# Model-based Human Performance Capture in Outdoor Scenes

A dissertation submitted towards the degree
Doctor of Engineering
of the Faculty of Mathematics and Computer Science
of Saarland University

by
Nadia Robertini

Saarbrücken
May, 2019

UNIVERSITÄT
DES
SAARLANDES

**Betreuender Hochschullehrer – Advisor:**
Prof. Dr.-Ing. Christian Theobalt

**Gutachter – Reviewers:**
Prof. Dr.-Ing. Christian Theobalt
Prof. Dr. Hans-Peter Seidel

**Dekan – Dean:**
Univ.-Prof. Dr. Sebastian Hack

**Kolloquium – Examination**

**Datum – Date:**
2019 - 05 - 21

**Vorsitzender – Chair:**
Prof. Dr. Philipp Slusallek

**Prüfer – Examiners:**
Prof. Dr.-Ing. Christian Theobalt
Prof. Dr. Hans-Peter Seidel

**Protokoll – Reporter:**
Dr. Rhaleb Zayer

To my beloved Alkhazur and Asalia

# Abstract

Technologies for motion and performance capture of real actors have enabled the creation of realistic-looking virtual humans through detail and deformation transfer at the cost of extensive manual work and sophisticated in-studio marker-based systems. This thesis pushes the boundaries of performance capture by proposing automatic algorithms for robust 3D skeleton and detailed surface tracking in less constrained multi-view outdoor scenarios. Contributions include new multi-layered human body representations designed for effective model-based time-consistent reconstruction in complex dynamic environments with varying illumination, from a set of vision cameras. We design dense surface refinement approaches to enable smooth silhouette-free model-to-image alignment, as well as coarse-to-fine tracking techniques to enable joint estimation of skeleton motion and fine-scale surface deformations in complicated scenarios. High-quality results attained on challenging application scenarios confirm the contributions and show great potential for the automatic creation of personalized 3D virtual humans.

# Kurzzusammenfassung

Technologien zur Bewegungs- und Verhaltenserfassung echter Schauspieler haben es ermöglicht, realistisch aussehende virtuelle Menschen zu erschaffen. Diese Technologien basieren auf Detail- und Deformationstransfers, entstanden aus umfangreicher Handarbeit und im Studio entwickelter, komplexer markerbasierter Systeme. Die vorliegende Arbeit sprengt die Grenzen der Verhaltenserfassung, indem sie automatische Algorithmen für ein robustes 3D-Skelett und detailliertes Oberflächen-Tracking in weniger eingeschränkten Outdoor-Szenarien mit Mehrfachansichten vorschlägt. Zu den Beiträgen gehören neue mehrschichtige Darstellungen des menschlichen Körpers, die für eine effektive modellbasierte und zeitlich konstante Rekonstruktion entwickelt wurden. Diese Darstellungen wurden in komplexen dynamischen Umgebungen mit unterschiedlicher Beleuchtung aus mehreren Vision-Kameras erzeugt. Die Ansätze zur Oberflächenverfeinerung einerseits ermöglichen eine ausgeglichene silhouettenfreie Ausrichtung des Modells an das Bild. Grobe bis feine Tracking-Techniken andererseits ermöglichen eine gemeinsame Schätzung von Skelettbewegungen und feinskaligen Oberflächendeformationen in komplexen Szenarien. Hochwertige Ergebnisse aus anspruchsvollen Anwendungsszenarien bestätigen die Beiträge und zeigen großes Potenzial für die automatische Erstellung von personalisierten virtuellen 3D-Menschen.

# Summary

Cutting-edge advances in technology in the digitization pipeline now allow the entertainment industry to create and animate realistic-looking 3D virtual humans with personalized appearance that look indistinguishable from real actors. The movie industry employs such technology for a range of purposes, from the complexity of live-action shots to the need for photo-realistic virtual characters that resemble an actor's appearance. Despite the great improvements introduced in available hardware and software, the creation of realistic characters still presents a number of restrictions and limitations. Since our visual perception is attuned to the appearance and behavior of human beings, the demand for realistic virtual look is particularly high.

To simplify the task of achieving realistic motion effects, producers make use of extensive motion capture as the main performance driver of virtual characters. Output of marker-based technologies is a set of sparse tracked feature points on the actor's body, also called markers, that are used to deform a 3D character, resulting in coarse-to-medium detailed deforming sequence reflecting the real action, through inverse kinematics (IK). The quality of the reconstruction strictly depends on the number of markers as well as on the quality and expressiveness of the actor's model employed and the re-targeting accuracy. High quality setups involving a large number of markers are capable to reconstruct personalized detailed surface deformations and expressions on top of the motion, capturing detailed performance (performance capture) at the cost of increased complexity in the capturing setup. Extensive manual work by experienced animators is still required to refine the reconstructions frame by frame accounting for missing details and setup limitations. The overall acquisition process, from the installation and calibration to cumbersome manual refinement, is extremely slow, expensive and most of the time infeasible, as it constrains performances in selected indoor studios with accurately calibrated settings (e. g. back-screens and calibrated illumination).

Motivated by the infeasibility of the current technologies for performance capture, recently the community started to look into markerless acquisition systems in less constrained environments, aiming at enabling performance capture of more natural and casual actions. Despite the efforts, none of the proposed approaches reconstruct reliable virtual performances in unconstrained outdoor setups, due to the strong assumptions on the capturing scene.

In this thesis, we take a leap forward and build some of the first model-based tracking methods to accurately and automatically capture the detailed surface deformation of performing actors in challenging outdoor environments with a possibly dynamic background and varying illumination conditions. Note that the given task is ambitious due to the overall uncertainty in the captured scene, e. g. complex and fast movements, wide clothing involving large non-rigid deformations, occlusions, lighting changes and dynamic background to name a few. The technical contributions of this thesis can be divided in four main areas: actor modeling, motion capture for skeleton configuration estimation, surface capture for personalized detailed deformation estimation and joint multi-layered skeleton-surface capture for robust coarse-to-fine reconstruction.

**Actor Modeling** Chapter 2 introduces a multi-layered mathematical body model of the actor, characterized by interconnected explicit and implicit coarse-to-fine representations. The body model can easily adapt to differently shaped and clothed humans and is well suited for motion and performance capture applications. The binding of coarse and fine layer is estimated automatically using a new efficient approach outlined in Section 2.3.2.2. The model is further augmented in Chapter 4 with a set of 3D colored Gaussian functions that approximate the surface geometry implicitly. This representation turns advantageous for surface refinement in unconstrained settings. Additional 3D Gaussian pairs -based surface approximation, discussed in Chapter 5, are used to align the model to the multi-view silhouettes without explicit background subtraction.

**Motion Capture** Chapter 3 outlines a method for robust motion capture in outdoor challenging scenarios with dynamic background and illumination changes. The approach generates illumination invariant input views and solves for the skeleton pose based on the segmentations iteratively. Moreover, this chapter presents a new adaptive combination of the lighting-invariant segmentation with CNN-based joint detectors used to increase the robustness to segmentation errors.

**Surface Capture** Chapter 4 presents a novel approach to recover true fine surface detail of deforming meshes of humans. Surface tracking is formulated as a global optimization problem of the densely deforming surface. The fine scale deformations for all mesh vertices, which maximize photo-temporal-consistency, can be effectively found by densely optimizing a new correspondences-free model-to-image consistency energy. Additionally, Chapter 5 outlines a method for unconstrained surface alignment to image contours without explicit background subtraction, based on a specifically designed implicit representation of the reprojected surface borders.

**Joint Skeleton-Surface Capture** Chapter 5 presents a new model-based method to accurately reconstruct coarse-to-fine human performances captured outdoors. The proposed approach fits a multi-layered actor model to unsegmented video frames, jointly optimizing for the coarse skeletal pose and the non-rigid surface shape simultaneously.

To summarize, this thesis presents several robust and automatic algorithms for tracking full body performances of a real actor from challenging outdoor multi-view videos. The proposed scientific contributions advance the state of the art in multi-view performance capture, thus improving the toolbox available for creating photo-realistic virtual 3D human avatars. Results attained on different application scenarios show great potential to automatize the digitization of photo-realistic virtual characters in movies and games, and possibly interactive applications in the near future. We hope that this thesis motivates the development of more advanced methods to digitize photo-realistic virtual human models of a quality comparable to the standard pipeline in post-production.

# Zusammenfassung

Modernste technologische Fortschritte im Digitalisierungsverfahren haben es der Unterhaltungsindustrie nun ermöglicht, realistisch aussehende virtuelle 3D-Menschen mit personalisiertem Erscheinungsbild zu erschaffen und zu animieren, die von echten Schauspielern nicht zu unterscheiden sind. Die Filmindustrie setzt solche Technologien für zahlreiche Aufgaben ein. Diese reichen von der Komplexität von Live-Action-Aufnahmen bis hin zur Notwendigkeit fotorealistischer virtueller Charaktere, die dem Aussehen eines Schauspielers ähneln. Trotz der großen Verbesserungen, die durch die verfügbare Hard- und Software eingeführt wurden, bringt die Erstellung realistischer Charaktere immer noch eine Reihe von Einschränkungen und Schwierigkeiten mit sich. Da unsere visuelle Wahrnehmung auf das Erscheinungsbild und Verhalten des Menschen abgestimmt ist, ist der Bedarf nach einem realistischen virtuellen Aussehen besonders hoch.

Zur Vereinfachung der Aufgabe, realistische Bewegungseffekte zu erzielen, nutzen Produzenten hauptsächlich die umfangreiche Bewegungserfassung als Leistungstreiber für virtuelle Charaktere. Die Ausgabe markerbasierter Technologien besteht aus wenigen nachverfolgten Merkmalpunkten auf dem Körper des Schauspielers, auch Marker genannt. Diese werden verwendet, um einen 3D-Charakter zu verformen, was zu einer grob- bis mittel-detaillierten Verformungssequenz führt, welche die reale Bewegung durch die Inverse Kinematik (IK) widerspiegelt. Die Qualität der Rekonstruktion hängt stark von der Anzahl der Marker sowie von der Qualität und dem Ausdruck des verwendeten Modells des Schauspielers sowie von der Genauigkeit des Re-Targeting ab. Hochwertige Setups mit einer großen Anzahl von Markern sind in der Lage, personalisierte, detaillierte Oberflächenverformungen und -ausdrücke über der Bewegung zu rekonstruieren und detaillierte Bewegungen auf Kosten einer erhöhten Komplexität im Aufzeichnungssetup zu erfassen (Performance Capture). Umfangreiche Handarbeit durch erfahrene Animatoren ist nach wie vor erforderlich, um die Rekonstruktionen Bild für Bild unter Berücksichtigung fehlender Details und Setup-Einschränkungen zu verfeinern. Der gesamte Erfassungsprozess, von der Installation und Kalibrierung bis hin zur umständlichen manuellen Verfeinerung, ist extrem langwierig, teuer und meist nicht realisierbar, da er die Durchführung auf ausgewählte Indoor-Studios mit genau kalibrierten Einstellungen einschränkt (z.B. Backscreens und kalibrierte Beleuchtung).

Motiviert durch die Unzulänglichkeit der aktuellen Technologien zur Performance Capture, wurde vor Kurzem begonnen, sich mit markerlosen Erfassungssystemen in weniger eingeschränkten Umgebungen zu befassen, um die Erfassung natürlicher und zufälliger Bewegungen zu ermöglichen. Trotz der Bemühungen rekonstruiert keiner der vorgeschlagenen Ansätze zuverlässige virtuelle Ergebnisse in uneingeschränkten Outdoor-Setups, da die Schätzungen der Aufnahmeszene stark sind.

Diese Arbeit bringt einen großen Fortschritt durch den Aufbau erster modellbasierter Tracking-Methoden, um die detaillierte Oberflächenverformung ausführender Schauspieler in anspruchsvollen Außenumgebungen mit einem möglicherweise dynamischen Hintergrund und unterschiedlichen Beleuchtungsbedingungen präzise und automatisch zu erfassen. Die gegebene Aufgabe ist sehr anspruchsvoll, aufgrund der allgemeinen Unklarheit in der aufgenommenen Szene durch z.B. kom-

plexe und schnelle Bewegungen, dynamische Hintergründe, weite Kleidung mit großen, unstarren Deformationen, Okklusionen und Lichtveränderungen. Die technischen Beiträge dieser Arbeit lassen sich in vier Hauptbereiche unterteilen: Modellierung des Schauspielers, Bewegungserfassung zur Schätzung der Skelettkonfiguration, Oberflächenerfassung zur personalisierten detaillierten Deformationsschätzung und gemeinsame mehrschichtige Skelett-Oberflächenerfassung für eine robuste Grob-/Feinrekonstruktion.

**Modellierung des Schauspielers** Kapitel 2 stellt ein vielschichtiges mathematisches Körpermodell des Schauspielers vor, das durch vernetzte explizite und implizite grobe bis feine Darstellungen charakterisiert ist. Das Körpermodell kann sich leicht an unterschiedlich geformte und bekleidete Menschen anpassen und eignet sich gut für Anwendungen zur Bewegungs- und Verhaltenserfassung. Die Bindung von Grob- und Feinschicht wird automatisch mit einem neuen effizienten Ansatz geschätzt, der in Abschnitt 2.3.2.2 beschrieben ist. Das Modell wird in Kapitel 4 um eine Reihe farbiger dreidimensionaler Gaußscher Funktionen erweitert, die sich der Oberflächengeometrie implizit annähern. Diese Darstellung wird für detailreichere Oberflächen in uneingeschränkten Umgebungen vorteilhaft. Zusätzliche Oberflächenannäherung basierend auf 3D-Gaußschen Paaren, wie in Kapitel 5 erläutert, wird verwendet, um das Modell ohne explizite Hintergrundsubtraktion an die Silhouetten mit mehreren Ansichten anzupassen.

**Bewegungserfassung** Kapitel 3 beschreibt eine Methode zur robusten Bewegungserfassung in anspruchsvollen Outdoor-Szenarien mit dynamischen Hintergrund- und Beleuchtungsänderungen. Der Ansatz erzeugt beleuchtungsinvariante Eingangsansichten und löst diese iterativ für die Skelettposition, basierend auf den Segmentierungen. Darüber hinaus stellt dieses Kapitel eine neue adaptive Kombination der beleuchtungsinvarianten Segmentierung mit CNN-basierten Gelenkdetektoren vor, um die Robustheit gegenüber Segmentierungsfehlern zu erhöhen.

**Oberflächenerfassung** Kapitel 4 stellt einen neuartigen Ansatz zur Verfügung, um echte feine Oberflächendetails des Menschen durch deformierende Netze zu erhalten. Die Oberflächenverfolgung ist als globales Optimierungsproblem der dicht verformten Oberfläche formuliert. Die feinskaligen Verformungen für alle Netzknoten, die die foto-zeitliche Konsistenz maximieren, können effektiv gefunden werden, indem eine neue korrespondenzfreie Modell-Bild-Konsistenzenergie dicht optimiert wird. Zusätzlich wird in Kapitel 5 ein Verfahren zur uneingeschränkten Ausrichtung der Oberfläche auf Bildkonturen ohne expliziten Hintergrundabzug skizziert, basierend auf einer speziell entwickelten impliziten Darstellung der re-projizierten Oberflächenränder.

**Gemeinsame Skelett-Oberflächenerfassung** Kapitel 5 präsentiert eine neue modellbasierte Methode zur genauen Rekonstruktion grober bis feiner menschlicher Bewegungen, die im Freien aufgenommen wurden. Der vorgeschlagene Ansatz passt ein mehrschichtiges Modell des Akteurs an unsegmentierte Videobilder an und optimiert gemeinsam und gleichzeitig für die grobe Skelettposition und die nicht starre Oberflächenform.

Zusammenfassend lässt sich sagen, dass diese Arbeit mehrere robuste und automatische Algorithmen für die Nachverfolgung des Ganzkörperverhaltens echter Schauspieler aus anspruchsvollen Outdoor-Videos mit mehreren Ansichten vorstellt. Die vorgeschlagenen wissenschaftlichen Beiträge bringen den Stand der Technik in der Multiview-Verhaltenserfassung voran und verbessern damit die zur Verfügung stehenden Mittel für die Erstellung fotorealistischer virtueller menschlicher 3D-Avatare. Die Ergebnisse verschiedener Anwendungsszenarien zeigen ein großes Potenzial, die Digitalisierung fotorealistischer virtueller Charaktere in Filmen und Spielen sowie möglicherweise interaktiver Anwendungen in naher Zukunft zu automatisieren. Wir hoffen, dass diese Arbeit zur Entwicklung fortgeschrittenerer Methoden zur Digitalisierung fotorealistischer virtueller Menschmodelle motiviert, die von vergleichbarer Qualität wie die der Standard-Pipeline in der Postproduktion sind.

# Acknowledgments

I would like to thank the fellow doctoral students of the Computer Graphics department at the Max Planck Institute and in particular the members of the Graphics, Vision and Video group for their feedback, cooperation and of course friendship. In particular I would like to express my gratitude to the following people: Edilson De Aguiar, Helge Rhodin, Dan Casas, Florian Bernard, Weipeng Xu, Dushyant Mehta, Edgar Tretschk, Abhimitra Meka, Ayush Tewari, Hyeongwoo Kim, Franziska Mueller, Michael Zollhoefer, Srinath Sridhar, Pablo Garrido, Oleksandr Sotnychenko, Ahmed Elhayek, Chenglei Wu, Thomas Helten, Carsten Stoll, Nils Hasler, Christian Richardt, Kwang In Kim, James Tompkin, Jozef Hladky, Thomas Leimkuehler, Bertram Somieski, Elena Arabadzhiyska. A big thank you to the secretaries Sabine Budde, Ellen Fries and Hanna Loger, for helping me out in a number of matters from the settling-in to the organization of the final ceremony. I am also grateful to the members of my committee and in particular to my advisor, Prof. Christian Theobalt, for his patience and support in overcoming numerous obstacles I have been facing through my research.

Last but not least, I would like to thank my family for always loving me “no matter what”, in particular my husband Alkhazur for his love and continuous support throughout writing this thesis. This thesis is dedicated to him and to our little treasure, Asalia.

# Contents

# Chapter 1

# Introduction

## 1.1 Motivation

Modeling and animating virtual humans has gained a lot of interest in the past few years. Realistic virtual humans and humanoids play a fundamental role in entertainment and artistic productions. In the movie industry, 3D characters substitute real actors in various scenes and tasks, see some examples in Figure 1.1(b,g,i,k). Personalized virtual selves or avatars enable immersive social interactions in virtual and augmented reality, e. g. video editing and interactive applications. Recently-developed *virtual mirrors* offer customers the possibility to try on various clothing or other accessories (e. g. glasses) in real-time, see Figure 1.1(h). Understanding human behavior and physiology is an important task in surveillance, sports medicine, design Realistic rendering of the human body and simulation of his natural movements (including gestures, facial expressions and so on) are required to enable the aforementioned applications.

During the past decades there has been impressive progress in the creation of entirely computer-generated virtual characters. Despite the great improvements introduced in available hardware and software, the creation of realistic characters is still an extremely challenging task with many physical limitations. Since our visual perception is attuned to the appearance and behavior of human beings, the demand for a realistic virtual look is particularly high. Such realistic appearances can be recreated only with a high degree of realistic rendering on top of a convincingly detailed three-dimensional model, which includes complex fine-scale features, such as hairs, skin wrinkles and fabric folds. Given the current level of technology, realistically animating virtual characters also proves a fundamental and challenging task because it is still difficult to capture subtle skin and cloth deformations. While most human expressiveness happens in the facial area, plausible body motions and interactions with the surrounding scene are crucial and serve to complete the picture. In consequence, convincing simulation of natural human motion typically involves tedious manual work by experienced animators.

To simplify the task to achieve realistic motion effects, producers typically make use of extensive motion capture as the main performance driver of virtual characters. *Motion capture*, sometimes referred as *mocap*, consists of recording the motion of real human actors and transfer the obtained data to virtual characters, such that they move similarly (re-targeting). The technique was introduced in the early 1970s initially as an optical analysis tool in bio-mechanics research [157]. Later, its use was expanded to wider fields of application and was recently being adopted in computer animation

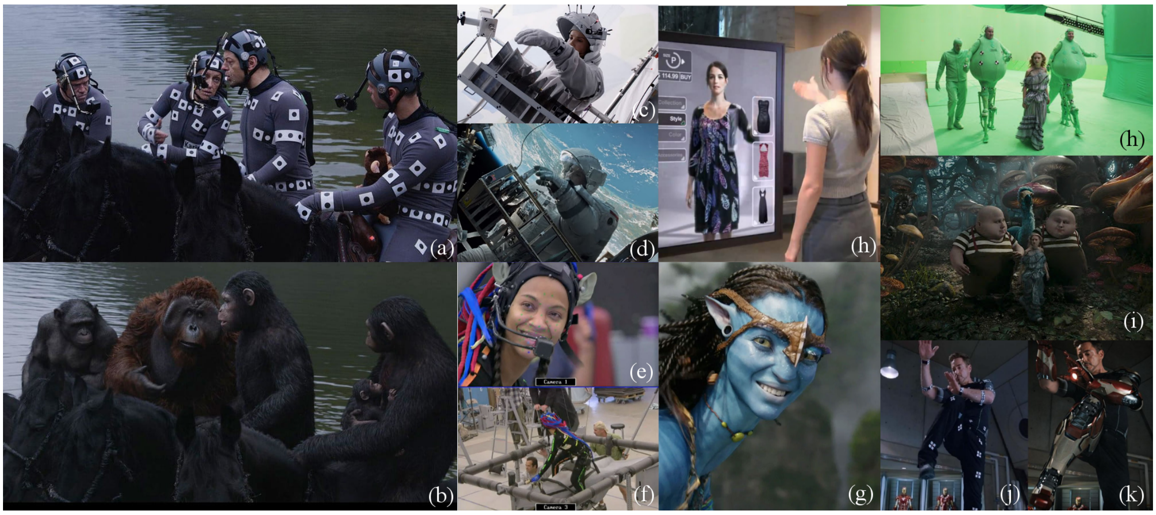


Figure 1.1: Examples showing the use of photo-realistic virtual humans in feature films and interactive applications. (a) "Planet of the Apes" featuring real actors with motion capture suits and (b) the corresponding rendered result where only the background is preserved. (c) "Gravity" actor in action and (d) the final rendering with additional features for the space suit and the background. (e) "Avatar" real actors with complex motion capture suites complete with hairs, (f) interactive infrastructure to guide the action and (g) a rendering result. (h) "Alice in wonderland" during capture and (i) after replacing the entire background with both static and moving objects, while the main actor is completely preserved. (j) "Ironman" real actor with motion capture suite to only estimate the rigid armor location, while the actor face is preserved at rendering time (k).

for television, cinema and video-games. Mocap techniques focus on tracking the rigid skeletal motion of human beings, i. e. the coarse bone-driven deformation of the actor's body, ignoring non-rigid detailed deformation, e. g. folds and creases on clothing.

The capturing setup typically involves the use of (active) distinguishable markers, strategically placed on the human actor's body. Complex optical systems, such as a collection of calibrated cameras placed around the actor, or non-optical systems, are used to track the markers' position in time. The output is a sparse set of tracked feature points that serve to animate virtual characters or differently shaped humanoids (e. g. cartoons) similarly through re-targeting. While the captured features give a good baseline of the overall motion, it is the animator's task to refine the quality of the deformations further to produce realistic effects. Virtual character modeling is also a fundamental and delicate task that determines tracking accuracy and fidelity before the actual deformation tracking and transfer. Models of characters are typically created by experienced 3D artists. In summary, three main steps are required for mocap: actor modeling, motion tracking and transfer, and (manual) refinement to introduce additional deformation detail.

When personalized detailed surface deformations and expressions are captured on top of the motion, the whole process is often referred to as *performance capture*. Performance capture of an actor's motions with full fidelity, as is required for avatars, typically requires more complex and finer-scale movements on top of skeleton-driven motion. These finer-scale movements include muscle bulging, cloth deformations and subtle facial expressions. These additional fine-scale details cannot be simply captured in their full extent with a few sparse markers. Increasing the number of markers results in higher setup constraints and costs. Due to the increased capture detail, the complexity of the performance capture pipeline is sensibly increased in all its steps, compared to that of mocap. On top of that, the performance capture pipeline requires an additional detailed surface tracking step, before the refinement. Capturing the detailed surface deformation is either performed as a separate step,

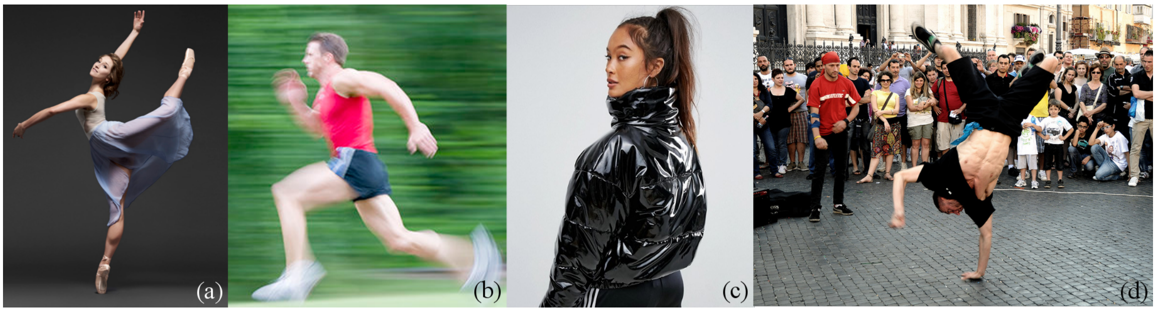


Figure 1.2: Challenges in human performance capture. (a) large non-rigid cloth deformations, (b) fast movements, (c) complex clothing reflectance, (d) cluttered outdoor capturing conditions with dynamic background and possible illumination changes.

after the coarse skeleton-driven character motion has been inferred, or in a joint skeleton-surface tracking fashion. This thesis proposes solutions for both strategies.

The overall capture process both for motion and performance capture, comprising of installation, calibration, recording and refining, is extremely slow, expensive and most of the time infeasible, as it constrains recordings in selected indoor studios with accurately calibrated settings (e. g. green-screens and calibrated illumination) to allow for the acquisition sensors to detect and track the markers. Less controlled (e. g. outdoor) capture settings easily prevent accurate markers acquisition, due to possible dynamic interfering (moving) objects as well as changing weather conditions, causing uncontrolled (abrupt) marker appearance variations. On top of such forbidding capture environment, actors are required to respect the (narrow) space limits and wear a marker-suit, which in turn makes it impossible to capture spontaneous acts or e. g. some long-distance sports.

Aiming at reducing the complexity of typical capture setups, the community has recently started to look into markerless acquisition systems in less constrained environments, thus enabling the performance capture of more natural and casual actions. While capturing the global skeleton motion has been demonstrated outdoors, assuming steady weather conditions [21, 63, 74, 24, 164, 154, 51], markerless performance capture of detailed surface motion has only succeeded in controlled studios with carefully calibrated light and back-screens [20, 40, 64, 162, 180]. In this thesis, we take a leap forward and build some of the first tracking methods that estimate surface deformations from input multi-view footage captured in outdoor scenes. Specifically, we introduce a new multi-layer model of the human body to fulfill our goal, as well as building full end-to-end optimization-based frameworks that address the performance capture challenges robustly. We test our approaches on different challenging outdoor scenarios including scenes with dynamic background and varying illumination conditions.

## 1.2 Overview

Given a multi-view video of an actor recorded in uncontrolled environment with unknown illumination settings, the goal of this thesis is to develop robust, accurate, and fully-automatic model-based methods to motion and performance capture the full temporally-coherent 3D body action, including fast movements and non-rigid deformations of the surface.

There exist a number of inherent challenges in achieving the goal stated above. To simplify the problem on the recording side, this thesis assumes the input cameras to be static and calibrated, such that their global position and orientation as well as intrinsic parameters (e. g. focal length) are known. Knowing where the cameras are looking at only minimally simplifies the task. The quality of the reconstructions are still limited by the recording quality, i. e. the pixel resolution, capturing frame rate and the resolution of the actor in the frames. Similar to other capture methods, the approaches in this thesis also assume a constrained recording space of few square meters with cameras equally distributed around, aiming at covering most of the action from different sides.

Apart from the acquisition quality, this thesis has to deal with challenges imposed by the captured scene itself. In uncalibrated outdoor scenarios, often scenes exhibit dynamic uncontrolled background, occlusions, illumination changes to name a few. Discriminating the actor (foreground) from the rest of the scene is extremely challenging in presence of unpredictably moving background and dynamic occlusions, see an example in Figure 1.1(d). Illumination changes, such as global light changes or shadows, result in (sudden) changes to the actor's visual appearance, causing photometric-based tracking approaches to easily fail.

Further on, actor actions might involve fast movements, resulting in motion blur and missing information, see Figure 1.1(b). Complex movements such as crossing arms cause self-occlusions for a few frames that are typically hard to track. This thesis does not explicitly set any constraint on the actor's clothing; however, tracking accuracy highly depends on the cloth's tightness to the actor's body and the stiffness, which affects the deformation behavior and complexity. Wide pieces of apparel typically involve large complex non-rigid deformations that are hard to track accurately, see an example in Figure 1.1(a).

Naturally, view-dependent clothing reflectance behavior (i. e. non-Lambertian), such as e. g. the shiny jacket in Figure 1.1(c), may cause the same surface locations to appear differently from different sides. Feature-less garments, e. g. uniformly colored shirts or pantsetc. , also lead to ambiguities in tracking. Another fundamental difficulty is given by possible shape changes due to removal of jackets or hats, for example. These require synthesizing topological changes in the initial body model by re-defining its shape and connections, unless already present. To cope with the latter, this thesis assumes no (large) topological changes happen during the recordings.

To robustly guide the tracking, especially designed models of the human actor are required to enable effective actor-to-image similarity measures and deformation. The tracking accuracy and the computational speed are primarily determined by the quality and storage efficiency of the body model. To enable the capture of differently shaped and dressed humans, a specifically designed body model has to be created for each new capture sequence, which is typically a cumbersome task. This thesis contributes with a new multi-layer human body model composed of skeleton, implicit volumetric and explicit mesh representation, automatically linked among each other using a new fast skinning-based approach, that can easily adapt to new human shapes and is best suited for motion and performance capture applications.

The remaining technical contributions of this thesis address a subset of the stated challenges at a time, by addressing several algorithmic core problems that need to be approached for performance capture in less constrained scenes. The second main contribution, after the actor modeling, is one of the first methods to accurately capture the actor's skeleton motion in presence of illumination changes and non-Lambertian clothing reflectance behavior. The third main contribution is a new correspondences-free approach to performance capture surface details that works outdoor without the need for background/foreground segmentation. The proposed method copes with (self-)occlusions, wide apparel and scenes that show people in clothing with only few features. Finally, we introduce a new model-to-image method that jointly optimizes for the coarse skeletal motion and the non-rigid surface deformation simultaneously in outdoor scenarios with possible occlusions and varying backgrounds. This thesis also demonstrates methods that can deal with fast and complex motions better than previous approaches.

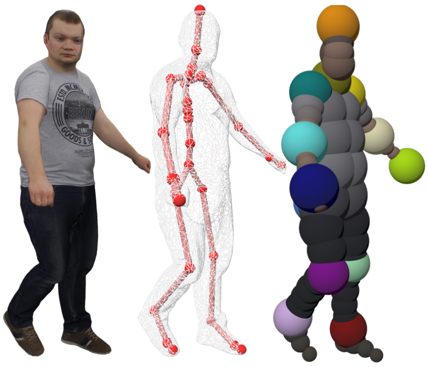
## 1.3 Structure of the Thesis

This thesis is divided into six chapters covering the main technical contributions in the areas of human motion and performance capture:

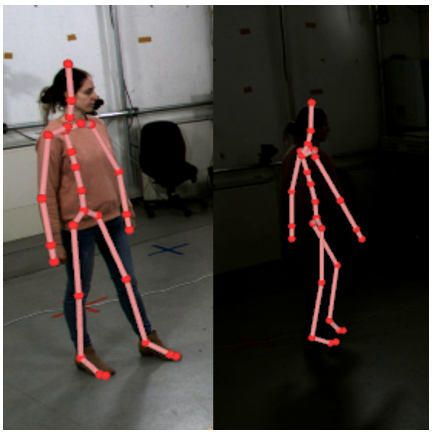
- Chapter 1 introduces the topic of this thesis, states the goals, outlines the structure of exposition, summarizes the technical chapters, and states the main technical contributions.
- Chapter 2 presents the required background for the research presented in this thesis. Among the contributions, it introduces a new multi-layer human body model of the actor, especially designed for easy adaptation to different body shapes, which is optimized for motion and performance capture applications.
- Chapters 3–5 present the main technical contributions, going from new methods to robustly track the human body motion to the reconstruction of detailed surface deformation in challenging scenarios. Chapter 3 proposes an effective solution to the motion capture problem in challenging illumination-changing scenarios. Chapter 4 is focused on surface detail capture given an initial coarse human body deformation sequence. Chapter 5 describes a complete solution to the problem of model-based human performance capture that jointly captures the skeletal motion and the detailed surface deformation in outdoor scenes. Each chapter gives an introduction and outline of the proposed solution compared to previous approaches. Results and comparisons to related work are discussed at the end of each chapter. Section 1.3.1 describes each chapter in more detail.
- Chapter 6 summarizes the core contributions and results achieved so far, and it briefly discusses unexplored future challenges.

The following section gives a more detailed overview of the technical chapters of this thesis.

### 1.3.1 Summary of Technical Chapters



**Chapter 2** introduces the mathematical body model of the human used throughout this thesis. In particular, it discusses the three typical model layer components that form the human body, namely skeleton, volumetric representation and triangular mesh surface. This chapter additionally discusses in detail an automatic approach for estimating mesh skinning weights using the underlying implicit volumetric layer. This method has been first introduced as part of the pipeline in the co-authored paper [142]. The resulting skinned mesh representation combined with underlying skeleton and Gaussian-based volumetric representation has been used extensively in the follow-up papers [146, 144].



**Chapter 3** presents a new solution to the problem of capturing human body motion under changing lighting conditions in a multi-view setup [144]. The key idea is to provide to the tracking approach robustly and dynamically segmented input frames, invariant to the scene lighting. Illumination-invariant frames are obtained by solving time-varying segmentation problems that use frame- and view-dependent appearance costs. Moreover, this chapter presents a new adaptive combination of the lighting-invariant segmentation with CNN-based joint detectors used to increase the robustness to segmentation errors. Experimental results demonstrate the capabilities of this approach to handle global light changes and shadows, outperforming existing works.

**Chapter 4** presents a novel approach to recover true fine surface detail of deforming meshes of humans [147, 145]. Surface refinement is formulated as a global optimization problem of the densely deforming surface, implicitly represented using a set of Gaussian functions placed at each vertex location. Similarly, the originally captured multi-view images are represented with a set of 2D Gaussians. The fine scale deformations for all mesh vertices, which maximize photo-temporal-consistency, can be effectively found by densely optimizing a new correspondences-free model-to-image consistency energy. We qualitatively and quantitatively demonstrate that our technique successfully reproduces finer detail than the input baseline geometry.

**Chapter 5** presents a new model-based method to accurately reconstruct human performances captured in less controlled outdoor settings [146]. Starting from a template of the actor model, a new unified implicit representation for both, articulated skeleton tracking and nonrigid surface shape refinement are introduced. The proposed approach fits the template to unsegmented video frames in two stages: first, the coarse skeletal pose is estimated, and subsequently nonrigid surface shape and body pose are jointly refined. The approach utilizes a new representation of the human body surface based on a combination of 3D Gaussians, designed to align the projected model with likely silhouette contours without explicit segmentation or edge detection. The obtained reconstructions are shown to outperform existing methods.

## 1.4 List of Contributions

In this section, we provide a more detailed list of technical contributions that enable the methods described above.

The main contributions of Chapter 2 are:

- Definition of a new multi-layered human body model composed of interconnected components, namely skeleton, volumetric shape representation using Sums-of-Gaussian 3D functions, and a triangular mesh.
- A method for automatically estimating mesh skinning weights using the underlying volumetric layer.

The main contributions of Chapter 3 are:

- A formulation of the time-varying frame segmentation problem that uses frame- and view-dependent appearance costs in order to obtain lighting-invariant representations of the individual frames.
- An adaptive combination of an abstract intermediate image representation with CNN-based joint detectors for robust pose estimation.
- The integration of the combined illumination-invariant segmentation with CNN-based detection into a robust model-based tracker.

The main contributions of Chapter 4 are:

- A new shape representation that models the mesh surface with a dense collection of 3D Gaussian functions centered at each vertex.
- A formulation of dense photo-consistency-based surface refinement on the basis of this new model.
- The integration of the new surface Gaussian based refinement method into a model-based performance capture method.

The main contributions of Chapter 5 are:

- A new unified implicit formulation for both, articulated skeleton tracking and non-rigid surface shape refinement.
- The formulation of a new surface refinement method to align the projected model to likely silhouette contours without explicit segmentation or edge detection.

## 1.5 List of Publications

The work presented in this thesis mainly encompasses four peer-reviewed scientific publications, published at conferences and journals in the fields of computer graphics and vision:

- **Nadia Robertini**, Edilson De Aguiar, Thomas Helten, Christian Theobalt. “Efficient Multi-view Performance Capture of Fine-Scale Surface Detail”. *Proceedings of the Second International Conference on 3D Vision (3DV)*, 2014 [147].
- **Nadia Robertini**, Dan Casas, Helge Rhodin, Hans-Peter Seidel, Christian Theobalt. “Model-based Outdoor Performance Capture”. *Proceedings of the Second International Conference on 3D Vision (3DV)*, 2016 [146].
- **Nadia Robertini**, Dan Casas, Edilson De Aguiar, Christian Theobalt. “Multi-view Performance Capture of Surface Details”. *International Journal of Computer Vision (IJCV)*, First Online. doi 10.1007/s11263-016-0979-1, 2017 [145].
- **Nadia Robertini**, Florian Bernard, Weipeng Xu, Christian Theobalt. “Illumination-invariant Robust Multiview 3D Human Motion Capture”. *Winter Conference on Applications of Computer Vision (WACV)*, 2018 [144].

Additionally Chapter 2 discusses in detail a new approach for estimating mesh skinning weights, first introduced in the co-authored paper below as part of the reconstruction pipeline. Details on the core techniques outlined in the paper are given in Chapter 6.

- Helge Rhodin, **Nadia Robertini**, Dan Casas, Christian Richardt, Hans-Peter Seidel, Christian Theobalt. “General Automatic Human Shape and Motion Capture Using Volumetric Contour Cues”. *Proceedings of the 2016 European Conference on Computer Vision (ECCV)*, 2016 [142].

Chapter 2 briefly discusses another co-authored paper focused on occlusions handling during model-based scene reconstruction:

- Helge Rhodin, **Nadia Robertini**, Christian Richardt, Hans-Peter Seidel, Christian Theobalt. “A Versatile Scene Model with Differentiable Visibility Applied to Generative Pose Estimation”. *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, 2015 [143].

# Chapter 2

# Background

## 2.1 Introduction

This chapter presents the required background for the research presented in this thesis. We start by introducing our multi-layer body model in Section 2.2 used throughout this thesis, especially designed for motion and performance capture applications. The model is formed by combining a personalized, detailed geometric model of the human body surface with a kinematic skeleton, augmented with an implicit volumetric representation, automatically linked among each other with a new approach, outlined in Section 2.3. Our human body representation can easily adapt to differently shaped and dressed humans and can be effectively employed for skeleton motion capture (mocap), as explained in Section 2.4. The mocap approach presented here serves as basis for the advanced performance capture techniques developed in this thesis.

## 2.2 Body Model

Scene modeling is a fundamental and cumbersome step in motion and performance capture applications. Models of the human actor are needed to enable effective actor-to-image similarity measures that robustly guide the tracking. The type and the resolution of the chosen actor representation primarily determine the computational speed at tracking time. It also determines robustness to outliers and tracking accuracy. Highly detailed surface mesh based representations [9, 64] enable fine scale model-to-image fitting in the number of available mesh vertices. However, determining optimal model fits with such a high number of independent model parameters, namely the 3D position of each vertex, is computationally demanding and easily becomes under-constrained. Simplifications of the actor model, either through volumetric [50, 164, 143] or implicit surface representation [137, 80] have shown impressive results in skeleton tracking thanks to their highly reduced space of unknown and elegant mathematical formulations. However, simplifying the surface representation typically reduces the reconstruction fidelity as it restricts the amount and quality of the captured surface details.

Aiming at fusing the beauty of an elegant mathematical representation without sacrificing surface detail, some methods use a combination of coarse, e. g. volumetric approximations, and fine-scale actor models, e. g. detailed surface meshes, to enable coarse-to-fine tracking [40]. In these approaches,

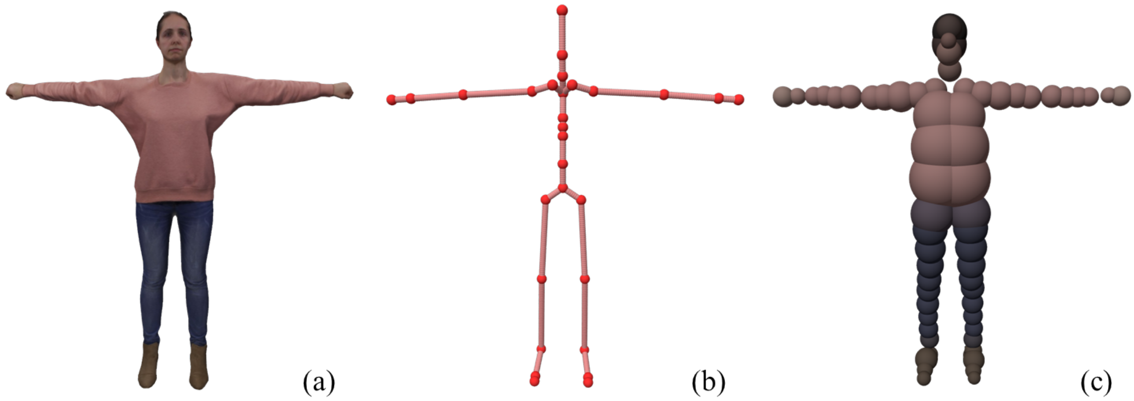


**Figure 2.1:** Visualization of the three layers of our body model in *natural pose* or *T-pose*. (a) Mesh with texture, (b) underlying skeleton structure, (c) underlying volumetric Gaussian-based representation.

the actor mesh is skinned to the underlying skeleton structure and deformed along with the estimated coarse pose. Robust surface refinement is then applied in a second step taking advantage of improved initial surface guesses. To improve tracking convergence, many methods proposed different ways to couple the articulated skeleton with the surface [88, 7, 136, 109, 159, 110, 14]. Most commonly used are linear blend shape models [5] and variants of the SCAPE model [7], both learned from databases of human scans. SCAPE is a popular data-driven human model able to synthesize realistic non-rigid surface deformations, e. g. muscle bulging, as a function of pose of the articulated skeleton and body characteristics, like mass and weight. However, these suggested parameterizations are not general enough to represent surface details such as exact wrinkles and general apparel deformation.

Aiming at combining the beauty of volumetric body representation with detailed surface geometry, we choose a multi-layer body model and make use of it throughout this thesis. The coarse layer is inspired by the model proposed by Stoll et al. [164] and it consists of a skeleton structure with volumetric Gaussian primitives rigidly attached to the underlying bones, see Figure 2.1. This representation is shown to be advantageous for human mocap (Section 2.4) and can easily adapt to differently shaped and clothed humans (Section 2.3). Our fine layer is a detailed rigged mesh obtained either from 3D full body scans or through image-based techniques. In the remainder of this section, we describe each layer of our proposed human body model as well as techniques to obtain personalized static reconstructions for each layer, compared to related models and approaches used in the literature.

### 2.2.1 Geometric Layer

In our three-layered actor representation, the geometric model is used to represent the last and most detailed layer. The geometry layer shall enable to describe the exact shape and appearance of the actor surface, in terms of points (vertices), connected polygons (typically triangles) and associated colors (vertex colors and textures), up to the tiniest surface detail depending on resolution, e. g. skin wrinkles. The amount of explained detail highly depends on the setup used for geometry acquisition. The actor body can be effectively scanned with a 3D body scanner to obtain a highly personalized reconstruction, see Section 2.2.1.2.

Generative image-based scanning approaches, described in Section 2.2.1.3, have been successfully employed to estimate accurate approximations of the human geometry from multi-view photographs,

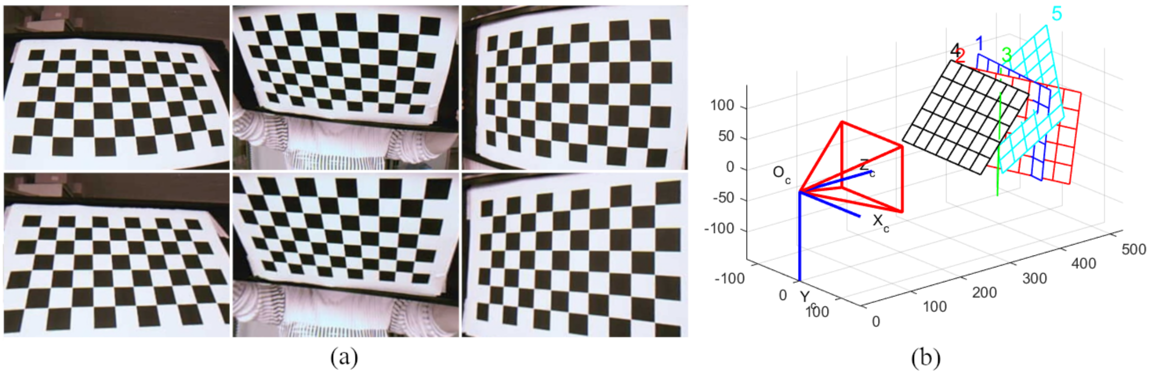


**Figure 2.2:** Example of single camera calibration images using a checkerboard. (a) The top row visualizes three input photographs of the checkerboard in different orientations, with the corresponding undistorted images in the bottom row. (b) Example of resulting extrinsic calibration showing the estimated camera location and orientation w. r. t. the 3D checkerboard.

when 3D full body scanners are unavailable [162, 60]. These kind of methods are extensively used in motion and performance capture applications as well as in this thesis.

In the remaining sections, we describe theory and methodologies for multi-view calibration of static vision cameras (Section 2.2.1.1), which is fundamental both for accurate static scanning and dynamic motion and performance capture. Next, we outline body scanning techniques as well as image-based scanning methods proposed in the literature to obtain static geometry and appearance reconstructions.

#### 2.2.1.1 Multi-view Calibration

This section describes theory and methodologies for *Multi-view Calibration (MVC)* of static cameras. The technique described is extensively used in this work as a preprocessing step prior to motion and performance capture. We start by outlining our parametric camera model to later introduce methods for geometry and color calibration of multi-view capture setups.

**Parametric Camera Model** A camera is a device that projects 3D world points onto the 2D image plane. Understanding how this imaging process takes place, or, in other words, what is the relation between world points and camera pixels, is essential for vision-based reconstruction and tracking, and is the primary task of calibration. Calibration aims at estimating the camera properties, such as, e. g. , focal length, lens distortion, location w. r. t. the world and so on, that affect its imaging process. To simplify the task of finding these quantities, we use a parametric camera model of the real camera and focus on estimating the (few) characterizing parameters. Most of the related work calibration approaches model the camera system as a *pinhole camera* with additional lens distortion parameters [195]. The pinhole camera model can be effectively used to approximate real-camera mapping from a 3D scene to a 2D image using a perspective projection. In such a perspective model, a 2D pixel point $p^h := [u\ v\ 1]^T$ in homogeneous coordinates is obtained from the corresponding 3D world point $\hat{p}^h := [x\ y\ z\ 1]^T$ as:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} := \begin{bmatrix} f_x \cdot m_x & \gamma & c_x \\ 0 & f_y \cdot m_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2.1}$$

where $(f_x, f_y)$ is the camera focal length in pixels, $(m_x, m_y)$ is a scale factor relating pixels to distance, $\gamma$ represents the skew coefficient between the $x$ and the $y$ axis and is often 0, $(c_x, c_y)$ represent the principal point, which would be ideally in the center of the image, $r_{..}$ are the rotation components of the global camera rotation matrix $R'_c$, whereas $t_.$ are the translation components of the global camera translation matrix $t'_c$. The matrix containing internal camera parameters is often called *intrinsic* matrix $K_c$, while the matrix describing the global position and orientation of the camera in the world is also called *extrinsic* matrix, $T'_c := [R'_c t'_c]$. The product of the intrinsic and extrinsic matrix is often called *camera projection matrix* $P_c := K_c \times T'_c$. This notation is used throughout this work.

Nonlinear parameters such as lens distortion cannot be included in the linear camera model described by the intrinsic parameter matrix. Radial distortion occurs when light rays bend more near the edges of a lens than they do at its optical center and are typically modeled by the radial distortion parameters $\rho_{1,2,3}$ as:

$$\begin{aligned}\tilde{u}_{\text{distorted}} &= \tilde{u}(1+\rho_1\cdot(\tilde{u}^2+\tilde{v}^2)^2+\rho_2\cdot(\tilde{u}^2+\tilde{v}^2)^4+\rho_3\cdot(\tilde{u}^2+\tilde{v}^2)^6)\\ \tilde{v}_{\text{distorted}} &= \tilde{v}(1+\rho_1\cdot(\tilde{u}^2+\tilde{v}^2)^2+\rho_2\cdot(\tilde{u}^2+\tilde{v}^2)^4+\rho_3\cdot(\tilde{u}^2+\tilde{v}^2)^6)\end{aligned} \tag{2.2}$$

Tangential distortion occurring when the lens and the image plane are not parallel, is modeled by the tangential distortion coefficients $\tau_{1,2}$ as follows:

$$\begin{aligned}\tilde{u}_{\text{distorted}} &= \tilde{u}+(2\cdot\tau_1\cdot\tilde{u}\cdot\tilde{v}\cdot+\tau_2\cdot((\tilde{u}^2+\tilde{v}^2)^2+2\cdot\tilde{u}^2))\\ \tilde{v}_{\text{distorted}} &= \tilde{v}+(\tau_1\cdot((\tilde{u}^2+\tilde{v}^2)^2+2\cdot\tilde{v}^2)+2\cdot\tau_2\cdot\tilde{u}\cdot\tilde{v})\end{aligned} \tag{2.3}$$

where $\tilde{u}$ and $\tilde{v}$ are respectively the undistorted pixel locations in normalized image coordinates, computed as:

$$\tilde{u} := \frac{u-c_x}{f_x}, \tilde{v} := \frac{v-c_y}{f_y}. \tag{2.4}$$

Once radial and tangential distortion parameters are estimated from calibration, we typically undistort all camera views once offline and work on the rectified images.

**Geometric Calibration** The most widely used approaches to geometrically calibrate a camera, i. e. estimate intrinsic and extrinsic as well as lens distortion parameters, also called *geometric calibration*, is to analyze several photographs of a set of known control points on a planar object, i. e. a *calibration target*, in focus in different locations and orientations [119, 201, 72]. Typically a checkerboard with white and black well-recognizable squares of known dimensions is employed. See an example of calibration sequence for a single camera setup in Figure 2.2.

Calibrating a static multi-view camera setup involves mainly three steps: automatic detection of the checkerboard corners, initial estimation of the camera parameters and final refinement. Image processing techniques, e. g. segmentation and edge detection, are employed to identify the checkerboard corners in all the multi-view photographs. Initial estimates of the camera parameters can then be obtained via RANSAC [56] or a similar iterative method, which is robust to outliers, applied to successive pairs of camera (stereo) correspondences. The error metric used is the reprojection error for the reconstructed points $i = 1 \ldots n_i$ for all camera views. To reduce the space of unknowns, often a subset of the parameters is considered at a time, i. e. intrinsics, lens distortion and extrinsics alone. To refine the calibration, iterative *bundle adjustment* [173] of tracked feature points across multiple views is applied until the re-projection errors are minimized. This aims at minimizing the total sum of squared distances between the observed feature points and the projected object

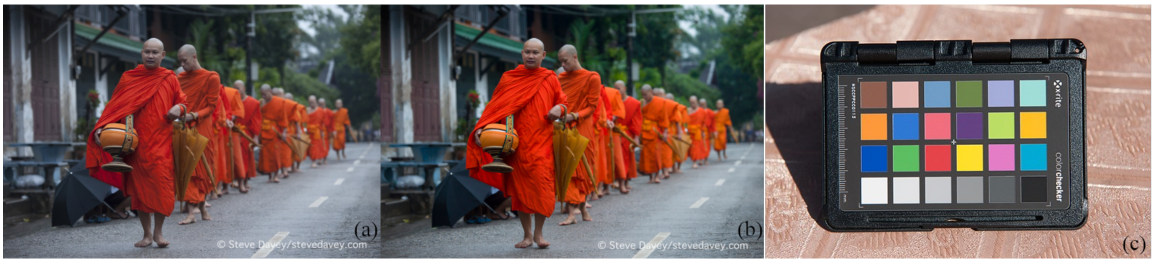


**Figure 2.3:** Example of photograph before and after color calibration. (a) Uncalibrated photograph, (b) calibrated photograph and (c) an example of calibration pattern photographed in bright sunlight.

points $i = 1 \dots n_i$ in all camera views $c = 1 \dots n_c$, using the current estimates for camera parameters:

$$\sum_{c=1}^{n_c} \sum_{i=1}^{n_i} v(i,c) || p_{ic}^h - P_c \cdot \hat{p}_i^h ||^2 \tag{2.5}$$

where $v(i,c) \in [0,1]$ equals 1 if point $i$ is visible from camera $c$ and 0 otherwise.

The accuracy of the calibration highly depends on the quality of the multi-view corner detection, the correctness of the found point correspondences across views and the amount and distribution in the camera space of the observations. To accurately estimate lens distortion parameters, for instance, a discrete amount of sample corners is needed, especially seen at very oblique angles close to the image edges, where the distortion is larger. High quality extrinsics, i. e. camera location and orientation w. r. t. the other cameras, for each camera in the multi-view system can be obtained with few samples, as soon as those are visible from many cameras at a time. While intrinsics and lens distortion affect each camera independently, extrinsics estimated for a multi-view setup must be computed in conjunction with as many cameras as possible to avoid inconsistencies. Since pinhole models ignore geometric distortions or blurring of unfocused objects caused by lenses and finite sized apertures, it is the cameramen task to assure that the object of interest, e. g. the calibration pattern or the performing actor, is in focus during the whole capturing time. This is especially important to accurately find the (sub-pixel) location of the checkerboard corners during calibration or the corresponding surface points during multi-view reconstruction.

In this thesis, we often use the software *The Captury Studio* [30] for semi-automatic accurate multi-view calibration using a moving checkerboard of known size. The software allows inspecting the reprojection error of each camera, perform error analysis and add manual input to improve the estimates. Alternatively, multi-view calibration of static cameras can be obtained using open source libraries, e. g. the *Camera Calibration Toolbox for MATLAB* [17] or *OpenCV* [140].

**Color Calibration** In order to improve the color consistency across views or also improve the color perception of a single view photograph (see an example in Figure 2.3), it is often necessary to perform so called *color calibration*, right after geometric calibration. The camera color perception varies from camera to camera as it depends both on the specific camera properties, i. e. optics and mechanics, as well as on the particular scene lighting. Color calibration typically involves the use of a color chart with known color distribution and RGB codes, see Figure 2.3$(c)$. Corresponding colors across views are then matched with the source color of each checkerboard patch, such that the seen digital color has the same properties.

In this thesis, we either rely on color pre-calibrated in-studio sequences (Chapter 4) or purposefully leave the multi-view system color uncalibrated to test the robustness of our motion and performance

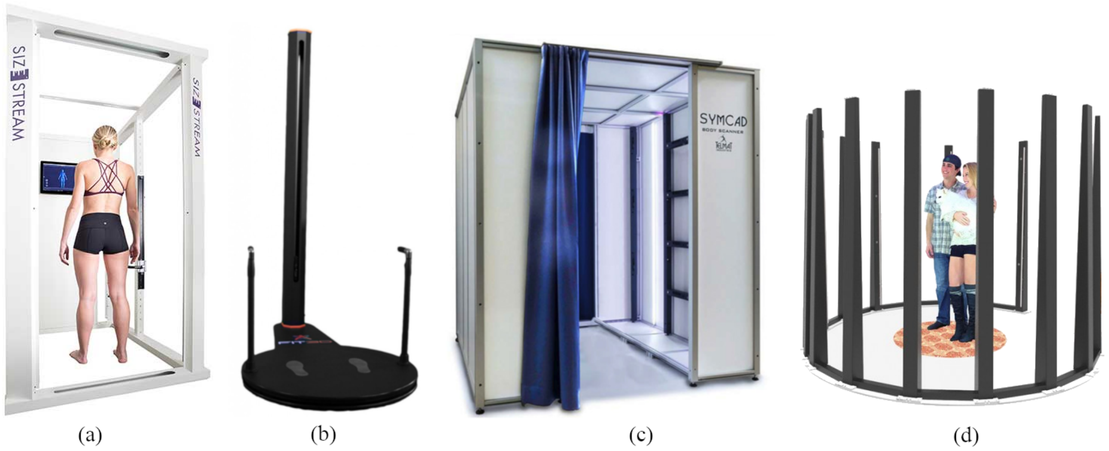


**Figure 2.4:** Full-body scanner examples. (a) Size Stream's SS20 3D Body Scanner [166], (b) Fit3D - Proscanner [57], (c) Telmat Industrie - Symcad III [83], (d) Twindom - Twinstant Mobile [176].

capture methods in complicated outdoor scenarios (Chapter 3 and Chapter 5). Reliable color calibration must handle dynamic color changes visible on the object of interest, i. e. the performing actor, which may be due to e. g. radiometric falloff, non-linear response of the camera sensor and non-uniform (changing) illumination. This is currently an active challenging research topic [54, 111, 199].

#### 2.2.1.2 Body Scanning

There exist several techniques to scan the human body reliably. The most rigorous technique to accurately capture bodies so far is through so called 3D body scanners, also called full-body scanners. Full-body scanners are designed to capture the 3D shape of a person's body and obtain a highly accurate 3D model based on the captured data. The obtained 3D model can be used to visualize the exact body shape and get accurate information regarding sizes, dimensions, posture and so on. Originally developed for the fashion industry, 3D scans are now used in various fields such as healthcare, 3D figurines and 3D photo, fitness and entertainment, as well as input for motion and performance capture applications.

To obtain a full body scan with a 3D scanner, the subject usually has to stand in the middle of a cabin and holds a pose for a few seconds, the time necessary for the scanner to capture sample data from many angles. A 3D software then reconstructs a highly detailed 3D model of the person's body, which can have colors and textures depending on the type of body scanners used. Typical resulting scanning holes caused by occlusion or noise can be easily filled-in using e. g. a *Poisson* reconstruction filter [97]. Figure 2.5 shows an example of triangulated scan mesh with holes at the armpits, feet and hairs, fixed using these kind of filters.

During the 3D capture process, the subject can either stand on a rotating turntable facing a fixed 3D body scanner, or stand still while the sensors located all around the body capture information from all angles. See examples in Figure 2.4. The fastest body scanners require only a few seconds of scanning time and typically deliver high-resolution point clouds of geometric samples that are then automatically triangulated and cleaned to obtain final detailed surface meshes with high fidelity [176, 83, 166, 57, 16]. High capturing speed is required since tiny movements can easily corrupt the scan accuracy by generating holes or mismatches in the samples. By greatly reducing

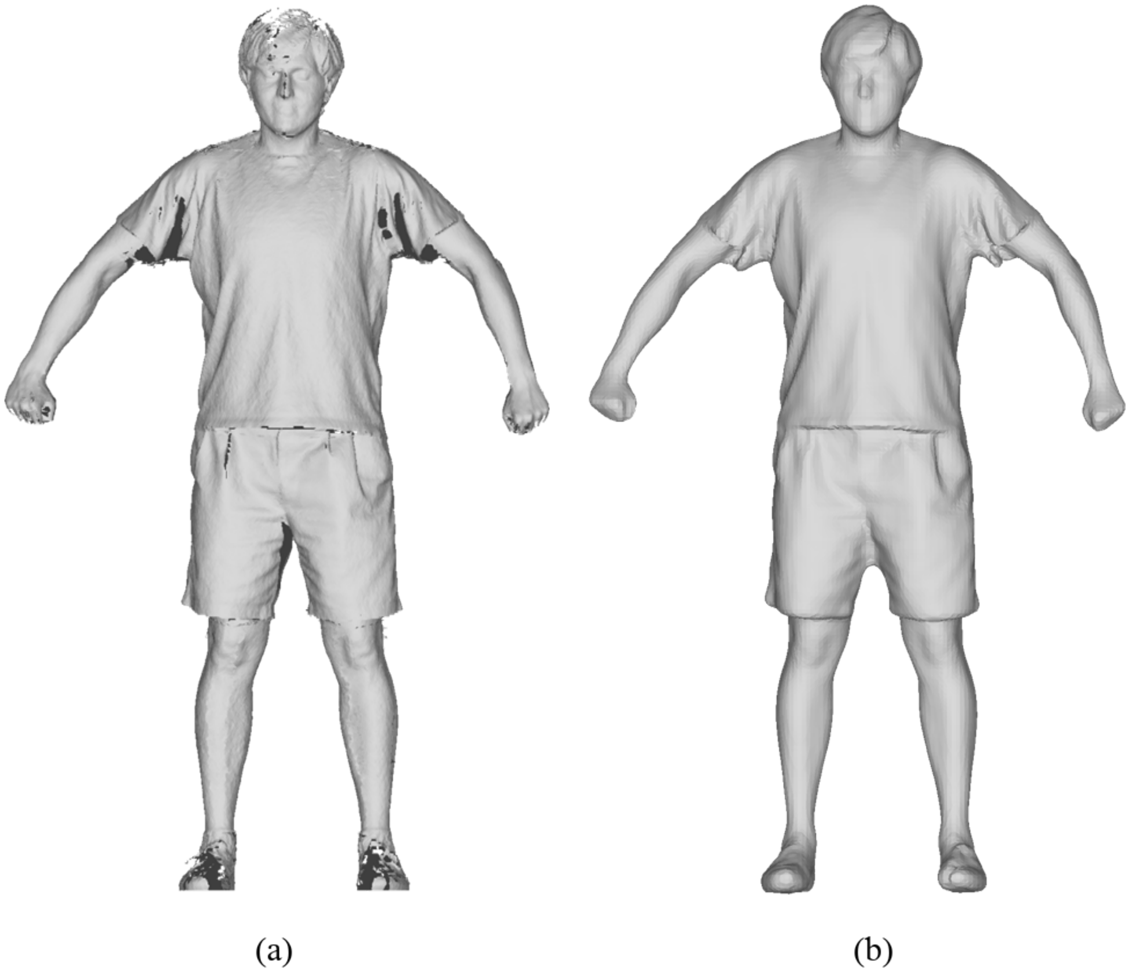


**Figure 2.5:** Example 3D triangulated scan of an actor before and after hole filling. (a) Initial scan of the actor, (b) Detailed Poisson reconstruction.

the acquisition time, down to e. g. a few milliseconds, the human body remains almost completely rigid and therefore can be captured at its highest resolution. One of the most recent and accurate body scanning technologies reaches accuracy levels within 0.1 mm with a maximum resolution of 20 million polygons and 0.01 seconds of scanning time [16].

The most popular acquisition technology is based on structured light sensors [150, 47]. These kind of sensors inspect the way a known light pattern projection onto the scene is deformed when striking object surfaces in a scene. The displacement of the pattern allows for an exact retrieval of the 3D coordinates of details on the object's surface. In its simplest form, structured light consists of a laser beam projecting a single dot onto the scene. Projecting a light plane, resulting in a single slit on the object surface, only requires scanning along a single axis, e. g. top to bottom, compared to the dot projection (requiring two axes scanning), to collect point correspondences over all the actor's body. This technology is commonly used for general 3D scanning. In order to avoid time consuming mechanical scanning, fast solutions consist in projecting multiple (phase shifting) stripes or dots [71].

Invisible infrared (IR) pattern projection allows non-invasive 3D scanning and is widely use for scanning the full 3D human body without altering its appearance [39]. IR structured light technologies

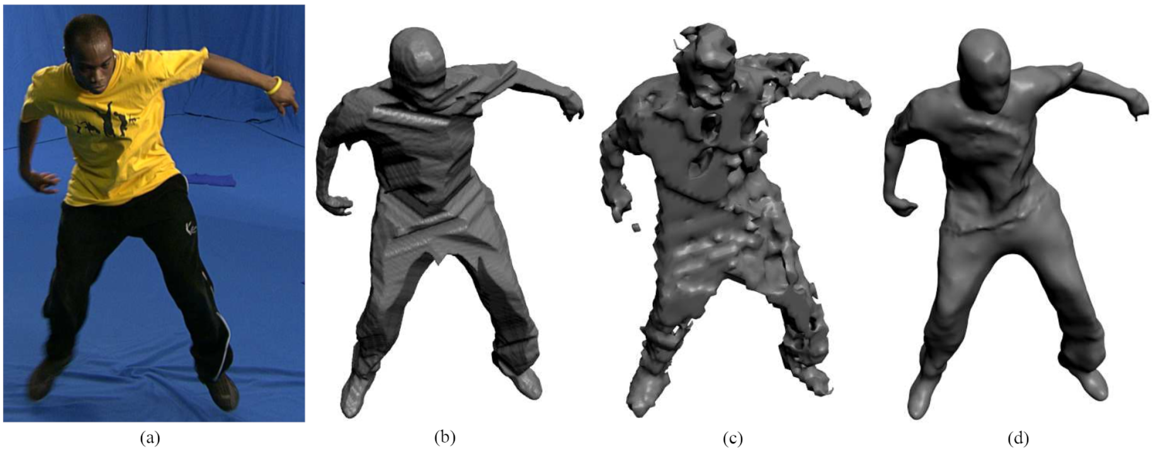


Figure 2.6: Examples of image-based static multi-view reconstruction taken from [162]. (a) One input view out of 8, (b) Shape-from-silhouette reconstruction, (c) Multi-view Stereo reconstruction, (d) Reconstruction obtained by fusing silhouette, feature and multi-view stereo cues.

combined with photogrammetry allow simultaneous 3D scanning and color (texture) acquisition. Photogrammetry stands for a series of techniques to obtain reliable information about physical objects through processes of recording, measuring and interpreting photographic images. Depth from stereo and depth from defocus are examples of photogrammetry aimed at reconstructing depth through the analysis of (multiple) photographs of the object of interest [153, 44, 82]. While these are typically combined with IR pattern projection, pure photogrammetry scanning solutions are also valid 3D scanning solutions. However, they typically require a very large number of high-resolution cameras with accurate calibration [16].

#### 2.2.1.3 Image-based Multi-view Reconstruction

There exist many cues that can be used to extract geometry from multi-view photographs: texture, defocus, shading, contours, and stereo correspondence. Techniques that use stereo correspondences as their main cue, so-called *Multi-view stereo (MVS)* approaches, have been the most successful in terms of robustness and application [162]. *Shape-from-silhouette* methods are also very popular techniques, which enable shape extraction from given silhouette contours [162, 60].

**Multi-view Stereo** Multi-view stereo algorithms are able to construct highly detailed 3D models from multi-view images of the scene [155, 61, 66]. The basic MVS method densely matches pairs of images by finding likely pixel correspondences across stereo views. Using the camera calibration parameters, the image matching problem is simplified from a 2D search over all the image space to a 1D search along the so called *epipolar line* [73]. A pixel in an image generates a 3D optic ray that passes through the pixel and the camera center of the image. The corresponding pixel on another image can only lie on the projection of that optic ray into the second image, seen by a second camera. By exploiting photo-consistency as well as regularization constraints, used to estimate the likelihood of two pixels or groups of pixels being in correspondence, it is possible to identify corresponding pixels across views and reconstruct a 3D point. The 3D point cloud resulting from multi-view stereo matches can be used to reconstruct the detailed shape of the object of interest, either by direct triangulation of the points or by different kinds of holes-free fitting methods, e. g. Poisson fitting [97].

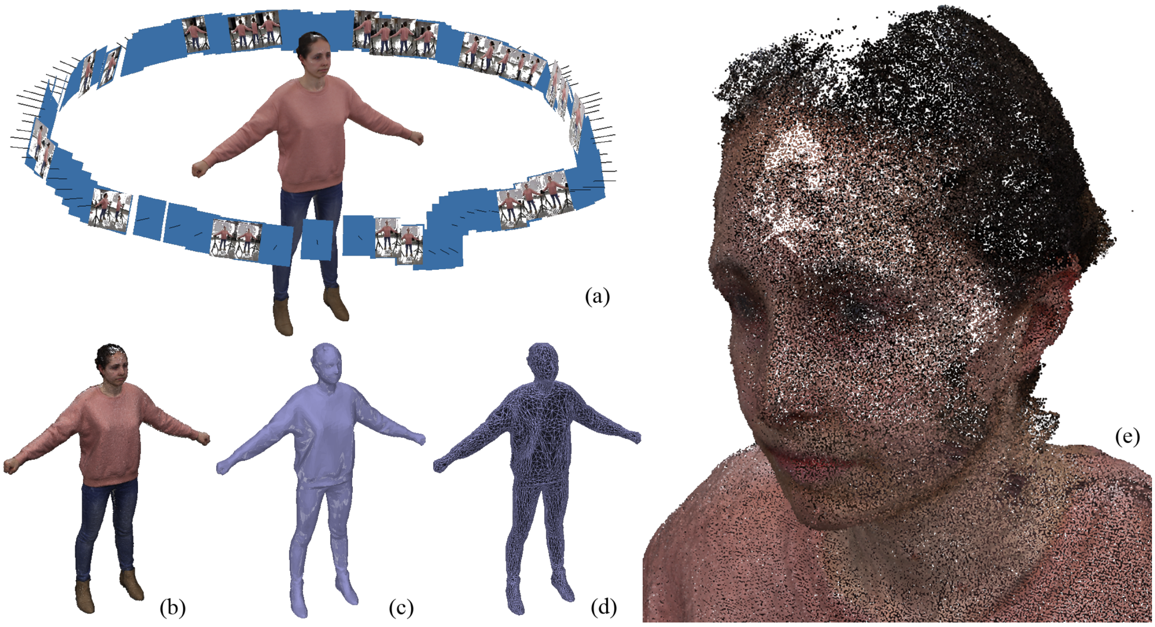


**Figure 2.7:** 3D reconstruction steps using the software *Agisoft PhotoScan* [2]. (a) The acquired multi-view photographs of the actor, (b) Dense estimated point cloud, (c) Resulting shaded mesh, (d) Resulting mesh with visible triangles, (e) Zoomed-in dense estimated point cloud.

MVS techniques reconstruct highly detailed point clouds when textures are available. In the presence of plain-colored objects or objects with complex view-dependent appearance properties, e. g. specularities or transparency, most of the proposed MVS techniques are unable to perform adequately. Humans wearing highly diffuse pieces of apparel are best suited for MVS. Visible ambiguities and noise, which cause mismatches at correspondence finding time, can however generate holes and ghosting geometry that require manual correction. See an example in Figure 2.6$(c)$.

In this thesis, we often make use of the MVS-based software *Agisoft PhotoScan* [2] to estimate a reliable colored mesh of the human actor from a collection of photographs. To obtain the actor mesh, we first capture images of the steady actor from different viewpoints. We typically ask our actors to hold a steady T-pose with slightly opened legs, while relaxing their arms on two sustaining tripods on the sides. The purpose of the tripods is to separate the arms from the sides while still ensuring steadiness of the entire body, allowing photographs to better capture hidden surface points such as the hips. The photographer typically walks in a circle around the actor's body, taking photographs from all sides. We typically adjust the camera focus once before the capture process starts, and then keep it fixed to enhance photograph capturing speed.

The acquired photos are then loaded into Agisoft PhotoScan. The program allows to automatically select good photograph candidate depending on their quality (e. g. sharpness) and to estimate the camera 3D location for each of them based on common features, see Figure 2.7$(a)$. After calibrating all cameras, the software allows to reconstruct a textured mesh out of the manually-segmented dense points. The quality of the model highly depends on the calibration accuracy and on the quality of the camera sensor. We typically reconstruct highly detailed textured meshes from densely estimated point clouds, see Figure 2.7$(c,d)$. See an example of the density of our point clouds in Figure 2.7$(e)$.

Agisoft PhotoScan only allows to obtain fragmented texture maps, as the one in Figure 2.8$(a)$. We typically use *Blender* [58] to map the texture islands to a more compact, human-readable

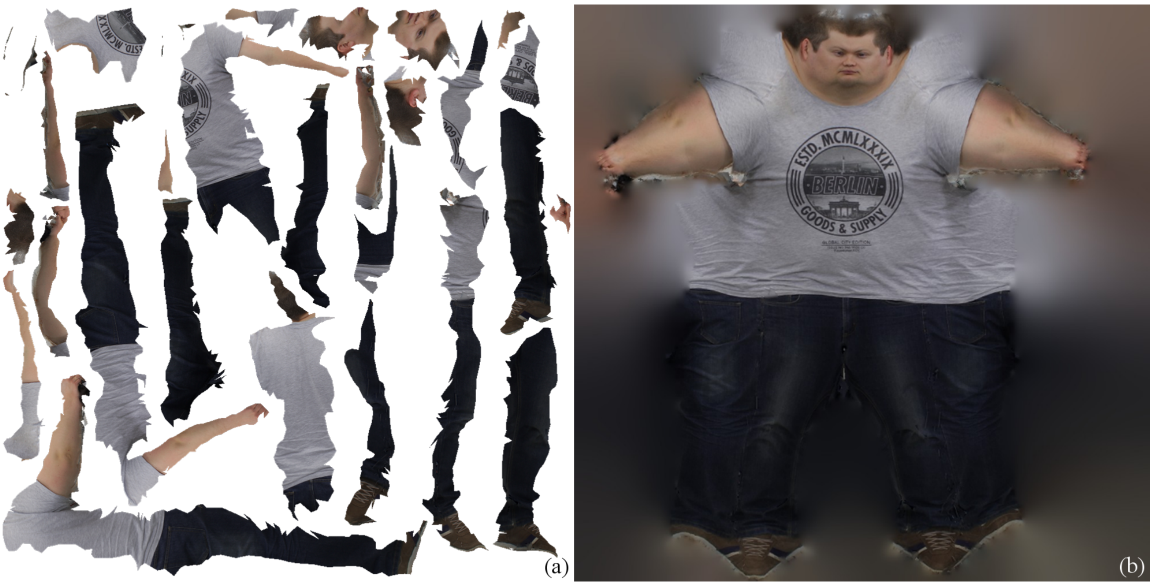


**Figure 2.8:** Example texture maps. (a) Fragmented texture map and (b) the same texture remapped to a single island.

representation, see Figure 2.8$(b)$.

**Shape-from-silhouette** Shape-from-silhouette reconstruction is a 3D reconstruction technique that is based on intersecting multiple 3D reprojected silhouette cones [101]. This class of approaches uses camera parameters to project multi-view silhouettes of an object of interest to the 3D space thereby creating multiple visual cones. The final *visual hull* is obtained as the intersection of all the visual cones generated for different points of view [59, 33]. Alternatively, instead of back-projecting each silhouette into 3D space, a different approach to obtain a visual hull consists in projecting all the voxels from a given voxel grid onto each object silhouette. Then, the visual hull is computed by taking all voxels lying inside a predefined number of silhouettes [114, 158, 134].

Unlike other 3D reconstruction methods, shape-from-silhouette techniques require neither constant object appearance nor the presence of textured regions. If exact silhouette contours are available, transparent objects or objects with complex reflectance properties can be reconstructed faithfully. The primary disadvantage of this kind of approach is the inability to obtain the exact detailed shape when, e. g. , the object presents cavities and internal details that never appear along the reprojection borders of any view. Although there exist methods able to find accurate silhouette boundaries in some uncontrolled scenarios [34, 36], the availability of exact silhouettes is typically restricted to highly controlled capturing setups with calibrated light and monochromatic back-screens for automatic foreground extraction. The scene depicted in Figure 2.6$(a)$ shows an example of ideal capturing setups for silhouette-based techniques. The corresponding low-quality reconstruction result obtained with as few as 8 cameras is depicted in Figure 2.6$(b)$.

### 2.2.2 Skeleton Layer

Given a static geometric model of the actor, the most wide way of defining surface deformations is using skeletal animation techniques. These techniques involve the generation of an embedded *skeleton* layer that can be used to animate the mesh. In the next sections, we describe the general skeleton definition as well as the template human skeleton structure employed throughout this thesis.

#### 2.2.2.1 General Skeleton Definition

The typical skeleton data structure, commonly used in character animation, is inspired by the actual anatomic skeleton of humans. It is usually implemented as a hierarchical structure of interconnected bones, each being a 3-dimensional segment of fixed size, having global 3D location and orientation determined by the parent bones motion. One effective way of storing a skeleton structure in memory is by saving only the connection between the bones, also called *skeletal joints*, with associated local transformation matrices and parental relationships. This algorithmic definition of the skeleton has computational advantages when it comes to animation, see Section 2.3 for details.

The concatenation of rigidly connected joints is called in mechanical engineering a *kinematic chain* [122]. The local 3D transformation of a joint $T_j$ includes a local offset to the parent joint, scale and rotation information. The 3D global transformation of a skeletal joint $T'_j$ is obtained by multiplying its parent's global joint transform $T'_{\text{parent}(j)}$, where $\text{parent}(j)$ indicates the parent joint of $j$, with its own local transform:

$$T'_j := T_j \times T'_{\text{parent}(j)} \tag{2.6}$$

Notice that the *root joint*, which is the upper most joint in the skeleton hierarchy, has also a fictional parent, which is defined as the global coordinate system, i. e. centered in $(0,0,0)$ and with a local transformation matrix corresponding to the identity.

Each joint has at most 6 degrees of freedom, being defined by local translations (max 3 degrees) and rotations (max 3 degrees) along the $x$, $y$ and $z$ axes in the corresponding local coordinate system. The collection of the values assigned to each degree of freedom $s_p, \forall p$ are the skeleton parameters used to describe a particular pose configuration $\mathcal{S} := \{s_p : p \in \{1 \ldots |\mathcal{S}|\}\}$. The *neutral pose*, also called *T-pose*, used throughout this thesis, is the skeleton pose obtained by setting all values in the joint configurations to zero, i. e. $s_p = 0, \forall p$, see Figure 2.1$(b)$. Simplification in the number of joints or degrees of freedom considered are generally possible, for example limiting the knee joint to only allow bending and to forbid twisting. Most of the joints are *spherical joints*, being able to generate rotations around their 3 corresponding local axis. The root joint has typically 6 degrees of freedom to additionally allow global 3D translations of the entire skeletal structure. Joints that allow translation along at least one axis are also called *prismatic joints*.

Given a joint axis $a_j$ and an assigned rotational angle $s_p$, the corresponding rotation matrix $T_j := R_j(s_p) \in \mathbb{R}^{3\times 3}$ in local coordinates can be found using the axis-angle representation:

$$R_j(s_p) := I + (sin(s_p)) \cdot A + (1 - cos(s_p)) \cdot A^2 \tag{2.7}$$

where $I$ is the identity, and $A$ is the cross-product matrix for the normalized axis vector $a_j$ defined as:

$$A := \begin{bmatrix} 0 & -[a_j]_z & [a_j]_y \\ [a_j]_z & 0 & -[a_j]_x \\ -[a_j]_y & [a_j]_z & 0 \end{bmatrix} \tag{2.8}$$

where $[a_j]_{x,y,z}$ are respectively the $x$, $y$ and $z$ coordinates of the normalized axis vector $a_j$. The corresponding $4 \times 4$ local rotation matrix in homogeneous space $R^h_j$ can be obtained as:

$$R^h_j := \begin{bmatrix} R_j & 0 \\ 0 & 1 \end{bmatrix} \tag{2.9}$$

The translation matrix $T_j := t_j(s_p)$ along an axis $a_j$ by $s_p$ in local coordinates is defined as:

$$t_j(s_p) := \begin{bmatrix} 1 & 0 & 0 & [a_j]_x s_p \\ 0 & 1 & 0 & [a_j]_y s_p \\ 0 & 0 & 1 & [a_j]_z s_p \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.10}$$

Multiple rotations and translations along multiple joint axes, i. e. $x$, $y$ and $z$, are obtained concatenating the corresponding matrices:

$$T_j := [t_j]_z \times [R^h_j]_z \times [t_j]_y \times [R^h_j]_y \times [t_j]_x \times [R^h_j]_x. \tag{2.11}$$

where $[R^h_j]_{x,y,z}$ and $[t_j]_{x,y,z}$ are respectively the rotation and translation matrices along either the $x$, $y$ or $z$ joint local axis. In principle, the global coordinates of a joint in homogeneous space $\hat{j}^{h'} \in \mathbb{R}^4$ can be computed as:

$$\hat{j}^{h'} := T'_j \cdot \hat{j}^h \tag{2.12}$$

where $\hat{j}^h$ is the joint location in local coordinates w. r. t. the parent joint.

The process of computing the location of the joints in global coordinates from a given skeleton configuration $\widetilde{\mathcal{S}}$ is called *forward kinematics*, while the inverse process, i. e. computing $\widetilde{\mathcal{S}}$ given the 3D location of the joints, is called *inverse kinematics*. The latter is more intuitive for end-user applications like animation, e. g. allowing animators to choose the exact 3D location of the joint, sparing the user the burden of setting the actual intricate joint parameters, which are instead automatically estimated by the program in the background.

Differently shaped and dressed humans are characterized by the same underlying skeleton structure, with the same bones, joint connections and possible configurations. Variations are possible though in the bone length and rotation amplitude. For example, some people can perform leg splits, being able to perform wide hip rotations, while others cannot. *Joint limits* define upper $L_p$ and lower $l_p$ bound for each degree of freedom, i. e. $s_p \in [l_p, L_p], \forall p$.

While techniques do exist that estimate skeleton structures for some vertebrates automatically [108, 8, 169], related work has shown that the results are unstable, especially in the presence of large apparel and complicated poses and shapes [102]. Since in this thesis we focus exclusively on the shape of humans, we fix the underlying general skeleton structure and focus on (semi-) automatic refinement of the bone lengths to find best fits for differently shaped actors.

#### 2.2.2.2 Our Template Human Skeleton Layer

In this section, we describe our template human skeleton structure we use throughout this thesis. The template represents the common human skeleton hierarchy, independent of body shape and clothing. To best adapt to differently shaped humans, the template skeleton must be embedded into the corresponding actor's mesh by adaptively changing bone lengths and pose configuration. Our automatic skeleton embedding technique is presented later in Section 2.3.1. The overall structure consists of 24 joints: 4 for each hip-toe sub-chain (*hip*, *knee*, *ankle*, *toe*), 4 for the clavicle-wrist sub-chain (*clavicle*, *shoulder*, *elbow*, *wrist*) and 8 for the head-pelvis (*head*, *neck* and 6 more *spine* joints, namely *spine1*...*spine6*). See a visualization of the skeleton structure in Figure 2.1($b$). The skeleton is parameterized with a 48-dimensional vector $\mathcal{S}$ consisting of 6 global translation and rotation parameters, and 42 more joint rotation parameters. Each parameter controls one or multiple skeleton joints and operates along the corresponding 3 local joint axes. Some parameters describe

**Table 2.1:** Active degrees of freedom for each skeletal joint used to describe skeleton pose configurations of our template human skeleton. For each skeletal joint, the table gives the active degrees of freedom expressed as $rx$, $ry$ and/or $rz$ for rotations and $tx$, $ty$, $tz$ for translations respectively around the $x$, $y$ and $z$ local joint axes. The last column shows the limits (radians for rotations and millimeters for translations) for each degree of freedom.

| Joint Name | Active Degrees of Freedom | Limits ($[lower, upper]$) |
|---|---|---|
| root | tx,ty,tz,rx,ry,rz | no limits |
| spine1… spine6 | rx,ry,rz | $[-0.5,0.5],[-0.55,0.55],[-0.8,0.25]$ |
| neck | rx,ry,rz | $[-0.667,0.667],[-0.3,0.3],[-0.5,0]$ |
| 2 × clavicles | ry,rz | $[-0.4,0.5],[-0.4,0.4]$ |
| 2 × shoulders | rx,ry,rz | $[-1.6,1.6],[-2.3,1.3],[-10,10]$ |
| 2 × elbows | rx,ry,rz | $[-2.7,-0.1],[-2.0,1.46],[-1.5,1.5]$ |
| 2 × wrists | ry,rz | $[-0.5,0.5],[-1.3,1.3]$ |
| 2 × hips | rx,ry,rz | $[-10,10],][-0.5,0.5],[-1,0.6]]$ |
| 2 × knees | rx | $[-2.5,-0.1]$ |
| 2 × ankles | rx,ry,rz | $[-0.4,0.4],[-0.5,0.5],[-0.25,0.25]$ |
| 2 × toes | rx | $[-0.7,0.7]$ |

rotations with respect of a subset of the available axes. For instance, the *knee* joint is only allowed a rotation along the x-axis, corresponding to bending.

Table 2.1 shows all skeleton joints with corresponding active degrees of freedom and limits. Most of the chosen parameters reflect the mechanical structure and behavior of real human skeletons. Some bones describing finger, toes and facial movements are excluded from our skeleton structure, as they fall out of the scope of this thesis in terms of level of captured detail. The effect of changing single pose parameters, within the limits, is visualized in Figure 2.9 on an example character.

### 2.2.3 Volumetric Layer

Volumetric actor model representations provide different shape information, compared to surface meshes, that can be effectively employed for volume-preserving deformation. In the fields of motion and performance capture, they are often used to approximate the human body volume, allowing advantageous model-to-image matching in a reduced space of unknowns, i. e. from a large amount of surface points to a few volumetric primitive locations [50, 164, 143]. The volumetric layer of our body model is described in detail in Section 2.2.3.2. Volume-based models presented in the literature are described in the next Section 2.2.3.1.

#### 2.2.3.1 Volumetric Models in the Literature

Many methods employ a specific set of primitives, such as ellipses, cylinders or cardboards, [189, 22, 90], which are joined together to form reliable human shape configurations. Primitives are typically either connected within each other or rigidly attached to an underlying skeleton structure [164, 50]. Some methods assign an additional color attribute to the primitives, which is used for improved model-to-image similarity assessment. Due to their reduced detail, their use is mostly restricted to skeleton (motion) tracking application rather than surface tracking. More detailed general volumetric shape models employ tetrahedral, multi-resolution voxels or even particle-based representations [40, 194, 112]. These have been successfully used to enable detail preserving, finer-scale 3D reconstruction. Still, detail reconstruction remains challenging.

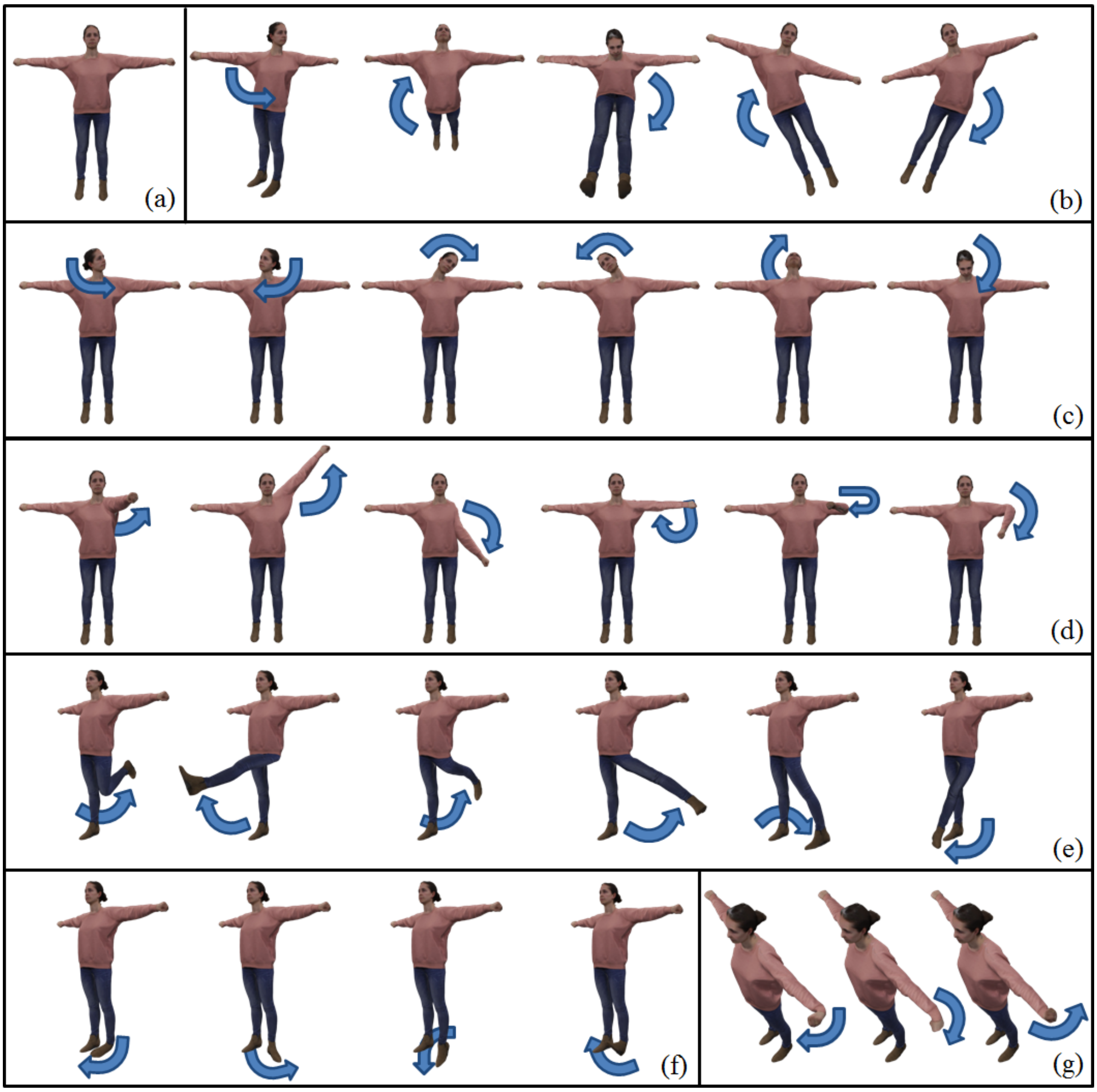


**Figure 2.9:** Visualization of pose configurations resulting from changing single skeletal parameters at a time on an example character. The blue arrows indicate the direction of the movement. (a) Neutral pose or T-pose, (b) changing the spine parameters resulting in upper body twist, lean and bend, (c) changing the neck parameters resulting in head movements, (d) changing the left arm parameters resulting in shoulder and elbow movements, (e) changing the left leg parameters resulting in hip and knee movements, (f) changing the left ankle parameters resulting in foot movements, (g) changing the left wrist parameters resulting in hand movements.

Among the volumetric representations, implicit models have recently become attractive due to their smooth analytical definitions. Implicitly defined surfaces represented through non-parametric *signed distance function (SDF)* are advantageous for volumetric reconstruction and tracking [128, 46, 137, 80, 131], also from single RGB-D input [84, 127, 128]. The use of implicit surfaces for shape and motion estimation from multi-view video was originally proposed by Plankers and Fua [137], who use smooth implicit surfaces attached to an articulated skeleton to approximate a human 3D shape, see an example in Figure 2.10$(a)$. This representation allows defining a distance function of data points and models that is smooth everywhere in space. Similarly, Ilic and Fua [80] create *implicit*

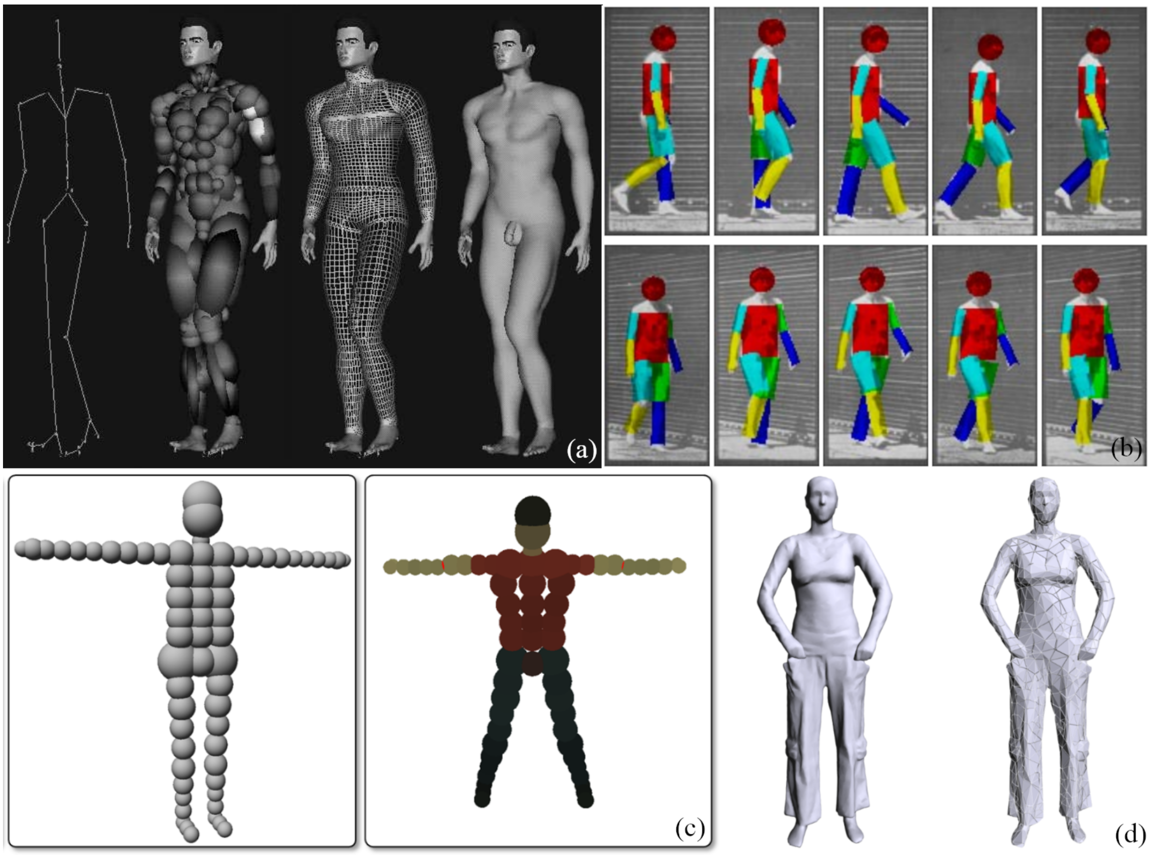


**Figure 2.10:** Examples of volumetric human actor representations. (a) Metaballs based model from Plankers et al. [137], (b) Cylinder based model fitting of [22], (c) Gaussian-based model introduced by Stoll et al. [164], (d) Tetrahedral model from Aguiar et al. [40].

*meshes* by attaching triangular primitives to the faces of explicit 3D meshes. In a recent work, Stoll et al. [164] employ a set of 3D parametric colored Gaussian functions rigidly attached to the skeleton, see Figure 2.10$(c)$. Their model formulation is extensively used throughout this thesis both for multi-view coarse and fine-scale tracking as well as an approximation of signed distance functions for skinning. Our volumetric Gaussian-based layer is described in detail in Section 2.2.3.2.

In order to accurately approximate an actor's mesh volume with a chosen set of primitives, some methods automatically optimize the degrees of freedom that characterize the chosen volume primitives, i. e. their 3D location, orientation and dimensions, such that they optimally fit the surface model or the multi-view silhouettes [164]. Alternatively, the primitives are manually fitted inside the surface model.

#### 2.2.3.2 Our Volumetric Layer

Our volumetric model layer that is used throughout this thesis is inspired by the implicit Gaussian-based representation introduced by Stoll et al. [164]. A set of 3D Gaussians with an additional color attribute are rigidly attached to the underlying skeleton to approximate coarsely the body volume. A

*Volumetric Gaussian* $\hat{G}_m$ is defined by its 3D mean $\hat{\mu}_m$ and 3D standard deviation $\hat{\sigma}_m$:

$$\hat{G}_m(x) := \frac{1}{\hat{\sigma}_m\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\hat{\mu}_m}{\hat{\sigma}_m}\right)^2\right) \tag{2.13}$$

Our template human volumetric layer is defined by 66 Volumetric Gaussians manually placed all around the template skeleton structure, introduced in Section 2.2.2.2. In particular, we employ 9 Volumetric Gaussians to approximate each leg volume, 5 for the feet, 10 for the abdomen, 6 for the head, 8 for the arms and 3 for the hands. The Volumetric Gaussians are distributed evenly along the limbs and the abdomen. The head is filled such that front and back are covered by different Volumetric Gaussians. This is used to allow different coloring of the face, i. e. skin color, w. r. t. the back side, i. e. hair color. After an initial manual fit of the template volumetric layer inside a given actor's mesh, the average color of the surrounding vertices of each $\hat{G}_m$ is assigned as additional color attribute in the HSV space $\eta_m$. Also the attachment location and the standard deviation of each Volumetric Gaussian are adjusted based on the input actor's shape. As shown later in Section 2.3.1, our volumetric layer representation can be fitted automatically inside a given actor's mesh volume using a technique proposed in the literature. The volumetric layer also serves for optimal skeleton structure fitting within a human mesh. A visualization of the volumetric 3D Gaussian-based layer can be seen in Figure 2.1$(c)$.

Each 3D Volumetric Gaussians is rigidly attached to one underlying skeleton bone. Consequently, bone motions move the affected Volumetric Gaussians as well. The formula to find the new location, i. e. mean $\hat{\mu}_m^{h'}$, of a Volumetric Gaussian in homogeneous depending on the skeleton pose parameters is given by:

$$\hat{\mu}_m^{h'} = T_j' \hat{\mu}_m^{jh} \tag{2.14}$$

where $\hat{\mu}_m^{jh}$ is the Volumetric Gaussian mean in local homogeneous coordinates w. r. t. the single parent joint $j$.

## 2.3 Rigging Pipeline

In animation, the process of describing surface mesh deformations with the motion of an underlying kinematic skeleton is called *rigging*. Given an actor's mesh, the rigging pipeline consists in three main steps: defining the underlying skeleton structure, performing skeleton fitting inside the mesh of the actor and finally binding the surface to the skeleton. In this thesis, we make extensive use of rigging to define links among the multiple human body layers. Since differently shaped and dressed humans share a similar hierarchical skeleton structure, we use the skeleton structure described in Section 2.2.2.2 and focus on the remaining two steps, namely fitting, also called *skeleton embedding*, and binding, also called *skinning*. After giving a brief introduction on proposed approaches for embedding and skinning, Section 2.3.2.2 outlines our proposed approach for reliably and efficiently estimating the required skinning weights, i. e. the weights that describe how the underlying bones affect each surface vertex of the human body model.

### 2.3.1 Skeleton embedding

Given a template human skeleton, the task of fitting it inside a given actor's geometry is called *skeleton embedding*. Skeleton embedding is the first step of the so called *rigging* process and consists

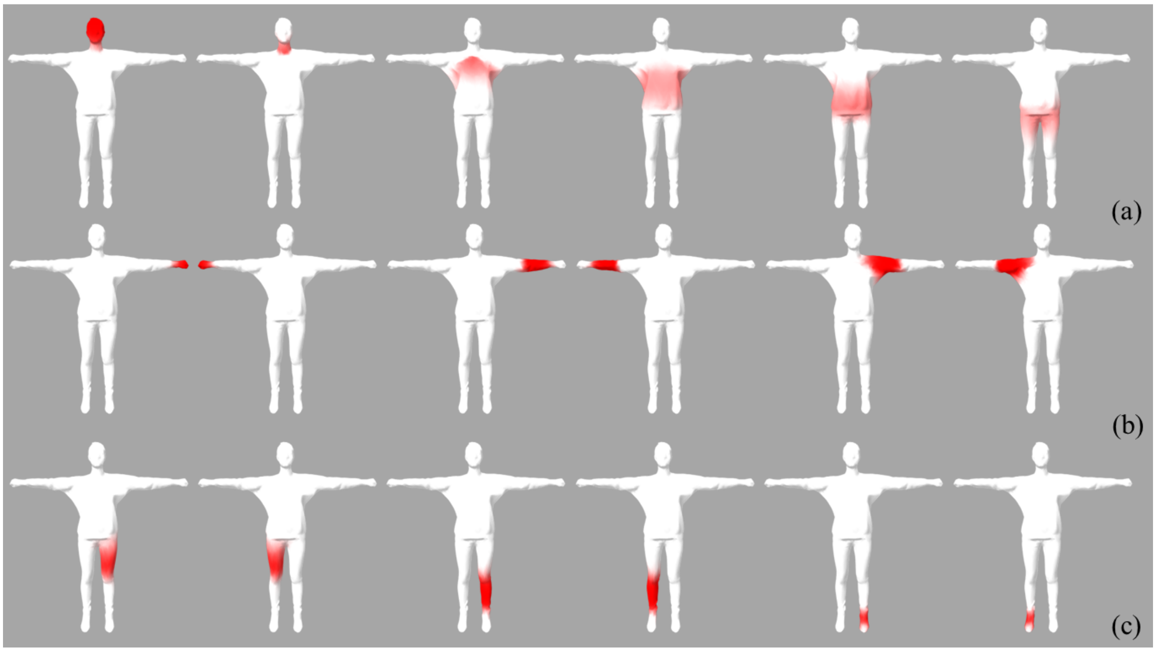


**Figure 2.11:** Visualization of skinning weights for different bones on a character. The skinning weights are obtained using the method explained in Section 2.3.2.2. (a) Skinning weights of the spine, (b) skinning weights of the arms, (c) skinning weights of the legs.

in estimating bone lengths as well as joint configurations that best fit the actor surface model and pose. Optimal skeleton structure fits place joints as close as possible to the correspondent real joint location. Solving such a problem automatically may lead to imperfect embedding solutions that require manual interventions. Errors at embedding time typically cause poor quality rigging solutions, which in turn result in distracting artifacts during animation.

Methods that provide approximate bone positions based on geometry surfaces and shapes are often insufficient for animation of photorealistic character models [91]. Embedding techniques especially designed for human shapes have shown more accurate results than general methods [55, 93, 120, 118]. Such methods are mostly based on initial template rigs or rely on a database of rigs, automatically adapted to the input human shape, to enable robust skeleton embedding. However for these techniques, model fitting may still be inaccurate in the presence of excessive clothing.

When the actor's geometry is available in different pose configurations, e. g. multiple 3D body scans, automatic skeleton embedding produces more accurate results. [102, 49]. The accuracy of the skeleton fit however still depends on the quality and amount of pose variations observed. Few observations or lots of similarly posed meshes still lead to unreliable results that require manual corrections. Despite the improvements introduced in the literature, skeleton embedding remains a semi-automatic process, where the user's intervention, aimed at correcting misplacement or providing prior annotations, is still needed.

In this thesis, we make extensive use of the additional volumetric layer, rigidly bounded to the underlying skeleton structure, to automatically fit both the skeleton and the Volumetric Gaussians inside a given actor's mesh. Specifically, we optimize for the joint configuration, i. e. skeletal pose, the bone lengths and the Volumetric Gaussians distribution with relative standard deviation, such that both coarse layers fit optimally the detailed surface layer. Due to the large amount of unknowns, we optimize iteratively for a subset of the parameters. To effectively fill in the volume of the actor's shape, we automatically generate multi-view silhouettes of the input mesh as seen from several fictional calibrated cameras placed around in a circle. The reprojected volume approximation given by the silhouettes is used to find the best set of reprojected Volumetric Gaussian, in terms of mean and standard deviation, across all camera views. This technique has been proposed and successfully employed by Stoll et al. [164] and is outlined in detail in Section 2.4.1.4. We make extensive use of their released software *The Captury Studio* [30] for automatic embedding.

### 2.3.2 Skinning

Skinning is the process of binding the actor surface mesh to the embedded skeleton structure, such that deformations of the surface vertices are determined by skeleton pose configurations. When an association between a surface mesh and a skeleton is given, simple bending of a joint displaces multiple vertices according to the underlying bone motion. A mesh skinned to a skeleton structure is also called *rigged mesh* and is widely used in the animation and movie industry.

One of the most popular geometric skinning methods widely used in animation and interactive applications is *linear blend skinning (LBS)* [113]. LBS defines a set of *blending weights* or *skinning weights* $w_{i,j}, \forall i, j$ that tell how much each vertex $i$, is affected by the joint $j$. See a visualization of skinning weights in Figure 2.11. A single bone is typically associated with a group of vertices. For instance, the thigh bone is associated with the vertices making up the polygons in the model's thigh. The deformed vertex location in 4D of vertex $v_i^h$, i. e. $v_i^{h'}$, in homogeneous space is computed using the following weighted sum:

$$v_i^{h'} = \sum_{j=1}^{n_j} w_{i,j} T_j' v_i^{jh} \tag{2.15}$$

where $T_j'$ is the global transform of joint $j$ in global coordinate system and $v_i^{jh}$ is the 4D vertex location in local coordinates w. r. t. the parent joint $j$. Note that vertices located close to bone intersections, i. e. joints, are likely to be associated to two or more bones at the same time and the mixed influence generates a nicely looking smooth deforming surface in those areas. LBS has real-time performance and provides acceptable deformations in a good range of cases. The main disadvantages of LBS are the loss of volume in bent areas and the unnatural *candy-wrapper* effect that appears when skeleton joints are twisted.

Geometric skinning with approximate *dual quaternion linear blending (DLB)* partially solves these issues [95]. The solution consists in converting each transformation into a unit dual quaternion $q_j$ [98]. Then, a normalized linearly blended dual quaternion $q_i$ is computed using the skinning weights:

$$q_i := \frac{\sum_j w_{i,j} q_j}{||\sum_j w_{i,j} q_j||} \tag{2.16}$$

This blended unit dual quaternion is guaranteed to represent a rigid body transformation $\tilde{T}_j'$, which can replace the original LBS term, to estimate skinned vertices locations as:

$$v_i^{h'} = \sum_{j=1}^{n_j} w_{i,j} \tilde{T}_j' v_i^{jh} \tag{2.17}$$

DLB exhibits less artifacts than LBS at minimal additional cost, and is the chosen skinning approach throughout this thesis. Alternative more powerful approaches typically involve complex and computationally expensive strategies either inspired by physics [29, 161, 172, 96, 149, 177] or based

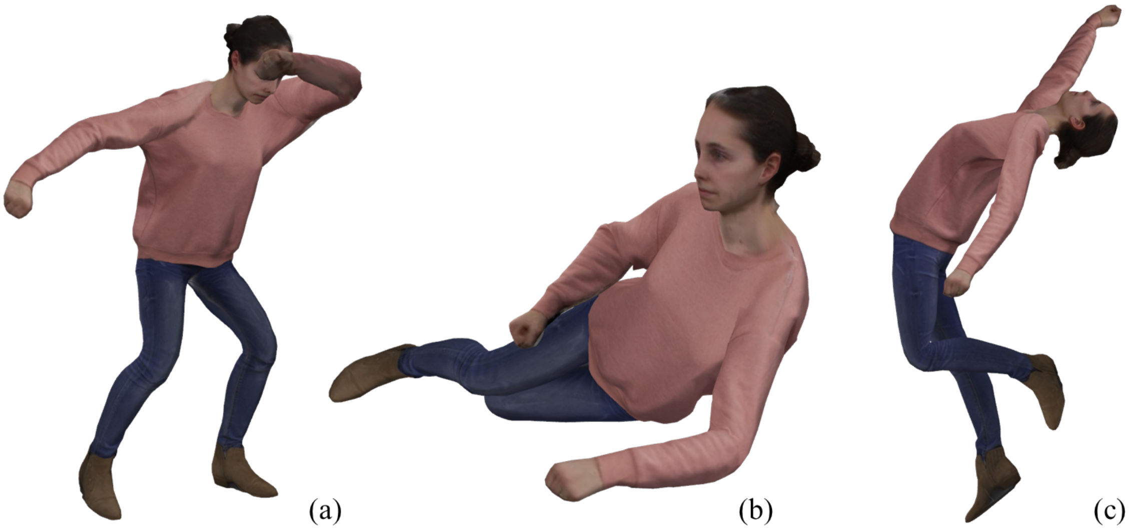


**Figure 2.12:** Reposing examples of a skinned actor model. The figure shows the expressiveness of our multi-layer human body model through realistic poses.

on multiple input meshes [182, 129]. These approaches deliver a high level of realism. Some even simulate realistic muscle bulges. However, the associated high computational costs together with the unusual settings restrict the applicability of these methods.

Another important step of the skinning process, apart from defining the algorithm to estimate global vertex locations given skeleton pose configuration, is the estimation of the skinning weights, described in the next section.

#### 2.3.2.1 Skinning Weights Estimation in the Literature

Few automatic methods have been proposed to estimate skinning weights from a static 3D model [10, 181, 94, 87]. This is a non-trivial problem due to a number of difficulties. For the chosen set of weights, the corresponding skin deformations caused by skeleton motion must produce natural looking surface displacements. A realistic look is given by convincing feature-preserving, non-rigid deformation, without distortions, loss of volume or other visible artifacts. The assignment of skinning weights must account for the resolution, the shape and the individual body parts. Assigned weights are typically distributed non-uniformly on the surface and present differently smoothed overlapping influence depending on the specific body part, i. e. varying smoothing window size for influence transition in bone intersection areas.

Some generalized methods that ignore the different body parts make simplifying assumptions on the latter: rather than producing a uniform weight distribution, which does not account for different body parts and shape, they tend to keep the width of a transition between two bones meeting at a joint proportional to the distance from the joint to the surface. Proximity to bones is an unreliable measure for skinning weight assignment, since it ignores the actual shape and can cause wrong attachments, e. g. part of the torso being attached to an arm.

More robust approaches make use of implicit 3D signed distance approximations of the surface model to assign skinning weights. This conversion helps in correctly estimating a valid set of skinning weights, assuming the input model is free from self-intersections. However, due to the cost of volumetric tessellation, some methods made simplifications by running distance computations only for areas close to the surface, which reduces the attachment time to a few seconds [10]. Approaches using skinning by example, i. e. estimating skinning based on a set of given reposed meshes, have demonstrated superior results when multiple actor models are provided as input [102, 49, 89, 75]. In the typical single model restricted conditions, approaches specifically designed for human shapes have also found reasonable weight assignment solutions, for example using automatic rigging attributes transferred from template [55]. The quality of this kind of data-driven approaches is however limited by the shape space spanned by models in the used database.

Despite the progresses made, automatically assessed skinning weights are still suboptimal and require some manual adjustment, also called *weight painting*, to account for imperfections and unnatural deformations resulting from poor automatic skinning weight assignment. The typical skinning weights assignment process involves iterative skeleton embedding adjustment and skinning weights repainting, given initial distribution guesses provided by an automatic skinning weight estimation method. After semi-automatic skeleton embedding, an initial set of weights is automatically estimated. At this point, artists begin moving each joint around and make sure the resulting deformation of the skin corresponds to the wanted surface displacement, see reposing examples in Figure 2.12. In case of deformation errors, e. g. wrapping artifacts around joint areas, interactive tools allow to apply manual corrections by local re-painting of weights. Alternatively, the skeleton embedding can be manually refined to obtain improved estimates of skinning to start with. This cumbersome iterative process is typically performed by experienced animators to guarantee outstanding skeleton-to-model agreement, which is required for realistic skinning. In the next section, we describe our highly simplified skinning weights assignment approach, specifically build for human actors, which exploits the potentials of the underlying volumetric layer.

#### 2.3.2.2 Our Volume based Skinning Weights Estimation

In this section, we describe our approach for assessing reliable skinning weights using the intermediate volumetric actor layer, introduced in Section 2.2.3.2. The volumetric layer is a 3D parametric representation of the actor's volume based on a set of 3D colored Gaussians attached to the underlying skeleton structure. Gaussians are by definition smooth functions with infinite support having global maxima located at the mean $\mu_m$, which usually lies nearby a skeleton bone segment. Implicitly, a Gaussians-based representation can be seen as a distance function defined everywhere in the space, where the gradient points toward the closest skeleton point. This makes it possible to evaluate the distance $d_m(v_i)$ from all Volumetric Gaussians $\hat{G}_m$ of the volumetric layer at the 3D mesh vertex locations $v_i, \forall i$:

$$d_m(v_i) := \hat{G}_m(v_i) := \frac{1}{\hat{\sigma}_m\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{v_i - \hat{\mu}_m}{\hat{\sigma}_m}\right)^2\right) \tag{2.18}$$

To enhance computational speed, we only consider the closest Volumetric Gaussians up to a certain user-defined threshold $\delta$, $D(v_i) := \{(m, d_m) : d_m < \delta\}$. Then, to effectively find the closest joints, we take the parent skeleton joint of the found Volumetric Gaussians. The skinning weights are then estimated as:

$$w_{i,j} := \sum_{m \in D(v_i)} d_m(v_i) : j \text{ is parent of } m \tag{2.19}$$

We additionally normalize the found skinning weights, such that $\sum_{j=1}^{n_j} w_{i,j} = 1, \forall i$.

This method for skinning weights assessment is used extensively throughout this thesis and was originally introduced in one of our co-authored works by Rhodin et al. [142]. The proposed approach

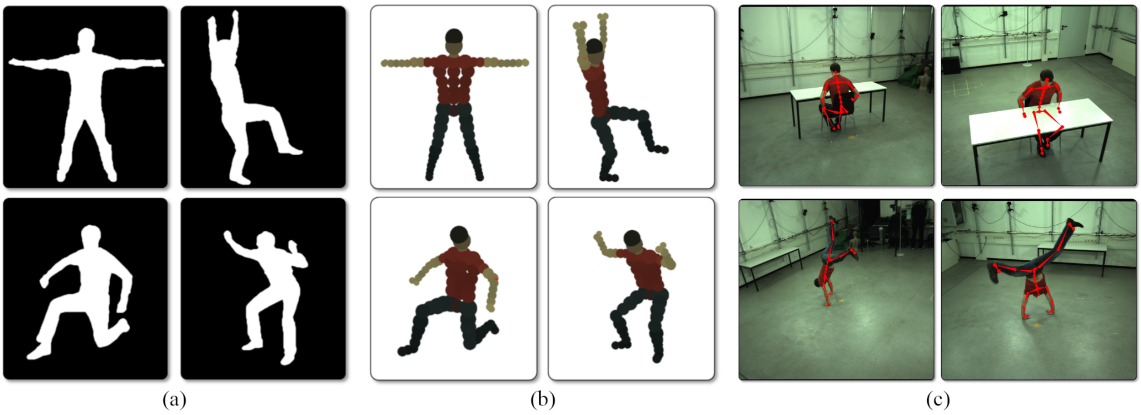


**Figure 2.13:** Actor modeling from segmented example pose images and tracking results from [164]. (a) Semi-automatically segmented input images showing different actor poses (1 image per pose shown out of 8) and (b) the resulting personalized actor body model. (c) Tracking results shown as skeleton overlay over the input images.

has computational advantages w. r. t. the work of Baran et al. [10]. First, it is based on available volumetric information, rather than requiring computationally expensive explicit estimation of heat distribution withing the surface mesh as it is done in [10]. Secondly, by fully exploiting the Volumetric Gaussian-based representation, which clearly distinguishes between different body parts, our method does not require any inside-outside surface check or handling of special cases, needed, e. g. , in presence of non-manifold geometry with holes, disconnected surfaces or multiple internal faces. Its high computational performance and independence from the geometric shape definition beats most of the related approaches in terms of time.

The skinning weight quality highly depends on the 3D Gaussian model resolution, i. e. the number of primitives as well as their distribution within the mesh surface. Poorly represented volume can easily lead to inaccurate skinning weights estimation. Nevertheless, this method robustly delivers adequate weight sets, which are good initial guesses that can be easily semi-automatically refined. Our interactive software tool, we built to accomplish this task, also allows to manually paint and adjust the automatically estimated skinning weights or (locally) smooth the weights over the surface with a user-specified smoothing kernel, if needed. In this work, we make few to no use of the weight painting tool, thus demonstrating the effectiveness of our automatic skinning weights assignment approach, compared to related methods.

## 2.4 Skeleton Tracking

In this section, we focus on relevant previous work, which exploits the potentials of the Gaussian-based representation both for multi-view mocap and differentiable visibility formulation. The remainder of this section is dedicated to the description of the skeleton tracking approach proposed by Stoll et al. [164], which is extensively used throughout this thesis, as well as conceptually extended to performance capture of surface detail later on in this thesis. In this section, we also discuss an extension of the work of [164] initially proposed by Rhodin et al. [143] in a co-authored work.

### 2.4.1 Volumetric Tracking by Stoll et al. [164]

Stoll et al. [164] present an approach for modeling the input multi-view video sequence and the human body by Sums of spatial Gaussians, that allow to perform fast and high-quality markerless mocap. Based on the Gaussians-based models of the input scene, [164] introduce a novel continuous and differentiable model-to-image similarity measure that can be used to estimate the skeletal motion of human actors at high frame rates.

#### 2.4.1.1 Scene Model

The body model representation employed by Stoll and colleagues corresponds to our skeleton and Gaussian-based volumetric layers, described respectively in Section 2.2.2.2 and Section 2.2.3.2. As shown in [164], the implicitly defined Gaussians-based representation $\hat{G}_m, \forall m = 1 \dots n_m$ is advantageous for skeleton mocap in outdoor scenarios, where scene conditions are less controlled.

The model Gaussians can be easily projected into the image space, to obtain 2D Gaussians $G_m$. The Gaussian mean in image space $\mu_m$ is obtained projecting the original mean in 3D $\hat{\mu}_m$ using the camera projection matrix $P$:

$$\mu_m = \begin{pmatrix} \frac{[P\hat{\mu}_m^h]_x}{[P\hat{\mu}_m^h]_z} \\ \frac{[P\hat{\mu}_m^h]_y}{[P\hat{\mu}_m^h]_z} \end{pmatrix} = \begin{pmatrix} \frac{[\mu_m^p]_x}{[\mu_m^p]_z} \\ \frac{[\mu_m^p]_y}{[\mu_m^p]_z} \end{pmatrix} \in \mathbb{R}^2, \tag{2.20}$$

where $[\hat{\mu}_m^h]_{x,y,z}$ are the respective coordinates of the mean in homogeneous coordinates (i. e. the $4^{th}$ dimension is set to 1), $[\mu_m^p]_{x,y,z}$ are the respective coordinates of the projected mean in homogeneous coordinates. Similarly, the corresponding 2D standard deviation $\sigma_m$ of a Volumetric Gaussian is obtained as follows:

$$\sigma_m = \frac{\hat{\sigma}_m f}{[P\hat{\mu}_m^h]_z} = \frac{\hat{\sigma}_m f}{[\mu_m^p]_z} \in \mathbb{R}, \tag{2.21}$$

where $f$ is the camera focal length and $\hat{\sigma}_m$ the standard deviation in 3D. The Volumetric Gaussian colors $\eta_m$ are stored in the HSV color space for later computations.

To obtain a similar mathematical definition of the input image pixels, Stoll et al. additionally convert the input multi-view images to an implicit Sums-of-Gaussians based representation. The implicit model for the input images $I(c)$ of all cameras $c \in \{1 \dots n_c\}$, $n_c$ being the number of cameras, is obtained by assigning an Image Gaussian $G_i(x)$, $x \in \mathbb{R}^2$ to each image patch, resulting from the decomposition of the input image to regions of similar color. The algorithm to assign Image Gaussians decomposes each input frame into squared regions of coherent color by means of quad-tree decomposition. After setting a maximum quad-tree depth $T_{\text{qt}}$, the image domain of each camera view is recursively split in smaller regions. Then, neighboring patches of similar color, up to a color similarity threshold $T_{\text{fuse}}$, are fused together. The color similarity between two regions having colors $\eta_1 \in \mathbb{R}^3$ and $\eta_2 \in \mathbb{R}^3$ in the HSV space is computed by simply taking the Euclidean distance $||\eta_1 - \eta_2||^2$. This approach allows generating a limited number of $G_i$, distributed densely at color feature locations and sparsely in the remaining plain colored regions, saving space allocation as well as computational performance.

After this bottom-up approach is completed, an Image Gaussian is assigned to each patch, such that its mean $\mu_i \in \mathbb{R}^2$ corresponds to the patch center, and its standard deviation $\sigma_i$ to half of the side length of the square patch. The underlying average HSV color $\eta_i$ is also assigned as an additional attribute. Figure 2.14 shows an example of implicit 2D Image Gaussians-based representation for an image frame, where each $G_i$ is visualized with a disk cantered at $\mu_i$, with radius $\sigma_i$.

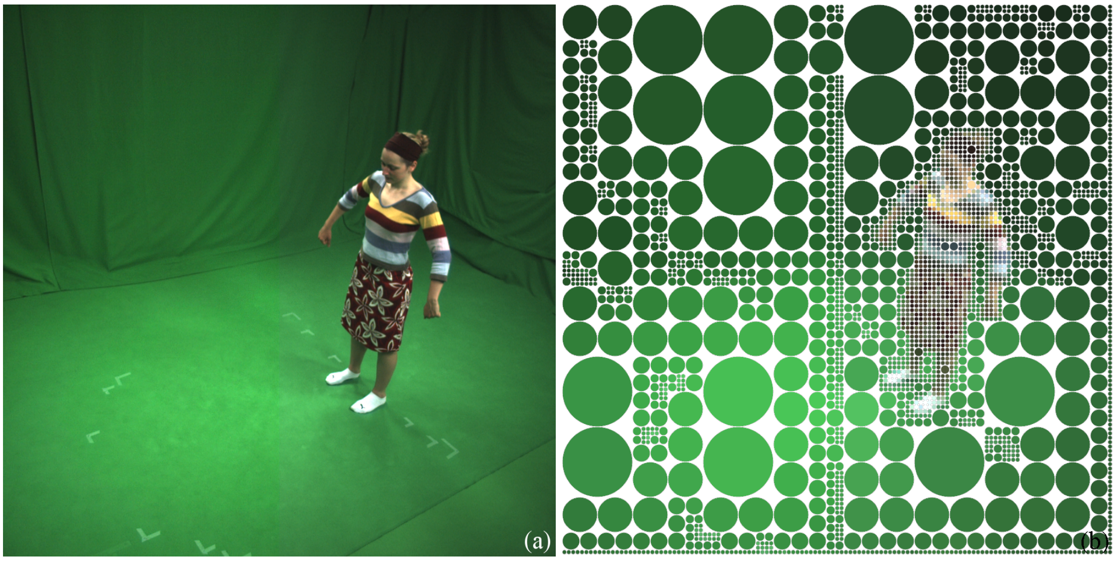


**Figure 2.14:** Example of implicit image representation. (a) The input image and (b) the estimated implicit representation as a collection of Image Gaussians (b).

#### 2.4.1.2 Pose Tracking

Analysis-by-synthesis strategies have been successfully employed for many tasks in computer vision, included pose estimation. General analysis-by-synthesis based approaches define a recognition process in which hypotheses are formulated and compared with input data until one of the hypotheses produces a match. In their work, Stoll et al. use analysis-by-synthesis to compare the multi-view image observations with the reprojected the human body model in different poses. In particular, the Gaussian-based approximation of the scene is used to formulate a smooth image-to-model consistency energy to effectively find the optimal actor pose configuration at each frame. The initial pose guess, either computed based on joint acceleration from previous frames or manually estimated, is projected into each camera view. Then, the consistency across all views of the 2D colored Volumetric Gaussians $G_m$ and the Image Gaussians $G_i$ is computed as:

$$\mathbf{E}(\mathcal{S}) = E_{\text{sim}} - E_{\text{temp}} - E_{\text{lim}} \tag{2.22}$$

where $E_{\text{sim}}$ is the core data term, $E_{\text{temp}}$ is the temporal smoothness term and $E_{\text{lim}}$ is used to check for parameter limits. In the following, we describe each term in detail.

**Similarity term** The core energy term estimates the similarity between the projected Volumetric Gaussians $G_m$ and Image Gaussians $G_i$ and is computed as:

$$E_{\text{sim}} = \frac{1}{n_c} \sum_{c=1}^{n_c} \left[ \frac{1}{\sum_{i=1}^{n_i^c} \Phi_{i,i}} \sum_{i=1}^{n_i^c} min \left( \sum_{m=1}^{n_m} \Phi_{i,m}, \quad \Phi_{i,i} \right) \right] \tag{2.23}$$

where $n_i^c$ is the number of Image Gaussians of camera $c$ and $\Phi_{i,m}$ measures the appearance and spatial similarity of an Image Gaussian $i$ and a Volumetric Gaussian $m$:

$$\Phi_{i,m} = T_{\Delta_c}(\delta_{i,m}) \left[ \int_{\Omega} G_i(x) G_m(x) \partial x \right]^2 \tag{2.24}$$

In the above equation $\delta_{i,m} = ||\eta_i - \eta_m||^2 \in \mathbb{R}^+$ measures the Euclidean distance between the colors in HSV color space, $\Delta_c$ is the maximum color difference allowed (beyond which the color similarity is set to 0), and $T_\Delta(\delta) : \mathbb{R} \to \mathbb{R}$ is the *Wendland* radial basis function [187] modeled by:

$$T_\Delta(\delta) = \begin{cases} \left(1 - \frac{\delta}{\Delta}\right)^4 \left(4\frac{\delta}{\Delta} + 1\right) & \text{if } \delta < \Delta \\ 0 & \text{otherwise} \end{cases} \tag{2.25}$$

Applying the $T_\Delta$ function on $\delta$ results in a smooth color similarity measure that is equal 1 if $\delta = 0$, i. e. $T_\Delta(0) = 1$ and smoothly decreases towards 0 as $\delta$ approaches $\Delta$, i. e. $\lim_{\delta \to \Delta} T_\Delta(\delta) = 0$.

A new approximation is proposed to handle self-occlusions without sacrificing the smoothness properties of their energy formulation. When multiple volumetric 3D Gaussians project onto the same screen space coordinate, their cumulative contribution in the sum can spoil the performance qualitatively, if not handled correctly. The authors employ a function minimization, visible in Equation 2.23, to account for the non-differentiability of the visibility function at occlusion boundaries. In a later co-authored work, described in the next section, a robust, fully differentiable visibility function is developed in the context of mocap.

One of the advantages of using a Gaussian representation is that the integral in Equation 2.24 has a closed-form solution, namely another Gaussian of the form:

$$\Phi_{i,m} = T_{\Delta_c}(\delta_{i,m}) 2\pi \frac{\sigma_m^2 \sigma_i^2}{\sigma_m^2 + \sigma_i^2} exp\left(-\frac{||\mu_i - \mu_m||^2}{\sigma_m^2 + \sigma_i^2}\right) \tag{2.26}$$

$$\Phi_{i,i} = \pi \sigma_m^2 \tag{2.27}$$

As explained in Section 2.2.3.2, the Volumetric Gaussians distribution in space is uniquely defined by the underlying skeletal joint configuration, which is in turn specified by the joint angles $s_p, \forall p = 1 \ldots |\mathcal{S}|$. Therefore, by optimizing for the model-to-image similarity in terms of the means $\hat{\mu}_m, \forall m = 1 \ldots n_m$, one actually estimates the optimal skeletal pose configuration for that frame. Optimization details are provided in Section 2.4.1.3.

**Temporal Term** The estimated skeletal joint angles in subsequent frames are regularized using a smoothness regularizer on the parameters:

$$E_{\text{temp}} = \sum_{p=1}^{|\mathcal{S}|} \left(\frac{1}{2}\left(s_p^{f-2} - s_p^f\right) - s_p^{f-1}\right)^2 \tag{2.28}$$

where $s_p^{f-2}$ and $s_p^{f-1}$ are the pose parameter values found at previous frames and $s_p^f$ is the parameter value at current frame. This term penalizes high acceleration in parameter space.

**Limits Term** Stoll et al. additionally include a check on the parameter limits:

$$E_{\text{lim}} = \sum_{p=1}^{|\mathcal{S}|} \begin{cases} ||l_p - s_p||^2 & \text{if } s_p \leq l_p \\ ||s_p - L_p||^2 & \text{else if } s_p \geq L_p \\ 0 & \text{otherwise.} \end{cases} \tag{2.29}$$

where $l_p$ and $L_p$ are respectively the lower and upper joint angle limits on parameter $s_p$.

#### 2.4.1.3 Optimization

In order to maximize $\mathbf{E}$, defined in Equation 2.22, the analytical derivatives with respect to the skeleton pose parameters $\mathcal{S}$ are computed. Then, conditioned gradient ascent is used to find the optimal skeleton pose in each frame. To improve the initial convergence to the correct pose, the initial actor skeletal configuration in frame 0 is typically manually initialized or assumed to be (close to) the neutral pose. In Chapter 3, we demonstrate the use of data-driven information to automatically initialize the actor pose and follow the motion without any manual intervention.

The approach shows impressive mocap results even in uncontrolled outdoor settings with a possibly dynamic background and occlusions. Thanks to the advantageous formulation, results are computed in real-time. The main disadvantage of this approach, which is due to the nature of the primitives, is the difficulty to accurately track twisting motions that typically characterize limbs and head. A possible solution to this issue is the explicit use of the information that is given by a skinned actor mesh. In particular, the surface mesh can be used to regularize the underlying skeleton motion, checking for sudden flips as well as for arising distortions or unnatural movements issues. Due to the plain skeleton visualization, i. e. dots and segments, unnatural effects are only visible when accompanied by skinning deformations.

#### 2.4.1.4 Actor Calibration

Calibrating the actor, i. e. computing a personalized body model of the performing actor, is an important step prior to tracking. The quality of the body model directly affects the accuracy of mocap. Stoll et al. [164] make extensive use of the analysis-by-synthesis approach, described in the previous section, to optimize for an enlarged set of the model parameters, which includes the Volumetric Gaussian mean, standard deviation and color, as well as bone lengths on top of the joint configuration. The algorithm decomposes the actor calibration problem in sub-problems aiming at fitting the template human body model into the reprojected actor's volume.

The reprojected actor's volume is obtained semi-automatically segmenting multi-view photographs of the actor in different poses, see Figure 2.13$(a)$. Alternatively, if a scan of the actor is available, it is possible to automatically generate multi-view silhouettes as seen from several fictional cameras with known calibration placed all around. The colors of the Volumetric Gaussians of the template human body model are initialized to the silhouette interior color. Then, the similarity energy in Equation 2.23 is iteratively minimized in terms of a subset of the model parameters: first the joint configuration (pose) alone, then the Volumetric Gaussian properties, i. e. means and standard deviations, and finally the bone lengths are estimated.

The initial skeletal pose can be automatically inferred by solving for single-frame model-to-image fitting as described in Section 2.4.1.3. Since the actor model is still uncalibrated at this point, manual corrections to the inferred pose are typically made to improve the estimated joint configuration. Once an initial skeleton pose has been computed, the set of reprojected Volumetric Gaussians is fit to the silhouettes across all views, in terms of the means $\hat{\mu}_m, \forall m$ and standard deviations $\hat{\sigma}_m, \forall m$. Finally, the obtained personalized volumetric layer is used to adjust the bone lengths.

To effectively find the corresponding bone lengths, additional scaling degrees of freedoms $\mathcal{S}_{\text{scaling}}$ are added to the skeletal parameter list, i. e. $\overline{\mathcal{S}} := \{\mathcal{S} \cup \mathcal{S}_{\text{scaling}}\}$. Scaling degrees of freedom uniquely

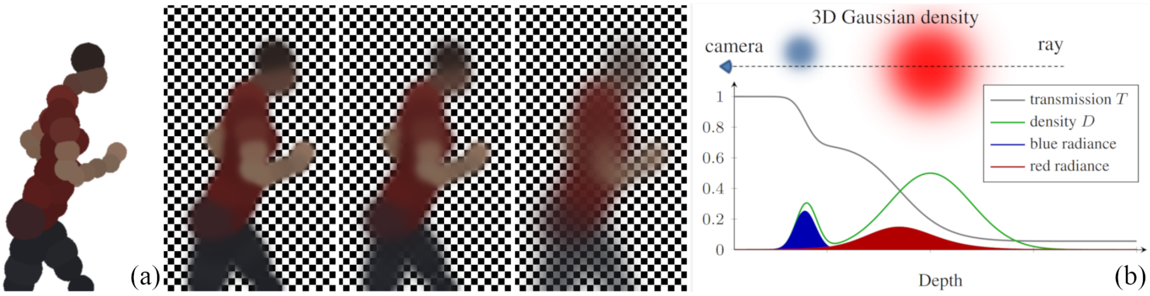


Figure 2.15: Translucent medium representation and visualization of radiance used to compute the visibility of the Volumetric Gaussians [143]. (a) Solid sphere actor model compared with its translucent medium density representation with increasing smoothness levels. (b) Ray-tracing of a Volumetric Gaussian density. The density along a ray is a sum of 1D Gaussians (green) and transmittance (gray) falls off from one for increasing optical depth. The radiance is the fraction of reflected light that reaches the camera (red and blue areas).

define joint scaling transformations $S_j(s_p)$ as:

$$S_j(s_p) := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & s_p & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.30}$$

Notice that the scaling factor $s_p$ only affects the $y$ axis, which by definition points towards the closest child joint along the bone connection. The effect of scaling is a bone stretch or compression, which exactly simulate variations in the bone lengths. Scaling transformations are concatenated to the remaining joint transformations, i. e. eventual translations and rotations, to form local transformation matrices, as explained in Section 2.2.2.1. The personalized bone lengths are found by computing the scaling parameters $\mathcal{S}_{\text{scaling}}$ that best fit to the new personalized volumetric layer within the convex hull, using Equation 2.23.

The actor calibration process can be iterated several times, i. e. re-estimating joint configuration, set of Volumetric Gaussians and bone length, until it converges to a good approximation of the actor's skeleton and volume, see Figure 2.13$(b)$. Generally, it is best to provide this embedding method several different poses, also called *calibration poses*, to best locate the position of all the skeletal joints. Four of the typically used calibration poses are shown in Figure 2.13$(a)$. The colors of the Volumetric Gaussians can be estimated at the end of the actor calibration cycle, by taking the underlying multi-view pixel color mean. This is obtained by computing the HSV color mean of the collection of all underlying pixels of each projected Volumetric Gaussian in all camera views. For each Volumetric Gaussian $\hat{G}_m$ and camera view, we typically take the set of pixels which lie in a 2D region of the image space enclosed in the approximated circle centered in the 2D $\mu_m$ with radius equal to $\sigma_m$. In case an actor's mesh is available together with associated vertex colors, the Volumetric Gaussians are assigned the close-by vertices color mean.

### 2.4.2 Improvements to Volumetric Tracking by Rhodin et al. [143]

A co-authored paper by Rhodin et al. [143] demonstrates the use of a volumetric Gaussian-based actor's body approximations to create an analytic, continuous and smooth visibility function that is differentiable everywhere in the scene. The main difficulty of related approaches lies in the handling of occlusions when projecting from 3D to the 2D space, where only visible parts of a 3D actor model should be considered when estimating e. g. model-to-image similarity. For simplicity, most other approaches check for camera visibility right before performing model projection to the image space. This kind of approaches are erroneous as they prevent occluded body parts from eventually reappearing in the view. Other approaches ignore multiple occluded object contribution in the energy [164].

The approach proposed by Rhodin and colleagues treats each Gaussian approximating the actor's volume as a translucent medium with smooth density distribution, see Figure 2.15$(a)$. The smooth Volumetric Gaussian visibility $\mathcal{V}_m$ along a pixel ray, from the camera origin $o$ towards a certain direction $n$ (depending on the pixel location) at distance $s$, is estimated as the accumulated transmittance $T$ of all encountered Volumetric Gaussians:

$$\mathcal{V}_m(o,n) := \sum_{m \in S_m} \lambda_m T(o,n,s) \hat{G}_m(o+sn) \tag{2.31}$$

In the above equation, $S_m$ is a compact sampling interval around the mean of each Volumetric Gaussian $G_s$ used to obtain an analytical closed form. Equation 2.31 is inspired by the radiance equation, where the sum of the product of the source radiance and transmission effectively measure the contribution of each Volumetric Gaussian to the pixel color, therefore approximating Volumetric Gaussian visibility, see plot in Figure 2.15$(b)$. The transmittance function $T$, i. e. , the percentage of light transmitted between two points in space, decays exponentially with the optical thickness of a medium, which is defined as the accumulated density $D$:

$$T(o,n,s) := \exp\left(-\int_0^s D(o+tn)\,\mathrm{d}t\right) \tag{2.32}$$

Using the Gaussian form of the density, one can rewrite the transmittance function in as:

$$T(o,n,s) := \exp\left(-\int_0^s \sum_m \hat{G}_m(o+tn)\,\mathrm{d}t\right) \tag{2.33}$$

The approximated transmittance of a Gaussian has an analytic closed form definition thanks to the appealing mathematical properties of Gaussians. Computationally, the proposed method has quadratic complexity in the number of Gaussians and is therefore relatively expensive. Nevertheless, the method delivers more accurate tracking and 3D geometry estimation w. r. t. previous methods based on non-differentiable visibility. In their experiment, the authors show that the new differentiable and well-behaved visibility function is essential for the success of the approach in markerless human mocap with very few cameras and/or strong occlusions. Setups with limited number of cameras or large occlusions highly rely on a proper visibility formulation to overcome failure at tracking time.

# Chapter 3

# Motion Capture

## 3.1 Introduction

Image-based human motion capture (mocap) is an important and long-standing problem in computer vision. In the past decades, there has been a constantly growing demand for robust human mocap algorithms from a wide range of application fields, such as computer animation, video effects (VFX) and biomechanical analysis. Recently mocap of performing humans in less constrained environment has gained increasing attention due to the potential of capturing casual actions in the wild. Despite the improved tracking quality of the previous methods in presence of challenging dynamic background, no known approach is able to reliably track the human motion in presence of changing illumination conditions, such as global light changes, or appearance changes, due to e. g. harsh shadows or view-inconsistent reflectance of pieces of apparel, which are typical in outdoor settings [21, 74, 24, 63, 154, 164, 53, 50, 141, 117, 197].

In this chapter, we describe a novel human mocap approach to reliably track the human action from multi-view videos with challenging temporally varying illumination (see an example in Figure 3.1), initially introduced in [144]. Our approach fits the personalized multi-layer human body model, described in Section 2.2, to a combination of an abstract intermediate image representation that is robust to lighting changes and 2D joint detections based on a convolutional neural network (CNN).

The main insight of our work is to factor out the lighting influence from the images by extracting the reflectance (or albedo) component of the image, and then perform motion tracking using the albedo component. To this end, for each frame of the multi-view sequence, the proposed method estimates the albedo component by solving a lighting-invariant segmentation problem. Our segmentation problem is phrased as a multi-labeling problem based on a pairwise Markov Random Field (MRF) [19], where each "material" of the human body is assigned a unique label that corresponds to its albedo. Lighting-invariance is achieved by dynamically updating the appearance information to reflect the changing lighting conditions. This is implemented by combining the image appearance and a pose prediction prior in a suitable way.

In order to increase the overall robustness, we additionally employ CNN-based joint position detections, which have been shown to have remarkable generalization ability [48, 204, 105, 171, 116, 133, 117]. Nevertheless, we have found that the CNN detector alone may be noisy and struggles under some conditions. In an attempt to combine the advantages of both worlds, we use an adaptive

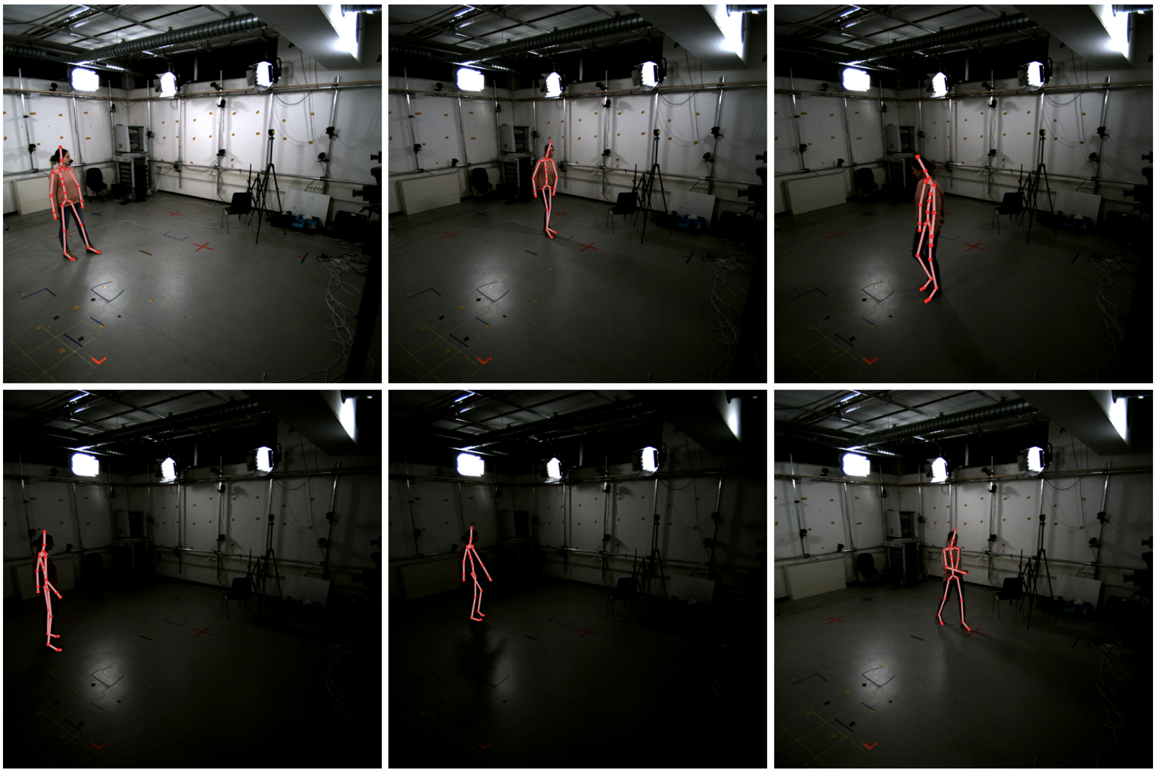


Figure 3.1: Our method estimates 3D human pose from multi-view image sequences. Even under extreme lighting conditions, our method is still able to track the motion successfully.

strategy for setting the relative weighting between the segmentation and the detectors, which are then embedded into the model-based mocap method described in Section 2.4.1. Our main technical contributions are:

1. the formulation of suitable time-varying segmentation problem that use frame- and view-dependent appearance costs in order to obtain lighting-invariant representations of the individual images,
2. the adaptive combination of an abstract intermediate image representation with CNN-based joint detectors, and
3. the integration of this combined information into a robust model-based tracker.

We demonstrate the effectiveness of our approach on several challenging sequences, including drastic lighting changes as well as harsh shadows. Our quantitative and qualitative results show that our approach accurately tracks the human pose and outperforms the existing methods in such challenging scenarios.

## 3.2 Related Work

Mocap of 3D humans has received a lot of attention in recent decades. Many vision-based markerless methods have been proposed to address this problem. A large portion of the literature focuses on generative model-based multi-view mocap, where the goal is to optimize the overlap of the projected 3D body model with multi-view images. Model-free approaches generally deliver worse qualitative performance compared to model-based approaches, as they employ automatically or semi-automatically estimated approximations of the tracked object. When the tracked object articulations are known a-priori (this is the case for a human model), the remaining dimensions, such as height, weight, lengths and so on, can generally be accurately estimated up to possible non-rigidity, e. g. loose clothing [142]. Most of the model-based approaches rely on manually estimated human models or 3D scans.

Given an input articulated body model of the actor, many methods for robust pose tracking rely on the silhouette input obtained by background subtraction [164, 13, 103, 62, 106, 9]. As the silhouette cue provides a strong constraint and drastically reduces the difficulty of the problem, those methods yield accurate mocap results. Multiple character tracking, even with complex interaction, has also been demonstrated. In the approach by Liu et al. [106] they first segment the provided foreground image region into different characters based on the color models and the pose prior. Then the different characters are tracked independently. In our method, the pose term for material segmentation is inspired by this work. However, our method segments the images with respect to different materials instead of different characters and does not rely on the silhouette input. This allows for automatic tracking as well as better estimation of the correspondences between the body model and the images. Although plausible results have been achieved by multiple character tracking methods, their application scenario is restricted to static backgrounds, since the background subtraction does not work well on dynamic background [107].

There also exist several generative model-based approaches that do not require explicit silhouette input [21, 74, 24, 63, 154, 164, 53]. Many approaches make extensive use of (dense) optical flow or region-based matching to track the articulated body motion [24, 63, 154]. More accurate methods build on top of indoor approaches by introducing a dynamic background segmentation step [21, 74]. The background is typically segmented based on temporal body motion cues obtained from previous frames. However, only results in well-controlled environments have been demonstrated. Cluttered background in a general outdoor scenario with illumination changes typically lead to tracking failures. Elhayek et al. [53] proposed an improved model-to-image consistency energy in weighted HSV color space, which is more resistant to intensity changes. Although the tracking failure due to illumination changes is alleviated to some extent, we have found in our experiments that using the color consistency energy in HSV space alone is not enough to handle unconstrained complex lighting conditions for outdoor tracking. To handle varying illumination, Wu et al. [193] proposed to simultaneously estimate the illumination and track the motion in a joint optimization framework. However, since the illumination estimation and motion tracking is performed in an alternating manner, and each step relies on the previous step being correct, their method is not able to recover from errors. Besides, it is also worth mentioning that the computational complexity of their method is rather high due the illumination estimation.

In contrast to generative model-based methods, data-driven (discriminative) methods address the 3D pose estimation problem from the perspective of image feature extraction, regression or classification [1, 85, 148, 121, 156]. With the enormous success of deep learning methods and the thus resulting growing popularity, many CNN-based approaches have recently been proposed to predict 3D human pose from monocular images. A common approach is to lift the 2D joint prediction to 3D using temporal constraints and/or pose priors [205, 202, 203], while many other methods directly estimate the 3D pose from single images [48, 204, 105, 171, 116, 133, 117]. Even real-time 3D pose estimation has been achieved [117]. The CNN-based methods are typically more robust to illumination changes in the outdoor scenario than generative methods due to the good generalization

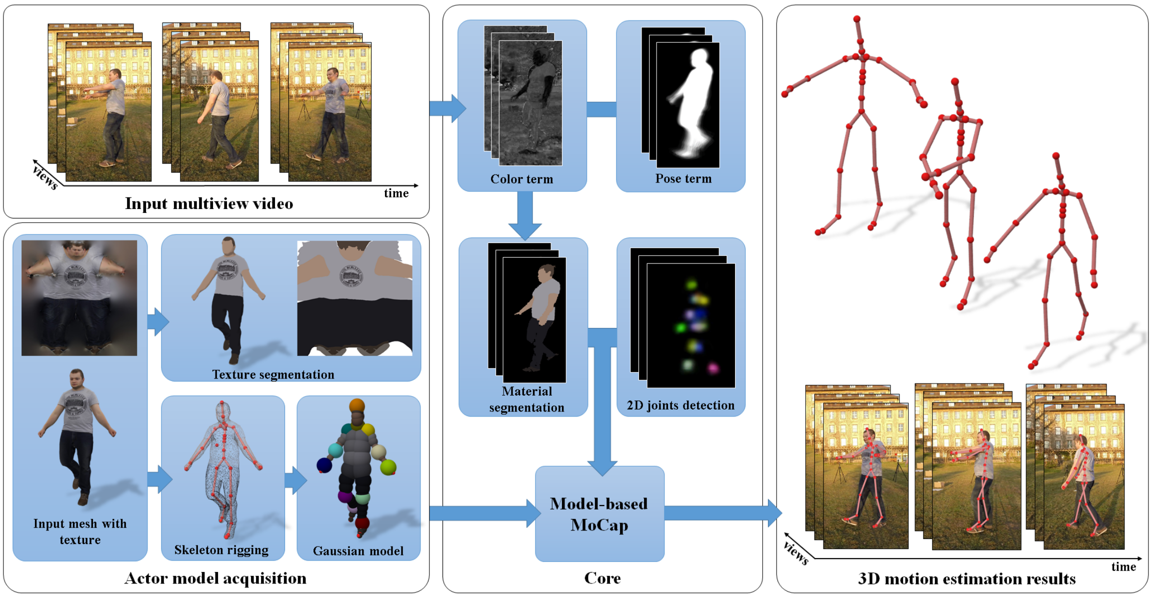


**Figure 3.2:** Pipeline of our method.

ability of the deep neural network, but those monocular-based methods are usually less accurate since they suffer from the inherent depth ambiguity. To address such problems, multi-view-based discriminative methods [160, 26, 6, 174] have been proposed. However, they typically have lower temporal stability than generative methods and, more importantly, they use a simplified body model with only few degrees of freedom.

Several recent methods combine the generative model and the discriminative approach to benefit from the merits of both sides [142, 53, 142, 15]. In particular, the methods in [142] and [53] share a similar outdoor multi-view mocap setting with our proposed method. In contrast to their methods, our approach iteratively tracks the skeletal motion and estimates the intrinsic segmentation to factor out the illumination changes. Our experimental results demonstrate that our approach outperforms the existing methods under varying illumination.

## 3.3 Overview

In this section, we describe our model-based mocap approach, which is summarized in Figure 3.2. Given a sequence of multi-view calibrated and synchronized images capturing the action of a single actor in varying lighting conditions, our goal is to consecutively track the human body pose in each frame, resulting in the temporally coherent skeletal motion across the entire sequence. This approach leverages the multi-layer human body model, described in Section 2.2.

To handle complex lighting conditions, for each frame we estimate an illumination-robust segmentation of the images by extracting the corresponding approximate albedo channel, and then incorporate the segmentation into the skeleton tracking. Segmented frames are an optimal input to appearance-based mocap approaches, since they are mostly invariant to illumination changes. Together with the segmentations, we provide 2D sparse joint detections as input to the mocap approach, that are estimated offline with CNN. As shown in Section 3.7, detections provide additional clues in presence of challenging lighting conditions and body poses, and are used to further improve the reliability of

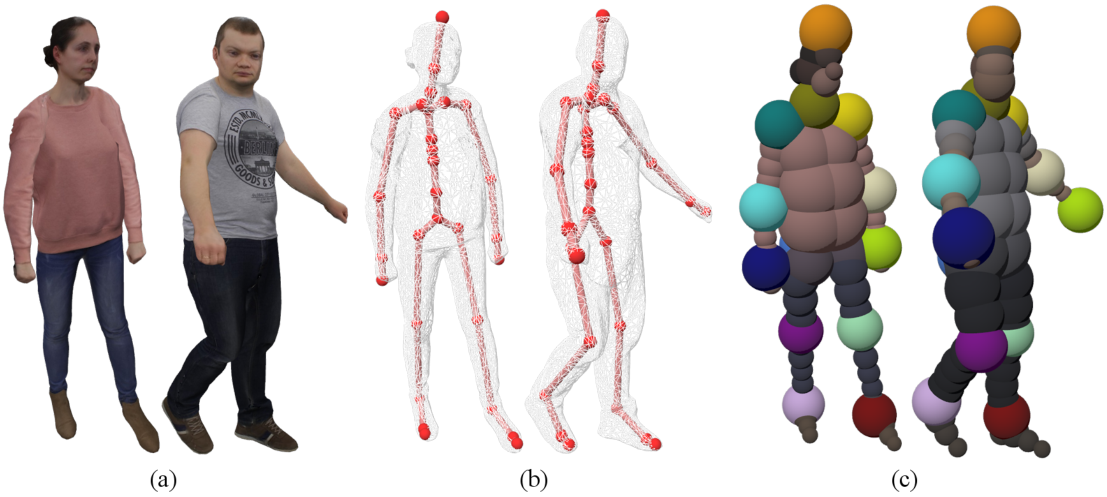


**Figure 3.3:** Examples of 2 input body models with all their layers. (a) Textured 3D models, (b) triangulated mesh model with kinematic skeleton, (c) Gaussian blob model showing the body material Gaussians as well as the special discriminative Gaussians attached to the 14 most prominent joints, i. e. head, neck, shoulders, elbows, wrists, hips, knees, and ankles.

the estimated body configurations.

In the rest of this chapter, we first discuss our approach for appearance model acquisition (Section 3.4), then describe our segmentation approach (Section 3.5), and finally present the skeleton tracking method (Section 3.6).

## 3.4 Scene Model

In this section, we describe our input actor representation, which we extensively use to estimate lighting-invariant image segmentation.

### 3.4.1 Actor

Our approach relies on the three-layer person-specific human body model outlined in Section 2.2. In this chapter, the volumetric layer is equipped with additional discriminative Gaussians attached to the most prominent joints in the skeleton, i. e. head, neck, shoulders, elbows, wrists, hips, knees, and ankles, see Figure 3.3$(c)$. These are used to enable skeleton matching to the 2D CNN-based joint detection heatmaps, as described in Section 3.6.

The detailed layer consists of a triangulated mesh with associated texture segmentation, see Figure 3.2. While retaining the valuable texture quality, we reduced the number of triangles for performance purposes. In order to define the model materials, we first semi-automatically segment the texture into different regions according to the albedos of different materials (e. g. skin, shirt, etc.) as in Figure 3.4$(c)$. To this end, we first apply the image smoothing method of [196] on the texture image to remove the high frequency shading components while preserving features in the image. Then we manually annotate the smoothed texture image, resulting in the material texture segmentation

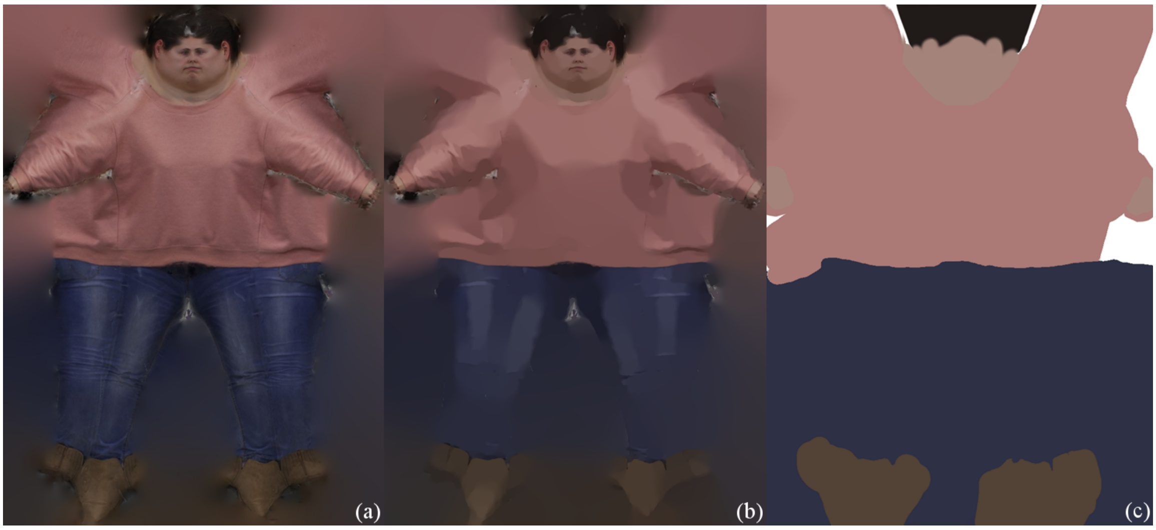


**Figure 3.4:** Visualization of the material clustering process on the texture map. (a) Initial texture image, (b) smoothed edge-preserving texture image to facilitate material clustering and (c) final material clustering.

that assigns a unique label to each material, see Figure 3.4$(c)$. Note that while we assume that each material is homogeneous, non-homogeneous parts (e. g. a shirt with a logo) can be modeled by introducing sub-materials.

Since in this chapter we are only interested in the skeleton motion estimation, i. e. discarding the surface details, we keep the number of materials limited to a small number and combine together slightly differently textured pieces of apparel or skin features. To this end, the shirt logo visible in Figure 2.8$(b)$ is fused together with the shirt material, while eyes and mouth are considered to be one material, i. e. skin material, see Figure 3.4$(c)$. All our body models have 5 materials in total, namely skin, hair, shirt, pants and shoes.

## 3.5 Lighting-invariant Segmentation

This section describes how to obtain a lighting-invariant representation (albedo channel) of a multi-view frame $(I_1^f, \ldots, I_{n_c}^f)$ for each camera view $c := 1 \ldots n_c$ and time $f := 1 \ldots F$. Notice that in our setting the same material may have a different color appearance when viewed from different cameras. This is the case for, e. g. , apparel with (piece-wise) non-Lambertian reflectance behavior, such as highly-specular apparel. A similar global effect may arise in case of incomplete or inconsistent photometric calibration of the cameras, which causes the same material to have different color appearance depending on the view. Consequently, we treat each camera view $c$ independently, and thus assume in the remainder of this section that $c$ and the frame $f$ are fixed. Hence, for notational convenience we omit the $c$ and $f$ indices and write $I$ instead of $I_c^f$.

The objective of the lighting-invariant segmentation is to assign to each pixel $i$ of image $I$ a label $\ell_i \in \mathcal{L}$ that indicates which material is seen in that pixel. The set of foreground materials includes all semi-automatically segmented materials of the actor's surface (e. g. shirt, skin, pants) with indices $\ell := 1 \ldots |\mathcal{L}| - 1$, described in the previous Section 3.4.1. We additionally consider the background to be a material and add it to the end of the labels' list, i. e. with index $|\mathcal{L}|$. A labeling $\boldsymbol{\ell} \in \mathcal{L}^{|I|}$ for

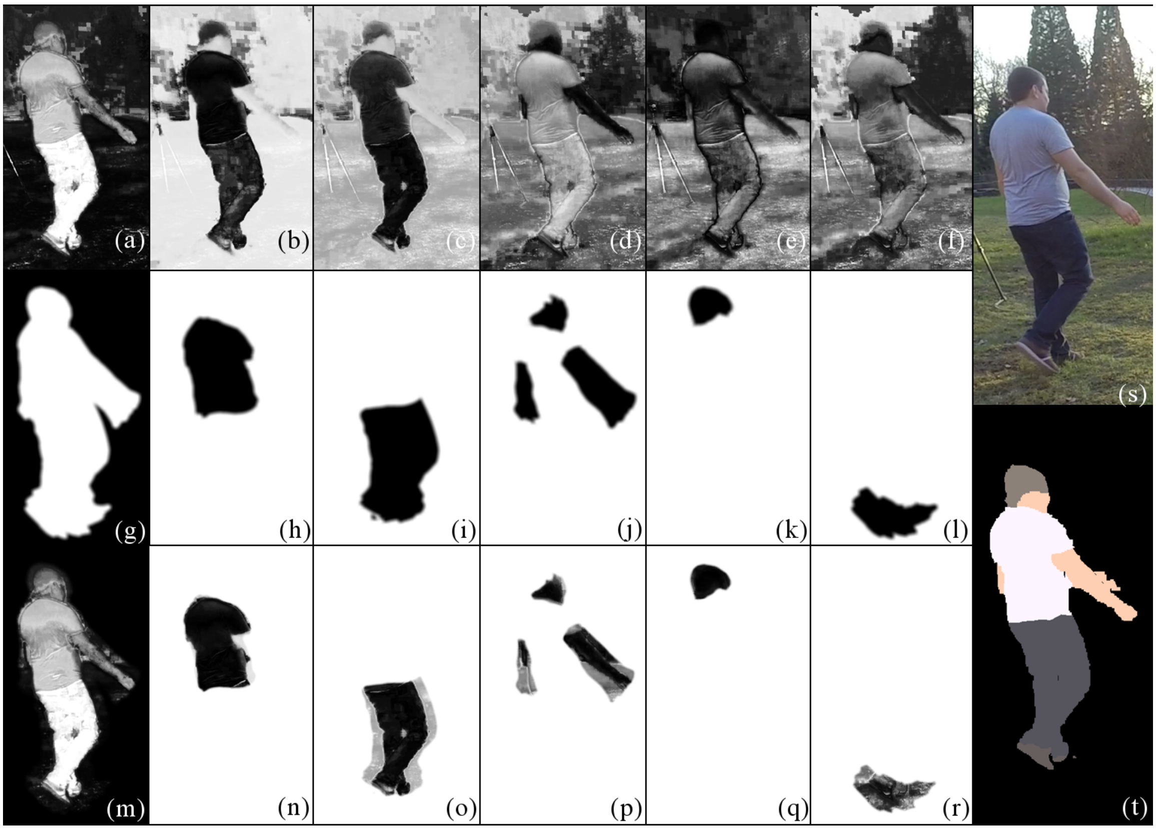


**Figure 3.5:** Visualization of color, pose and combined costs. Top row (a-f) shows the appearance costs, middle row (g-l) shows the pose costs, and bottom row (m-r) the combined appearance-pose costs for different materials: (a,g,m) background, (b,h,n) shirt, (c,i,o) pants, (d,j,p) skin, (e,k,q) hair and (f,l,r) shoes. The input frame is shown on the top-right (s), while the resulting segmentation is depicted below in (t).

image $I$ is obtained by minimizing an energy of the form

$$E(\boldsymbol{\ell}) = \sum_{i=1}^{|I|} E_i(\ell_i) + \sum_{i \sim j} E_{ij}(\ell_i, \ell_j)\,, \tag{3.1}$$

where $E_i(\ell)$ is the data term that measures the cost for assigning label $\ell \in \mathcal{L}$ to pixel $i$, and $E_{ij}$ is the smoothness term that penalizes neighboring pixels $i$ and $j$ that are assigned different labels ($i \sim j$ indicates that $i$ and $j$ are neighbors).

The data term combines the appearance $E_i^{\mathrm{a}}(\ell)$ and pose cost $E_i^{\mathrm{p}}(\ell)$ defined for each pixel $i$ as follows:

$$E_i(\ell) = E_i^{\mathrm{a}}(\ell) \cdot E_i^{\mathrm{p}}(\ell)\,, \tag{3.2}$$

The appearance term measures the visible color similarity to the material color. In fact, we make use of an enhanced appearance representation of pixels rather than using their simple RGB color, which is shown to be more robust to illumination effects. A visualization of the appearance costs for each material can be seen in Figure 3.5, first row (reference input image in Figure 3.5$(s)$). In the gray-scale figure darker colors correspond to lower costs for the corresponding material. For instance, the skin appearance costs in Figure 3.5$(d)$ alone can be used to successfully identify arms and face as skin, since the pixel at the corresponding regions have the lowest costs. However, misleading outliers with similar appearance in the background, e. g. the shoes, would cause a segmentation approach for the skin to be infeasible at this stage.

To cope with similar looking materials, we additionally define the pose term. The pose term robustly identifies possible material locations based on previously observed poses, see Figure 3.5, middle row. Body motion is assumed to be sufficiently smooth between subsequent frames. The pose term is designed such that it considers a (sufficiently large) uncertainty range of possible poses, making the aforementioned assumption valid in most of the cases. By weighting the appearance term with the pose term, most of the far regions in terms of pixel distance from the query material (e. g. , the skin material in the example) are correctly set to high cost, see Figure 3.5 last row. In this case, segmentation leads to accurate material labeling, see Figure 3.5($t$).

The appearance representation as well as the detailed description of the energy term is described in Section 3.5.2. Details for the pose term are discussed in the next Section 3.5.1.

### 3.5.1 Pose costs

**Algorithm 1:** Function to estimate pose costs $E^{\mathrm{p}}$ for all materials of a single camera view $c$, given as input skinned mesh $M$ with pose parameters $\mathcal{S}$.

```
function $E^{\mathrm{p}} := PoseCosts(M, \mathcal{S}, c)$:
for $n := 1$ to $50$ do
    for $p := 1$ to $|\mathcal{S}|$ do
        $\widetilde{s}_p := \min(\max(rnd(s_p, \sigma), l_p), L_p)$;
    end
    $\widetilde{\mathcal{S}}_n := \{\widetilde{s}_p, \forall p := 1 \dots |\mathcal{S}|\}$;
    for $\ell := 1$ to $|\mathcal{L}| - 1$ do
        $M.enableMaterial(\ell)$;
        $H_\ell := H_\ell + M.render(\widetilde{\mathcal{S}}_n, c)$;
    end
end
for $\ell := 1$ to $|\mathcal{L}| - 1$ do
    $H_{|\mathcal{L}|} := H_{|\mathcal{L}|} + (1 - H_\ell)$;
end
$E^{\mathrm{p}} := \{1 - k \cdot H_\ell, \forall \ell := 1 \dots |\mathcal{L}|\}$
```

In order to define body pose costs at a given frame $f$, we start by predicting the body pose configuration $\mathcal{S}^f$, which is obtained based on the acceleration computed from the pose parameters of the previous two frames, i. e. $\mathcal{S}^{f-1}$ and $\mathcal{S}^{f-2}$:

$$\mathcal{S}^f := \mathcal{S}^{f-1} + \frac{\mathcal{S}^{f-1} - \mathcal{S}^{f-2}}{2} \tag{3.3}$$

The pose parameters for the first frame $\mathcal{S}^1$, are either user-specified or assumed to be all equal to zero as a first guess, and $\mathcal{S}^2 := \mathcal{S}^1$. Figure 3.14 shows a successfull fast recovery example when $\mathcal{S}^1$ remains unspecified. The predicted pose, computed using the above formula, gives already a good approximation of the actual human pose in the multi-view images, as most of the human natural movements have close-to-constant acceleration across frames. However to be able to capture some pose variation, we additionally allow a window of uncertainty around each predicted joint parameter.

In particular, we sample 50 random pose parameters from a Gaussian distribution around the current pose parameter prediction $\mathcal{S}^f$. Random joint parameters $\widetilde{\mathcal{S}}_n := \{\widetilde{s}_p : p \in \{1 \dots |\mathcal{S}|\}\}$ are chosen in respect of the joint limits $[l_p, L_p]$, as shown in Function 1, lines 3-6. The standard deviation $\sigma$ for random sampling $rnd(s_p, \sigma)$ is typically chosen aiming at covering a sufficiently large space of the complex body pose variations around the predicted pose, and is set to be 10cm for the global skeleton translation components, and 0.2 radians for the rotational skeleton joints. The effect of accumulated small rotations going from parent to children joints ,e. g. , going from the shoulder to the wrist, result in naturally larger end-points movement, especially visible for the shoulder-wrist chain in Figure 3.5$(j)$.

Given $\mathcal{S}^f$ and a set of random joint parameters $\widetilde{\mathcal{S}}_n^f, \forall n := 1 \ldots 50$, we estimate for each material $\ell \in \mathcal{L}$ a binary pose probability image $H_\ell : \Omega \to \{0,1\}$, where $\Omega$ is the 2D image pixel space and $H_\ell(x_i) := 1$ if pixel $i$, with position $x_i \in \Omega$, belongs to material $\ell$, and $H_\ell(x_i) := 0$ otherwise. $H_\ell$ can be easily and efficiently estimated using mesh rendering techniques. To this end, we project all the 50 estimated random mesh configurations, resulting from skinning based on the random skeleton poses $\widetilde{\mathcal{S}}_n^f$, onto the same image plane using the camera projection matrix of the chosen view. To distinguish between different material projections, we assign $|\mathcal{L}|-1$ (foreground) textures to the mesh. Each texture is colored with white color value $(1,1,1,1)$ in the RGBA color space at the corresponding material locations represented. The remaining surface locations, assigned to different materials, are set to transparent $(0,0,0,0)$. For instance, the texture for the skin material is white on the arms and face, and transparent in the remaining surface mesh locations. We, then, render each texture projection in separated blending buffers initialized to $(0,0,0,1)$, one per foreground material, taking care of enabling depth test, disabling lighting, and setting the blending function such that new overlaying renderings are summed up in the buffer. This results in accumulated material buffers $H_\ell : \Omega \to \{0,1\}, \forall \ell := 1 \ldots |\mathcal{L}|-1$, which are black everywhere except at those locations filled by randomly re-posed re-projected material textures, representing in other words binary pose probability masks $\in \{0,1\}$. The binary mask for the background $H_{|\mathcal{L}|} : \Omega \to \{0,1\}$ is obtained by summing up all resulting material masks and inverting their values, see Function 1, lines 12-14 . The method used to estimate binary pose probability images $H_\ell$ gives a view, pose and shape guided approximation, which is more accurate than considering e. g. fixed-sized 2D bounding boxes around each material re-projection.

The pose cost masks for each material are finally obtained by inverting the probability masks and by slightly smoothing the borders using a small Gaussian smoothing kernel $k$, simulating smoothly decreasing/increasing costs around the border's mask:

$$E_i^{\mathrm{p}}(\ell) := 1 - k * H_\ell \,. \tag{3.4}$$

### 3.5.2 Frame-dependent appearance costs

Given a definition for pixel $i$ appearance $\Psi(x_i), \forall i := 1 \ldots |I|$ and material $\ell$ appearance $\Psi(\ell), \forall \ell := 1 \ldots |\mathcal{L}|$, the cost of assigning $\ell$ to $i$ is given by:

$$E_i^{\mathrm{a}}(\ell) := dist(\Psi(x_i), \Psi(\ell)) \,. \tag{3.5}$$

where the function $dist$ computes the distance between their appearances. While the pixel appearance $\Psi(x_i)$ can be defined by their color at the corresponding pixel location, the material appearance $\Psi(\ell)$ needs to be re-estimated at each frame according to the new material location in the images and its possibly changing visual appearance due to varying lighting conditions. For estimating $\Psi(\ell)$, using the un-smoothed mask $H_\ell$ introduced in the previous section, we extract all pixel appearance vectors at the reprojected location of material$\ell$, which we denote as $X_\ell := \{\Psi(x_i) : H_\ell(x_i) = 1\}$. Then, we

**Algorithm 2:** Function to estimate appearance costs $E^{\mathrm{a}}$, materials mean $\mu_\ell$ and covariance $C_\ell$ for all materials of a single view, given input RGB image $I$ and pose costs $E^{\mathrm{p}}_\ell$.

```
function $[E^a, \mu_\ell, C_\ell] := AppearanceCosts(I, E^p_\ell)$:
$\Psi := Rgb2Feature(I)$;
for $\ell := 1$ to $|\mathcal{L}| - 1$ do
    $X_\ell := \{\Psi(i) : 1 - E^{\mathrm{p}}_\ell \geq \frac{1}{2}\}$;
    $\mu_\ell := \arg\min_y \sum_{x \in X_\ell} \|x - y\|_2$;
    $C_\ell := \frac{1}{|X_\ell|-1} \sum_{x \in X_\ell} (x - \mu_\ell)(x - \mu_\ell)^T$;
    for $i := 1$ to $|I|$ do
        $E^{\mathrm{a}}_i(\ell) := (\Psi(i) - \mu_\ell)^T C_\ell (\Psi_i - \mu_\ell)$;
    end
end
$\Psi_{\mathrm{bg}} := [\Psi^T, E^{\mathrm{a}}_i(\ell_1), \ldots, E^{\mathrm{a}}_i(\ell_{|\mathcal{L}|-1})]^T$;
$X_{|\mathcal{L}|} := \{\Psi_{\mathrm{bg}}(i) : 1 - E^{\mathrm{p}}_{|\mathcal{L}|} \geq \frac{1}{2}\}$;
$\mu_{|\mathcal{L}|} := \arg\min_y \sum_{x \in X_{|\mathcal{L}|}} \|x - y\|_2$;
$C_{|\mathcal{L}|} := \frac{1}{|X_{|\mathcal{L}|}|-1} \sum_{x \in X_{|\mathcal{L}|}} (x - \mu_{|\mathcal{L}|})(x - \mu_{|\mathcal{L}|})^T$;
for $i := 1$ to $|I|$ do
    $E^{\mathrm{a}}_i(|\mathcal{L}|) := (\Psi_{\mathrm{bg}}(i) - \mu_{|\mathcal{L}|})^T C_{|\mathcal{L}|} (\Psi_{\mathrm{bg}}(i) - \mu_{|\mathcal{L}|})$;
end
```

compute the *geometric median* $\mu_\ell \in [0,1]^3$ of the vectors in $X_\ell$, which is given by

$$\mu_\ell := \mu(X_\ell) = \arg\min_y \sum_{x \in X_\ell} \|x - y\|_2 \,, \tag{3.6}$$

The above formula admits an efficient solution [186, 79]. Note that when the $\ell_2$-norm in (3.6) is replaced by the squared $\ell_2$-norm, one obtains the mean, whereas using the $\ell_1$-norm results in the coordinate-wise median. The geometric median gives a robust and reliable estimate of material visual appearance even in presence of inaccuracies in the estimated pose mask $H_\ell$, outperforming median or average. In addition to $\mu_\ell$, we estimate a robust *covariance matrix* of $X_\ell$ based on the geometric median $\mu_\ell$, computed as

$$C_\ell := \frac{1}{|X_\ell|-1} \sum_{x \in X_\ell} (x - \mu_\ell)(x - \mu_\ell)^T \,. \tag{3.7}$$

The geometric median $\mu_\ell$ and the covariance matrix $C_\ell$ together define the material appearance, i. e. $\Psi(\ell) := \{\mu_\ell, C_\ell\}$, and are updated at each time frame $f$.

Figure 3.6 illustrates appearance costs $E^{\mathrm{a}}_i$ when changing the definition of appearance $\Psi(x_i)$ for each pixel as well as the formula for assessing the costs, i. e. $dist$. When using simple RGB colors, i. e. $\Psi(x_i) := RGB(x_i)$, and norm distances, i. e. $dist(\Psi(x_i), \Psi(\ell)) := ||\Psi(x_i) - \mu_\ell||^2$, we obtain the pixel costs depicted in Figure 3.6$(a, f, k)$. Notice that in this case we completely ignore valuable information on the material given by its covariance $C_\ell$. The lack of complete information as well as the use of the weak RGB color space for the distance computations, result in unreliable appearance costs, since the target material in the pixels has mixed costs. The skin costs depicted in Figure 3.6$(a)$, for instance, are very sensitive to shading, e. g. around the arms, and have even higher costs on the arms and face locations than certain far sides in the background. The pants costs in Figure 3.6$(f)$ are

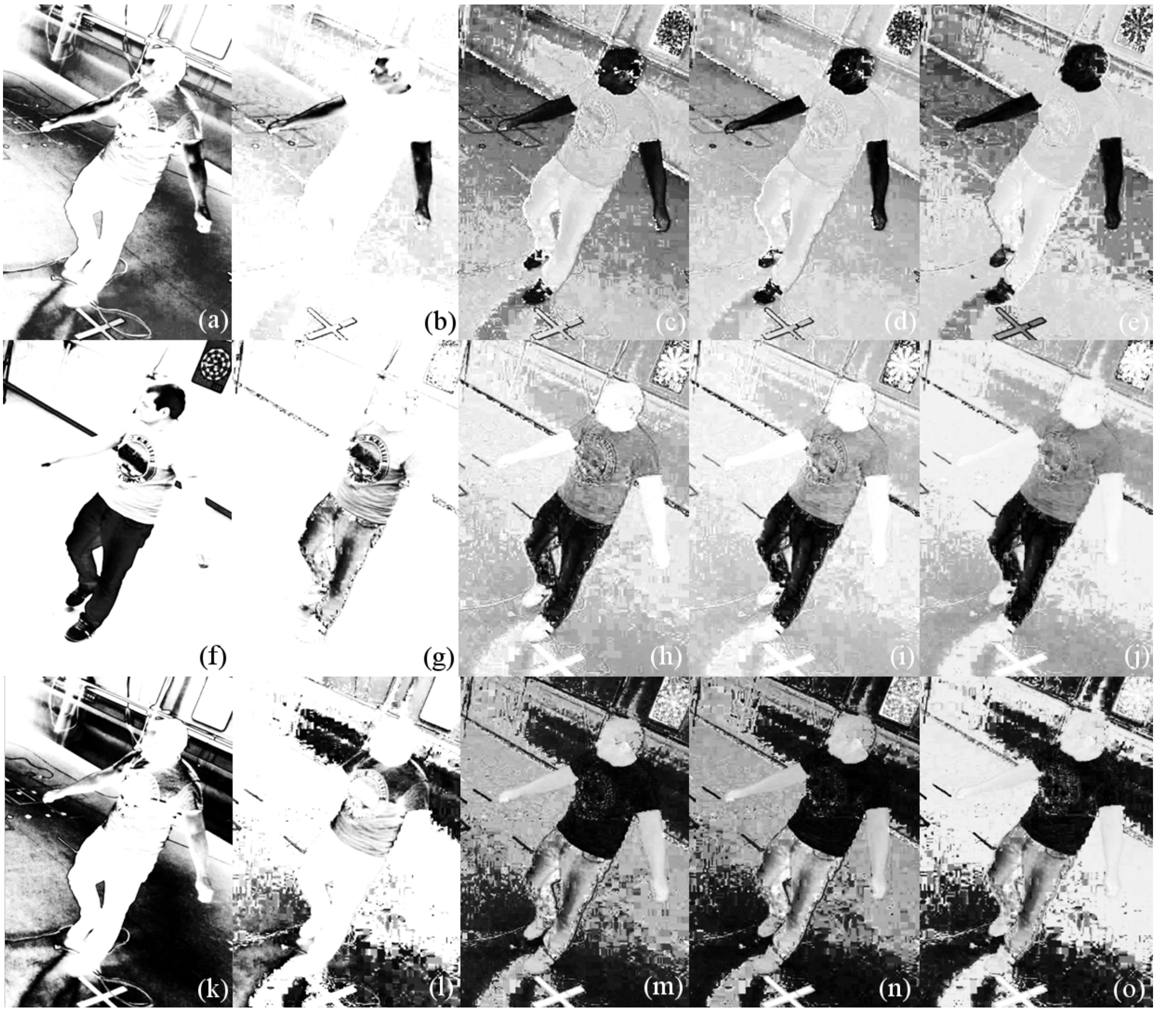


**Figure 3.6:** Color costs (the darker the lower) for skin (a,b,c,d,e), pants (f,g,h,i,j) and shirt (k,l,m,n,o). Each column from left to right shows the respective costs using simple RGB space distance (a,f,k), HSV space distance (b,g,l), Mahalanobis distance from the average material color (c,h,m), Mahalanobis distance from the median material color (d,i,n) and Mahalanobis distance from the geometric median material color (e,j,o).

better estimated, since the pants are well marked till the boundaries. Still, outliers on the shirt and on the background are present. The illustrated costs for the shirt material in Figure 3.6$(k)$ show a complete failure, due to the high RGB similarity of the gray shirt to the gray floor.

When replacing the RGB space with the HSV color space instead, we gain more robustness to shading, since similar colors that are differently illuminated have smaller distances than in the RGB space. Thanks to this fact, all depicted costs are much improved, see Figure 3.6$(b,g,l)$. Especially for the skin material Figure 3.6$(b)$ the simple conversion to HSV space mostly resolves the shading problem around the arms, and better captures the face, eliminating most of the surrounding outliers, i. e. setting the remaining pixels costs to higher values (closer to white). While the costs for the pants material itself (Figure 3.6$(g)$) looks worse than in the RGB case, the remaining body parts and the background have now been well separated from the target material in terms of costs.

In order to improve upon the still limited lighting-invariance that we gain in the HSV color space, we resort to keeping the perceived color information as well as the saturation, given respectively by the *hue* and *saturation* components, ignoring the shading *value* component. The feature image $\Psi : \Omega \to [0,1]^3$ is then obtained by setting the first and second components to the sine and cosine of the *hue*, respectively, and the third component to the *saturation*. The choice of sine and cosine helps to better deal with the periodicity of the *hue* component and, therefore, to ease further processing. On top of that, rather than using simple norm distances for the function *dist*, we consider a (frame-dependent) robust version of the *Mahalanobis distance* to measure the discrepancy between the observed appearance vector $\Psi(x_i)$ of a given pixel $i$ and material $\Psi(\ell)$ appearance. Using both the dynamically updated geometric median $\mu_\ell$ and the covariance $C_\ell$, for all foreground materials $\ell_1,\dots,\ell_{|\mathcal{L}|-1}$, we define the appearance cost $E^{\mathrm{a}}$, in the spirit of the *Mahalanobis distance*, as

**Algorithm 3:** Function to convert an RGB image to feature vector.

```
function $\Psi := Rgb2Feature(I)$:
for $i := 1$ to $|I|$ do
    $[H,S,V] := Rgb2Hsv(I(i)) \in \{(0,2\pi),(0,1),(0,1)\}$;
    $\Psi(i,1) := \frac{sin(H)+1}{2}$;
    $\Psi(i,2) := \frac{cos(H)+1}{2}$;
    $\Psi(i,3) := S$;
end
```

$$E_i^{\mathrm{a}}(\ell) := (\Psi(x_i) - \mu_\ell)^T C_\ell^{-1} (\Psi(x_i) - \mu_\ell)\,. \tag{3.8}$$

Using the new outlined color space together with the new appearance distance definition above, we obtain much more accurate appearance costs, see Figure 3.6, columns 3-5, where the corresponding target material pixels are assigned the lower costs. The figure compares resulting costs $E_i^{\mathrm{a}}(\ell)$ computed using the material appearance average (column 3) with median (column 4) and with the chosen geometric median (column 5), visually proving the effectiveness of the latter against the remaining two. The shirt costs in Figure 3.6$(m,n,o)$ still presents some limitation at this stage, due to undistinguished appearances of the shirt and the floor. Nevertheless, this is fixed when combining the appearance term with the pose term.

Since the background (having label $\ell_{|\mathcal{L}|}$) is in general inhomogeneous, a single-modal model in the $\Psi$-feature space as assumed in (3.8) is inappropriate. Instead of using a multi-modal model, for the background we consider a lifted feature vector $\Psi_{\mathrm{bg}}$ obtained by augmenting $\Psi$ with the already predicted foreground costs, i. e.

$$\Psi_{\mathrm{bg}}(x_i) = [\Psi(x_i)^T, E_i^{\mathrm{a}}(\ell_1),\dots,E_i^{\mathrm{a}}(\ell_{|\mathcal{L}|-1})]^T\,. \tag{3.9}$$

By embedding the computed foreground costs into the $(3{+}(|\mathcal{L}|-1))$-dimensional background feature vector $\Psi_{\mathrm{bg}}$, we have found that a single-modal model in this higher-dimensional feature space is able to provide sufficient discriminability for the background. The background appearance costs are then computed as in (3.8) with $\Psi_{\mathrm{bg}}$ in place of $\Psi$, and $\mu_{|\mathcal{L}|}$ and $C_{|\mathcal{L}|}$ being computed from the predicted background mask using $X_{|\mathcal{L}|} = \{\Psi_{\mathrm{bg}}(x) \;:\; H_{|\mathcal{L}|}(x) = 1\}$. The used method for the background segmentation is proved to be superior to alternative ways involving for instance the direct use of the pose costs, as shown in Figure 3.7$(c)$ compared to $(d)$.

Results of the segmentation when using the appearance costs only (ignoring the pose costs) are shown in Figure 3.8. Notice that, thanks to the robustness of our approach, all foreground materials have been segmented successfully, with the exception of some isolated pixels present in the background. These are fixed when additionally considering the pose costs.

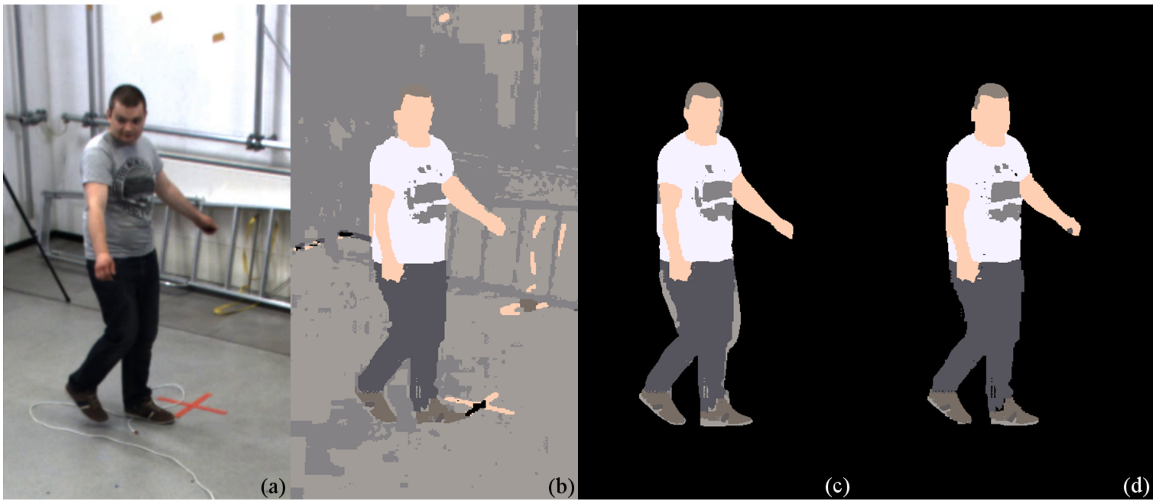


**Figure 3.7:** Background segmentations using different approaches. (a) Input image, (b) segmentation resulting by assigning background labels the maximum costs (in this case almost no pixel is assigned the background label), (c) segmentation with label costs taken from the pose prior alone (silhouette mismatches arise), (d) segmentation using the proposed background labeling costs.

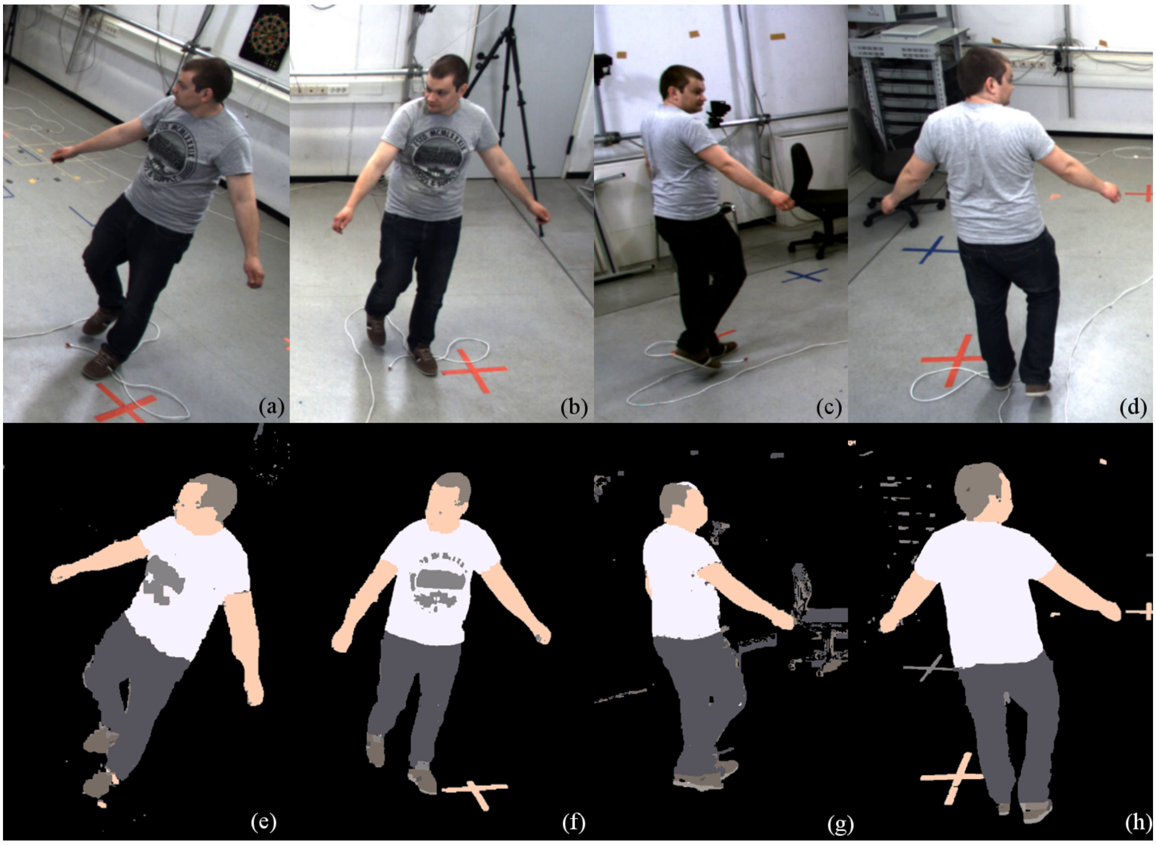


**Figure 3.8:** Segmentation results considering only the appearance cost term (without the pose costs): (a,b,c,d) input views, (e,f,g,h) resulting segmentations.

### 3.5.3 Smoothness term

In order to achieve a piece-wise constant labeling $\boldsymbol{\ell} \in \mathcal{L}^{|I|}$ of image $I$, we use a smoothness term that penalizes neighboring pixels that are assigned different labels. The Potts model $s$ [139] is a robust

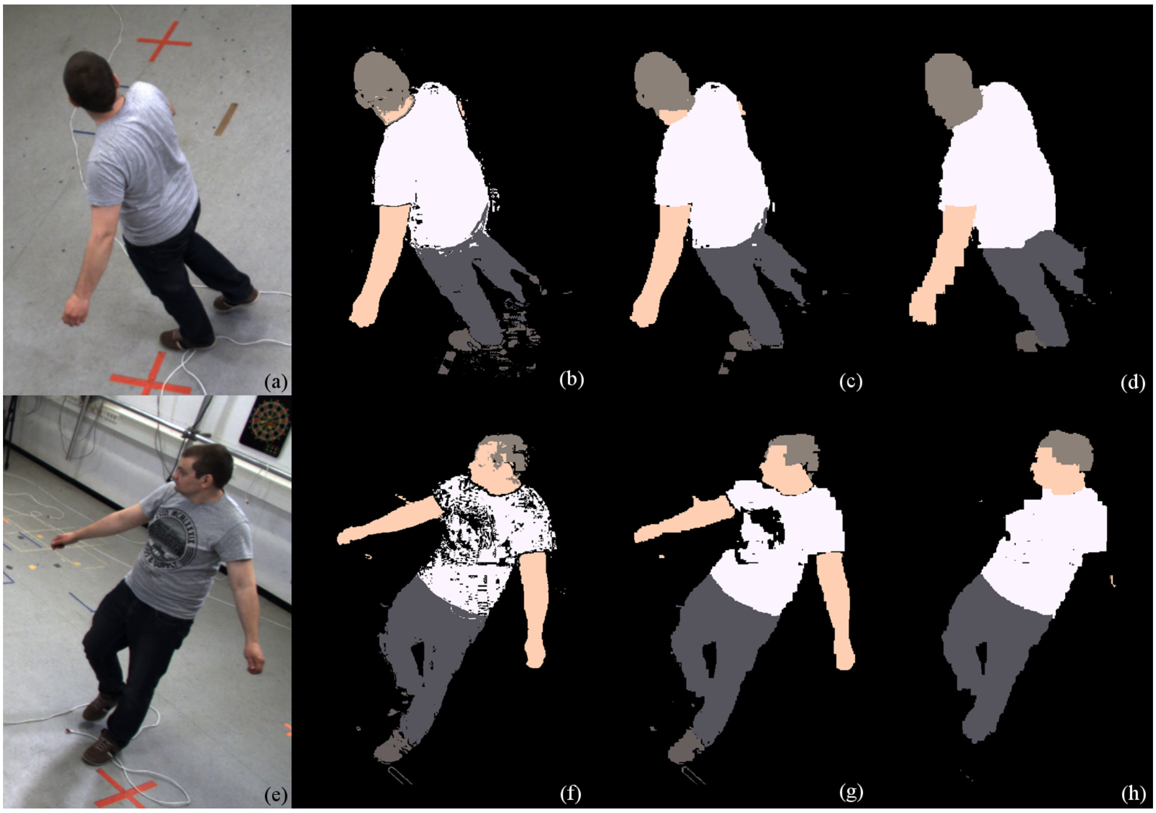


**Figure 3.9:** Segmentation results using different smoothness weights. (a,e) Input views, (b,f) no smoothness weight ($\lambda_s := 0.0$), (c,g) used smoothness weight ($\lambda_s := 0.5$), (d,h) high smoothness weight ($\lambda_s := 10$)

discontinuity-preserving interaction potential that is defined such that $s(\ell_i, \ell_j) := 1$ if $\ell_i = \ell_j$ and $s(\ell_i, \ell_j) := 0$ otherwise. The pairwise term used in (3.1) is given by the generalized Potts model [18]

$$E_{ij}(\ell_i, \ell_j) := w_{\text{smooth}} \cdot \omega_{ij} \cdot s(\ell, \ell') \quad \forall \ i \sim j\,. \tag{3.10}$$

where $w_{\text{smooth}}$ is the smoothness weight and $\omega_{ij} \geq 0$ is a weight that depends on appearance similarity among neighboring pixels $i, j$ and is defined as

$$\omega_{ij} = \exp\left(\frac{\|\Psi(x_i) - \Psi(x_j)\|_2^2}{2}\right)\,. \tag{3.11}$$

The purpose of the weights $\omega_{ij}$ is to increase the cost for assigning different labels to neighboring pixels that have similar color appearance, and to decrease the cost if the color appearance is different. Figure 3.9 shows segmentation results with varying smoothness weights $w_{\text{smooth}}$. Segmentations obtained without smoothness term are clearly noisy, while those with high smoothness weights tend to have over-smoothed material boundaries, causing thin materials, such as arms and shoes, to vanish, see Figure 3.9$(d,h)$. We used $w_{\text{smooth}} := 0.5$ for all resulting sequences, which gives a good trade-off retaining material boundaries accuracy with smoother overall look.

### 3.5.4 Minimization of the MRF Energy

In order to minimize (3.1), we use the alpha-expansion algorithm [19] that has appealing properties both from a theoretical and from a practical point-of-view. On the one hand, when minimizing

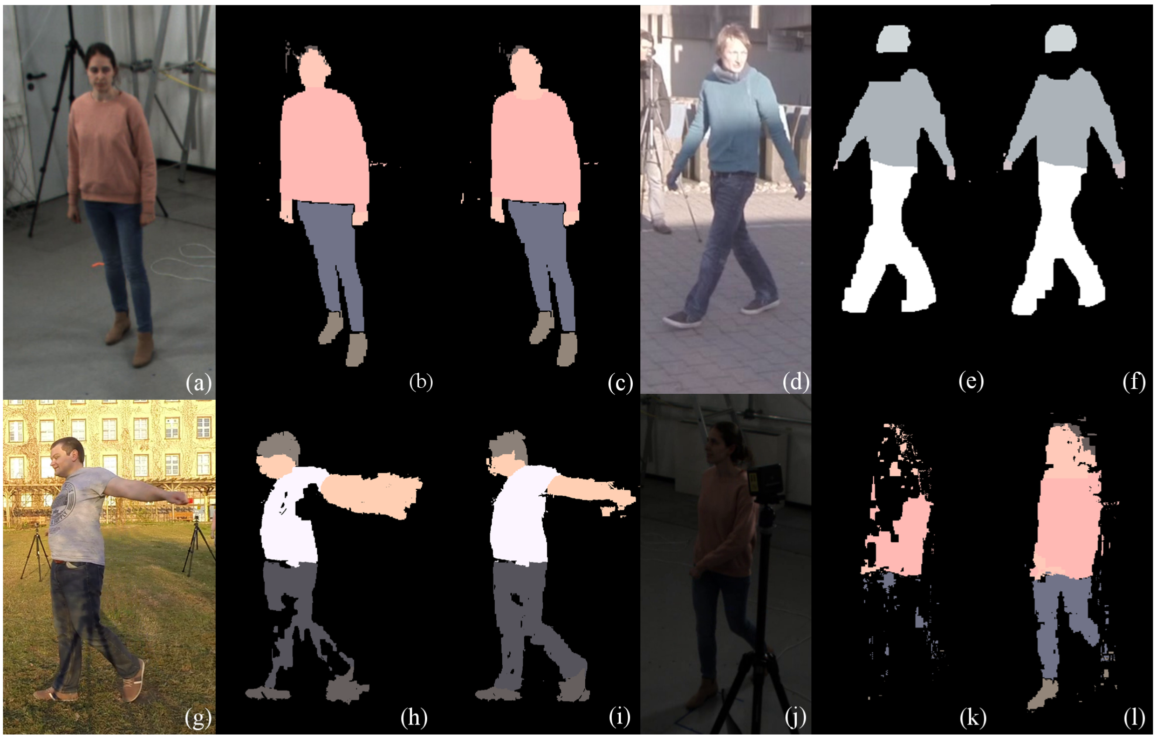


**Figure 3.10:** Examples of our segmentation for 4 sequences. (a,d,g,j) Input frames, (b,c,h,k) initial segmentation guesses and (c,f,i,l) final refined segmentations.

the energy in Equation (3.1) with a generalized Potts model as smoothness term (as in (3.10)), the alpha-expansion algorithm has the guarantee that the so-obtained local optimum lies within a factor of the global optimum [178]. Moreover, the alpha-expansion algorithm is very efficient and is known to produce good solutions in practice. In order to improve the performances, we typically reduce the search space by cropping the input image views based on the reprojected model location at the current frame. In particular, we consider an enlarged bounding box around the skeleton re-projected location, with a margin of 100 pixels.

For all frames $f > 1$ we estimate an initial segmentation guess $\widetilde{\boldsymbol{\ell}}^f$, which is obtained using the unchanged material appearance information from the previous frame $\Psi(\ell)^{f-1} := \{\mu_\ell^{f-1}, C_\ell^{f-1}\}$ and the pose prediction formula described in Section 3.5.1 for the costs. Once the actual skeleton pose is estimated using the mocap approach described in the next section, the initial segmentation guess is also refined and the new material appearance information is updated. Examples of the resulting segmentations with their corresponding initial guess are visualized in Figure 3.10. Notice how the refined estimates resolve most of the issues present in the initial guesses for all shown sequences in Figure 3.10$(c, f, i, l)$. There are cases, where the initial guess already gives high quality segmentation results, as in Figure 3.10$(b)$, since the predicted pose in that case is already very accurate. Figure 3.10$(i, l)$ show great improvements in the refined versions, where most of the illumination-changing issue and shadows have been correctly recognized and labeled. Also in the extreme harsh shadows case Figure 3.10$(d)$ there are improvements, although, as depicted, the method failed to recognize the overexposed face as skin. The quality of the segmentations highly depends on the quality of the skeleton pose, since both material appearance and pose costs are estimated based on the skeleton configuration. Overall results clearly show the robustness of our segmentation method in challenging illumination-changing scenarios.

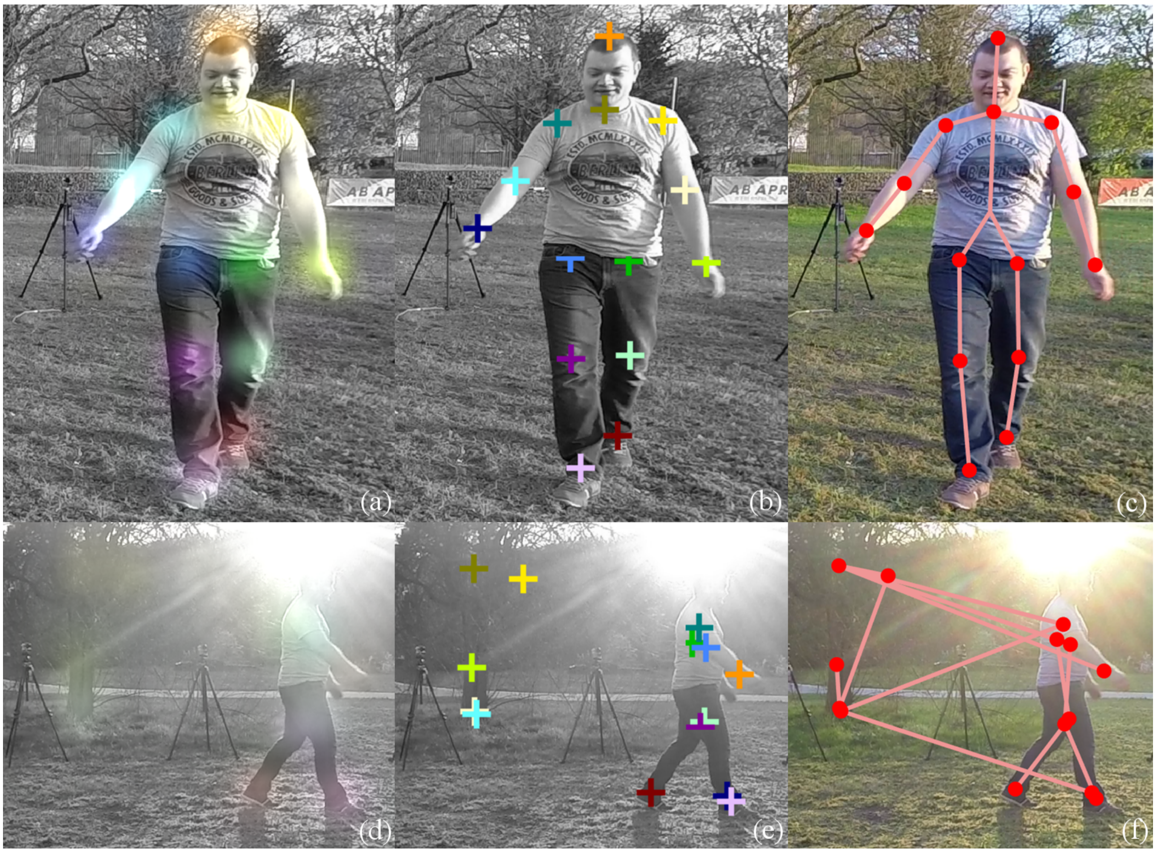


**Figure 3.11:** Visualization of the input 2D CNN-based detections. (a,d) Colored heat-maps obtained from each joint, (b,e) heat-map maxima identified by crosses, (c,f) 2D corresponding skeleton fit to the detections.

## 3.6 Pose Tracking

This section describes the pose tracking algorithm used to successfully track the human performance in presence of challenging illumination changes. The goal is to estimate accurate, temporally-smooth skeleton pose parameters $\mathcal{S}^f$ at current frame $f$. We give as input to our mocap approach both the pre-computed actor model, described in Section 3.4.1, and the input multi-view material segmentations $\boldsymbol{\ell}^f$ estimated using the method described in Section 3.5. By using the material segmentations directly instead of the raw images, we gain illumination robustness which helps guiding the skeleton tracking regardless of possible appearance and light changes.

Such multi-view illumination-invariant images are an optimal input to appearance-based tracking approaches. Aiming at efficient skeleton recovery as well as robustness to possible input segmentation outliers, e. g. imprecise silhouette borders, missing materials or other failures as described in Section 3.5, we adopt the Sums of Gaussians based skeleton tracking method described in detail in Section 2.4.1. The use of volumetric representation of the body model, through a set of Gaussians, as well as the smooth optimization used, helps in successfully recovering the skeleton pose even in presence of qualitatively poor segmentations. Typically if at least a few views have accurately labeled the corresponding materials, then accurate pose tracking is still feasible. In extreme cases, however, when, e. g. , the input skeleton pose is wrongly initialized, the tracking method is unable to recover, due to the segmentation procedure getting stuck to wrong material appearance estimate.

To avoid the aforementioned situation, we additionally provide the tracking method with CNN-based 2D detections $\delta$ obtained by the *convolutional pose machine* approach [185]. The 2D sparse detections which are available for 14 joints only, included end-effectors such as hands and feet, provide additional cues on the human pose in case of material segmentation failure. Figure 3.11$(a,d)$ show examples of input 2D detections given as distinct heatmap for each joint. CNN-based detections are typically robust to varying background and illumination-changes, however, they are temporally and spatially unstable (see a failure case in Figure 3.11$(d)$). Nevertheless a combination of both 2D detections and material segmentations turned to be an effective solution to the pose tracking problem in challenging illumination-changing scenes.

The 2D joint heatmaps are treated as images similarly to the material segmentation and given as input to the mocap approach. The additional detection Gaussians, described in Section 3.4.1, are used to match the skeleton pose with CNN-based 2D detections, in a similar way the Volumetric Gaussians are fit to the segmented images. As shown in Section 3.7, the additional use of the detections allows to recover the correct skeleton pose even when uninitialized. The next section describes in detail how to combine the benefits of the detections with the segmentations using an adaptive weighting scheme that adjusts their importance based on their reliability.

The pseudo-code of the complete proposed tracking method is outlined in Function 4. Our approach goes through all frames in the sequences, estimating the skeleton pose if $f > 1$ (lines 3-5) based on the currently available segmentations $\boldsymbol{\ell}^f$ and detections $\delta^f$. Lines 6-11 and 17-23 depict respectively the estimation the material segmentations for the current frame $f$ (refined or initial) and the next frame $f+1$ (initial guess). Finally lines 12-16 perform weights updating described in the next Section 3.6.1.

### 3.6.1 Adaptive weighting

In order to improve upon the robustness and to prevent wrong segmentations to cause error propagation, we employ an adaptive weighting strategy to set the relative importance between the material segmentations $\boldsymbol{\ell}^f$ and the CNN-based joint detections $\delta^f$ at current frame $f$. Let $w_s$ and $w_d$ be respectively the weights of the segmentation and joint detections, initially set to $w_s = 0.8$ and $w_d = 0.2$. In order to check for unreliable segmentations, we compute for each material $\ell_1, \dots \ell_L$ the $\ell_2$-norm of the distance between the $\mu_\ell^f$ at the current and $\mu_\ell^{f-1}$ at the previous frame, i. e.

$$dist(\ell) := \|\mu_\ell^f - \mu_\ell^{f-1}\|_2 \tag{3.12}$$

If any of the $dist(\ell)$ for $\ell = \ell_1, \dots \ell_L$ is larger than the threshold $\theta := 0.08$, it means that the corresponding material appearance has changed suddenly (i. e. unsmoothly), due to either abrupt illumination changes or wrong skeleton poses. In this case, we update $w_s = \frac{w_s}{2}$ to decrease the relative importance of the segmentation, otherwise we set $w_s := \min(w_s + 0.1, 0.8)$. The weight $w_d$ is obtained as $w_d = 1 - w_s$, such that when $w_s$ is decreased the tracking method relies more on the detections.

## 3.7 Experimental Results

In this section, we evaluate the proposed tracking approach by running it on several multi-view sequences with illumination changes. Among the test sequences, we specifically capture several outdoor videos with harsh and soft shadows caused by sun and tree interactions, as well as in-studio

**Algorithm 4:** Main function to estimate pose parameters $\mathcal{S}^f$ for each frame $f := 2 \ldots F$, given input skinned mesh $M$, multi-view images $I_c^f$ and detections $\delta_c^f$ and initial pose parameters $\mathcal{S}^1$.

```
function $\mathcal{S}^f := Solve(M, I_c^f, \delta_c^f, \mathcal{S}^1)$:
for $f := 1$ to $F$ do
    if $f$ != $1$ then
        $\mathcal{S}^f := mocap(M, S^f, \boldsymbol{\ell}^f, \delta^f, w_d, w_s)$;
    end
    for $c := 1$ to $n_c$ do
        $E_c^{\mathrm{p}} := PoseCosts(M, \mathcal{S}^f, c)$;
        $[E_c^{\mathrm{a}}, \mu_\ell^{f,c}, C_\ell^{f,c}] := AppearanceCosts(I_v^f, E_c^{\mathrm{p}})$;
        $E_c^f := E_c^{\mathrm{a}} \times E_c^{\mathrm{p}}$;
        $\boldsymbol{\ell}_c^f := Optimize(E_c^f)$;
    end
    if $\exists \ell : ||\mu_\ell^{f,c} - \mu_\ell^{f-1,c}|| > \theta$ then
        $w_s := \frac{w_s}{2}$, $w_d := 1 - w_s$;
    else
        $w_s := \min(w_s + 0.1, 0.8)$, $w_d := 1 - w_s$;
    end
    $\mathcal{S}^{f+1} := \mathcal{S}^f + \frac{\mathcal{S}^f - \mathcal{S}^{f-1}}{2}$;
    for $c := 1$ to $n_c$ do
        $E_c^{\mathrm{p}} := PoseCosts(M, \mathcal{S}^{f+1}, c)$;
        $E_c^{\mathrm{a}} := AppearanceCosts(I_c^{f+1}, E_c^{\mathrm{p}})$;
        $E_c^{f+1} := E_c^{\mathrm{a}} \times E_c^{\mathrm{p}}$;
        $\boldsymbol{\ell}_c^{f+1} := Optimize(E^{f+1})$;
    end
end
```

| Sequence | *boy, girl*<br>*_indoor* | | *boy, girl1, girl2*<br>*_outdoor* | | | *walk1, walk2*<br>*_outdoor* | |
|---|---|---|---|---|---|---|---|
| Published by | Us | | | | | *GVVPerfCapEva* [68] | |
| Cameras | 8 | | | | | | |
| Frames | 330 | 770 | 400 | 270 | 300 | 700 | 600 |
| Frame rate | 40 Hz | | | | | | |
| Camera type | PhaseSpace Camera [81] | | GoPro Camera [67] | | | | |
| Resolution | 1004 × 1004 | | 1920 × 1080 | | | | |

**Table 3.1:** Details for each sequence. This table summarizes the technical details of the sequences used in this chapter.

videos with simulated global light-changes. The indoor global illumination changes are simulated by randomly switching on and off a subset of the studio lights, which are placed around the captured scene. We additionally run our approach on two challenging sequences from the *GVVPerfCapEva* dataset [68]. For one of the outdoor captured sequences, we computed ground truth joint locations and performed quantitative evaluation as well as comparison to different tracking approaches. We show that our method produces reliable and accurate tracking, robust to illumination changes and capable of quickly recovering from temporal failures due to challenging motions or bad pose initialization.

### 3.7.1 Test Sequences

Due to the lack of available multi-view calibrated footage with illumination changes, for evaluating our approach we capture 5 new sequences with the wanted actor-light interactions, see Table 3.1. 3 sequences are captured outdoor at sunset on a grass field surrounded by trees. As the sun goes slowly down behind the trees, the initially soft shadows produced by the branches and leaves interacting with the sunlight cause soft shadow spots to appear on the actor's body, temporally changing his appearance. We design the captured scene such that half of it is covered by the shadows of the trees, and half is directly hit by the sun. Two differently dressed actors are recorded while moving in such complex illuminated scene, producing 3 sequences: *boy_outdoor*, *girl1_outdoor* and *girl2_outdoor*. In particular, in *boy_outdoor* the recorded actor performs complex arm crossing movements while stepping from the shadow to the illuminated area. The remaining sequences involve the same girl actor performing respectively simple walk and jumping motions. The acquisition system for this scene is composed of 8 steady synchronized and calibrated*GoPro* cameras [67] placed in a circle around.

To test our method on global illumination changes as well, we switch the capturing setup in an indoor studio with a different acquisition setup and a semi-professional studio light system, that help to simulate wider light changes, both in terms of scene coverage and intensity. The light system is composed of several white lights placed all around the capture scene, that can be individually operated by (smoothly) change their intensity from strong to off. While capturing the actors moving around, the studio lights were randomly operated to generate global illumination variations and therefore appearance variations of the actor surface. The sequence *boy_indoor* shows smooth light reduction across time, while the actor performs similar arm exercises as in *boy_outdoor*. The longer *girl_indoor* sequence involves extreme low-light conditions, where the girl's body is barely visible in the dark. Switches from bright to dark illumination happen discontinuously and faster, making this sequence one of the most challenging to work with. The acquisition system indoor is composed of 8 *PhaseSpace* cameras [81] placed around at different heights, including some of them attached to the ceiling.

We additionally make use of 2 available sequences from the *GVVPerfCapEva* dataset: $walk1\_outdoor$ and $walk2\_outdoor$. The first footage includes strong harsh shadows cutting the actor's body, while he walks in a circle. The second one, although recorded in steady weather conditions, shows cluttered dynamic background, with a lot of people walking around, causing trouble to the CNN-based 2D joint detection, as it is discussed in Section 3.7.3.

### 3.7.2 Runtime

Our current implementation takes around half a minute per multi-view frame when using 8 views. The performance time decreases linearly when decreasing the number of camera views considered. The biggest overhead is the estimation of color and pose costs, which has to be computed for each view and pixel. These subroutines are called in lines 7, 8, 19 and 20 in Function 4. We believe the computation time can be drastically decreased using parallel computing techniques. The appearance costs can be computed in parallel for all the image pixels as they solely depend on the single pixel appearance. The pose costs estimation could be further simplified to still obtain reliable pose costs at much higher frame-rate. The CPU implementation of the tracker, in Function 4, line 4, has proven to have real-time performances, as described in Section 2.4.1.3. Overall, we believe a systematic GPU implementation of the entire tracking approach, together with the online estimation of 2D joint heatmaps, can lead to much improved performance.

### 3.7.3 Qualitative results

In Figure 3.12 and Figure 3.1, we provide some examples of the pose estimation results obtained with our method. Input to our tracking approach, apart from the multi-view sequence and the calibration, are the actor models with manually estimated skeletal pose at frame 1. We can see from these results that our method accurately tracks the skeletal motion of the actors in both outdoor and indoor scenarios with different kind of illumination changes.

In the *girl_indoor* sequence, shown in Figure 3.12, row 1, our approach produces smooth reliably mocap results, even in presence of extremely dark illumination conditions, where the actor is hardly visible for a human observer. In dark frames, the segmented materials are accurately identified and provide the tracker valuable information regarding the skeleton location. CNN-based 2D joint locations instead are unreliable in those frames due to the extreme light conditions. This confirms the effectiveness of the material segmentation in such challenging scenarios.

Figure 3.12, row 2, shows some resulting frames of the *boy_indoor* sequences. While the darkest illumination conditions reached during this sequence remain brighter than the sequence in row 1, here the tracking has to deal with more challenging arm crossing motions, that typically cause (temporal) failures for most of the approaches. Our method tracks the whole motion reliably regardless of the complex movements.

In the outdoor sequences of Figure 3.12, rows 3-5, namely $girl1\_outdoor$, $girl2\_outdoor$ and $boy\_outdoor$, although the appearance of the actors changes significantly as the actors walk into or out of the shadow, our method is still able to track the skeletal motion stably. The strong appearance changes of the upper body surface area, e. g. shirt and sweater, from dark to piece-wise over-exposed and vice-versa, typically causes tracking failure in common color-based approaches. The use of

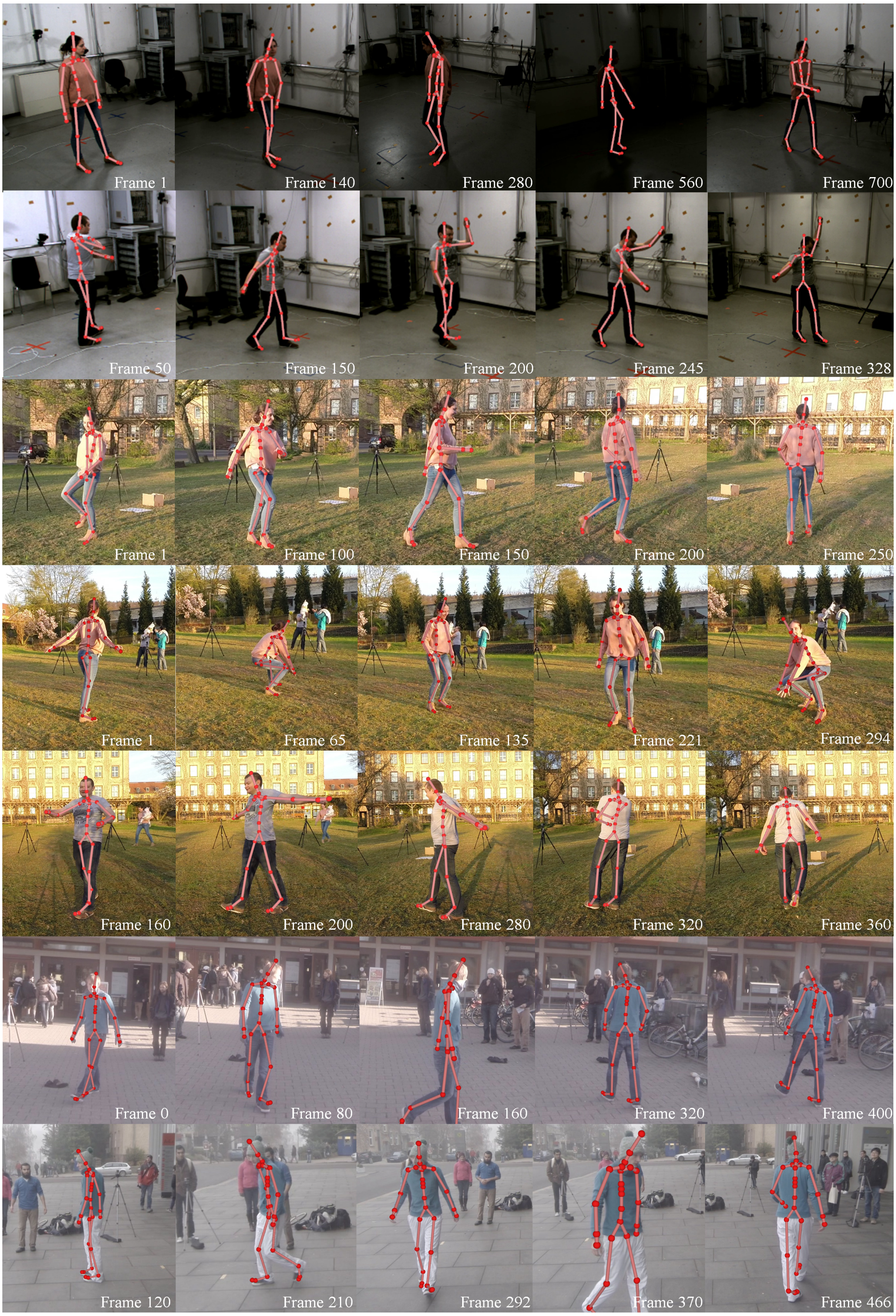


**Figure 3.12:** Qualitative results. The figure shows tracking results obtained with our approach on 2 indoor and 5 outdoor sequences (columns) and different frames of the same view (rows). From the top to the bottom: *girl_indoor*, *boy_indoor*, *girl1_outdoor*, *girl2_outdoor*, *boy_outdoor*, *walk1_outdoor*, *walk2_outdoor*.

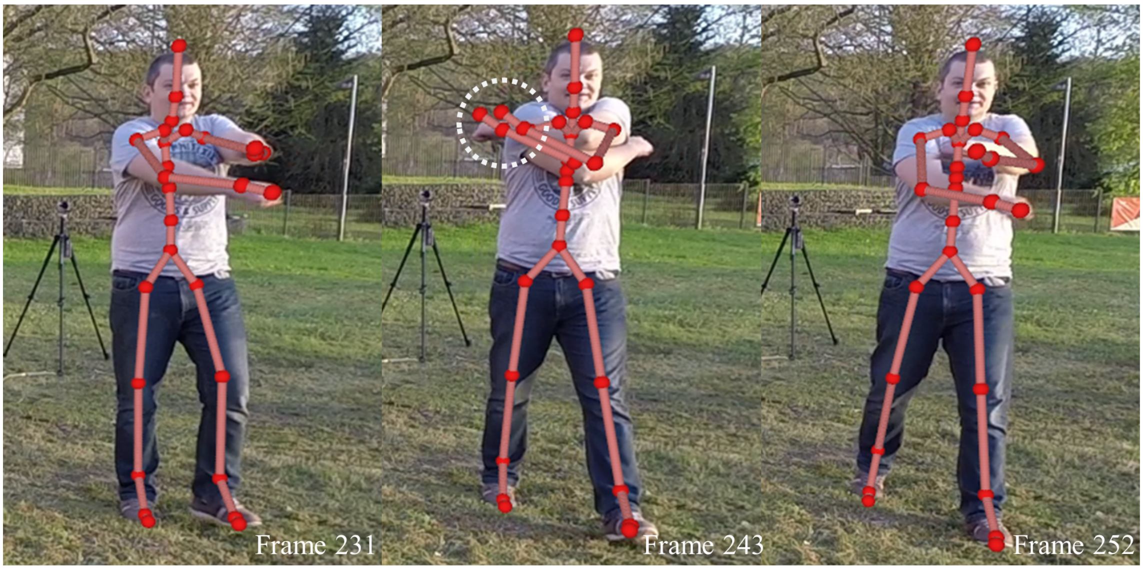


**Figure 3.13:** The image shows a tracking failure during a challenging motion in the *boy_outdoor* sequence. Our method recovers in the next frames. The white circles point to tracking failures.

material segmentations combined with the CNN-based 2D joint detections provide sufficiently reliable data to overcome challenging appearance shifts and complex motions, like arm crossing and jumps.

In the *boy_outdoor* sequence the actor crosses his arm while walking from the shadow to the illuminated area. The combination of challenging motion and illumination at the same time, causes temporal tracking failure during those few frames, see Figure 3.13. Despite the failure, our tracking approach quickly recovers after few frames, thanks to the adaptive weighting of the material segmentations against the 2D joint detections. Loosing tracking results in abrupt changes in the estimated material appearance, which in turn results in reduced weighting (confidence) for the segmentations and increased weighting for the detections. 2D joint detections can recover the tracking back to stability thanks to the fact they are not strictly relying to temporal information, instead the joint location probability is estimated individually for each frame in 2D. In the next section, we show that relying exclusively on the detections results in temporally inaccurate tracking. The temporal stability of our motion results is visible in the resulting video sequence.

In the sequence of Figure 3.12, row 6, *walk*1*_outdoor*, our method yields successful tracking results, even in presence of harsh shadow. Segmentations for this sequence are successfully estimated, independently to the appearance changes of the blue sweater. The last resulting *walk*2*_outdoor* sequence, shown in Figure 3.12, row 7, provides a good example where 2D joint detections alone become unreliable due to the presence in the background of several moving people that tend to disguise the detector at times. Our tracking approach for this sequence relied almost exclusively on the material segmentations, resulting in temporally reliable and accurate skeleton tracking.

As additional test, we run our method on the *boy_outdoor* sequence without providing the initial skeleton configuration, see Figure 3.14. Testing revealed the importance of the CNN-based joint detections during this experiment as well as the effectiveness of the adaptive weighting strategy. The material segmentations alone get stuck to wrong material appearances, due to the body model being re-projected to highly inaccurate location. Because of the typical high variability of the material appearance across frames, due to the overlap of the badly re-posed model with different materials

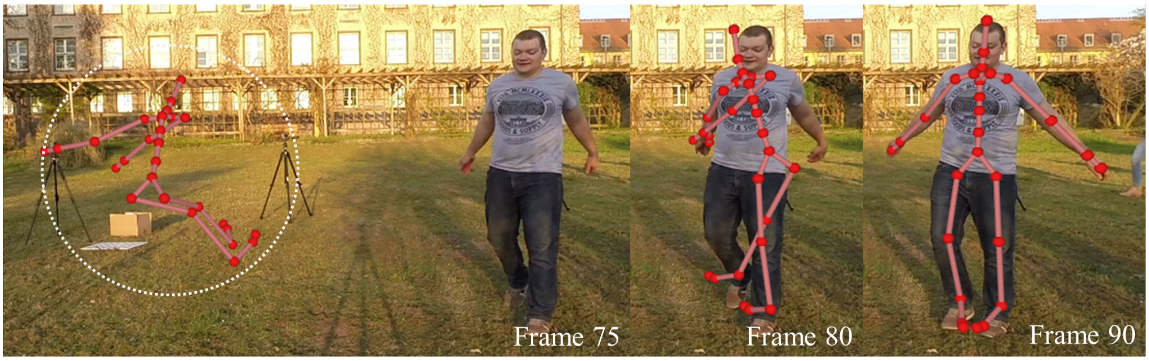


**Figure 3.14:** Bad pose initialization example. The image shows skeleton tracking resulting from bad pose initialization in the *boy_outdoor* sequence. Even in presence of strong misalignments, our method is able to successfully recover the correct pose after few frames.

(e. g. building, grass and so on) as well as the pose being discontinuously re-estimated, the 2D detections weight increases greatly, suggesting the tracker to rely more on the detections and less on the biased segmentations. The process of adaptive weighting eventually allows the tracking approach to converge to the correct body pose after few frames, see Figure 3.14.

### 3.7.4 Comparison

We compare our tracking approach with 3 different tracking methods, specifically chosen to prove the effectiveness of the algorithmic choices for our mocap algorithm. In particular, we pick the original Gaussian based tracker, described in Section 2.4.1, denoted as *Images-only*, which relies exclusively on the input images, without considering any CNN-based detections or material segmentations. We additionally run the aforementioned mocap method on the 2D detections only (*Detections-only*). Finally, we chose the tracking method proposed by Rhodin et al. [142], which is a representative approach for tracking relying both on input unsegmented images and 2D joint detections estimated by CNN (*Image+Detections*). We run all the chosen tracking approaches to the *boy_outdoor* sequence for which we manually estimated the ground truth joint locations. Given the ground truth, we qualitatively as well as quantitatively evaluated the proposed approach against the related work.

Figure 3.15 qualitatively compares the estimated poses resulting from the aforementioned tracking approaches against our results. The *Images-only* method delivers the least accurate results due to the challenging appearance changes and complex body motions, i. e. arm crossing. Apart from temporally loosing tracking of the limbs during the crossing motion, the overall body location and orientation remains stable until the color appearance of the various materials changes due to the body being hit by direct sunlight, after it leaves the shadow. Examples of resulting skeleton configuration are shown in Figure 3.15$(d,h)$.

The *Detections-only* method is more robust to complex motions and illumination, however its temporal instability leads to frequent sudden tracking failures. The jittering is due to the high variability of the joint heatmaps, which are computed independently for each frame and camera. 2D joint detections often contain outliers due to the presence of other people in the scene or person-like objects. As soon as for more than half of the views the estimated heatmaps can be used to identify correct joint locations, the correct pose is typically also well identified, although jittery. Another limitation of the *Detections-only* method is the missing tracking information for some joints, like feet and hands, which therefore cannot be recovered. These are also visible in Figure 3.15$(b,f)$.

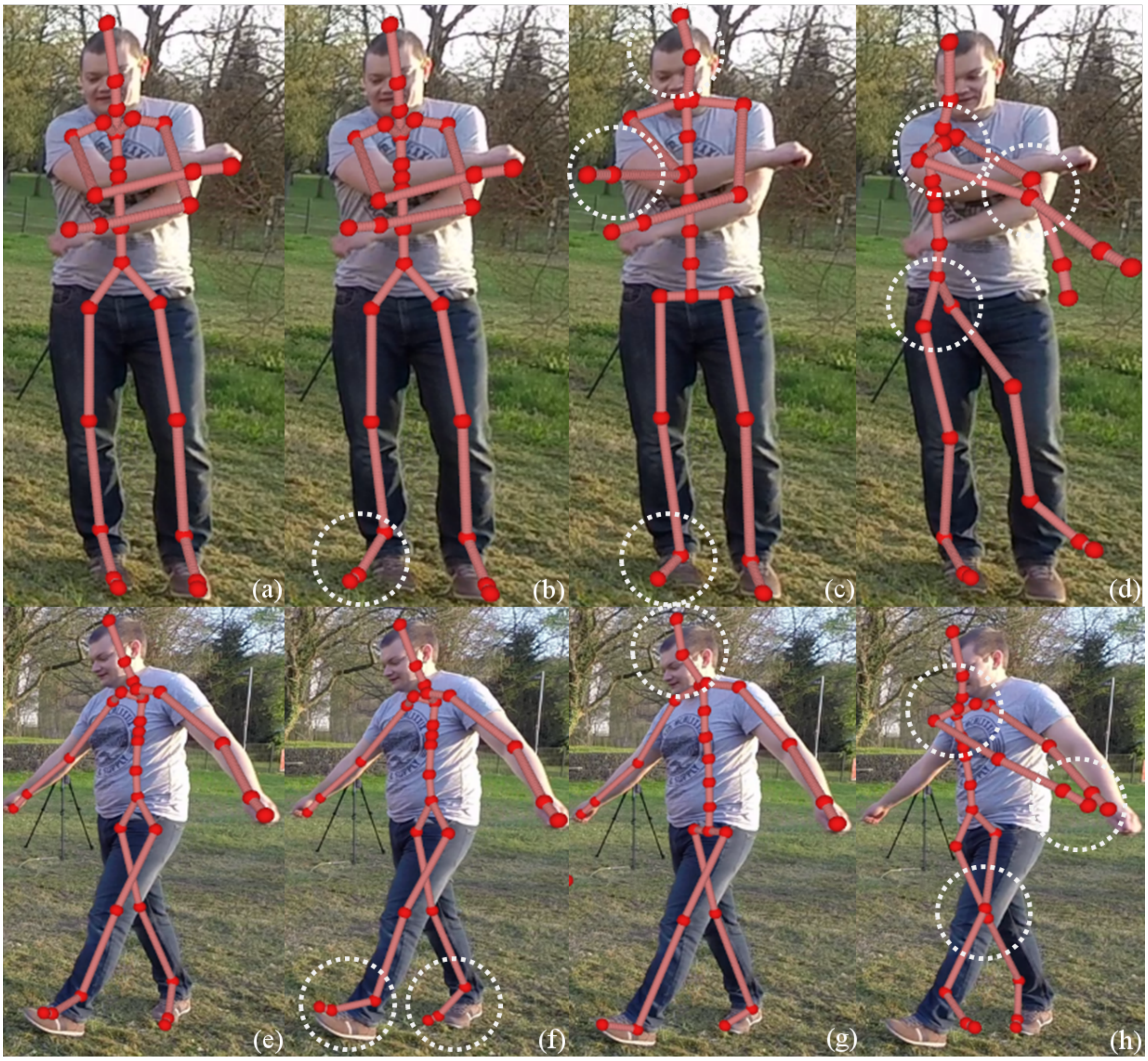


**Figure 3.15:** Comparison of estimated poses in 2 frames of the *boy_outdoor* sequence for different methods. (a,e) Our reconstruction, (b,f) *Detections-only*, (c,g) *Image+Detections* and (d,h) *Images-only*. The white circles point to tracking failures.

The *Image+Detections* method highly relies on the (jittery) CNN-based detections, however the resulting sequence is smoother than *Detections-only* due to the additional raw images input for guiding the tracking. The jittering is attenuated by combining the detections with the denser information given by the raw images. However, temporal pose estimation failures are still present due to the unreliable illumination changing images. Figure 3.15$(c,g)$ shows an example frame where the estimated pose is incorrectly estimated, due to high input color variability caused by the shadows. End-effector joints like the top of the head and the feet remain sometimes unexplained because of the lack of reliable information at those locations.

Our tracking approach successfully estimates the entire body motion outperforming all the related approaches, both in terms of motion estimation accuracy and smoothness across time. The resulting skeleton estimations compared to the other discussed methods are visible in Figure 3.15$(a,e)$.

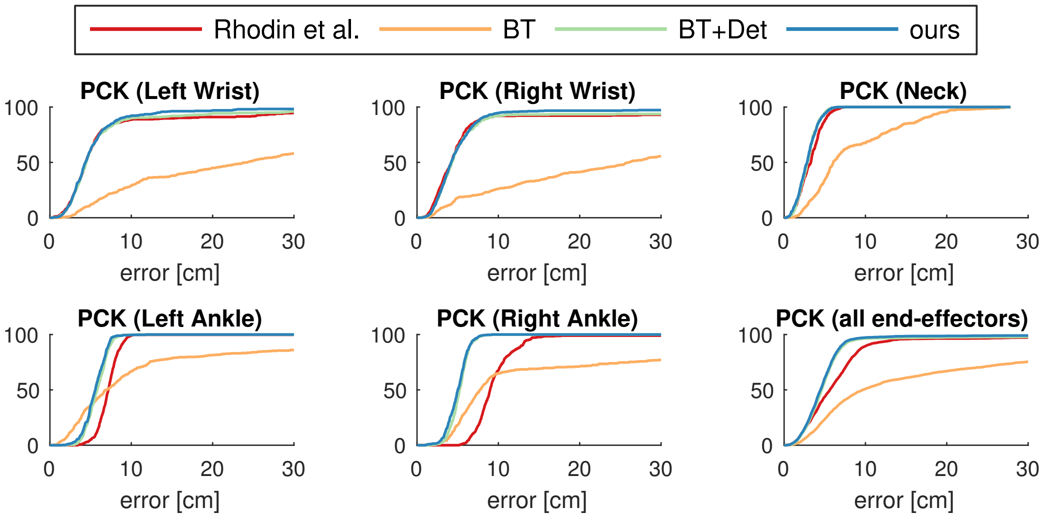


**Figure 3.16:** 3DPCK values of end-effector joint positions for *boy_outdoor* sequence when using 8 cameras. The value on the vertical axis shows the percentage of frames where the error is smaller than or equal to the value on the horizontal axis. The corresponding AUC values are shown in Table 3.2.

**Table 3.2:** Area under curve (AUC) values of 3DPCK curves in Figure 3.16, as well as the corresponding numerical summary of average ground truth errors (in cm) and standard deviation for five different joints.

| | *Images+Detections* | | *Images-only* | | *Detections-only* | | Ours | |
|---|---|---|---|---|---|---|---|---|
| Joint | AUC | Error | AUC | Error | AUC | Error | AUC | Error |
| Left Wrist | 0.9249 | 7.35±9.68 | 0.6858 | 30.92±26.18 | 0.9295 | 6.89±8.77 | **0.9428** | **5.59**±4.91 |
| Right Wrist | 0.9298 | 7.20±11.39 | 0.6023 | 41.02±35.09 | 0.9326 | 6.91±9.73 | **0.9451** | **5.61**±6.32 |
| Left Ankle | 0.8839 | 7.28±3.05 | 0.7979 | 12.69±14.35 | 0.9075 | 5.79±1.28 | **0.9114** | **5.55**±1.33 |
| Right Ankle | 0.8279 | 9.60±3.34 | 0.7003 | 16.73±16.82 | 0.9061 | 5.22±1.14 | **0.9105** | **4.98**±1.25 |
| Neck | 0.8836 | 3.23±1.45 | 0.7001 | 8.34±5.71 | 0.8951 | 2.91±1.18 | **0.8970** | **2.86**±1.26 |

### 3.7.5 Quantitative evaluation

Quantitative results evaluating those methods on the *boy_outdoor* sequence are shown in Figure 3.16, as well as in Table 3.2. In Figure 3.16, we show the percentage of correct keypoints (3DPCK [116]) for four end-effector joints across the entire sequence comprising 400 frames from 8 different views. The end-effector joints, e. g. hands and feet, provide a good measure for the overall skeleton pose accuracy, since the corresponding joints typically move faster than the remaining ones and thus are most prone to tracking failures and jittering. Among the chosen joints for the evaluation, we selected the wrists, the ankles and the neck. Table 3.2 summarizes the corresponding area under the curve (AUC) as well as the 3D ground truth error (in *cm*) for the chosen methods and joints. Overall, our method and the *Detections-only* method significantly outperform the other two approaches. As also discussed in the qualitative evaluation of the methods in Section 3.7.4, our method yields more accurate motion estimation results. This assessment is also quantitatively confirmed in Table 3.2 with a reduced AUC as well as reduced overall error.

We also evaluate the performance of our method compared to *Detections-only* depending on the number of views, see Figure 3.17. For the evaluation we compute the end-effector joints errors in *cm* as before. As we decrease the number of views used, we keep the most challenging camera views,

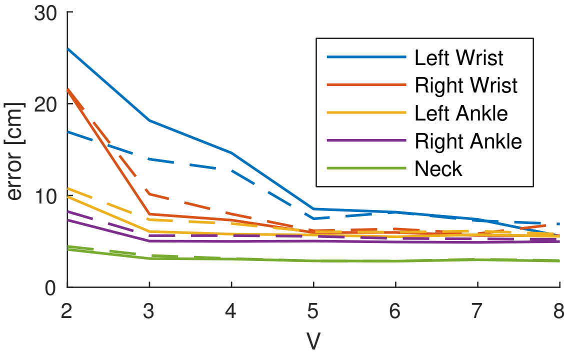


**Figure 3.17:** Ground truth error (vertical axis) of our method (solid lines) and *Detections-only* (dashed lines) depending on the used number of views (horizontal axis).

where the appearance changes are best visible. We also maintain a good distribution of the chosen cameras, to favor a good coverage of the scene. As expected, skeleton pose estimation in acquisition setups with fewer cameras are less accurate. Our method outperforms *Detections-only* for all joints, except the left wrist joint alone, that is shown to be more accurately estimated by *Detections-only* when fewer than 6 cameras are used. Apart from being invisible from the selected camera views for most of the time, the left arm happen to casually overlap to the same background material, causing segmentation trouble to distinguish it from the skin through unchanged adaptive weighting, thus resulting in long lasting tracking failure of the corresponding joints.

## 3.8 Discussion and Limitations

The proposed tracking approach is robust to illumination changes thanks to the combination of material segmentations and CNN-based joint detections. Results on various testing sequences with different illumination conditions show the effectiveness of the proposed method to accurately track the human motion in outdoor challenging scenarios with a possible dynamic background, occlusions and appearance changes. Our approach outperforms existing related approaches as demonstrated both qualitatively and quantitatively in Section 3.7.

Illumination-invariant material segmentation is robust thanks to the proposed dynamic appearance term. Our method is, in general, reliable in cases where the foreground and the background appearance coincide, thanks to the additional pose prediction term accounting for plausible motions. Results demonstrated the effectiveness of the tracking approach even in presence of imperfect material segmentation boundaries. Since the used mocap approach is based on smooth Gaussian based scene representation, it does not strictly consider sharp object boundaries or edges, making incomplete or inexact segmentations suffice for good tracking results. Moreover, the missed information about a certain material location can still be inferred by the location of the other surrounding materials. Typically, in a multi-view setting, the background varies a lot and the combination of the segmentations of all the views suffices to converge to the right pose.

The additional use of the 2D joint detections and the adaptive weighting scheme further increase the robustness of our tracking approach. Detections have shown their effectiveness in presence of inaccurate segmentations and strong initial skeleton pose misalignment. Thanks to the proposed scheme, our method can quickly recover from segmentation errors.

The initial model acquisition and semi-automatic material segmentation have a direct impact on the quality of the results. Our approach does not require an exact shape match to the images, however, accurate input shapes lead to higher-quality segmentation and tracking. To automatically identify the actor materials, a simple color clustering might suffice. Automatic identification of the materials could, however, produce poor quality segmentations, e. g. in presence of highly textured apparel.

Our method cannot directly handle non-homogeneous as well as highly specular foreground materials, due to their high variability within the same material. A multi-modal color term, e. g. Gaussian mixture models, could help in improving segmentation of such materials. Alternatively, it might be necessary to define multiple sub-materials to better handle highly variable material appearances.

## 3.9 Conclusion

In this chapter, we have presented a novel approach for illumination-invariant human mocap in a multi-view setup. The goal is to capture the continuous temporally-consistent skeleton pose in a multi-view sequence with illumination changes, such as global light changes, or appearance changes, due to e. g. harsh shadows or view-inconsistent reflectance of e. g. pieces of apparel. We employ an intermediate image representation that factors out variations in lighting across the sequence in time, or variations in appearance across the views. In order to obtain this invariant representation, for each frame and each view we solve a segmentation problem that uses previous tracking results in order to infer cues about the individual materials' appearances in the current frame. By fusing this approach with CNN-based joint detectors as well as with the model-based tracker described in Section 2.4.1, we demonstrate superior performance compared to other methods, even under difficult conditions.

The proposed mocap method is focused on recovering the coarse skeleton based deformation of the actor, ignoring personalized non-rigid detailed deformation, which are crucial in performance capture applications. In the next chapter, we fill this lack by discussing a novel approach for dense surface tracking. The described surface capture method densely refines input coarse (skeleton based) actor deformation sequences by adding visible non-rigid deformation detail from multi-view images.

# Chapter 4

# Surface Capture

## 4.1 Introduction

In the previous chapter, we focused on skeletal motion capture in challenging illumination-varying scenarios, ignoring the tracking of fine-scale surface details. This chapter fills this gap for performance capture applications by focusing on a novel surface capture method that densely refines input coarse animations. Over the last decade, performance capture techniques have enabled the 3D reconstruction of the motion and the appearance of real-world scenes, such as human motion or facial expressions, from multiple video recordings [40, 64, 20, 179]. Captured sequences are reconstructed using a space-time coherent 3D mesh, also known as 4D model, that reproduces the original scene or motion. Such temporal reconstructions are essential for creating high-quality animations in entertainment and real-time applications, e. g. virtual mirrors. 4D models can be obtained by deforming a mesh or a rigged template such that it aligns with the images [40, 179], or by a per-frame independent reconstruction of the scene [162, 35] followed by a surface alignment step to convert the temporally inconsistent mesh geometries to a coherent model [25]. However, many reconstructing methods only produce coarse-to-medium scale 4D models that do not reproduce the high-frequency geometry detail present in the original scene. A second refinement step serves as to recover the finer scale shape detail on top of the coarse geometry.

For fine surface detail reconstruction, some methods have used photo-consistency constraints via stereo-based refinement [40, 162]. However, such methods often turn to discrete sampling of local displacements, since formulating dense stereo based refinement as a continuous optimization problem has been more challenging [100]. Other methods for mesh refinement align the surface to a combination of silhouette constraints and sparse image features [64]. But such approaches merely recover medium scale detail and may suffer from erroneous feature correspondences between images and shape. Recently, shading-based techniques such as shape-from-shading or photometric stereo [192, 190, 179] have been also proposed to capture small-scale displacements. However, these methods are either limited to be used in controlled and calibrated lighting setups, or they require a complex inverse estimation of lighting and appearance when they are applied in uncontrolled recording conditions.

This chapter describes an effective solution to the dynamic shape refinement step using multi-view photo-consistency constraints. As input, our method expects synchronized and calibrated multiple video of a scene and a reconstructed coarse mesh animation, as it can be obtained with previous

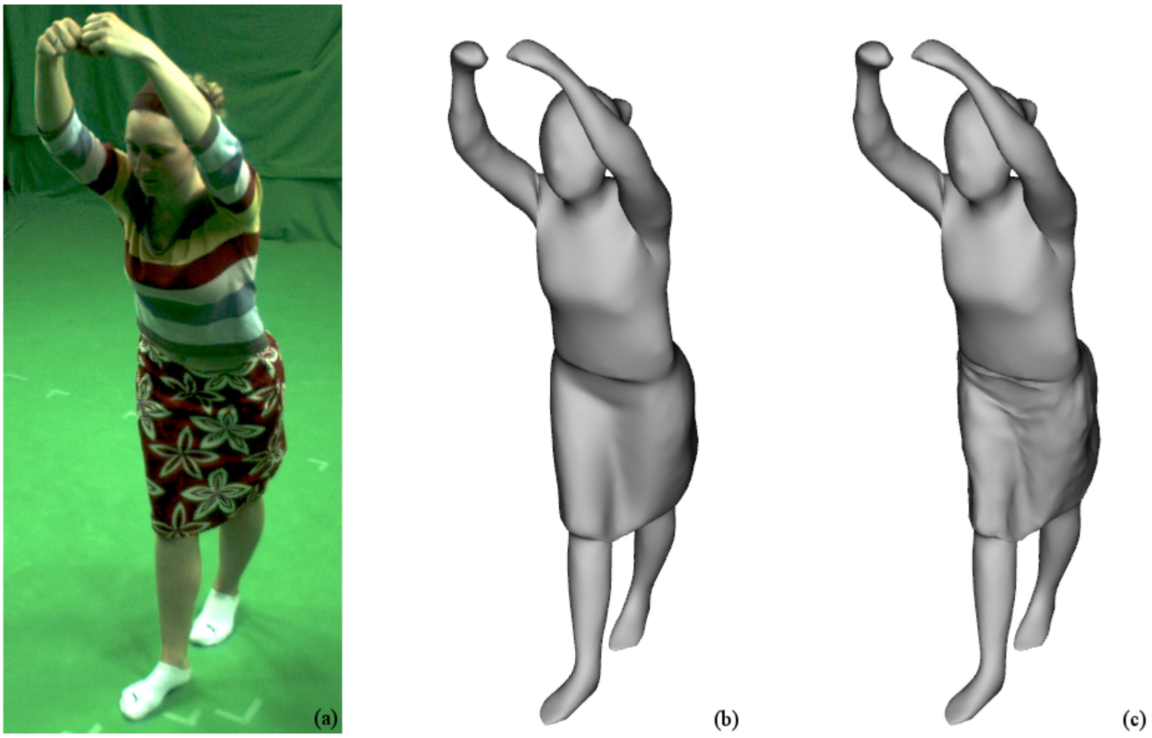


Figure 4.1: From a multi-view video sequence (a) and a coarse mesh animation (b), our method efficiently reconstructs fine scale surface details (c) on the skirt, e. g. wrinkles and folds.

methods from the literature. Background subtraction or image silhouettes are not required for refinement.

Our main technical contributions are:

1. a new implicit surface representation that models the mesh surface with a dense collection of 3D Gaussian functions centered at each vertex,
2. the formulation for dense photo-consistency based surface refinement, which we formulate as a global optimization problem in the position of each vertex on the surface,
3. and the integration of the proposed refinement method into a model-based performance capture pipeline.

The model-to-image photo-consistency formulation can be effectively minimized in terms of dense local surface displacements with standard gradient-based solvers. In contrast to previous approaches, this formulation enables a correspondence-free continuous optimization of the dense surface deformation.

We test our approach on performing actors wearing loose clothing, e. g. skirt or large t-shirt. The input coarse geometry of the scenes is obtained using a template-based method [64, 40], or a template-free method [162] followed by a surface alignment step [25]. Qualitative and quantitative results demonstrate that our approach captures more fine-scale detail, e. g. , the wrinkles in the skirt in Figure 4.1, than both of the baseline methods.

## 4.2 Related Work

Existing methods for surface capture and refinement can be split into two categories. On one hand, *model-based* methods [31, 40, 64, 179, 107, 191, 193, 9] deform a static template of an actor or a person's apparel [20], usually obtained by a laser scan or image-based reconstruction, to best fit it into synchronized multi-camera input. On the other hand, *model-free* methods [162, 180, 60, 35] remove the need of an initial template by reconstructing a per-frame independent geometry using variants of shape-from-silhouette or active or passive stereo [60, 206, 115, 162, 184, 175]. In contrast to model-based methods, the output geometry is typically temporally incoherent, e. g. has a different number of vertices and edges per frame. A subsequent step, known as *surface alignment*, is used to convert the temporally incoherent geometry proxy to a coherent geometry that deforms over time to fit the reconstructed shapes. Cagniart et al. [27] solve such free-form alignment sequentially by iterative close point matching of overlapping rigid-patches. Similarly, Budd et al. [25] use a combination of geometric and photometric features in a non-sequential alignment framework. Volumetric constraints have also been used to formulate the surface tracking problem [4, 41]. Nevertheless, regardless the accuracy of the method used for surface tracking, most of them result in a coarse-to-medium scale 4D model in which most of the high-frequency details are lost.

Most of the methods mentioned so far suffer from two main limitations: on one hand, the template-based methods that use an initial highly detailed scan model usually do not deform the fine-detail of the template according to the acquired per-frame imagery. This leads to incorrect surface detail on the final reconstructed meshes that does not match the captured dynamics [40, 179]. On the other hand, methods that require the deformation of the per-frame reconstructed meshes to achieve temporal consistency [162, 25, 27] tend to output a coarse reconstruction of the sequence. In both cases, true fine detail needs to be recovered in a subsequent step.

Medium scale non-rigid 4D surface detail can be estimated by using a combination of silhouette constraints and sparse feature correspondences [64]. Other approaches use stereo-based photo-consistency constraints in addition to silhouettes to achieve denser estimates of finer scale deformations [162, 40]. It is an involved problem to phrase dense stereo-based surface refinement as a continuous optimization problem, as it is done in variational approaches [100]. Thus, stereo-based refinement in performance capture often resorts to discrete surface displacement sampling which are less efficient, and with which globally smooth and coherent solutions are harder to achieve. Alternatively, fine-detail surface deformation caused by garment wrinkling has been investigated using cloth-specific approaches. Popa et al. [138] enhance reconstructed surfaces using a wrinkle generation method based on the recorded shadows.

Fine-scale detail can be also recovered using methods acquiring active lighting, e. g. shape-from-shading or photometric stereo [190]. Many of these approaches require controlled and calibrated lighting [77, 180, 3], which reduces their applicability in real-world scenarios. For example, Hernandez et al. [77] use red, green and blue lights from different directions to estimate surface normals with a photometric stereo approach. This enables detailed reconstruction of cloth, even untextured. Ahmed et al. [3] estimate surface reflectance and time-varying normal fields on a coarse template mesh to incorporate garment details such as wrinkles and folds. Vlasic et al. [180] reproduce high-resolution geometric detail by capturing multi-view normal maps in a large lighting dome that provides a series of novel spherical lighting configurations. More recently, shading-based refinement of dynamic scenes captured under more general lighting was shown [192, 191], but these approaches are computationally challenging as they require to solve an inverse rendering problem to obtain estimates of illumination, appearance and shape at the same time.

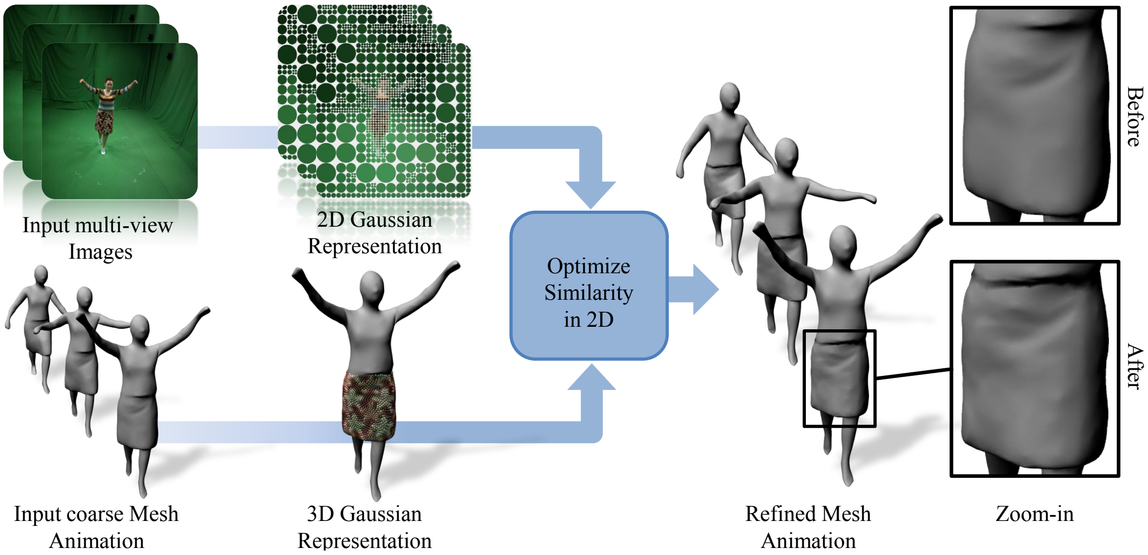

Input multi-view Images
2D Gaussian Representation
Optimize Similarity in 2D
Before
After
Input coarse Mesh Animation
3D Gaussian Representation
Refined Mesh Animation
Zoom-in

**Figure 4.2:** Our approach takes as input a coarse topologically consistent mesh animation and a set of input images from a multi-view calibrated and synchronized camera setup. Both the input mesh and image sequences are converted to an implicit representation resembling respectively a collection of 3D Surface Gaussians and 2D Image Gaussians. Then, the color consistency between the two representations is optimized in 2D for each camera view, resulting in a refined mesh animation with additional captured surface details.

Our technique has some similarity to the work of Sand et al. [151] who capture skin deformation as a displacement field on a template mesh. However, they require marker-based skeleton capture, and only fit the surface to match the silhouettes in multi-view video. In this chapter, we describe a surface refinement approach that does not require input silhouettes and has therefore the capabilities of being applied on unconstrained outdoor scenarios.

## 4.3 Overview

In this section, we describe our model-based surface capture approach, also depicted in Figure 4.2. Given a sequence of multi-view calibrated and synchronized images capturing the action of a single actor, as well as a deformation sequence, composed of a set of spatiotemporally coherent meshes, i. e. with the same topology over time, coarsely reproducing the visible surface deformation, our goal is to refine the initial set of coarse geometries by incorporating missing visible fine-scale dynamic surface details to the meshes. The proposed refinement method is applicable to any deformation sequence. For consistency with the scope of this thesis, we focused on refinement of performing humans, for which an initial deformation sequence is available.

The proposed approach extends the mocap method described in Section 2.4.1 to surface capture using a similar Gaussian based representation both for the actor model and the input multi-view images. To enable capture of finer-scale deformation detail, we employ a much denser collection of 3D colored Gaussian functions, referred to as *Surface Gaussians*, to approximate the surface. Each Surface Gaussian implicitly describes the location and color of each vertex in the input mesh. Similarly to Section 2.4.1, we maximize the color consistency between the collection of projected Surface Gaussians and the images, properly converted to a set of Image Gaussians, obtained using the approach described in Section 2.4.1.1, for all views. Because of the additional detail required for refinement, we employ a much denser set of Image Gaussians w. r. t. to those used in Section 2.4.1.

The optimization displaces the Surface Gaussians along the associated vertex normal of the coarse mesh, yielding the necessary vertex displacement. On top of an extended similarity energy functional, initially introduced in Section 2.4.1.2, we employ a new regularization term, to account for temporally and spatially smooth surface deformation tracking. Thanks to the advantageous mathematical formulation, our optimization method inherits all advantages of the initial mocap method: our smooth energy function can be expressed in closed form, and it allows to analytically computing derivatives, enabling the possibility of using efficient gradient-based solvers.

## 4.4 Scene Model

In order to enable effective model-to-image consistency optimization, we convert the actor model and the images to an implicit representation, similarly to Section 2.4.1.1. Both the actor and the input multi-view images are converted to Sums-of-Gaussian functions with additional color attributes.

### 4.4.1 Actor

Our refinement approach requires an input deformation sequence that coarsely represents the recorded action, visible in the multi-view video. The input deformation sequence, i. e. the sequence of topologically consistent meshes, is typically the result of vertex displacements applied on a given actor geometry, i. e. having fixed topology over time. Alternatively, in presence of complete three-layered actor models, as described in Section 2.2, with given skeletal joint configurations per frame, obtained, e. g. , from the mocap approach described in Chapter 3, the deformation sequence can be obtained applying skinning frame-by-frame.

Our implicit model for the input actor geometry is obtained by placing a Surface Gaussian at each mesh vertex $v_s$, $\forall s \in \{1 \dots n_s\}$, $n_s$ being the number of vertices. An isotropic Gaussian function on the surface is defined with a mean $\hat{\mu}_s$, that coincides with the vertex location, and a standard deviation $\hat{\sigma}_s$ as follows:

$$\hat{G}_s(\hat{x}) = \frac{1}{\sqrt{\hat{\sigma}_s\sqrt{\pi}}} exp\left(-\frac{||\hat{x}-\hat{\mu}_s||^2}{2\hat{\sigma}_s^2}\right), \tag{4.1}$$

with $\hat{x} \in \mathbb{R}^3$. The standard deviation $\hat{\sigma}_s$ is chosen to guarantee surface coverage while minimizing overlap. We typically regularize the input mesh triangulation beforehand, such that each vertex is equidistant and then choose the same $\hat{\sigma}_s, \forall s$. Figure 4.3 shows an example of implicit model representation for the skirt region. Note that although assigned Surface Gaussians $\hat{G}_s(\hat{x})$ have infinite support, for visualization purposes we visualize them as a sphere centered at $\hat{\mu}_s$ with $\hat{\sigma}_s$ *mm* radius. In contrast to the unnormalized representation used for the Volumetric Gaussians, shown in Section 2.2.3.2, the Surface Gaussians are normalized using the normalization factor of $\frac{1}{\sqrt{\hat{\sigma}_s\sqrt{\pi}}}$. In Section 4.5, we mathematically validate this choice and show quantitative improvements on the overall energy formulation.

We further assign an HSV color value $\eta_s$ to each Surface Gaussian. In case the geometry model of the actor is equipped with colors, it is possible to directly assign the corresponding vertex colors to the Surface Gaussians. When vertex colors are unavailable, for each projected Surface Gaussian, we compute the underlying pixel color mean, in a similar way as it is done for Volumetric Gaussians in Section 2.4.1.4. However, when computing the color mean, due to possible occlusions and other

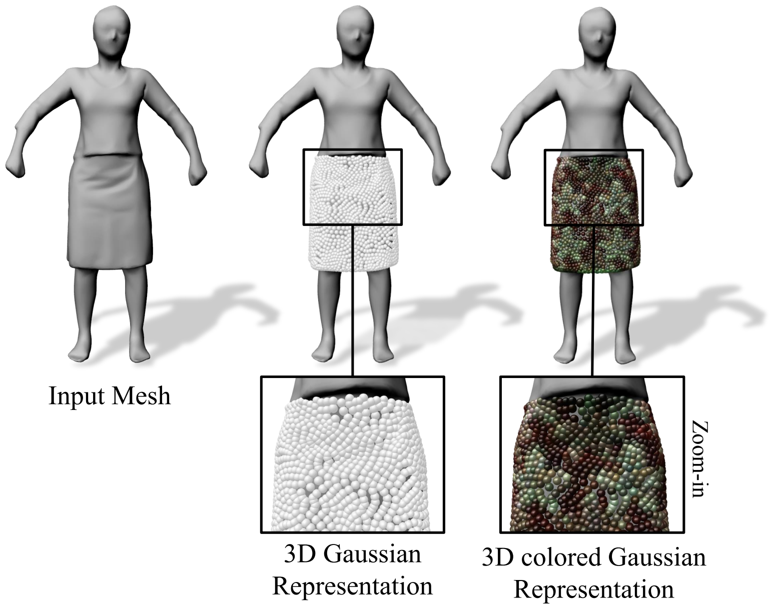

Input Mesh
3D Gaussian Representation
3D colored Gaussian Representation
Zoom-in

**Figure 4.3:** Surface Gaussian initialization pipeline. Starting from the input mesh (left), we populate the regions in which we want to recover the fine geometric detail (in this case, the skirt) with 3D Gaussians (center). Finally, a color is assigned to each 3D Gaussian based on the underlying reprojected pixel average from the best-camera view (right).

vertex visibility issue, we only consider the most direct camera, i. e. where the corresponding vertex normal and camera viewing direction align best, instead of taking all the camera views. The Surface Gaussians are then projected to the image from the best camera view and the underlying pixel color average is assigned as additional color attribute.

3D Surface Gaussians $\hat{G}_s$ can be easily and efficiently projected to the 2D image space of each camera view using the corresponding camera projection matrix $P$. The projected 2D mean $\mu_s$ and standard deviation $\sigma_s$ are computed using the formulas shown in Section 2.4.1.1. The same Surface Gaussian based representation is propagated to subsequent meshes in the deformation sequence. Temporal correspondences across topologically consistent meshes are easily inferred.

### 4.4.2 Images

Our implicit model for the input images $I(c)$ of all cameras views $c \in \{1 \dots n_c\}$, $n_c$ being the number of cameras, is obtained by assigning an Image Gaussian $G_i(x)$, $x \in \mathbb{R}^2$ to each image patch of all camera views. Section 2.4.1.1 describes in detail this procedure. Since we aim at capturing fine-scale details, we choose the quad-tree depth threshold $T_{\text{qt}}$ such that the smallest Image Gaussians has standard deviation $\sigma_i := 1$ pixel.

## 4.5 Surface Refinement

In this section, we describe our surface refinement approach. We employ an analysis-by-synthesis approach to refine the input coarse mesh animation, at every frame, by optimizing the following energy $\mathbf{E}(\mathcal{M})$ w. r. t. the collection of Surface Gaussian means $\mathcal{M} = \{\hat{\mu}_1, \dots \hat{\mu}_{n_s}\}$:

$$\mathbf{E}(\mathcal{M}) = E_{\text{sim}}(\mathcal{M}) - w_{\text{reg}} E_{\text{reg}}(\mathcal{M}) - w_{\text{temp}} E_{\text{temp}}(\mathcal{M}). \tag{4.2}$$

The term $E_{\text{sim}}$ measures the color similarity of the projected 2D Surface Gaussians with the 2D Image Gaussians for each camera view. $E_{\text{reg}}$ is used to keep the distribution of the Surface Gaussians geometrically smooth, whereas $w_{\text{reg}}$ is a user defined smoothness weight. The additional term $E_{\text{temp}}$ is used to temporally smooth the displacements of the Surface Gaussians over time to avoid visual artifacts such as jittering, whereas $w_{\text{temp}}$ is a user defined temporal smoothing weight.

We constrain the Surface Gaussians to only move along the corresponding vertex (normalized) normal direction $\hat{\mathbf{n}}_s$, i. e. the 3D mean $\hat{\mu}_s$ is defined by:

$$\hat{\mu}_s = \hat{\mu}_s^{\text{init}} + \hat{\mathbf{n}}_s k_s \in \mathbb{R}^3 \tag{4.3}$$

where $\hat{\mu}_s^{\text{init}}$ is the initial Surface Gaussian mean initialized as the vertex position $v_s$ at the beginning of each frame, and $k_s$ is the unknown vertex displacement. Constraining the motion of $\hat{\mu}_s$ along the corresponding normal brings two main advantages. On the one hand, it implicitly forces the Surface Gaussians to maintain a regular distribution on the surface, reducing the risk of self intersections due to crossing triangles. On the other hand, the consequent reduction of the optimization space, i. e. $n_s$ ($k_s$) unknown instead of $3 \times n_s$ ( $[\hat{\mu}_s]_x, [\hat{\mu}_s]_y$ and $[\hat{\mu}_s]_z$ ), results in higher performance as well as better-posed convergence.

Optimal displacements $k_s$ for each Surface Gaussian can be directly applied as vertex displacements along the corresponding surface normal to refine the model surface, both in terms of reprojected surface-image appearance agreement and space-time smoothness. We define each term of $\mathbf{E}(\mathcal{M})$ analytically and compute the corresponding derivatives with respect to the unknown displacements $k_s, \forall s \in \{1 \dots n_s\}$. The derivatives are:

$$\frac{\partial \mathbf{E}}{\partial \mathcal{M}} = \frac{\partial \mathbf{E}}{\partial k_s} = \frac{\partial}{\partial k_s}(E_{\text{sim}} - w_{\text{reg}} E_{\text{reg}} - w_{\text{temp}} E_{\text{temp}}) = \frac{\partial E_{\text{sim}}}{\partial k_s} - w_{\text{reg}} \frac{\partial E_{\text{reg}}}{\partial k_s} - w_{\text{temp}} \frac{\partial E_{\text{temp}}}{\partial k_s} \tag{4.4}$$

In the next sections, we describe each term in detail.

### 4.5.1 Similarity Term

We exploit the power of the implicit Gaussian representation of both input images and surface in order to derive a closed-form analytical formulation for our similarity term. The appearance and spatial similarity $\Phi_{i,s}$ of an Image Gaussian $G_i$ and projected Surface Gaussian $G_s$ pair can be mathematically formulated as the integral of the Gaussian product, weighted by the color similarity $T_{\Delta_c}(\delta_{i,s})$, as follows:

$$\Phi_{i,s} = T_{\Delta_c}(\delta_{i,s}) \left[ \int_{\Omega} G_i(x) G_s(x) \partial x \right]^2 \tag{4.5}$$

This equation has a similar mathematical definition to the overlap energy used for Volumetric Gaussians, explained in detail in Section 2.4.1.2. One of the advantages of using a Gaussian representation is that the integral in Equation 4.5 has a closed-form solution:

$$\Phi_{i,s} = T_{\Delta_c}(\delta_{i,s}) \left[ \int_{\Omega} \frac{1}{\sqrt{\pi \sigma_s \sigma_i}} exp\left( -\frac{1}{2} \frac{||x-\mu_i||^2}{\sigma_i^2} \right) \cdot \, exp\left( -\frac{1}{2} \frac{||x-\mu_s||^2}{\sigma_s^2} \right) \partial x \right]^2 \tag{4.6}$$

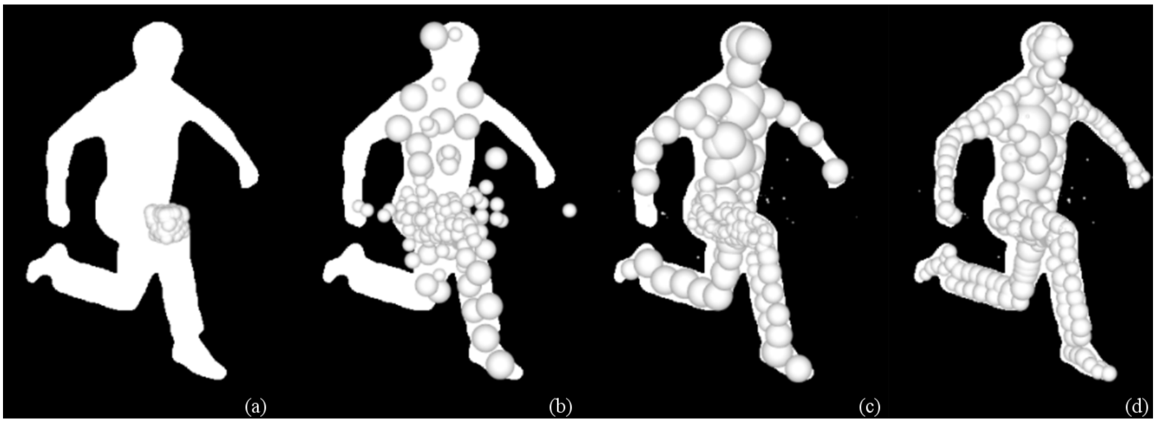


**Figure 4.4:** Volumetric Gaussians based shape approximation from silhouette at different optimization steps [143]. (a) Initialization, (b) after 100 iterations, (b) after 300 iterations and (d) after 10000 iterations.

$$\begin{aligned} &= T_{\Delta_c}(\delta_{i,s})\left[\frac{\sqrt{2\sigma_s\sigma_i}}{\sqrt{(\sigma_s^2+\sigma_i^2)}}exp\left(-\frac{1}{2}\frac{||\mu_i-\mu_s||^2}{\sigma_s^2+\sigma_i^2}\right)\right]^2 \\ &= T_{\Delta_c}(\delta_{i,s})2\frac{\sigma_s\sigma_i}{\sigma_s^2+\sigma_i^2}exp\left(-\frac{||\mu_i-\mu_s||^2}{\sigma_s^2+\sigma_i^2}\right) \end{aligned} \tag{4.7}$$

Notice that, due to the different mathematical definition of Surface Gaussians compared to Volumetric Gaussians, $\Phi_{i,s}$ has a more advantageous closed-form definition. The use of normalized Surface Gaussians with the chosen normalization factor allows to mathematically constraining the overlap $\Phi_{i,s}$ in the interval $[0,1]$, which has appealing properties concerning the next formulations' steps. $\Phi_{i,s}$ estimated with normalized Gaussian can be also used to optimize the standard deviation $\hat{\sigma}_s$ along with the mean $\hat{\mu}_s$ of the Surface Gaussians. This can be used for example when estimating optimal Surface Gaussians properties, i. e. location and size, during model initialization, as it has been demonstrated in Figure 4.4 for Volumetric Gaussians. Figure 4.5 compares the similarity landscape in 3D with and without normalization, visually proving the effectiveness of this improved formulation.

To compute the similarity of the surface model to the multi-view images $E_{\text{sim}}$, we first calculate the overlap of the set of Surface Gaussians against the set of Image Gaussians for each camera view, obtained by summing-up all overlaps $\Phi_{i,s}$, $\forall i,s$. Then, we normalize the result considering the number of cameras $n_c$ and the maximum obtainable overlap, which can be found counting out the Image Gaussians $\sum_i \Phi_{i,i} = \sum_i 1 = n_i^c$, $\forall c$:

$$E_{\text{sim}} = \frac{1}{n_c}\sum_{c=1}^{n_c}\left[\frac{1}{n_i^c}\sum_{i=1}^{n_i^c} min\left(\sum_{s=1}^{n_s}\Phi_{i,s},\ \ 1\right)\right] \tag{4.8}$$

such that $E_{\text{sim}} \in [0,1]$. The use of normalized Gaussians contributes in an improvement in performance (3% w. r. t. the unnormalized version), due to the reduced computational demand for $E_{\text{sim}}$. In this equation, the inner minimization implicitly handles occlusions on the surface as it prevents occluded Gaussian projections into the same image location to contribute multiple times to the energy. This is an elegant way for handling occlusion while preserving at the same time energy smoothness.

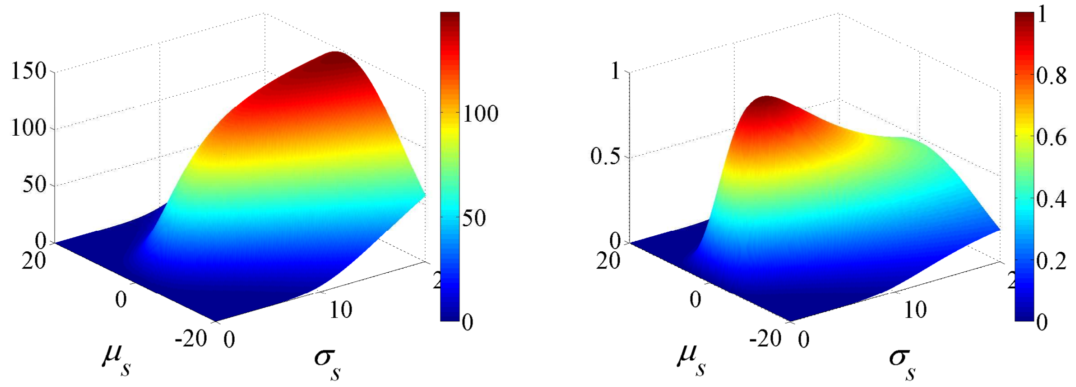


**Figure 4.5:** Similarity $E_{\text{sim}}$ evaluated for a Surface Gaussian with varying mean $\mu_s$ and standard deviation $\sigma_s$ against a fixed Image Gaussian having $\sigma_i = 5$ and $\mu_i = 0$. Left: the energy obtained without Gaussian normalization as in [147]. Right: the energy obtained by normalizing the Gaussians as explained in this chapter. Both plots have a maxima in $\mu_s = 0 = \mu_i$, however only the normalized energy on the right has $max$ $E_{\text{sim}} = 1$ for $\sigma_s = 5 = \sigma_i$, while the un-normalized plot has $\lim_{\sigma_s \to \infty} E_{\text{sim}} = \infty$.

This formulation has appealing analytical derivatives:

$$\frac{\partial E_{\text{sim}}}{\partial k_s} = \frac{1}{n_c} \sum_{c=1}^{n_c} \frac{1}{n_i^c} \sum_{i=1}^{n_i^c} \begin{cases} \frac{\partial \Phi_{i,s}}{\partial k_s} & \text{if} \sum_{s=1}^{n_s} \Phi_{i,s} < 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.9}$$

For the full derivation, refer to Section A.0.1.

### 4.5.2 Regularization Term

The regularization term constraints the Surface Gaussians in the local neighborhood such that the final reconstructed surface is sufficiently smooth. This is accomplished by keeping neighboring vertices displacements $k_s$ and $k_j, j \in \Psi(s)$, $\Psi(s)$ being the set of Surface Gaussians indices that are neighbors of $\hat{G}_s$, close to each other:

$$E_{\text{reg}} = \sum_{s=1}^{n_s} \frac{1}{|\Psi(s)|} \sum_{j \in \Psi(s)} T_{\Delta_d}(\delta_{s,j}) \, (k_s - k_j)^2 \tag{4.10}$$

The relevance of a pair $k_s$ and $k_j$ similarity is adjusted by the fixed weight $T_{\Delta_d}(\delta_{s,j})$, that is small ($\approx 0$) when the geodesic distance $\delta_{s,j} \in \mathbb{R}^+$ (measured in number of edges) between the vertices is large, and large ($\approx 1$) otherwise. In particular, $T_\Delta(\delta)$ is the Wendland radial basis function defined in Section 2.4.1 and $\Delta_d$ is the maximum allowed geodesic distance after which $T_{\Delta_d}$ drops to 0. Since we assume fixed surface topology for our experiments, $\delta_{sj}$ does not change, and in particular is constant with respect to the variable $k_s$. We compute the geodesic distance among all vertices and all possible neighbors only once for each sequence.

The effect of the minimization of $E_{\text{reg}}$ is to maintain a smooth surface where all close neighbors show similar displacements the more they are close to each other. Notice that the initial model surface with $k_s = 0, \forall s$ is assumed to be already smooth. Baked-in details cannot be smoothed out after optimization. A similar formulation in the case of free motion of the Surface Gaussian without any constraints along the normal would be harder to formulate, as it would require additional complex terms to guarantee smooth and regular surface distribution of the resulting vertex positions. The smooth derivatives of the regularization term are:

$$\frac{\partial E_{\text{reg}}}{\partial k_s} = \frac{4}{|\Psi(s)|} \sum_{j \in \Psi(s)} T_{\Delta_d}(\delta_{s,j})\,(k_s - k_j) \tag{4.11}$$

The corresponding derivation rules can be found in Section A.0.2.

### 4.5.3 Temporal Term

The temporal smoothing term is used to constraint the displacements $k_s$ over time, generating a smooth temporal deformation and avoiding jitter and artifacts. This term is defined as follows:

$$E_{\text{temp}} = \sum_{s=1}^{n_s} \left( \frac{1}{2}(k_s^{f-2} + k_s^{f}) - k_s^{f-1} \right)^2 \tag{4.12}$$

where $k_s^f$, $k_s^{f-1}$ and $k_s^{f-2}$ are respectively the normal displacement $k_s$ computed a current, previous and one before the previous frame. This formulation is inspired by the acceleration law, aiming at obtaining time consistent results with smooth acceleration. The smoothing term comes into play after computing the displacements for the first 2 frames, when the constants for the first frame $k_s^1$, and second frame $k_s^2$ are known.

Smooth derivatives of the temporal term are computed as:

$$\frac{\partial E_{\text{temp}}}{\partial k_s} = \frac{1}{2}(k_s^{f-2} + k_s^{f}) - k_s^{f-1} \tag{4.13}$$

The complete derivation rules are in Section A.0.3.

### 4.5.4 Optimization

Our energy function $\mathbf{E}$ can be efficiently optimized using an iterative gradient-based approach. For each iteration $t$ of the maximization process, we compute the derivative of $\mathbf{E}^t$ with respect to each $k_s, s \in \{1 \ldots n_s\}$, obtained summing-up all energy term derivatives together, following Equation 4.4.

To improve computational efficiency, we evaluate the overlap $\Phi_{i,s}$ only for visible Surface Gaussians from each camera view. Explicit visibility computation is performed only once at the beginning of each frame, by considering each Surface Gaussian as simple vertices. The implicit occlusion handling approximation takes care of consistently handling new occlusions that might arise during optimization. The Gaussian overlap is then computed against visible projected Surface Gaussians and Image Gaussians in a local neighborhood, by considering only the closest Image Gaussians up to a distance threshold $T_{\text{dist}}$ in number of pixels and an HSV color distance threshold $T_{\text{color}}$.

We efficiently optimize our energy function $\mathbf{E}$ using a *conditioned gradient ascent* approach. The general gradient ascent method is a first-order optimization procedure that aims at finding local maxima by taking steps proportional to the energy gradient. It uses a scalar factor, the conditioner $\gamma$, associated to the analytical derivatives that increases (respectively decreases) step-by-step when the gradient sign is constant (respectively fluctuating).

At each optimization step $t$ we update the displacements $k_s^t$ based on the current normalized gradient $\overline{\nabla}(\mathbf{E})^t$ and conditioner $\gamma^t$

$$k_s^t = k_s^{t-1} + \overline{\nabla}(\mathbf{E})^t \gamma^t \tag{4.14}$$

where initial $k_s^1 = 0, \forall s = 1 \ldots n_s$, and $\overline{\nabla}(\mathbf{E})^t$ is the normalized gradient computed considering the maximum $\nabla(\mathbf{E})^t$ among all $s = 1 \ldots n_s$ at the current step to ensure values in the interval $[0,1]$:

$$\overline{\nabla}(\mathbf{E})^t = \frac{\nabla(\mathbf{E})^t}{max\left(\nabla(\mathbf{E})^t, s = 1 \ldots n_s\right)} \tag{4.15}$$

The conditioner is initially set to $\gamma^1 = 0.1$, then we update it based on the gradients at previous and current step as follows:

$$\gamma^{t+1} = \begin{cases} min\left(1.2\gamma^t, \frac{\Delta_\gamma}{\overline{\nabla}(\mathbf{E})^t}\right) & \text{if } \left(\overline{\nabla}(\mathbf{E})^{t-1}\overline{\nabla}(\mathbf{E})^t\right) > 0 \\ 0.5\gamma^t & \text{otherwise} \end{cases} \tag{4.16}$$

where $\Delta_\gamma = 1\ mm$ is the maximum step size. We additionally check if the gradient has dramatically decreased in magnitude, and if so further reduce the conditioner based on the gradient ratio:

$$\gamma^{t+1} = 0.25\frac{\overline{\nabla}(\mathbf{E})^{t-1}}{\overline{\nabla}(\mathbf{E})^t}\gamma^t \tag{4.17}$$

The use of the conditioner brings three main advantages: faster convergence to the final solution, smoother convergence slope without undesired zigzagging, and controlled size of the analytical derivative. Such benefits are depicted in Figure 4.6, showing the impact of the conditioner on the convergence curve trend. For each frame, we perform at least $t := 5$ and at most $t := 1000$ iterations, and stop when

$$\frac{|\mathbf{E}^t - \mathbf{E}^{t-1}|}{max(1, \mathbf{E}^t, \mathbf{E}^{t-1})} \leq 1e^{-8}. \tag{4.18}$$

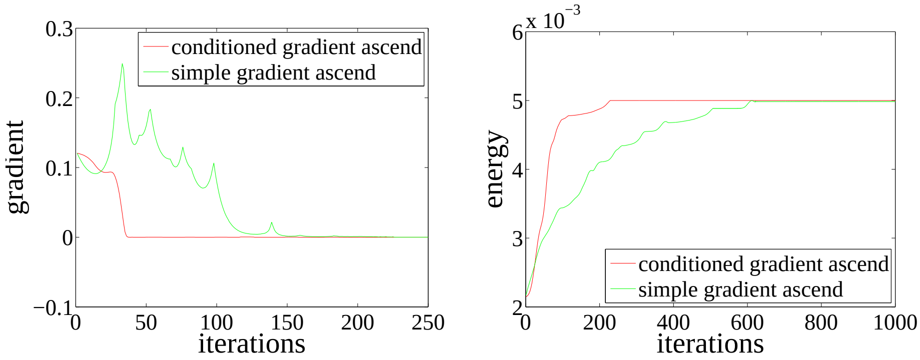


**Figure 4.6:** Comparison between conditioned gradient ascent (red) used in this paper, and simple gradient ascent (green) optimization. Left: gradient intensity $\overline{\nabla}(\mathbf{E})_s$ per iteration of a single parameter $k_s$. Right: the energy **E** per iteration. The conditioned gradient ascent has faster convergence to the local maxima while keeping a smooth gradient curve.

Once the convergence has been reached (typically around iteration $n_t := 200$ for all sequences, see Figure 4.6), we update the vertex positions of the input mesh at the current frame by simply

| Sequence | *skirt* | *dance* | *handstand* | *wheel* | *pop2lock* | *synthetic* |
|---|---|---|---|---|---|---|
| Published by | Gall et al. [64] | | | | S.&H. [162] | Us |
| Cameras | 8 | | | | | 10 |
| Frames | 640 | 574 | 150 | 120 | 250 | 1 |
| Frame rate | 40 Hz | | | | | n/a |
| Camera type | PhaseSpace Vision Camera | | | | | n/a |
| Resolution | 1004×1004 | | | | 1920 × 1080 | 1280 × 720 |
| Vertices | 12095 | 3430 | 12095 | 12095 | 3880 | 42 |

**Table 4.1:** Details for each sequence. This table summarizes the main settings for all the used sequences, as well as optimization-related settings, e. g. approximate amount of 3D and 2D Gaussians per frame.

displacing them along the corresponding normal using the found optimal $k_s^f := k_s^{n_t}$. Note that in practice, when rendering the final resulting mesh sequence, we add an extra $\varepsilon$ to the computed vertex displacement $k_s^f$. This is needed to compensate for the small surface bias (shrink along the normal during optimization) that is due to the spatial extent of the Gaussians. More details are given in Section 4.6. Hence, we update the vertex position as:

$$v_s' = v_s + \hat{\mathbf{n}}_s \cdot (k_s^f + \varepsilon) \tag{4.19}$$

where $v_s$ is the original location of the vertex, $\hat{\mathbf{n}}_s$ is the corresponding unchanged normal and $\varepsilon = \sigma_s$ throughout this work.

## 4.6 Experimental Results

In this section, we describe our datasets and provide detailed qualitative and quantitative evaluation of the results compared to previous work.

### 4.6.1 Test Sequences

We test our approach on five different datasets: *skirt*, *dance*, *pop2lock*, *handstand* and *wheel*. Input multi-view video sequences, as well as camera settings and initial coarse mesh reconstruction were provided by Gall et al. [64] and Starck and Hilton [162]. All sequences are recorded with 8 synchronized and calibrated cameras and number of frames ranging between 120 and 640, see Table 4.1. Input coarse meshes are obtained using techniques based on sparse feature matching, shape-from-silhouette and multi-view 3D reconstruction, described in [64, 162], and therefore lack of surface details.

In order to refine the input mesh sequences, we first subdivide the coarse input topology by inserting additional triangles and vertices until we reach sufficient density. This preprocessing step is performed in order to increase the amount of surface details that can be captured during optimization. We typically subdivide more in presence of highly textured apparel, and less in case of plain colored surfaces. The skirt in the *skirt* sequence for instance is highly subdivided, due to the high details recovering potentials, while the monochromatic shirt and pants in the *dance* sequence are mostly left with their original coarse subdivision. We then generate a collection of Gaussians on the surface, as described in Section 4.3. Since most of the fine-scale deformations happen on the clothing, we focus on the refinement on those areas, generating Surface Gaussians only for the corresponding vertices. The number of vertices indicated in Table 4.1 correspond to the amount of Surface Gaussians created for each sequence.

To visually enhance the capabilities of our refinement approach in capturing fine-scale surface details, we create 3 additional datasets by smoothing the input meshes of the sequences *dance*, *skirt* and *pop2lock*. By doing so, we eliminate most of the baked-in surface details and use the over-smoothed mesh animations as input to our system. For the quantitative evaluation of our technique, we additionally generate a *synthetic* sequence. See Section 4.6.5 for more details.

### 4.6.2 Runtime

The biggest overhead of our Gaussian-based refinement is the evaluation of the similarity term, which has to be computed for each pair surface-image Gaussian in all the views. This quantity has to be estimated iteratively for each gradient-descent based optimization step until convergence. The complexity of our approach depends on the number of iterations $n_t$ performed by the gradient-based solver, as well as the number of Image Gaussians $n_i^c$ per view and Surface Gaussians $n_s$, $O(n_t \sum_{c=1}^{n_c} n_i^c n_s)$. These 3 quantities variate depending on the sequence, frame and specific camera. The order of complexity is obtained considering the maximum among the possible values for each sequence. For each frame, at most $n_t = 1000$ iterations are performed, see Section 4.5.4, and at most each pixel is taken as Image Gaussian, resulting in $n_i^c = 1004 \times 1004$ for the sequences from Gall et al. [64] and $n_i^c = 1920 \times 1080$ for the sequences provided by Starck and Hilton [162].

However, convergence to a solution is typically reached in less iterations, after around $n_t = 200$ iterations, see Figure 4.6. The number of Image Gaussians is also smaller than the number of pixels for each camera, as for each Surface Gaussian we only take into account the closest Image Gaussians in terms of spacial and color distance, as described in Section 4.5.4. The average number of Image Gaussians per frame is close to $n_i^c = 10000$, which is $\approx 2$ order of magnitude smaller than the total number of pixels. It can be easily shown that our approach has linear complexity with respect to varying Surface Gaussians.

We evaluate the performance of our system on an Intel Xeon Processor E5-1620, Quad-core with Hyperthreading and 16GB of RAM. Table 4.2 summarizes the performance we obtain for the 6 tested sequences. The computation time can be further reduced by parallelizing orthogonal optimization steps, such as the evaluation of the overlap $\Phi_{i,s}, \forall i,s$, described in Section 4.5.1. Our algorithm is particularly suited for implementation on GPU, which we believe has a strong impact on the time performance.

### 4.6.3 Qualitative Results

In Figure 4.7 and Figure 4.8, we provide some examples of surface refinement results obtained with our method. Input to our refinement approach, on top of the multi-view images and corresponding camera calibration, are the temporally consistent deformation sequences obtained from the methods of [64, 162]. Our visual results demonstrate that our approach is able to plausibly reconstruct additional fine-scale details, e. g. the wrinkles and folds in the skirt, and it produces closer model alignment to the images than the baseline methods.

The *skirt* sequence is the most suited for our multi-view surface refinement technique thanks to its highly textured regions and accurate alignment with respect to the images. We focus on the

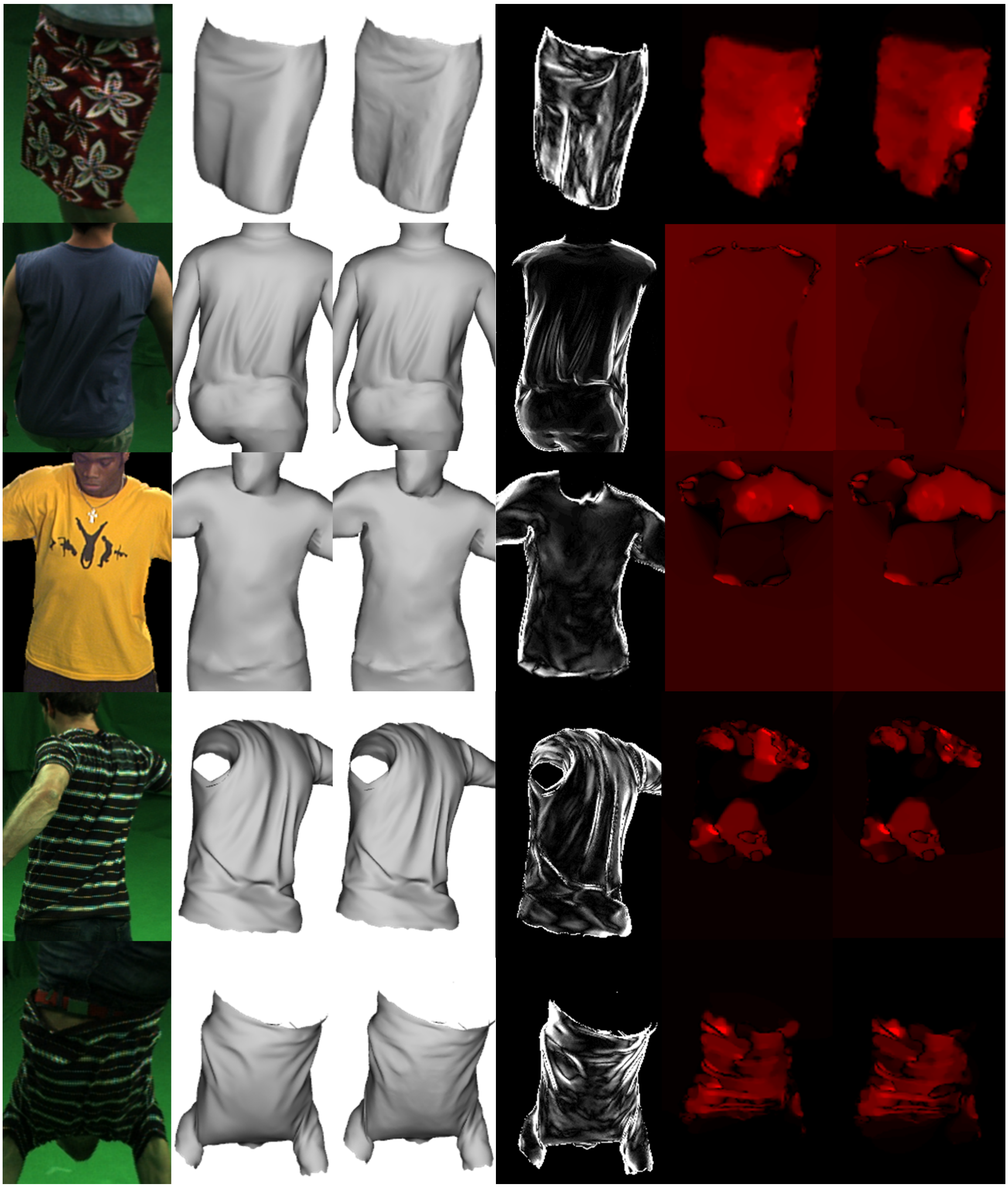


**Figure 4.7:** Results of our refinement approach for (from top to bottom) the *skirt*, *dance*, *pop2lock*, *wheel* and *handstand* animation sequences from an unused camera. From left to right: input image, baseline mesh, refined mesh, surface difference between baseline and refined, flow magnitude for the reprojected input mesh and the reprojected output mesh against the original input image.

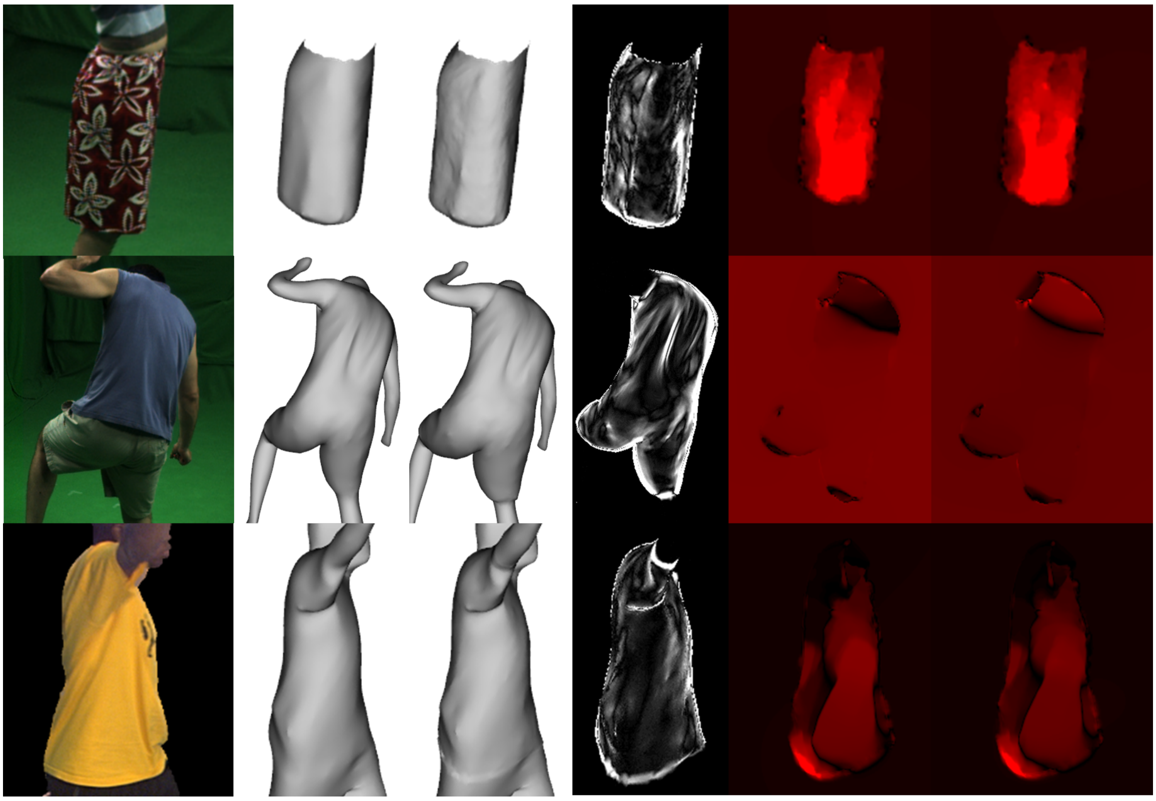


**Figure 4.8:** Results of our refinement approach for the smooth *skirt* (top), *dance* (middle) and *pop2lock* (bottom) animation sequences from an unused camera. From left to right: input image, baseline mesh, refined mesh, surface difference between baseline and refined, flow magnitude for the reprojected input mesh and the reprojected output mesh against the original input image.

refinement of the skirt surface region, which is highly textured. Textures lead to high-color variance of the Surface Gaussians, which in turn allows to estimate finer scale details and more accurate alignment, even in the inner parts, far from the projected borders. As shown in Figure 4.7, row 1, and Figure 4.8, row 1, our method is able to capture additional fine-scale details for both the *skirt* and the smoothed *skirt* sequences using the default parameter settings depicted in Table 4.3. The incorporated details are best visible in the smoothed sequence version.

The *dance* sequence shows strong deformations on the subject's clothing during the jumping motions, see Figure 4.7, row 2, and Figure 4.8, row 2. We successfully refine the input geometry to include additional cloth dynamics, e. g. floating shirt, also visible in the video. Inner details are only marginally captured when a shading effect appears, due to lack of textures, and are best visible in the smoothed *dance* sequence. Most of the captured details are visible along the reprojected borders, where the alignment with respect to the images appears to be improved. For both the original and the smoothed animation, we use the default parameters setting in Table 4.3.

The *pop2lock* sequence is challenging to refine with our approach, due to the large homogeneously colored regions on the subject's shirt. On top of that, the input mesh presents severe misalignments with respect to the input multi-view images with several time-consistency artifacts throughout the multi-view video. Similar to the *dance* sequence, the majority of the captured details are localized on the reprojected borders, as seen in Figure 4.7, row 3, and Figure 4.8, row 3. Our refinement approach is unable for this sequence to improve the silhouette overlap consistently, due to the strong shirt deformations, e. g. folding and creasing.

Both the *handstand* and the *wheel* sequences contain textured regions, represented by the stripes in the shirt, which help in improving the surface details by effectively guiding the Surface Gaussian along the shirt deformation. The *handstand* sequence is particularly challenging for our refinement approach, since the shirt is strongly deformed during the performance, especially in the upside-down pose after frame 50. On the other hand, the *wheel* sequence shows an example of fast motion, where the underlying geometry is temporally misplaced, resulting in false underlying colors and therefore inaccurate refinement, as it is shown in the quantitative evaluation. Nevertheless, our refinement approach captures inner details consistently for both of the sequences.

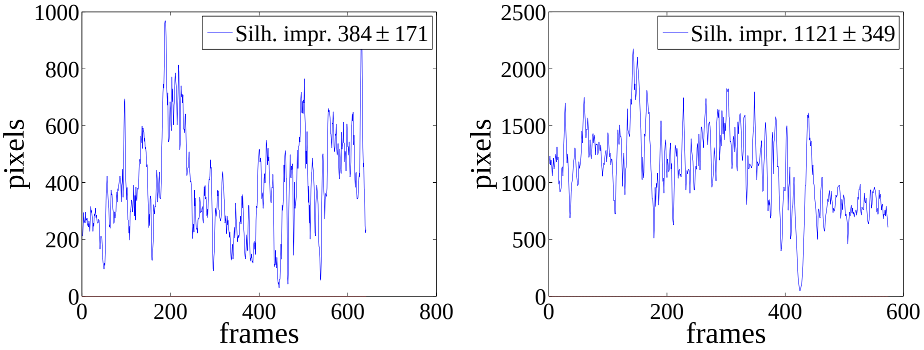
### 4.6.4 Comparison

In order to compare our approach with the baseline methods on sequences for which no ground-truth geometry is available, we compute the optical flow error. To this end, we first texture the baseline and our resulting mesh models by assigning Surface Gaussian colors to the corresponding vertices. Then, we use the classical variational approach by Brox et al. [23] to generate displacement flow vectors between the input images of a single camera view and the reprojected textured mesh models for each frame and pixel. For each frame, we then compute the average displacement error, by dividing the sum of the displacement norm by the number of pixels. The evaluation is performed on a camera view that is excluded from the optimization.

As shown in the Table 4.2, row 3, our method significantly decreases the average flow displacement error for all the sequences, leading to quantitatively more accurate results compared to the baseline (input) methods. In terms of percentage, the registered improvement lies between 3% and 11%. High percentage of flow error reduction between the baseline and the refined meshes is also obtained in the purposely smoothed datasets. This confirms that our approach successfully recovers true surface detail even when using very coarse geometry. Figure A.2 shows the corresponding normalized flow error per frame compared to the unrefined input meshes and visually shows the average improvement

| Sequence | *skirt* | | *dance* | | *handstand* | *wheel* | *pop2lock* | |
|---|---|---|---|---|---|---|---|---|
| Input Meshes | orig | smooth | orig | smooth | orig | orig | orig | smooth |
| Flow Improvement [%] | 4.8 | 4.7 | 9.1 | 2.7 | 11 | 5.1 | 4.9 | 3 |
| Silh. Improvement [pixel] | 59 | 384 | 294 | 1121 | −80 | 35 | 96 | −16 |
| Total [pixel] | ≈16K | | ≈35K | | ≈19K | | ≈30K | |
| Timing [min/frame] | ≈20 | | ≈3 | | ≈30 | | ≈3 | |

**Table 4.2:** Flow and silhouette average improvements produced by our refinement approach compared to the baseline methods. The average flow displacements improvement is express in percent, while the silhouette improvement in number of pixels over the approximate total amount of true re-projected cloth pixels over all the views.



(a) *smooth skirt*, Cloth px ≈ $16K$ (b) *smooth dance*, Cloth px ≈ $35K$

**Figure 4.9:** Silhouette improvement of the refined meshes for the smoothed *skirt* and *dance* datasets. In each graph: pixel improvement with respect to the specified input sequence per frame. Manually estimated true cloth pixels to be used as a reference for the silhouette pixel improvement are stated in the captions of each sequence.

per frame also stated in Table 4.2. Notice that most of the baseline input mesh sequences, e. g. the *skirt* sequence, already provide closely-aligned input meshes. In these cases, improvements introduced after refinement are marginal in terms of flow displacement error. Qualitative figures nevertheless show that our approach is able to capture the deformation dynamics happening on the clothing in all sequences, as well as fine-scale details, especially for highly textured regions. Details on the inner plain-colored regions, e. g. shirt folds in the *dance* sequence, are hard to reproduce with our approach, since they lack of sufficiently distinctive color information in the Surface Gaussians neighborhood.

Especially for the latter cases, quantitative evaluation demonstrates that our approach also improves the boundary alignment of the reconstructed meshes with respect to the input image, without explicitly relying on any precomputed foreground segmentation. This is especially visible in the pre-smoothed sequences, also shown in Figure A.1 and Figure 4.10. As opposite to most of the silhouette-based approaches [162], our optimization approach smoothly and continuously improves the alignment to the silhouettes, without using error-prone correspondence finding, discrete optimization or complex background subtraction. We visually show silhouette reprojection improvement in Figure 4.10. The silhouette improvement per frame is computed by counting the false positive and false negative (i. e. wrong) pixels in the original and refined sequences, obtained by subtracting the ground-truth silhouette image pixels with the reprojected mesh pixels from an unused camera. Then, we take the difference between original and refined false pixels, which implicitly resembles the number of newly

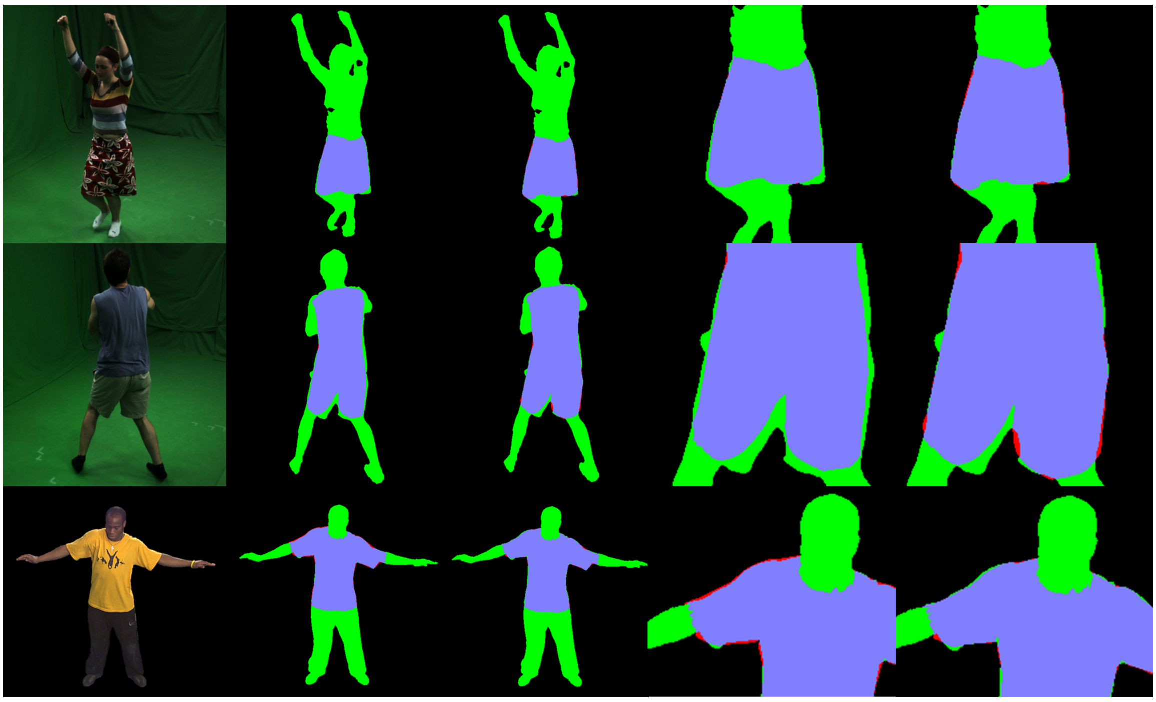
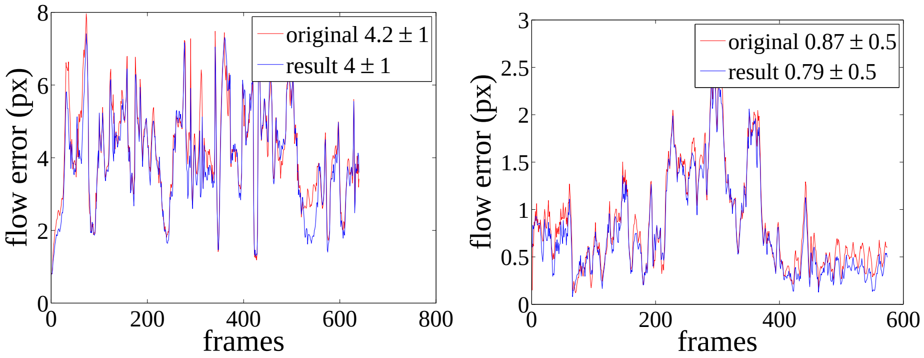


**Figure 4.10:** Silhouette overlap evaluation of the refined meshes shown for *skirt* (top), *dance* (middle) and *pop2lock* (bottom) sequences. From left to right: input image, silhouette overlap of the baseline (input), silhouette overlap of the refined mesh, zoomed-in baseline overlap, zoomed-in refined overlap. The overlap is estimated from unused views. We used the following color code for the silhouette overlap: Purple: true positives, Green: false negatives, Red: false positives.



(a) *skirt* (b) *dance*

**Figure 4.11:** Average flow displacement error of the refined meshes for the *skirt* and *dance* sequence.The plots show normalized flow error per frame for the baseline (red) and refined (blue) mesh sequences.

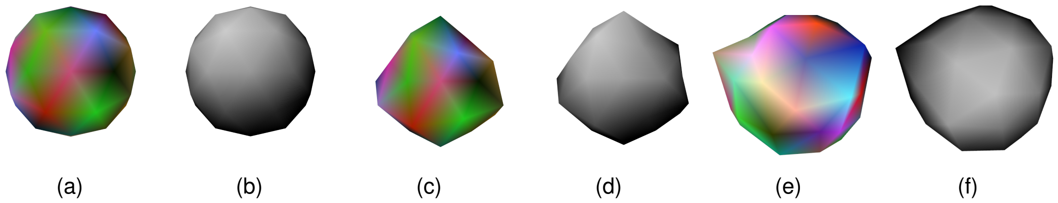


**Figure 4.12:** Results of our refinement approach for the synthetic sequences. (a) target shape with no additional displacement, (b) resulting shape, (c) target with normal displacement, (d) resulting shape, (e) target with random displacement and (f) resulting shape.

found correct pixels. Notice, that our computations are performed using the available full-body silhouette images, although, we just refine the clothing parts. For comparison, we manually count the average number of true pixels in the clothing areas for each sequence as a reference, see Table 4.2, row 5.

### 4.6.5 Quantitative Evaluation

We quantitatively evaluate the performance of our approach on a *synthetic* sequence, consisting of a 42-vertices sphere model with randomly generated colors. We build three different target scenarios: the unchanged sphere, the sphere with randomly displaced vertices along the corresponding vertex normals, and the sphere with randomly displaced vertices without any normal constraint. Each scenario is rendered from 10 arbitrary points of view with a resolution of $1280x720$ and used as input multi-view images for our refinement approach. The initial unchanged sphere geometry with true pre-assigned colors is taken as the starting surface to be refined. Finally, we compare the output of each scenario with the ground truth geometry to numerically assess the quality of our results.

Figure 4.12 shows the described target $(a,c,e)$ and refined $(b,d,f)$ mesh for each scenario. For the first unchanged scenario, we use the default parameter settings from Table 4.3, while for the remaining two we used $w_{\text{reg}} = 0$ aiming at capturing the randomly generated displacements, without introducing any temporal nor spatial smoothness constraints. For all scenarios we have used $T_{\text{dist}} = 90$ pixels to capture also larger displacements. We compute exact average Euclidean displacement error,

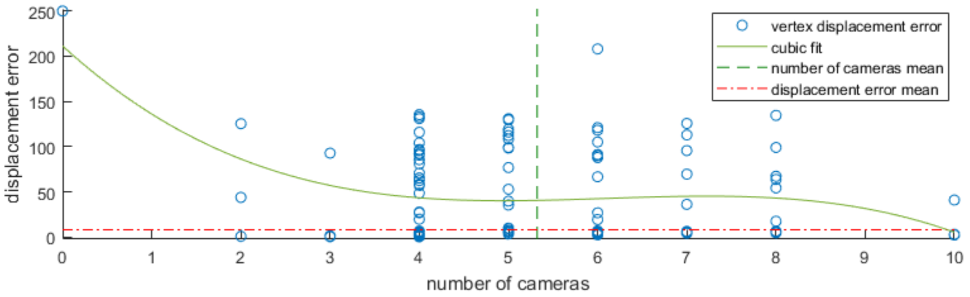


**Figure 4.13:** Vertex displacement error curve (green) when using different number of cameras. The plot shows $3 \times 42$ blue circles, each sampling the displacement error of a single optimized vertex in the 3 *synthetic sphere* sequences described in Section 4.6.5. The $y$-axis indicates the number of cameras that see a certain vertex sample. The plot also shows the average displacement error (dashed red) as well as average visibility (dashed green).

by summing up the vertex displacement norm from the ground truth divided by the number of vertices. The average error is then estimated as a percentage of the total object volume size. Even with large distortions, e. g. the severe displacement introduced in the third scenario with random displacements, applied to the original geometries, our method is able to recover the fine geometric details with average errors lower than $\leq 7\%$ of the volume size. For the remaining scenarios, the estimated error lies below $1.84\%$ and $0.22\%$ respectively for the normal constrained random displacements and the unchanged case. Notice that due to the surface normal constraints introduced in the design of our refinement approach, for the third scenario, i. e. with random normal-unconstrained displacements, we got a normal-constrained approximation of the target shape, which justifies the increased error w. r. t. the shape volume.

While experimenting with the scenarios, we make several observations. The initially assigned color of the Surface Gaussians has a direct impact to the performance of the proposed surface capture approach. While the used HSV color space helps in keeping our surface capture approach relatively robust to slight color changes due to illumination, larger appearance changes caused by e. g. shadows and global light changes cannot be handled properly with the current formulation.

Testing revealed that the error increases when reducing the number of views, see Figure 4.13. This finding suggests that without providing more advanced surface smoothness term to handle missing depth, our approach is impractical for settings with monocular, few-cameras or badly distributed camera settings. Randomly placed cameras around the *synthetic sphere* resulted in an average of visibility of 5.23 cameras per vertex, see the plot in Figure 4.13. Evidently little visible vertices cause a consistent loss on the overall quantitative performance average.

We further validate the additional $\varepsilon$ used to correct the displacements, described in Section 4.5.4. An increase of $\varepsilon = \sigma_s$ of the found displacements, helps to accommodate for the bias in the displacements, as it is also shown in Figure 4.14. The average errors relative to the total volume size for each refined mesh with added epsilon are $0.22\%$, $1.84\%$ and $7.10\%$, against the unchanged $0.69\%$, $1.95\%$ and $7.13\%$ estimated from the optimizer only, i. e. without additional epsilon increase.

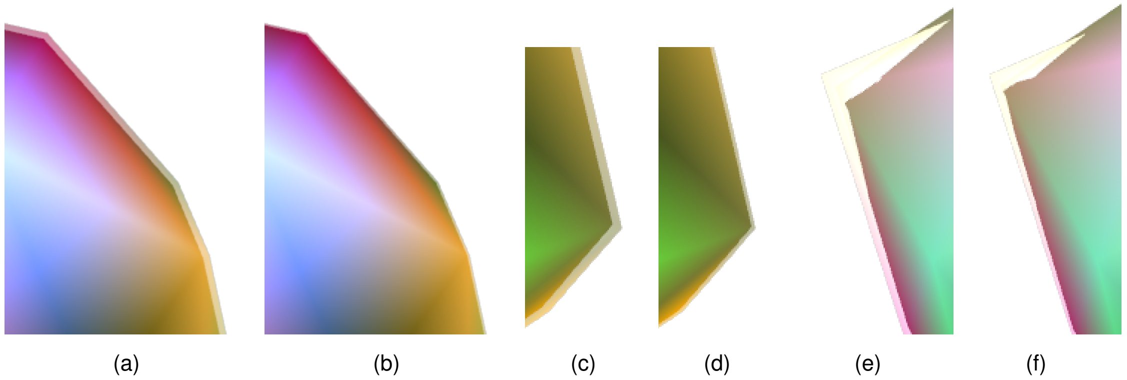


**Figure 4.14:** Effects of adding a displacement along the normal of the refined meshes equal to the standard deviation of the Surface Gaussians. The figure shows the zoomed-in overlap between the input (semi-transparent) and resulting (colored) mesh of the synthetic sequence first without and then with additional displacement. (a,b) overlap from the first unchanged scenario with and without epsilon, (c,d) overlap from the normal-displaced scenario with and without epsilon and (e,f) overlap from the randomly displaced scenario with and without epsilon.

| Parameter | Description | Value interval | Default Value |
|---|---|---|---|
| $w_{\text{reg}}$ | Regularization weight | $[0,1]$ | $5e^{-7}$ |
| $w_{\text{temp}}$ | Temporal weight | $[0,1]$ | $1e^{-7}$ |
| $\sigma_s$ | Standard deviation, $\forall G_s$ | $(0,\infty)$, $[mm]$ | 5 |
| $T_{\text{qt}}$ | Quad-tree depth threshold | $[0, log_2(min(I.width, I.height))]$ | $max$ |
| $T_{\text{fuse}}$ | Color fusion threshold | $[0,\infty)$ | 0.05 |
| $T_{\text{color}}$ | Color similarity threshold | $[0,\infty)$ | 0.15 |
| $T_{\text{dist}}$ | Threshold on pixel distance | $[0,\infty)$, $[px]$ | 30 |
| $\Delta_d$ | Max geodesic distance | $[0,\infty)$, $[\#edges]$ | 2 |

**Table 4.3:** User-defined parameters of the energy function **E**, together with their description, value interval and default value.

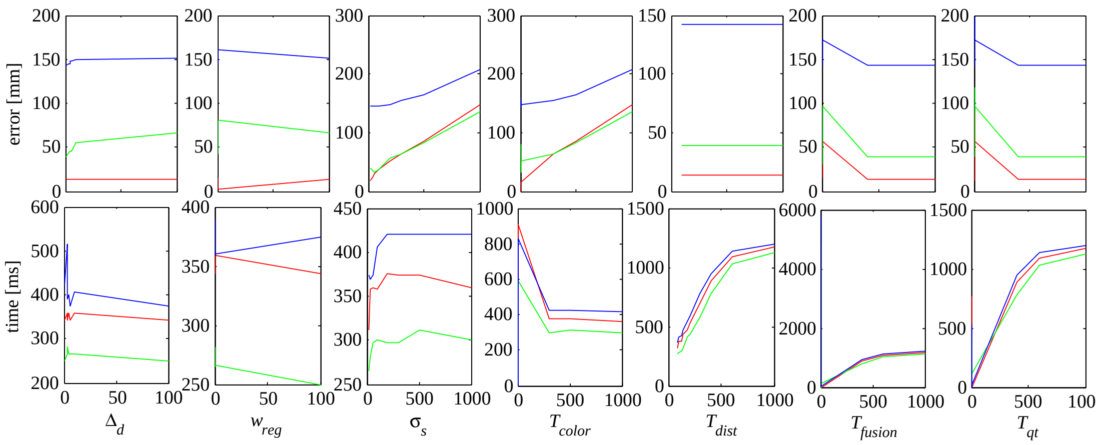


**Figure 4.15:** Influence of the parameters of the energy function **E** on the reconstructed error and computational time, in the *synthetic* dataset. The reconstruction error is estimated as average Euclidean displacement error from the given ground-truth in three different scenarios: input and target meshes are equal (no displacement, red lines), the target has a random displacement along the corresponding normal (normal displacement, green lines), the target has random displacement which may deviate from a normal displacement (random displacement, blue lines).

### 4.6.5.1 Parameters

In order to test the influence of the parameters used in our energy formulation, we run several experiments based on the *synthetic* dataset described in Section 4.6.5. For each user-defined parameter, see Table 4.3, we change the parameter value within its pre-defined interval, while keeping the remaining parameters fixed, and run our refinement approach with this setting. The set of plots in Figure 4.15 show the error and time performance obtained by varying each parameter.

As expected increasing $w_{\text{reg}}$ improves the static sphere scenario results, deteriorating the remaining ones. To enable the capture of details while preserving a certain degree of surface smoothness, we must choose a proper non-null regularization weight, which avoids over-smoothing.

By increasing the standard deviation $\sigma_s$ size, we obtain slightly worse reconstructions. We believe this is due to the fact that the Gaussian color is taken from the corresponding vertex color of the ground-truth mesh, and not from the underlying pixel average. In this case, the underlying pixel color poorly matches the Surface Gaussian color, leading to displacement errors.

The color similarity threshold $T_{\text{color}}$ restricts the evaluation of our similarity function w. r. t. a Surface Gaussian to a subset of Image Gaussians with similar color properties. The larger the threshold the more Image Gaussians are included in the evaluation, leading to an increasing computational time. The distance threshold $T_{\text{dist}}$ value highly depends on the sequence and mesh scale. Further increasing its value beyond its default ($T_{\text{dist}} = 90$ pixels for the *synthetic* sequence) does not improve qualitatively the reconstruction, instead it slows down the performance. On the other hand the fusion threshold $T_{\text{fusion}}$ and the quad-tree depth threshold $T_{\text{qt}}$ only produce accurate reconstruction results for high values, which in turn decrease the time performance. The smoothing weight $w_{\text{temp}}$ improves the stability of the refinement in time, without further improving the quality of the resulting surface.

## 4.7 Discussion and Limitations

The proposed refinement approach smoothly captures the surface details and improves the alignment of the input mesh sequences with the multi-view images consistently and robustly. Results on various testing sequences qualitatively and quantitatively show the effectiveness of the proposed approach to capture small details such as cloth wrinkles on dynamic sequences with loose clothing. The proposed formulation has a smooth correspondence-free energy that can be efficiently and effectively optimized using gradient-based optimization.

For the applicability of the proposed refinement approach, there are few assumptions. We assume the input mesh sequence to be sufficiently accurate, such that smaller details can be easily and correctly captured by displacing vertices along their corresponding vertex normals. In cases where the input reconstructed meshes present misalignments with respect to the images (e. g. in the *pop2lock* sequence) or if it is necessary to reconstruct stronger deformations, then our method is unable to perform adequately. Our refinement might be reformulated allowing more complex displacements, e. g. , removing the normal constraint. However, such weaker prior requires more complex regularization formulation in order to maintain a smooth surface, also to handle unwanted self-intersections and collapsing vertices. An example of more advanced regularization formulation to handle free surface deformation is described in the next Chapter 5. The increased number of parameters to optimize for (i. e. 3 times more, when optimizing for all 3 vertices dimensions, $x$, $y$ and $z$) would spoil computational efficiency and raise the probability of getting stuck in local maxima solutions. The risk of returning local maxima solutions is still high when employing local solvers (e. g. gradient ascent) on non-convex problems as in our case. A possible solution is to use more advanced solvers, e. g. global solvers, when computational efficiency is not a requirement.

As we demonstrate in the previous section, our approach is unable to densely refine plain colored surfaces with little texture (e. g. the *pop2lock* and *dance* sequences). In fact, in these cases, the corresponding color variability of the Surface Gaussians in the inner surfaces is too small to provide any clue for the optimizer. A solution here could be to employ a more complex color model that takes into account, e. g. , illumination and shading effects, at the cost of increased computational expenses. For sequences with little texture, we have although found that our approach improves the reprojected surface borders alignment with the images consistently. The employed surface-image similarity formulation is well suited to recover inner details in presence of texture, nevertheless the estimated border alignment resulting from optimization is sub-optimal and requires adjustments, i. e. adding $\varepsilon$ to the found displacements. In fact, by design, Gaussians tend to align towards higher density regions, rather than along the borders, making them unsuitable for exact extremities approximation. A better mathematical formulation for border alignment is described in the next Chapter 5.

The used temporal smoothing term analytically formulates smoothing in time based on a window of 3 frames, that might be insufficient to fully eliminate time inconsistencies, e. g. for videos captured at high frame rates. We would like to further investigate into the impact of a larger window size on the overall time consistency of the reconstruction. On top of that our temporal smoothing approach only keeps the optimized displacements smooth in time and cannot correct time inconsistencies in the original input geometry. This limitation is particularly visible in the sequence $pop2lock$.

## 4.8 Conclusion

In this chapter, we have presented an effective method for capture of deforming meshes with fine-scale time-varying surface detail from multi-view video recordings. Our approach refines the input coarse meshes on all vertex positions by adding fine-scale deformation present in the original video. The proposed model-to-image consistency energy function uses an implicit representation of the deformable mesh using a collection of Gaussians for the surface and a set of Gaussians for the input images, enabling a smooth closed-form energy with implicit occlusion handling and analytic derivatives.

We qualitatively and quantitatively evaluated our refinement strategy on 5 input sequences, initially obtained using 2 different state-of-the-art 3D reconstruction approaches: a template-free method and template-based method, which originally tend to produce smooth results that lack fine-scale details. We demonstrated that in both cases the proposed method successfully recovers true fine-scale detailed geometry and improves the surface border alignment w. r. t. to the multi-view images. Additionally, we have also shown the performance of our method on synthetic data, which we manually modeled and smoothed to create the ground truth. Numerical evaluation confirms the effectiveness of our refinement approach compared to previous work quantitatively.

This chapter focused on a surface capture solution based on input coarse animations. The next chapter demonstrates the capabilities of the refinement approach, described in this chapter, to be successfully applied in less constrained outdoor scenarios. Additionally, the next chapter shows how combining skeleton with surface capture can boost robustness to various outdoor environments.

# Chapter 5

# Joint Skeleton-Surface Capture

## 5.1 Introduction

In the previous chapter, we introduced a new approach for dense surface refinement of coarse animations. This chapter extends this refinement approach by proposing a full performance capture method that simultaneously recovers both the coarse skeletal and the fine-scale surface of a performing actor in general outdoor scenarios. Several markerless human performance capture methods have been proposed to reconstruct people in their general, potentially loosely deforming, apparel, from multi-view RGB video [20, 40, 64, 162, 180]. Some state-of-the-art methods reconstruct highly detailed 3D mesh sequences by fitting a 3D template to the observed performance [40, 64, 179]. More general methods reconstruct per-frame geometry independently and without a prior model [60, 162, 180]. Most high-quality reconstruction methods fail on footage recorded in general outdoor scenes, as they expect constant lighting and crisp foreground-background subtraction, which is best achieved in front of static indoor green screens. Aiming to overcome this limitation, recent research in joint segmentation and reconstruction [170, 123, 124] successfully reconstructed deforming objects in less constrained setups. However, resulting 3D mesh detail is significantly lower than for the previous in-studio silhouette-based methods.

In this chapter, we propose a new model-based performance capture method that takes a leap forward, and captures detailed human performance, including accurate motion and loose non-rigid surface shape, in less controlled and outdoor environments with moving background, *without* explicit silhouette extraction. To meet the challenges of less controlled environments, we use a new unified implicit formulation for both, articulated skeleton tracking and non-rigid surface shape refinement. Our method fits an initial static surface mesh of an actor to unsegmented video frames by jointly optimizing for skeleton pose and non-rigid surface detail. To this end, we introduce a mathematical formulation as the minimization of an objective function that estimates the agreement of model and observation.

The coarse volumetric 3D body shape and body appearance used for skeletal pose estimation, as well as the fine-scale 3D surface geometry and appearance, which is coupled to the coarse body representation, are described in Section 2.3. Particularly for surface shape refinement we introduce a new combination of 3D Gaussians, which we refer to as *Border Gaussians*, designed to align the projected model with likely silhouette contours without explicit segmentation or edge detection. Also the input images are transformed to an implicit representation, similar to the previous chapter, but at

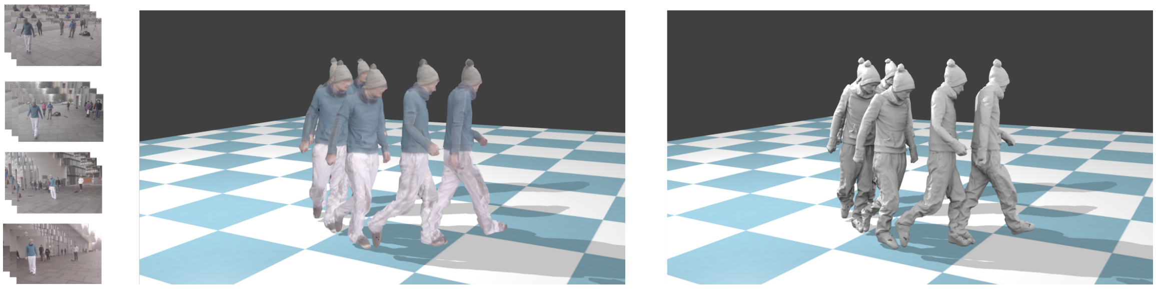


Figure 5.1: From a set of multi-camera input images (left), we reconstruct the human performance as a temporally consistent 3D mesh that accurately matches the captured motion, here visualized textured (center) and untextured (right) to better appreciate the results. Our novel formulation for model-based human performance capture enables reconstruction of outdoor performances without explicit silhouette segmentation.

much higher resolution to model details. This scene representation enables effective and efficient optimization and analytically differentiable smooth objective functions. We introduce a coarse-to-fine objective that integrates coarse fitting and detail refinement, where robust skeleton tracking is used to improve the performance of fine-scale surface detail recovery, both computationally and qualitatively.

Ours is the first integrated template representation and fitting approach for both coarse and fine geometry capable of accurately reconstructing the articulated motion and deforming surface geometry of actors in challenging outdoor scenes. Our main technical contributions are:

1. a unified implicit formulation for both, articulated skeleton tracking and non-rigid surface shape refinement,
2. a formulation of a new surface refinement method to align the projected model to likely silhouette contours without explicit segmentation or edge detection.

We obtain reconstructions of much higher quality in outdoor settings than existing methods, and show that we are on par with state-of-the-art methods on indoor scenes for which they were designed.

## 5.2 Related Work

Existing methods for performance capture involve subsequent optimization of coarse tracking and detailed reconstruction. During coarse tracking, the global motion of the actor is estimated through skeleton or coarse (volumetric) surface fitting. Finer deformation details are recovered in a subsequent step using either stereo constraints, features, or shading-based refinement. Section 3.2 and Section 4.2 describe previous approaches focused respectively on coarse, i. e. skeleton driven, and fine-scale surface tracking. While independent fitting of skeleton and derived detailed surface have proven impressive results especially in indoor controlled scenarios, failures of individual steps often cause temporal tracking mismatches from which it is hard to recover.

Joint optimization of skeleton and shape is often advantageous as it creates structural stability and allows to correct mismatches in skeleton and surface simultaneously [179, 9, 64]. However, optimization with such a high number of independent model parameters is computationally demanding and easily becomes under-constrained. Related works tend to simplify the input model aiming at reducing the number of parameters to optimize for. In the work of Straka et al. [165], the actors shape is constrained by a linear dependency on skeleton bone positions and is optimized efficiently to silhouette observations. Rigidity weights are manually assigned. Kanazawa et al. [92] learn stiffness values for a volumetric deformation energy during fitting to manually obtained correspondence annotations. Most commonly used are linear blend shape models [5] and variants of the SCAPE model [7], both learned from databases of human scans. In particular, SCAPE represent a popular data-driven human model able to synthesize realistic non-rigid surface deformations as a function of skeletal pose and body shape. However these and many other suggested parameterizations [88, 7, 136, 109, 110, 14, 142] are constrained to a low-dimensional set of deformation parameters and not general enough to represent surface details such as exact wrinkles and general apparel deformation.

Many methods demonstrate their capability to recover fine scale surface details by iteratively optimizing over detailed and multi-layered models. Iterative local optimization allows at each iteration to optimize for a lower dimensional subspace of the entire parameter set, which improves the overall performance especially in terms of stability. Global optimization is too expensive for large data sets and skeletons with many degrees of freedom. Gall et al. [64] describe a method to solve for skeleton mismatches through local iterative optimization. Stoll et al. [163] segment the result of [64] into rigid and non-rigid parts to separate near-rigid body parts from clothing exhibiting free deformation.

Despite all recent progress in performance capture, existing model-based methods do not cope well in uncontrolled outdoor scenes, due to background-foreground ambiguities among others. In this chapter, we use a coarse-to-fine strategy where the shape reconstructions increases in detail from iteration to iteration, while inferred constraints such as skeleton dependencies are maintained, but refined by generative matching to the input video without intermediate simplifications.

A key element of this chapter is a method for surface border refinement without silhouettes. It overcomes the limitation of many existing performance capture methods that rely on explicit background segmentation for accurate silhouette alignment, an error-prone step which hinders their usage in uncontrolled environments. Progress has been made by multi-view segmentation [183, 45], joint segmentation and reconstruction [167, 69, 21, 123, 124, 37], and also aided by propagation of a manual initialization [74, 191, 170]. In uncontrolled environments, the obtained segmentation is still noisy, enabling only skeleton pose [74] and rather coarse 3D reconstructions [123, 124]. Rhodin et al. [142] propose a volumetric contour model and directly fit a parametric shape model to image edges, circumventing silhouette extraction entirely. However, only coarse shape without cloth-level detail is reconstructed. Non-rigid shape and silhouettes can be also directly captured with depth cameras, whose richer RGB-D data ease pose optimization and allow even real-time reconstructions with template based [207, 70, 198] and parametric shape models [14, 76, 37]. However, the specialized RGB-D sensors are not as commonly available as RGB video cameras, e. g. in mobile devices, may have higher cost and energy consumption, and do not work well in direct sunlight. In contrast, the proposed segmentation-free method only requires RGB videos as input and is thereby applicable in more general outdoor environments.

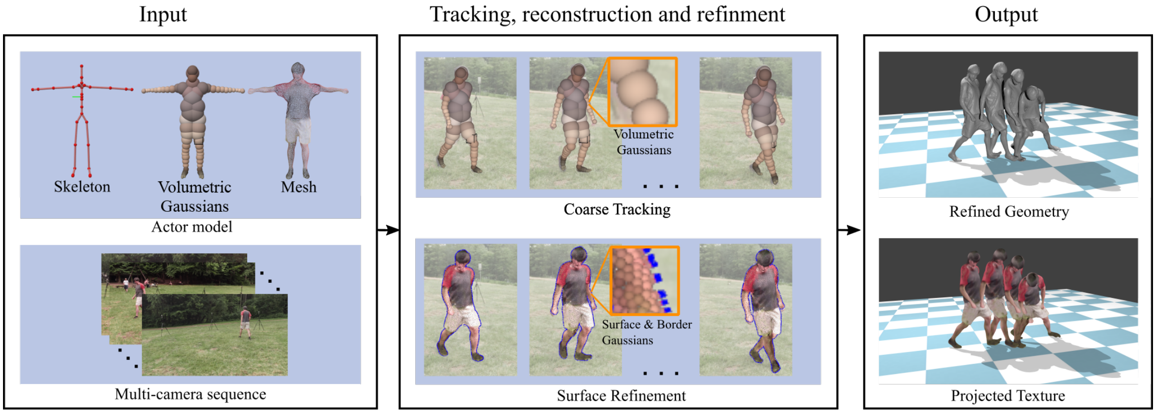


**Figure 5.2:** Overview of our method. Input to our optimization approach is an actor model and a multi-view sequence obtained from synchronized and calibrated cameras (right). We optimize the model-to-image agreement in two stages (depicted in the middle) where we subsequently estimate the skeleton pose and then refine the surface using a new tracking approach. Output of our method (left) is a sequence of refined geometry and texture, which best resembles the input performance in terms of pose and surface details.

## 5.3 Overview

In this section, we describe our model-based performance capture approach, also summarized in Figure 5.2. Given a sequence of multiview calibrated and synchronized images capturing the action of a single actor in an outdoor environment, our goal is to deform a given personalized body model of the actor such that it accurately reproduces the performance filmed. Details on the three-layer human body model used are given in Section 2.2.

The initial skeleton motion of the performance is tracked based on the approach described in Section 2.4.1, which does not require segmentation and attains high performance through a Gaussian representation of image and actor model. Output is an intermediate skeleton motion, which is used to drive the mesh layer by skinning. The resulting mesh sequence only accounts for the rigid motion derived from the skeleton, therefore it fails in reproducing non-rigid deformations caused by cloth and soft tissue deformation, and suffers from skinning artifacts.

In order to recover high-detailed surface deformation, we maximize the agreement between a fine-scale Gaussian-based implicit representation of the surface mesh and the image in a similar way to Chapter 4. The previous chapter described a surface capture approach to densely refine an initial coarse animation using a set of small 3D *Surface Gaussians* on the mesh surface. The capabilities of the method were mostly demonstrated in indoor controlled settings, due to the lack of outdoor reconstructed sequences of sufficient quality. In this chapter, we demonstrate the effectiveness of the Surface Gaussian based approach in less constrained outdoor scenarios. Additionally, to refine the input skinned animation, we introduce a new set of 3D *Border Gaussians* designed to improve the contour alignment in presence of a dynamic background without explicit background segmentation.

The full performance capture approach described in this chapter actively employs the refined surface deformation output to improve the initially reconstructed skeletal pose iteratively. This chapter demonstrates that joined estimation of skeletal and shape deformation is advantageous over independent optimization in terms of robustness. The output of our method is a sequence of skeletal poses alongside with refined surface meshes that reproduce the input videos.

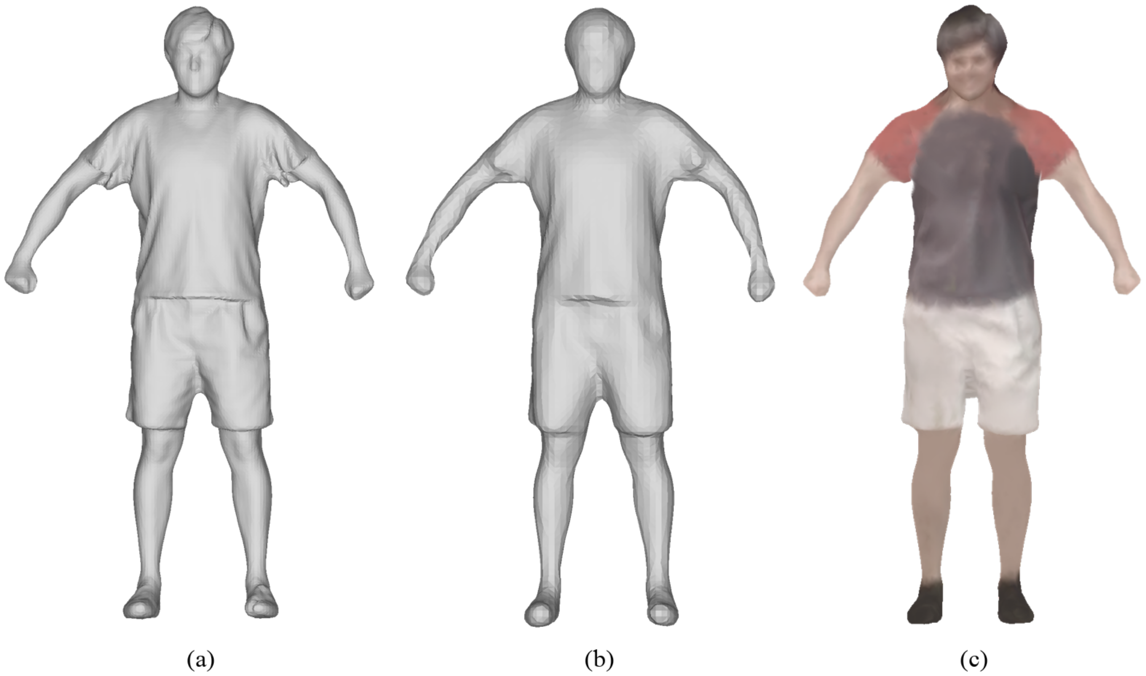


**Figure 5.3:** Actor mesh model creation for the *pablo* sequence. (a) Initial scan of the actor, (b) Poisson reconstruction, (c) resulting mesh after regularization and simplification, (d) resulting mesh with vertex colors.

## 5.4 Scene Model

Similarly to the previous chapter, we convert both the actor model and images to an implicit representation to enable effective model-to-image consistency estimation and optimization.

### 5.4.1 Actor

We use the three-layer representation, described in Section 2.2, to define our personalized actor model. Before converting the mesh surface layer to an implicit mathematical representation, we regularize its geometry such that the vertices are regularly distributed on the surface. We additionally smooth out the high-frequency surface details, see Figure 5.3$(b)$. Smoothing is applied in order to remove most of the baked-in details, which are due to pose-related deformations on the clothing, e. g. flapping sleeves, wrinkles or facial expressions. Notice that by equally smoothing out the entire mesh, we may additionally get rid of important rigid details such as pockets and detailed arm or leg shape. To avoid this, a rigid-details aware semi-automatic surface smoothing approach is a better choice. Our final textured mesh is obtained by reprojecting the multi-view pixel colors from the input images to the manually reposed mesh in the first frame, see Figure 5.3$(c)$, using a similar procedure as explained in Section 2.4.1.4.

We transform the vertices of the geometric layer into an implicit representation $\mathcal{G} = \{G_s^c, G_b^c\}$ consisting of two subsets of Gaussians: *Surface Gaussians* $G_s^c, \forall s = 1 \dots n_s^c$ and *Border Gaussians* $G_b^c, \forall b = 1 \dots n_b^c$. In this chapter, we often use the term *Model Gaussian* referring to the collection of Surface and Border Gaussians. The Surface Gaussians are assigned to the visible vertices from camera $c$ that do not fall onto an occluding mesh contour at a given frame. A small 3D Gaussian is placed at each vertex position and is assigned the color of the static mesh vertex, as also explained in

Chapter 4. Notice that the Surface Gaussians have the same mathematical definition introduced in the previous chapter, however they are now only placed at inner vertices, away from the reprojection borders. For the vertices falling on a contour a different subset of Gaussians is used, with a different mathematical representation, which we call Border Gaussians.

The Border Gaussians are assigned to the vertices that lie on an outer occluding mesh contour in the original camera view $c$, and are explicitly used for contour alignment. Their mathematical formulation consists of a pair of connected Surface Gaussians designed to have opposing forces respectively towards the foreground and background. By minimizing the product of these opposing forces, Border Gaussians are designed to align exactly at the image contours. Details about the design and the formulation for optimization are given in Section 5.5.2.

**Algorithm 5:** Function to initialize a given actor model with Surface and Border Gaussians for each camera view $c$, given as input current mesh $M$, camera projection matrices $P_c, \forall c = 1 \ldots n_c$ and distance threshold $\Delta$ to the closest image borders.

```
function $\mathcal{G}^c := Init(M, P_c, \Delta)$:
for $c := 1$ to $n_c$ do
    for $v := 1$ to $n_v$ do
        $v^c := P^c v$;
        if $v^c$ visible then
            if $dist(v^c, border) \leq \Delta$ then
                $G^c_b.insert(BorderGaussian(v))$;
            else
                $G^c_s.insert(SurfaceGaussian(v))$;
            end
        end
    end
    $\mathcal{G}^c := \{G^c_s, G^c_b\}$;
end
```

We preserve the mesh connectivity information to define the surface topology of Surface and Border Gaussians. Model Gaussians are also coupled to the skeleton by the original mesh skinning weights, in order to formulate skinning-based surface regularization, see Section 5.5.3. The pseudo-code used for initializing the actor model is depicted in Function 5. While these sets of Gaussians are defined per each camera, in the rest of this chapter, we drop the camera index for notation simplification.

### 5.4.2 Images

Our input image set is approximated with a set of colored 2D *Image Gaussians* $G^c_i, \forall i = 1 \ldots n^c_i$. The Image Gaussians are estimated using the method explained in Section 2.4.1.1.

## 5.5 Surface Tracking

The articulated skeleton pose $\mathcal{S}$ is estimated using the approach described in Section 2.4.1, which is shown to recover reliable poses in complicated outdoor recordings. In particular, the initial skeletal configuration in the first frame is estimated manually. Once the skeletal motion has been recovered, the initial actor surface, defined by $\mathcal{M} := \{\mathcal{M}_s, \mathcal{M}_b\}, \mathcal{M}_s := \{\hat{\mu}_s : s = 1 \dots n_s\}, \mathcal{M}_b := \{\hat{\mu}_b : b = 1 \dots n_b\}$, is obtained through skinning, i. e. by rigidly deforming the actor body limbs following the skeleton motion $\mathcal{S}$. In this section, we describe how to incorporate the missing fine-scale non-rigid surface deformations to the coarse geometry. In order to refine the initial skinned actor model, we maximize the following energy:

$$\mathbf{E}(\mathcal{M},\mathcal{S}) = E_{\text{sim}}(\mathcal{M}_s) + E_{\text{cont}}(\mathcal{M}_b) - w_{\text{skin}} E_{\text{skin}}(\mathcal{M},\mathcal{S}) - w_{\text{smooth}} E_{\text{smooth}}(\mathcal{M}) - w_{\text{temp}} E_{\text{temp}}(\mathcal{S}), \quad (5.1)$$

initialized with the estimated pose $\mathcal{S}$ and the associated skinned mesh $\mathcal{M}$. The combination of the proposed energy terms helps in minimizing the discrepancy between the actor model geometry and the multi-view images, while keeping the surface deformations smooth both spatially and temporally. We define each term of $\mathbf{E}(\mathcal{M},\mathcal{S})$ analytically and compute the corresponding derivatives with respect to the unknown surface $\mathcal{M}$ and skeleton pose $\mathcal{S}$. The derivatives are:

$$\frac{\partial \mathbf{E}}{\partial \mathcal{M}_s} = \frac{\partial E_{\text{sim}}}{\partial \mathcal{M}_s} - w_{\text{skin}} \frac{\partial E_{\text{skin}}}{\partial \mathcal{M}_s} - w_{\text{smooth}} \frac{\partial E_{\text{smooth}}}{\partial \mathcal{M}_s} \quad (5.2)$$

$$\frac{\partial \mathbf{E}}{\partial \mathcal{M}_b} = \frac{\partial E_{\text{cont}}}{\partial \mathcal{M}_b} - w_{\text{skin}} \frac{\partial E_{\text{skin}}}{\partial \mathcal{M}_b} - w_{\text{smooth}} \frac{\partial E_{\text{smooth}}}{\partial \mathcal{M}_b} \quad (5.3)$$

$$\frac{\partial \mathbf{E}}{\partial \mathcal{S}} = -w_{\text{skin}} \frac{\partial E_{\text{skin}}}{\partial \mathcal{S}} - w_{\text{temp}} \frac{\partial E_{\text{temp}}}{\partial \mathcal{S}} \quad (5.4)$$

Notice that $\partial \mathcal{M}_s := \partial [\hat{\mu}_s]_{x,y,z}$ and $\partial \mathcal{M}_b := \partial [\hat{\mu}_b]_{x,y,z}$ correspond to derivative w. r. t. the 3D Model Gaussian means, i. e. Surface Gaussians $\hat{\mu}_s, \forall s = 1 \dots n_s$ and Border Gaussians $\hat{\mu}_b, \forall b = 1 \dots n_b$, and $[\cdot]_{x,y,z}$ are the corresponding coordinates on the global axis $x$, $y$ and $z$.

The unified energy representation enables joint and efficient optimization of surface interior, contour alignment, and skeleton pose, using a traditional fast gradient-based optimizer. In the next sections, we describe each term in detail.

### 5.5.1 Similarity Term

The similarity term $E_{\text{sim}}$ measures the photo consistency of the *visible* Surface Gaussians with the input images. It is implemented as a generalization of the volumetric Gaussian tracking, described in Section 2.4.1:

$$E_{\text{sim}}(\mathcal{M}_s) = \frac{1}{n_c} \sum_{c=1}^{n_c} \left[ \frac{1}{n_i^c} \sum_{i=1}^{n_i^c} min \left( \sum_{s=1}^{n_s} \Phi_{i,s}, \ 1 \right) \right] \quad (5.5)$$

where $\Phi_{i,s}$ measures the similarity between the image and the reprojected Surface Gaussians, described in detail in Section 4.5.1. This term accounts for fine-scale detail refinement of the surface interior, especially when texture cues are available. As shown in the previous chapter, this formulation is sub-optimal for contour alignment. The corresponding analytical derivatives are defined as:

$$\frac{\partial E_{\text{sim}}}{\partial \mathcal{M}_s} = \frac{1}{n_c} \sum_{c=1}^{n_c} \frac{1}{n_i^c} \sum_{i=1}^{n_i^c} \begin{cases} \frac{\partial \Phi_{i,s}}{\partial \mathcal{M}_s} & \text{if} \sum_{s=1}^{n_s} \Phi_{i,s} < 1 \\ 0 & \text{otherwise} \end{cases} \quad (5.6)$$

Notice that in contrast to the previous chapter, the Model Gaussians are allowed free displacement in space. Therefore the derivatives have to be computed w. r. t. each global axis $x$, $y$ and $z$. For the full derivation of the derivatives, refer to Section B.0.1.

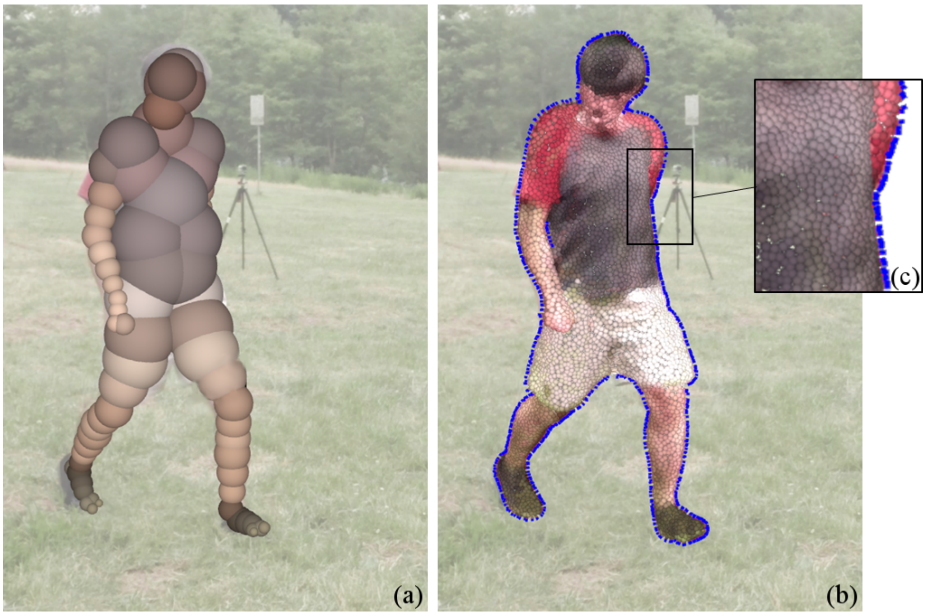


**Figure 5.4:** Visualization of the Gaussian based representation for the actor volume (volumetric layer) and surface in the *pablo* sequence. (a) Volumetric Gaussians, (b) Surface Gaussians on the inner surface locations and Border Gaussians along the image contours, (c) zoom in on a reprojection border to better visualize the Border Gaussians. Notice that only the outer Gaussian in blue is well visible, while the inner Gaussian with the surface color is hidden below the surface.

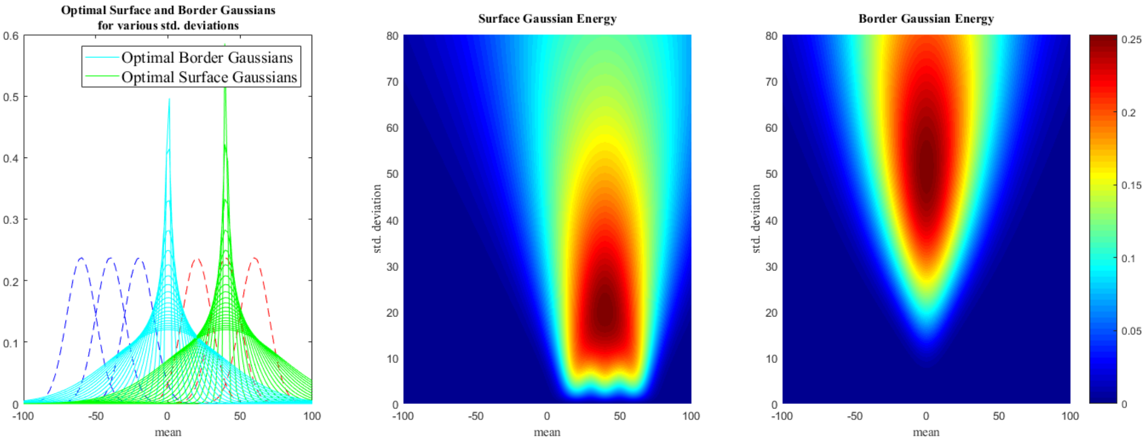
### 5.5.2 Contour Term

The contour term $E_{\text{cont}}$ measures the model-to-image contour alignment. Our goal is to align each border vertex in color, space and direction with nearby image gradients, i. e. move border vertices such that their projection satisfies the following constraints: it spatially coincides with a strong edge, it shows a strong gradient from vertex color to background color, and its edge orientation aligns with the mesh contour direction. In our setting, we have an accurate shape and appearance model of the actor, but face unknown background, e. g. moving scenes, which hinders direct foreground-background gradient computation.

We propose a formulation that neither requires pre-computations nor knowledge of the background color and is nevertheless efficient to optimize. For each mesh vertex $v$ that is within a certain distance to the contour of the mesh in the camera plane, we create an implicit representation, which we refer to as *Border Gaussian*. Border Gaussians $\hat{G}_b$ are centered at vertices closest to the reprojection surface borders and consist of an *inside* and a connected *outside* Gaussian, respectively $\hat{G}_{b_{\text{in}}}$ and $\hat{G}_{b_{\text{out}}}$, meeting at the corresponding vertex location, i. e. $\hat{G}_b := \{\hat{G}_{b_{\text{in}}}, \hat{G}_{b_{\text{out}}}\}$. Both inside and outside Gaussian attributes, i. e. mean $\hat{\mu}_{b_{\text{in}}}$ and $\hat{\mu}_{b_{\text{out}}}$ and standard deviation $\hat{\sigma}_{b_{\text{in}}}$ and $\hat{\sigma}_{b_{\text{out}}}$, are computed from the Border Gaussian attributes, i. e. $\hat{\mu}_b$ and $\hat{\sigma}_b$:

$$\hat{\mu}_{b_{\text{in}}} := \hat{\mu}_b - \hat{\sigma}_b \hat{\mathbf{n}}_b, \quad \hat{\mu}_{b_{\text{out}}} := \hat{\mu}_b + \hat{\sigma}_b \hat{\mathbf{n}}_b, \quad \hat{\sigma}_{b_{\text{in}}} := \hat{\sigma}_{b_{\text{out}}} := \frac{\hat{\sigma}_b}{2} \tag{5.7}$$

where $\hat{\mathbf{n}}_b$ is the corresponding vertex normal, obtained by interpolation the neighboring face normals. Notice that for later optimization, this quantity is kept as a constant. Inside and outside Gaussian are



(a) (b) (c)

**Figure 5.5:** Response of a Surface Gaussian (b) compared to a Border Gaussian (c) in an example scenario. (a) Shows an example distribution of red and blue-colored 1D Gaussians meeting at an image border located in $\mu = 0$. Optimal Surface and Border Gaussians are sampled respectively in green and cyan for different standard deviations. These are extracted from the energies in (b) and (c).

designed to have opposing color similarity. We therefore optimize:

$$E_{\text{cont}}(\mathcal{M}_b) = \frac{1}{n_c}\sum_{c=1}^{n_c}\left[\frac{1}{n_i^{c2}}\sum_{i=1}^{n_i^c} min\left(\sum_{b=1}^{n_b^c}\Phi_{i,b_{\text{in}}},1\right)\cdot min\left(\sum_{b=1}^{n_b^c}\overline{\Phi}_{i,b_{\text{out}}},1\right)\right] \tag{5.8}$$

where $n_b$ is the number of Border Gaussian, and $\Phi_{i,b_{\text{in}}}$ and $\overline{\Phi}_{i,b_{\text{out}}}$ are respectively the similarity between the inside and outside Gaussians in a reprojected Border Gaussian and the Image Gaussian. In particular, $\overline{\Phi}$ takes the opposing color similarity $T_\Delta(\delta_{b,i})$, expressed as $(1 - T_\Delta(\delta_{b,i}))$. This allows evaluating the outside Gaussian contribution to the energy without explicitly defining its color, which is variable w. r. t. view and time depending on the currently hit background at the reprojection surface borders. Instead, the outside Gaussian color is assumed to be sufficiently far from the color of the connected inside Gaussian. Notice that this leaves out possible cases where the foreground and background color match. Nevertheless these extreme cases cannot be handled by state-of-the-art computer vision approaches either. Optimizing for $E_{\text{cont}}$ causes implicit attraction of the inside Gaussian to the model color and the outside Gaussian to the background color. These opposing attraction forces push the Border Gaussian towards the foreground-background boundaries as desired.

Figure 5.5 shows the response of a Border Gaussian compared to a single Surface Gaussian in an example scenario in 1D with an image border. The scenario involves 6 Image Gaussians, 3 blue and 3 red meeting at an image border. While an optimal Surface Gaussian slides away from the image borders during optimization, a Border Gaussian finds its maxima at the desired location.

Additionally, Figure 5.6 visualizes the responses of Surface and Border Gaussians w. r. t. rotation in a similar scenario in 2D. Surface Gaussians are insensitive to rotation as shown in the figure and have only response to similarly colored Image Gaussians. Surface Gaussians can be effectively used to refine inner parts of the surface, as shown in Chapter 4. On the other hand, Border Gaussians, thanks to the pair structure, have responses both to the foreground and to the background and are sensitive to rotations. Maximum response is located at the edge of the Image Gaussians, therefore pulling the Border Gaussian at the image edges with optimal rotation. More examples and properties of the Border Gaussians are discussed in Section B.0.3. The analytical derivatives are defined below. Full

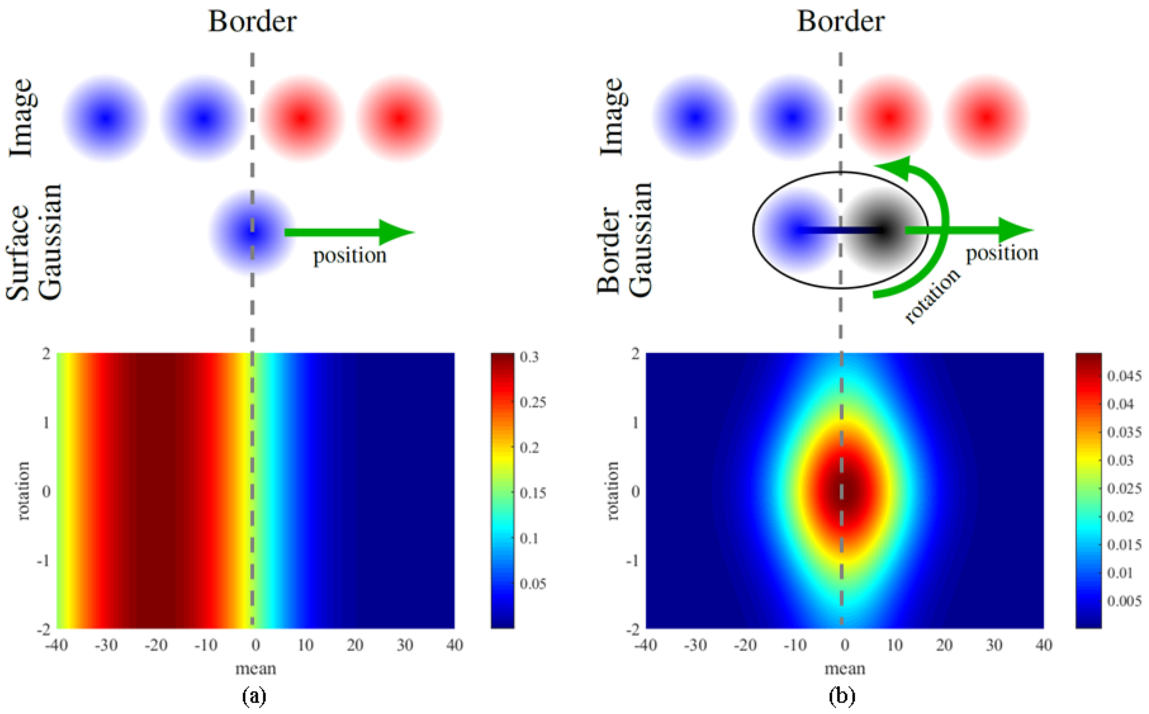


**Figure 5.6:** Response of a Surface Gaussian (a) compared to a Border Gaussian (b) sampled along different positions and orientations in an example scenario. The example scenario is visualized in 2D on top of the responses.

derivation can be found in Section B.0.2.

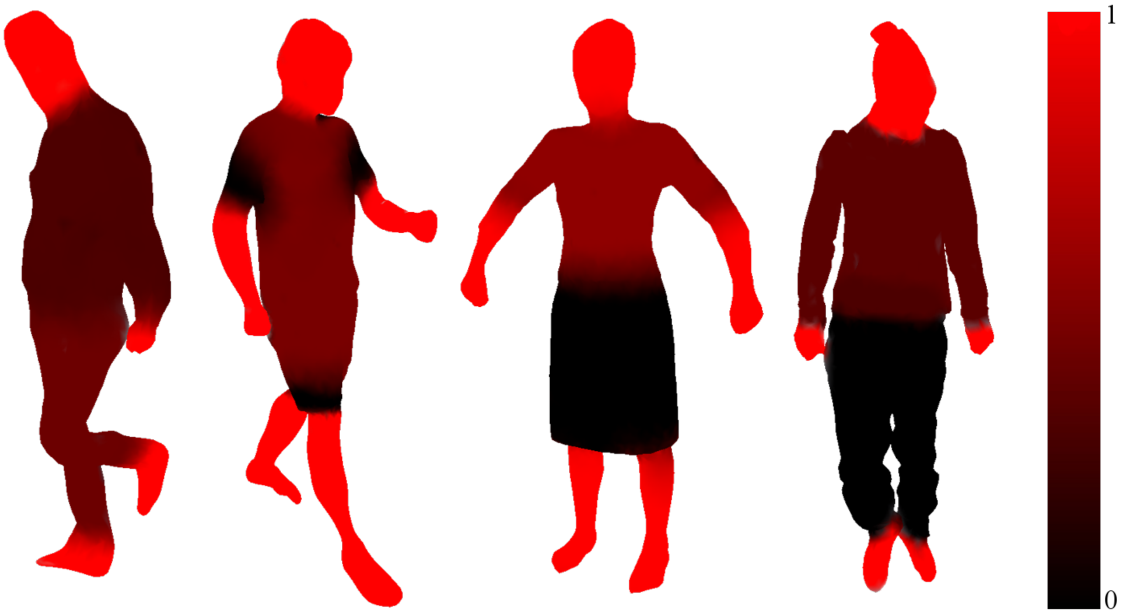
$$\frac{\partial E_{\text{cont}}}{\partial \mathcal{M}_b} = \frac{1}{n_c} \sum_{c=1}^{n_c} \frac{1}{n_i^{c2}} \sum_{i=1}^{n_i^c} \begin{cases} \overline{\Phi}_{i,b_{\text{out}}} \frac{\partial \Phi_{i,b_{\text{in}}}}{\partial \mathcal{M}_b} + \Phi_{i,b_{\text{in}}} \frac{\partial \overline{\Phi}_{i,b_{\text{out}}}}{\partial \mathcal{M}_b} & \text{if } \sum_{b=1}^{n_b} \Phi_{i,b_{\text{in}}} < 1 \text{ and } \sum_{b=1}^{n_b} \overline{\Phi}_{i,b_{\text{out}}} < 1 \\ \Phi_{i,b_{\text{in}}} \frac{\partial \overline{\Phi}_{i,b_{\text{out}}}}{\partial \mathcal{M}_b} & \text{else if } \sum_{b=1}^{n_b} \Phi_{i,b_{\text{in}}} \geq 1 \\ \overline{\Phi}_{i,b_{\text{out}}} \frac{\partial \Phi_{i,b_{\text{in}}}}{\partial \mathcal{M}_b} & \text{else if } \sum_{b=1}^{n_b} \overline{\Phi}_{i,b_{\text{out}}} \geq 1 \\ 0 & \text{otherwise} \end{cases} \tag{5.9}$$

### 5.5.3 Skinning Term

The skinning term $E_{\text{skin}}$ is used to keep the 3D Model Gaussians means 3D, i. e. $\mathcal{M} := \{\mathcal{M}_s, \mathcal{M}_b\}$, close to the correspondent rigid location $\mu_{\check{q}}, \forall q = 1 \ldots n_s + n_b$, which is determined by the estimated skeleton pose $\mathcal{S}$ and the global joint transformations $T'_j$:

$$\hat{\mu}_{\check{q}} = \sum_{j=1}^{n_j} w_{q,j} T'_j \hat{\mu}^j_{\check{q}} \tag{5.10}$$

where $\hat{\mu}^j_{\check{q}}$ is expressed in local coordinates w. r. t. the parent joint $j$. Details on skinning are presented in Section 2.3.2. Given the rigid Gaussian locations $\hat{\mu}_{\check{q}}$, the skinning term $E_{\text{skin}}$ is defined as follows:



(a) (b) (c) (d)

**Figure 5.7:** Rigidity masks for different actor models. Black stands for no rigidity constraint, while red stands for strong rigidity. (a) Actor from the *cathedral* sequence, (b) actor from the *pablo* sequence, (c) actor from the *skirt* sequence, (d) actor from the *unicampus* sequence introduced in Section 5.6.1.

$$E_{\text{skin}} = \frac{1}{n_s + n_b} \sum_{q=1}^{n_s+n_b} \lambda_q \left(1 - \widehat{\Phi}_{q\check{q}}\right)^2 \tag{5.11}$$

where $\lambda_q$ is a rigidity weight associated to the Gaussian $q$ and $\widehat{\Phi}_{q\check{q}}$ is the similarity in 3D of the optimized Model Gaussian $\hat{G}_q$ with its rigid correspondent $\hat{G}_{\check{q}}$:

$$\widehat{\Phi}_{q\check{q}} = T_\Delta(\delta_{q\check{q}}) 2 \frac{\hat{\sigma}_q \hat{\sigma}_{\check{q}}}{\hat{\sigma}_q^2 + \hat{\sigma}_{\check{q}}^2} e^{-\frac{||\hat{\mu}_q - \hat{\mu}_{\check{q}}||^2}{\hat{\sigma}_q^2 + \hat{\sigma}_{\check{q}}^2}} = e^{-\frac{||\hat{\mu}_q - \hat{\mu}_{\check{q}}||^2}{2\hat{\sigma}_q^2}} \tag{5.12}$$

A rigidity weight $\lambda_q \in [0,1]$ is defined for each vertex by providing a mask, which enables to regularize different parts of the body differently, e. g. allow loose clothing to move *more* freely than hands or feet. See examples in Figure 5.7.

As a consequence of jointly optimizing $\mathcal{M}$ and $\mathcal{S}$, the $E_{\text{skin}}$ term also refines the skeleton pose. The corresponding derivatives w. r. t. to the surface are:

$$\frac{\partial E_{\text{skin}}}{\partial \mathcal{M}} = \frac{2}{n_s + n_b} \sum_{q=1}^{n_s+n_b} \lambda_q \left(1 - \widehat{\Phi}_{q\check{q}}\right) \frac{-\partial \widehat{\Phi}_{q\check{q}}}{\partial \mathcal{M}} = \frac{2}{n_s + n_b} \sum_{q=1}^{n_s+n_b} \lambda_q \left(1 - \widehat{\Phi}_{q\check{q}}\right) \widehat{\Phi}_{q\check{q}} \left( \frac{\hat{\mu}_q - \hat{\mu}_{\check{q}}}{\hat{\sigma}_q^2} e_{x,y,z} \right) \tag{5.13}$$

where $e_{x,y,z}$ is a 3-dimensional vector defined depending on the dimensionality, i. e. either $x$, $y$ or $z$, with respect of which the derivative is estimated. In particular, $e_x := (100)$, $e_y := (010)$ and

$e_z := (001)$. The derivatives w. r. t. each skeleton degree of freedom $s_p, \forall p = 1 \dots |\mathcal{S}|$ are:

$$\begin{aligned} \frac{\partial E_{\text{skin}}}{\partial s_p} &= \frac{2}{n_s + n_b} \sum_{q=1}^{n_s+n_b} \lambda_q \left(1 - \widehat{\Phi}_{q\check{q}}\right) \frac{-\partial \widehat{\Phi}_{q\check{q}}}{\partial s_p} \\ &= \frac{2}{n_s + n_b} \sum_{q=1}^{n_s+n_b} \lambda_q \left(1 - \widehat{\Phi}_{q\check{q}}\right) \widehat{\Phi}_{q\check{q}} \left( - \frac{\hat{\mu}_q - \hat{\mu}_{\check{q}}}{\hat{\sigma}_q^2} \sum_{j=1}^{n_j} w_{q,j} \frac{\partial T'_j}{\partial s_p} \hat{\mu}_{\check{q}}^j \right) \end{aligned} \tag{5.14}$$

The joint $j$ is either prismatic, i. e. $T'_j = t'_j$ is a translation, revolute, i. e. $T'_j = R'_j$ is a rotation or a combination of these. For different kinds of joints, we obtain different derivations:

$$\frac{\partial t'_j}{\partial s_p} = a_j, \quad \frac{\partial R'_j}{\partial s_p} = a_j \times (\hat{\mu}_q - c_j) \tag{5.15}$$

The derivative with respect to a translation is the joint axis $a_j \in \mathbb{R}^3$ in global coordinates. For revolute joint we have to take the angular velocity given by the rotation matrix, which is by definition the tangential perperdicular to both the joint global axis and the vector connecting the joint center in global coordinate $c_j$ and the optimized vertex position $\hat{\mu}_q$.

### 5.5.4 Smoothness Term

The surface smoothness term $E_{\text{smooth}}$ regularizes unnatural surface deformations with a smoothness prior term. We use a Laplacian smoothness term which penalizes deviations of the optimized Laplacian coordinates from the Laplacian coordinates of the skinned mesh at current frame. The smoothness term is defined as:

$$E_{\text{smooth}} = \frac{1}{n_s + n_b} \sum_{q=1}^{n_s+n_b} \left( \mathcal{L}_q(\mathcal{M}) - \mathcal{L}_q(\mathcal{M}^{\text{init}}) \right)^2 \tag{5.16}$$

where $\mathcal{L}_q(\mathcal{M})$ is the Laplacian operator applied to the currently optimized mean of the $q$-th Model Gaussian, while $\mathcal{L}_q(\mathcal{M}^{\text{init}})$ is the same operator applied to the *initial* Model Gaussian mean resulting from skinning. The Laplacian operator is mathematically defined by:

$$\mathcal{L} = \begin{pmatrix} \sum_{q \in \xi(1)} g_{1,q} & -g_{1,2} & \cdots & -g_{1,n_s+n_b} \\ -g_{2,1} & \sum_{q \in \xi(2)} g_{2,q} & \cdots & -g_{2,n_s+n_b} \\ \vdots & \vdots & \ddots & \vdots \\ -g_{n_s+n_b,1} & -g_{n_s+n_b,2} & \cdots & \sum_{q \in \xi(n_s+n_b)} g_{n_s+n_b,q} \end{pmatrix} \tag{5.17}$$

where $\xi(q)$ is the set of Model Gaussians neighbor to $q$ and the weights $g_{m,n}$ are defined for each opposite edge $(m,n)$ as:

$$g_{m,n} = \frac{1}{2}(cot \alpha_{m,n} + cot \beta_{m,n}) \tag{5.18}$$

taking the angles $\alpha_{m,n}$ and $\beta_{m,n}$ to the mesh edge $(m,n)$. By applying the Laplacian operator $\mathcal{L}$ to the surface vertices, i. e. multiplying $\mathcal{L}$ with a vertices matrix obtained by stacking the vertex coordinates row by row, one obtains the so called Laplacian coordinates. Note that, in the previous Chapter 4, we constrained vertex motion explicitly to the surface normal direction, suppressing tangential corrections entirely and precluding convergence from large displacements, while the Laplacian regularization based on the Laplacian coordinates only ensures coherent motion of nearby vertices.

The derivative of the smoothness term w. r. t. the surface at any vertex location is defined as:

$$\frac{\partial E_{\text{smooth}}}{\partial \mathcal{M}} = \frac{2}{n_s + n_b} \sum_{q=1}^{n_s+n_b} \left( \mathcal{L}_q(\mathcal{M}) - \mathcal{L}_q(\mathcal{M}^{\text{init}}) \right) \mathcal{L}_q(\mathcal{M}) \tag{5.19}$$

### 5.5.5 Temporal Term

We include a temporal term $E_{\text{temp}}$ in the optimization in order to obtain smooth skeleton movements. The energy to be minimized is given by:

$$E_{\text{temp}} = \frac{1}{|\mathcal{S}|} \sum_{p=1}^{|\mathcal{S}|} \left( 1 - e^{-\frac{\left[\frac{1}{2}\left(s_p^{f-2}+s_p^{f}\right)-s_p^{f-1}\right]^2}{2}} \right) \tag{5.20}$$

The exponential is used to smooth the pose acceleration in previously observed frames as well as to keep the quantity between 0 and 1. Notice that the temporal term is applied from $f > 2$ for the previous pose configurations, i. e. $f-1$ and $f-2$, to be defined. The derivative with respect to the degrees of freedom $s_p, \forall p = 1 \dots |\mathcal{S}|$ is computed as follows:

$$\frac{\partial E_{\text{temp}}}{\partial s_p} = \frac{1}{|\mathcal{S}|} e^{-\frac{\left[\frac{1}{2}\left(s_p^{f-2}+s_p^{f}\right)-s_p^{f-1}\right]^2}{2}} \cdot \left( \frac{1}{4}\left(s_p^{f-2} + s_p^{f}\right) - \frac{1}{2} s_p^{f-1} \right) \tag{5.21}$$

### 5.5.6 Optimization

To solve for the surface $\mathcal{M}$ and the skeleton pose $\mathcal{S}$, we employ a fast conditioned gradient ascent. In general, gradient-based optimization finds locally optimal solutions, and (close to) global optima solutions when accurate surface initialization guesses are provided.

The initial skeleton pose guess is estimated manually, while the initial surface guess is obtained from skinning. In the first iteration, we refine the surface, by keeping the skeleton degrees of freedom fixed. This improves the model-to-image similarity having the initial skeleton pose regularize the vertices location. Once the surface has been refined, we step back to solve for the skeleton pose aiming at fixing eventual model-skeleton agreement issues that appeared after surface refinement, e. g. bones wrongly placed withing the body or even exiting rigid body parts. Notice that through the use of a rigidity mask, we guide the alignment more strictly for arms and legs than, e. g. , clothing.

Given an updated skeleton pose, we may iterate to improve the surface alignment to the images with an improved smoothness term. We typically set up to 3 iterations, where one iteration consists of subsequent surface and skeleton pose optimization. While refining the surface, we separate the optimization of the Surface Gaussians and the Border Gaussians in two subsequent steps, aiming at reducing the number of unknowns, see Function 6. Section 5.6 quantitatively validates this choice, showing improved performances w. r. t. to simultaneous optimization of all Model Gaussians at once. The conditioned gradient ascent optimization steps are explained in Section 4.5.4.

## 5.6 Experimental Results

We qualitatively and quantitatively evaluate our method and compare our results with state-of-the-art human performance capture methods, both indoor and outdoors.

**Algorithm 6:** Function to refine an input actor mesh, given as input the actor's skeleton parameters $\mathcal{S}$ and 3D Model Gaussian means $\mathcal{M}$.

```
function $\mathcal{M}' := Refine(\mathcal{S}, \mathcal{M})$:
$\mathcal{M} := \{\mathcal{M}_s, \mathcal{M}_b\} := \mathcal{M}.update(\mathcal{S})$;
for $n := 1$ to $3$ do
    $\mathcal{M}_s' := Optimize(\mathcal{M}_s)$;
    $\mathcal{M}_b' := Optimize(\mathcal{M}_b)$;
    $\mathcal{M}.update(\mathcal{M}_s', \mathcal{M}_b')$;
    $\mathcal{S}' := Optimize(\mathcal{S})$;
    $\mathcal{M}.update(\mathcal{S}')$;
end
```

| Sequence | *cathedral* | *pablo* | *unicampus* | *skirt* | *synthetic* |
|---|---|---|---|---|---|
| Published by | Kim et al. [99] | Us | Elhayek et al. [50] | Gall et al. [64] | Us |
| Cameras | 8 | | | | 9 |
| Frames | 65 | 200 | 133 | 200 | 1 |
| Frame rate | 40 Hz | | | | n/a |
| Camera type | GoPro Camera [67] | | | PhaseSpace Camera [81] | n/a |
| Resolution | 1920 × 1080 | | 1280 × 720 | 1004×1004 | 1280 × 720 |
| Vertices | 4473 | 9161 | 6155 | 3862 | 160 |

**Table 5.1:** Details for each sequence. This table summarizes the technical details of the sequences used in this chapter.

### 5.6.1 Test Sequences

Table 5.1 presents the details of the sequences used in this chapter. The *cathedral* sequence, published by Kim et al. [99], shows an actor walking down the outdoor stairs of a cathedral. The sequence presents some dynamic variations in the unconstrained background, while the illumination remains constant during the whole action. For this sequence there is no actor scan available. To obtain an actor mesh, we manually segment one frame of the sequence and perform silhouette-based reconstruction to obtain a static reconstruction, which we use as a template, see Section 2.2.1 for details. This method allows obtaining a rough estimate of the actor's surface, which lacks surface details. Despite this, later in this section, we show high-quality reconstruction results of the whole sequence with additional deformations being captured.

The newly captured *pablo* sequence shows alternating short jumps and walking motions performed by a single person on a grass field. While the illumination stays constant in time, individual automatic color adjustment of the GoPro cameras causes the actor to appear differently when switching to different views. Since the videos lack color calibration, the tracking approach has to deal with possible color mismatches across the views. On top to this, the shadows of the surrounding trees sometimes alter the appearance of the actor, introducing additional challenges in the tracking. During the jumping motions the shirt and shorts undergo strong deformations. For this and the remaining sequences, a detailed actor scan is available.

The *unicampus* sequence, published by Elhayek et al. [50], also used as testing sequence in Chapter 3, shows a circular walk outside of large building with a dynamic background, featuring people walking around and cars driving close by. In terms of capturing setup, this sequence is the most challenging due to the unconstrained environmental features that surrounds the action. The main actor partially leaves at times (some of) the camera views and appearance-wise can get confused with other people in the scene.

The *skirt* sequence has been used also in the previous Chapter 4. For this sequence automatically estimated silhouettes are available thanks to the back-screened capturing setup. The performing girl wears highly textured clothing with a highly deforming skirt, which is perfectly suited for Gaussians based optimization. Although the controlled background is less challenging in this sequence, we include it in our evaluation setup to perform both evaluation and comparison to previous methods. For the quantitative evaluation of our technique, we additionally generate a *synthetic* sequence with exact ground-truth. See Section 5.6.5 for more details.

### 5.6.2 Runtime

All results presented in this chapter are computed on a desktop computer with a NVIDIA GeForce Titan X. Using unoptimized code, our method takes about 5-30 minutes on the CPU and 1-4 minutes on the GPU, per frame. The biggest overhead is the refinement step, where the similarity between all Model Gaussians and the Image Gaussians has to be estimated. In particular, the similarity computation of a single Border Gaussian takes twice the time of a Surface Gaussian, since it has to be estimated for both the inside and the outside Gaussian. As demonstrated in Chapter 4, the computational time is directly proportional to the amount of Model Gaussians. Notice that, in contrast to the previous chapter, we now optimize w. r. t. all the 3 global coordinates of each Model Gaussians. Thus, the order of complexity is at least three times worse in the worst case. Depending on the number of gradient descent iterations $n_t$ and the number of vertices $n_v$, the order of complexity is $O(3n_t \sum_{c=1}^{n_c} n_i^c 2n_v)$, when the entire surface is approximated with Border Gaussians only. Compared to the previous chapter, where only single 1-degree Surface Gaussians are considered, the method in this chapter involves six times more operations in the worst case. This explains the increased runtime.

### 5.6.3 Qualitative Results

In Figure 5.1 and Figure 5.8, we qualitatively show reconstruction results on the 4 different sequences, introduced in Section 5.6.1. Our results demonstrate that reconstructed meshes are temporally coherent, do not suffer from temporal noise, and maintain the level of detail of the input template without suffering from unnatural geometric deformation artifacts.

We also show textured models in Figure 5.8, last column, computed by reprojecting the original image frames onto the refined models, which implicitly demonstrate the accuracy of our surface reconstruction. More advanced view-dependent texturing techniques [32] would alleviate some of the remaining ghosting artifacts in the appearance. We use $w_{\text{skin}} = 0.001$ for all the sequences and variable $w_{\text{smooth}} \in [0.001, 0.045]$.

The *cathedral* sequence shows accurate tracking of the cloth deformation along the rigid actor's motion. Due to the lack of baked-in surface details, because of absence of scans for the performing actor, the reconstructed deformations are better appreciated over the textured model. With such accurate geometry alignment to the multi-view images, it is a fairly trivial task to add the missing fine-scale surface details, e. g. , using shading-based refinement approaches. While the overall reconstruction aligns well with the multi-view images input, implausible deformations are still

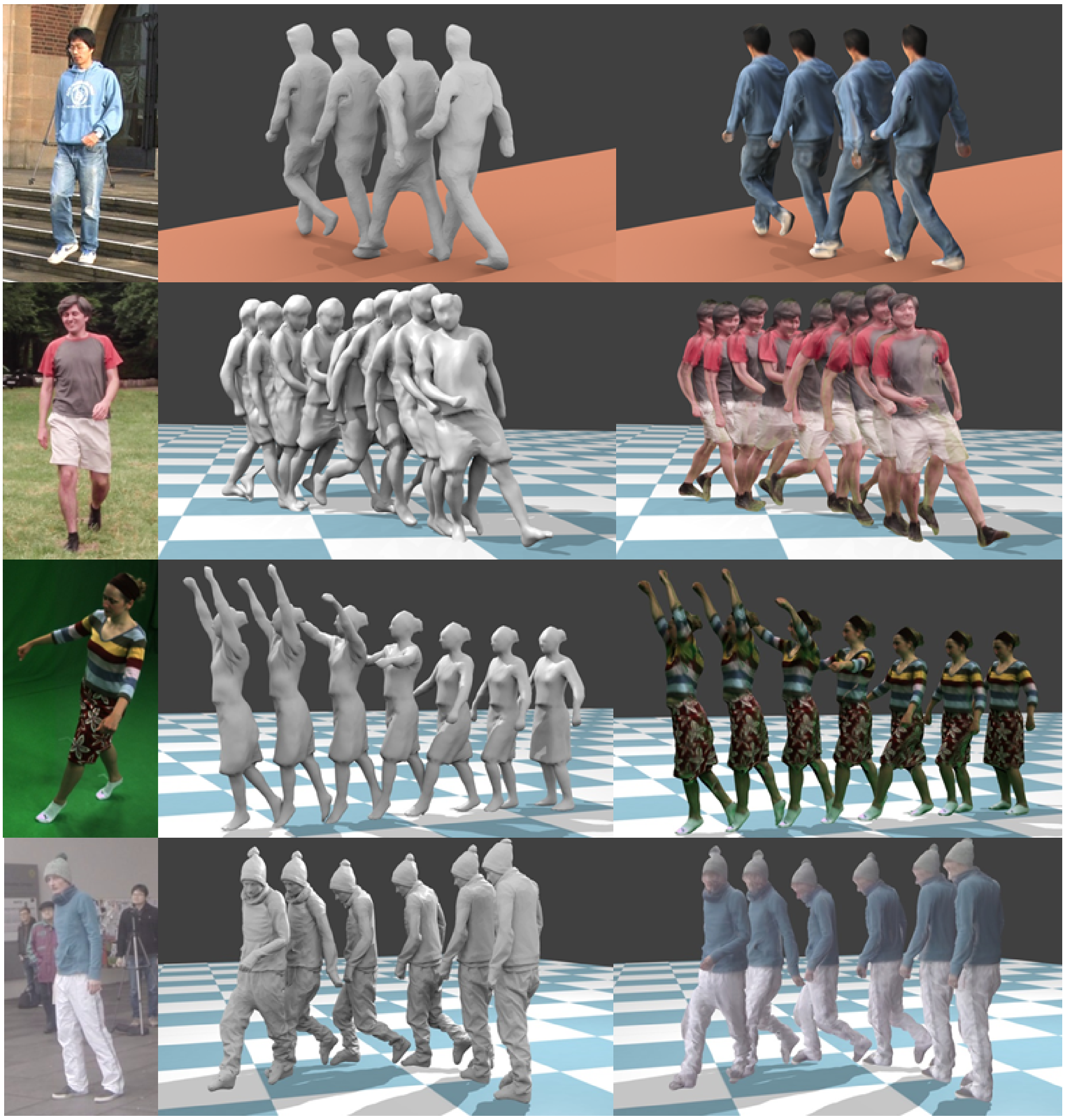


**Figure 5.8:** Qualitative results of our human performance capture approach. On the left, a representative input frame of each sequence. For each sequence, we show untextured and textured reconstructions of various frames. From top to bottom: *cathedral*, *pablo*, *skirt* and *unicampus*.

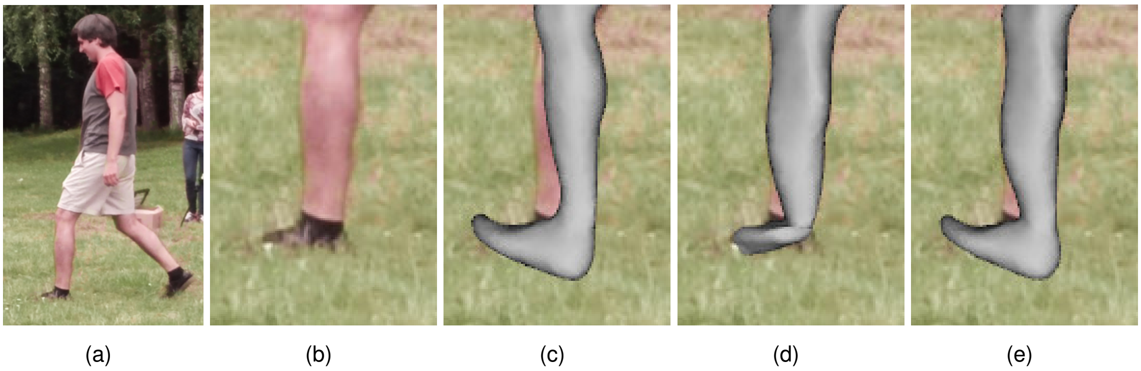


**Figure 5.9:** Influence of the rigidity mask on the skinning $E_{\text{skin}}$ term. (a) Input image, (b) zoom in, (c) initial skinned surface, (d) refined surface without rigidity mask, (e) refined surface with rigidity mask.

visible in the deformation sequence, e. g. in the hair. A higher rigidity weight for the head region would fix these issues.

The *pablo* sequence is particularly challenging to track, due to the appearance changes of the actor across cameras. The reconstructed sequence is apparently unaffected by these inconsistencies, as the found estimated surface is the result of multiple camera fitting, where few uncalibrated views get implicitly ignored (or down-weighted) during optimization. Another challenge in this sequence is the frequent occlusions of the feet in the high grass. In this challenging situations, our refinement step, strongly influenced by the similarity term $E_{\text{sim}}$, tends to implausibly squash the geometry to maximize model-to-observation similarity. When using a rigidity mask, we can enforce vertices on the feet to deform rigidly and maintain its original volume, while improving the contour alignment at the same time, see figure 5.9$(e)$. This proves the effectiveness of our skinning term $E_{\text{skin}}$ to deal with such difficult situations plausibly. In the *pablo* reconstructed sequence, some wobbling in the upper head are visible due to difficulty to distinguish between the hair color and the dark background. Results demonstrate plausible tracking and reconstruction of the shirt and shorts deformation along with the actor's movement.

The *skirt* sequence is one of the least challenging sequences we tested on in this chapter. Monochromatic static background is ideal to refine the mesh contours. Also the strong textures on the clothing allow the proposed approach to track surface details at a finer resolution than the remaining sequences, mostly facing plain-colored apparel. For this sequence we obtain impressively accurate reconstructions including tracking of tiny feet motions and matching flapping of the skirt with the images. Notice that, due to the stronger surface regularization based on Laplacian coordinates, we obtain slightly over-smoothed reconstruction w. r. t. those obtained in Chapter 4, even with small regularization weight, i. e. $w_{\text{smooth}} = 0.005$.

The *unicampus* sequence is one of the most challenges to reconstruct, due to the highly dynamic background. Our results show that our actor model keeps high fidelity matches with the multi-view images regardless of the sudden changes in the background. The actor model of this sequence has the highest resolution compared to the actors in the other sequences and holds most of the initially scanned baked-in details, e. g. wrinkles on the clothing and face features. Our refinement approach keeps surface details rigid while accounting for contour and feature match with the images. While removing baked-in surface details is hard with our formulation, e. g. to capture new forming cloth wrinkles of different shape, we demonstrate that our rigid mask constraints in this sequence help maintaining a plausible-looking shape across the sequence.

### 5.6.4 Comparison

We compare our approach to the silhouette-based method proposed by Gall et al. [64], as well as to the motion capture approach, describe in Section 2.4.1, augmented with a skinned mesh, which corresponds to our initial surface guess. The approach proposed by Gall et al. [64] entirely relies on pre-computed accurate silhouettes and can therefore only be applied to controlled capturing setups. For the comparison we launched all mentioned methods on the indoor *skirt* sequence for which exact silhouettes are available.

To quantitatively assess the performance of our method compared to different approaches, we use a silhouette overlap metric between the actor model projected to the camera plane and the ground truth silhouette obtained with manual segmentation. We label as *positive* and *negative* the foreground and background pixels, respectively. When the label of the projected model and the ground truth image pixel agree, it is a *true* pixel, and *false* otherwise. The combination of true and false pixels is expressed with the $\mathrm{F}_1$ score [152], commonly used in statistical analysis for binary classification, which can be interpreted as a weighted average of *precision* and *recall*, i. e. :

$$\mathrm{F}_1 := 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \tag{5.22}$$

The Figures in this section visualize the resulting overlap labels using green for false negative $FN$, red for false positive $FP$, purple for true positive $TP$ and black for true negative $TN$. Precision and recall are defined respectively as:

$$\text{precision} := \frac{TP}{TP+FP}, \text{recall} := \frac{TP}{TP+FN}. \tag{5.23}$$

Figure 5.10 presents a visualization of the qualitative evaluation of the *skirt* sequence, showing the overlap of the mesh and the ground truth contour ($top$ row), as well as the resulting silhouette overlap labels ($middle$ row), in different settings. Zoomed in visualizations are given in the $bottom$ row. As expected, the mesh generated from skinning after applying the motion capture approach, described in Section 2.4.1 ($first$ column), is incapable of capturing the non-rigid skirt contour and suffers from skinning artifacts in the shoulder area. Our approach (second column) significantly improves these shortcomings, resulting in a much more accurate alignment.

Additionally, we also validate the performance of our method in ideal conditions, where the background is known (*third* column). Instead of working with the input color images, we use the silhouette images, i. e. , with white foreground and black background, and assign the outside Gaussians of the Border Gaussians also a black color, instead of the inverse of the inner Gaussian color as we do in uncontrolled conditions. Results under such ideal conditions, shown in the third column, further confirm the effectiveness of our new implicit representation: *perfect* color assignment of the inner and outer Gaussians refines the mesh such that it perfectly matches the ground truth. Our results are in fact comparable to Gall et al. [64] ($fourth$ column). Notice that this state-of-the-art method requires explicit silhouette segmentation and can be only launched in indoor sequences with controlled capturing setup, whereas our method does not suffer from this restriction.

Figure 5.11 visualizes the $\mathrm{F}_1$ scores across 60 frames and 8 cameras of the *skirt* sequence, for different approaches. Average $\mathrm{F}_1$ score values when enforcing known color background ($0.9676 \pm 0.0056$) are comparable to the silhouette-based method from Gall et al. [64] ($0.9683 \pm 0.0045$).

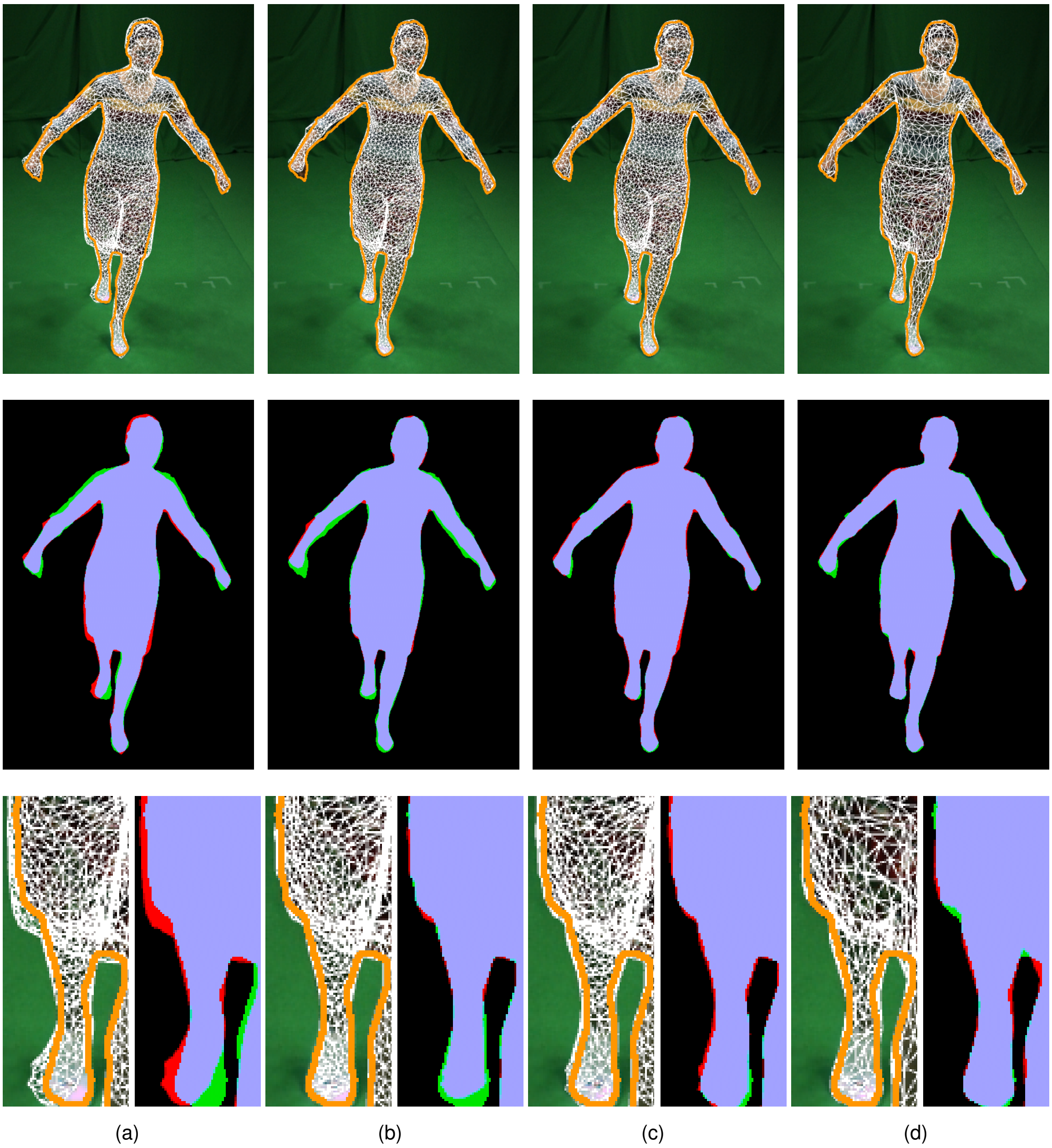


**Figure 5.10:** Silhouette overlap evaluation in the *skirt* sequence. (Top row) Meshes with ground truth contour in orange, (Middle row) Silhouette overlap. See Section 5.6.4 for color scheme description. (Bottom row) Zoomed-in results. (a) Initial skinned surface, (b) refined surface, (c) refined surface with silhouettes, (d) results of Gall et al. [64].

| Sequence | *cathedral* | | *pablo* | | *unicampus* | | *skirt* | |
|---|---|---|---|---|---|---|---|---|
| Actor Model | convex hull | | scan | | | | | |
| Surface | Initial | Refined | Initial | Refined | Initial | Refined | Initial | Refined |
| $F_1$ score | 0.9114±<br>0.0077 | 0.9362±<br>0.0033 | 0.8812±<br>0.0156 | 0.9212±<br>0.0096 | 0.8962±<br>0.0149 | 0.9223±<br>0.0083 | 0.9271±<br>0.0122 | 0.9676±<br>0.0056 |
| Timing | ≈14 | | ≈30 | | ≈20 | | ≈12 | |

**Table 5.2:** Quantitative evaluation of the sequences tested in this paper. Timing is expressed in minutes per frame. The $F_1$ score of the refined surface is consistently higher than for the initial surface resulting from skinning.

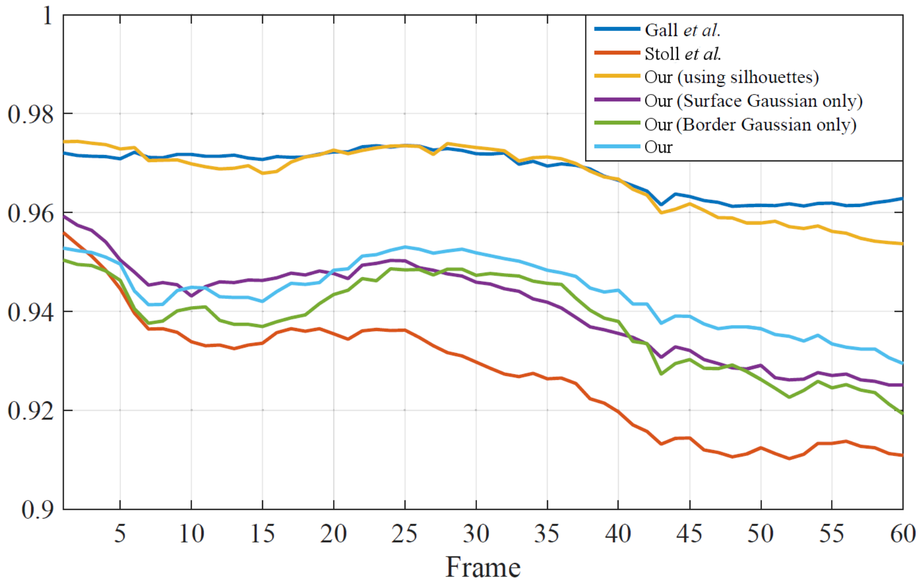


**Figure 5.11:** Quantitative evaluation of the silhouette overlap for the *skirt* sequence. The figure shows the $F_1$ scores ($y$-axis) across frames ($x$-axis) for different settings. Blue: Gall et al. [64], Red: Stoll et al. [164], Yellow: Our approach using explicit silhouettes, Violet: Our approach using Surface Gaussians only (corresponding to Chapter 4 without the normal constraint), Green: Our approach using Border Gaussians only, Cyan: Our complete proposed approach.

### 5.6.5 Quantitative Evaluation

Extensive quantitative evaluation on outdoor footage is difficult due to the lack of ground truth data, which can only be generated with laborious manual segmentation. We manually segment 10 frames of the publicly available *cathedral* dataset [99], as well as of our new sequences *unicampus* and *pablo*. Figure 5.12 shows the silhouette overlap evaluation in these sequences, demonstrating consistent improvement after mesh refinement. Despite the challenging scenes, with uncontrolled background, our method successfully reconstructs and refines the surface of the actor model, without requiring explicit manual silhouette segmentation. Table 5.2 presents the $F_1$ mean and standard deviation of evaluated frames in these sequences, consistently showing that our performance capture method achieves high scores even in such challenging datasets.

In Figure 5.11, we evaluate each of the components of our energy, demonstrating that mesh refinement using Surface Gaussians and Border Gaussians significantly improves over the initial skinning guess, as well as using Surface or Border Gaussian alone.

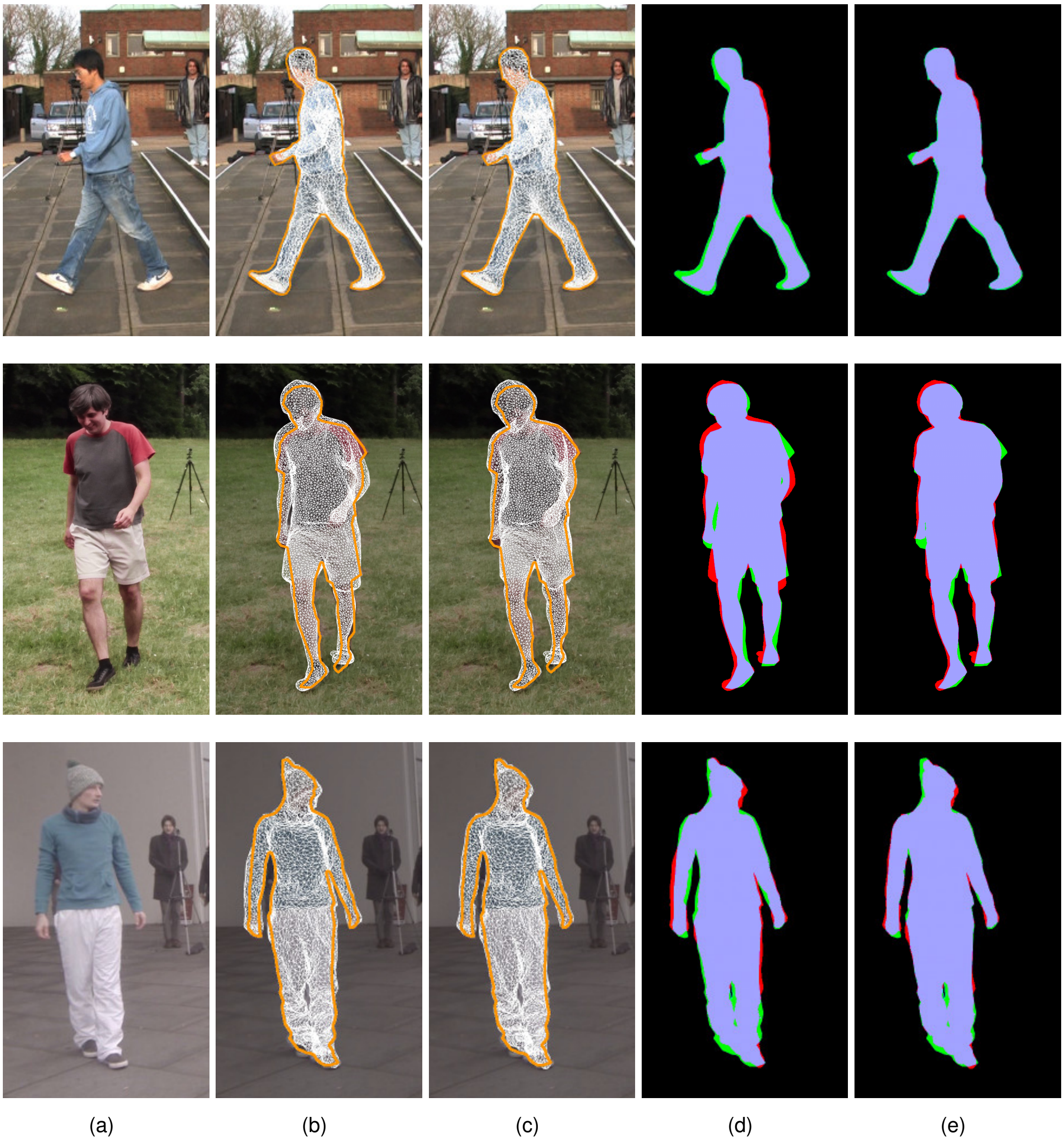


**Figure 5.12:** Silhouette overlap evaluation in outdoor sequences. (a) Original frame, (b) initial skinned mesh and ground truth contour in orange, (c) final refined mesh and and ground truth contour in orange, (d) initial skinned mesh silhouette overlap and (e) final refined silhouette overlap. See Section 5.6.4 for color scheme description.

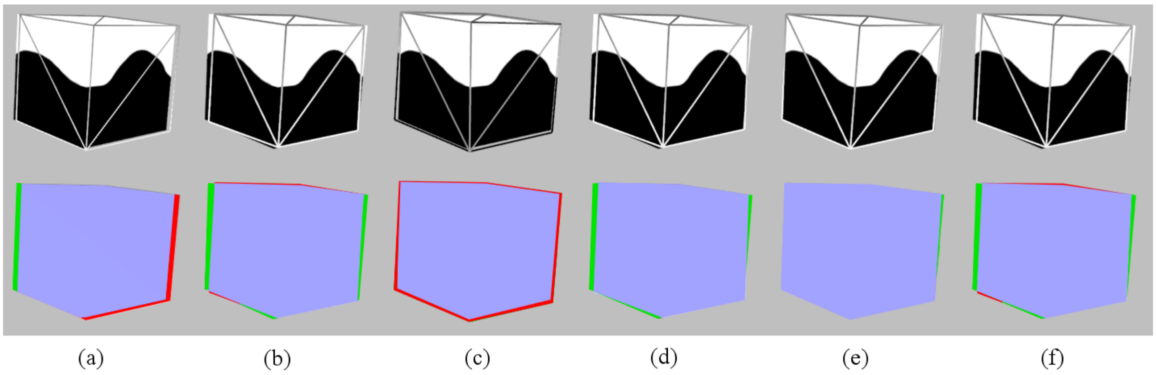


**Figure 5.13:** Results of our approach for the *synthetic* sequence with different configurations. (Top row) Input view featuring a black and white textured cube together with the resulting reprojected wire-frame cube after refinement with different settings: (a) initial state, (b) result optimizing only for the Border Gaussians, (c) result optimizing only for the Surface Gaussians, (d) result obtained with simultaneous optimization for Surface and Border Gaussians, (e) result obtained by optimizing first for the Surface Gaussians only and then for the Border Gaussians only (Our), (f) result as in (d) when using few Surface/Border Gaussians, i. e. at most 8 per views. (Bottom row) Corresponding silhouette overlap. See Section 5.6.4 for color scheme description.

We additionally test our refinement step on a *synthetic* sequence consisting of a 8-vertices textured cube, further subdivided to obtain up to 160 vertices in total, with side size equal to 1 meter, i. e. the volume size equals $1\text{m}^3$. We render 9 random views of the same mesh cube and feed it to our refinement approach together with an initial slightly translated cube model, moved 50 mm along the $x$ axis, as initial surface guess, corresponding to a mean displacement error of 50 mm. Figure 5.13 shows the resulting refined cubes reprojected to a single chosen view using different settings. The initial guess, which is used for refinement, is translated by 50 along the x axis. The corresponding overlap is depicted in Figure $5.13(a)$. Note that for clarity the figure always visualizes the reprojected resulting cubes with 8 vertices, although up to 160 are used for refinement. The best aligned result given by Figure $5.13(e)$, which corresponds to the method described in this chapter, has mean displacement error equal to 13.8 mm with a percentage improvement over the initial guess of 72.3%.

All the remaining test result in less accurate alignment w. r. t. to the input images. The Surface Gaussian only result in Figure $5.13(c)$ has better performances (41.3% improvement, 29.3 mm error) than the Border Gaussians only result in Figure $5.13(b)$ (5.0% improvement, 47.4 mm error). The textures in the inner surface guide the Surface Gaussians toward the correct overlap and implicitly adjust the contours as well, although imperfectly. This finding has been also confirmed in the evaluation of previous Chapter 4.

In the Figure $5.13(d)$, we also test our method when simultaneously optimizing for both Surface and Border Gaussians and find out this leads to worse performances than for our approach, with an average improvement over the initial guess equal to 3.5%, with error 48.2 mm. One reason behind this behavior is that simultaneous optimization has to solve for a larger number of unknowns at the same time, i. e. Surface and Border Gaussians, easily leading to local minima solutions Figure $5.13(e)$. To prove the last assertion, we run simultaneous optimization of Surface and Border Gaussians using only as few as 8 vertices. This clearly leads to inaccurate matches, due to the missing tracking information. However, it has slightly improved performances w. r. t. the test with a larger number of unknowns, i. e. 5.3% improvement and 47.3 mm error.

## 5.7 Discussion and Limitations

Testing demonstrated improved silhouette alignment results when both the inner and outer Gaussian colors in a Border Gaussian are known and fixed, see Section 5.6. In a dynamic background scenario, the color of outer Gaussians is hard to infer as it is subject to sudden changes. In an attempt to identify the actual visible color for each view and frame, an idea could be to use the available information given by the inner surface colors and look for the closest color change, which corresponds to a border cross, along the surface normal, e. g. , using image gradient based techniques. This simple approach may be enough to identify the outer Gaussian color and guide the surface refinement step accurately. Nevertheless, the chosen approach delivers close to optimal silhouette alignment even when surface and background do not present opposing colors.

Especially in outdoor scenarios where the illumination can undergo strong changes, e. g. , global light changes or large harsh shadows hitting the actor, it is important to keep tracking of the appearance changes of the performer and accommodate when needed. The current implicit surface formulation has fixed colors taken from the initial template. However, to account for possible piece-wise appearance changes, the Surface Gaussian colors as well as the inner Gaussians in the Border Gaussians, have to be updated accordingly. Resetting the surface color by multi-view re-projection at each frame easily leads to tracking failures due to the possibly accumulated color mismatches that can lead to increased error over time. Chapter 3 describes a method for robustly creating input illumination-invariant images, which is an effective solution to the appearance changes problem. Another approach to deal with light variations as well as time-varying shading effects happening on the surface could be explicit illumination estimation.

As demonstrated also in the previous chapter, our implicit model formulation enables fine-scale detail reconstruction in presence of highly textured surfaces. Plain colored apparel for instance cannot be tracked down to full detail due to lack of color information in the homogeneous areas. High-frequency geometric detail in these cases could be recovered using inverse rendering techniques, which also consider shading effects. These techniques often require estimating the illumination and might be unsuited for highly shadowed areas, where a shadow edge is easily disguised for a shape detail.

## 5.8 Conclusion

This chapter presented one of the first model-based methods for outdoor human performance capture. Our new unified implicit representation for both skeleton tracking and non-rigid surface refinement allows to jointly optimize for pose and shape, even in scenes with unknown moving background. Our method fits the template to unsegmented video frames by first optimizing for skeletal pose and subsequently refine both the pose and the non-rigid surface shape, such that they match the multi-view images.

During surface refinement, the approach optimizes the silhouette alignment of the reprojected mesh to the multi-view images without explicit background subtraction, by using a new implicit surface border formulation. This new formulation based on Border Gaussians allows formulating silhouette alignment without explicitly defining the vary background color and is proven to deliver improved results even in presence of a unknown dynamic background.

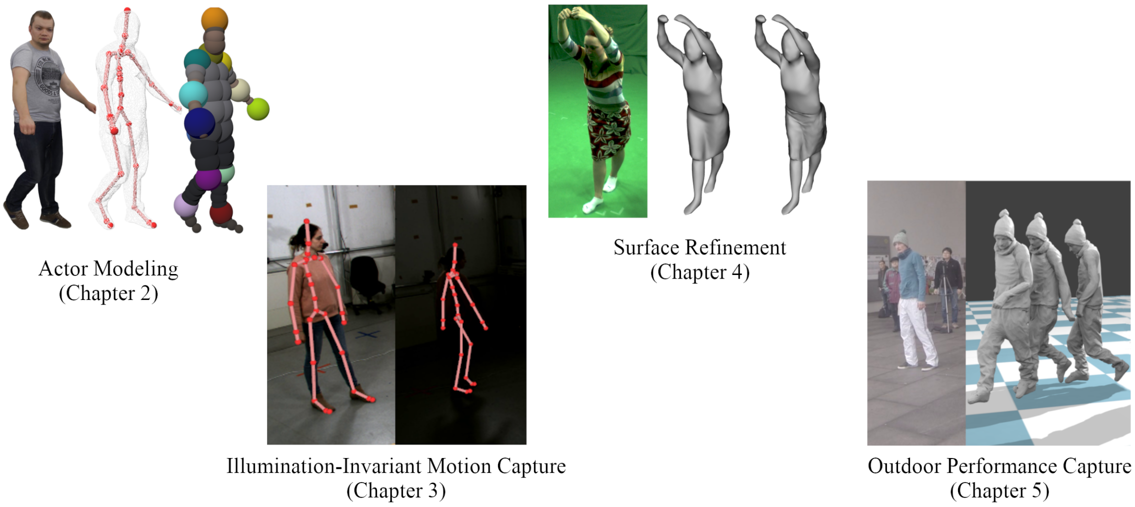
We qualitatively and quantitatively evaluated our performance capture approach on several outdoor sequences, demonstrating the effectiveness of our approach in unconstrained outdoor settings with a possibly dynamic background. We compare our approach with the previous methods and numerically show improved performances.

# Chapter 6

## Conclusion

Cutting-edge advances in technology in the digitization pipeline now allow the entertainment industry to create and animate realistically looking 3D virtual humans with personalized appearance that look indistinguishable from real actors. Technologies for motion and performance capture of real actors have enabled the creation of convincingly looking virtual humans through detail and deformation transfer. Capturing detailed human bodies, however, comes at the price of extensive manual work and sophisticated marker-based systems that are expensive to build and that only work with in-studio controlled settings. Lightweight approaches for motion and performance capture have introduced great simplifications in the acquisition and tracking process through markerless setups. However, outdoor solutions still assume steady weather conditions and fail to reproduce the desired amount of detail in presence of uncontrolled appearance variations of the performing actor, due to illumination changes.

In this thesis, we took a leap forward and built one of the first model-based tracking methods to automatically capture the surface deformation of performing actors in challenging outdoor environment with possibly dynamic background and varying illumination conditions at high accuracy. Novel technical advances have been developed in four different areas, namely actor modeling, motion capture for skeleton configuration estimation, surface capture for personalized detailed deformation estimation and joint multi-layered skeleton-surface capture for robust coarse-to-fine reconstruction. We developed a new multi-layer body model of the actor, composed of a skeleton, volumetric and a geometry layer, that can be easily personalized to represent differently shaped and dressed humans and is well suited for motion and performance capture applications (Section 2.2). The model is further augmented with an implicit Gaussian-based surface representation to favor surface capture in less controlled outdoor settings (Chapter 4 and Chapter 5). Our markerless capture setup consists of a set of synchronized and calibrated vision cameras capturing the overall performance from different viewpoints. We have developed an illumination-invariant motion capture method to reliably estimate skeletal motion in challenging scenarios with illumination changes, such as global light changes and shadows (Chapter 3). For transferring the estimated rigid motion to the geometric layer, we have proposed a fast and effective method to automatically skin the actor's mesh to the underlying skeleton (Section 2.3). Additionally, we have developed a surface capture approach to include additional visible deformation and best match the model to the input multi-view images (Chapter 4). To complete the picture, we have developed unified implicit formulation for both the articulated skeleton tracking and non-rigid surface refinement to jointly track the overall actor's motion from coarse to fine scale resolution (Chapter 5).



Actor Modeling
(Chapter 2)

Illumination-Invariant Motion Capture
(Chapter 3)

Surface Refinement
(Chapter 4)

Outdoor Performance Capture
(Chapter 5)

**Figure 6.1:** Overview of the main contributions presented in this thesis [147, 146, 145, 144].

In this chapter, we conclude the efforts achieved so far by restating and discussing the presented contributions. Additionally, we examine some open challenges towards full body performance capture in the wild.

## 6.1 Summary and Discussion

In this section, we summarize the core contributions presented in Chapters 2 and 5 and provide a brief discussion of some remaining challenges that were not addressed in this thesis.

Chapter 2 introduced the mathematical body model of the human actor used throughout this thesis for motion and performance capture. The multi-layered model consists of a skeleton structure equipped with a volumetric representation of the actor and detailed geometry, which can be easily personalized to differently shaped and clothed human actors. This chapter additionally outlined a novel effective approach for rigging, i. e. defining mesh deformation based on the underlying rigid skeleton motion, which is computationally cheap and generates reliable skinning weights without the need for extensive manual adjustment. The proposed rigging method has been successfully used in several published work [146, 142, 144] to define reliable rigid motion transfer for topologically different actors.

Most of the markerless motion capture approaches are unable to deliver reliable skeleton tracking in presence of strong illumination changes. To overcome this limitation, Chapter 3 introduced a state-of-the art model-based approach that captures accurate skeletal motion by iteratively generating illumination-invariant representations of the input multi-view images. The proposed graph-cut based approach estimates a unique material label for each pixel in the input images, representing its *albedo*. Synthesized albedo pictures are by definition illumination-invariant and can be effectively used to robustly capture the skeletal motion in presence of strong illumination-changes affecting the actor's appearance. Results in unconstrained capture setup demonstrated qualitatively and quantitatively improved skeleton configuration estimations, outperforming related approaches.

Motivated by the lack of general methods for dense surface refinement in (possibly) general environment, Chapter 4 introduced a simple yet robust approach for reliably capturing dense surface details without the need for explicit vertex-to-pixel correspondence [147, 145]. Instead, we built a smooth energy functional, differentiable everywhere in space, that exploits the potentials of the volumetric representation of the actor model to optimally fit it to the multi-view images. A fine-scale Sums-of-Gaussians body model of the actor is matched to similarly approximated 2D Gaussians-based multi-view images using gradient-based optimization. Error-prone correspondence finding is unnecessary in this context, thus enabling better posed model-to-image fitting, possibly in unconstrained background settings.

The potentials of the smooth energy functional for model-to-image fitting in outdoor dynamic environments, is demonstrated in Chapter 5. In this chapter, an improved differentiable energy is introduced to account both for the inner surface details and for the silhouette alignment without the explicit need for background subtraction [146]. The approach is based on a novel smooth formulation for surface reprojection borders that allows silhouette and correspondence-free accurate contour alignment to the multi-view images using gradient based optimization. Additionally, this chapter defined an iterative approach to jointly estimate skeletal motion and refine the surface geometry, enabling robust and accurate solutions to the problem of performance capture. The method proposed in this chapter is one of the first to tackle the performance capture problem in outdoor dynamic scenes.

As stated in Chapter 3, the quality of the input actor model has a direct impact on the accuracy of the results. Because of the lack of fully-automatic methods to estimate personalized geometry and rigging, the proposed approaches require cumbersome manual adjustment and can become infeasible for real-time/interactive applications. While the proposed illumination-invariant segmentation of the input image is robust to different kinds of material and appearance, highly specular materials can easily spoil the performances. The same deficiency applies to the proposed Gaussian-based surface refinement approaches. Nevertheless our smooth volumetric based tracking demonstrated robustness to sporadic unhandled local effects, both in terms of appearance changes and complex deformation.

While Gaussian-based geometry refinement had encountered difficulties in tracking complex cloth folding and tiny details, mainly due to the large number of unknowns, the overall coarse-to-medium scale smooth deformation of the actor closely matches the actual performance, even in outdoor dynamic capture environments. Apart from the geometry resolution, another important factor that affects the quality of the reconstruction is the presence of texture. While typical coarse skeleton tracking prefers plain colored foregrounds, the discovery of fine-scale detail is best performed in presence of highly textured surfaces, where the color distribution produces lots of reliable features to be tracked. When texture are unavailable, shading is another choice for cues that can be used to refine monochromatic pieces of apparel. The reconstruction of fine-scale details covering any kind of surface, i. e. , textured and texture-less, remains an open scientific question, that will be examined in the next section.

## 6.2 Future Work

In this section, we discuss other remaining aspects not covered in this thesis, including challenges in unconstrained full-body performance and motion capture. In this thesis, we have worked with multi-view synchronized and calibrated camera settings. Unconstrained multi-view performance capture, however, has to deal with possibly moving cameras with different technical characteristics and synchronization (Section 6.2.1).

Our proposed model-based performance and motion capture approaches require as input a personalized body model of the actor. The proposed multi-layer actor model is able to semi-automatically adapt to new human shapes. In Section 6.2.2, we outline a method, introduced in a co-authored work by Rhodin et al. [142], that further automatizes the creation of personalized models. Also handling of apparent topology changes, which are due to inaccurate initial actor modeling, is an interesting future direction. More details on related approaches and possible solutions are given in Section 6.2.3.

Our refined tracking solution showed accurate results, especially in presence of strong texture. Remaining monochromatic surface areas can be refined using shading-based refinement. While approaches have been proposed to tackle the problem of understanding geometry from shading, no existing approach is able to synthesize general fine-scale correct detail in unconstrained outdoor scenarios. An overview of the challenges that characterize shading-based methods are outlined in Section 6.2.4.

### 6.2.1 Dealing with Moving Cameras

The past few years have observed significant advances in mobile camera technology. The widespread use of smart-phones facilitated casually capturing and sharing any scenes of interest. Multi-view capture of the same scene, e. g. street performances, has become nowadays more common than ever. Methods for motion and performance capture from sparse mobile videos, have to deal with independent camera movements that lack calibration and synchronization.

Due to the inherent complexities, only few existing works deal with the problem of tracking humans from moving unsynchronized cameras [51, 52, 74, 198, 65]. Hasler et al. [74] perform accurate synchronization using the audio recordings, while camera parameters for each set of pre-synchronized video frames are estimated by performing *Structure from Motion (SfM)* [132]. The actor's pose configurations are estimated in a second step, relying on the estimated camera settings. However, SfM fails in case of cluttered scenes with dense moving background. Another limitation of SfM is its unreliability in presence of motion blur due to e. g. hand-held camera shaking, and small camera translation up to pure rotational motion. Ye et al. [198] proposed a model-based method to simultaneously optimize skeletal pose and sensor position from image features and geometric correspondences tracking between the point clouds and the performer's surface, using multiple consumer depth sensors. Unfortunately, depth sensors have trouble working in outdoor scenarios, when the IR sun rays may introduce interference in the IR-projection pattern. For the method to work reliably, stable background features are also required.

In contrast to these and related approaches, Elhayek et al. [53] proposed a new energy formulation to jointly optimize for the skeleton pose and the camera parameters. Their on-the-fly bundle adjustment method for multi-view calibration is based on few sparse 3D points corresponding to the visible skeleton joints. Resulting camera paths can be less accurate than SfM based approaches. On top of this, their method requires some manual input in the first frame, i. e. initial camera calibration and the actor's poses initialization. Nevertheless, theirs is one of the first method to provide good guesses of camera extrinsics in dynamic and possibly cluttered scenes.

An improved estimation of the camera parameters could be obtained utilizing the larger set of point correspondences across views provided by the denser surface information of the reconstructed geometries, which are refined using the approaches proposed in this thesis in Chapter 4 and Chapter 5. The additional cues could be also used to refine the camera intrinsics along with the extrinsics. Due to the increased amount of parameters to optimize for w. r. t. to [53], specific optimization strategies and temporal regularizers must be explored.

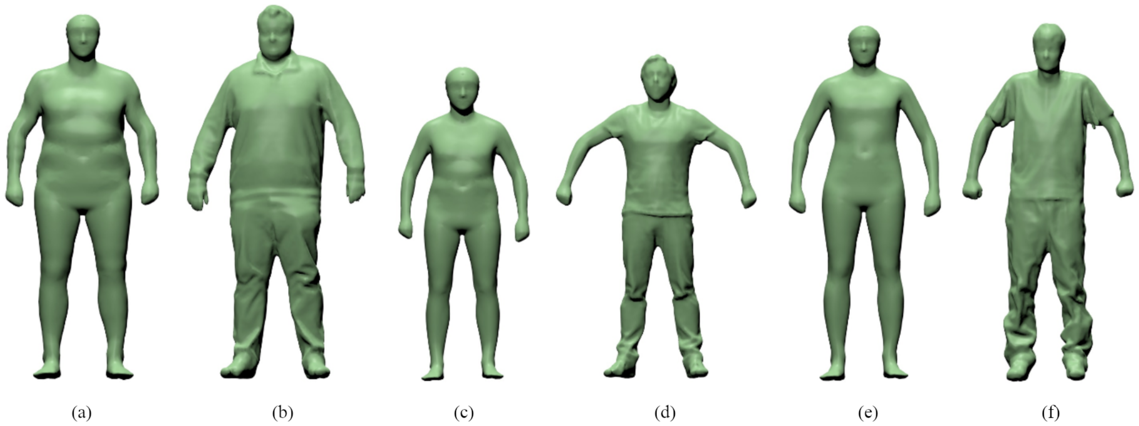


**Figure 6.2:** Parametric body model examples estimated using the method of Rhodin et al. [142] compared to 3D body scans (a,c,e) Estimated parametric body model, (b,d,f) Corresponding 3D scans.

### 6.2.2 Automatic Creation of Personalized Body Models

In this thesis, we introduced a new multi-layer actor body model, which can easily adapt to differently shaped and clothed humans and is well suited for motion and performance capture applications. Among the layers, the geometry layer is the hardest to generate, due to the amount of personalized detail required. High quality geometry is typically achieved through the use of complex 3D scanners or semi-automatic image-based scanning techniques that are strongly affected by imperfect input, see Section 2.2.1. In this thesis, we have seen that most of the backed-in surface detail resulting from scanning are due to the specific body pose (or motion) captured during scanning. In order to obtain a more neutral shape geometry, i. e. independent from specific pose, we typically smooth-out the geometry before tracking. This suggests that the level of detail we are seeking to capture is way smaller than that obtainable from 3D scanners. Semi-automatic image-based scanning techniques applied to few cameras would generally suffice for capturing coarse geometry that is, nevertheless, well suitable for motion and performance capture.

Several methods have been proposed in the literature to reconstruct general high-quality geometry from multi-view photographs [155, 61, 66, 162, 60], see Section 2.2.1.3. Due to the generality of these methods, the reconstructed humans, especially in absence of sufficient camera views, lack of important shape details. To improve the capture of human-like shapes from multi-view, some human-specific methods have been proposed that fit a parametric body model of a human to a multi-view photograph of the actor [159, 110]. In a co-authored work by Rhodin et al. [142], we fully automatically build a personalized 3D model of the actor using a statistical model representing the human body surface in combination with a few multi-view image clues. Our data-driven body model is learned from a database of 228 registered 3D scans of humans in neutral pose. Then, we set one of the scans as a reference mesh, and create a rigged skeleton as well as an implicit volumetric representation of the shape in a similar way as in Section 2.2.3.2. We finally build a PCA model on the variation space given by bone length and volume dimensions and determine a few major PCA coefficients to span the principal shape variations of the database.

To obtain an initial skeleton-based parametric shape of the actor seen in the images, we find optimal PCA parameters that best fit the parametric model to a given estimation of the person's global location, complete with the most prominent joint position guesses obtained using CNN-based approach, in a few frames. To further account for non-rigid shape variations, e. g. determined by fat and body

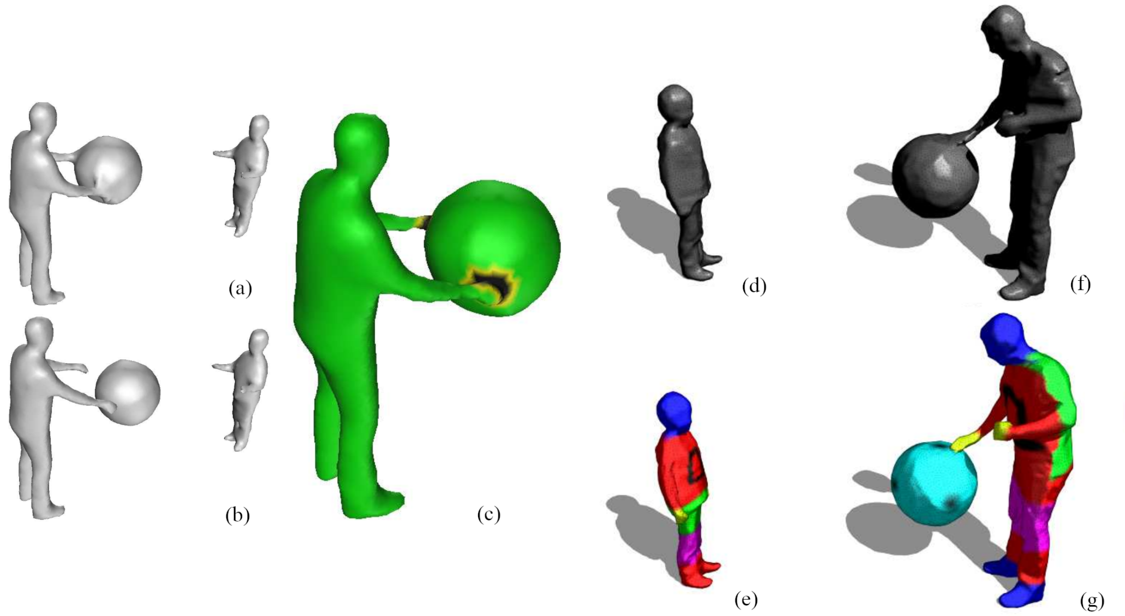


**Figure 6.3:** Example results of related approaches synthesizing apparent topology changes [104, 27]. (a,b) Input 3D reconstructions of two subjects playing with a large ball, (c) Canonical time-consistent actor showing topological separation of the actor's hands from the ball, (d,f) Input model views, (e,g) Corresponding time-consistent canonical model reconstructions with automatic segmented body parts.

mass index (*BMI*), we additionally refine the estimated shape using multi-view image contour cues. Our smooth energy formulation for silhouette-free contour alignment matches reprojected model border gradients with the closest image gradients, which implicitly resemble likely contours. This final refinement step incorporates additional shape features, which may deviate from the database and therefore best adapt to the actor's actual proportions.

Figure 6.2 shows examples of this method's body approximations. As shown in the figure, the estimated parametric model closely aligns to the corresponding 3D scans in terms of body sizes and general shape. A disadvantage of data-driven human body approximations is mainly the lack of some surface features, e. g. clothing deformations and facial detail, which cannot be captured faithfully unless present in the database. As discussed before, optimal personalized human body models must be free of pose specific deformations and these human approximations mostly respect this constraint. A disadvantage of this approach is the inability to capture less-tightly dressed human, e. g. , wearing dresses or skirts, or any other additional add-on, e. g. hats. In fact, this approach synthesizes tightly clothed human body models despite the large variation in clothing, e. g. , correctly locating the hips in presence of a large skirt. A combination of data-driven and image-based method could help integrating unseen body shapes due to clothing. Nevertheless, automatic creation of personalized actor models is an interesting research direction for performance capture.

### 6.2.3 Capturing Apparent Topology Changes

Our proposed model-based approaches require a colored template mesh for initialization and cannot cope well with complex shape or topology changes. Topology changes (in physics) are the changes in the geometric structure of objects due to splits or merges, e. g. water particles. Apparent topological changes are defined as the apparent change of the initially assumed topology due to inconsistencies to the reality. The actor model's geometry is a complex multi-layered structure of interacting materials, e. g. skin and clothing, that are typically assumed to be merged together in a unique surface. In these cases, apparent topology changes occurring when a hidden body part is suddenly revealed, e. g. a hat is taken off, cause strong model-to-image mismatches that cannot be accounted for, unless the topology of the initial actor model is also updated, i. e. the hat is split from the head geometry.

Most methods for performance capture simply ignore apparent topology changes and treat these sudden changes as outliers [40, 64, 20, 179]. Other approaches aimed at reconstructing the full performance, remove the need for an input actor model and reconstruct geometry frame-by-frame including apparent changes in the topology [46, 130, 162, 35]. The main drawback for these techniques is the lack of temporal consistency. On top of this, the reconstructed geometry lacks important details due to the underconstrained settings. Additionally, because of the need for trackable features and background subtraction, model-free approaches have only been successful in indoor controlled studios, often with complicated, expensive multi-view setups.

Since synthesizing realistic multi-layered personalized human models, complete with clothing geometry and physical interactions, is extremely challenging with the current technology, methods to capture apparent topology changes have focused on algorithms to consistently and accurately update the input coarse human model. Advanced techniques explicitly synthesize apparent topology changes on the given input geometry, and then propagate the changes back in time, aiming at keeping the geometry temporally consistent [104, 200, 12, 38, 135], see examples in Figure 6.3. This kind of approaches are ideal for performance capture applications. However, they have only been demonstrated on controlled settings with low resolution geometry.

Implicit surfaces and hybrid implicit-explicit representations are well suited to handle topology changes, thanks to the fact they naturally adapt to typical topological operations like split and merge [188, 131, 43]. Alternatively volumetric particle, voxel or patch based representations can serve well for this purpose [112, 168, 27, 28]. Macklin et al. [112] demonstrate the use of particles with fixed size to represent any kind of object and matter, i. e. solid, liquid or gas. Actor models can be formed by joining together solid particles and model splitting and merging at the particles connections. However, recovering fine-scale detail as well as representing thin material surfaces, e. g. pieces of apparel, is challenging in a low resolution setting, and computationally demanding otherwise. This also applies for voxel or other primitive -based volumetric representation. Nevertheless, this is an interesting future direction.

A solution to the problem of capturing apparent topology changes could utilize the proposed multi-layer body model, synthesized using a collection of Gaussians on the surface, to track the possible variations in topology using the model-to-image fitting techniques discussed in Chapter 4 and Chapter 5. The surface Gaussians described in Chapter 5 are structurally linked by the underlying triangle connections in the geometry using a surface Laplacian regularizer. To enable stronger deformation possibly caused by apparent topology changes, a new less-constraining regularization term must be formulated, that keeps the surface smooth while allowing typical topological operations, such as cuts, fusion and hole formations. Sums-of-Gaussians based models simulate implicit surfaces, which are by definition topologically-free and thus optimal for tracking topology changes. Possibly, new Gaussians on the surface must be seeded and properly constrained to the neighbors to explain newly appearing areas. However, meshes extracted from implicit representations must be remapped to a common topology for spatio-temporal consistency. Large topology changes could be propagated back in time to favor temporal consistency as well as used to refine the previously estimated deformations. As discussed in the thesis, Gaussians on the surface are primarily fit to similarly colored areas. Therefore, topology changes are captured mostly when they result in a

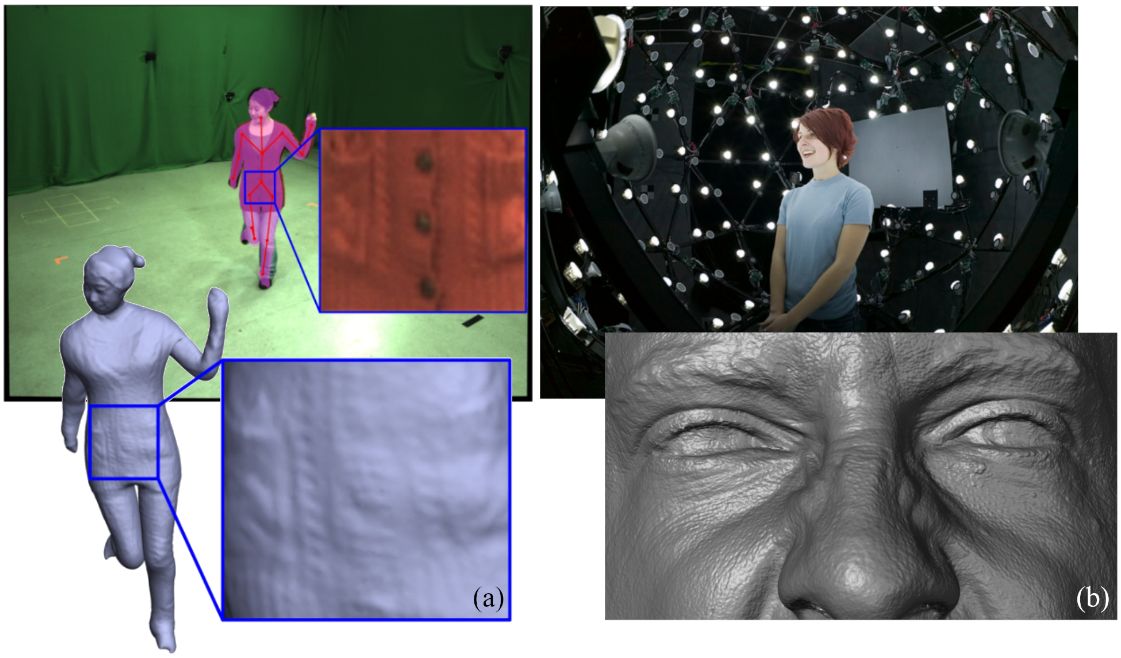


**Figure 6.4:** Results of shading based refinement for different methods. (a) Less-constrained indoor solution by Wu et al. [193], (b) Complex light-stage based reconstruction result by Debevec et al. [42].

visible appearance, i. e. color, change. Nevertheless, the use of Gaussians on the surface for capturing apparent changes in topology is an interesting direction for future research.

### 6.2.4 Shading-based Refinement

Most feature based approaches, like multi-view stereo (*MVS*), reconstruct coarse-to-medium over-smoothed models mainly due to lack of trackable image structures, multi-view averaging into a canonical model or strong regularization [126, 86, 155]. *Shape-from-shading (SfS)* based approaches overcome some of these resolution limits and also succeed on texture-less objects. For SfS to deliver correct results, reliable information on surface reflectance (albedo) of the object of interest and illumination must be provided.

SfS is extensively used in the literature to refine coarse image-based shape models, e. g. obtained from MVS, even under general uncontrolled lighting [192, 193, 191, 11]. In these works, illumination and albedo distributions are estimated via inverse rendering optimizations, a cumbersome and ill-posed step in uncontrolled scenarios. In an attempt to simplify computationally expensive non-linear inverse rendering for scene illumination understanding, strong scene and lighting assumptions are made, e. g. prior knowledge on reflectance, geometry and well-controlled complex lighting setups [42, 78, 125]. Many methods employ *spherical harmonics* to approximate the lighting model and assume distant, monochromatic light sources. Surfaces are typically assumed to be Lambertian, ignoring specularities. Handling of self-occlusions, spatially-varying illumination and high-frequency textures are few examples of additional unhandled cases in SfS refinement, that restrict the applicability of the these kind of techniques.

One of the primary challenges in general scene understanding is distinghishing between albedo and shading variations. Abrupt changes in the albedo, e. g. due to texture, might erroneously hallucinate geometric detail as a result of misinterpretation of local albedo changes as variations in surface shading. Texture-copy artifacts are very common inaccuracies in shading based surface reconstruction. Despite the impressive advances made in this direction, shading based refinement in the wild remains an open problem.

In Chapter 3, we proposed a method that simultaneously tracks both surface deformation and local illumination changes. After initially assigning an albedo color to the various materials, the method learns the appearance change of each material using temporal pose priors and color consistency cues. The accumulated information on different body parts could be used to approximate a reflectance model for each material to better study light interactions from different view points and find the correct albedo to use for SfS. The formulation of surface refinement through SfS could be combined with the actual surface tracking to derive better regularized deformation.

## 6.3 Closing Remarks

The work presented in this thesis has been motivated by current limitations in performance capture, which involve skeleton and fine-scale surface tracking of real actors for 3D realistic virtual human synthesis. Restrictive capturing setup and extensive manual work greatly reduce the applicability of these techniques. Several scientific contributions, which advance the state of the art in multi-view performance capture, have been proposed in Chapters 2–5 to deal with the limitations mentioned above.

Results attained on challenging scenarios have confirmed the scientific advances in the field and have shown potential to automatize the digitization process for realistic virtual characters. Animation artists can now better utilize the results obtained by our algorithms as high-quality prototypes to sketch realistic human animations thereby streamlining the entire conventional digitization process in post-production, thus saving money and weeks of strenuous effort. There are still many challenges that need to be solved to digitize characters in unconstrained settings at high fidelity. We hope that this thesis motivates the development of more follow-up methods to digitize photo-realistic virtual human models of a quality comparable to the standard pipeline in post-production.

# Appendices

# Appendix A

# Capturing Surface Details

### A.0.1 Derivative of the Similarity Term

In order to calculate the derivative of $E_{sim}$, we note that most of its terms are constant with respect to $k_s$, except the projected means $\mu_s$ and the variances $\sigma_s$, within the term $\Phi_{i,s}$.

Using homogeneous coordinates, expressed throughout the paper using the superindex $h$, we first compute the Surface Gaussian mean in 2D image space, $\mu_s^h$, by projecting the constrained Surface Gaussian mean $\hat{\mu}_s^h$ from Equation 4.3, using the camera projection matrix $P \in \mathbb{R}^{4\times 4}$:

$$\mu_s^h = P\hat{\mu}_s^h = P(\hat{\mu}_s^{init} + \hat{\mathbf{n}}_s^h k_s) \in \mathbb{R}^3, \tag{A.1}$$

where $\hat{\mu}_s^{init}$ is the initial Surface Gaussian mean, initialized as the vertex position $v_s$, in homogeneous coordinates. The derivative of $\mu_s^h$ with respect to $k_s$ is defined as:

$$\begin{aligned} \frac{\partial \mu_s^h}{\partial k_s} &= \frac{\partial}{\partial k_s}(P(\hat{\mu}_s^{init} + \hat{\mathbf{n}}_s^h k_s)) = P\frac{\partial}{\partial k_s}(\hat{\mu}_s^{init} + \hat{\mathbf{n}}_s^h k_s) \\ &= P(0 + \hat{\mathbf{n}}_s^h \frac{\partial}{\partial k_s}(k_s)) = P\hat{\mathbf{n}}_s^h. \end{aligned} \tag{A.2}$$

Combining the above equations, the derivative of $\mu_s$ evaluates to:

$$\begin{aligned} \frac{\partial \mu_s}{\partial k_s} &= \begin{pmatrix} \frac{\partial}{\partial k_s}\left(\frac{[\mu_s^h]_x}{[\mu_s^h]_z}\right) \\ \frac{\partial}{\partial k_s}\left(\frac{[\mu_s^h]_y}{[\mu_s^h]_z}\right) \end{pmatrix} = \begin{pmatrix} \frac{\frac{\partial[\mu_s^h]_x}{\partial k_s}[\mu_s^h]_z - [\mu_s^h]_x \frac{\partial[\mu_s^h]_z}{\partial k_s}}{{[\mu_s^h]_z}^2} \\ \frac{\frac{\partial[\mu_s^h]_y}{\partial k_s}[\mu_s^h]_z - [\mu_s^h]_y \frac{\partial[\mu_s^h]_z}{\partial k_s}}{{[\mu_s^h]_z}^2} \end{pmatrix} \\ &= \begin{pmatrix} \frac{\partial}{\partial k_s}\left([\mu_s^h]_x\right) - [\mu_s]_x \frac{\partial}{\partial k_s}\left([\mu_s^h]_z\right) \\ \frac{\partial}{\partial k_s}\left([\mu_s^h]_y\right) - [\mu_s]_y \frac{\partial}{\partial k_s}\left([\mu_s^h]_z\right) \end{pmatrix} \frac{1}{[\mu_s^h]_z} \\ &= \begin{pmatrix} [P\hat{\mathbf{n}}_s^h]_x - [\mu_s]_x [P\hat{\mathbf{n}}_s^h]_z \\ [P\hat{\mathbf{n}}_s^h]_y - [\mu_s]_y [P\hat{\mathbf{n}}_s^h]_z \end{pmatrix} \frac{1}{[P(\hat{\mu}_s^{init} + \hat{\mathbf{n}}_s^h k_s)]_z}. \end{aligned} \tag{A.3}$$

The derivative with respect to $k_s$ of the projected variance $\sigma_s$ is calculated by applying simple derivation rules:

$$\begin{aligned}\frac{\partial \sigma_s}{\partial k_s} &= \frac{\partial \sigma_s}{\partial [\mu_s^h]_z}\frac{\partial [\mu_s^h]_z}{\partial k_s} = \frac{-f\hat{\sigma}_s}{([\mu_s^h]_z)^2}\frac{\partial [\mu_s^h]_z}{\partial k_s} \\ &= \frac{-\sigma_s}{[\mu_s^h]_z}\frac{\partial [\mu_s^h]_z}{\partial k_s} = \frac{-\sigma_s}{[\mu_s^h]_z}[P\hat{\mathbf{n}}_s^h]_z \in \mathbb{R}.\end{aligned} \tag{A.4}$$

Therefore, the derivative of the term $\Phi_{i,s}$ with respect to $k_s$ is obtained by substituting Equation A.3 and Equation A.4 in Equation 4.7, which generates:

$$\begin{aligned}\frac{\partial}{\partial k_s}(\Phi_{i,s}) &= T_{\Delta_c}(\delta_{i,s})2\frac{\partial}{\partial k_s}\left(\frac{\sigma_s\sigma_i}{\sigma_s^2+\sigma_i^2}e^{-\frac{||\mu_i-\mu_s||^2}{\sigma_s^2+\sigma_i^2}}\right) \\ &= T_{\Delta_c}(\delta_{i,s})2\left\{2\frac{\sigma_s\sigma_i}{\sigma_s^2+\sigma_i^2}e^{-\frac{||\mu_i-\mu_s||^2}{\sigma_s^2+\sigma_i^2}}\left[\frac{\partial[\mu_s^h]_z}{\partial k_s}\left(-\frac{1}{2}+\right.\right.\right. \\ &\left.\left.\left.+\frac{\sigma_s^2}{\sigma_s^2+\sigma_i^2}-\frac{||\mu_i-\mu_s||^2\sigma_s^2}{(\sigma_s^2+\sigma_i^2)^2}\right)\frac{1}{[\mu_s^h]_z}+\frac{(\mu_i-\mu_s)\frac{\partial\mu_s}{\partial k_s}}{\sigma_s^2+\sigma_i^2}\right]\right\} \\ &= T_{\Delta_c}(\delta_{i,s})4\frac{\sigma_s\sigma_i}{\sigma_s^2+\sigma_i^2}e^{-\frac{||\mu_i-\mu_s||^2}{\sigma_s^2+\sigma_i^2}}\left[[P\hat{\mathbf{n}}_s^h]_z\left(-\frac{1}{2}+\right.\right. \\ &\left.\left.+\frac{\sigma_s^2}{\sigma_s^2+\sigma_i^2}-\frac{||\mu_i-\mu_s||^2\sigma_s^2}{(\sigma_s^2+\sigma_i^2)^2}\right)\frac{1}{[\mu_s^h]_z}+\frac{(\mu_i-\mu_s)\frac{\partial\mu_s}{\partial k_s}}{\sigma_s^2+\sigma_i^2}\right].\end{aligned} \tag{A.5}$$

Finally, the derivative of $E_{sim}$ with respect to $k_s$ is:

$$\frac{\partial E_{sim}}{\partial k_s} = \frac{1}{n_c}\sum_{c=1}^{n_c}\frac{1}{n_i^c}\sum_{i=1}^{n_i^c}\begin{cases}\frac{\partial\Phi_{i,s}}{\partial k_s} & \text{if}\sum_{s=1}^{n_s}\Phi_{i,s}<1 \\ 0 & \text{otherwise.}\end{cases} \tag{A.6}$$

### A.0.2 Derivative of the Regularization Term

The derivative of $E_{reg}$ with respect to $k_s$ is calculated by simple derivation rules as follows:

$$\begin{aligned}\frac{\partial E_{reg}}{\partial k_s} &= \frac{\partial}{\partial k_s}\left(\sum_{s=1}^{n_s}\frac{1}{|\Psi(s)|}\sum_{j\in\Psi(s)}T_{\Delta_d}(\delta_{s,j})(k_s-k_j)^2\right) \\ &= \frac{1}{|\Psi(s)|}\sum_{j\in\Psi(s)}T_{\Delta_d}(\delta_{sj})\left(\frac{\partial\,(k_s-k_j)^2}{\partial k_s}+\frac{\partial\,(k_j-k_s)^2}{\partial k_s}\right) \\ &= \frac{1}{|\Psi(s)|}\sum_{j\in\Psi(s)}T_{\Delta_d}(\delta_{s,j})\,(2\,(k_s-k_j)-2\,(k_j-k_s)) \\ &= \frac{4}{|\Psi(s)|}\sum_{j\in\Psi(s)}T_{\Delta_d}(\delta_{s,j})\,(k_s-k_j)\,.\end{aligned} \tag{A.7}$$

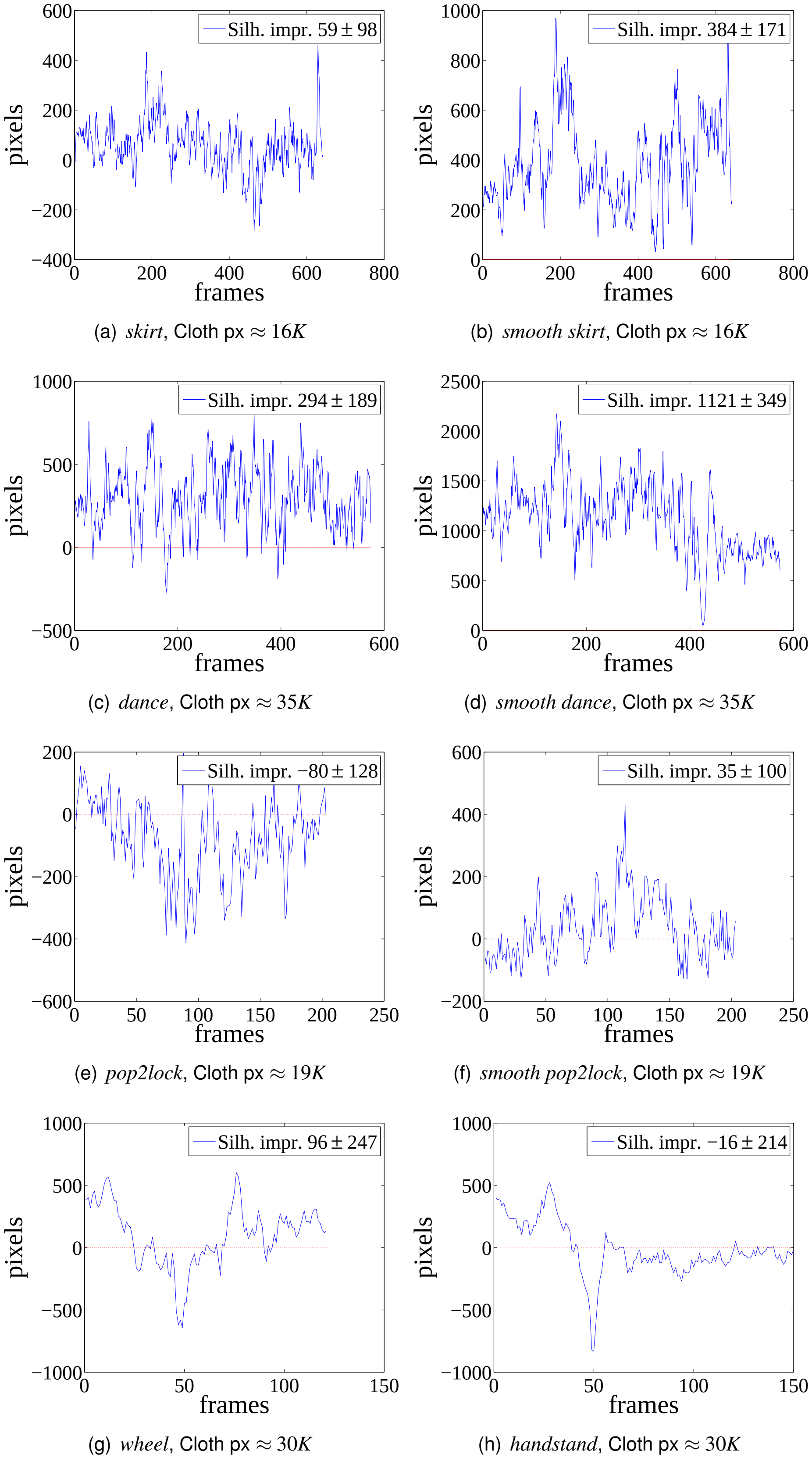
### A.0.3 Derivative of the Temporal Term

The derivative of $E_{temp}$ with respect to $k_s^f$ at the current frame $f$ is calculated by simple derivation rules as follows:

$$\begin{aligned}\frac{\partial E_{temp}}{\partial k_s} &= 2\left(\frac{1}{2}(k_s^{f-2}+k_s^f)-k_s^{f-1}\right)\left(\frac{1}{2}(0+1)-0\right)\\ &= \frac{1}{2}(k_s^{f-2}+k_s^f)-k_s^{f-1}.\end{aligned} \tag{A.8}$$

### A.0.4 Comparative Graphs

In this Section, we show all comparative graphs for the tested sequences.



(a) *skirt*, Cloth px $\approx 16K$

(b) *smooth skirt*, Cloth px $\approx 16K$

(c) *dance*, Cloth px $\approx 35K$

(d) *smooth dance*, Cloth px $\approx 35K$

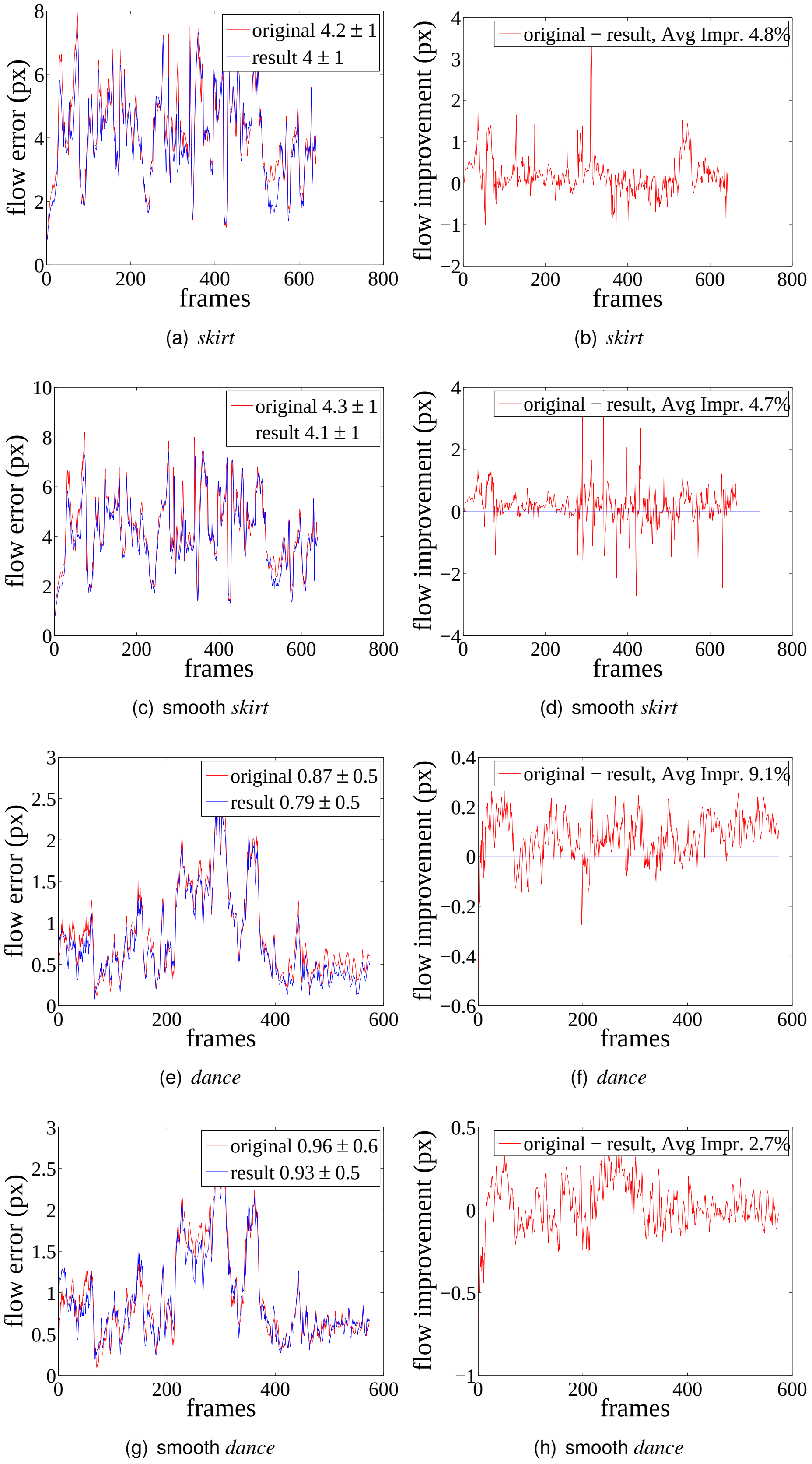
(e) *pop2lock*, Cloth px $\approx 19K$

(f) *smooth pop2lock*, Cloth px $\approx 19K$

(g) *wheel*, Cloth px $\approx 30K$

(h) *handstand*, Cloth px $\approx 30K$

**Figure A.1:** Silhouette improvement of the refined meshes for all the tested sequences. In each graph: pixel improvement with respect to the specified input sequence per frame. Manually estimated true cloth pixels to be used as a reference for the silhouette pixel improvement are stated in the captions of each sequence.



(a) *skirt*

(b) *skirt*

(c) smooth *skirt*

(d) smooth *skirt*

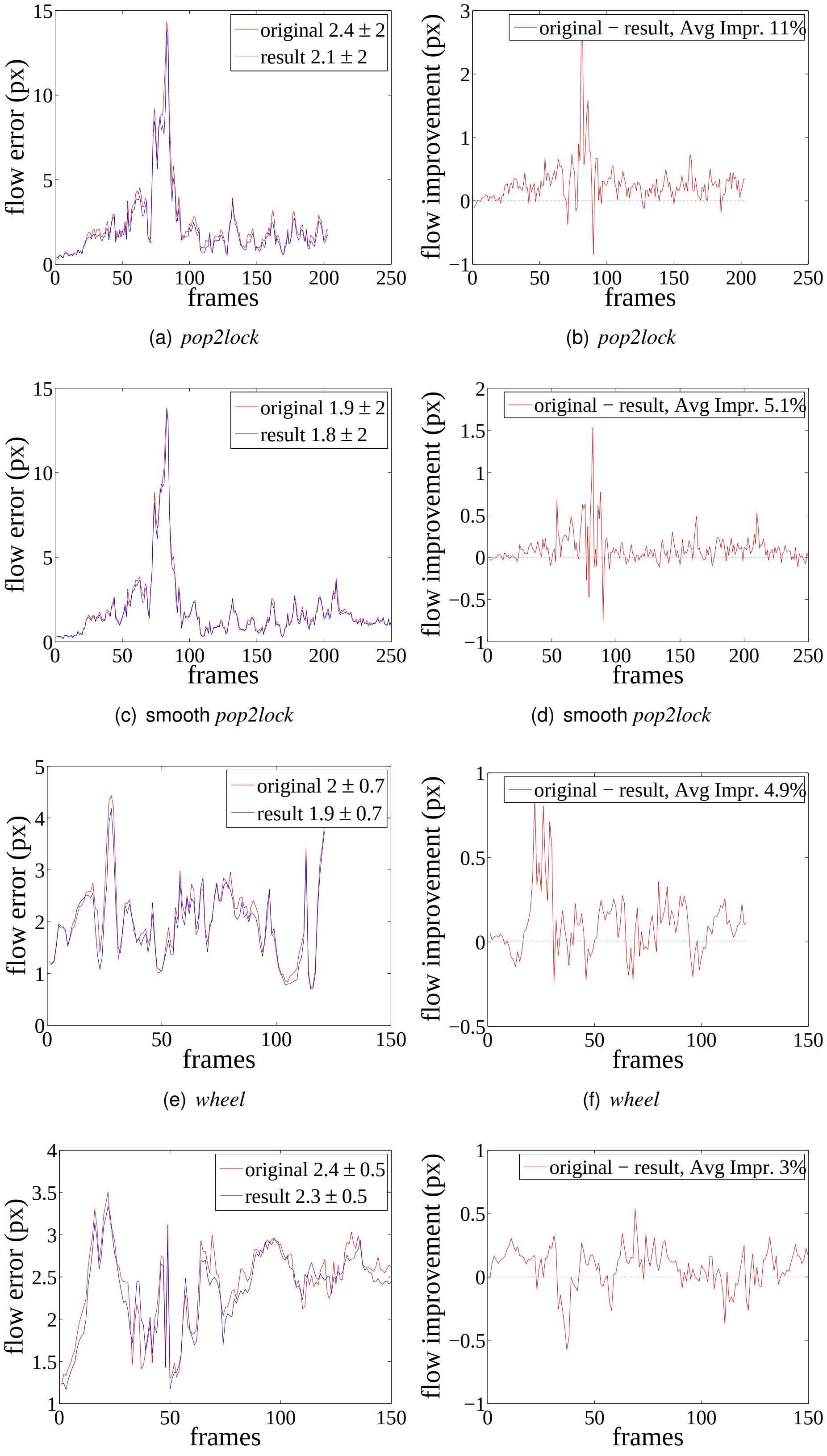
(e) *dance*

(f) *dance*

(g) smooth *dance*

(h) smooth *dance*

**Figure A.2:** Average flow displacement error of the refined meshes, for the *skirt* and *dance* datasets, including over-smoothed version. In each subfigure: (Left) normalized flow error per frame for original, in red, and refined, in blue, mesh sequences. (Right) Difference between input and resulting flow error. Notice how the resulting error is consistently decreased across all sequences. We register about $2.7\%$ to $11\%$ average improvement of the sequence quality with respect to the input flow error. See the text for detailed evaluation.



(a) *pop2lock*

(b) *pop2lock*

(c) smooth *pop2lock*

(d) smooth *pop2lock*

(e) *wheel*

(f) *wheel*

(g) *handstand*

(h) *handstand*

**Figure A.3:** Average flow displacement error of the refined meshes, for the *pop2lock*, *wheel* and *handstand* datasets, including over-smoothed version. In each subfigure: (Left) normalized flow error per frame for original, in red, and refined, in blue, mesh sequences. (Right) Difference between input and resulting flow error. Notice how the resulting error is consistently decreased across all sequences. We register about $2.7\%$ to $11\%$ average improvement of the sequence quality with respect to the input flow error. See the text for detailed evaluation.

# Appendix B

# Outdoor Performance Capture

### B.0.1 Derivative of the Similarity Term

The derivation of the similarity term $E_{sim}$ follows similar derivation rules explained in Section A.0.1. The derivatives of the projected surface Gaussian mean $\mu_s^p = P\hat{\mu}_s^h$ with respect to all dimensions are:

$$\begin{aligned} \frac{\partial \mu_s^p}{\partial [\hat{\mu}_s]_{x,y,z}} = \frac{\partial (P\hat{\mu}_s^h)}{\partial [\hat{\mu}_s]_{x,y,z}} &= \frac{\partial}{\partial [\hat{\mu}_s]_x} \left( \begin{pmatrix} p_{0,0} & p_{0,1} & p_{0,2} & p_{0,3} \\ p_{1,0} & p_{1,1} & p_{1,2} & p_{1,3} \\ p_{2,0} & p_{2,1} & p_{2,2} & p_{2,3} \\ p_{3,0} & p_{3,1} & p_{3,2} & p_{3,3} \end{pmatrix} \begin{pmatrix} [\hat{\mu}_s]_x \\ [\hat{\mu}_s]_y \\ [\hat{\mu}_s]_z \\ 1 \end{pmatrix} \right) \\ &= \frac{\partial}{\partial [\hat{\mu}_s]_x} \left( \begin{pmatrix} p_{0,0}[\hat{\mu}_s]_x + p_{0,1}[\hat{\mu}_s]_y + p_{0,2}[\hat{\mu}_s]_z + p_{0,3} \\ p_{1,0}[\hat{\mu}_s]_x + p_{1,1}[\hat{\mu}_s]_y + p_{1,2}[\hat{\mu}_s]_z + p_{1,3} \\ p_{2,0}[\hat{\mu}_s]_x + p_{2,1}[\hat{\mu}_s]_y + p_{2,2}[\hat{\mu}_s]_z + p_{2,3} \\ p_{3,0} + p_{3,1} + p_{3,2} + p_{3,3} \end{pmatrix} \right) = \begin{pmatrix} p_{0,0} \\ p_{1,0} \\ p_{2,0} \\ 0 \end{pmatrix} = [P]_{x,y,z}, \end{aligned} \tag{B.1}$$

where $P \in \mathbb{R}^{4\times4}$ is the camera projection matrix, $\hat{\mu}_s^h \in \mathbb{R}^4$ is the Gaussian mean in homogeneous coordinates (i.e. the $4th$ additional dimension is set to 1) and $[P]_{x,y,z}$ are respectively the first, second and third column of the projection matrix. The mean in 2D $\mu_s$ is obtained from the projected mean $\mu_s^p$ by performing a dimensionality reduction:

$$\mu_s = \begin{pmatrix} \frac{[\mu_s^h]_x}{[\mu_s^h]_z} \\ \\ \frac{[\mu_s^h]_y}{[\mu_s^h]_z} \end{pmatrix} \in \mathbb{R}^2. \tag{B.2}$$

Therefore its derivative with respect to x, y and z-dimension is obtained as follows:

$$\begin{aligned}\frac{\partial \mu_s}{\partial [\hat{\mu}_s]_x} &= \begin{pmatrix} p_{0,0} - [\mu_s]_x p_{2,0} \\ \\ p_{1,0} - [\mu_s]_y p_{2,0} \end{pmatrix} \frac{1}{[\mu_s^p]_z}, \\ \frac{\partial \mu_s}{\partial [\hat{\mu}_s]_y} &= \begin{pmatrix} p_{0,1} - [\mu_s]_x p_{2,1} \\ \\ p_{1,1} - [\mu_s]_y p_{2,1} \end{pmatrix} \frac{1}{[\mu_s^p]_z}, \\ \frac{\partial \mu_s}{\partial [\hat{\mu}_s]_z} &= \begin{pmatrix} p_{0,2} - [\mu_s]_x p_{2,2} \\ \\ p_{1,2} - [\mu_s]_y p_{2,2} \end{pmatrix} \frac{1}{[\mu_s^p]_z},\end{aligned} \tag{B.3}$$

where $p_{\cdot,\cdot}$ are the matrix components of the projection matrix $P$. The projected standard deviation in 2D $\sigma_s$ is obtained from the correspondent standard deviation in 3D $\hat{\sigma}_s$ as:

$$\sigma_s = \frac{f\hat{\sigma}_s}{\left[\mu_s^p\right]_z} \in \mathbb{R}, \tag{B.4}$$

where $f$ is the camera focal length. The derivatives with respect to $[\hat{\mu}_s]_x$, $[\hat{\mu}_s]_y$ and $[\hat{\mu}_s]_z$ are then defined as:

$$\begin{aligned}\frac{\partial \sigma_s}{\partial\left[\hat{\mu}_s\right]_x} &= \frac{\partial \sigma_s}{\partial [\hat{\mu}_s^p]_z}\frac{\partial\left[\mu_s^p\right]_z}{\partial [\hat{\mu}_s]_x} = \frac{-f\hat{\sigma}_s}{(\left[\mu_s^p\right]_z)^2}\frac{\partial\left[\mu_s^p\right]_z}{\partial [\hat{\mu}_s]_x} = \frac{-\sigma_s}{\left[\mu_s^p\right]_z}\frac{\partial\left[\mu_s^p\right]_z}{\partial [\hat{\mu}_s]_x} = \frac{-\sigma_s}{\left[\mu_s^p\right]_z}p_{2,0} \in \mathbb{R}, \\ \frac{\partial \sigma_s}{\partial\left[\hat{\mu}_s\right]_y} &= \frac{-\sigma_s}{\left[\mu_s^p\right]_z}p_{2,1} \in \mathbb{R}, \\ \frac{\partial \sigma_s}{\partial\left[\hat{\mu}_s\right]_z} &= \frac{-\sigma_s}{\left[\mu_s^p\right]_z}p_{2,2} \in \mathbb{R}.\end{aligned} \tag{B.5}$$

Next, we find all sub-term required for the derivation of $\Phi_{i,s}$ (in $\mathbb{R}$) with respect to $[\hat{\mu}_s]_x$ only:

$$\frac{\partial}{\partial\left[\hat{\mu}_s\right]_x}(\sigma_i\sigma_s) = \frac{\partial}{\partial \sigma_s}(\sigma_s^2\sigma_i^2)\frac{\partial \sigma_s}{\partial\left[\hat{\mu}_s\right]_x} = \sigma_i\frac{\partial \sigma_s}{\partial\left[\hat{\mu}_s\right]_x} = \sigma_i\frac{-\sigma_s}{\left[\mu_s^p\right]_z}p_{2,0} = \frac{-\sigma_i\sigma_s}{\left[\mu_s^p\right]_z}p_{2,0}, \tag{B.6}$$

$$\frac{\partial}{\partial\left[\hat{\mu}_s\right]_x}(\sigma_i^2+\sigma_s^2) = \frac{\partial}{\partial \sigma_s}(\sigma_i^2+\sigma_s^2)\frac{\partial \sigma_s}{\partial\left[\hat{\mu}_s\right]_x} = 2\sigma_s\frac{\partial \sigma_s}{\partial\left[\hat{\mu}_s\right]_x} = 2\sigma_s\frac{-\sigma_s}{\left[\mu_s^p\right]_z}p_{2,0} = \frac{-2\sigma_s^2}{\left[\mu_s^p\right]_z}p_{2,0}, \tag{B.7}$$

$$\begin{aligned}\frac{\partial}{\partial\left[\hat{\mu}_s\right]_x}\left(\frac{\sigma_i\sigma_s}{\sigma_i^2+\sigma_s^2}\right) &= \frac{\frac{\partial}{\partial[\hat{\mu}_s]_x}(\sigma_i\sigma_s)(\sigma_i^2+\sigma_s^2) - (\sigma_i\sigma_s)\frac{\partial}{\partial[\hat{\mu}_s]_x}(\sigma_s^2+\sigma_i^2)}{(\sigma_s^2+\sigma_i^2)^2} \\ &= \frac{\frac{-\sigma_i\sigma_s}{\left[\mu_s^p\right]_z}p_{2,0}(\sigma_i^2+\sigma_s^2)}{(\sigma_i^2+\sigma_s^2)^2} - \frac{\sigma_i\sigma_s\frac{-2\sigma_s^2}{\left[\mu_s^p\right]_z}p_{2,0}}{(\sigma_i^2+\sigma_s^2)^2} = \frac{\frac{-\sigma_i\sigma_s}{\left[\mu_s^p\right]_z}p_{2,0}}{\sigma_i^2+\sigma_s^2} + \frac{\frac{2\sigma_i\sigma_s^3}{\left[\mu_s^p\right]_z}p_{2,0}}{(\sigma_s^2+\sigma_i^2)^2} \\ &= 2\frac{\sigma_s^2\sigma_i^2}{\sigma_s^2+\sigma_i^2}\left(\frac{-p_{2,0}}{2\left[\mu_s^p\right]_z} + \frac{\sigma_s^2 p_{2,0}}{\left[\mu_s^p\right]_z(\sigma_i^2+\sigma_s^2)}\right),\end{aligned} \tag{B.8}$$

$$\begin{aligned}\frac{\partial}{\partial\left[\hat{\mu}_s\right]_x}(||\mu_i-\mu_s||^2) &= \frac{\partial}{\partial\left[\hat{\mu}_s\right]_x}((\mu_i-\mu_s)(\mu_i-\mu_s)^T) = 2(\mu_i-\mu_s)\left(-\frac{\partial}{\partial\left[\hat{\mu}_s\right]_x}(\mu_s)\right)\\ &= -2(\mu_i-\mu_s)\frac{\partial}{\partial\left[\hat{\mu}_s\right]_x}(\mu_s),\end{aligned} \tag{B.9}$$

$$\begin{aligned}\frac{\partial}{\partial\left[\hat{\mu}_s\right]_x}\left(-\frac{||\mu_i-\mu_s||^2}{\sigma_s^2+\sigma_i^2}\right) &= -\left(\frac{\frac{\partial}{\partial[\hat{\mu}_s]_x}(||\mu_i-\mu_s||^2)(\sigma_i^2+\sigma_s^2)}{(\sigma_i^2+\sigma_s^2)^2}-\frac{||\mu_i-\mu_s||^2\frac{\partial}{\partial[\hat{\mu}_s]_x}(\sigma_i^2+\sigma_s^2)}{(\sigma_i^2+\sigma_s^2)^2}\right)\\ &= -\frac{-2(\mu_i-\mu_s)\frac{\partial}{\partial[\hat{\mu}_s]_x}(\mu_s)(\sigma_i^2+\sigma_s^2)}{(\sigma_i^2+\sigma_s^2)^2}+\frac{||\mu_i-\mu_s||^2\frac{-2\sigma_s^2}{\left[\mu_s^p\right]_z}p_{2,0}}{(\sigma_i^2+\sigma_s^2)^2}\\ &= -\frac{-2(\mu_i-\mu_s)\frac{\partial}{\partial[\hat{\mu}_s]_x}(\mu_s)}{\sigma_i^2+\sigma_s^2}+\frac{||\mu_i-\mu_s||^2\frac{-2\sigma_s^2}{\left[\mu_s^p\right]_z}p_{2,0}}{(\sigma_i^2+\sigma_s^2)^2}\\ &= 2\left(\frac{(\mu_i-\mu_s)\frac{\partial}{\partial[\hat{\mu}_s]_x}(\mu_s)}{\sigma_i^2+\sigma_s^2}-\frac{||\mu_i-\mu_s||^2\frac{\sigma_s^2}{\left[\mu_s^p\right]_z}p_{2,0}}{(\sigma_s^2+\sigma_i^2)^2}\right),\end{aligned} \tag{B.10}$$

$$\begin{aligned}\frac{\partial}{\partial\left[\hat{\mu}_s\right]_x}\left(e^{-\frac{||\mu_i-\mu_s||^2}{\sigma_s^2+\sigma_i^2}}\right) &= e^{-\frac{||\mu_i-\mu_s||^2}{\sigma_s^2+\sigma_i^2}}\frac{\partial}{\partial\left[\hat{\mu}_s\right]_x}\left(-\frac{||\mu_i-\mu_s||^2}{\sigma_s^2+\sigma_i^2}\right)\\ &= e^{-\frac{||\mu_i-\mu_s||^2}{\sigma_s^2+\sigma_i^2}}2\left(\frac{(\mu_i-\mu_s)\frac{\partial}{\partial[\hat{\mu}_s]_x}(\mu_s)}{\sigma_s^2+\sigma_i^2}-\frac{||\mu_i-\mu_s||^2\frac{\sigma_s^2}{\left[\mu_s^p\right]_z}p_{2,0}}{(\sigma_s^2+\sigma_i^2)^2}\right),\end{aligned} \tag{B.11}$$

$$\begin{aligned}\frac{\partial}{\partial\left[\hat{\mu}_s\right]_x}\left(\frac{\sigma_i\sigma_s}{\sigma_i^2+\sigma_s^2}e^{-\frac{||\mu_i-\mu_s||^2}{\sigma_s^2+\sigma_i^2}}\right) &= \frac{\partial}{\partial\left[\hat{\mu}_s\right]_x}\left(\frac{\sigma_i\sigma_s}{\sigma_i^2+\sigma_s^2}\right)e^{-\frac{||\mu_i-\mu_s||^2}{\sigma_s^2+\sigma_i^2}}+\frac{\sigma_i\sigma_s}{\sigma_i^2+\sigma_s^2}\frac{\partial}{\partial\left[\hat{\mu}_s\right]_x}\left(e^{-\frac{||\mu_i-\mu_s||^2}{\sigma_s^2+\sigma_i^2}}\right)\\ &= 2\frac{\sigma_s^2\sigma_i^2}{\sigma_s^2+\sigma_i^2}\left(\frac{-p_{2,0}}{2\left[\mu_s^p\right]_z}+\frac{\sigma_s^2 p_{2,0}}{\left[\mu_s^p\right]_z(\sigma_i^2+\sigma_s^2)}\right)e^{-\frac{||\mu_i-\mu_s||^2}{\sigma_s^2+\sigma_i^2}}+\frac{\sigma_i\sigma_s}{\sigma_i^2+\sigma_s^2}e^{-\frac{||\mu_i-\mu_s||^2}{\sigma_s^2+\sigma_i^2}}2\left(\frac{(\mu_i-\mu_s)\frac{\partial}{\partial k_s}(\mu_s)}{\sigma_s^2+\sigma_i^2}\right.\\ &\quad\left.-\frac{||\mu_i-\mu_s||^2\frac{\sigma_s^2}{[\mu_s^h]_z}[P\hat{\mathbf{n}}_s^h]_z}{(\sigma_s^2+\sigma_i^2)^2}\right)\\ &= 2\frac{\sigma_i\sigma_s}{\sigma_i^2+\sigma_s^2}e^{-\frac{||\mu_i-\mu_s||^2}{\sigma_s^2+\sigma_i^2}}\left(\frac{-p_{2,0}}{2\left[\mu_s^p\right]_z}+\frac{\sigma_s^2 p_{2,0}}{\left[\mu_s^p\right]_z(\sigma_s^2+\sigma_i^2)}+\frac{(\mu_i-\mu_s)\frac{\partial}{\partial[\hat{\mu}_s]_x}(\mu_s)}{\sigma_i^2+\sigma_s^2}-\frac{||\mu_i-\mu_s||^2\sigma_s^2 p_{2,0}}{\left[\mu_s^p\right]_z(\sigma_i^2+\sigma_s^2)^2}\right)\\ &= 2\frac{\sigma_i\sigma_s}{\sigma_i^2+\sigma_s^2}e^{-\frac{||\mu_i-\mu_s||^2}{\sigma_s^2+\sigma_i^2}}\left[p_{2,0}\left(-\frac{1}{2}+\frac{\sigma_s^2}{\sigma_i^2+\sigma_s^2}-\frac{||\mu_i-\mu_s||^2\sigma_s^2}{(\sigma_i^2+\sigma_s^2)^2}\right)\frac{1}{\left[\mu_s^p\right]_z}+\frac{(\mu_i-\mu_s)\frac{\partial}{\partial[\hat{\mu}_s]_x}(\mu_s)}{\sigma_i^2+\sigma_s^2}\right].\end{aligned} \tag{B.12}$$

Therefore, the derivative of the similarity $\Phi_{i,s}$ is obtained as:

$$
\begin{aligned}
\frac{\partial \Phi_{i,s}}{\partial\,[\hat{\mu}_s]_x} &= T(\delta_{is})2\frac{\partial}{\partial\,[\hat{\mu}_s]_x}\left(\frac{\sigma_i\sigma_s}{\sigma_i^2+\sigma_s^2}e^{-\frac{||\mu_i-\mu_s||^2}{\sigma_i^2+\sigma_s^2}}\right)\\
&= T(\delta_{is})2\left\{2\frac{\sigma_i\sigma_s}{\sigma_i^2+\sigma_s^2}e^{-\frac{||\mu_i-\mu_s||^2}{\sigma_s^2+\sigma_i^2}}\left[p_{2,0}\left(-\frac{1}{2}+\frac{\sigma_s^2}{\sigma_i^2+\sigma_s^2}-\frac{||\mu_i-\mu_s||^2\sigma_s^2}{(\sigma_i^2+\sigma_s^2)^2}\right)\frac{1}{\left[\mu_s^p\right]_z}+\frac{(\mu_i-\mu_s)\frac{\partial}{\partial[\hat{\mu}_s]_x}(\mu_s)}{\sigma_i^2+\sigma_s^2}\right]\right\}\\
&= T(\delta_{is})4\frac{\sigma_i\sigma_s}{\sigma_i^2+\sigma_s^2}e^{-\frac{||\mu_i-\mu_s||^2}{\sigma_s^2+\sigma_i^2}}\left[p_{2,0}\left(-\frac{1}{2}+\frac{\sigma_s^2}{\sigma_i^2+\sigma_s^2}-\frac{||\mu_i-\mu_s||^2\sigma_s^2}{(\sigma_i^2+\sigma_s^2)^2}\right)\frac{1}{\left[\mu_s^p\right]_z}+\frac{(\mu_i-\mu_s)\frac{\partial}{\partial[\hat{\mu}_s]_x}(\mu_s)}{\sigma_i^2+\sigma_s^2}\right]\\
&= 2\Phi_{i,s}\left[p_{2,0}\left(-\frac{1}{2}+\frac{\sigma_s^2}{\sigma_i^2+\sigma_s^2}-\frac{||\mu_i-\mu_s||^2\sigma_s^2}{(\sigma_i^2+\sigma_s^2)^2}\right)\frac{1}{\left[\mu_s^p\right]_z}+\frac{(\mu_i-\mu_s)\frac{\partial\mu_s}{\partial[\hat{\mu}_s]_x}}{\sigma_i^2+\sigma_s^2}\right],\\
\frac{\partial \Phi_{i,s}}{\partial\,[\hat{\mu}_s]_y} &= 2\Phi_{i,s}\left[p_{2,1}\left(-\frac{1}{2}+\frac{\sigma_s^2}{\sigma_i^2+\sigma_s^2}-\frac{||\mu_i-\mu_s||^2\sigma_s^2}{(\sigma_i^2+\sigma_s^2)^2}\right)\frac{1}{\left[\mu_s^p\right]_z}+\frac{(\mu_i-\mu_s)\frac{\partial\mu_s}{\partial[\hat{\mu}_s]_y}}{\sigma_i^2+\sigma_s^2}\right],\\
\frac{\partial \Phi_{i,s}}{\partial\,[\hat{\mu}_s]_z} &= 2\Phi_{i,s}\left[p_{2,2}\left(-\frac{1}{2}+\frac{\sigma_s^2}{\sigma_s^2+\sigma_i^2}-\frac{||\mu_i-\mu_s||^2\sigma_s^2}{(\sigma_i^2+\sigma_s^2)^2}\right)\frac{1}{\left[\mu_s^p\right]_z}+\frac{(\mu_i-\mu_s)\frac{\partial\mu_s}{\partial[\hat{\mu}_s]_z}}{\sigma_i^2+\sigma_s^2}\right].
\end{aligned}
\tag{B.13}
$$

Finally the derivative of the similarity term $E_{sim}$ is:

$$
\frac{\partial E_{sim}}{\partial[\hat{\mu}_s]_{x,y,z}} = \frac{\partial E_{sim}}{\partial\mathcal{M}_s} = \frac{1}{n_c}\sum_{c=1}^{n_c}\frac{1}{n_i^c}\sum_{i=1}^{n_i^c}\begin{cases}\frac{\partial\Phi_{i,s}}{\partial\mathcal{M}_s} & \text{if}\sum_{s=1}^{n_s}\Phi_{i,s}<1\\ 0 & \text{otherwise.}\end{cases}
\tag{B.14}
$$

### B.0.2 Derivative of the Countour Term

The contour term $E_{cont}$ can be considered as an extension of the $E_{sim}$ term, where the overlap is estimated simultaneously for two Surface Gaussians, i. e. the inside and outside Gaussian components of the Border Gaussians, at a time. The partial derivatives for the inside and the outside Gaussian similarity, i. e. $\Phi_{i,b_{in}}$ and $\overline{\Phi}_{i,b_{out}}$ can be computed using the formula in Section B.0.1. In particular, the derivatives w. r. t. $\overline{\Phi}_{i,b_{out}}$ have inverted color similarity weight:

$$
\begin{aligned}
\frac{\partial\overline{\Phi}_{i,b_{out}}}{\partial[\hat{\mu}_b]_{x,y,z}} &= (1-T_{\Delta_c}(\delta_{i,b_{out}}))\frac{\partial}{\partial[\hat{\mu}_b]_{x,y,z}}\left[\int_\Omega\frac{1}{\sqrt{\pi\sigma_{b_{out}}\sigma_i}}exp\left(-\frac{1}{2}\frac{||x-\mu_i||^2}{\sigma_i^2}\right)\cdot\, exp\left(-\frac{1}{2}\frac{||x-\mu_{b_{out}}||^2}{\sigma_{b_{out}}^2}\right)\partial x\right]^2 =\\
&= 2\overline{\Phi}_{i,b_{out}}\left[p_{2,\{0,1,2\}}\left(-\frac{1}{2}+\frac{\sigma_{b_{out}}^2}{\sigma_s^2+\sigma_i^2}-\frac{||\mu_i-\mu_s||^2\sigma_{b_{out}}^2}{(\sigma_i^2+\sigma_s^2)^2}\right)\frac{1}{\left[\mu_s^p\right]_z}+\frac{(\mu_i-\mu_{b_{out}})\frac{\partial\mu_b}{\partial[\hat{\mu}_b]_{x,y,z}}}{\sigma_i^2+\sigma_s^2}\right].
\end{aligned}
\tag{B.15}
$$

Finally, the derivative of $E_{cont}$ is:

$$\frac{\partial E_{cont}}{\partial [\hat{\mu}_s]_{x,y,z}} = \frac{\partial E_{cont}}{\partial \mathcal{M}_b} = \frac{1}{n_c}\sum_{c=1}^{n_c}\left[\frac{1}{n_i^{c2}}\sum_{i=1}^{n_i^c}\frac{\partial}{\partial \mathcal{M}_b}\left\{min\left(\sum_{b=1}^{n_b^c}\Phi_{i,b_{in}},1\right)\cdot min\left(\sum_{b=1}^{n_b^c}\overline{\Phi}_{i,b_{out}},1\right)\right\}\right] =$$

$$= \frac{1}{n_c}\sum_{c=1}^{n_c}\frac{1}{n_i^{c2}}\sum_{i=1}^{n_i^c}\begin{cases}\overline{\Phi}_{i,b_{out}}\frac{\partial \Phi_{i,b_{in}}}{\partial \mathcal{M}_b}+\Phi_{i,b_{in}}\frac{\partial \overline{\Phi}_{i,b_{out}}}{\partial \mathcal{M}_b} & \text{if } \sum_{b=1}^{n_b}\Phi_{i,b_{in}} < 1 \text{ and } \sum_{b=1}^{n_b}\overline{\Phi}_{i,b_{out}} < 1\\ \Phi_{i,b_{in}}\frac{\partial \overline{\Phi}_{i,b_{out}}}{\partial \mathcal{M}_b} & \text{else if } \sum_{b=1}^{n_b}\Phi_{i,b_{in}} \geq 1\\ \overline{\Phi}_{i,b_{out}}\frac{\partial \Phi_{i,b_{in}}}{\partial \mathcal{M}_b} & \text{else if } \sum_{b=1}^{n_b}\overline{\Phi}_{i,b_{out}} \geq 1\\ 0 & \text{otherwise.}\end{cases} \quad . \tag{B.16}$$

### B.0.3 Properties of the Contour Term

Notice that the contour term evaluates in the interval $[0,1]$, in fact:

$$\begin{aligned} max(E_{cont}) &= \frac{1}{n_c}\sum_{c=1}^{n_c}\frac{1}{|I(c)|^2}\left[\left(\sum_{i=1}^{n_i^c}1\right)\cdot\left(\sum_{i=1}^{n_i^{c2}}1\right)\right] = \\ &= \frac{1}{n_c}\sum_{c=1}^{n_c}\frac{1}{n_i^{c2}}\left[n_i^c\cdot n_i^c\right] = \frac{1}{n_c}\sum_{c=1}^{n_c}1 = \frac{n_c}{n_c} = 1, \\ min(E_{cont}) &= \frac{1}{n_c}\sum_{c=1}^{n_c}\frac{1}{n_i^{c2}}\cdot 0 = 0. \end{aligned} \tag{B.17}$$

Figure B.1 shows various example plots of $E_{cont}$ obtained by finding the optimal Border Gaussians for various standard deviations (sigma), given a set of 4 Image Gaussians with random sigmas, mean and color. The color of the Image Gaussians is randomly chones to be either blue or red. These energy plots are maximized where the Image Gaussian form a border: for all cases, the optimal Border Gaussian have mean at the Image Gaussian border. Notice how smaller standard deviations in general produce more precise alignment to the border.

In the first example (top) the only border has been detected sucessfully. In the middle example the most prominent border of the two is detected using different standard deviations. Notice that the resulting energy in this case has a double spike for smaller standard deviations, which suggests where the the borders are located. In the bottom more complex case, one can see that by varying the standard deviations of the optimal Border Gaussian one can easily recognize two out of the three borders.

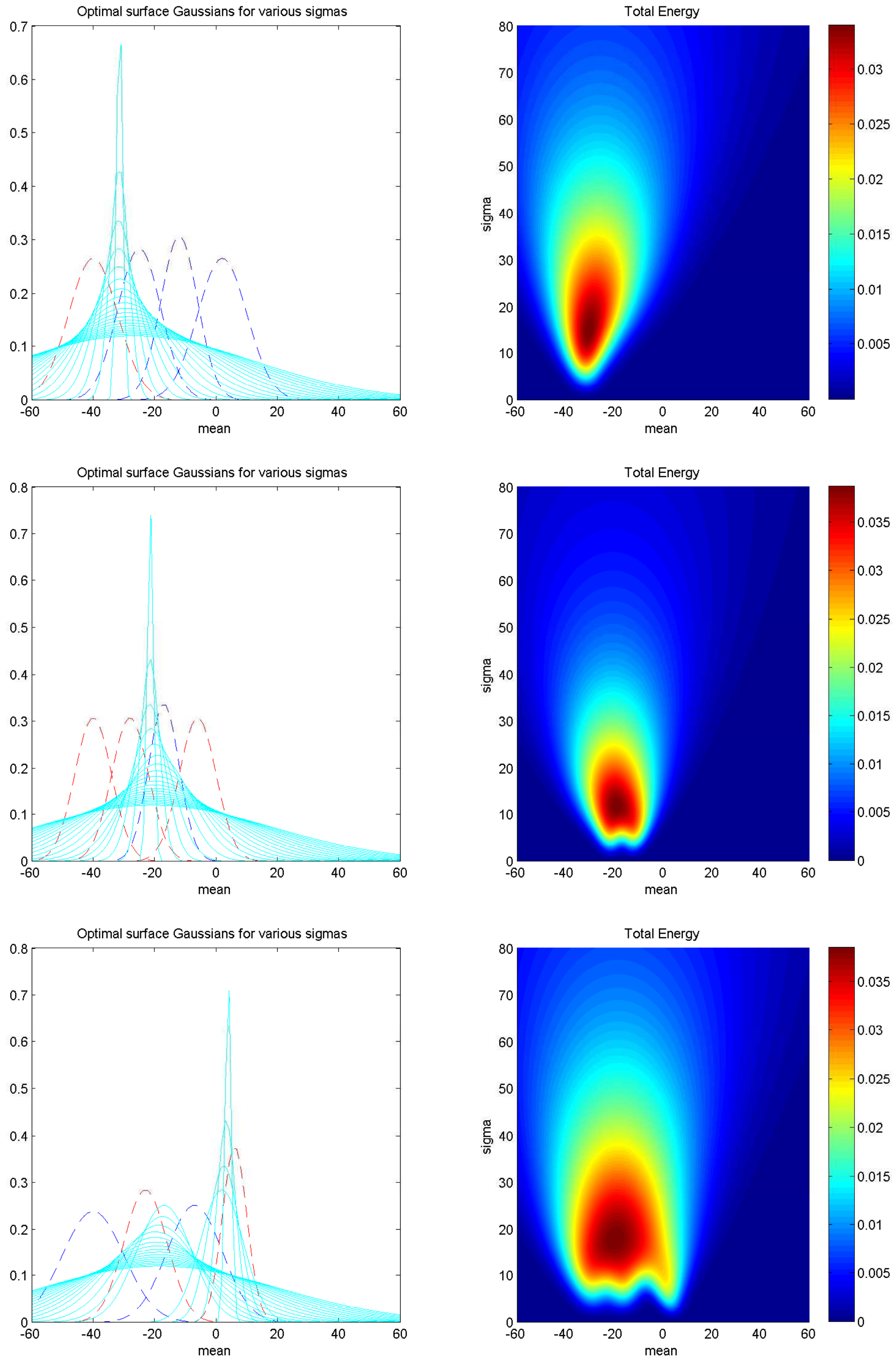


**Figure B.1:** On the left: Plots of 1D image Gaussians (dashed blue and red corresponding to opposite color similarity wrt a pair of Gaussian) together with the optimal set of Border Gaussians in cyan obtained for different standard deviations (sigmas). On the right: the Border Gaussian energy plotted for various means and sigmas.

# Bibliography

[1] A. Agarwal and B. Triggs. Recovering 3D human pose from monocular images. *TPAMI*, 2006. 39

[2] Agisoft. Photoscan: photogrammetric processing software, 2006. 17

[3] N. Ahmed, C. Theobalt, P. Dobrev, and H. P. Seidel. Robust fusion of dynamic shape and normal capture for high-quality reconstruction of time-varying geometry. In *CVPR*, 2008. 67

[4] B. Allain, J.-S. Franco, and E. Boyer. An Efficient Volumetric Framework for Shape Tracking. In *CVPR 2015 - IEEE International Conference on Computer Vision and Pattern Recognition*, 2015. 67

[5] B. Allen, B. Curless, and Z. Popović. The space of human body shapes: reconstruction and parameterization from range scans. In *ACM Transactions on Graphics (TOG)*. ACM, 2003. 10, 91

[6] S. Amin, M. Andriluka, M. Rohrbach, and B. Schiele. Multi-view Pictorial Structures for 3D Human Pose Estimation. In *BMVC*, 2013. 40

[7] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. In *ACM Transactions on Graphics (TOG)*. ACM, 2005. 10, 91

[8] O. K.-C. Au, C.-L. Tai, H.-K. Chu, D. Cohen-Or, and T.-Y. Lee. Skeleton extraction by mesh contraction. *ACM Trans. Graph.*, 2008. 20

[9] L. Ballan and G. M. Cortelazzo. Marker-less motion capture of skinned models in a four camera set-up using optical flow and silhouettes. In *3DPVT*, 2008. 9, 39, 67, 90

[10] I. Baran and J. Popovic. Automatic rigging and animation of 3d characters. *ACM TOG (Proc. SIGGRAPH)*, 2007. 27, 28, 29

[11] T. Beeler, D. Bradley, H. Zimmer, and M. Gross. Improved reconstruction of deforming surfaces by cancelling ambient occlusion. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part I*, ECCV'12, 2012. 120

[12] G. L. Bernstein and C. Wojtan. Putting holes in holey geometry: Topology change for arbitrary surfaces. *ACM Trans. Graph.*, 2013. 119

[13] L. Bo and C. Sminchisescu. Twin gaussian processes for structured prediction. *IJCV*, 2010. 39

[14] F. Bogo, M. J. Black, M. Loper, and J. Romero. Detailed full-body reconstructions of moving people from monocular RGB-D sequences. In *ICCV*, 2015. 10, 91

[15] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *ECCV*, 2016. 40

[16] Botspot. Optaone, 2017. 14, 15, 16

[17] J.-Y. Bouguet. Camera calibration toolbox for matlab, 1998. 13

[18] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR '98. IEEE Computer Society, 1998. 50

[19] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *TPAMI*, 2001. 37, 50

[20] D. Bradley, T. Popa, A. Sheffer, W. Heidrich, and T. Boubekeur. Markerless garment capture. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 2008. 3, 65, 67, 89, 119

[21] M. Bray, P. Kohli, and P. Torr. PoseCut: simultaneous segmentation and 3D pose estimation of humans using dynamic graph-cuts. In *ECCV*, 2006. 3, 37, 39, 91

[22] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *CVPR*, 1998. 21, 23

[23] T. Brox, A. Bruhn, N. Papenberg, and J. Weicker. High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision, ECCV*, pages 25–36, 2004. 80

[24] T. Brox, B. Rosenhahn, J. Gall, and D. Cremers. Combined Region and Motion-Based 3D Tracking of Rigid and Articulated Objects. *TPAMI*, 2010. 3, 37, 39

[25] C. Budd, P. Huang, M. Klaudiny, and A. Hilton. Global non-rigid alignment of surface sequences. *IJCV*, 2013. 65, 66, 67

[26] M. Burenius, J. Sullivan, and S. Carlsson. 3D pictorial structures for multiple view articulated pose estimation. In *CVPR*, 2013. 40

[27] C. Cagniart, E. Boyer, and S. Ilic. Free-form mesh tracking: a patch-based approach. In *Proc. IEEE CVPR*, 2010. 67, 118, 119

[28] C. Cagniart, E. Boyer, and S. Ilic. Probabilistic deformable surface tracking from multiple videos. In *Proc. ECCV*, 2010. 119

[29] S. Capell, S. Green, B. Curless, T. Duchamp, and Z. Popović. Interactive skeleton-driven dynamic deformations. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '02, 2002. 26

[30] T. Captury. The captury, 2011. 13, 26

[31] J. Carranza, C. Theobalt, M. Magnor, and H.-P. Seidel. Free-viewpoint video of human actors. In *ACM TOG (Proc. SIGGRAPH '03)*, 2003. 67

[32] D. Casas, M. Volino, J. Collomosse, and A. Hilton. 4D Video Textures for Interactive Character Appearance. *Computer Graphics Forum*, 2014. 103

[33] B. Chang, S. Woo, and I. Ihm. Gpu-based parallel construction of compact visual hull meshes. *Vis. Comput.*, 2014. 18

[34] D. C. Chen, S. Denman, and C. B. Fookes. Accurate silhouette segmentation using motion detection and graph cuts. In *10th International Conference on Information Science, Signal Processing and their Applications*, 2010. 18

[35] A. Collet, M. Chuang, P. Sweeney, D. Gillett, D. Evseev, D. Calabrese, H. Hoppe, A. Kirk, and S. Sullivan. High-quality streamable free-viewpoint video. *ACM Trans. Graph.*, 2015. 65, 67, 119

[36] C. Coniglio, C. Meurie, O. Lézoray, and M. Berbineau. A graph based people silhouette segmentation using combined probabilities extracted from appearance, shape template prior, and color distributions. In *Advanced Concepts for Intelligent Vision Systems*. Springer International Publishing, 2015. 18

[37] Y. Cui, W. Chang, T. Nöll, and D. Stricker. KinectAvatar: Fully automatic body capture using a single kinect. In *ACCV Workshops*, 2012. 91

[38] F. Da, C. Batty, and E. Grinspun. Multimaterial mesh-based surface tracking. *ACM Trans. on Graphics (SIGGRAPH 2014)*, 2014. 119

[39] Y. V. David Fofi, Tadeusz Sliwa. A comparative survey on invisible structured light, 2004. 15

[40] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. *ACM TOG (Proc. of SIGGRAPH)*, 2008. 3, 9, 21, 23, 65, 66, 67, 89, 119

[41] E. de Aguiar, C. Theobalt, C. Stoll, and H.-P. Seidel. Marker-less deformable mesh tracking for human shape and motion capture. In *CVPR*, 2007. 67

[42] P. Debevec. The Light Stages and Their Applications to Photoreal Digital Actors. In *SIGGRAPH Asia*, 2012. 120

[43] M. Desbrun and M.-P. Gascuel. Animating soft substances with implicit surfaces. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, 1995. 119

[44] Y. Ding, J. Xiao, and J. Yu. A theory of multi-perspective defocusing. In *CVPR 2011*, 2011. 16

[45] A. Djelouah, J.-S. Franco, E. Boyer, F. Le Clerc, and P. Pérez. Sparse multi-view consistency for object segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015. 91

[46] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S. R. Fanello, A. Kowdle, S. O. Escolano, C. Rhemann, D. Kim, J. Taylor, P. Kohli, V. Tankovich, and S. Izadi. Fusion4d: Real-time performance capture of challenging scenes. *ACM Trans. Graph.*, 2016. 22, 119

[47] S. J. N. Drvar. The assessment of structured light and laser scanning methods in 3d shape measurements. In *Proceedings of the 4 th International Congress of Croatian Society of Mechanics*, 2003. 15

[48] Y. Du, Y. Wong, Y. Liu, F. Han, Y. Gui, Z. Wang, M. Kankanhalli, and W. Geng. Marker-less 3D human motion capture with monocular image sequence and height-maps. In *ECCV*, 2016. 37, 39

[49] D. A. Edilson, T. Christian, T. Sebastian, and S. Hans-Peter. Automatic conversion of mesh animations into skeleton-based animations. *Computer Graphics Forum*, 2008. 25, 28

[50] A. Elhayek, E. Aguiar, A. Jain, J. Tompson, L. Pishchulin, M. Andriluka, C. Bregler, B. Schiele, and C. Theobalt. Efficient convnet-based marker-less motion capture in general scenes with a low number of cameras. In *CVPR*, 2015. 9, 21, 37, 102

[51] A. Elhayek, C. Stoll, N. Hasler, K.-i. Kim, H.-P. Seidel, and C. Theobalt. Spatio-temporal motion tracking with unsynchronized cameras. In *Proc. CVPR*, CVPR '12, 2012. 3, 116

[52] A. Elhayek, C. Stoll, K. I. Kim, H. P. Seidel, and C. Theobalt. Feature-based multi-video synchronization with subframe accuracy. In *Pattern Recognition*, 2012. 116

[53] A. Elhayek, C. Stoll, K. I. Kim, and C. Theobalt. Outdoor Human Motion Capture by Simultaneous Optimization of Pose and Camera Parameters. In *CGF*, 2015. 37, 39, 40, 116

[54] U. Fecker, M. Barkowsky, and A. Kaup. Histogram-based prefiltering for luminance and chrominance compensation of multiview video. *IEEE Transactions on Circuits and Systems for Video Technology*, 2008. 14

[55] A. Feng, D. Casas, and A. Shapiro. Avatar reshaping and automatic rigging using a deformable model. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, MIG '15. ACM, 2015. 25, 28

[56] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 1981. 12

[57] Fit3D. Proscanner, 2014. 14

[58] B. Foundation. Blender: free and open source 3d creation suite, 1998. 17

[59] J.-S. Franco and E. Boyer. Exact polyhedral visual hulls. In *British Machine Vision Conference*, 2003. 18

[60] J.-S. Franco and E. Boyer. Efficient polyhedral modeling from silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009. 11, 16, 67, 89, 117

[61] S. Fuhrmann, F. Langguth, and M. Goesele. Mve: A multi-view reconstruction environment. In *Proceedings of the Eurographics Workshop on Graphics and Cultural Heritage*, 2014. 16, 117

[62] J. Gall, B. Rosenhahn, T. Brox, and H. P. Seidel. Optimization and filtering for human motion capture: A multi-layer framework. *IJCV*, 2010. 39

[63] J. Gall, B. Rosenhahn, and H. P. Seidel. Drift-free tracking of rigid and articulated objects. In *CVPR*, 2008. 3, 37, 39

[64] J. Gall, C. Stoll, E. Aguiar, C. Theobalt, B. Rosenhahn, and H.-P. Seidel. Motion capture using joint skeleton tracking and surface estimation. In *Proc. IEEE CVPR*, 2009. 3, 9, 65, 66, 67, 76, 77, 89, 90, 91, 102, 106, 107, 108, 119

[65] M. Germann, A. Sorkine-Hornung, R. Keiser, R. Ziegler, S. Würmlin, and M. H. Gross. Articulated billboards for video-based rendering. *Comput. Graph. Forum*, 2010. 116

[66] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz. Multi-view stereo for community photo collections. In *2007 IEEE 11th International Conference on Computer Vision*, 2007. 16, 117

[67] GoPro. Gopro: The world's most versatile action cameras, 2002. 55, 102

[68] G. Group. Gvvperfcapeva: Human shape and performance capture datasets, 2013. 55

[69] J.-Y. Guillemaut and A. Hilton. Joint multi-layer segmentation and reconstruction for free-viewpoint video applications. *International Journal of Computer Vision*, 2011. 91

[70] K. Guo, F. Xu, Y. Wang, Y. Liu, and Q. Dai. Robust non-rigid motion tracking and surface reconstruction using l0 regularization. In *ICCV*, 2015. 91

[71] M. Gupta and S. K. Nayar. Micro phase shifting. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 15

[72] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. 12

[73] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, 2004. 16

[74] N. Hasler, B. Rosenhahn, T. Thormählen, M. Wand, J. Gall, and H. P. Seidel. Markerless motion capture with unsynchronized moving cameras. In *CVPR Workshops*, 2009. 3, 37, 39, 91, 116

[75] N. Hasler, T. Thormählen, B. Rosenhahn, and H.-P. Seidel. Learning skeletons for shape and pose. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, 2010. 28

[76] T. Helten, A. Baak, G. Bharaj, M. Müller, H.-P. Seidel, and C. Theobalt. Personalization and evaluation of a real-time depth-based full body tracker. In *3DV*, 2013. 91

[77] C. Hernandez, G. Vogiatzis, G. J. Brostow, B. Stenger, and R. Cipolla. Non-rigid photometric stereo with colored lights. In *Proc. ICCV*, 2007. 67

[78] C. Hernandez, G. Vogiatzis, and R. Cipolla. Multiview photometric stereo. 2008. 120

[79] O. Hössjer and C. Croux. Generalizing univariate signed rank statistics for testing and estimating a multivariate location parameter. *Journal of Nonparametric Statistics*, 1995. 46

[80] S. Ilic and P. Fua. Implicit meshes for surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2006. 9, 22

[81] P. Inc. Phasespace vision camera, 1994. 55, 56, 102

[82] P. R. Induchoodan, M. J. Josemartin, and P. R. Geetharanjin. Depth recovery from stereo images. In *2014 International Conference on Contemporary Computing and Informatics (IC3I)*, 2014. 16

[83] T. Industrie. Symcad iii, 2017. 14

[84] M. Innmann, M. Zollhoefer, M. Niessner, C. Theobalt, and M. Stamminger. Volumedeform: Real-time volumetric non-rigid reconstruction, 2016. 22

[85] C. Ionescu, L. Bo, and C. Sminchisescu. Structural svm for visual localization and continuous state estimation. In *2009 IEEE 12th International Conference on Computer Vision*, 2009. 39

[86] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In *Proc. UIST*. ACM, 2011. 120

[87] A. Jacobson, I. Baran, L. Kavan, J. Popović, and O. Sorkine. Fast automatic skinning transformations. *ACM Trans. Graph.*, 2012. 27

[88] A. Jain, T. Thormählen, H.-P. Seidel, and C. Theobalt. Moviereshape: tracking and reshaping of humans in videos. *ACM TOG*, 2010. 10, 91

[89] D. L. James and C. D. Twigg. Skinning mesh animations. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, 2005. 28

[90] S. X. Ju, M. J. Black, and Y. Yacoob. Cardboard people: A parameterized model of articulated image motion. In *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on*, 1996. 21

[91] P. JunJun, Y. Xiaosong, X. Xin, W. Philip, and Z. J. J. Automatic rigging for animation characters with 3d silhouette. *Computer Animation and Virtual Worlds*, 2009. 25

[92] A. Kanazawa, S. Kovalsky, R. Basri, and D. W. Jacobs. Learning 3d articulation and deformation using 2d images. *arXiv preprint arXiv:1507.07646*, 2015. 91

[93] M. Kasap and N. Magnenat-Thalmann. Parameterized human body model for real-time applications. *2007 International Conference on Cyberworlds (CW'07)*, 2007. 25

[94] S. Katz and A. Tal. Hierarchical mesh decomposition using fuzzy clustering and cuts. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, 2003. 27

[95] L. Kavan, S. Collins, J. Žára, and C. O'Sullivan. Skinning with dual quaternions. In *I3D*, 2007. 26

[96] L. Kavan and O. Sorkine. Elasticity-inspired deformers for character articulation. *ACM Trans. Graph.*, 2012. 26

[97] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP '06, 2006. 14, 16

[98] B. Kenwright. A beginners guide to dual-quaternions: What they are, how they work, and how to use them for 3d. In *The 20th International Conference on Computer Graphics, Visualization and Computer Vision*, 2012. 26

[99] H. Kim and A. Hilton. Influence of colour and feature geometry on multi-modal 3d point clouds data registration. In *International Conference on 3D Vision (3DV)*, 2014. 102, 108

[100] K. Kolev, M. Klodt, T. Brox, and D. Cremers. Continuous global optimization in multiview 3d reconstruction. *International Journal of Computer Vision*, 2009. 65, 67

[101] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1994. 18

[102] B. H. Le and Z. Deng. Robust and accurate skeletal rigging from mesh sequences. *ACM Trans. Graph.*, 2014. 20, 25, 28

[103] C. S. Lee and A. Elgammal. Coupled visual and kinematic manifold models for tracking. *IJCV*, 2010. 39

[104] A. Letouzey and E. Boyer. Progressive shape models. In *Proc. CVPR*. IEEE, 2012. 118, 119

[105] S. Li, A. B. Chan, and A. B. C. Sijin Li. 3D Human Pose Estimation from Monocular Images with Deep Convolutional Neural Network. In *ACCV*, 2014. 37, 39

[106] Y. Liu, J. Gall, C. Stoll, Q. Dai, H.-P. Seidel, and C. Theobalt. Markerless motion capture of multiple characters using multiview image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013. 39

[107] Y. Liu, C. Stoll, J. Gall, H.-P. Seidel, and C. Theobalt. Markerless motion capture of interacting characters using multi-view image segmentation. In *CVPR*, 2011. 39, 67

[108] Y. Livny, F. Yan, M. Olson, B. Chen, H. Zhang, and J. El-Sana. Automatic reconstruction of tree skeletal structures from point clouds. *ACM Trans. Graph.*, 2010. 20

[109] M. Loper, N. Mahmood, and M. J. Black. MoSh: Motion and shape capture from sparse markers. *ACM Transactions on Graphics*, 2014. 10, 91

[110] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 2015. 10, 91, 117

[111] S.-P. Lu, B. Ceulemans, A. Munteanu, and P. Schelkens. Spatio-temporally consistent color and structure optimization for multiview video color correction. *IEEE Transactions on Multimedia*, 2015. 14

[112] M. Macklin, M. Müller, N. Chentanez, and T.-Y. Kim. Unified particle physics for real-time applications. *ACM Trans. Graph.*, 2014. 21, 119

[113] N. Magnenat-Thalmann, R. Laperrière, and D. Thalmann. Joint-dependent local deformations for hand animation and object grasping. In *Proceedings on Graphics Interface '88*, 1988. 26

[114] W. Matusik, C. Buehler, and L. McMillan. Polyhedral visual hulls for real-time rendering. In *Rendering Techniques 2001*. Springer Vienna, 2001. 18

[115] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan. Image-based visual hulls. In *SIGGRAPH*, 2000. 67

[116] D. Mehta, H. Rhodin, D. Casas, O. Sotnychenko, W. Xu, and C. Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3DV*, 2017. 37, 39, 61

[117] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. In *SIGGRAPH*, 2017. 37, 39

[118] C. Miller, O. Arikan, and D. Fussell. Frankenrigs: Building character rigs from multiple sources. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '10, 2010. 25

[119] J. R. Mitchelson and A. Hilton. Wand-based multiple camera studio calibration. 2007. 12

[120] L. Moccozet, F. Dellas, N. Magnenat-Thalmann, S. Biasotti, M. Mortara, B. Falcidieno, P. Min, and R. Veltkamp. Animatable human body model reconstruction from 3d scan data using templates. In *Proceedings CapTech Workshop on Modelling and Motion Capture Techniques for Virtual Environments*, 2004. 25

[121] G. Mori and J. Malik. Recovering 3D human body configurations using shape contexts. *TPAMI*, 2006. 39

[122] R. M. Murray, S. S. Sastry, and L. Zexiang. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., 1994. 19

[123] A. Mustafa, H. Kim, J.-Y. Guillemaut, and A. Hilton. General dynamic scene reconstruction from multiple view video. In *ICCV*, 2015. 89, 91

[124] A. Mustafa, H. Kim, J.-Y. Guillemaut, and A. Hilton. Temporally coherent 4d reconstruction of complex dynamic scenes. In *CVPR*, 2016. 89, 91

[125] D. Nehab, S. Rusinkiewicz, J. Davis, and R. Ramamoorthi. Efficiently combining positions and normals for precise 3D geometry. 2005. 120

[126] R. A. Newcombe and A. Davison. Live dense reconstruction with a single moving camera. In *CVPR*, 2010. 120

[127] R. A. Newcombe, D. Fox, and S. M. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *CVPR*, 2015. 22

[128] R. A. Newcombe, S. Izadi, O. Hilliges, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. 2011. 22

[129] W. Ofir, S. Olga, L. Yaron, and G. Craig. Context-aware skeletal shape deformation. *Computer Graphics Forum*, 2007. 27

[130] S. Orts-Escolano, C. Rhemann, S. Fanello, W. Chang, A. Kowdle, Y. Degtyarev, D. Kim, P. L. Davidson, S. Khamis, M. Dou, V. Tankovich, C. Loop, Q. Cai, P. A. Chou, S. Mennicken, J. Valentin, V. Pradeep, S. Wang, S. B. Kang, P. Kohli, Y. Lutchyn, C. Keskin, and S. Izadi. Holoportation: Virtual 3d teleportation in real-time. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, 2016. 119

[131] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer Verlag, 2003. 22, 119

[132] O. Özyesil, V. Voroninski, R. Basri, and A. Singer. A survey on structure from motion. *CoRR*, 2017. 116

[133] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis. Coarse-to-fine volumetric prediction for single-image 3D human pose. In *CVPR*, 2017. 37, 39

[134] J. M. Perez, P. G. Aledo, and P. P. Sanchez. *Real-time voxel-based visual hull reconstruction*. 2012. 18

[135] T. Pfaff, R. Narain, J. M. de Joya, and J. F. O'Brien. Adaptive tearing and cracking of thin sheets. *ACM Transactions on Graphics*, 2014. 119

[136] L. Pishchulin, S. Wuhrer, T. Helten, C. Theobalt, and B. Schiele. Building statistical shape spaces for 3D human modeling. 2015. 10, 91

[137] R. Plankers and P. Fua. Articulated soft objects for multiview shape and motion capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2003. 9, 22, 23

[138] T. Popa, Q. Zhou, D. Bradley, V. Kraevoy, H. Fu, A. Sheffer, and W. Heidrich. Wrinkling captured garments using space-time data-driven deformation. *Computer Graphics Forum (Proc. Eurographics)*, 2009. 67

[139] S. Prince. Computer vision: models, learning, and inference, 2012. 49

[140] I. Research. Opencv: Camera calibration and 3d reconstruction, 1999. 13

[141] H. Rhodin, C. Richardt, D. Casas, E. Insafutdinov, M. Shafiei, H.-P. Seidel, B. Schiele, and C. Theobalt. EgoCap: Egocentric Marker-less Motion Capture with Two Fisheye Cameras. In *SIGGRAPH Asia*, 2016. 37

[142] H. Rhodin, N. Robertini, D. Casas, C. Richardt, H.-P. Seidel, and C. Theobalt. General automatic human shape and motion capture using volumetric contour cues. In *Proceedings of the 2016 European Conference on Computer Vision (ECCV 2016)*, 2016. 5, 8, 28, 39, 40, 59, 91, 114, 116, 117

[143] H. Rhodin, N. Robertini, C. Richardt, H.-P. Seidel, and C. Theobalt. A versatile scene model with differentiable visibility applied to generative pose estimation. In *Proceedings of the 2015 International Conference on Computer Vision (ICCV 2015)*, 2015. XVII, 8, 9, 21, 29, 34, 72

[144] N. Robertini, F. Bernard, W. Xu, , and C. Theobalt. Illumination-invariant robust multiview 3d human motion capture. In *Proceedings of the Winter Conference on Applications of Computer Vision, WACV 2018*, 2018. 5, 6, 8, 37, 114

[145] N. Robertini, D. Casas, E. D. Aguiar, and C. Theobalt. Multi-view performance capture of surface details. *International Journal of Computer Vision*, 124(1):96–113, 8 2017. 6, 8, 114, 115

[146] N. Robertini, D. Casas, H. Rhodin, H.-P. Seidel, and C. Theobalt. Model-based outdoor performance capture. In *Proceedings of the 2016 International Conference on 3D Vision (3DV 2016)*, 2016. 5, 6, 8, 114, 115

[147] N. Robertini, E. de Aguiar, T. Helten, and C. Theobalt. Efficient multi-view performance capture of fine-scale surface detail. 2014. 6, 8, 73, 114, 115

[148] G. Rogez, J. Rihan, S. Ramalingam, C. Orrite, and P. H. S. Torr. Randomized trees for human pose detection. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 39

[149] D. Rohmer, S. Hahmann, and M.-P. Cani. Exact volume preserving skinning with shape control. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '09, 2009. 26

[150] J. Salvi, J. Pages, and J. Batlle. Pattern codification strategies in structured light systems. *PATTERN RECOGNITION*, 2004. 15

[151] P. Sand, L. McMillan, and J. Popović. Continuous capture of skin deformation. *ACM TOG*, 2003. 68

[152] Y. Sasaki. The truth of the f-measure. 2007. 106

[153] Y. Y. Schechner and N. Kiryati. Depth from defocus vs. stereo: How different really are they? *International Journal of Computer Vision*, 2000. 16

[154] C. Schmaltz, B. Rosenhahn, T. Brox, and J. Weickert. Region-based pose tracking with occlusions using 3D models. *Mach. Vision Appl.*, 2012. 3, 37, 39

[155] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, CVPR '06, 2006. 16, 117, 120

[156] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *ICCV*, 2003. 39

[157] J. Shelton and J. Orr. *Optical Measurement Methods in Biomechanics*. Springer, 1997. 1

[158] Z. Shu-Jun and W. Wei. Optimized volumetric visual hull reconstruction method based on cuda. In *International Conference on Audio Language and Image Processing*, 2010. 18

[159] L. Sigal, A. Balan, and M. J. Black. Combined discriminative and generative articulated pose and non-rigid shape estimation. In *Advances in Neural Information Processing Systems (NIPS)*, 2007. 10, 117

[160] L. Sigal, M. Isard, H. Haussecker, and M. J. Black. Loose-limbed people: Estimating 3D human pose and motion using non-parametric belief propagation. *International Journal of Computer Vision*, 2012. 40

[161] P.-P. J. Sloan, C. F. Rose, III, and M. F. Cohen. Shape by example. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, I3D '01, 2001. 26

[162] J. Starck and A. Hilton. Surface capture for performance based animation. *IEEE Computer Graphics and Applications*, 2007. 3, 11, 16, 65, 66, 67, 76, 77, 81, 89, 117, 119

[163] C. Stoll, J. Gall, E. de Aguiar, S. Thrun, and C. Theobalt. Video-based reconstruction of animatable human characters. *ACM TOG (Proc. SIGGRAPH Asia)*, 2010. 91

[164] C. Stoll, N. Hasler, J. Gall, H. P. Seidel, and C. Theobalt. Fast articulated motion tracking using a sums of Gaussians body model. In *ICCV*, 2011. XVII, 3, 9, 10, 21, 23, 26, 29, 30, 33, 35, 37, 39, 108

[165] M. Straka, S. Hauswiesner, M. Rüther, and H. Bischof. Simultaneous shape and pose adaption of articulated models using linear optimization. In *Computer Vision–ECCV 2012*. Springer Berlin Heidelberg, 2012. 91

[166] S. Stream. Ss20 3d body scanner, 2017. 14

[167] R. Szeliski and P. Golland. Stereo matching with transparency and matting. In *ICCV*, 1998. 91

[168] R. Szeliski and D. Tonnesen. Surface modeling with oriented particle systems. In *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*, 1992. 119

[169] A. Tagliasacchi, H. Zhang, and D. Cohen-Or. Curve skeleton extraction from incomplete point cloud. SIGGRAPH '09, 2009. 20

[170] A. Taneja, L. Ballan, and M. Pollefeys. Modeling dynamic scenes recorded with freely moving cameras. In *Asian Conference on Computer Vision*, 2010. 89, 91

[171] B. Tekin, I. Katircioglu, M. Salzmann, V. Lepetit, and P. Fua. Structured Prediction of 3D Human Pose with Deep Neural Networks. In *BMVC*, 2016. 37, 39

[172] J. Teran, E. Sifakis, S. S. Blemker, V. Ng-Thow-Hing, C. Lau, and R. Fedkiw. Creating and simulating skeletal muscle from the visible human data set. *IEEE Transactions on Visualization and Computer Graphics*, 2005. 26

[173] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ICCV '99, 2000. 12

[174] M. Trumble, A. Gilbert, A. Hilton, and J. Collomosse. Deep convolutional networks for marker-less human pose estimation from multiple views. In *CVMP*, 2016. 40

[175] T. Tung, S. Nobuhara, and T. Matsuyama. Complete multi-view reconstruction of dynamic scenes from probabilistic fusion of narrow and wide baseline stereo. In *Proc. IEEE ICCV*, 2009. 67

[176] Twindom. Twinstant mobile scanner, 2016. 14

[177] R. Vaillant, L. Barthe, G. Guennebaud, M.-P. Cani, D. Rohmer, B. Wyvill, O. Gourmel, and M. Paulin. Implicit skinning: Real-time skin deformation with contact modeling. *ACM Trans. Graph.*, 2013. 26

[178] O. Veksler. *Efficient Graph-based Energy Minimization Methods in Computer Vision*. PhD Thesis, 1999. 51

[179] D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated mesh animation from multi-view silhouettes. *ACM TOG (Proc. SIGGRAPH '08)*, 2008. 65, 67, 89, 90, 119

[180] D. Vlasic, P. Peers, I. Baran, P. Debevec, J. Popović, S. Rusinkiewicz, and W. Matusik. Dynamic shape capture using multi-view photometric stereo. In *ACM TOG (Proc. SIGGRAPH Asia '09)*, 2009. 3, 67, 89

[181] L. Wade and R. E. Parent. Automated generation of control skeletons for use in animation. *Vis. Comput.*, 2002. 27

[182] R. Y. Wang, K. Pulli, and J. Popović. Real-time enveloping with rotational regression. *ACM Trans. Graph.*, 2007. 27

[183] T. Wang, J. Collomosse, and A. Hilton. Wide baseline multi-view video matting using a hybrid markov random field. In *ICPR*, 2014. 91

[184] M. Waschbüsch, S. Würmlin, D. Cotting, F. Sadlo, and M. Gross. Scalable 3D video of dynamic scenes. In *Proc. Pacific Graphics*, 2005. 67

[185] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*. IEEE Computer Society, 2016. 53

[186] E. Weiszfeld. Sur le point pour lequel la somme des distances de n points donnÃ©s est minimum. *Tohoku Mathematical Journal, First Series*, 1937. 46

[187] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in computational Mathematics*, 1995. 32

[188] C. Wojtan, N. Thürey, M. Gross, and G. Turk. Deforming meshes that split and merge. In *ACM SIGGRAPH 2009 Papers*, SIGGRAPH '09, 2009. 119

[189] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997. 21

[190] C. Wu, Y. Liu, Q. Dai, and B. Wilburn. Fusing multiview and photometric stereo for 3d reconstruction under uncalibrated illumination. *IEEE TVCG*, 2011. 65, 67

[191] C. Wu, C. Stoll, L. Valgaerts, and C. Theobalt. On-set performance capture of multiple actors with a stereo camera. *ACM TOG (Proc. SIGGRAPh Asia)*, 2013. 67, 91, 120

[192] C. Wu, K. Varanasi, Y. Liu, H.-P. Seidel, and C. Theobalt. Shading-based dynamic shape refinement from multi-view video under general illumination. In *Proc. IEEE ICCV*, 2011. 65, 67, 120

[193] C. Wu, K. Varanasi, and C. Theobalt. Full body performance capture under uncontrolled and varying illumination: A shading-based approach. In *ECCV*, 2012. 39, 67, 120

[194] Z. Wu, S. Song, A. Khosla, X. Tang, and J. Xiao. 3d shapenets for 2.5d object recognition and next-best-view prediction. 2014. 21

[195] G. Xu and Z. Zhang. *Epipolar Geometry in Stereo, Motion, and Object Recognition: A Unified Approach*. Kluwer Academic Publishers Norwell, 1996. 11

[196] L. Xu, C. Lu, Y. Xu, and J. Jia. Image smoothing via L0 gradient minimization. In *SIGGRAPH*, 2011. 41

[197] W. Xu, A. Chatterjee, M. Zollhöfer, H. Rhodin, D. Mehta, H.-P. Seidel, and C. Theobalt. MonoPerfCap: Human Performance Capture from Monocular Video. *arXiv:1708.02136*, 2017. 37

[198] G. Ye, Y. Liu, N. Hasler, X. Ji, Q. Dai, and C. Theobalt. Performance Capture of Interacting Characters with Handheld Kinects. In *Proc. ECCV*, 2012. 91, 116

[199] S. Ye, S.-P. Lu, and A. Munteanu. Color correction for large-baseline multiview video. *Image Commun.*, 2017. 14

[200] A. Zaharescu, E. Boyer, and R. Horaud. Topology-adaptive mesh deformation for surface evolution, morphing, and multiview reconstruction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2011. 119

[201] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999. 12

[202] F. Zhou and F. De La Torre. Spatio-temporal matching for human detection in video. In *ECCV*, 2014. 39

[203] X. Zhou, S. Leonardos, X. Hu, and K. Daniilidis. 3D shape estimation from 2D landmarks: A convex relaxation approach. In *CVPR*, 2015. 39

[204] X. Zhou, X. Sun, W. Zhang, S. Liang, and Y. Wei. Deep kinematic pose regression. In *ECCV*, 2016. 37, 39

[205] X. Zhou, M. Zhu, S. Leonardos, K. G. Derpanis, and K. Daniilidis. Sparseness meets deepness: 3D human pose estimation from monocular video. In *CVPR*, 2016. 39

[206] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. *ACM Trans. Graph.*, 2004. 67

[207] M. Zollhöfer, M. Nießner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, C. Loop, C. Theobalt, and M. Stamminger. Real-time non-rigid reconstruction using an RGB-D camera. *ACM Transactions on Graphics*, 2014. 91