Recovery of the 3D Virtual Human: Monocular Estimation of 3D Shape and Pose with Data Driven Priors

Author(s):
Dibra, Endri

Publication Date:
2018

Permanent Link:
https://doi.org/10.3929/ethz-b-000266852 →

Diss. ETH No. 25131

Recovery of the 3D Virtual Human: Monocular Estimation of 3D Shape and Pose with Data Driven Priors

A dissertation submitted to

ETH Zurich

for the Degree of

Doctor of Sciences
(Dr. sc. ETH Zurich)

presented by

Endri Dibra
MSc in Robotics, Systems and Control, ETH Zurich, Switzerland
born April 19, 1988
citizen of Albania

accepted on the recommendation of

Prof. Dr. Markus Gross, examiner
Prof. Dr. Michael Black, co-examiner
Prof. Dr. Otmar Hilliges, co-examiner

2018

Abstract

The virtual world is increasingly merging with the real one. Consequently a proper human representation in the virtual world is becoming more important as well. Despite recent technological advances in making the virtual human presence more realistic, we are still far from having a fully immersive experience in the virtual world, in part due to the lack of proper capturing and modeling of a virtual double. Thus, new methods and techniques are needed to obtain and recover a realistic virtual doppelgänger. This thesis aims to make virtual human representation accessible for every person, by showcasing how it can be obtained under inexpensive minimalistic sensor requirements. Potential fields of application of the findings could be the estimation of body shape from selfies, health monitoring and garment fitting.

In this thesis we investigate the problem of reconstructing the 3D virtual human from monocular imagery, mainly coming from an RGB sensor. Instead of focusing on the full avatar at once, we separately consider three constituting parts of it: the naked body, clothing and the human hand. The preeminent focus is on the estimation of the 3D shape and pose from 2D images, e.g. taken from a smart-phone, making use of data-driven priors in order to alleviate this ill-posed problem. We utilize discriminative methods, with a focus on CNNs, and leverage existing and new realistically rendered synthetic datasets to learn important statistics. In this way, our presented data-driven methods can generalize well and provide accurate reconstructions on unseen real input data. Our research is not only based on single views and annotated groundtruth data for supervised learning, but also shows how to utilize multiple views simultaneously, or leverage from them during training time, in order to boost performance achieved from a single view at inference time. In addition, we demonstrate how to train and refine unsupervised with unlabeled real data, by integrating lightweight differentiable renderers into CNNs.

In the first part of the thesis, we aim to estimate the intrinsic body shape, regardless of the adopted pose. Under assumptions of uniform background colours and poses under minimal self-occlusion, we show three different approaches for estimating the body shape: Firstly, by basing our estimation on handcrafted features in combination with CCA and random forest regressors, secondly by basing it on simple standard CNNs, and thirdly by basing it on more involved CNNs with generative and cross-modal components. We show robustness to pose changes, silhouette noise and state-of-the-art performance on existing datasets, outperforming also optimization based approaches.

The second part of the thesis tackles the estimation of garment shape from one or two images. Two possible estimations of the garment shape are provided: one that gets deformed from a template garment (i.e. from a t-shirt or a dress) and second one that gets deformed from the underlying body. Our analysis includes empirical evidence which shows the advantages and disadvantages of utilizing either of the estimation methods. We adopt lightweight CNNs in combination with a new realistically rendered garment dataset, synthesized under physically correct dynamic assumptions, in order to tackle the very difficult problem of estimating 3D shape from an image. Training purely on synthetic data, we are the first to show that garment shape estimation from real images is possible through CNNs.

The last and concluding part of the thesis focuses on the problem of inferring a 3D hand pose from an RGB or depth image. To this end, our proposal is an end-to-end CNN system that leverages data from our newly proposed realistically rendered hand dataset, consisting of 3 million samples of hands in various poses, orientations, textures and illuminations. Utilizing this dataset in a supervised training setting, helped us not only with pose inference tasks, but also with hand segmentation. We additionally introduce network components based on differentiable renderers that enabled us to train and refine our networks with unlabeled real images in an unsupervised fashion, showing clear improvements. We demonstrate on-par and improved performance over state-of-the-art methods for two input modalities, under various tasks varying from 3D pose estimation to gesture recognition.

Zusammenfassung

Die reale Welt, in der wir leben, verschmilzt immer stärker mit der virtuellen Welt und erfordert eine angemessene menschliche Repräsentation in letzterer. Trotz der jüngsten technologischen Fortschritten, die es uns ermöglichen, die menschliche virtuelle Präsenz realistischer zu gestalten, ohne unser virtuelles "Double" komplett zu vermessen und zu modellieren, sind wir noch weit von vollständig immersive Erlebnisse in der virtuellen Welt entfernt. Es werden neue Methoden und Techniken benötigt, um einen realistischen virtuellen Doppelgänger zu erstellen. Um diese Techniken für jede Person zugänglich zu machen, müssen diese unter kostengünstigen und minimalistischen Sensoranforderungen arbeiten können.

Diese Arbeit untersucht das Problem der Rekonstruktion des virtuellen 3D-Menschen aus monokularen Bildern, die hauptsächlich von RGB-Sensoren stammen. Anstatt sich auf den vollen Avatar auf einmal zu konzentrieren, betrachten wir die folgenden drei Teile getrennt: den nackten Körper, Kleidung und die menschliche Hand. Unser Hauptaugenmerk liegt auf der Ermittlung der dreidimensionalen Form und Haltung von Personen anhand von 2D-Bildern. Diese stammen z.B. von Smartphones und werden gemeinsam mit datengetriebenen Priors verwendet, um dieses unterbestimmte Problem zu lösen. Wir verwenden diskriminierende Methoden mit Fokus auf CNNs und nutzen existierende und neue realistisch gerenderte synthetische Datensätze, um zugrundeliegende Statistiken zu lernen. Auf diese Weise können unsere vorgestellten datengetriebenen Methoden gut verallgemeinern und genaue Rekonstruktionen für ungesehene reale Eingabedaten liefern. Unsere Forschung basiert nicht nur auf einzelnen Ansichten und annotierten Groundtruth-Daten für Supervised Learning, sondern zeigt auch, wie mehrere Ansichten gleichzeitig oder während der Trainingszeit genutzt werden können, um die Resultate aus einer einzigen Ansicht zur Inferenzzeit zu verbessern. Darüber hinaus demonstrieren wir, wie unsupervised mittels nicht annotierter realer Daten trainiert und verfeinert werden kann, indem leichtgewichtige differenzierbare Renderer in CNNs integriert werden.

Im ersten Teil der Arbeit haben wir uns zum Ziel gesetzt, die Körperform, unabhängig von der Haltung des Körpers zu ermitteln. Reale Anwendungen hierfür sind die Bestimmung der Körperform aus Selfies, Gesundheitsüberwachung oder die personalisierte Anpassung von Kleidung. Unter der Annahme einheitlicher

Hintergrundfarben und Körperhaltungen unter minimaler Selbstokklusion wird dieses Problem mit drei verschiedenen Ansätzen angegangen: einer basiert auf handgefertigten Features in Kombination mit CCA und Random Forest Regressors, ein zweiter basiert auf einfachen Standard-CNNs und der dritte basierend auf komplexeren CNNs mit generativen und cross-modalen Komponenten. Wir zeigen Robustheit gegenüber veränderter Körperhaltung, verrauschter Silhouetten und erreichen State of the Art Ergebnisse bei bestehenden Datensätzen und übertreffen dabei auch optimierungsbasierte Methoden.

Der zweite Teil der Arbeit beschäftigt sich mit der Abschätzung der Form von Kleidungsstücken aus einem oder zwei Bildern. Zwei mögliche Vorgehen zur Bestimmung der Kleidungsstückform sind vorgesehen, eine, die von einem Template-Kleidungsstück (eines T-Shirts oder eines Kleides) deformiert wird, und eine, welche vom darunterliegenden Körper deformiert wird. Wir liefern empirische Daten für die Vor- und Nachteile der Verwendung der beiden Modelle. Wir verwenden einfache CNNs in Kombination mit einem neuen realistisch gerenderten Kleidungsstück-Datensatz, der unter physikalisch korrekten dynamischen Annahmen synthetisiert wurde, um dieses schwierige Problem anzugehen. Nach einem Training auf rein synthetischen Daten sind wir, unseres Wissens nach, die ersten, die zeigen, dass die Bestimmung der Kleidungsstückform auch von realen Bildern durch CNNs möglich ist.

Der letzte und abschliende Teil der Arbeit beschäftigt sich mit dem Problem, eine 3D-Handhaltung aus einem RGB- oder Tiefenbild abzuleiten. Zu diesem Zweck ist unser Vorschlag ein End-to-End-CNN-System, das aus unserem neu vorgeschlagenen realistisch gerenderten Hand-Datensatz besteht, der aus 3 Millionen Handmustern in verschiedenen Haltungen, Orientierungen, Texturen und Beleuchtungen besteht. Durch die Verwendung dieses Datensatzes in einer überwachten Trainingsumgebung können nicht nur Inferenzaufgaben, sondern auch Handsegmentierungen durchgeführt werden. Darüber hinaus stellen wir Netzwerkkomponenten auf Basis differenzierbarer Renderer vor, die es uns ermöglichen, unsere Netzwerke unsupervised, ohne annotierte reale Bilder zu trainieren und zu verfeinern. Wir demonstrieren eine gleichwertige und verbesserte Leistung gegenüber Methoden nach dem aktuellen Stand der Technik für zwei Eingabemodalitäten bei verschiedenen Aufgaben, die von der 3D-Haltungs-Schätzung bis zur Gestenerkennung reichen.

Acknowledgments

Starting, pursuing and completing an academic path is due to a weighted mixture between opportunity, inspiration, possibility and support. All these factors were important throughout my academic life and different people have played different roles in this equation. I would initially like to express my immense gratitude to my parents, sister and grandparents, for contributing to all of the above factors. Thank you for making me who I am and giving me your unconditional support in all the decisions I have taken so far.

I would like to sincerely thank my thesis advisor, Prof. Markus Gross, for giving me the opportunity to be part of the Computer Graphics Lab and explore this very exciting field, allowing me to pursue my own research ideas. I would also like to thank him for convincing me to come back to academia, after having started my career in industry.

I am also thankful to Prof. Michael Black and Prof. Otmar Hilliges, not only for refereeing and examining my dissertation, but also for their inspiration through their work and thought provoking conversations I was delighted to be a part of during my PhD. It was due to some of their very best works that I was pushed to think out of the box, and produce something hopefully relevant for the community.

I would like to thank Paul Beardsley for being my first academic mentor and preparing me for a PhD during my time at Disney Research Zurich. Thank you for the opportunity and support. My gratitude goes to Stelian Coros and Bernhard Tomasewski for their friendship and putting me in contact with CGL during that game of pool.

My PhD was initially coupled with the Vizrt company, that gave me an opportunity to work on research strongly tied with industrial applications. I would like to thank the whole Vizrt team (Richard, Urs, Christoph, Stephan, Lars, Danni, Yannick, Hadi etc.) for making me feel part of them. Special thanks go to Remo Ziegler for his larger involvement on the research projects and to Jens Puwein for initially guiding me through a PhD in collaboration with an industrial partner.

I would like to thank Cengiz Öztireli for supporting me from the beginning and being my long-lasting collaborator throughout my whole PhD work and Prof.

Yoshihiro Kanamori for the insightful comments and inputs during his time in CGL.

Most of the work and publications would not have been possible without the help of my co-authors and students: Himanshu Jain, Radek Danecek, Thomas Wolf, Silvan Melchior, Ali Balkis, Matej Hamas, collaborators: Prof. Ladislav Kavan, Petr Kadlecek and very helpful discussions with Martin Oswald, Johannes Schönberger, Torsten Sattler, Nikolay Savinov, Lubor Ladicky, Tobias Günther, Vinicius Azevedo, Brian McWilliams, Alex Wan-Chun, Andrea Tagliasacchi.

Special thanks go to Thorsten Steinbock for supporting me in my lab job as an IT Coordinator, but also for helping all of us with cluster support during these GPU intensive CNN times.

I would like to thank my sister Kejda and my friend Simone Meyer and Sunnie Groeneveld for their comments on the thesis.

I consider myself very lucky to have spent these amazing years around brilliant researchers, friends and colleagues from the Institute of Visual Computing and Disney Research Zurich. I would like to mention all of them but I hope in forgiveness if anyone is missing: Simone Meyer, Riccardo Roveri, Christian Schumacher, Vinicius Azevedo, Tobias Günther, Byungsoo Kim, Pelin Dogan, Rafael Wampler, Yeara Kozlov, Virag Varga, Marco Ancona, Severin Klingler, Niko Ben Huber, Irene Baeza, Pascal Berard, Christian Schüller, Oliver Glauser, Olga Diamanti, Michael Rabinovic, Olga Sorkine-Hornung, Kaan Yücer, Vittorio Megaro, Federico Perazzi, Martina Häfeli, Leo Helminger, Alexandre Chapiro, Loic Ciccone, Thomas Müller, Ivan Ovinnikov, Romain Prevost, Claudia Plüss, Barbara Solenthaler, Markus Portmann, Danielle Luterbacher, Ian Cherabier, Yagiz Aksoy, Thabo Beeler, Roland Möerzinger, Marcel Lancelle, Derek Bradley, Amit Bermano, Bernd Bickel, Jan Novak, Maurizio Nitti, Alessia Marra, Robert Sumner, Fabio Zünd and Antoine Milliez. Big up goes to the Secaf chat for the super funny videos and posts.

Lastly, I would like to thank all of my friends from Zurich, Bremen and Albania who made my time even more enjoyable during my PhD.

Contents

List of Figures

List of Algorithms

List of Tables

CHAPTER 1

Introduction

Humans are constantly trying to bring the virtual world as close as possible to the real one. The most recent technological advances of the last decades have played an important role in bridging the gap between these two very related and forked concepts about the world we live in. As a matter of fact this is true. Since the invention of the internet, a lot of advances and achievements have been seen in the online world e.g. electronic retailing, ease of acquiring information, voice assistance, fast queries about questions and needs and most importantly an easy communication medium that virtually brings people together without the need to physically travel. Initial communication attempts were achieved through e-mail or instant messaging, and more recently through video conferencing and virtual meeting rooms with cartoon-like avatars. The most natural way for humans to communicate with each other though, are through close face-to-face physical interactions, and the above-mentioned approaches lack such emotional presence or immersion, due to the current inability to properly represent the virtual human body.

Despite recent promising attempts and achievements to properly model, capture and place the human in the virtual world, in its current state, the latter can be more described as a disembodied one, where humans are parted from their bodies. With bodies, we do not only mean the naked body as a whole, but also the clothes and garments that typically cover it and its parts. With respect to the body parts, we distinguish the hands, due to the high importance they play in mutual communication and manipulating the world. Thus, there is an imminent need for the presence of virtual humans or avatar copies of one-self to make such an immersive experience richer.

The more recent fields of virtual, augmented and mixed reality have seen impressive advances in creating plausible digital human avatars, however we are yet far from a compelling realism, and the more realistically looking ones are typically obtained from expensive scanners, extensive and time-costly manual labor or a multitude of synchronized sensors (typically in the form of RGB, Range and Depth cameras). For practical applications, it is essential to have an automatic and interactive system, that can work with sensor input acquired from less restrictive conditions, e.g. RGB images from cheaper and fewer sensors. An example of such sensor is a smartphone camera, which could be metaphorically thought of as a portal between the real (image) and virtual world (reconstructed human avatar).

In this thesis, we break down the 3D virtual human into three of its constituent parts, namely the naked body, the garment fitting on it and one of the body parts - the human hand. Consecutively, we present several methods, as in Figure 1.1, on how to recover either the intrinsic 3D shape, pose or both from monocular 2D imagery depicting the aforementioned body parts. This is achieved with the help of statistical priors learned from datasets of virtual or real humans.

Once our digital double is obtained, there exist a myriad of applications where it can be used. From a holistic viewpoint, games would benefit a lot from personalized 3D human avatars to increase the realism and enrich the players' experience, as it is also the case with VR immersion [1]. Free view-point video, e.g. applied to sports, would leverage from 3D avatars of players for the tools utilized to analyze and showcase instant replays from intricate angles [2]. Tele-presence or Holoportation [3] are very important in face-to-face communications. AR applications, are not only limited to instant avatars consisting of the face and hair [4], or hands in free-form interaction with virtual objects [Hilliges et al., 2012], but can also benefit from the full body shape and pose, as in the case of Amazon's recently acquired company, *Body Labs*. Visual effects and animations in film productions would leverage from ready made actor avatars, due to the difficulty of automating many steps in this process, which typically require extensive amounts of man-hours. One such example is the reincarnation of Paul Walker in the movie Fast and Furious 7.

If one focuses on its constituent parts, e.g. the reconstruction of the naked body shape, the variety of applications where a recovered shape can be

[1] https://www.oculusconnect.com/

[2] www.vizrt.com

[3] https://www.microsoft.com/en-us/research/project/holoportation-3/

[4] https://www.pinscreen.com/

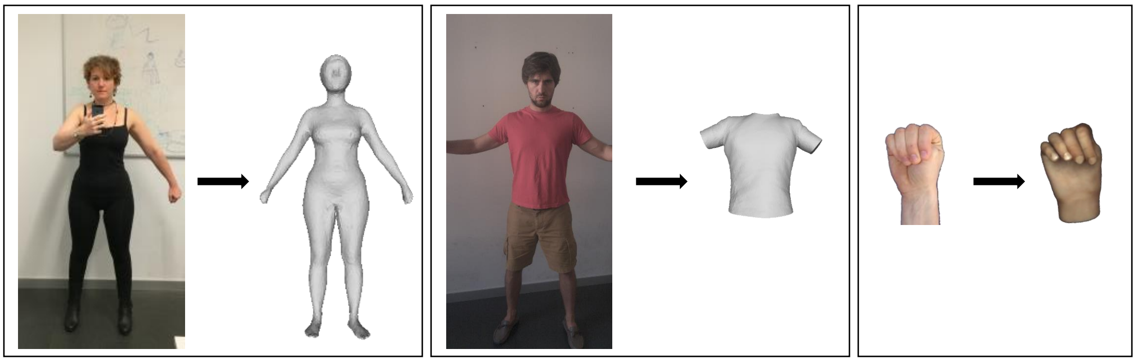


Figure 1.1: *We learn a mapping from image pixels to 3D meshes that come in the form of human body shape and pose (Chapter 3), garment shape or pose (Chapter 4) and hand pose (Chapter 5).*

utilized expands even further. It finds applications ranging from security (e.g. surveillance and biometric authentication) and the medical field (e.g. body health monitoring due to visual cues [Piryankova et al., 2014; Mölbert et al., 2017; Fleming et al., 2017] or automatic estimation of personal measurements) to ergonomics, image retouching, and clothing retail [5]. Clothing is an important part of virtual human modeling too. Capturing and modeling garments are fundamental steps for applications ranging from online retail and virtual try-on to virtual character and avatar creation. Last but not least, knowledge of a hand pose, not only facilitates communication between (digital) humans, it also enables the recognition and automatic translation of hand gestures into meaning. This is very relevant for the growing research and applied field of human computer interaction (HCI).

1.1 Stating the Technical Problem

Retrieving 3D (human) information from 2D (image) observations is a long-known, very challenging and ill-posed task. Many 3D objects can explain the same observation, e.g. due to variations in shape and pose. Since in this thesis we focus on humans, we distinguish between an intrinsic shape and pose, especially in the case of the naked human body or hand. A pose is defined in terms of transformation matrices (rotation and translation) of the limbs or fingers, e.g. running or a hand gesture, while an intrinsic shape captures changes in the body or hand that are independent of pose changes

[5] https://www.fision-technologies.com/

(e.g. thickness, height, waist circumference, finger breadth etc.). Delving deeper into the naked human body, there exist general (soft-tissue) deformations of the shape due to pose changes and dynamics, however for simplicity, throughout this thesis, we decouple human pose and shape from each other and we discard dynamics, similarly to many previous works. For garments on the other hand, we couple the shape and pose into one general deformation, which is affected from dynamics.

The problem of estimating 3D shape and pose from 2D images has been investigated and various solutions have been proposed (Chapter 2). These are mainly based on iterative processes that tend to minimize discrepancies between synthesized observations stemming from 3D human reconstructions to the very real-world observations that they try to explain. Human bodies and limbs, in the real world, are controlled by a large number of degrees of freedom (DOF), hence trying to solve for all of them simultaneously becomes almost infeasible. In order to alleviate the problem, researchers represent humans through parametric models, by formulating generative models of the human that have low degrees of freedom, which we call parameters throughout the thesis. There has been a decade of research on how to learn such parametric models, however the principle is as follows: A template triangular mesh is fit to a large number of real scans [Robinette and Daanen, 1999], by putting them into correspondence through co-registration e.g. [Hirshberg et al., 2012]. Once the meshes are in correspondence, one could compute the vertex or triangle [Anguelov et al., 2005] deformations of each fitted sample from a template mean mesh and apply dimensionality reduction techniques such as the Principal Component Analysis (PCA) to obtain a low dimensional parametric model. One of the most compact parametric models based on this principle is that of [Loper et al., 2015], however semantically meaningful body models exist too [6]. Under such paradigm, throughout this work, we represent shapes and poses as deformations from a template model, as it can be seen in Figure 1.2.

Estimating 3D shape and pose from 2D images boils down to estimating model parameters such that the model output is similar to real-world observations (e.g. the silhouette of an estimated human body should match the mask of the person depicted in a picture). Generative or optimization based methods have tackled this problem traditionally, with the advantage of achieving low reconstruction errors, however on the expense of high running times. Another group of works, a discriminative one, that heavily relies on training data, has been used to approach this problem too. These methods are fast, however they achieve lower inference accuracy as com-

[6] http://www.makehuman.org/

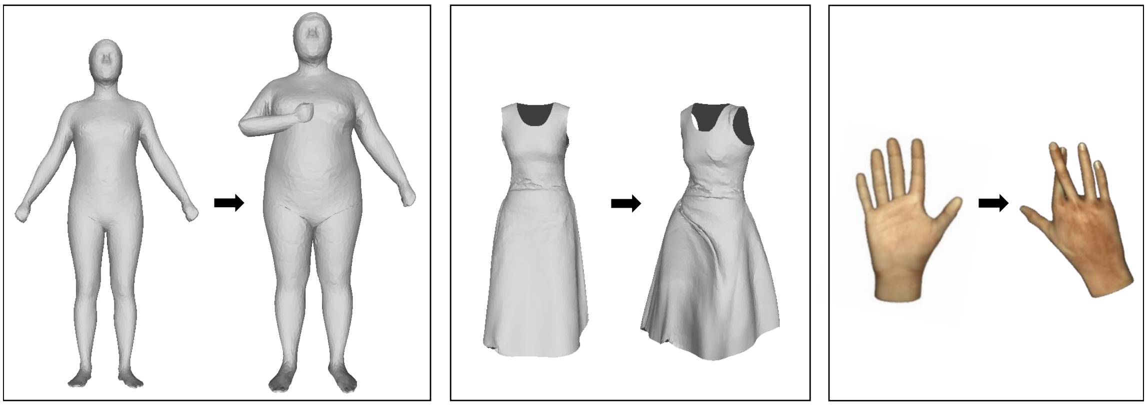


Figure 1.2: *Demonstration of training samples (right) generated from a template mesh (left) by changing the human body shape and pose (Chapter 3), garment shape or pose (Chapter 4) and hand pose (Chapter 5).*

pared to their generative counterparts. These methods have typically relied on handcrafted features, however with the advent of Convoultional Neural Networks (CNNs) a new opportunity arose for these data hungry methods to climb the throne. This was due to better machines, more available real training data and an increase in realism of synthetically generated data. In this thesis, we attempt to show that with a correct training and carefully generated synthetic data, it is possible to achieve reconstructions on par or even better than those from generative methods, while preserving the speed of discriminative methods. In order to do so, we offload the computations at inference time, by loading knowledge coming from different modalities and views (Chapter 3) at training time and adding components based on differentiable rendering (Chapter 5).

Supervised discriminative methods are based on annotated groundtruth datasets in order to be trained properly. Obtaining 3D annotations from 2D images is a very tedious process, especially when shapes and poses of bodies, garments and hands are considered, due to occlusions, articulations etc. In order to overcome this challenge, utilizing generative processes, we generate realistically looking rendered synthetic datasets. This rendering process, which can be thought of as the inverse of what we are trying to achieve in this work, maps parameters such as pose, shape, texture and dynamics along with lighting and camera parameters into pixels. In this way, annotated groundtruth data between images and parameters can be easily obtained.

There is a discrepancy though between synthetic and real data. Utilizing only synthetically generated data and training solely based on those, can result in systems learning statistics from the data which are not representative of the real ones. Despite the fact that our data is generated from a statistically-learned human body model, and our predictions always fall within the natural space of human bodies (or clothes, hands) the accuracy can be tremendously improved if real training data is infused into the system. Hence, we explore ways of incorporating real-world unlabeled data in a semi-supervised and unsupervised fashion, in order to improve our predictions. Additionally, it is clear that having more than one view (image) representing the same object should improve predictions. We present methods of leveraging from multiple views at training time, in order to boost inference from a single view at test time.

A schematic description of the general pipeline for obtaining the 3D virtual human, with respect to this thesis, is depicted in Figure 1.3. The human avatar constitutes of its various parts: the naked human body, the garment and going in more detail, the hand. In chronological order, we train for and aim to obtain intrinsic body shape, garment shape and hand pose parameters. Through a forward mapping between these parameters and images, we generate training data. With the help of our methods based on such training data and presented in the following chapters, we obtain a mapping from real world monocular images to these parameters, which in turn are utilized to reconstruct the 3D virtual human.

1.2 Principal Contributions

In terms of the bigger picture, this thesis aims to advance the field of reconstructing and tracking of the human avatar utilizing the least amount of sensors. Tackling this all at once is a very hard problem, hence, as previously mentioned, we focus on and contribute to three crucial and representative parts that constitute the virtual human, namely the naked body shape, the garment shape and the hand pose. Our goal is to provide solutions for practical applications that utilize RGB monocular cameras and run at interactive rates, in order to make them tangible and integrate them in today‘s smartphones. In order to achieve this, we contribute to the community threefold: (a) by introducing several discriminative methods, largely based on CNN-s, that have lower run-times than but perform on par with and even better than acknowledged optimization based methods, (b) by providing realistically looking synthetically generated datasets that are crucial for the training of fully supervised discriminative based methods, due to the scarcity of annotated real datasets for the problems that we tackle and (c) by presenting ways of utilizing multiple modes and views during training, in a supervised

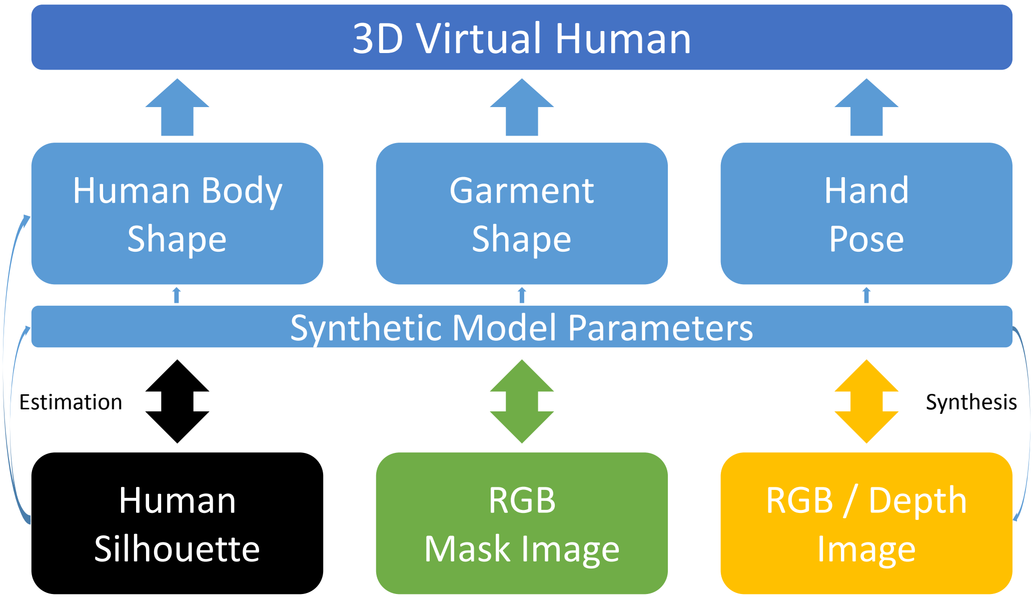


Figure 1.3: *A 3D virtual human is obtained through a combination of its various parts, namely the human body shape and pose, the garment and going in more detail, the hand pose. These, in turn, are obtained by estimating parameters of parametric mesh models from real world images. In order to train this mapping, synthetic images are generated through a forward rendering and synthesis process that starts from these parameters and obtains the image in a silhouette form (Chapter 3), RGB mask (Chapter 4) and RGB or Depth mask (Chapter 5).*

or un-supervised (with unlabeled data) fashion, in order to boost predictions achieved during inference or testing time.

Below, we list the main contributions of the work presented in this thesis. More specifically:

- We introduce a fast and automatic system for human body shape estimation from monocular silhouettes/s under no fixed pose or known camera assumptions, thanks to novel features that capture robust global and local information simultaneously. We further demonstrate how Canonical Correlation Analysis (CCA) for multi-view learning combined with Regression Forests can be applied to the task of shape estimation, leveraging synthetic data at training time and improving prediction at test time as compared to training random forests with raw feature data. Extensive validation on thousands of body shapes are provided via thorough comparisons to state-of-the-art methods on synthetic meshes generated by fitting meshes to real human scans.

- We present another system for human shape and body measurements estimation, from silhouettes (or shaded images) of people in garment fitting like poses, by learning a global mapping to shape parameters. To the best of our knowledge, this is the first system that utilizes CNN-s to accurately reconstruct human body shapes from images. Thus, we show how to train from scratch an end-to-end fully supervised regression from CNNs with binary silhouette images as input, and demonstrate how to incorporate more evidence (e.g. a second view) in order to improve prediction.

- Building up on the above, we introduce a novel neural network architecture for 3D body shape estimation from silhouettes consisting of three main components, (a) a generative component that can invert a pose-invariant 3D shape descriptor to reconstruct its neutral shape, (b) a predictive component that combines 2D and 3D cues to map silhouettes to human body shapes, (c) a cross-modal component that leverages multi-view information to boost single view predictions. This combination achieves state-of-the-art performance for human body shape estimation that significantly improves accuracy as compared to existing methods.

- We provide an end-to-end 3D garment shape estimation algorithm. The algorithm automatically extracts 3D shape from a single image captured with an uncontrolled setup, that depicts a dynamic state of a garment at interactive rates. To the best of our knowledge, we introduce the first regressor system based on convolutional neural networks (CNN-s) combined with statistical priors and a specialized loss function for garment shape estimation. In order to enable its training, we provide a new realistically rendered physically based synthetic garment dataset for a shirt and dress case. We further validate our approach by presenting experiments with several architectures, including those for single and multi-view setups.

- We demonstrate how to train or refine a CNN-based 3D hand pose estimation architecture, on unseen and unlabeled depth images, avoiding the need for annotated real data. This is achieved due to a new training pipeline that can accurately estimate 3D hand pose with the ability to refine itself on unlabeled depth images, using a depth loss component with a physical and collision regularizer. The advantage of utilizing such a method, to enhance estimations of simple candidate CNN models, is demonstrated through extensive evaluations and comparisons to state-of-the-art methods.

- Lastly, based on the approach from above that expects a depth image as an input, we introduce a complete system for 3D hand pose estimation and gesture recognition from monocular RGB data. We show how refining of an RGB-based network trained on synthetic data is achieved with unlabeled RGB hand images and the corresponding depth maps. In order to achieve this, we initially depend on a new realistically rendered hand dataset with 3D annotations that we provide. This dataset helps both hand segmentation and 3D pose inference. We validate our method on available datasets showing superior performance to related works for three different pose inference tasks.

1.3 Thesis outline

The remainder of this thesis is organized as follows:

- **Chapter 2** gives a general overview of previous methods, focusing on estimating the virtual human and in more detail on body shape, garment shape and hand pose estimation from monocular imagery.
- **Chapter 3** introduces three methods that can estimate the human body shape from monocular binary silhouette images. These methods make use of Random Forest Regressors with specialized features and Convolutional Neural Networks along with supervised and cross-modal learning to map images to meshes that represent the human body.
- **Chapter 4** describes a CNN based method that can map garment RGB images to garment shapes with the help of a proposed physically simulated synthetic garment dataset.
- **Chapter 5** introduces two discriminative methods for hand pose estimation from either depth or monocular RGB images, that can be trained unsupervised from unseen and unlabeled real data, due to a specialized differentiable renderer loss during training.
- **Chapter 6** concludes this thesis with a discussion of the contributions and a more elaborated outlook of potential methods that could complement and improve the ones introduced here.

1.4 Publications

In the context of this thesis, the technical contributions have led to top-tier peer-reviewed conference publications:

- **E. Dibra**, C. Öztireli, R.Ziegler and M. Gross (2016). Shape from Selfies: Human Body Shape Estimation Using CCA Regression Forests. *Proceedings of the 14th European Conference of Computer Vision (ECCV), Amsterdam, The Netherlands, October 11-14, 2016* (Chapter 1).
- **E. Dibra**, H. Jain, C. Öztireli, R.Ziegler and M. Gross (2016). HS-Nets: Estimating Human Body Shape from Silhouettes with Convolutional Neural Networks. *Fourth International Conference on 3D Vision (3DV), Stanford, CA, USA, October 25-28, 2016* (Chapter 1).
- **E. Dibra**, H. Jain, C. Öztireli, R.Ziegler and M. Gross (2017). Human Shape from Silhouettes Using Generative HKS Descriptors and Cross-Modal Neural Networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, July 21-26, 2017* (**Spotlight**) (Chapter 1).
- R. Danecek*, **E. Dibra***, C. Öztireli, R.Ziegler and M. Gross (2017). DeepGarment : 3D Garment Shape Estimation from a Single Image. *Computer Graphics Forum (Eurographics), Lyon, France, April 24-28, 2017* (Chapter 2).
- **E. Dibra***, T. Wolf*, C. Öztireli and M. Gross (2017). How to Refine 3D Hand Pose Estimation from Unlabelled Depth Data ? *Fifth International Conference on 3D Vision (3DV), Qingdao, China, October 10-12, 2017* (Chapter 3).
- **E. Dibra**, S. Melchior, A. Balkis, T. Wolf, C. Öztireli and M. Gross (2018). Monocular RGB Hand Pose Inference from Unsupervised Refinable Nets. *CVPR 1st International Workshop on Human Pose, Motion, Activities and Shape in 3D (3D Humans 2018), Salt Lake City, UT, USA, June 18-22, 2018* (Chapter 3).

Additional implementation and evaluation details not present in the above papers are included in this thesis, in addition to their contents. Although not directly linked to the work presented here, the following peer-reviewed publications were published during my PhD studies:

- **E. Dibra**, J. Maye, O. Diamanti, R. Siegwart and P.A. Beardsley (2015). Extending the Performance of Human Classifiers Using a

Viewpoint Specific Approach. *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloha, HI, USA, January 5-9, 2015.*

CHAPTER 2

Related Work

In the recent years, there has been a lot of progress from the Computer Vision and Graphics community in producing a myriad of excellent methods that attempt to estimate the virtual 3D human. The human body is very complex, and capturing it in 3D, without the need of expensive capturing equipment, but through low-cost 2D sensors is a very relevant research task. Researchers, thus, have looked at this problem from various perspectives and tackled it from various angles. Some have focused on the static surface representation of the body as a whole [Balan et al., 2007; Bălan and Black, 2008; Guan et al., 2009; Zhou et al., 2010; Jain et al., 2010; Hasler et al., 2010; Chen et al., 2013; Rhodin et al., 2016; Bogo et al., 2016a] utilizing simplified body models [Anguelov et al., 2005; Hasler et al., 2009; Neophytou and Hilton, 2013; Loper et al., 2015], others have attempted to explore dynamics [Pons-Moll et al., 2015] and anatomically correct body models [Kadlecek et al., 2016]. In parallel to works focusing on the human body in its naked form, or under assumptions of tight clothing, several methods that tackle looser garment capturing [Hahn et al., 2014; Pons-Moll et al., 2017; Zhang et al., 2017; Yang et al., 2016; Jeong et al., 2015; Zhou et al., 2013; Guan et al., 2012; Bradley et al., 2008] under static or dynamic assumptions have been presented, as humans are typically seen in clothing apparel, for most everyday scenarios. Last but not least, due to the need of covering details that are typically not captured when the human body is considered as a whole, there has been quite some work recently that has focused on the capturing and estimation of smaller and more targeted body parts, such as hands [Zimmermann and Brox, 2017; Spurr et al., 2018; Tagliasacchi et al., 2015; Oikonomidis et al., 2011; Tompson et al., 2014], faces [Cao et al., 2017; Beeler et al., 2010; Kim et al., 2017; Tewari et al., 2017], sometimes going even deeper into eyes [Bérard et al., 2016; Bérard et al., 2014; Sugano et al., 2014; Zhang et al., 2015] and hair [Hu et al., 2015], with potential combinations of some of those components [Hu et al., 2017].

With respect to the general attributes of a virtual human, previous methods have focused on shape [Balan et al., 2007; Bălan and Black, 2008; Guan et al., 2009; Zhou et al., 2010; Jain et al., 2010; Hasler et al., 2010; Chen et al., 2013], pose [Song et al., 2017; Mehta et al., 2016; Mehta et al., 2017], texture [de Aguiar et al., 2008; Eisemann et al., 2008] separately and in combination with each other [de Aguiar et al., 2008; Stoll et al., 2010; Gall et al., 2009; Carranza et al., 2003]. The majority of works focusing on pose have tackled the human body [Mehta et al., 2016; Mehta et al., 2017; Song et al., 2017], and more recently the human hand [Zimmermann and Brox, 2017; Spurr et al., 2018; Panteleris et al., 2017; Song et al., 2015; Mueller et al., 2017]. With respect to the shape, the human body as a whole has had quite some attention, however less than its pose counterpart, mainly due to data modeling, the limited availability of shape datasets and difficulty of capturing.

In this chapter, we focus on three sub-parts in more detail, namely the Human Body Shape (Section 2.1), Garment Shape (Section 2.2) and Hand Pose (Section 2.3). For each part, we touch on the works adopted prior to and during the deep learning latest advent era, mainly focusing on 2D Monocular RGB Images as input, but also covering depth, multi-view and video based works.

2.1 Body Shape Estimation Methods

General Methods for Shape Estimation. It is an ill-posed problem to estimate the 3D geometry of a human body from 2D imagery. Early methods used simplifying assumptions such as the visual hull [Laurentini, 1994] by considering multiple views or simple body models with geometric primitives [Delamarre and Faugeras, 1999; Kakadiaris and Metaxas, 1998; Mikic et al., 2003]. Although these work well for coarse pose and shape approximations, an accurate shape estimation cannot be obtained.

Human Body Shape Statistical Priors. As scanning of a multitude of people in various poses and shapes was made possible [Robinette and Daanen, 1999], more complete, parametric human body shape models were learned [Anguelov et al., 2005; Hasler et al., 2009; Neophytou and Hilton, 2013; Loper et al., 2015] that capture deformations due to shape and pose. Instead of assuming general geometry, human body shape model based methods started to rely on the limited degrees of freedom for the possible body shapes. Utilizing such a prior allows us to always stay within the space of realistic body shapes, and reduces the problem of estimating the parameters of the model. Such models can also be combined with articulation models to simultaneously represent pose as joint angles or transformations, and shape with parameters [Anguelov et al., 2005; Hasler et al., 2009; Neophytou and Hilton, 2013; Loper et al., 2015]. In our methods, we combine state-of-the-art 3D body shape databases [Yang et al., 2014; Pishchulin et al., 2015] containing thousands of meshes, and utilize a popular human body shape deformation model based on SCAPE [Anguelov et al., 2005].

Fitting Body Shapes by Silhouette Matching. The effectiveness of parametric models with human priors, gave rise to methods that try to estimate the human body shape from single [Guan et al., 2009; Zhou et al., 2010; Jain et al., 2010; Hasler et al., 2010; Chen et al., 2013; Rhodin et al., 2016] or multiple input images [Balan et al., 2007; Bălan and Black, 2008; Hasler et al., 2010; Rhodin et al., 2016], by estimating the parameters of the model, through matching projected silhouettes of the 3D shapes to extracted image silhouettes by correspondence. Although this leads to accurate matching, despite promising results on deformable 2D shape matching [Schmidt et al., 2007; Schmidt et al., 2009], establishing correspondences between the input and output silhouettes is a very challenging problem especially when the body pose is not known or self occlusions are present. The simultaneous estimation of pose and shape is typically addressed by manual interaction to establish and refine matching or pose estimation [Chen et al., 2013; Zhou et al., 2010; Jain et al., 2010], and under certain assumptions on the error metric, camera calibration and views [Guan et al., 2009; Bălan and Black, 2008; Jain et al., 2010]. [Lahner et al., 2016] aim at automatically finding a correspondence between 2D and 3D deformable objects by casting it as an energy minimization problem, demonstrating good results however tackling only the problem of shape retrieval. A very recent work [Bogo et al., 2016a] attempts at estimating both the 3D pose and shape from a single 2D image with given 2D joints, making use of a 3D shape model based on skinning weights [Loper et al., 2015]. It utilizes a human body prior as a regularizer, for uncommon limb lengths or body inter-penetrations, achieving excellent results on 3D pose estimations, however, lacking accuracy analysis on the generated body shapes.

In contrast to previous methods that directly match silhouettes, we formulate shape estimation from silhouettes as a regression problem where global and semantic information on the silhouettes are incorporated either through handcrafted features (Section 3.3.1) or by utilizing CNNs (Section 3.4 and Section 3.5). This leads to accurate, robust, and fast body shape estimations without manual interaction, resulting in a practical system.

Fitting Body Shapes by Mapping Statistical Models. While the abovementioned works tackle the shape estimation problem by iteratively minimizing an energy function, another body of works estimate the 3D body shape by first constructing statistical models of 2D silhouette features and 3D bodies, and then defining a mapping between the parameters of each model [Xi et al., 2007; Sigal et al., 2007; Chen and Cipolla, 2009; Chen et al., 2010; Chen et al., 2011; Boisvert et al., 2013]. In terms of silhouette representation they vary from PCA learned silhouette descriptors [Chen and Cipolla, 2009; Boisvert et al., 2013] to handcrafted features such as the Radial Distance Functions and Shape Contexts [Sigal et al., 2007]. In Section 3.3.1 we additionally introduce the Weighted Normal Depth and Curvature combined features. The statistical 3D body model is learned by applying PCA on triangle deformations from an average human body shape [Anguelov et al., 2005]. With respect to the body parameter estimations, [Xi et al., 2007] utilize a linear mapping, [Sigal et al., 2007] a mixture of kernel regressors and [Chen and Cipolla, 2009] a shared Gaussian process latent variable model. In our method from Section 3.3.1 we utilize a combination of projections at Correlated Spaces and Random Forest Regressors, while [Boisvert et al., 2013] an initial mapping with the method from [Chen and Cipolla, 2009] which is further refined by an optimization procedure with local fitting. The mentioned methods target applications similar to ours, however they are lacking practicality for interactive applications due to their running times, and have been evaluated under more restrictive assumptions with respect to the camera calibration, poses, and amount of views required.

Under similar settings, our method from Section 3.4 attempts at finding a mapping from one or two images directly, by training an end-to-end Convolutional Neural Network to regress to body shape parameters. On the other hand, the method from Section 3.3.1 finds a fast mapping from specialized silhouette features, projected at correlated spaces, to shape parameters utilizing random forest regressors.

In contrast to these methods, in our third method from Section 3.5, we first learn an embedding space from 3D shape descriptors, that are invariant to isometric deformations, by training a CNN to regress directly to 3D body shape vertices. Then, we learn a mapping from 2D silhouette images to this new embedding space. We demonstrate improved performance over the previous methods working under restrictive assumptions (two views and known camera calibration) with this set-up. Finally, by incorporating cross-modality learning from multiple views, we also outperform our first method (Section 3.3.1) under a more general setting (one view and unknown camera calibration).

CNNs on Applications and 3D Shapes. With the rebirth of neural networks, classification and recognition tasks were revised [Krizhevsky et al., 2012a; Simonyan and Zisserman, 2014; He et al., 2015] and demonstrated more accurate results than previous works. Building on them, there have been recent works using CNNs with 3D shapes for tasks like shape classification and retrieval [Wu et al., 2015; Su et al., 2015; Fang et al., 2015], pose estimation [Toshev and Szegedy, 2014], image semantic segmentation [Long et al., 2015; Girshick et al., 2014] and human re-identification [Cheng et al., 2016]. Most of the methods working on shapes though, tackled retrieval or classification applications and were geared towards rigid shapes (like chairs, tables etc.). To a smaller extent, works like [Toshev and Szegedy, 2014] and [Kendall et al., 2015] tackle regression with CNNs, however for human or camera pose estimation. It has also been a common theme for most of the previous methods that accept a 2D input to use an RGB or grayscale image, often fine-tuning previous architectures trained on similar inputs.

Unlike the above, we newly introduce a method that tries to solve a regression problem, for accurate human shape estimation, by training a CNN from scratch, on binary input images (Section 3.4). We distinguish from other CNN attempts like [Savva et al., ; Su et al., 2015], in that they utilize rigid 3D shapes for matching and retrieval. We further illustrate that our architectures work for different types of inputs such as multiple silhouettes, or images with shading information.

While the improvement in accuracy and performance by utilizing Convolutional Neural Networks for 2D image related tasks is almost undisputed by now, there have been various efforts to adapt CNN-s also for 3D representations. One of the main paradigms is to represent the data as a low resolution voxelized grid [Wu et al., 2015; Su et al., 2015; Rohit Girdhar, 2016]. This representation has been mainly utilized for shape classification and retrieval tasks [Wu et al., 2015; Su et al., 2015; Savva et al.,] or to find a mapping from 2D view representations of those shapes [Rohit Girdhar, 2016], and has been geared towards rigid objects (like chairs, tables, cars etc.). Another possibility to represent the 3D shape, stemming more from the Computer Graphics community is that of 3D Shape Descriptors, which have been extensively studied for shape matching and retrieval [Iyer et al., 2005; Tangelder and Veltkamp, 2008; Vranic et al., 2001].

Various shape descriptors have been proposed, with most recent approaches being diffusion based methods [Sun et al., 2009; Bronstein et al., 2010; Rustamov, 2007]. Based on the Laplace-Beltrami operator that can robustly characterize the points on a meshed surface, some of the proposed descriptors are the global point signature (GPS) [Rustamov, 2007], the heat kernel signature (HKS) [Sun et al., 2009] and the Wave Kernel Signature (WKS) [Aubry et al., 2011]. Further works build on these and related descriptors and learn better descriptors, mainly through CNN-s that are utilized in shape retrieval, classification and especially shape matching [Pickup et al., 2014; Boscaini et al., 2016a; Boscaini et al., 2016b; Masci et al., 2015a; Masci et al., 2015b; Wei et al., 2016; Xie et al., 2016; Litman and Bronstein, 2014; Fang et al., 2015]. Their main objective is either to maximize the inter class variance or to design features that find intra-class similarities.

In our third method (Section 3.5), on the other hand, we want to find suitable descriptors that maximize intra-class variance (here human body shapes), and learn a mapping by regression to 3D body shapes, which to the best of our knowledge has not been explored. Due to the properties of the HKS, such as invariance to isometric deformations and insensitivity to small perturbations on the surface, which are very desirable in order to consistently explain the same human body shape under varying non-rigid deformations, we start from the HKS and encode it into a new shape embedding space, from which we can decode the full body mesh or to which we can regress possible views of the bodies. In this way, our method can be thought of as a generative technique that learns an inverse mapping, from the descriptor space to the shape space.

Multi-View and Cross-Modality Learning. Throughout our methods we attempt to utilize information not necessarily present at inference time, such as multiple views, and we look into techniques that leverage from this additional information during training time, to make predictions more robust. In the presence of multiple views or modalities representing the same data, unsupervised learning techniques have been proposed that leverage such modalities during training, to learn better representations that can be useful when one of them is missing at test time. There exist a couple of applications that rely on learning common representations, including 1) transfer learning, 2) reconstruction of a missing view, 3) matching across views, and directly related to our work 4) boosting single view performance utilizing data from other views or otherwise called cross-modality learning.

Early works, like Canonical Correlation Analysis (CCA) [Hotelling, 1936] and it's kernelized version [Hardoon et al., 2004] are statistical learning techniques that find maximally correlated linear and non-linear projections of two random vectors with the intention of maximizing mutual information and minimizing individual noise. The projected spaces learn representations of two data views such that each view's predictive ability is mutually maximized. Hence, information present in either view that is uncorrelated with the other view is automatically removed in the projected space. That is a helpful property in predictive tasks, that we utilize in Section 3.3.1. The aforementioned methods have been used for unsupervised data analysis with multiple views [Hardoon et al., 2007], fusing learned features for better prediction [Sargin et al., 2007], reducing sample complexity using unlabeled data [Kakade and Foster, 2007], hallucinating multiple modalities from a single view [McWilliams et al., 2013] as well as a generalized version of CCA [Sharma et al., 2012] for a classification and retrieval task. Despite its power, CCA in combination with regression has found little usage since its proposal [Kakade and Foster, 2007]. It has only been empirically evaluated for linear regression [McWilliams et al., 2013], and utilized for an action recognition classification task [Kim et al., 2007]. More generally, except for a few works [McWilliams et al., 2013], utilizing cross-modality learning to improve regression has had little attention. In our first method from Section 3.3.1 we demonstrate the application of CCA to cross-modality learning for body shape estimation, showing that the prediction accuracy can be increased by fusing multi-view information at training time. To tackle some of the shortcomings of the CCA, such as the inability to scale well to large datasets, a deep version of CCA [Andrew et al., 2013] that does not require memorizing the whole training data has been developed, along with its GPU counterpart implementation applied to the problem of matching images and text [Yan and Mikolajczyk, 2015]. More recently Multimodal Autoencoders (MAEs) [Ngiam et al., 2011] have been proposed that also attempt to find common representations for two views/modalities by learning two kinds of reconstructions - self-reconstruction and cross-reconstruction (reconstruction of the other view). Combining the advantages of both MAEs and CCA, the Correlational Neural Networks [Chandar et al., 2016] were presented, but these methods do not focus on boosting single view predictions.

Unlike these techniques, in our third method from Section 3.5, we present a way to perform cross-modality learning by first learning representative features through CNN-s, and then passing them through shared encoding layers, with the objective of regressing to the embedding space. We demonstrate significant increase in performance over uni-modal predictions, and scalability to higher dimensional large scale data.

2.2 Garment Shape Estimation Methods

Following the growing interest in online apparel shopping, virtual reality, and virtual cloth fitting for avatar creation, a wide variety of approaches have been presented that tackle the problem of 3D cloth estimation and modeling. With respect to the input expected, they could be divided into pose-based [Hahn et al., 2014], pose and shape based [Guan et al., 2012], single RGB image based [Zhou et al., 2013], [Yang et al., 2016], single silhouette based [Jeong et al., 2015], multiple RGB images based [Popa et al., 2009] or RGB and Depth image based [Chen et al., 2015]. In terms of the estimation techniques utilized, the methods can be classified as follows: Some of them are based on optimization routines that deform a cloth model to fit to image-space information (e.g. contour [Yang et al., 2016]), others find a mapping to cloth panels or measurements that in turn are used to reconstruct the meshes with Physically Based Simulations (PBS) [Jeong et al., 2015], or directly find a mapping to 3D shape or shape deformations [Guan et al., 2012].

Our method, presented in Section 4.2, takes a single RGB image as the input, and estimates 3D vertex deformations. The current single image based methods come with various limitations such as the need for manual interaction and assumptions on the camera poses and lighting conditions [Yang et al., 2016; Jeong et al., 2015; Zhou et al., 2013], restriction on the complexity of human poses [Jeong et al., 2015; Zhou et al., 2013], symmetry assumptions for the back of the cloth [Zhou et al., 2013], inability to handle self occlusions [Zhou et al., 2013], high run-time [Bradley et al., 2008; Yang et al., 2016; Zhou et al., 2013], and the assumption of a statically stable physical state on the cloth and underlying human body [Yang et al., 2016; Jeong et al., 2015] that prohibits the estimation of clothes under dynamic scenes. Our method aims at overcoming these limitations, making single-image 3D garment capture practical.

The cloth shape estimation techniques can be further split into several categories based on the general approach utilized as follows:

Structure-from-Motion-based Techniques. These methods modify and extend the standard SfM setup to estimate the shape of the garment. Some of the techniques rely on special markers depicted on the garment to make the process easier, with early work focusing on reconstruction of just single sheets of cloth or smaller pieces of garments from single images [Pritchard and Heidrich, 2003; Scholz and Magnor, 2004; Guskov et al., 2003]. The first work [Scholz et al., 2005] to solve the reconstruction problem for the entire garment assumed special markers that are easily detectable and localizable in 3D via a standard multi-view camera setup. [White et al., 2007] optimized the quality of the results further with a smarter marker selection and a new hole filling strategy producing high quality results with a speed of several minutes per frame. [Bradley et al., 2008] utilized anchor points set to special garment locations that can be easily detected (e.g. sleeves or the neckline) in a controlled setup, eliminating the need of special markers. In a follow up work [Popa et al., 2009], the final garment shape is further augmented utilizing edge detection to deform areas using a handcrafted non-rigid local transformation that can reconstruct higher frequency plausible wrinkles. Unlike these works, we target a single image-based setting with a minimally controlled setup.

Shape-from-Shading-based Techniques. [Zhou et al., 2013] propose garment modeling from a single image, utilizing statistical human body models and having the user outline the silhouette of the garment and set the body pose. The initial shape is then estimated by constructing oriented facets for each bone [Robson et al., 2011], and assuming symmetry in order to model the garment from the back as well. Then, shape-from-shading is used to recover higher frequency folds, achieving comparable results to the method from [White et al., 2007], however with considerable user interaction, runtime in order of minutes, and the inability to handle self occlusions of the character.

Data-driven Techniques. Most data-driven works have focused on estimating the naked human body shape from images, mainly utilizing statistical human body shape priors learned from 3D naked human body scans with techniques similar to SCAPE [Anguelov et al., 2005]. [Bălan and Black, 2008] utilized such a model to infer human body under clothing with a multi-camera setup. Other works estimate the human body from a single image by mapping silhouette features with random forests [Sigal et al., 2007], regression forest with canonical correlation analysis (Section 3.3.1), or Gaussian process latent variables [Chen et al., 2010]. Other works focus on face shape estimation [Cao et al., 2015], that enhance the quality by mapping clustered wrinkle patterns to mesh height fields through local regressors.

Earlier data-driven techniques estimate 2D garment shapes based on computed 2D silhouette cloth descriptors and difference from naked body silhouettes [Guan et al., 2010]. Applying this idea to 3D cloth modeling, a generative model (DRAPE [Guan et al., 2012]) was proposed that allows to dress any person in a given shape and pose, by learning a linear mapping from SCAPE [Anguelov et al., 2005] parameters to DRAPE cloth parameters. A similar approach was taken by [Alexandros Neophytou, 2014], utilizing another technique for modeling human shapes, and a clothing model that is treated as a transformation factor from a reference body shape to the final clothed shape. This is in contrast to DRAPE that learns separate models for every type of clothing. [Hahn et al., 2014] take a different approach, where instead of modeling clothing as an "offset" from the naked body shape, they approximate physically based cloth simulators by clustering in the pose space and performing PCA on the simulated garment per cluster to reduce the complexity and speed up the process. Similarly, we model a garment as a deformation from a body or from a template garment shape. However, unlike these methods, we tackle the problem of 3D garment estimation from images.

Other works aim at estimating 3D garment shape from a single image. Some of these methods assume depth is known [Sekine et al., 2014; Chen et al., 2015], while others work for restricted mannequin poses and given cloth panels [Jeong et al., 2015], or assume considerable manual interaction and statically stable physical state of the garment and the underlying human body to map wrinkle patterns and segmented garments to cloth parameters and materials through fitting, taking several hours to compute [Yang et al., 2016]. In contrast, our method from Section 4.2 can estimate dynamic garment shapes that are not in steady-state, minimizes user interaction, and runs at interactive rates.

Deep Learning. In recent years, there has been a massive uptake of deep learning in all of the applied machine learning fields thanks to advances in parallel computing on GPUs and the concept of stacking multiple layers to learn richer representations of data. CNN-s have been proven to outperform state-of-the art techniques in computer vision applications such as image classification [Krizhevsky et al., 2012a; Szegedy et al., 2015; Simonyan and Zisserman, 2014], feature detection (Overfeat [Sermanet et al., 2013]) and description (LIFT [Yi et al., 2016]), optical flow estimation [Fischer et al., 2015; Ilg et al., 2017], 2D pose estimation [Toshev and Szegedy, 2013], denoising, segmentation etc. Since the creation of AlexNet [Krizhevsky et al., 2012a], deeper architectures have been developed, such as the Deep Residual Net [He et al., 2015] which introduced the "shortcut connections" to achieve state-of-the-art, along with smaller architectures like SqueezeNet [Iandola et al., 2016a] that achieves AlexNet performance utilizing 50× less parameters. While there have been recent works on 2D cloth recognition and retrieval [Liu et al., 2016], 3D shape classification, retrieval and representation such as [Su et al., 2015], [Wu et al., 2015] and [Wang et al., 2015] targeted to rigid 3D objects or cloth capturing, modeling and re-targeting from 3D scans [Pons-Moll et al., 2017], except for the methods from Chapter 3 that infer 3D human body shape from silhouettes, to the best of our knowledge, there has been no previous work that attempts to infer 3D garment shape from monocular images, as in here.

2.3 Hand Pose Estimation Methods

Hand pose and human pose estimation are highly related fields, with the former having gained quite some popularity in the recent years, while utilizing and applying many of the principles from the latter. We would like to refer to [Sarafianos et al., 2016; Ye et al., 2013] as well as most recent works [Mehta et al., 2016; Mehta et al., 2017; Song et al., 2017] for a more comprehensive analysis of human pose estimation, and focus on 3D hand pose inference relevant methods, which can be primarily classified with respect to the input as depth, monocular RGB, multi-view, and video-based. Given the low cost of RGBD sensors, there has been a vast amount of work on hand pose estimation based on depth images, which can be further classified as being either generative (model-based), discriminative (appearance based), or both (hybrid) [Erol et al., 2007]. An additional classification can be made based on how the input is mapped to the output : 2D-to-3D lifting [Zimmermann and Brox, 2017; Tomè et al., 2017; Zhao et al., 2016; Bogo et al., 2016b; Tompson et al., 2014; Panteleris et al., 2017] or direct 3D mapping based methods [Zhou et al., 2016; Oberweger et al., 2015a]. Our methods can be classified as discriminative, direct 3D mapping methods with a monocular Depth (Section 5.1) or RGB (Section 5.3) as input.

Generative, Optimization-Based Approaches. Many methods in this category utilize gradient-based optimization approaches and attempt to solve an iterative closest point (ICP) problem. In this context, [Melax et al., 2013] formulate the hand optimization as a constrained rigid body problem. [Schröder et al., 2014] suggest optimizing in a reduced parameter space and [Tagliasacchi et al., 2015] combine previous results, to show that ICP in combination with temporal, collision, kinematic and data-driven terms can be utilized to track with high robustness and accuracy from a depth video. Following up on this, [Sharp et al., 2015] enhance this approach utilizing a smooth model and [Tkach et al., 2016] present a new hand model based on sphere meshes.

A non-gradient, particle swarm optimization (PSO) approach has been suggested by [Oikonomidis et al., 2011], minimizing "the discrepancy between the appearance and 3D structure of hypothesized instances of a hand model and actual hand observations". This requires extensive rendering of an explicit hand model in various poses. [Tompson et al., 2014] use an (offline) PSO based approach to find the ground truth for the NYU dataset [Tompson et al., 2014]. Since PSO depends highly on a good initialization, [Qian et al., 2014] increase its robustness by combining it with ICP, while [Taylor et al., 2016] suggest minimizing a truncated L1 error norm between the synthesized and real depth image while also rendering a more realistic-looking mesh through linear blend skinning (LBS) [Lewis et al., 2000]. In general, these techniques focus on tracking, requiring a good initialization or GPU implementations, while our method from Section 5.1 focuses on single depth image 3D pose estimation and can run real-time on CPU.

Discriminative, Data-Driven Approaches. Recently, many methods based on convolutional neural networks (CNNs) have been proposed. [Oberweger et al., 2015a] evaluate different CNN architectures and propose a pose prior by adding a bottleneck layer showing that a projection to a reduced subspace before the final regression boosts the prediction performance. [Zhou et al., 2016] propose a forward kinematic layer to create a loss function on the joint positions while predicting rotation angles of the joints. Using those angles, a physical loss is introduced, which penalizes angles outside a specified range, similar to what we do. Instead of directly using depth images as input, [Ge et al., 2016] show that projecting the point cloud onto three orthogonal planes and feeding the projections into three different CNN-s enhances the prediction performance. [Deng et al., 2017] convert the depth map to a 3D volumetric representation first, and then feed it into a 3D CNN to produce the pose in 3D, requiring no further processing. Apart from CNN-s, there exist also methods that utilize decision forests to make a 3D pose prediction [Keskin et al., 2012; Tang et al., 2017; Xu et al., 2016]. These methods are typically fully supervised, except for [Wan et al., 2017; Tang et al., 2013]. We show semi-supervised and unsupervised adaptations, with real RGB and depth data, applied to RGB and depth input.

Hybrid Methods. Often, neural networks are used as an intermediate prediction step which requires optimization afterward. [Tompson et al., 2014] predict various key positions and optimize for the actual pose using inverse kinematics. [Mueller et al., 2017] fit the hand skeleton to 2D and 3D joint predictions from a CNN. [Ye et al., 2016] combine a spatial attention mechanism and PSO in a cascaded and hierarchical way. [Sinha et al., 2016] utilize a CNN to reduce the dimensionality of the depth input and optimize for the final pose via a matrix completion approach considering also temporal information. [Oberweger et al., 2015b] use a deep generative neural network to synthesize depth images, and a separate optimization network to iteratively correct the pose predicted by a third convolutional model. Similar in spirit to the PSO approaches, starting from a rigged 3D hand model, we synthesize depth images in order to compare to the input depth images. We, however, do not do this externally, but rather integrate it in a conventional gradient based learning architecture, similar to [Loper and Black, 2014; Roveri, 2018].

Our base CNN architecture from Section 5.1 initially predicts joint rotations from a base reference pose, similar to [Zhou et al., 2016]. This allows us to completely reconstruct the articulated hand pose, whereas predicting just joint positions needs a further optimization step to do the same. Our method builds on top of [Zhou et al., 2016], since we also calculate a forward kinematic chain from the predicted pose and we also utilize a physical loss (Section 5.1). However, we go a step further and do not minimize a loss on the joint positions but on the actual hand depth image, enabling us to adapt to unlabeled images.

At first glance our method might appear to be similar to the feedback loop proposed by [Oberweger et al., 2015b], but there are some important differences we want to emphasize to avoid confusion. [Oberweger et al., 2015b] synthesize depth images too, however such images are utilized to iteratively optimize a pose prediction during testing, whereas we optimize our base model during training only, by backpropagating errors on depth images. Our prediction employs only a single forward pass through the CNN. Furthermore, our method is completely independent of labeled real depth images, whereas [Oberweger et al., 2015b] highly depends on well labeled data to adapt to a dataset (e.g. training on ICVL dataset fails because of annotation errors). Our method allows for simple end-to-end training, but the method from [Oberweger et al., 2015b] requires to train three different neural networks. We will elaborate more on the differences between the state-of-the-art methods in Section 5.1.

All in all, our method from Section 5.1 could be seen as a network extension to data-driven methods, in order to boost predictions by training at a minimal cost (from unlabeled depth data). Our goal thus, slightly differs from that of most of the abovementioned works, which mainly focus on maximizing pose estimation accuracy on available datasets. Our method from Section 5.3 on the other hand, can be seen as its adaptation to RGB images, expanding also on other tasks such as gesture recognition, 2D joint estimation etc.

Video-Based Methods. Since RGBD sensors are not always available, further methods have been proposed, that utilize RGB images in combination with temporal information. [de La Gorce et al., 2011] use texture, position and pose information from the previous frame to predict the current pose. [Romero et al., 2009] exploit temporal knowledge to guide a nearest-neighbor search. [Song et al., 2015] perform a joint estimation of the 3D hand position and gestures for mobile interaction. In general, these methods have to solve the problem of obtaining a first estimate.

Multi-View-Based Methods. Another approach involves the use of multiple cameras to compensate for the lack of depth data, alleviating the problems with occluded parts. [Zhang et al., 2016] utilize stereo matching for hand tracking, [Simon et al., 2017] apply multi-view bootstrapping for keypoint detection, and [Sridhar et al., 2014] estimate 3D hand pose from multiple RGB cameras, with a hand shape representation based on a sum of Anisotropic Gaussians, whereas [Sridhar et al., 2013] combine RGB and Depth data to obtain a richer input space.

Image-Based Methods. Due to the larger availability of regular color cameras, opposed to the abovementioned methods, in our method from Section 5.3, we make use of neither depth nor multi-camera or temporal information. One of the first single frame based hand detection works, from [Athitsos and Sclaroff, 2003] utilize edge maps and Chamfer matching. With the exception of concurrent works [Spurr et al., 2018; Panteleris et al., 2017; Mueller et al., 2017], it was only recently that one of the first monocular RGB based methods [Zimmermann and Brox, 2017] for 3D hand pose estimation was presented, utilizing CNN-s and synthetic datasets. In contrast to our method, they split the prediction into a 2D joint localization step followed by a 3D up-lifting, and use their own synthetic dataset to complement the scarcity of existing datasets. We utilize our new, high quality, hand synthetic dataset to predict 3D joint angles directly from an RGB image and strongly compare to [Zimmermann and Brox, 2017] on various tasks in Section 5.4.

CHAPTER 3

Human Body Shape Estimation

Estimating the human body shape from imagery is an important problem in computer vision with diverse applications. The estimated body shape provides an accurate proxy geometry for further tasks such as rendering free viewpoint videos [Xu et al., 2011; Stoll et al., 2010; Casas et al., 2014; Starck et al., 2005], surveillance [Chen et al., 2013], tracking [Guan et al., 2008], biometric authentication, medical and personal measurements [Boisvert et al., 2013; Thaler et al., 2018], virtual cloth fitting [Guan et al., 2012; Wuhrer et al., 2014; Rogge et al., 2014; Alexandros Neophytou, 2014], and artistic image reshaping [Zhou et al., 2010]. Pose estimation is also tightly coupled with shape estimation. Knowing the body shape significantly reduces the complexity and improves the robustness of pose estimation algorithms and thus expands the space of poses that can be reliably estimated [Ye and Yang, 2014; de Aguiar et al., 2008].

However, as opposed to pose estimation, body shape estimation has received less attention from the community. The majority of existing algorithms are typically based on generative approaches that minimize an error fitting term. They rely on either manual input [Zhou et al., 2010; Jain et al., 2010; Rogge et al., 2014], restrictive assumptions on the acquired images [Boisvert et al., 2013], or require information other than just 2D images (e.g. depth) [Weiss et al., 2011; Perbet et al., 2014; Helten et al., 2013]. Furthermore, some of the methods have prohibitive complexity for real-time applications [Jain et al., 2010; Boisvert et al., 2013; Weiss et al., 2011]. A practical human body shape estimation algorithm should be accurate, robust, efficient, automatic and fast. Additionally, it should work with images acquired under less restrictive conditions and body poses.

In contrast to optimization based approaches, it has been shown repeatedly that utilizing neural networks can lead to superior results for many problems such as classification [Krizhevsky et al., 2012a], segmentation [Long et al., 2015; Girshick et al., 2014], pose estimation [Toshev and Szegedy, 2014] and shape classification or retrieval [Su et al., 2015; Wu et al., 2015; Fang et al., 2015]. Despite discriminative attempts to estimate the human body shape [Sigal et al., 2007; Chen et al., 2010], applying CNN techniques to body shape estimation has had very little attention.

In this chapter, we present three discriminative methods that tackle the problem of estimating the human body shape from monocular images. We focus on the silhouette as the most important visual cue, due to its relevance in describing the human body, and the ease of extraction, as demonstrated in recent works [Varol et al., 2017]. We initially introduce a method based on silhouette features, that get mapped to the human body shape through random forest regressors, and for the remaining two main sections, we newly explore the utilization of CNN-s to tackle the very same task. We start with simpler AlexNet [Krizhevsky et al., 2012b] like architectures and enhance them with generative and cross-modal components, observing an improvement in accuracy as compared to both discriminative and generative works. All the three methods rely on a parametric body shape model, learned from human body scans, hence, below we present it at first. Due to their discriminative nature and the variety of shapes that they need to capture, such methods leverage from training data abundance. Since the amount of real training data is very limited, we recur to synthetic ones, whose generation we explain next and then introduce the three methods. In the end we conclude the chapter with a discussion on the advantages and limitations of each method.

3.1 Shape as a Geometric Model

Deformable shape models are a common choice to tackle the problem of human shape estimation from a few camera images [Balan et al., 2007; Sigal et al., 2007; Bălan and Black, 2008; Guan et al., 2009; Zhou et al., 2010; Jain et al., 2010; Guan et al., 2012], in particular SCAPE [Anguelov et al., 2005]. While there exist other, more recent body models [Neophytou and Hilton, 2013; Loper et al., 2015], we chose it mainly due to its simplicity and ease of comparison with related methods that adopt it as well. It is a low-dimensional parametric model that captures correlated deformations due to

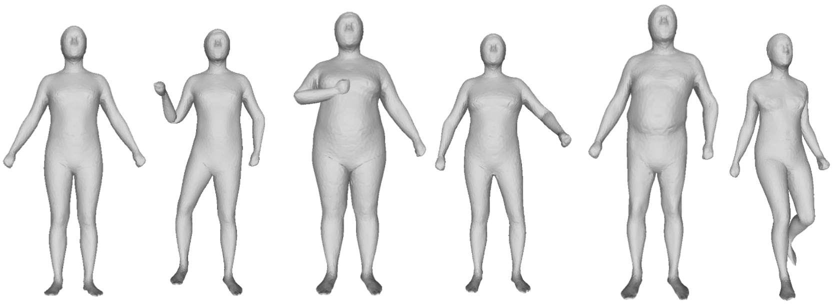


Figure 3.1: 6 *meshes from our database. The leftmost one is the mean mesh in the rest pose. The others are from different people in various poses.*

shape and pose changes simultaneously. Specifically, SCAPE is defined as a set of triangle deformations applied to a reference template 3D mesh and learned from 3D range scans of different people in different poses. More concretely, throughout our methods, SCAPE is defined as a set of 12894 triangle deformations applied to a reference template 3D mesh consisting of 6449 vertices. Estimating a new shape requires estimating parameters α and β, which determine the deformations due to pose and intrinsic body shape, respectively. Given these parameters, each of the two edges $\mathbf{e}_{i1}$ and $\mathbf{e}_{i2}$ of the i^{th} triangle of the template mesh (defined as the difference vectors between the vertices of the triangle), is deformed according to the following expression:

$$\mathbf{e}_{ij}^{'} = \mathbf{R}_i(\alpha)\mathbf{S}_i(\beta)\mathbf{Q}_i(\mathbf{R}_i(\alpha))\mathbf{e}_{ij}, \tag{3.1}$$

with $j \in \{1,2\}$. The matrices $\mathbf{R}_i(\alpha)$ correspond to joint rotations, and $\mathbf{Q}_i(\mathbf{R}_i(\alpha))$ to the pose induced non-rigid deformations, e.g. muscle bulging. $\mathbf{S}_i(\beta)$ are matrices modeling shape variation as a function of the shape parameters β. The body shape deformation space is learned by applying PCA to a set of meshes of different people in full correspondence and same pose, with transformations written as $\mathbf{s}(\beta) = \mathbf{U}\beta + \mu$, where $\mathbf{s}(\beta)$ is obtained by stacking all transformations $\mathbf{S}_i(\beta)$ for all triangles, $\mathbf{U}$ is a matrix with orthonormal columns, and μ is the mean of the triangle transformations over all meshes. For further details please refer to [Anguelov et al., 2005]. We therefore obtain the model by computing per-triangle deformations for each mesh of the dataset from a template mesh, which is the mean of all the meshes in the dataset (Figure 3.1 left), and then applying PCA in order to extract the components capturing largest deformation variations. We chose to use 20 components ($\beta \in R^{20}$) which are enough to capture more than 95% of the energy.

For all the methods, and more specifically the first two methods, we would like to estimate the shape parameters β regardless of the pose (since the third one attempts to directly estimate 3D mesh vertices). We take the common assumption that the body shape does not significantly change due to the range of poses we consider. Hence, we ignore pose dependent shape changes given by $\mathbf{Q}_i(\mathbf{R}(\alpha))$. Decoupling pose and shape changes allows us to adopt a fast and efficient method from the graphics community for pose changes, known as Linear Blend Skinning (LBS) [Lewis et al., 2000], similar to previous works [Pishchulin et al., 2015; Jain et al., 2010; Feng et al., 2015; Yang et al., 2016]. Starting from a rest pose shape with vertices $\mathbf{v}_1, ..., \mathbf{v}_n \in \mathbf{R}^4$ in homogenous coordinates, LBS computes the new position of each vertex by a weighted combination of the bone transformation matrices $\mathbf{T}_1, ..., \mathbf{T}_m$ in a skeleton controlling the mesh, and skinning weights $w_{i,1}, ..., w_{i,m} \in \mathbf{R}$ for each vertex $\mathbf{v}_i$, as given by the following formula:

$$\mathbf{v}_i^{'} = \sum_{j=1}^{m} w_{i,j}\mathbf{T}_j\mathbf{v}_i = \left(\sum_{j=1}^{m} w_{i,j}\mathbf{T}_j\right)\mathbf{v}_i \tag{3.2}$$

In our model, the skinning weights are computed for a skeleton of 17 body parts (1 for the head, 2 for the torso, 2 for the hips and 3 for each of the lower and upper limbs) for the mean shape mesh using the heat diffusion method [Baran and Popovic, 2007a]. It has to be noted that $w_{i,j} \geq 0$ and $w_{i,1} + \cdots + w_{i,m} = 1$. Once the shape parameters β are estimated, reconstructing a new shape utilizing SCAPE involves solving a least-squares system over such parameters, which runs in milliseconds.

3.2 Data Generation

Our training based methods require numerous training and validation data. Gathering a big number of human shapes is a highly non-trivial task - due to the need of specialized equipment for scanning people, the difficulty of finding a large number of them, and more importantly, due to the necessity of scanning them under minimalistic clothing, in order to better capture the intrinsic shapes. Unfortunately, there exists no freely available dataset of real human body shapes along with measurements. A feasible solution though, would be to learn a parametric shape model from a small subset of body shapes that capture body shape variances and generate synthetic data from it, as explained in Section 3.1.

Taking advantage of the commercially available CAESAR dataset [Robinette and Daanen, 1999] [1], containing people in an almost naked apparel, researchers have released two datasets [Yang et al., 2014; Pishchulin et al., 2015], consisting of meshes obtained by fitting a template mesh to subsets of the CAESAR dataset. We merge these two datasets and construct a larger one, to enable learning a more general shape model. One of them [Yang et al., 2014], consists of around 1500 registered meshes in correspondence, however of higher resolution than the other dataset [Pishchulin et al., 2015]. The resolutions respectively are 12500 vertices 25000 triangles and 6449 vertices 12894 triangles. Mesh resolution is not so important for our application, hence we map the higher resolution meshes to the lower resolution ones. This also improves the computation time. To achieve that, we first extract a template mesh, as the mean mesh of each dataset, and then apply non-rigid ICP [Amberg et al., 2007] to the two template meshes. Afterwards, closest points in both meshes are computed, using barycentric coordinates in the closest triangle. The retrieved mapping can be applied to all the remaining meshes due to the same mesh connectivity.

We select 2900 meshes from the combined dataset for learning the shape model, leaving out around 1500 meshes for testing and experiments. In order to synthesize more training meshes, we sample from the 20 dimensional multivariate normal distribution spanned by the PCA space (Section 3.1), such that for a random sample $\beta = [\beta_1, \beta_2, ..., \beta_{20}]$, it holds that $\beta \sim \mathcal{N}(\mu, \Sigma)$ with μ being the 20-dimensional mean vector and Σ the 20×20 covariance matrix of the parameters. To synthesize meshes in different poses, we gather a set of animations comprising of various poses (e.g. selfie, walking, running, etc.). After transferring a generated pose to the template mesh using LBS, we compute the resulting per-triangle deformations $\mathbf{R}_i$. For a given mesh with parameters β, the final pose is then given by $\mathbf{e}'_{ij} = \mathbf{R}_i \mathbf{S}_i(\beta) \mathbf{e}_{ij}$, where $\mathbf{e}_{ij}$ are the edges of the template mesh (Section 3.1).

As the training set, we randomly generate around 100000 samples from the multi-variate distribution over the β parameters, and restrict them to fall into the $\pm 3 \times \mathit{Std.Dev}$ range for each dimension of the PCA projected parameters to avoid getting unrealistic human shapes. After applying LBS on various template poses, in order to enrich the dataset, we can easily obtain silhouettes, shaded images and even mesh descriptors, which are inputs utilized in the following methods. The silhouette is computed by projecting all the mesh edges for which two coinciding triangles have normals pointing in opposite directions, as seen by the camera viewpoint. For testing, we evaluate our method with images from the meshes left out from the training dataset, as well as on real ones.

[1] http://store.sae.org/caesar/

3.3 Shape Estimation with Handcrafted Features and CCA

In this first section, we propose a fast and automatic method for estimating the 3D body shape of a person from images, utilizing *multi-view* semi-supervised learning. This method relies on extracting novel features from a given silhouette of a single person under minimal self-occlusion, like in a selfie, and a parametric human body shape model [Anguelov et al., 2005]. The latter is utilized to generate meshes spanning a spectrum of human body shapes, from which silhouettes are computed over multiple views, as also explained in Section 3.2, in poses compliant with the target applications for training. We firstly estimate viewing direction with high accuracy, by solving a classification task. Utilizing the information simultaneously captured in multiple synthetic views of the same body mesh, we apply Canonical Correlation Analysis (CCA) [Hotelling, 1936] to learn informative bases where the extracted features can be projected. A random forest regressor is then adopted to learn a mapping from projected feature space to parameter space. This results in lower feature dimensionality, reducing the training and test time drastically, and improves prediction as compared to plain regression forests. We demonstrate our results on real people and comprehensively evaluate our method by validating it on thousands of body shapes.

3.3.1 Method Overview

The goal of our system is to infer the 3D body shape of a person from a single or multiple monocular images fast and automatically. Specifically, we would like to estimate the parameters of a 3D body shape model (Section 3.1) such that the corresponding body shape best approximates the 3D body of the subject depicted in the input images. Despite the ambiguity that the 2D silhouette withholds, the projection of the transformed mesh in the image should at least best explain it.

An overview of our system is depicted in Figure 3.2. The input to the shape estimation algorithm is a 2D silhouette of the desired individual under minimal self-occlusion (e.g. a selfie), which can be computed accurately for our target scenarios, by learning a background model through Gaussian mixture models and using Graphcuts [Boykov and Jolly, 2001]. The word “selfie” here is used interchangingly to describe the activity of taking a selfie in front of a mirror, and also as a label for poses representing mild self-occlusion (Figure 3.1). We then compute features extracted from the silhouettes (Section 3.3.2). These are first used to train a classifier on the camera viewing direction (Section 3.3.3). The features from silhouettes of a particular view

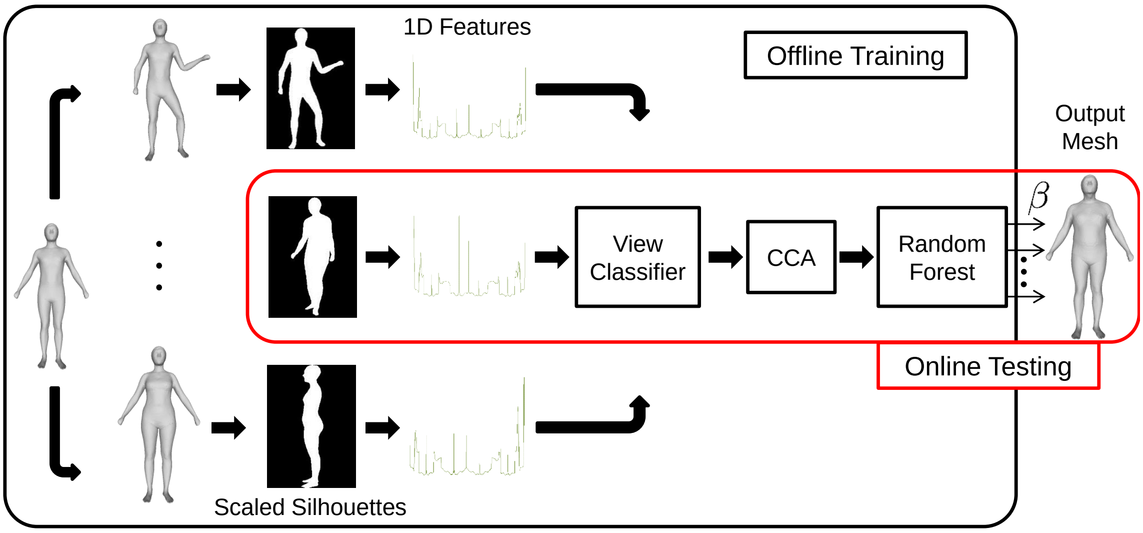


Figure 3.2: *Overview of the first method.* ***Training:*** *Silhouettes from 36 views are extracted from meshes generated in various shapes and poses (Section 3.1). A View Classifier is learned (Section 3.3.3) from extracted silhouette features (Section 3.3.2). View specific Regression Forests are then trained to estimate shape parameters by first projecting features in CCA correlated spaces (Section 3.3.4).* ***Testing:*** *The extracted features from an input silhouette are used to first infer the camera view, and then the shape parameters by projecting them into CCA spaces and feeding them into the corresponding Regression Forest.*

are then projected into bases obtained by CCA, such that the view itself and the most orthogonal one to it (e.g. front and side) are used to capture complementary information into the CCA correlated space, and fed to a Random Forest Regressor (Section 3.3.4) trained for each camera view. At test time, the extracted features from an input silhouette are used to first infer the camera view, and then the shape parameters by projecting them into CCA spaces and feeding them into the corresponding Regression Forest. The parameters are used to generate a mesh by solving a least-squares system on the vertex positions (Section 3.1). The generated mesh can then be utilized for various post-processing tasks such as human semantic parameter estimation, free view-point video with projective texturing, further shape refinement [Zhou et al., 2010; Boisvert et al., 2013], or pose refinement [Jain et al., 2010].

3.3.2 Feature Extraction

We extract novel features from the scaled silhouettes, with height equal to 528 pixels and width to 384 pixels, as the input to our learning method.

These features are designed to capture local and global information on the silhouette shape, and be robust to pose and slight view changes. For each point in the silhouette, two feature values are calculated, namely the *(weighted) normal depth* and the *curvature*. In order to extract these, we first compute the 2D point normal for every point in the silhouette, and then smooth all normals with a circle filter of radius of 7 pixels. As different people have different silhouette lengths, we sample 1704 equidistant points from each silhouette starting from the topmost pixel of the silhouette. The sample size is set according to the smallest silhouette length over all our training data. Our feature vector per silhouette then consists of 3408 real valued numbers.

The normal depth is computed as follows. For any point from the sampled set, we send several rays starting from the point itself and oriented along the opposite direction of its normal, until they intersect the inner silhouette boundary. The lengths of the ray segments are defined as the normal depths as illustrated in Figure 3.3. The normals are represented in green and the ray segments in red for two different points in the silhouette. We allow an angle deviation of 50 degrees from the silhouette normal axis. The feature for a point is defined as the weighted average of all normal depths falling within one standard deviation from the median of all the depths, with weights defined as the inverse of the angle between the rays and the normal axis.

The *normal depth* is a feature inspired by 3D geodesic shape descriptors [Shapira et al., 2008; Slama et al., 2013], differing from the *Inner-Distance* 2D descriptor [Ling and Jacobs, 2007] used for classification of different object types while being noise sensitive, and the spectral features utilized in [Lahner et al., 2016] for a shape retrieval task. The main ideas behind our feature are (a) for the same individual in different poses, under mild self-occlusions, the features look very similar with small local shifts, (b) each point feature serves as a robust body measurement, correlated with the breadth of the person in various parts of the body, which is analogous to estimating body circumference at each vertex of the real body mesh, and (c) the feature is robust to silhouette noise due to the median and averaging steps. The measure might differ though in some parts of the silhouette (e.g. elbow) for the same person in different poses. In order to alleviate this limitation, we apply smoothing on small neighborhoods of the silhouette. The *curvature* on the other hand is estimated as the local variance of the normals. Despite being a local feature, it provides a measure of roundness, especially around the hips, waist, belly and chest, which helps in discriminating between various shapes.

We illustrate that the combination of normal depth that captures global

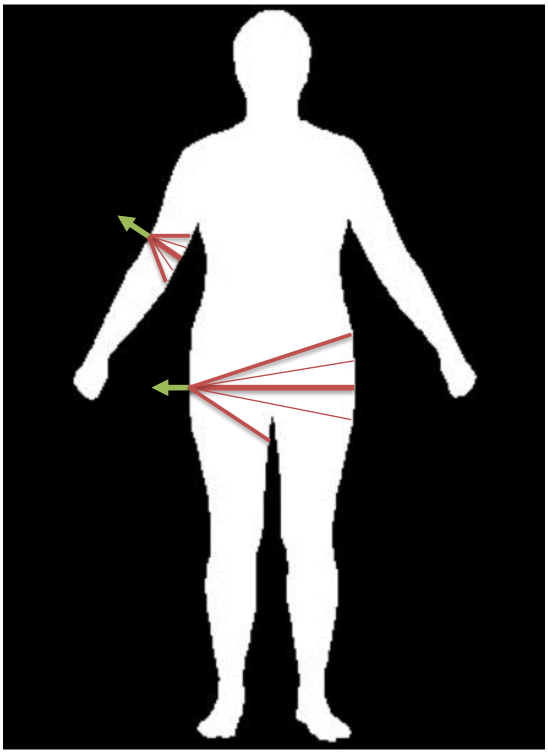


Figure 3.3: *Normal depth computation in 2 different points. The arrows are the silhouette normals. The normal depth is computed as the weighted mean of the lengths of the red lines.*

information on the silhouette and curvature encoding local details leads to estimators robust to limited self-occlusions, and discriminative enough to describe the silhouette and reconstruct the corresponding shape in Section 3.3.5.

3.3.3 View Direction Classification

To increase robustness with respect to view changes, we decided to train view-specific Regression Forests for 36 viewing directions around the body. In order to discriminate between the views, we train a Random Forest Classifier utilizing the 3408 features extracted (Section 3.3.2) from $100,000$ silhouettes of people in multiple poses, shapes and views, having as labels the views numbered 1 to 36. We achieve a high accuracy of 99% if we train and test on neutral and selfie-like poses. The accuracy decreases to 85.7% if more involved poses (e.g. walking, running etc) are added. However, by investigating class prediction probabilities, we observed that false positives are assigned only to the views that are contiguous to the view with the correct label. As it will be shown in Section 3.3.5, Table. 3.2, a 10 degree view difference has a low reconstruction error when the features are projected into CCA bases.

3.3.4 Learning Shape Parameters

We pose shape parameter estimation as a regression task. Given the silhouette features, using supervised learning, we would like to estimate the shape parameters such that the reconstructed shape best explains the silhouette. To make the features more discriminative, we propose to correlate features extracted from silhouettes viewed from different directions. More specifically, we apply Canonical Correlation Analysis (CCA) [Hotelling, 1936] over features extracted from a pair of silhouettes from two camera views.

At training time, the views are selected such that they capture complementary information. While the first one is the desired view from which we want to estimate the shape (one of 36 views), the second one is chosen to be as orthogonal as possible to the first, e.g. (front and side view). Because the human body is symmetric, a complementary view to a desired one is always searched in the zero to 90 degree angle range to that view. In practice, we round the complementary view to the closest extreme (i.e front or side view) to ease the offline computations.

We first apply PCA to reduce the dimensionality of the extracted features from 3408 to 300 in each view. Then, we stack the PCA projected features for all mesh silhouettes from the first and second views into the columns of the matrices $\mathbf{X}_1$ and $\mathbf{X}_2$, respectively. Then, CCA attempts to find basis vector pairs $\mathbf{b}_1$ and $\mathbf{b}_2$, such that the correlations between the projections of the variables onto these vectors are mutually maximized by solving:

$$\arg\max_{\mathbf{b}_1,\mathbf{b}_2 \in R^N} corr(\mathbf{b}_1^T\mathbf{X}_1, \mathbf{b}_2^T\mathbf{X}_2), \tag{3.3}$$

where $N = 300$. This results in a coordinate free mutual basis unaffected by rotation, translation or global scaling of the features. The features projected onto this basis thus capture mutual information coming from both views. The subsequent basis vector pairs are computed similarly, with the assumption that the new projected features are orthogonal to the existing projected ones. We use 200 basis pairs with CCA projections covering 99% of the energy.

The final training is done on the 200 projected features extracted from one view, which is one of the 36 views we consider. These projected features are

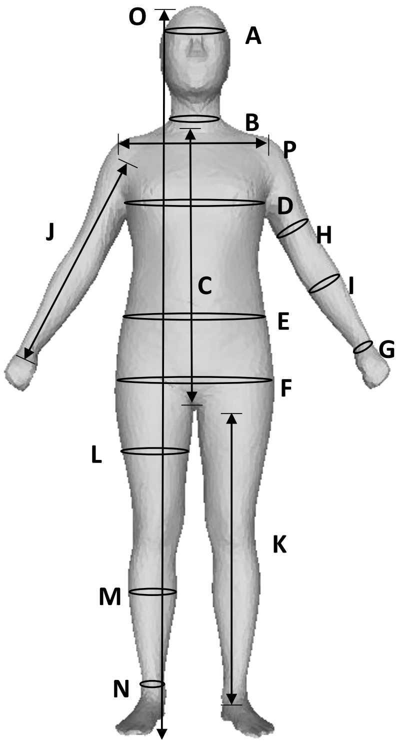


Figure 3.4: *3D measurements on the meshes used for validation.*

input to a Random Forest Regressor [Breiman, 2001] of 4 trees and a maximum depth of 20. The labels for this regressor are the 20-dimensional shape parameter vectors β. Each component of β is weighted with weights set to the eigenvalues of the covariance matrix defined in Section 3.1 in the computation of the shape deformation space, and normalized to 1, to emphasize the large scale changes in 3D body shapes. At test time, the raw features extracted from a single given silhouette are first classified into a view. These are then projected with the obtained PCA and CCA matrices for that view to obtain a 200 dimensional vector. The projected features are finally fed into the corresponding Random Forest Regressor, in order to obtain the desired shape parameters β.

3.3.5 Validation and Results

Previous shape-from-silhouette methods lack extensive evaluation. [Xi et al., 2007] demonstrate results on two real images of people and 24 subjects in synthetic settings, [Sigal et al., 2007] validate on two measurements and two subjects in monocular settings, and [Balan et al., 2007] report silhouette errors for a few individuals in a sequence and height measurement for a single individual. To the best of our knowledge, only [Boisvert et al., 2013]

Measurement	RF	CCA-RF-1	CCA-RF-2	GT
A. Head circumference	16±13	13±10	**8 ± 8**	13±9
B. Neck circumference	13 ±10	10±8	**7 ± 7**	6±6
C. Shoulder-blade/crotch length	22±18	18±9	**18 ± 17**	14±11
D. Chest circumference	38 ±31	30±24	**25 ± 24**	24±24
E. Waist circumference	35 ±28	29±25	**24 ± 24**	16±14
F. Pelvis circumference	33 ±26	30±25	**26 ± 25**	14±12
G. Wrist circumference	10 ±8	6±5	**5 ± 5**	5±5
H. Bicep circumference	16 ±13	13±11	**11 ± 11**	9±10
I. Forearm circumference	14 ±11	11±9	**9 ± 8**	8±8
J. Arm length	15±21	15±12	**13 ± 12**	8±8
K. Inside leg length	26 ±19	23±18	**20 ± 19**	9±9
L. Thigh circumference	22 ± 18	19±16	**18 ± 17**	11±11
M. Calf circumference	18 ±13	14±12	**12 ± 12**	7±8
N. Ankle circumference	10 ±7	18±6	**6 ± 6**	5±5
O. Overall height	60 ± 45	50±42	**43 ± 41**	14±11
P. Shoulder breadth	15 ± 14	13±6	**6 ± 6**	12±11

Table 3.1: *Comparisons to variations of our method (*RF, CCA-RF-1, CCA-RF-2*) and ground truth, via various measurements. The measurements are illustrated in Figure 3.4. Errors are represented as Mean±Std. Dev and are expressed in millimeters.*

perform a more extensive validation, for 220 synthetic humans consisting of scans from the CAESAR database [Robinette and Daanen, 1999], and four real individuals' front and side images. We present the largest validation experiment with 1500 synthetic body meshes as well as real individuals.

Quantitative Experiments. We distinguish two test datasets, D1 and D2. D1 consists of 1500 meshes neither used to learn the parametric shape model nor to train the regression forests (RF) and D2 of 1000 meshes used to learn the parametric model but not to train the RF. These meshes consist of 50% males and 50% females, and are in roughly the same rest pose. In order to properly quantify our method, similar to [Boisvert et al., 2013], we perform 16 three-dimensional measurements on the meshes, which are commonly used in garment fitting as illustrated in Figure 3.4. For the measurements represented with straight lines, we compute the Euclidean distance between the two extreme vertices. The ellipses represent circumferences and are measured on the body surface. For each of the 16 measurements, we compute the difference between the one from the ground truth mesh and the estimated mesh. We report the mean error and the standard deviation for each of the measurements in Table 3.1. We name our main method *CCA-RF*, with CCA

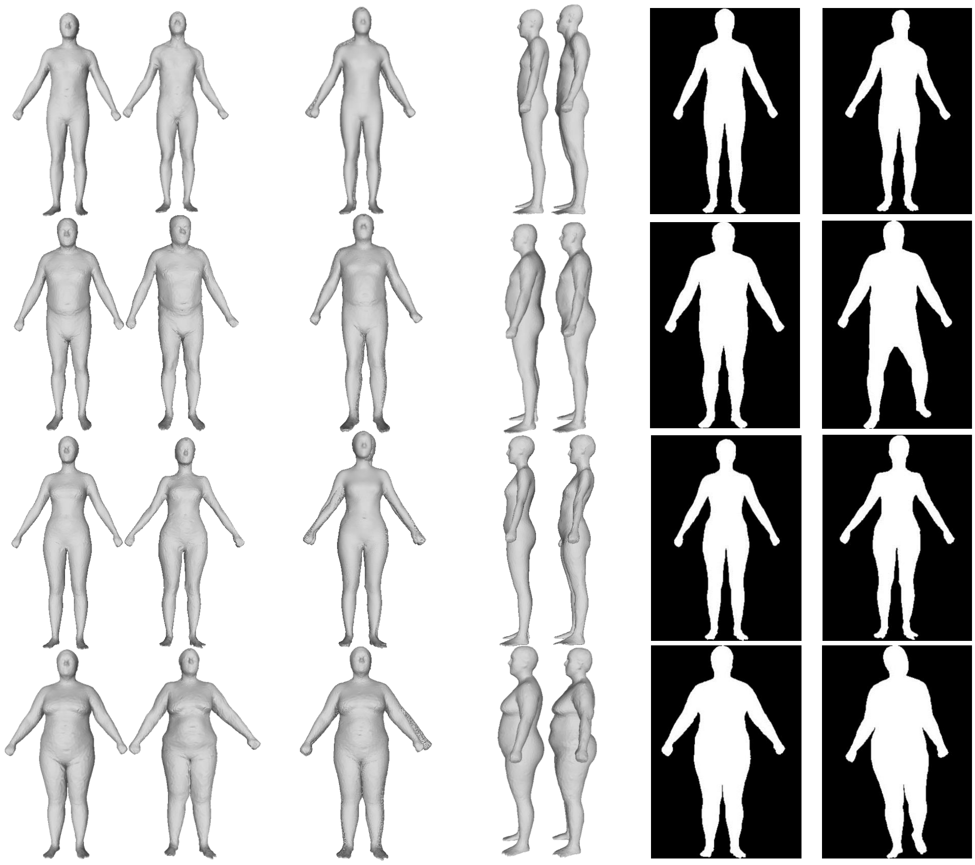


Figure 3.5: *Visual results for predictions on* 4 *test meshes. From left to right: predicted mesh, ground truth mesh, the two meshes frontally overlapping, the two meshes from the side view, silhouette from the predicted mesh, input silhouette.*

applied to the features before passing them to the random forest, specifically *CCA-RF-1* and *CCA-RF-2* respectively tested on D1 and D2. Similarly, *RF*, for the method trained on raw features and tested on D1. The last table column provides the ground truth (GT) mean errors for D1, computed between the original test meshes and their reconstructions obtained by projecting them into the learned PCA space. This provides a lower limit for the obtainable errors with our 20 parameters shape model.

Before analyzing the results, it is crucial to highlight the differences between the settings and goals of the methods we compare to. [Boisvert et al., 2013] employ a setting where the pose is fixed to a rest pose and the distance from the camera is also fixed. The shape estimation method is based on utilizing silhouettes from two different views (front and side), with the application of garment fitting in mind. The same setting is considered for the other two methods mentioned above [Chen et al., 2010;

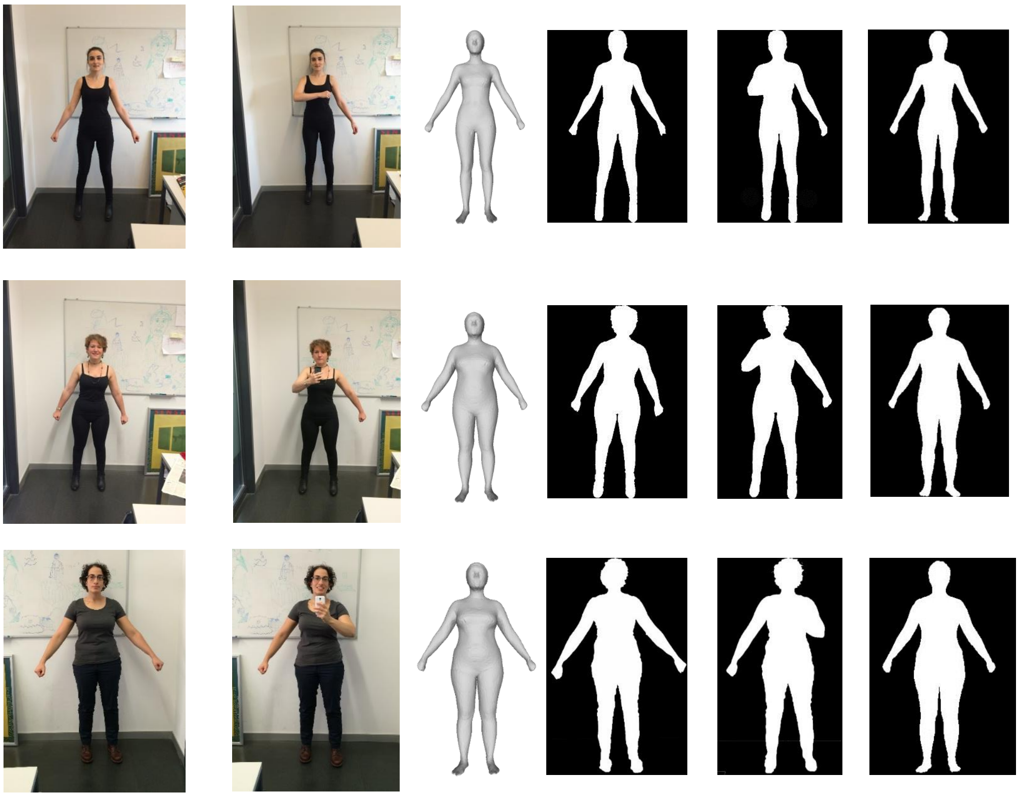


Figure 3.6: *Visual results for predictions on 3 females. From left to right: the two input images in a rest and selfie pose, the estimated mesh - same estimation is obtained for both poses, the two silhouettes from which features are extracted for each pose, the silhouette of the estimated mesh.*

Xi et al., 2007]. In contrast, we train and test for a more general setting, where we have a single silhouette as the input at test time, the pose can change, and no assumptions on the distance from the camera are made. Furthermore, our tests involve a significantly larger dataset with high variations.

Even though our method operates under a significantly more general setting than the previous works [Boisvert et al., 2013; Chen et al., 2010; Xi et al., 2007], with a single silhouette input and no distance information, it outperforms the non-linear and linear mapping methods, as shown in the single view Table 3.7 and two-view Table 3.8. The mean absolute error for all the models is 19.4 mm for *CCA-RF-1* and 16.18 mm for *CCA-RF-2*. The errors are very close to those of GT, illustrating the accuracy of our technique. Note that some errors for *CCA-RF-2* are smaller than those of the GT, due to the different training as explained above. The higher error for D1 is due to the body shapes that cannot be represented with the parametric model learned from the rest of the shapes. The error is higher for the overall height, due to the fixed scale in the training and testing silhouettes that we use. It is important to note the differences in errors between the *RF* and *CCA-RF-1*. There is an overall decrease of error when CCA is utilized, which shows that the projection with the CCA bases significantly improves prediction. Additionally, we evaluate the performance of our method when the input comes from a less favorable view, the side view, achieving an error of 22.45 mm which is very close to the one from only the frontal view. For completeness, we compare also to [Helten et al., 2013], who utilize an RGB-D camera for capturing the body shapes, and a full RMSE map per vertex to measure the differences. Using two depth maps, fitting to the pose and testing only on 6 individuals they report a mean error of 10.1 mm while we have a mean error of 19.19 mm on 1500 meshes.

Qualitative Results. In Figure 3.5, we show example samples from our tests. In each row, first the predicted mesh is shown along with the ground truth test mesh. Then, their overlap is illustrated. This is followed by the side views, and the silhouette of the estimated mesh and the input silhouette. Note that the input silhouettes are in different poses, but we show the estimated meshes in rest poses for easy comparisons. Our results are visually very close to the ground truth shapes even under such pose changes.

Finally, we show an experiment where real pictures of three females are taken in a rest and a selfie-like pose along with the estimated meshes in Figure 3.6. It is important to note that despite the pose change, the retrieved mesh for each person is the same. Another important observation is that even though the input is scaled to the same size, the estimated parameters yield statistically plausible heights, which turned out to be sufficient in obtaining an ordering based on relative height between the estimated meshes. We believe that this is due to the statistical shape model, where semantic parameters like height and weight are correlated in the PCA parameter space. To the best of our knowledge, no previous work can resolve this task. For example, in the work by [Sigal et al., 2007], the mesh needs to be scaled if no camera calibration is provided.

Poses, Views and Noise. We investigated accuracy in the presence of silhouette noise, various poses, and different or multiple views. We run the experiments with the data setup D1, explained above. For each experiment, we show the mean and standard deviation either of the accuracy gain or of the errors over all the body measurements in Table 3.2.

The first three columns show the *accuracy gain* of applying *CCA-RF* to the front view (F), side view (S) or when concatenating both views together (FS), as compared to *RF*. A larger gain is obtained in the side view as compared to the front view, due to additional information that is injected from the frontal

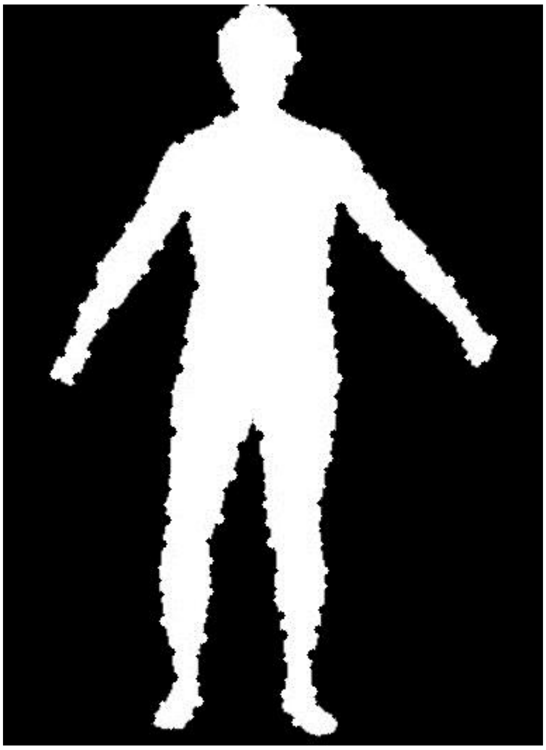


Figure 3.7: *Noisy silhouette.*

view (the most representative one) in the projected space. An even bigger gain is obtained if both views are utilized for training and testing. This is very important, as it shows that having potentially more views improves the predictor. In fact, we have observed that utilizing the same amount (100000) of training data, and training and testing on two views with the raw features, degrades the result as compared to just one view. This is alleviated with the CCA projection, improving the results as singular view noise in the data is removed.

The fourth column (VE), displays the *errors* obtained by testing on features extracted from a view 10 degrees rotated from the frontal view, for a *CCA-RF* trained on the frontal view. The column for (VG) displays the *gain* of *CCA-RF* over *RF* for the same scenario. The CCA-RF is again more accurate, however the error for both is generally low, implying that a classification error of the camera view of 10 degrees can be allowed in our system. (N) demonstrates the *error* due to random noise added to the silhouettes, as in Figure 3.7, showing robustness to noise to a certain extent. (P12) shows the *error* induced by training only on a rest pose, and testing on 12 different poses as in Figure 3.1, as compared to testing on the same meshes in a rest pose, and (P1) describes the same measurement, however by training on 12 poses and testing on a different unseen one, demonstrating robustness to

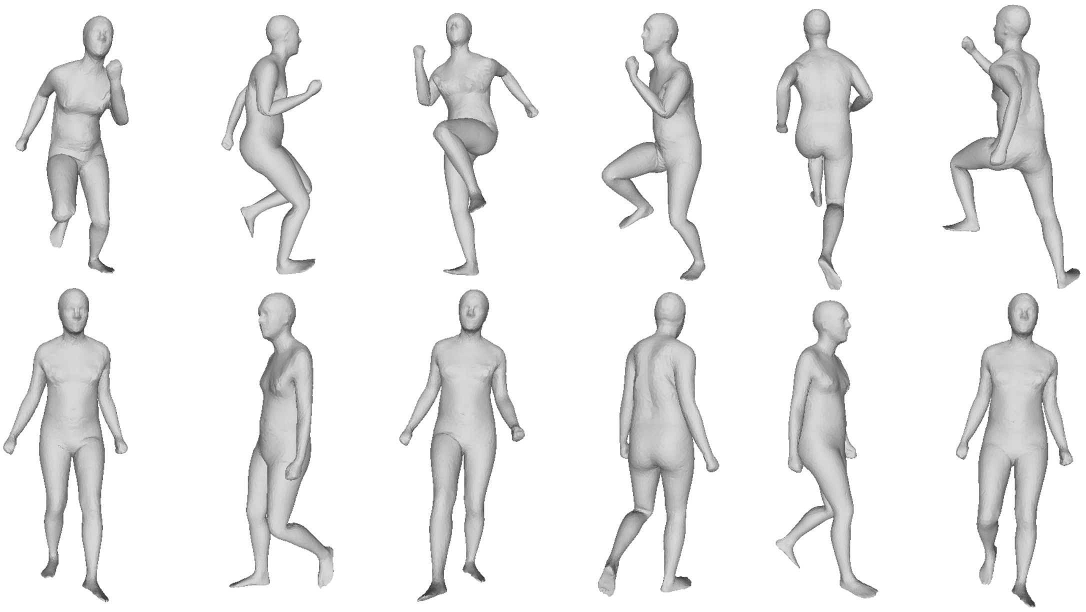


Figure 3.8: *(top) Meshes in running poses. (bottom) Meshes in walking poses.*

pose changes under minimal self occlusions. The last three columns demonstrate similar measurements, however, by increasing the articulations in the poses, with (W) consisting of poses from a walking sequence, (R) from a running sequence (Figure 3.8), and (PWR) combining all poses we have. The error increases in the latter case especially due to the introduction of poses with more self occlusions. However, when trained on individual sequences, the errors are lower, implying that for an application where a certain activity is known, one could adapt specialized regressors, especially due to the very fast training in the low dimensional spaces.

Known Camera Distance. In the previous experiments we made no assumption on the absolute scale of the silhouette from which we estimate the body shape. Here, we present results of an additional experiment performed under the assumption that the absolute scale of the silhouette is known, which is equivalent to knowing the distance of a person from the camera. The experiment is performed on Dataset 1, both with (CCA-RF-S-1) and without (RF-S-1) projecting the features onto the CCA bases. As it can be observed in Table 3.3, we get significant reductions in error for many measurements as compared to the case with no known absolute scale (RF-1), especially for the height. These errors are close to the ground truth (GT) error, which is the lowest error possible with the body shape model we use. Additionally better predictive results are noticed when the CCA is applied to the extracted features.

Measurement	(F)	(S)	(FS)	(VE)	(VG)	(N)	(P12)	(P1)	(W)	(R)	(PWR)
Mean (mm)	4.9	5.2	6.6	2.2	1.8	2.3	9.3	1.7	1.6	3.9	8.5
Std. Deviation (mm)	2.4	2.6	4.0	1.9	1.5	1.8	5.6	1.0	1.0	2.3	5.2

Table 3.2: *Columns 1-3 show accuracy gain of applying CCA for the Frontal, Side and Frontal Side view altogether, over raw features. (VE) shows the error due to 10 degree view change and (VG), the gain of applying CCA. (N) is the error due to silhouette noise. (P12) shows the error of testing on 12 poses different from the training one, and the rest (Columns 8-11) demonstrate the errors while gradually adding more difficult poses from the training ones. Mean and Std. Deviation is computed over all the body measurements.*

Algorithm Speed. The method is significantly faster than previous works, allowing for interactive applications. The method of [Boisvert et al., 2013] needs 6 seconds for body shape regression, 30 seconds for the MAP estimation, and 3 minutes for the silhouette based similarity optimization, with 6 seconds for their implementation of sGPLVM [Chen et al., 2010] (on an Intel Core i7 CPU 3GHz and single-threaded implementation). We, on the other hand, reach 0.3 seconds using a single threaded implementation on an Intel Core i7 CPU 3.4GHz (0.045 seconds for feature computation, 0.25 seconds for mesh computation, and 0.005 seconds for random forest regression), with even more speed-up opportunities as the feature computation and mesh vertices computation can be highly parallelized.

3.4 Shape Estimation with Neural Networks (HS-Net)

In the previous section, we proposed a method based on handcrafted features that in combination with random forest regressors and multi-view CCA can estimate the human body shape from silhouettes. In this section, we tackle the same problem with neural networks. To this end, we propose an accurate, fully automatic, and fast method that avoids handcrafted features and pose fitting by utilizing Convolutional Neural Networks (CNNs) to estimate the 3D body shape of a person. We have in mind applications such as garment fitting and personal measurements, hence pose variety, as compared to the previous section is more restricted. We analyze four possible cases as inputs to the network (a) a single frontal binary silhouette of the person scaled to a fixed size, needed in case of missing camera calibration information (b) the shaded image of the person scaled to a fixed size, with the motivation that shading withholds information complementary to the silhouette (c) a frontal silhouette which assumes known camera parame-

Measurement	RF-1	RF-S-1	CCA-RF-S-1	GT
A. Head circumference	16±13	14±11	13±11	13±9
B. Neck circumference	13±10	8±7	7±8	6±6
C. Shoulder-blade/crotch length	31±24	18±16	17±16	14±11
D. Chest circumference	38±31	28±25	25±23	24±24
E. Waist circumference	35±28	25±23	23±23	16±14
F. Pelvis circumference	33±26	19±17	18±17	14±12
G. Wrist circumference	10±8	6±6	6±6	5±5
H. Bicep circumference	16±13	10±11	10±10	9±10
I. Forearm circumference	14±11	10±9	10±8	8±8
J. Arm length	19±14	14±12	13±12	8±8
K. Inside leg length	26±19	18±15	16±13	9±9
L. Thigh circumference	22± 18	16±15	15±14	11±11
M. Calf circumference	18±13	11±9	11±9	7±8
N. Ankle circumference	10±7	7±7	7±7	5±5
O. Overall height	60±45	36±29	29±25	14±11
P. Shoulder breadth	15±14	13±15	13±13	12±11

Table 3.3: *Comparisons of the complementary results via various measurements. The measurements are illustrated in Figure 3.4. Errors represent Mean±Std. Dev and are expressed in millimeters. From left to right: Our results without applying CCA, the new results under the known scale assumption, the same with CCA applied to the features, the ground truth error defined as the error between the original model and its projection to the shape space spanned by the* 20 *parameters we utilize.*

ters and (d) two silhouettes simultaneously (front and side) under known distance from the camera, which in fact is a realistic assumption for the intended use-cases. In compliance with the applications, we make the assumption that people are wearing tight clothes and pose in a neutral stance that allows mild pose changes. Our method relies on advances made in the field of Neural Networks and a human body shape model [Anguelov et al., 2005], as in the previous section, obtained from thousands of 3D scans [Yang et al., 2014; Pishchulin et al., 2015]. Utilizing a CNN of roughly the size of AlexNet [Krizhevsky et al., 2012a] our method learns a global mapping from the input to the shape parameters. In fact, we learn an end-to-end regression from an input silhouette to 20 parameters that are used to recover the underlying body shape, as in Section 3.3. In addition, we show how to combine body views from two silhouettes to improve prediction over a single view. In order to comprehensively evaluate our method we validate it on thousands of body shapes, by computing error metrics on measurements

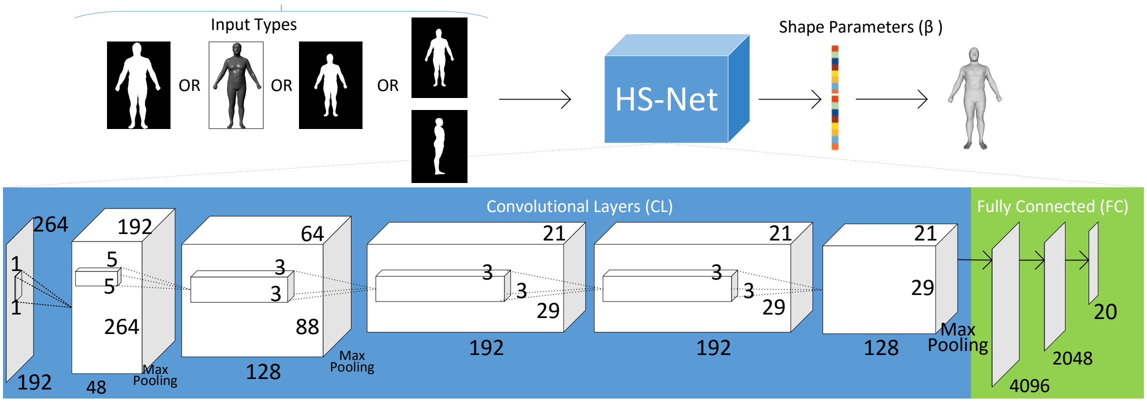


Figure 3.9: *Overview of the second method.* ***Top:*** *One of the four input types (scaled frontal silhouette to a fixed height, shaded image, one or two unscaled silhouettes) are fed to the Human Shape Network (HS-Net), to learn a global mapping and estimate human shape parameters (β), which can be used to reconstruct the human body shape.* ***Bottom:*** *The HS-Net architecture for the one view case.*

used in garment fitting, showing robustness to noise and comparing it to state-of-the-art methods that work under the same restrictive assumptions as (d).

3.4.1 Method Overview

Our goal is to design a fast and automatic system to accurately estimate the 3D human shape from silhouettes or images with shading information, for the garment fitting application in mind. More specifically, we would like to learn a global mapping from image evidence to parameters representing the 3D shape utilizing CNNs. With respect to the requirements (and privacy), we categorize image evidence in two groups: silhouette and shaded image. For the first, and least revealing case, extracting silhouettes in general images is not yet fully-automatic, but for our application it is realistic to assume that the person is wearing tight clothes and posing in front of a uniform color background, which simplifies the problem. For the second case on the other hand, the requirement is that the clothing is as minimalistic as possible, due to the fact that our training is based on naked body shapes (Section 3.2). In practice, a shaded image of a real person can be obtained by recovering the intrinsic image [Shen et al., 2011]. A neutral pose, allowing mild changes, is a reasonable assumption in both cases. While it is true that a 2D image withholds ambiguity per se, our goal is to generate the best approximating 3D mesh that explains the evidence.

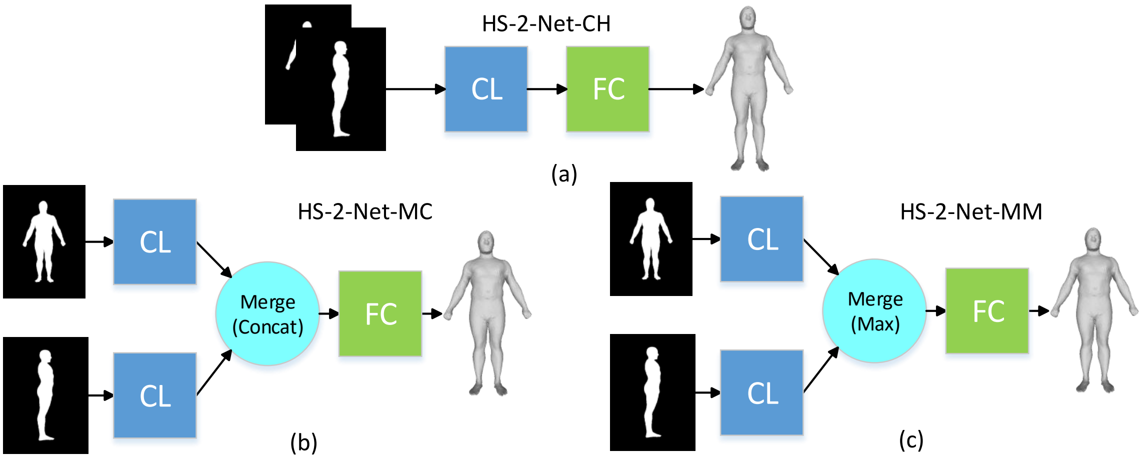


Figure 3.10: *The three architectures considered for two input silhouettes. (a) both silhouettes are input as two channels (b) each silhouette is input into two separate convolutional layer (CL) blocks and outputs of the CL are concatenated through a* Merge *layer (c) same scenario however with a* Max *operation performed instead.*

A system overview is depicted in Figure 3.9 (top), with the input being one of the four input types : scaled frontal silhouette to a fixed height, shaded image, one or two unscaled silhouettes, and as output the reconstructed 3D human shape. We pose shape estimation as an instance of supervised learning. Specifically, we solve a regression problem, where data is generated using a statistical human shape model (Section 3.1) based on SCAPE [Anguelov et al., 2005]. Utilizing parameters spanning from the human shape space, various meshes are reconstructed, from which we obtain silhouettes or shaded images. The parameters themselves are the output, and are used to reconstruct the 3D human shapes. In order to learn a global mapping from the data to the parameters, we do not need to handcraft features as in previous works [Ling and Jacobs, 2007; Sigal et al., 2007]. We also do not apply local fitting as in [Boisvert et al., 2013]. Instead, inspired by recent trends and outstanding results on various computer vision topics, we train CNNs (Section 3.4.2) from scratch, to find the most representative features and a mapping from the image evidence to the human shape. This results in a very fast and automatic system that clearly outperforms methods based on global mapping [Xi et al., 2007; Chen and Cipolla, 2009] and strongly competes with expensive methods that adopt local fitting [Boisvert et al., 2013].

3.4.2 Learning A Global Mapping

We pose the global mapping as an end-to-end regression problem, from 2D input image to shape parameters. We achieve this by training from scratch a CNN similar to that of AlexNet [Krizhevsky et al., 2012a] and adapting it to our inputs and regression task, as depicted in Figure 3.9 (bottom). Regarding the number of input images we distinguish two cases : A frontal single view image, coming in different forms, and two images simultaneously, from front and side.

3.4.2.1 Single View

The frontal view image can come in three forms. Firstly, a frontal binary silhouette of the human in a neutral pose, scaled to a fixed height is considered. This is the most general case, and assumes unknown camera calibration, hence the need for a fixed scaling. Second, if the camera parameters are known, e.g. when the person stands a known distance away from the camera, the input is a fixed size image of varying silhouette size and height. Estimating the real 3D shape from a 2D input silhouette is an ill-posed problem per se, due to the fact that a silhouette can represent various body shapes, even though we strive to reconstruct the shape that best explains it. Utilizing silhouettes only, has the advantage that no personal information is revealed, which is important for privacy protection. Allowing the problem to be a bit more relaxed, by adding further information, we lastly consider the case of using additional image cues such as shading, complementary to the scaled silhouette, similar to [Guan et al., 2009]. In order to synthetically generate training data, we render images with shading under Lambertian assumptions. In practice a similar result could be achieved by extracting the intrinsic image [Shen et al., 2011]. The input size for all the mentioned methods is set to 264×192 pixels. For each case, the single channel input images, along with the known shape parameters, are fed into our *Human Shape Network (HS-Net)*, which learns the mapping from input to the shape parameters β. *HS-Net* is a modification of Alexnet [Krizhevsky et al., 2012a] customized to a regression problem, our various input types, intended application and the available hardware. The network consists of five convolutional blocks, followed by three fully connected layers as illustrated in Figure 3.9 (bottom). Each layer is followed by an activation layer (ReLu). In addition, dropout layers are utilized between fully connected layers to avoid overfitting and max pooling is used after the first, second and fifth convolutional blocks. The network is trained from scratch, since the available pre-trained models are geared towards classification and RGB or grayscale images, while we tend to learn regression from binary images. We experimented with different optimization algorithms and observed that the best results were obtained using *RMSProp* [2] and *Adadelta* [Zeiler, 2012]. We decided to utilize the Adadelta optimizer due to its capacity to automatically adjust the learning rate and prevent it from becoming too small.

3.4.2.2 Two Views

In compliance with the realistic scenario of estimating the body shape and the body parts measurements as accurately as possible, we additionally opted for the usage of two silhouettes simultaneously, where the person is seen from a full frontal and side view. This setting also assumes known camera parameters, same as the methods we compare to [Xi et al., 2007; Chen and Cipolla, 2009; Boisvert et al., 2013], which translates to knowing the distance from the camera. One of the challenges of this case is how to combine multiple view inputs in a way that the convolutional network can use them coherently. We explore and evaluate three different approaches to achieve this. The first approach, utilizes a model architecture very similar to the one view case, however the input images from the different views are stacked along the channel dimension to form two channel images, see Figure 3.10 (a). These two channel images are then fed into the network for training. By visualizing the output filters (Figure 3.17) for different layers on various test images, we observed that the network learns some filters more pronounced towards frontal views, while others favor the side views. For the second approach, the architecture differs from the previous case, in that we add a *Merge* layer similar to the view pooling layer of [Su et al., 2015], after two sets of convolutional layers with shared weights for each view, followed by fully connected layers. The input images from each view are fed into two separate five layer convolutional networks and merged using a concatenation operation, see Figure 3.10 (b). The third approach distinguishes from the second one in that the merge layer performs a *Max* operation over each dimension, see Figure 3.10 (c). The motivation behind the last two approaches, was to allow the network to separately learn features from individual images and then fuse them more discriminatively through a merging layer. The merge layer with max operation improves learning and subsequently the estimation accuracy (see Table 3.4) over the two channel network, as it combines evidence at a later stage of learning. All three methods lead to improvements over the one view case, which we demonstrate in Section 3.4.3, where the merging with max operation performs the best.

[2] http://www.cs.toronto.edu/ tijmen/csc321/slides/lecture_slides_lec6.pdf

3.4.3 Validation and Results

Our method targets the application of human body shape and body parts estimation. In order to assess its reliability, one can not rely only on the visual reconstruction of the mesh. Rigorous quantitative experiments are necessary, especially of measurements over various important body parts. If the latter can be estimated accurately, fitting clothes virtually or even buying clothes online becomes more intuitive and appealing. Measuring different body parts consistently in real datasets is difficult, as even the most trained individuals are reported to deviate up to 10 mm [Gordon et al., 1989]. Hence, we evaluate on synthetic meshes, obtained by fitting a parametric model to real people scans from the CAESAR dataset, similar to the methods we compare to [Boisvert et al., 2013]. Performing the evaluations on this dataset, in addition to the shapes being very close to the real ones, has the advantage that they are in full correspondence. Thus, it becomes easy to automatically measure various body parts. Additionally, the poses adopted from the real human scans, deviate from the neutral pose specified by experimenters while they are being scanned, Figure 3.11 (top-left). These meshes are quite realistic and in compliance with the variation of the poses that people adopt for our target applications. Different openings of the arms, legs and even shoulders can be noticed. In our experiments, we apply the same measurements as in [Boisvert et al., 2013] and the previous section, repeated for viewers courtesy in Figure 3.11 (top-right). For our evaluation we use 1500 meshes and 4 real people on 16 body measurements, which to the best of our knowledge is the most complete one so far, as compared to related work. [Boisvert et al., 2013] evaluate on 220 meshes and 4 real people, [Xi et al., 2007] on 24 meshes and two real people, [Sigal et al., 2007] for two measurements only on two subjects and [Balan et al., 2007] for silhouette errors and height measurement on a few individuals.

3.4.3.1 General Training and Set-Up Details

For each of the 100000 generated meshes, Section 3.2, we generate silhouettes from frontal and side views, as well as shaded images under lambertian assumptions with Maya[3]. As a preprocessing step, the images are centered, normalized to the [0,1] interval and fixed to the 264 × 192 pixels resolution for all the cases. The resolution was chosen such that it neither impedes learning of shape variations, nor is too big, due to the hardware and time constraints we had. We use 95000 images for training and 5000 for the net-

[3] http://www.autodesk.com/products/maya/

work validation. As explained above, the testing is performed on 1500 unseen samples and real human ones. The network architecture is detailed in Figure 3.9 (bottom) for the one view input case. For the various experiments that we perform, we change the networks as explained in Section 3.4.2 and adopt the following nomenclature : *HS-1-Net-S* and *HS-1-Net* for the scaled and unscaled input silhouette and *HS-1-Net-Im* for the scaled shaded image input. Training usually converges between 15-25 epochs depending on the experiment. The batch size was set to 32, to not be a proper divisor of the number of training samples per epoch, which is equal to half of total training samples. This provides an easy way to simulate shuffling without hitting memory constraints for such big datasets. We also experimented with batch normalization [Ioffe and Szegedy, 2015] right after the convolutional layers, resulting in slight error increase. Applying batch normalization after the fully connected layers though, caused the network to converge to constant functions.

We experimented with the RMSprop, Adagrad [Duchi et al., 2011] and Adadelta [Zeiler, 2012] optimizers, in order to minimize the manual learning rate adjustments. We observed that RMSprop (with an initial learning rate of 0.001) and Adadelta (with decay rate of 0.95) converged faster than Adagrad, also with a smaller test error. Thus, all the reported experiment results are for the models trained using Adadelta. We experimented with the squared loss, with and without multiplying the last fully-connected layer by custom weights. The weights are set to be the eigenvalues of the covariance matrix obtained from PCA, during the data generation step Section 3.1 and normalized to 1, such that we emphasize large scale changes in 3D body shapes, as for the previous method. As expected, using squared loss with custom weights performed better. For all the networks, we utilized Glorot uniform weight initialization [Glorot and Bengio, 2010].

For the two view case, we used the best performing network configurations from the one view case, however the architectures were modified to fit the input extension, as shown in Figure 3.10. The two selected views were the frontal and the side one. We also distinguish between three cases here : *HS-2-Net-CH* for the input silhouettes passed as two channels of a single image, *HS-2-Net-MM* for separately training the two inputs as different single channel images and applying a merge layer, that performs a max operation over each dimension right after the output of the last convolutional layers (CL), and *HS-2-Net-MC* for the same architecture that concatenates the output of CL, instead of max operation. All the CL have shared weights.

3.4.3.2 Quantitative Experiments

We perform 16 3D measurements on the test meshes which consist of males and females in roughly equal numbers, similar to [Boisvert et al., 2013]. The measurements are illustrated in Figure 3.11 (top-right) and are widely used in garment fitting. We compute the Euclidean distance between two extreme vertices for the straight line measurements, while for the ellipsoidal ones, the perimeter is computed on the body surface. For each measurement we calculate the difference between the value estimated and the ground truth, and report the mean error and standard deviation computed over the error values for all the test meshes in Table 3.4. Additionally, we show how the mean error over all measurements varies, for each different input type that we consider, in Figure 3.11. *HS-2-Net-MM* has the lowest error of 4.02 mm, as compared to 11 mm of [Boisvert et al., 2013], which utilizes a more expensive local fitting algorithm. For completeness, we compare to the work of [Helten et al., 2013], that utilizes an RGB-D camera for capturing the body shapes, and a full RMSE map per vertex to measure the differences. They report an error of 10.1 mm, evaluating on 6 individuals from two depth maps, while we report an error of 7.4 mm on 1500 meshes.

We observed that using weights with squared loss function increases the accuracy of the model. The model trained on silhouettes with known camera parameters performs significantly better than the one with unknown camera calibration. The shaded images network *HS-1-Net-Im*, performs also slightly better than the corresponding silhouette one *HS-1-Net-S*, implying that shading information possibly improves the shape estimation accuracy, but could also be related to added information due to grayscale input as opposed to a binary one. Lastly, *HS-2-Net-CH* demonstrates more accuracy for the ellipsoidal errors while *HS-2-Net-MC* for the euclidean ones, despite their overall similar performance. In comparison to the other methods, our network clearly outperforms the global methods [Xi et al., 2007; Chen and Cipolla, 2009] (Figure 3.8), and strongly compares to the method from [Boisvert et al., 2013]. Adding a second view gives better results than a single view with noticeable improvements in the height and waist estimation.

We perform three additional experiments to show the extensibility of the approach: (1) an experiment with more pronounced poses (2) partially visible silhouettes and (3) images rendered under specularity assumption. All the following experiments were performed assuming unknown camera calibration, hence scaled silhouettes similar to *HS-1-Net-S*.

Poses. We generated 95000 meshes in 10 different poses, Figure 3.12 and

Measurement	HS-1-Net-S	HS-1-Net-S-Im	HS-1-Net	HS-2-Net-MC	HS-2-Net-CH	HS-2-Net-MM
A. Head circumference	4±4	4±4	2±4	2±3	2±3	**2±3**
B. Neck circumference	8±5	6±4	3±1	2±1	3±1	**2±1**
C. Shoulder-blade/crotch length	20±15	20±14	7±7	5±6	4±5	**3±5**
D. Chest circumference	13±7	13±6	4±1	2±1	4±2	**2±1**
E. Waist circumference	19±13	19±13	8±7	6±7	8±7	**7±5**
F. Pelvis circumference	19±14	19±12	6±5	5±4	6±5	**4±4**
G. Wrist circumference	5±3	5±3	3±2	2±1	3±2	**2±2**
H. Bicep circumference	8±4	8±3	2±1	2±1	2±1	**2±1**
I. Forearm circumference	7±4	6±3	2±1	2±1	2±1	**1±1**
J. Arm length	12±8	12±8	6±4	5±4	5±4	**3±2**
K. Inside leg length	20±14	19±13	12±8	13±9	11±7	**9±6**
L. Thigh circumference	13±8	12±7	8±5	7±4	7±4	**6±4**
M. Calf circumference	12±7	11±6	5±2	5±2	4±2	**3±1**
N. Ankle circumference	6±3	5±2	3±1	2±1	3±1	**2±1**
O. Overall height	50±39	49±37	20±15	19±15	16±13	**12±10**
P. Shoulder breadth	4±4	3±4	3±4	2±4	2±4	**2±4**

Table 3.4: *Error comparisons on body measurements for the various inputs and presented training modalities. The measurements are illustrated in Figure 3.11 (top-right). Errors are represented as Mean±Std. Dev and are expressed in millimeters. Our best achieving method* HS-2-Net-MM *is highlighted.*

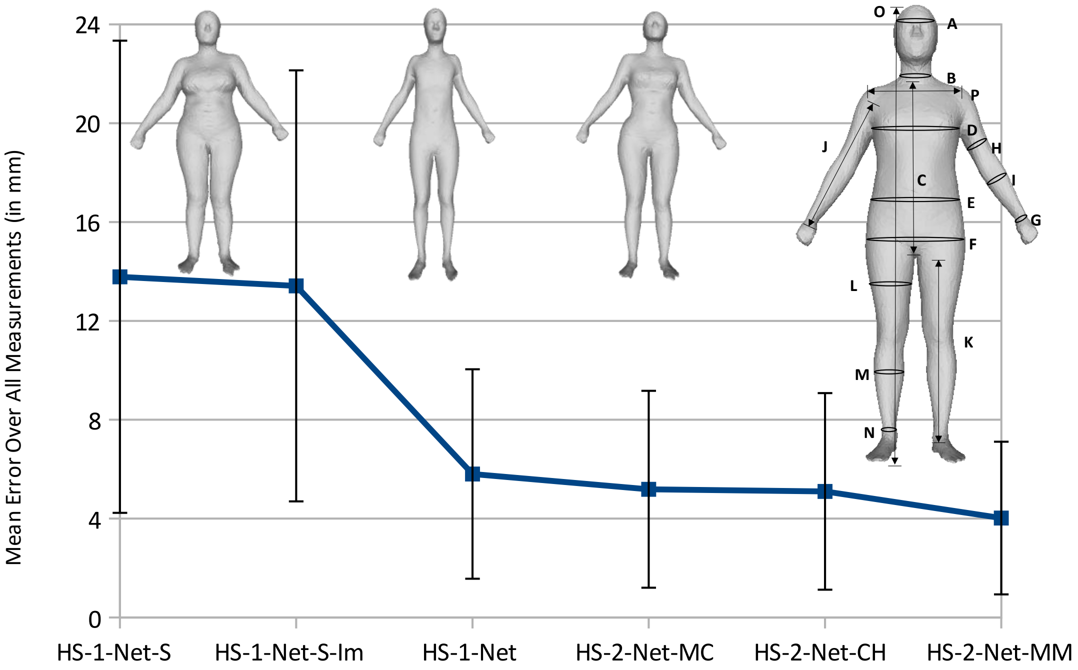


Figure 3.11: *Mean error over all measurements for different input types. (top-left) 3 test meshes in slightly changing poses. (top-right) Illustration of the body measurements (A - P) on the template mesh.*

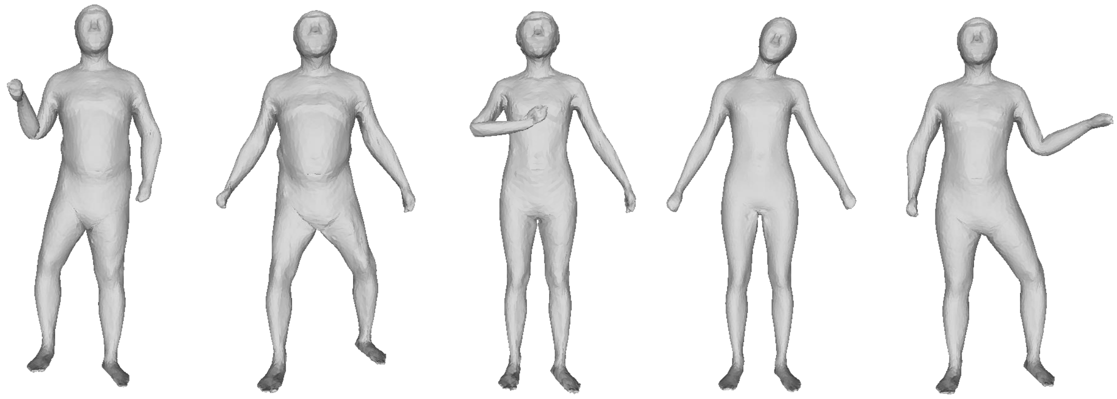


Figure 3.12: *Meshes in various poses*

compare to the results of *HS-1-Net-S* Table 3.5, Column 1. Except for the Arm Length (J) measurement (which has an added error of 40 mm), we observe very similar results. The added error for J can be due to the fact that we include poses similar to the one from Figure 3.12 (middle). This pose introduces self occlusions, handling of which is a limitation of this method.

Half Body. We think that an interesting stress case is that of partially visible people, e.g. an upper body selfie, or without loss of generality, a female wearing a skirt that would impede the reliable estimation of the lower body part. Directly applying *HS-1-Net-S* to such inputs resulted in increased errors, implying that with the current training set, it is not possible to accurately estimate full bodies from partial silhouettes.

To tackle this, we train a network similar to *HS-1-Net-S*, however with silhouettes from the upper half of the body, as input, (Figure 3.13, Left). As illustrated in Table 3.5, Column 1 and 2, the results for this network (*HS-1-Net-SH*) are similar to that of the full body case (*HS-1-Net-S*). We think that the ability of this network to accurately estimate full body shapes from partial (half body) images is due to the high correlation between different body measurements, as represented by the shape parameters β.

Finally, we train a network that can have either full or half body silhouette as possible input (*HS-1-Net-SHS*), and test separately on each input type. We demonstrate the results in Table 3.5, Column 4 and 5, and notice that the performance for both full and half body silhouettes is similar to that of the network trained separately on each input type (*HS-1-Net-S* and *HS-1-Net-SH*), with a maximum added error of 2 millimeters for individual measurements. This shows that our network can simultaneously learn to accurately esti-

Measurement	HS-1-Net-S	HS-1-Net-SH	HS-1-Net-SHS-Half	HS-1-Net-SHS-Full	HS-1-Net-Im	HS-1-Net-IP
A. Head circumference	4±4	5±5	5±5	5±5	4±4	4±4
B. Neck circumference	8±5	8±5	8±5	8±5	6±4	6±4
C. Shoulder-blade/crotch length	20±15	20±15	21±16	21±16	20±14	**17±12**
D. Chest circumference	13±7	14±7	15±7	14±6	13±6	13±8
E. Waist circumference	19±13	19±14	20±15	20±14	19±13	19±14
F. Pelvis circumference	19±14	20±14	21±15	20±14	19±12	19±14
G. Wrist circumference	5±3	6±3	6±4	6±4	5±3	5±3
H. Bicep circumference	8±4	8±4	9±4	9±4	8±3	8±4
I. Forearm circumference	7±4	7±4	7±4	7±4	6±3	6±4
J. Arm length	12±8	12±8	13±8	12±7	12±8	12±8
K. Inside leg length	20±14	19±13	19±13	19±13	19±13	19±14
L. Thigh circumference	13±8	13±8	13±8	12±8	12±7	12±8
M. Calf circumference	12±7	12±6	12±6	12±6	11±6	11±6
N. Ankle circumference	6±3	6±3	5±3	5±3	5±2	5±3
O. Overall height	50±39	51±38	52±39	52±39	49±37	**47±37**
P. Shoulder breadth	4±4	4±4	4±4	4±4	3±4	4±4

Table 3.5: *Results of the additional experiments with errors represented as Mean±Std. Dev (in mm). All experiments are done with the input scaled to a fixed height. Experiments (from* left *to* right*): full body silhouettes; only half body silhouettes; trained on both half and full body silhouettes but tested only on half; same as previous but tested on full body silhouettes; grayscale images with shading under Lambertian assumptions; grayscale images with Phong shading (we highlight the most significant decrease in error due to Phong shading as compared to the Lambertian case).*

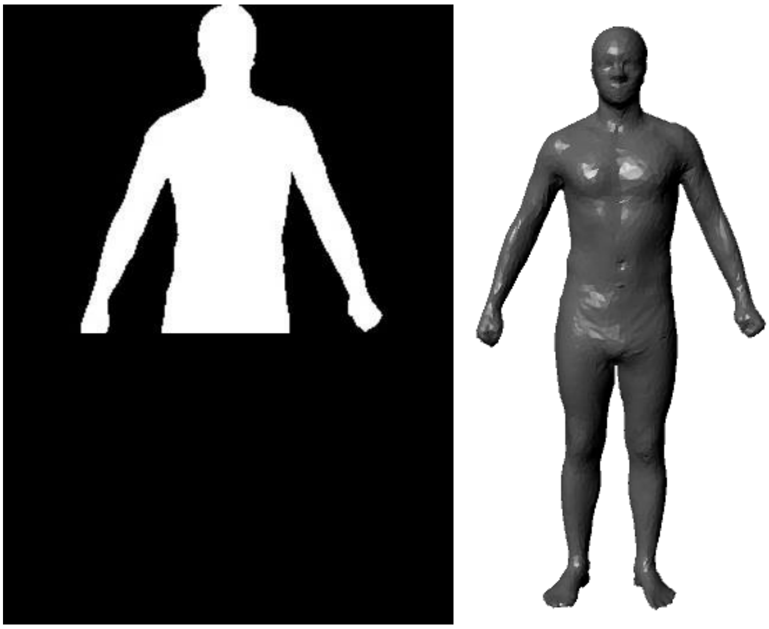


Figure 3.13: *Examples of a half body silhouette (*left*) and a grayscale image with Phong shading (*right*)*

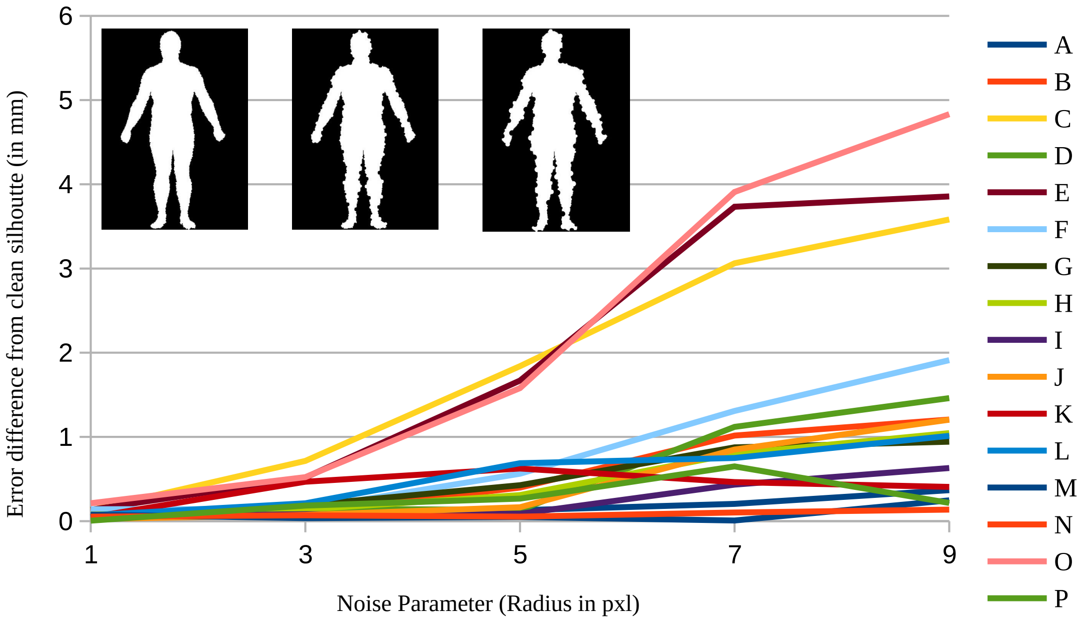


Figure 3.14: *Error plots for each of the body measurements (A - P) when noise is applied, as compared to clean silhouettes. (top-left)* 3 *silhouettes with noise parameters* 1, 5 *and* 9. *(Figure best seen in colors).*

mate 3D body shapes, from both full and partial images, unlocking further applications.

Images under Phong Shading. We perform a final experiment to illustrate the effects of specularity on the grayscale shaded images, e.g. selfies with flash on or people wearing clothes of specular materials. For this, we train a network (*HS-1-Net-IP*) with images extracted using Phong shading (Figure 3.13, right) and observe a slight improvement over the experiment with shading under Lambertian assumptions (*HS-1-Net-Im*), Table 3.5 Column 5 and 6. We think that this is because of the extra information from specularity in Phong shading. This again shows that the network is able to utilize the added information from shaded images.

Noise. Due to the imperfection of silhouette extraction algorithms, we evaluated the robustness of our model under the influence of noise. We apply noise to the silhouette by randomly eroding or dilating the silhouette at the border, with filters of various radii, evaluating it for 1,3,5,7 and 9 pixels. We plot the errors of each body measurements and show examples of noisy silhouettes for a radius of 1, 5 and 9 pixels in Figure 3.14. The method achieves performance similar to the noiseless case within a reasonable noise radius, where even for the highest noise parameter the maximum error (in the height) is below 5 mm, implying robustness to this noise type.

3.4.3.3 Qualitative Results

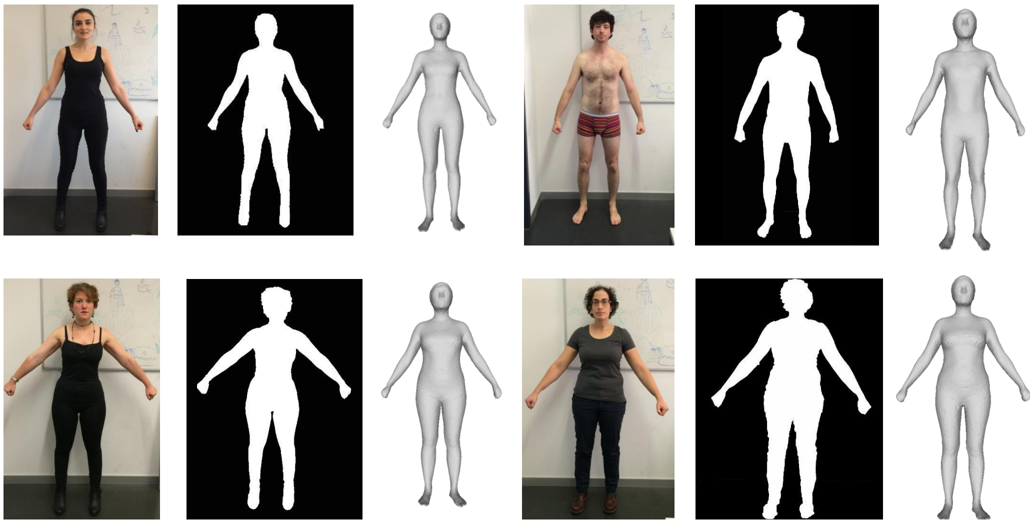


Figure 3.15: *Mesh reconstruction for 4 real subjects in mildly varying poses. (left) Input image (middle) Extracted Silhouette (right) Reconstruction of the estimated shape.*

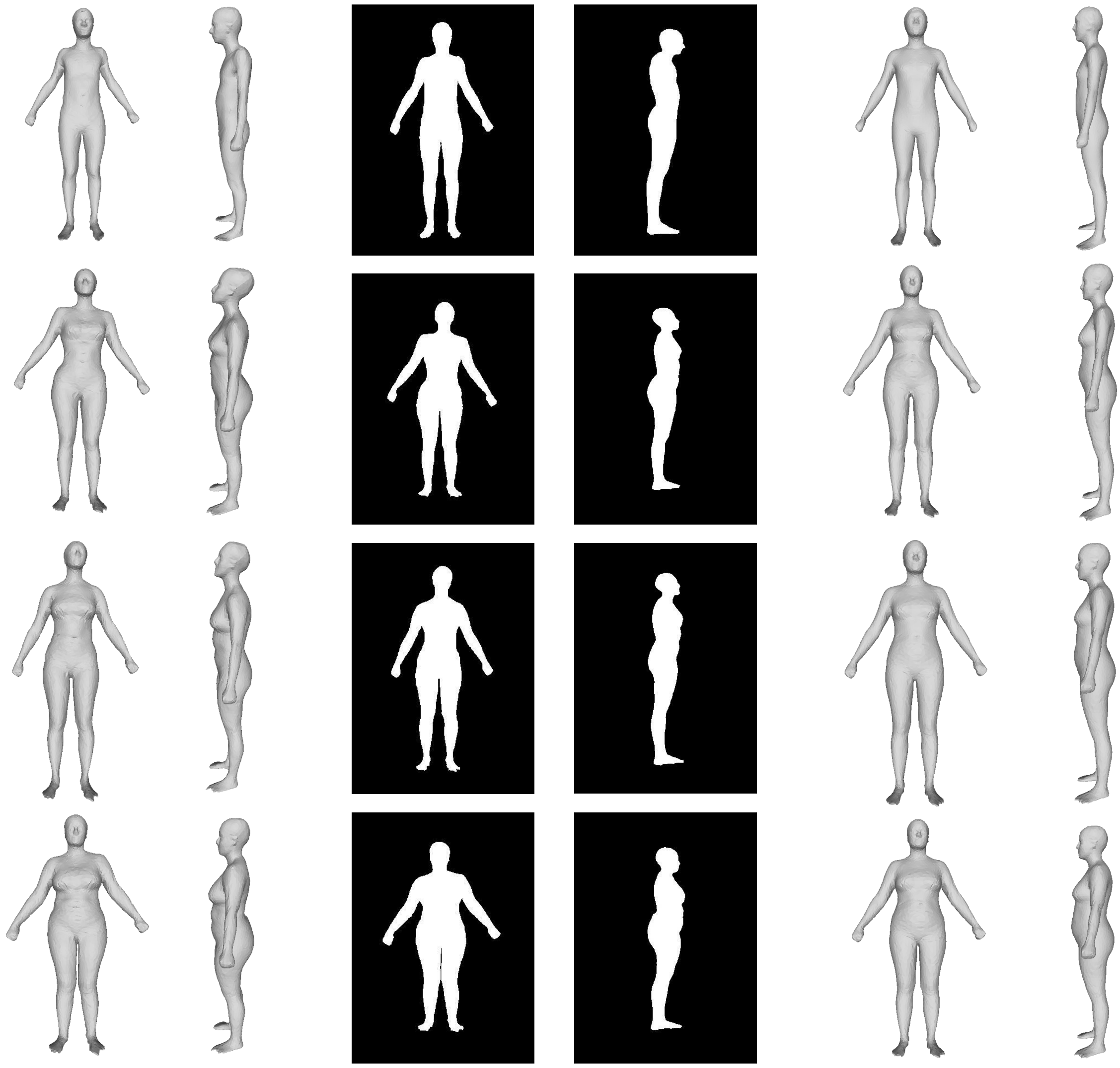


Figure 3.16: *Reconstructed meshes for various test inputs for two views network* HS-2-Net-MM. *(*left *to* right*) Original meshes (front and side view); Silhouettes (front and side view); Reconstructed Meshes (front and side view)*

We demonstrate results of frontal body shapes, obtained by applying *HS-1-Net-S* over scaled silhouettes, extracted from images of real people in Figure 3.15. The individuals adopt a neutral pose, however please note the variations in the arms and legs openings. Our method manages to reconstruct accurate shapes, also backing up our claim that mild pose changes do not affect our robustness.

Synthetic Meshes. Estimated meshes for four synthetic test examples with slight pose changes are presented in Figure 3.16.

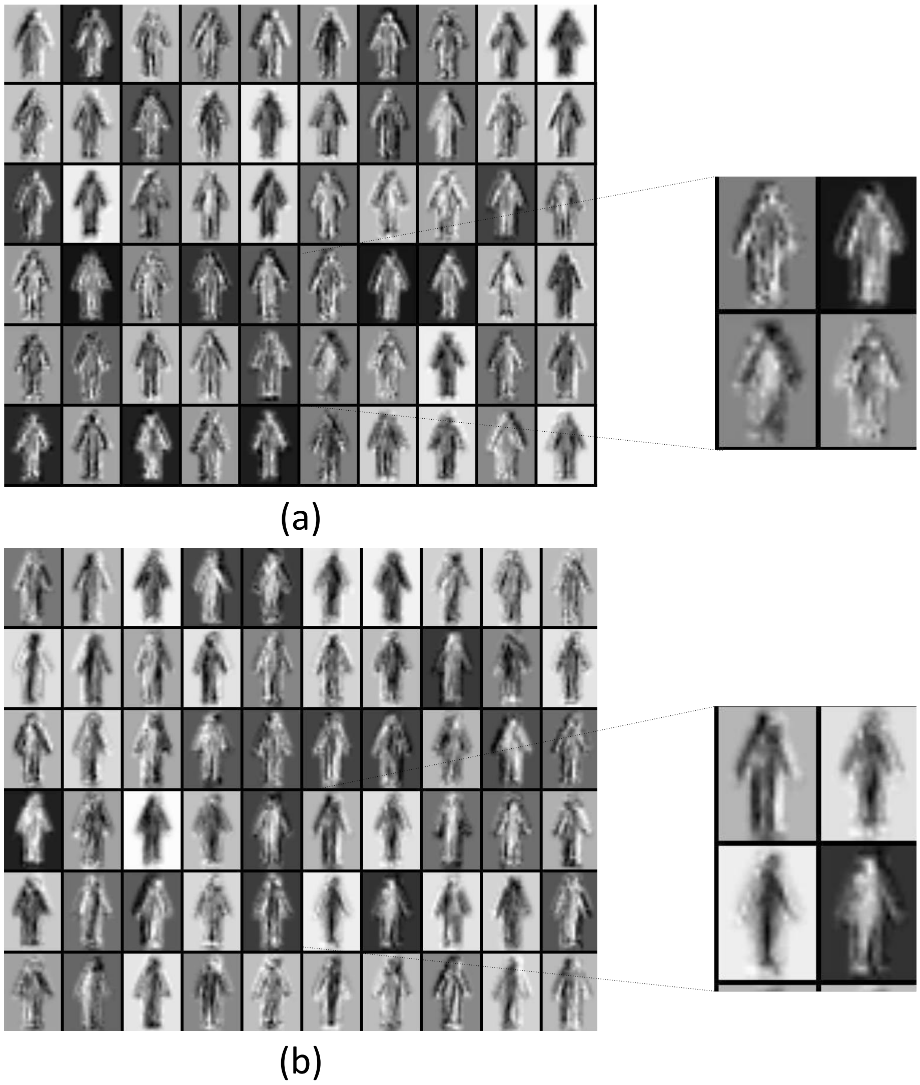


Figure 3.17: *Visualization of randomly chosen 60 convolutional filters on a test input for 3rd layer (*left*) and zoomed in view of four selected filters (*right insets*) for (a) one view case (HS-1-Net) and (b) two views case (HS-2-Net-CH).*

Filters. We illustrate some of the convolution filters from the third convolutional layer, visualized on a sample test input (Figure 3.17). For the zoomed in version of two views treated as two channels of an image (HS-2-Net-CH) (Figure 3.17(b), *right inset*), we can observe that the side view is more pronounced in the *bottom-left* (Figure 3.17(b), *right inset*) filter while the front view is more pronounced in the *top-left* (Figure 3.17(b), *right inset*) one. The other two have combined features from both views.

3.4.3.4 Method Speed

We conducted our experiments on an Intel(R) Core(TM) i7 CPU 950 3.0 GHz with NVIDIA GTX 980TI (6GB) GPU. The training code was implemented in Python using Keras framework [4] with Tensorflow as backend. The usual training time is around 30 minutes per epoch and the testing time was about 0.2 seconds per image. Generating a mesh from the estimated parameters takes around 0.25 seconds (significant further speed-up is possible via parallelization of this step). Our full algorithm runs in 0.45 seconds and is significantly faster than the methods we compare to with 3 minutes and 36 seconds for the full optimization of [Boisvert et al., 2013] and 6 seconds for the global mapping of [Chen et al., 2010].

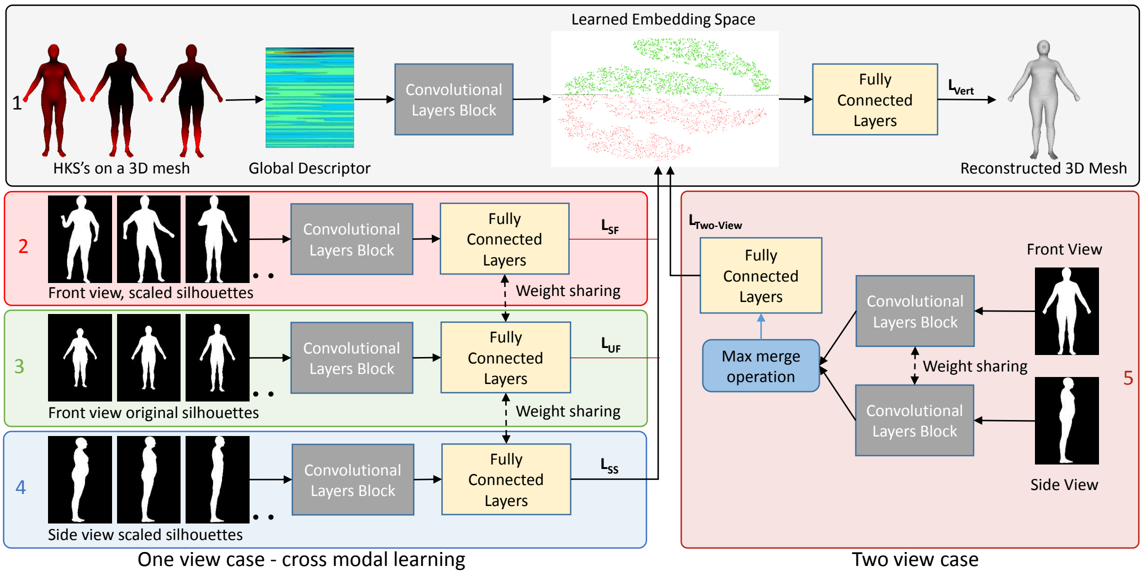


Figure 3.18: *Overview of the third method. (1) HKS-Net: HKS projected features as input, generates an embedding space which is mapped to 3D meshes. (2),(3) and (4) Three modes of the Cross-Modal Neural Network (CMNN) (only (2) is used at test time). (5) An architecture that requires both views at test time. The method uses either CMNN or (5), depending on the number of available input views.*

3.5 The Generative and Cross-Modal Estimator of Body Shape

In this section, we leverage from the shortcomings of the previously introduced methods and propose a more robust method based on CNN-s

[4] http://keras.io/

to estimate the human body shape. We attempt to achieve it from a single or multiple silhouettes of a human body with poses compliant with the two main applications explained in the previous sections: virtual garment fitting assuming a neutral pose, as in [Chen and Cipolla, 2009; Boisvert et al., 2013] and Section 3.4, and shape from individually taken pictures or "Selfies" (e.g. through a mirror or a long selfie stick), assuming poses that exhibit mild self occlusion (Section 3.3). Compared to state-of-the-art in this domain, we achieve significantly higher accuracy on the reconstructed body shapes and simultaneously improve in speed if a GPU implementation is considered, or obtain similar run-times to the method from Section 3.3 on the CPU. This is achieved thanks to a novel Neural Network architecture (Figure 3.18) consisting of various components that (a) are able to learn a body shape representation from 3D shape descriptors and map this representation to 3D shapes, (b) can successfully reconstruct a 3D body mesh from one or two given body silhouettes, and (c) can leverage multi-view data at training time, to boost predictions for a single view at test time through cross-modality learning.

As referred and compared to in the previous sections, there exist methods that attempt to find a mapping from silhouettes to the parameters of a statistical body shape model [Anguelov et al., 2005], utilizing handcrafted features as in Section 3.3, silhouette PCA representations [Chen and Cipolla, 2009] possibly with local fine tuning [Boisvert et al., 2013], or CNN-s as in Section 3.4. Based on the obtained parameters, a least squares system is solved to obtain the final mesh. We also use CNN-s to learn silhouette features, but unlike in the previous section, we first map them to a shape representation space that is generated from 3D shape descriptors (Heat Kernel Signature (HKS) [Sun et al., 2009]) invariant to isometric deformations and maximizing intra-human-class variation, and then decode them to full body vertex positions. Regressing to this space improves the predictions and speeds up the computation.

In Section 3.3, we demonstrated how to boost features coming from one view (scaled frontal) during test time, utilizing information from two views (front and side) at training time, by projecting features with Canonical Correlation Analysis (CCA) [Hotelling, 1936]. CCA comes with shortcomings though as (1) it computes a linear projection, (2) it is hard in practice to extend it to more than two views, and (3) suffers from lack of scalability to large datasets as it has to "memorize" the whole training data set. As part of the method presented in this section, we propose an architecture (which we call Cross-Modal Neural Network (CMNN)) that is able to overcome the mentioned challenges, by first generating features from various views separately, and then combining them through shared layers. This leads to improvements in predictive capabilities with respect to the uni-modal case. Abstracting away from silhouettes, this network can be used as-is for other tasks where multiple views on the data are present, such as image and text retrieval, or audio and image matching.

3.5.1 Method Overview

The main goal of our method is to accurately estimate a 3D body shape from a silhouette (or two) of a person adopting poses in compliance with two applications - virtual cloth fitting and self shape monitoring. On par with the related work, we consider either a single frontal silhouette scaled to a fixed size (no camera calibration information) with poses exhibiting mild self occlusions, or two views simultaneously (front and side, scaled or unscaled) of a person in a neutral pose. We propose to tackle this problem with a deep network architecture (Figure 3.18). Our network is composed of three core components: a generative component that can invert pose-invariant 3D shape descriptors, obtained from a multitude of 3D meshes (Section 3.1) to their corresponding 3D shape, by learning an embedding space (Section 3.5.2); a cross-modal component that leverages multi-view information at training time to boost single view predictions at test time (Section 3.5.3); and a combination of losses to perform joint training over the whole network (Section 3.5.4).

3.5.2 Generating 3D Shapes from HKS (HKS-Net)

The first part of our architecture aims at learning a mapping from 3D shape descriptors to 3D meshes via a shape embedding space. We start by extracting Heat Kernel Signatures (HKS) and then projecting them to the eigenvectors of the Laplace-Beltrami operator to obtain a global descriptor. This is used to learn the embedding space, as well as an inverse mapping that can generate 3D shapes in a neutral pose given the corresponding descriptor.

Heat Kernel Signatures (HKS). Let a 3D shape be represented as a graph $G = (V, E, W)$, where V, E and W represent the set of vertices, edges, and some weights on the edges, respectively. The weights encode the underlying geometry of the shape, and can be computed via standard techniques from the mesh processing literature [Sun et al., 2009]. Given such a graph constructed by connecting pairs of vertices on a surface with weighted edges, the heat kernel $H_t(x, y)$ is defined as the amount of heat that is transferred from the vertex x to vertex y at time t, given a unit heat source at x [Sun et al., 2009]:

$$H_t(x,y) = \sum_i e^{-\lambda_i t} \phi_i(x) \phi_i(y), \tag{3.4}$$

where H_t denotes the heat kernel, t is the diffusion time, λ_i and ϕ_i represent the i^{th} eigenvalue and the corresponding eigenvector of the Laplace-Beltrami operator, respectively, and x and y denote two vertices. Heat kernel has various nice properties that are desirable to represent human body shapes under different poses. In particular, it is invariant under isometric deformations of the shape, captures different levels of detail and global properties of the shape, and it is stable under perturbations [Sun et al., 2009].

The heat kernel at vertex x and time t can be used to define the heat kernel signature $HKS_x(t)$ for this vertex:

$$HKS_x(t) = H_t(x,x) = \sum_i e^{-\lambda_i t} \phi_i^2(x). \tag{3.5}$$

Hence, for each vertex x, we have a corresponding function $HKS_x(t)$ that provides a multi-scale descriptor for x. As the scale (i.e. t) increases, we capture more and more global properties of the intrinsic shape. In practice, the times t are sampled to obtain a vector $HKS_x(t_j), j \leq J$ for each vertex x. In our technique, we use $J = 100$ time samples. Then for each t_j, we can form the vectors $\mathbf{h}_j := [HKS_{x_1}(t_j), HKS_{x_2}(t_j) \cdots]^T$.

Projected HKS Matrix. To learn the embedding space, the HKS for all vertices at a given time t_j are projected onto the eigenvectors of the Laplace-Beltrami operator in order to obtain a 2D image capturing the global intrinsic shape. Specifically, we compute a matrix $\mathbf{M}$ with $\mathbf{M}_{ij} = \phi_i^T \mathbf{h}_j$, i.e. the dot product of the i^{th} eigenvector of the Laplace-Beltrami operator and the heat kernel vector defined over the vertices for time t_j. Since we use 300 eigenvectors ϕ_i, we thus get a 300 × 100 matrix $\mathbf{M}$.

This is then used as input to the top part of our network (that we call HKS-Net, Figure 3.18 (1)) to learn an embedding space of about 4000 dimensions, by minimizing the per-vertex squared norm loss L_{Vert}. A simplistic representation of this embedding, computed utilizing T-SNE [Maaten and Hinton, 2008], is also presented in Figure 3.18, where female meshes are depicted in green dots and male meshes in red. An important property of HKS-Net is that we can reconstruct a 3D mesh in a neutral pose when HKS-Net is presented with a computed $\mathbf{M}$. Hence, HKS-Net can invert the HKS descriptors. Although we do not utilize this property in the scope of this work, we believe that this could be a valuable tool for geometry processing applications. But instead, we use the embedding space with 4000 dimensions as the target space for the cross-modal silhouette-based training of our network, which we explain next.

3.5.3 Cross-Modal Neural Network (CMNN)

The second component thus consists of finding a mapping from silhouettes to the newly learned embedding space. We generate five types of silhouettes that can be referred to as *modes* : frontal view scaled in various poses with minor self occlusion, frontal view scaled in a neutral pose, side view scaled in a neutral pose and front and side view unscaled in a neutral pose (Figure 3.18).[5] Here, unscaled implies known camera calibration, and scaled means we resize the silhouettes such that they have the same height. Frontal means that the plane of the bones that form the torso is parallel to the camera plane, and side is a 90 degrees rotated version of the frontal view. At test time, our results are not affected by slight deviations from these views. We thus center the silhouettes, and resize them to an image of resolution 264×192 before inputting them to the CMNN. We, of course, do not expect to use all the modes/views at once during testing, but our intention is to leverage the vast amount of data from various modes at training time for robust predictions at test time.

We start by training a network similar in size to the one from the method of Section 3.4 (5 convolutional and 3 dense layers), with AdaMax optimizer [Kingma and Ba, 2014], and learning rate of e^{-4}, to map each mode individually to the embedding space by minimizing squared losses on the 4000 embedding space parameters (Figure 3.18 (2),(3) and (4) with the respective losses L_{SF}, L_{UF} and L_{SS}). As shown in Table 3.7, we already achieve better results for the one-view case as compared to related works. This pre-training serves as an initialization for the convolutional weights of the Cross-Modal Neural Network (CMNN). The final cross-modal training is performed by starting from the weights given by the pre-training, and optimizing for the shared weights for the fully connected layers with a combined loss, e.g. for scaled-front and scaled-side we minimize $L_{SF} + L_{SS}$, or for three modes, the loss is $L_{SF} + L_{UF} + L_{SS}$.

The idea is to let each single convolutional network compute silhouette features separately first, and then correlate these high-level features at later stages. We observed that we obtain significant improvements when cross-correlating various combinations of 2 modes and 3 modes during training (Table 3.7) as compared to the uni-modal results. CMNN offers several advantages as compared to CCA. First, we obtain a non-linear correlation between high-level features. Second, we can add as many modes as we want,

[5] Please note that throughout the text *mode* and *view* are used interchangeably to emphasize different ways of representing the same 3D mesh.

while it is not trivial to correlate more than two spaces with CCA. Finally, we do not need to store all training data in memory as in the case of CCA.

One of the main focuses of this section is estimating a 3D shape for the scaled-frontal case, with similar application scenarios as in the previous sections. Hence, our desired test time mode, i.e. the desired input at test time, is a silhouette from a frontal view with unknown camera parameters. Without loss of generality, we consider the unscaled-frontal and scaled-side as the other additional modes. Note that this can be extended with more views and further variations.

3.5.4 Joint Training

Finally, we would like to jointly train HKS-Net and CMNN for obtaining the final generative network. This is done by using all losses at the same time and backpropagating them to all parts of the architecture. We thus perform a joint training with the HKS-Net by minimizing $L_{SF} + L_{UF} + L_{SS} + L_{Vert}$. This training not only improves the mappings from 2D silhouettes to the 3D meshes, but also improves the generative capabilities of the HKS-Net by learning a better embedding space (Table 3.7 and Table 3.8).

Two-View Case. We also consider the case when two complementary input silhouette images (front and side) are given simultaneously, which further allows comparisons to some of the related works [Xi et al., 2007; Chen and Cipolla, 2009; Boisvert et al., 2013] and the method from Section 3.4. For this case, we mainly consider neutral poses. As the architecture, we use the HKS-Net along with a network similar to the one used in Section 3.4 (Figure 3.18 (5)) where, unlike in CMNN, the weight sharing is performed at early stages during convolutions, and the last convolutional layers are merged through a max-pooling operation. This is then trained with the sum of squared losses $L_{Two-View} + L_{Vert}$, on the embedding space and the mesh vertex locations, as before. Similarly, the mapping to the embedding space is decoded to a 3D mesh space through a forward pass in the dense layers of the HKS-Net. This achieves better results than that of the previous method, due to the newly learned embedding (Table 3.8).

3.5.5 Experiments and Results

We have run an extensive set of experiments to ensure the reliability of our technique. In this section, we report results of our qualitative and quantitative tests, with thorough comparisons to the state-of-the-art. In order

Name	Training Input	Test Input	Architecture
SF-1	Scaled Frontal View (SFV), Neutral Pose	SFV	$\boxed{2}$
SF-1-P	SFV, Various Poses	SFV	$\boxed{2}$
SFU-1	SFV, Unscaled Frontal View (UFV)	SFV	$\boxed{2}$ $\boxed{3}$
SFS-1	SFV, Scaled Side View (SSV)	SFV	$\boxed{2}$ $\boxed{4}$
SFUS-1	SFV, UFV, SSV	SFV	$\boxed{2}$ $\boxed{3}$ $\boxed{4}$
SFUS-HKS-1	SFV, UFV, SSV, projected HKS (PHKS)	SFV	$\boxed{1}$ $\boxed{2}$ $\boxed{3}$ $\boxed{4}$
SF-SS-2	SFV, SSV	SFV, SSV	$\boxed{5}$
UF-US-2	UFV, Unscaled Side View (USV)	UFV, USV	$\boxed{5}$
UF-US-HKS-2	UFV, USV, PHKS	UFV, USV	$\boxed{1}$ $\boxed{5}$

Table 3.6: *Nomenclature for the various experiments. For the architecture components highlighted in colors and with numbers, please refer to Figure 3.18.*

to quantitatively assess our method, we perform experiments on synthetic data similar to previous works [Boisvert et al., 2013], as in the previous two sections, by computing errors on the same 16 body measurements widely utilized in tailor fitting, as shown in Figure 3.19 (repeated for viewer‘s courtesy). Since all the methods we compare to, as well as ours, make use of the same shape model [Anguelov et al., 2005], the comparisons become more reliable through these measurements on estimated meshes in full correspondence.

From the combined datasets [Yang et al., 2014; Pishchulin et al., 2015] of meshes fitted to real body scans, where duplicate removal is ensured as in the previous methods we presented, we set 1000 meshes apart for testing, and utilize the rest for generating the human body model and training data (Section 3.1). For these left-out meshes we then extract HKS descriptors and silhouettes in various views and poses. We apply LBS [Lewis et al., 2000] to deform the meshes into desired poses compliant with our applications.

We run the methods from the previously introduced methods on the silhouettes extracted from these meshes, while for others methods we compare to [Xi et al., 2007; Chen and Cipolla, 2009; Boisvert et al., 2013], we report the numbers from their experiments performed on similar but fewer meshes (around 300). In addition to comparisons with the state-of-the-art, we thoroughly evaluate the added value of each component in our network. In the end we conclude with qualitative results and run-time evaluations.

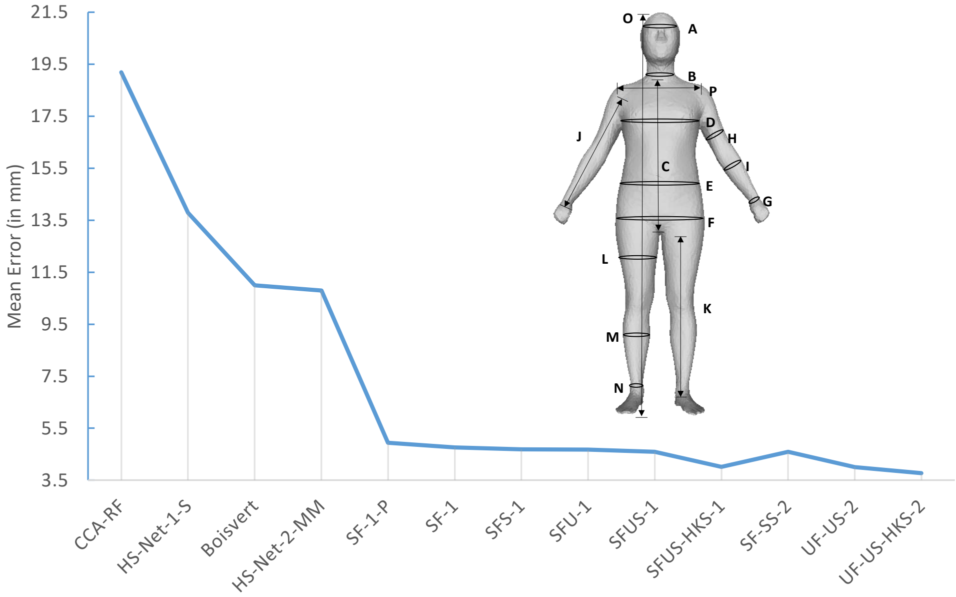


Figure 3.19: *Plot of the mean error over all body measurements illustrated on a mesh, for the methods from Table 3.7 and Table 3.8.*

3.5.5.1 Quantitative Experiments

The 16 measurements are calculated as follows: straight line measurements are computed by Euclidean distances between two extreme vertices, while for the ellipsoidal ones, we calculate the perimeter on the body surface. For each measurement, we report the mean and standard deviations of the errors over all estimated meshes with respect to the ground truth ones.

We report errors when only the frontal view silhouette is utilized at test time in Table 3.7, and if both frontal and side view silhouettes are available at test time in Table 3.8. For both tables, we distinguish between two cases: known camera distance (unscaled) and unknown camera distance (called scaled in the subsequent analysis, since we scale the silhouettes to have the same height in this case, as elaborated in Section 3.5.3). The nomenclature for our experiments is summarized in Table 3.6. Note that for all methods in the tables, the errors are for a neutral pose, except for $SF-1-P$, where we show the error measures when we train and test using different poses. The mean error over all body measurements for the methods we consider is depicted in Figure 3.19. Our best mean error for the one view cross-modal case is 4.01 mm and for the two-view case is 3.77 mm, showing a very high accuracy for the tasks we consider. These are significantly better than the mean error of the second method 19.19 mm, first method 10.8 mm, 11 mm [Boisvert et al.,

Measurements	SF-1-P	SF-1	SFS-1	SFU-1	SFUS-1	SFUS-HKS-1	HS-Net-1-S (Method 2)	CCA-RF (Method 1)
A. Head circumference	4.3±3.5	3.9±3.1	3.7±2.9	3.7±2.9	3.9±2.9	**3.1±2.6**	4±4	8±8
B. Neck circumference	2.2±1.8	2.3±1.8	2.3±1.8	2.3±1.8	2.2±1.7	**2.1±1.7**	8±5	7±7
C. Shoulder-blade/crotch length	6.2±4.9	6.1±4.8	5.3±4.2	5.3±4.1	5.4±4.1	**4.9±3.8**	20±15	18±17
D. Chest circumference	6.7±5.4	6.7±5.3	5.9±4.9	5.9±4.7	5.8±4.8	**5.8±4.8**	13±7	25±24
E. Waist circumference	8.1±6.1	7.8±6.2	7.5±5.9	7.5±5.9	7.5±5.7	**6.4±5.2**	19±13	24±24
F. Pelvis circumference	9.3±7.5	8.8±7.2	8.4±6.7	8.2±6.6	8.1±6.5	**7.1±5.9**	19±14	26±25
G. Wrist circumference	2.1±1.7	2.1±1.7	1.9±1.6	1.9±1.6	1.9±1.6	**1.7±1.5**	5±3	5±5
H. Bicep circumference	3.9±3.1	3.3±2.6	2.9±2.4	2.9±2.4	2.9±2.5	**2.9±2.5**	8±4	11±11
I. Forearm circumference	3.1±2.4	2.9±2.3	3.1±2.3	2.7±2.3	2.9±2.3	**2.6±2.2**	7±4	9±8
J. Arm length	4.1±3.1	3.8±2.9	3.3±2.5	3.3±2.5	3.2±2.5	**2.9±2.4**	12±8	13±12
K. Inside leg length	7.3±5.1	6.8±5.2	6.2±4.8	6.5±4.9	5.7±4.5	**5.4±4.3**	20±14	20±19
L. Thigh circumference	6.3±4.9	6.3±5.5	5.8±4.9	5.7±4.7	5.8±4.8	**5.8±4.9**	13±8	18±17
M. Calf circumference	3.6±2.9	3.5±3.1	3.3±2.7	3.3±2.6	3.5±2.8	**2.9±2.5**	12±7	12±12
N. Ankle circumference	2.1±1.5	2.1±1.7	1.9±1.5	1.8±1.4	2.1±1.5	**1.6±1.3**	6±3	6±6
O. Overall height	12.6±9.9	12.4±9.9	11.2±8.6	10.9±8.4	10.4±8.1	**9.8±7.7**	50±39	43±41
P. Shoulder breadth	2.3±1.9	2.3±1.8	2.2±1.2	2.2±1.9	2.1±1.7	**1.9±1.7**	4±4	6±6

Table 3.7: *Body measurement errors comparison with shapes reconstructed from one scaled frontal silhouette. The nomenclature is presented in Table 3.6. Last two columns show the results of the state-of-the-art methods. The measurements are illustrated in Figure 3.19 (top-right). Errors are expressed as Mean±Std. Dev in millimeters. Our best achieving method* SFUS-HKS-1 *is highlighted.*

Measurements	SF-SS-2	UF-US-2	UF-US-HKS-2	HS-2-Net-MM (Method 2)	[Boisvert et al., 2013]	[Chen et al., 2010]	[Xi et al., 2007]
A. Head circumference	3.9±3.2	3.3±2.6	**3.2±2.6**	7.4±5.8	10±12	23±27	50±60
B. Neck circumference	1.9±1.7	2.0±1.6	**1.9±1.5**	5.3±3.1	11±13	27±34	59±72
C. Shoulder-blade/crotch length	5.1±4.1	4.3±3.5	**4.2±3.4**	9.9±7.0	4±5	52±65	119±150
D. Chest circumference	5.4±4.8	5.8±4.3	**5.6±4.7**	19.1±12.5	10±12	18±22	36±45
E. Waist circumference	7.5±5.7	7.6±5.9	**7.1±5.8**	18.4±13.2	22±23	37±39	55±62
F. Pelvis circumference	8.0±6.4	8.0±6.4	**6.9±5.6**	14.9±11.3	11±12	15±19	23±28
G. Wrist circumference	1.9±1.6	1.6±1.4	**1.6±1.3**	3.8±2.7	9±12	24±30	56±70
H. Bicep circumference	3.0±2.6	2.6±2.1	**2.6±2.1**	6.5±4.9	17±22	59±76	146±177
I. Forearm circumference	3.0±2.4	2.9±2.1	**2.2±1.9**	5.5±4.2	16±20	76±100	182±230
J. Arm length	3.3±2.6	2.4±1.9	**2.3±1.9**	8.1±6.4	15±21	53±73	109±141
K. Inside leg length	5.6±5.1	4.3±3.8	**4.3±3.8**	15.6±12.4	6±7	9±12	19±24
L. Thigh circumference	5.8±5.1	5.1±4.3	**5.1±4.3**	13.7±10.8	9±12	19±25	35±44
M. Calf circumference	3.9±3.2	3.1±2.1	**2.7±1.9**	8.5±6.5	6±7	16±21	33±42
N. Ankle circumference	2.1±1.5	1.6±1.1	**1.4±1.1**	4.6±3.2	14±16	28±35	61±78
O. Overall height	10.6±8.6	7.2±6.1	**7.1±5.5**	25.9±20.4	9±12	21±27	49±62
P. Shoulder breadth	2.2±1.8	2.1±1.8	**2.1±1.8**	5.6±3.9	6±7	12±15	24±31

Table 3.8: *Same as in Table 3.7, however with shapes reconstructed from two views at the same time. Last four columns show the results of the other state-of-the-art methods for the same task. Our best achieving method* UF-US-HKS-2 *is highlighted.*

2013], and 10.1 mm [Helten et al., 2013], even though some of these methods operate under more restrictive assumptions. Our best results, that achieve state-of-the-art, are highlighted in bold.

For the **one view** case (Table 3.7), one can see that as we go from uni-modal to cross-modal training, by using multiple views at training time and sharing weights in the fully connected layers, the errors constantly decrease. We show the effect of adding a side scaled view only ($SFS-1$), an unscaled frontal view only ($SFU-1$), and combining all three ($SFUS-1$). The lowest errors are achieved through joint training ($SFUS-HKS-1$) of the CMNN and HKS-Net (Section 3.5.4). In this case, not only the accuracy of predictions from silhouettes, but also the accuracy of the HKS-Net itself is improved as compared to when it is separately trained, reducing the mean error over all the meshes from 4.74 to 3.77 mm. We further report results when different poses are applied on the test meshes ($SF-1-P$), in contrast to all other methods considered. Even in this case, the errors do not differ much from the neutral pose case ($SF-1$), implying robustness to variations for the pose space we consider.

For the **two view** case, we compare to the results of the works that require two views at test time [Boisvert et al., 2013; Xi et al., 2007; Chen and Cipolla, 2009], as well as the second method from Section 3.4. We utilize the same camera calibration assumptions, and again achieve significant improvements in accuracy ($UF-US-HKS-2$), due to the new shape embedding space jointly trained with the prediction network. For the two view-case, we do not test on multiple poses, since the previous works we compare to are also tested on neutral poses for this particular application. One interesting observation here is that the results for the single view cross-modal case ($SFUS-1$ in Table 3.7) are comparable to, and in some measurements even better than those of the two-view network ($SF-SS-2$ in Table 3.8). Since no joint training was performed in either case, and the loss for both cases is in the shape embedding space, this demonstrates the importance of the shared fully connected layers and cross-modal training for boosting prediction performance at test time.

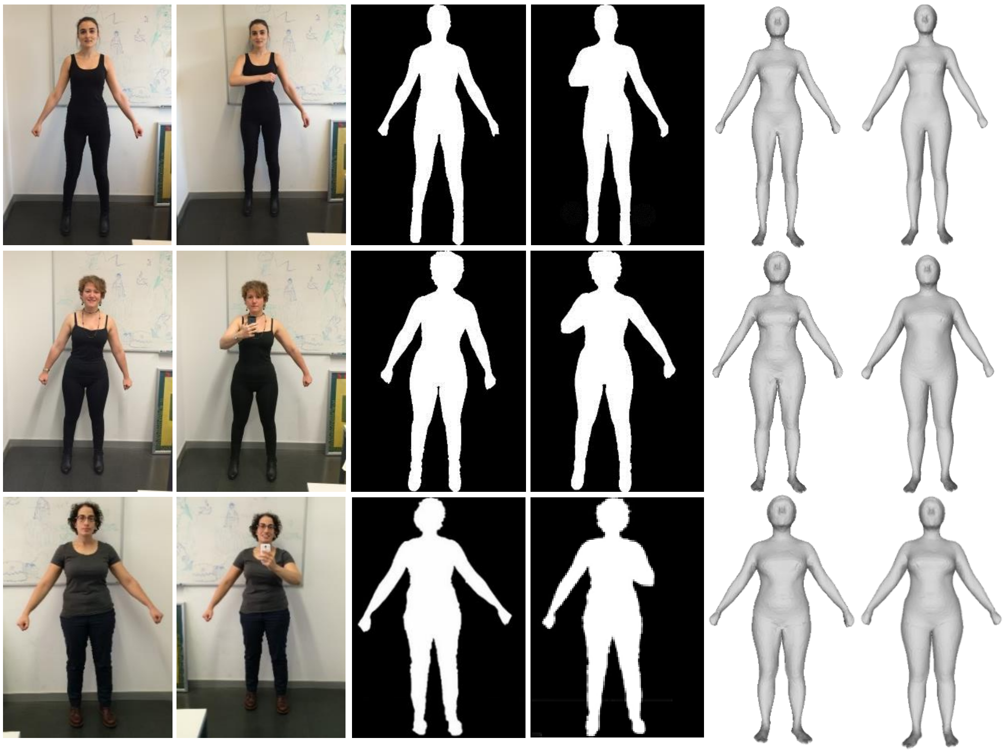


Figure 3.20: *Results for predictions on the test images from Method 1 of Section 3.3. From left to right: the two input images in a rest and selfie pose, the corresponding silhouettes, the estimated mesh by our method $SF-1-P$, and by the first method.*

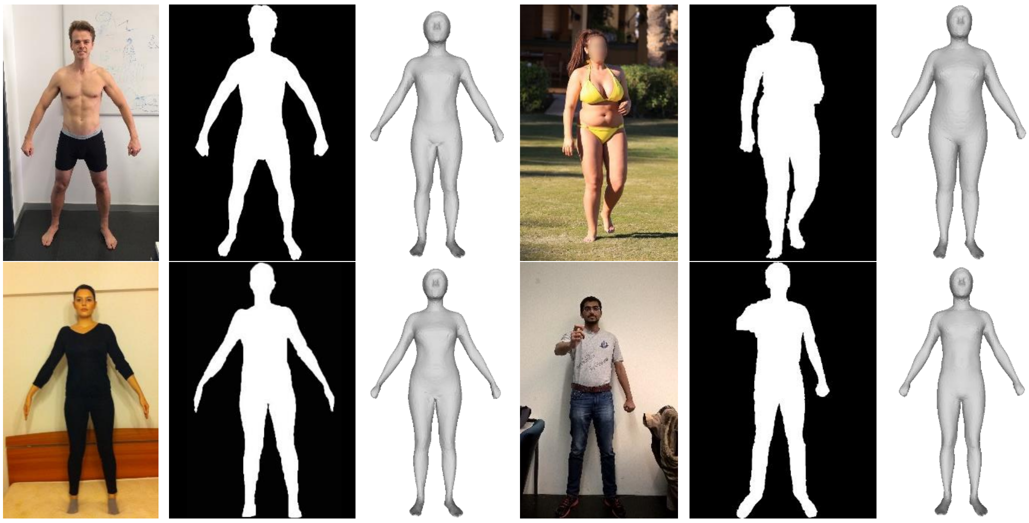


Figure 3.21: *Predictions on four test subjects in different poses and with clothes. From left to right: input image, the corresponding silhouette, the estimated mesh by our method $SF-1-P$.*

3.5.5.2 Qualitative Experiments

We evaluate our method on three test subjects in a neutral and selfie pose (also presented in Section 3.3), and four new subjects with other poses. As it can be observed in Figure 3.20, our reconstructions resemble the real individuals more closely, as compared to those achieved with the first method (last column), especially for the second subject. The results in Figure 3.21 illustrate harder cases, where the silhouettes differ more from those of the training data due to clothing, poses, and occlusions. Our results still explain the silhouettes well for all cases. We additionally show mesh overlays over the input images, applied also to the method from [Bogo et al., 2016a] below.

3.5.5.3 Mesh Overlaps

We show the estimated meshes (third column in gray), utilizing our method $SF-1-P$, for three input photos from the individuals of Section 3.3 along with the estimated meshes (last column in pink) from the method of [Bogo et al., 2016a], in Figure 3.22. We also show the meshes estimated with our method and that from [Bogo et al., 2016a] overlaid on the input images, in Figure 3.23 and Figure 3.24 respectively, in order to emphasize the differences in estimations from both methods. It can be noticed that our method gives more accurate estimations for these individuals, with a tendency of the method from [Bogo et al., 2016a] to overestimate, also visible by the difference in silhouette projection, especially on the torso and around the waste in Figure 3.24. Additionally, in Figure 3.23 we show the overlay on the scanned mesh of another individual from the testing dataset. Please note that we did not apply linear blend skinning to change the neutral pose to fit perfectly the input silhouette, in order to enhance the fact that for the application of automatic body measurement a fixed pose is not needed. The method from [Bogo et al., 2016a] on the other hand attempts to more accurately estimate the 3D body pose, which is also the main purpose of their work.

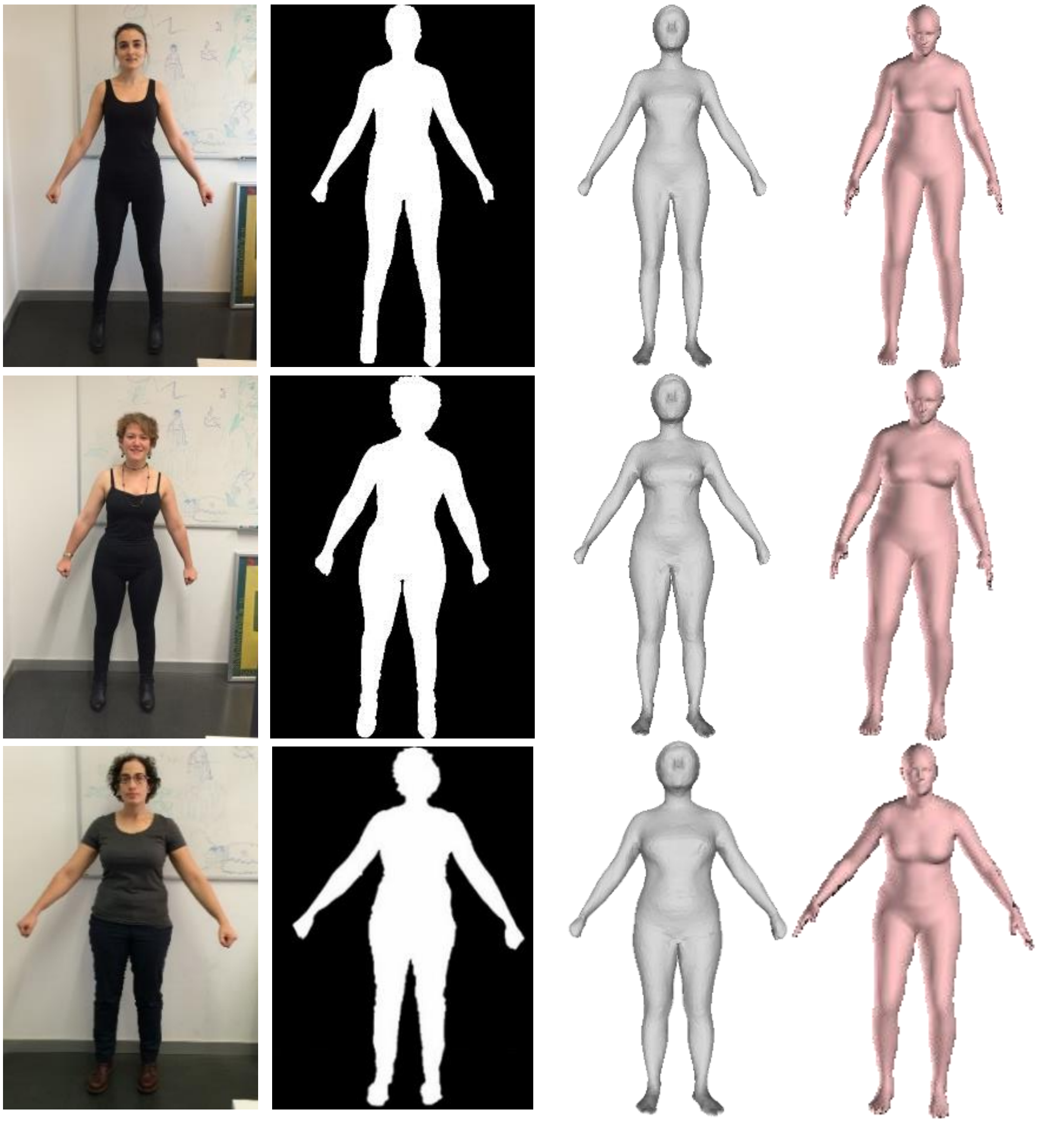


Figure 3.22: *Results for predictions on the test images from Method 1 of Section 3.3. From left to right: the input image in a rest pose, the corresponding silhouette, the estimated mesh by our method $SF-1-P$, and by the method of* [Bogo et al.,2016a].

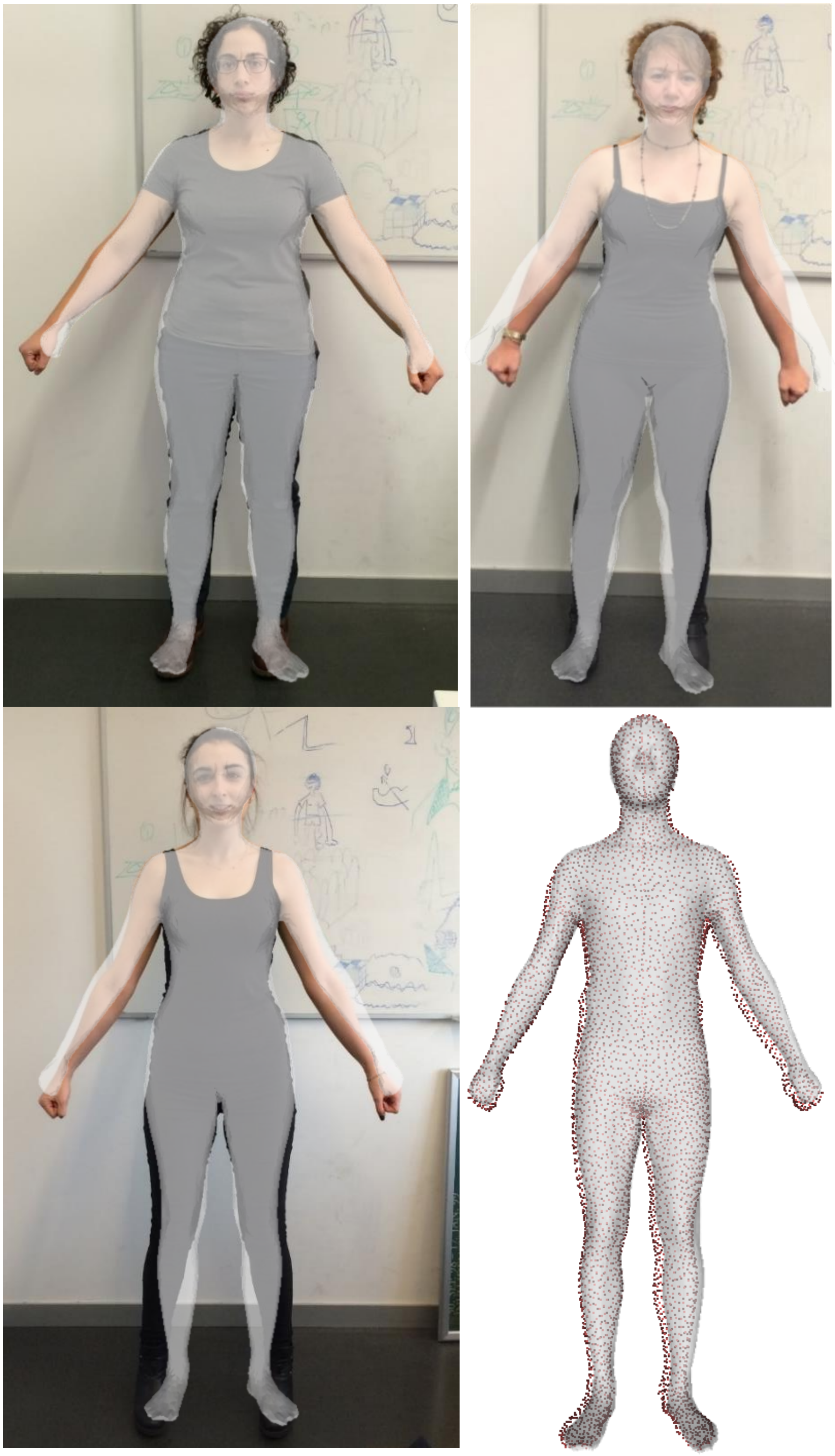


Figure 3.23: *Estimated overlayed meshes utilizing our method overlayed on the input images or scans (bottom-right).*

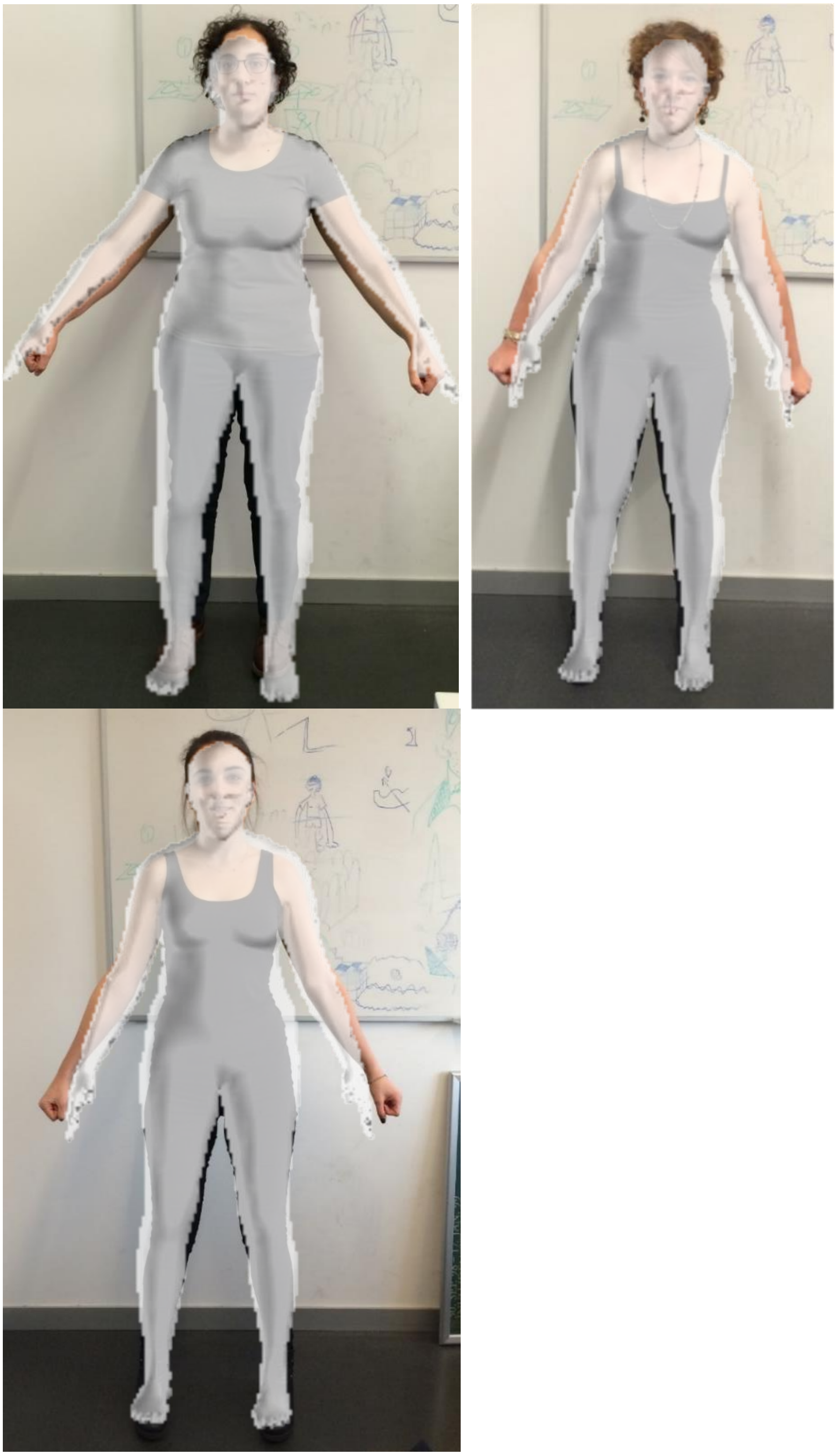


Figure 3.24: *Estimated overlayed meshes utilizing the method from* [Bogo et al.,2016a] *overlayed on the input images.*

3.5.5.4 Failure Cases

One typical example of a failure case is that of a single view ambiguity, e.g. Figure 3.25 (bottom), where we show a synthetic mesh of a man with pot belly that is not captured from the frontal silhouette, hence the reconstruction (on the right) tries to best explain it. Other examples are bodies that do not reside in the shape space from which we generate the data, e.g. the muscular male in Figure 3.25 (top).

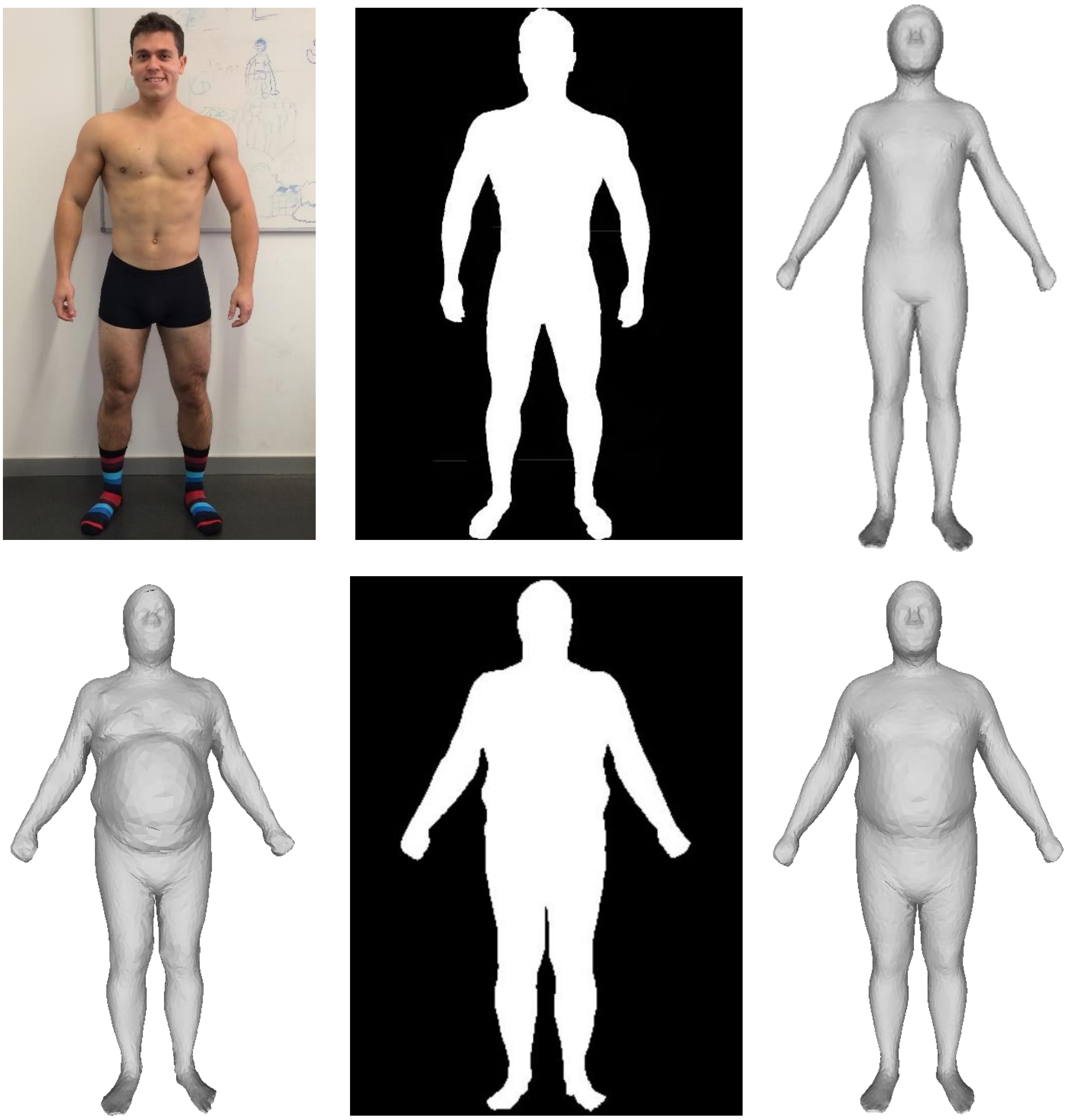


Figure 3.25: *Examples of two failure cases.*

3.5.5.5 Speed

The training of our network was performed on an Intel(R) Core(TM) i7 CPU 4770 3.4 GHz with NVIDIA GTX 1080 (8G) GPU. It took around 50 min per epoch, with one epoch consisting of roughly $50,000$ samples. The total training time for the various architectures considered in the experiments varies from 15-30 epochs. We conducted our test time experiments on an Intel(R) Core(TM) i7 CPU 950 3.0 GHz with NVIDIA GTX 940 (2GB) GPU. Since our method directly outputs the vertices of a mesh, and does not need to solve a least squares system, it is much faster (0.15 seconds) than other methods when using the GPU for prediction. Even when using a CPU, our method takes about 0.3 seconds, similar to the fastest method presented in Section 3.3, and less than 6 seconds [Boisvert et al., 2013] and 0.45 seconds from the second method (Section 3.4). As a result, our method scales to higher mesh resolutions, and can be directly used as an end-to-end pipeline, outputting a full 3D mesh. With the advances in compressed deep networks (e.g. [Han et al., 2015; Iandola et al., 2016a]), this can potentially be ported to mobile devices, which is in line with our targeted application of shape from selfies.

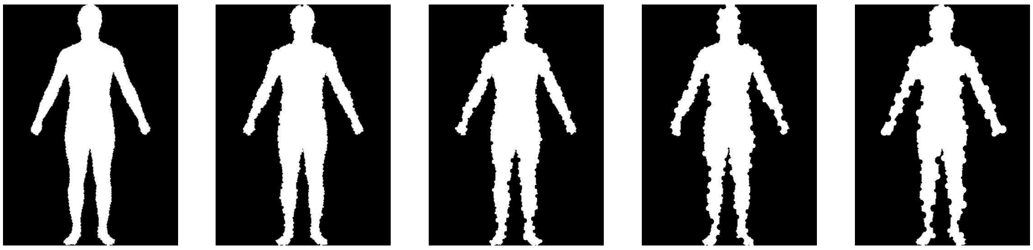


Figure 3.26: *5 silhouettes representing the same person with noise applied to them. Noise parameters (radii) considered 1,3,5,7 and 9 pixels.*

3.5.5.6 Parametric Space and Final Error Components

We chose to use 20 PCA components to generate the shape space, for fairness to the first two presented methods that utilize the same number of components, but also because it was enough to capture 95% of the energy and avoid low-variance datasets. Starting from the original meshes and others spanning such a space, we learn a 4000 dimensional internal representation space, extracted from the HKS features and used to decode mesh vertices directly. Despite the fact that the embedding space is of higher dimensionality than the 20 parameters used in the previous works, we believe that it's higher accuracy stems from compact pose-invariant features, needed here to learn non-linear mappings to higher dimensional mesh vertex spaces. This is a better learned representation than just pure PCA applied on the triangle deformations. Furthermore, our method is also faster than the other methods for the same input and output resolution. This is one factor that contributes to the final error estimation, and also demonstrated e.g. in Table 3.8, where $SF - 1$ is compared to $HS - Net - 1S$. Another factor, that plays an important role ,is the mapping from silhouette images to the embedding space. We demonstrate the decrease in error as the number of view/modes is increased during training, but remains uni-modal at testing, utilizing our novel CMNN network (e.g. Table 3.8, $SF - 1$ vs $SFS - 1$). If we consider the influence of the input image resolution, we believe that it does not play a role in comparison to the previous works, as we used the same input image size as in the second method, which is half of the resolution utilized in the first one. Last but not least, the combination of the above two factors through joint training also helps decreasing the errors, as we show in Table 3.7 $SFUS - HKS - 1$.

3.5.5.7 Noise

An important evaluation factor for real world systems is robustness to noise. Although for our target applications this is less of a concern, in general this is important. Hence, we generate noisy silhouettes by non-uniformly eroding or dilating the silhouette at the border, with filters of various radii (we consider 1,3,5,7 and 9 pixels). An illustration of such noise applied to the same silhouette for the various radii is depicted in Figure 3.26. The mean error obtained over all the body measurements when noise is applied to every input test silhouette for the $SF - 1$ network, is shown in Figure 3.27 (top line), computed as the difference from the clean silhouette errors. As it can be observed, the increase in error for reasonable noise radius is small, and even for highest noise radius, the maximum error is below 2 cm.

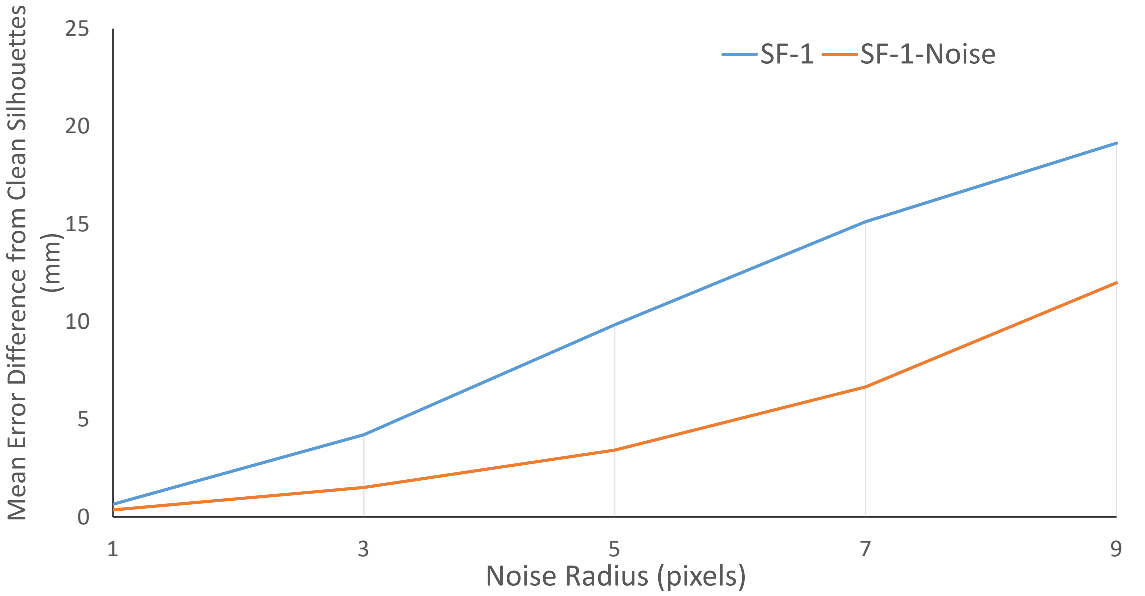


Figure 3.27: *Error plots for the increase in the mean errors as compared to the silhouettes without noise. The top line ($SF-1$) demonstrates the errors when training is performed on clean silhouettes and testing on noisy ones. On the other hand, the bottom line ($SF-1-Noise$) demonstrates the errors when noise is inflicted into the training data. The mean errors are computed over all body measurements. The noise parameter (radii) varies from 1 to 9 pixels.*

Missing Limb. In addition, we perform a further experiment, where silhouette noise is understood as a missing limb part, which could represent difficulties in silhouette extraction over various body parts, due to motion blur, occlusion or similar foreground-background color (e.g. when a person stays in front of a wall or uniform background color, quite often the feet project onto the floor/pavement which could be of the same color as the shoes, e.g. Figure 3.22). For this, we evaluate the $SF-1$ on test data of the form depicted in Figure 3.28, where a limb part is missing. We observe a little increase of 3.77 mm in the overall mean error, as compared to $SF-1$ evaluated on the complete silhouettes. This could be due to the human body prior and its symmetric properties.

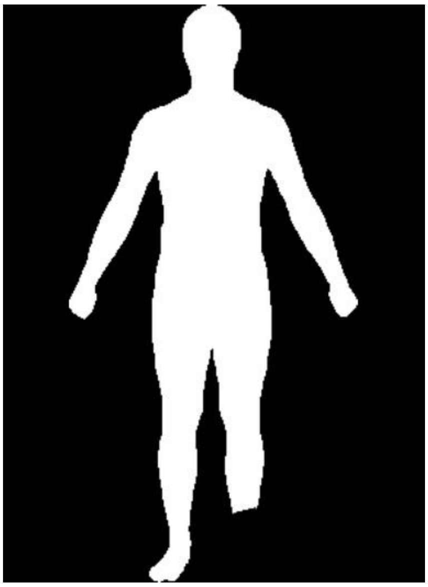


Figure 3.28: *Visualization of the input silhouettes when a limb part is missing.*

Train with Noise. Lastly, we perform an experiment, where instead of only testing with noisy silhouettes, we also train with noisy ones. For this experiment the amount of training data grows linearly with the amount of noise radii we consider. Once again the network trained is similar to $SF-1$, and we call it $SF-1-Noise$. Evaluating on the same test silhouettes as the first noise experiment, we observe a decrease in the mean error, as shown in Figure 3.27 (bottom line), which shows that adding perturbations and noise to the training data makes the method more robust to it.

3.5.5.8 Comparison to CCA

We demonstrated that cross-correlating features during training time at later stages, by sharing weights in fully connected layers, improved predictions for our test samples. One main advantage of cross-view learning through neural networks is that the training data does not need to be stored in memory and especially, one can add as many views as desired, as compared to CCA [Hotelling, 1936] that has been practically shown for two views only. Nevertheless, for fairness also to our first method, we compare to a version of our network that utilizes CCA for correlation, and only considers two views (the front and side silhouette scaled and in a neutral pose). The training goes as follows : 1. We first train two networks separately, one for the front $SF-1$ and one for the side $SS-1$ to map view specific silhouettes to the embedding space. This is utilized to learn view specific features directly from the network (as opposed to the first method that extracts handcrafted

Measurements	SFS-1	SF-1-CCA	SFUS-1	SFUS-1-SH
A. Head circumference	3.7±2.9	4.3±3.5	3.9±2.9	4.2±3.4
B. Neck circumference	2.3±1.8	2.8±2.1	2.2±1.7	2.2±1.9
C. Shoulder-blade/crotch length	5.3±4.2	7.2±5.5	5.4±4.1	5.8±4.5
D. Chest circumference	5.9±4.9	7.8±6.9	5.8±4.8	6.6±5.5
E. Waist circumference	7.5±5.9	9.2±7.2	7.5±5.7	8.5±6.6
F. Pelvis circumference	8.4±6.7	9.7±8.1	8.1±6.5	8.6±7.1
G. Wrist circumference	1.9±1.6	3.1±2.1	1.9±1.6	2.1±1.7
H. Bicep circumference	2.9±2.4	4.2±3.4	2.9±2.5	3.3±2.6
I. Forearm circumference	3.1±2.3	3.3±2.6	2.9±2.3	3.2±2.5
J. Arm length	3.3±2.5	4.5±3.6	3.2±2.5	3.5±2.9
K. Inside leg length	6.2±4.8	7.4±6.0	5.7±4.5	6.2±5.1
L. Thigh circumference	5.8±4.9	7.1±5.9	5.8±4.8	6.2±5.3
M. Calf circumference	3.3±2.7	4.3±3.6	3.5±2.8	3.9±3.3
N. Ankle circumference	1.9±1.5	2.1±1.6	2.1±1.5	2.3±1.7
O. Overall height	11.2±8.6	14.1±11.1	10.4±8.1	11.9±9.5
P. Shoulder breadth	2.2±1.2	2.6±2.1	2.1±1.7	2.2±1.9

Table 3.9: *Body measurement errors comparison over the various experiments considered here. Errors are expressed as Mean±Std. Dev in millimeters.*

features); 2. Then, we extract 8064 features from the last convolutional layer over each view, for all of our training data, and since the dimensionality is quite high, we apply dimensionality reduction through PCA up to 500 dimensions that capture most of the energy. Starting from these 500 dimensional vectors we apply CCA, to find linear projection bases where the correlation of the projected features is maximized; 3. In the end, we train a smaller network of three fully connected layers $SF-1-CCA$, to map from the 500 CCA projected features of the frontal view only (the desired one) to the embedding space. At test time, a new frontal view silhouette is first input to $SF-1$ that performs a forward pass to extract 8064 features, which are then projected onto PCA and CCA. The projection is mapped through $SF-1-CCA$ to the embedding space, which in turn reconstructs the mesh with the help of the $HKS-Net$. We demonstrate the results of this procedure for the same synthetic meshes and we compare to our cross-modal training over two views $SFS-1$ in Table 3.9. It can be noticed that our method outperforms the CCA based one. The latter still performs well, however on the expense of added memory footprint and unscalability to more than two views. Furthermore, it is not trivial to train the network end-to-end without splitting it into various components. And lastly, we think that most of the learning is due to the non-linear mapping performed from $SF-1-CCA$, rather than from the linear CCA mapping.

3.5.5.9 Late Sharing

We perform a further experiment, to demonstrate the need of sharing weights at later stages in the network for the cross-modal training, as opposed to sharing at earlier stages. The motivation behind late sharing was that we first wanted to let the network separately figure out the appropriate filters to apply to the various views, and then combine higher level and more meaningful features through shared fully connected layers. To demonstrate this, we train a network considering three views, similar to $SFUS-1$, however here the weight sharing starts from the first convolutional layers, all the way to the end, which we call $SFUS-1-SH$. The evaluations of this network for the same synthetic data, with frontal scaled silhouettes as input, are depicted in Table 3.9. It can be seen that the results are worse than $SFUS-1$, demonstrating the need for late sharing.

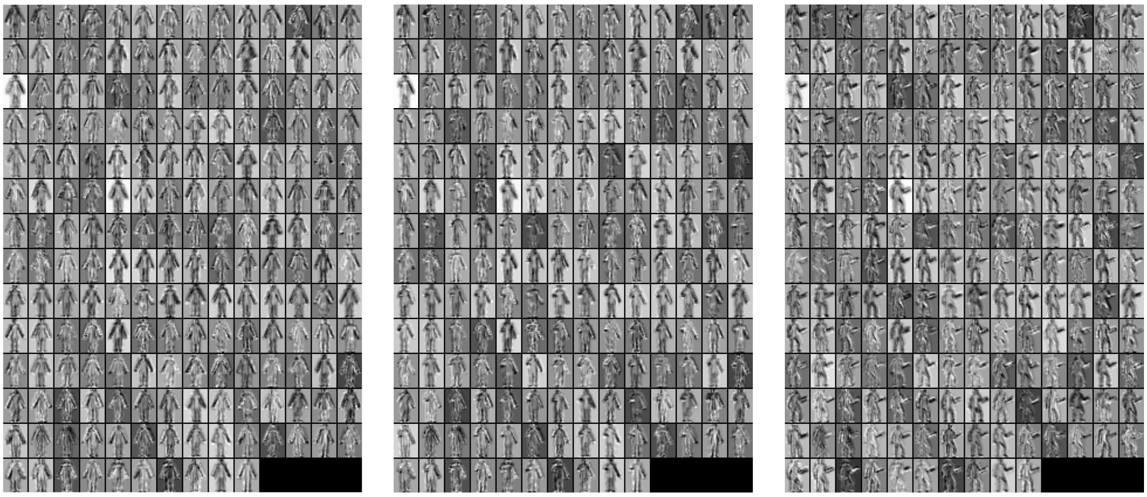


Figure 3.29: *Visualization of filter responses on the last convolutional layers of $SF-1-P$. The same person in three various poses is shown.*

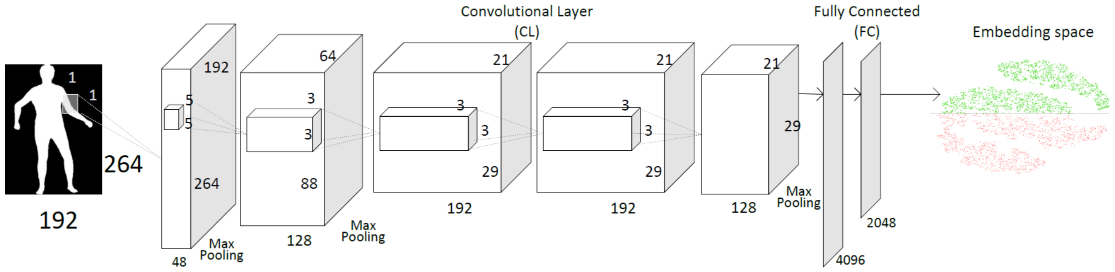


Figure 3.30: *Network architecture for a single view case trained with $SF - 1 - P$ architecture. For other types of inputs, such as side view etc., the architecture is the same.*

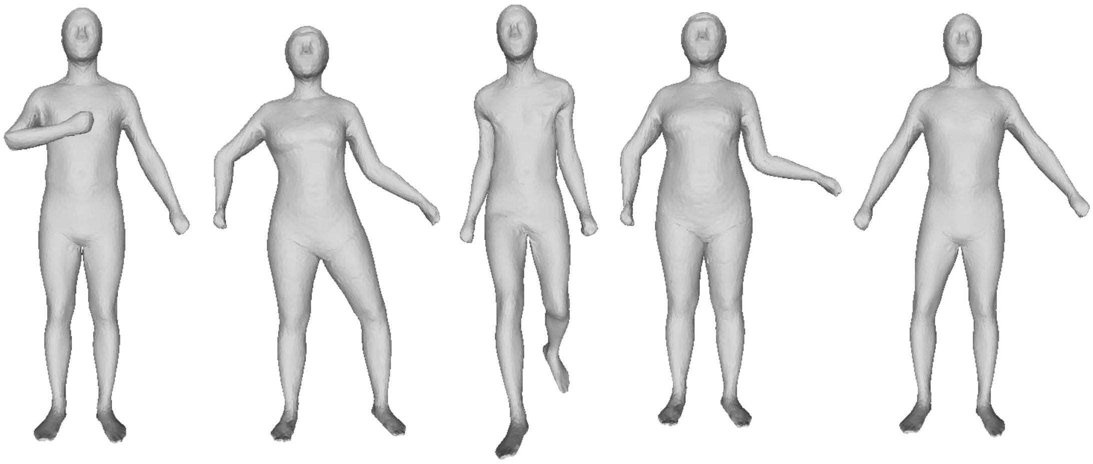


Figure 3.31: *Illustration of people in various poses considered throughout our experiments.*

3.5.5.10 Convolutional Filters

For illustrative purposes, we also demonstrate the filter responses of one of the last convolutional layers for $SF - 1 - P$ when the input silhouette is of a person in three various poses (Figure 3.29). A more detailed version of the single view architecture is depicted in Figure 3.30. The network internally learns to distinguish between various body parts (e.g. limbs), as similar looking filters are applied to the same parts (e.g. hands), even though the poses vary. An illustrative figure of the various poses we consider is depicted in Figure 3.31.

3.5.5.11 Experiments on HKS-Net

Despite our intention in this lastly introduced method to demonstrate accurate estimation of human body shapes from silhouettes, here we present some further experiments that show some of the nice properties of the $HKS - Net$. We demonstrate visual results of the reconstructions as well as mean errors computed over all the body measurements. For each input mesh we first compute the HKS descriptor. That is then fed into the $HKS - Net$ to reconstruct the final mesh. For the quantitative results, we compute the difference of errors for each measurement, obtained for each of the experiments that we consider (which modify the original mesh), from the errors obtained when the original meshes in neutral pose are input to the $HKS - Net$.

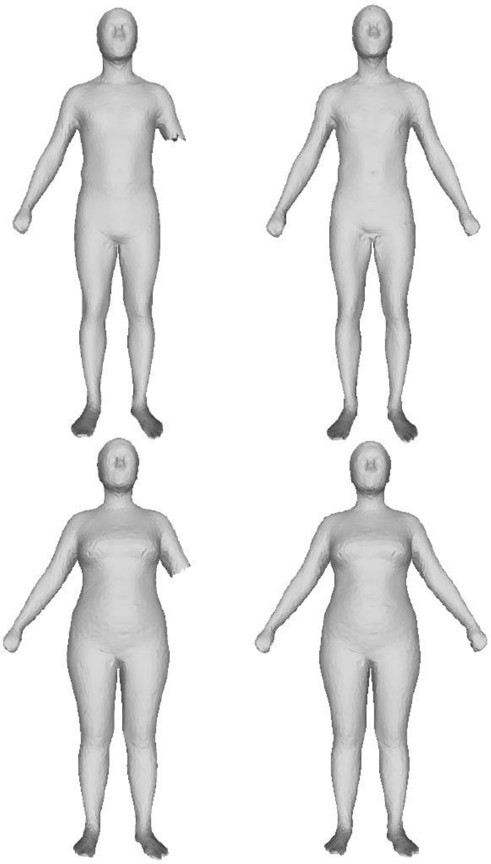


Figure 3.32: *Mesh reconstruction (right) when a partial mesh (left) is input into the $HKS - Net$.*

Partial Mesh. First, we assume the mesh in a neutral pose comes with missing parts (limbs etc.). We remove the left hand over all the test meshes. The qualitative reconstructions are depicted in Figure 3.32. The mean overall added error is 6.72 mm. We can observe that the network has reconstructive abilities despite missing extremities.

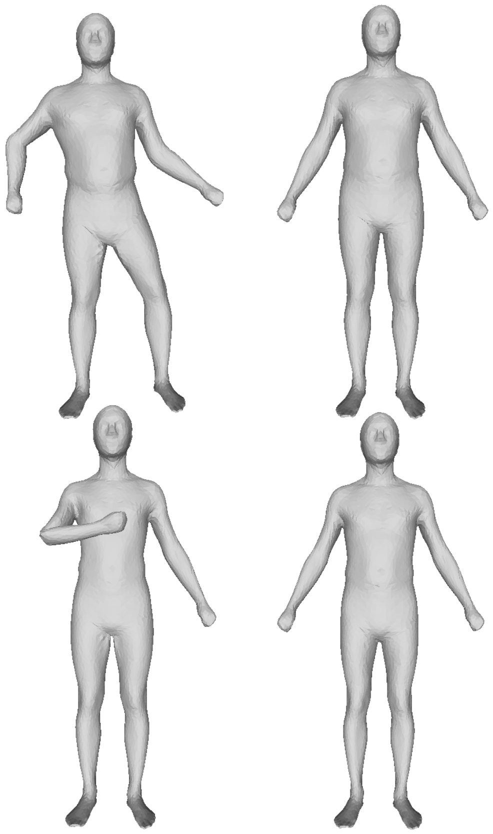


Figure 3.33: *Mesh reconstruction (right) when a posed mesh (left) is input into the $HKS - Net$.*

Posed Mesh. Secondly, we test over meshes coming in poses obtained from Linear Blend Skinning (LBS) and different from the neutral one. This experiment is important for applications where the computation of a neutral pose of a given posed-mesh is needed. This would allow for mesh alignment, matching as well as consistent measurement computations. Some qualitative reconstructions are depicted in Figure 3.33. The mean overall added error is 3.72 mm. This almost implies invariance to isometric deformations, however due to LBS artifacts the errors increase a bit as opposed to the neutral pose reconstruction.

Noisy Mesh. Lastly, we evaluate robustness to mesh noise for the $HKS - Net$. For this we apply random vertex displacements to the original ground truth meshes. The qualitative reconstructions are depicted in Figure 3.34. The mean overall added error is almost negligible, 0.2 mm, which implies robustness to mesh noise.

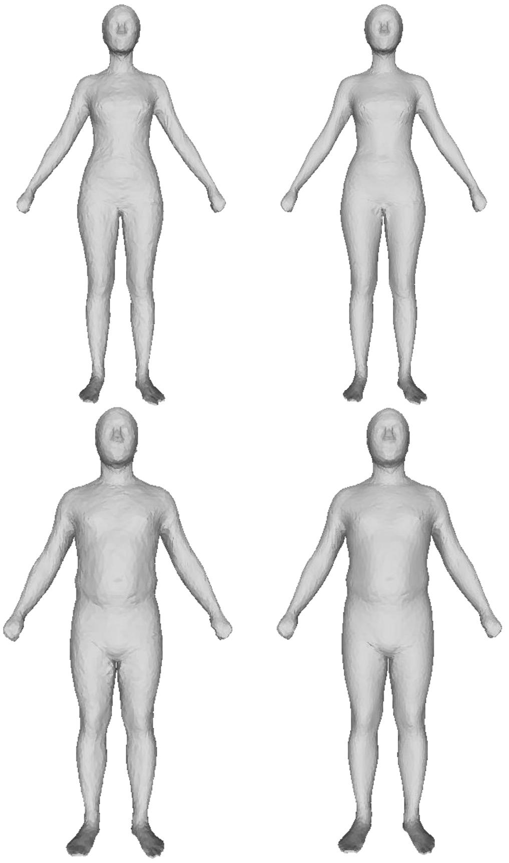


Figure 3.34: *Mesh reconstruction (right) when a noisy mesh (left) is input into the* $HKS - Net$.

3.6 Discussion and Conclusion

In this chapter we presented three discriminative methods that estimate the human body shape given monocular images.

CCA-RF. The first method estimated 3D human body shape from silhouette features mapped through random forest regressors. It allowed different views, poses with mild occlusions, various body shapes to be estimated, and it was extensively evaluated on thousands of human bodies, by utilizing one of the biggest databases available to the community.

Withing the scope of this work, shape extraction from a single silhouette was in focus, because of its various applications such as selfies or utilizing limited video footage. However, this is an inherently ill-posed problem. Further views can be incorporated to obtain more accurate reconstructions, similar to methods we compare to. This would lead to a better estimation especially in the areas around the belly and chest, hence decrease the elliptical body measurement errors.

The accuracy of this method is tied to silhouette extraction. For the difficult cases of dynamic backgrounds or very loose clothes, the large scale silhouette deformations would skew our results. This could be tackled by fusing results over multiple frames. Unlike [Chen et al., 2010] though, our results always remain in the space of plausible human bodies. For small scale deformations, as in Figure 3.7, we show in Table 3.2 (N) that our results stay robust.

We assume that the silhouettes come in poses with limited partial occlusion. Under this assumption, we showed robustness, as the same mesh estimation is achieved from different poses (e.g. Figure 3.6). However, under more pronounced occlusions, our results start degrading (Table 3.2 (PWR)), which could be alleviated by increasing the number of training poses and utilizing deeper learning.

Although we aimed at precise measurements for the evaluation, errors due to discretization are inevitable, hence a standardized procedure on a standard mesh dataset is needed as a benchmark. We believe that this work along with that of [Boisvert et al., 2013] has set an important step towards this direction.

Since our system is designed for a general setting, we apply a fixed scale to the silhouette, losing height information. We showed a fairly good performance on estimating the relative height and demonstrated better absolute height estimation, given that camera calibration is incorporated.

Our fast system, running in minutes for training and milliseconds for execution in single core CPU's, while being memory lightweight due to the low feature dimensionality, could be integrated into smart phones, allowing body shapes to be reconstructed with one click of a button. Simultaneously, it can be used for 3D sport analysis, where estimation of a 3D shape of a player seen from a sparse set of cameras can improve projections of novel-views.

Finally, we showed how CCA, which captures relations in an unsupervised linear way, can be used to correlate different views in the data to improve the prediction power and speed of the algorithm. We believe that capturing non-linear relations with Kernel CCA's or deep architectures, such as auto-encoders, should lead to even better results. Our method illustrates the utility of CCA for other vision applications where two or more views describing the same object or event exist, such as video-to-text matching or shape from various sources of information.

HKS-Net. With the second method, we abstracted away from handcrafted features, and handled the accurate estimation of 3D human body shape from silhouettes (or shaded images) utilizing CNNs. As with the previous method, the problem was posed as a regression that finds a global mapping from the various inputs that we presented to shape parameters. We extensively evaluated our technique on thousands of human bodies and real people.

In compliance with our main target applications, e.g. garment fitting and as opposed to the previous method, we mainly focused on shape estimation of people in neutral poses allowing mild pose changes, from one or two binary silhouettes as well as shaded images as input. We showed that we outperform methods based on global mapping and achieve similar results to more expensive methods that employ local fitting.

In the scope of the networks that we experimented with, we showed how to simultaneously combine two binary silhouettes in order to improve prediction over a single one, and evaluated three different methods. This also set a ground for the method presented last.

We also demonstrated in a synthetic experiment, that if shading information is present, better results are achievable. Due to lack of real data though, it is difficult to assess its performance on real humans and believe that as intrinsic image extraction algorithms improve, it will lead to future works in this domain.

Even though silhouette extraction is not a bottleneck for the target scenarios assumed by the method, e.g. due to potential uniform backgrounds, we evaluated the performance under the influence of noise of different levels, and showed that our method is robust to silhouette noise under reasonable assumptions. We further assumed humans in tight clothes. Applying our method to a scenario where clothed people are present deteriorates the results, however in contrast to previous works, the reconstructions remain in the space of plausible human bodies, which is mainly due to the statistical body model which we utilize.

A limitation to our method is that with the current training, it can not handle poses that differ significantly from the neutral pose and contain self-occlusions. We could handle that by generating a larger training set including more pronounced poses, which is tackled in the last method.

Lastly, we showed that our system is orders of magnitude faster than the methods we compare to. Based on recent works [Chen et al., 2015] that try to compress Neural Networks as well as the possible speed-up of our mesh computation, our algorithm, which already runs at interactive rates, could also be integrated into smartphones in the foreseeable future.

HKS and Cross-Modal Net. Lastly, we built up on the previous two methods and presented a novel method for capturing a 3D human body shape from a single silhouette with unknown camera parameters. This was achieved by combining deep correlated features capturing different 2D views, and embedding spaces based on 3D shape descriptors in a novel CNN-based architecture. We extensively validated our results on synthetic and real data, demonstrating significant improvement in accuracy as compared to the state-of-the-art methods. We illustrated that each component of the architecture is important to achieve these improved results. Combined with the lowest running times over all the state-of-the-art, we thus provided a practical system for detailed human body measurements with millimetric accuracy.

The proposed cross-modal neural network enhances features by incorporating information coming from different modalities at training time. The idea of such correlating networks can be extended for many other problems where privileged data is available, or correlations among different data types (e.g image, text, audio) are to be exploited. HKS-Net like architectures can be used for inverting shape descriptors, which can have various applications for understanding and generating shapes.

Inferring 3D shapes from 2D projections is an ill-posed problem. As in the previous methods, we also operate under mild occlusions and a certain level of silhouette noise, which are realistic assumptions for many scenarios including ours. However, especially for severe occlusions, we would need stronger priors to infer correct 3D shapes. We believe that extending our techniques for images with shading cues can provide accurate estimations even for such cases. A training, covering different environments and textures, would be necessary for this case.

CHAPTER 4

Garment Shape Estimation

In the previous chapter, we tackled the capturing of human bodies, under no or tight clothing assumptions. Clothing, especially in a loose form, is an important part of virtual human modeling. Capturing and modeling garments are fundamental steps for many applications ranging from online retail to virtual character and avatar creation. There exist many options to model or capture garments with professional tools used by talented artists, digitalized traditional garment sewing patterns, 3D meshes and expensive physically based simulations, or 3D capture with advanced hardware in controlled setups. However, such tools and setups are not available to most content generators and users. Instead, a practical approach is allowing the users to utilize commodity cameras and capture clothing from a single image or video. Such simple and practical capture systems have recently been developed for human faces [Cao et al., 2015; Kim et al., 2017; Tewari et al., 2017], hair [Hu et al., 2015], eyes [Bérard et al., 2016; Bérard et al., 2014], body shapes [Balan et al., 2007; Bălan and Black, 2008; Guan et al., 2009; Zhou et al., 2010; Jain et al., 2010; Hasler et al., 2010; Chen et al., 2013; Rhodin et al., 2016; Bogo et al., 2016a] and the techniques presented in the previous chapter, or hands [Tkach et al., 2016; Spurr et al., 2018; Zimmermann and Brox, 2017]. Cloth capturing from dynamic scenes with monocular imagery remains a challenge due to the complex deformations.

A successful approach to solve such ill-posed capturing problems is utilizing data-driven priors. With the recent advances in machine learning techniques, it has been demonstrated that accurate reconstructions of various types and classes of objects can be obtained even from a single image, as in

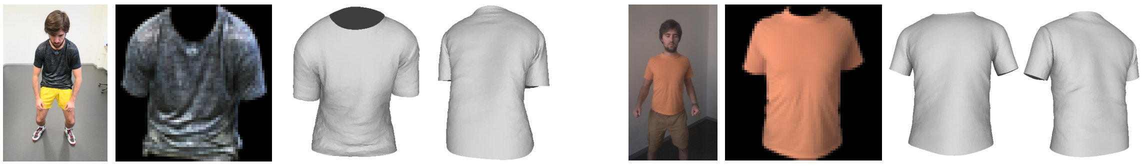


Figure 4.1: *Garment 3D shape estimation using our CNN model and a single-view. From left to right: real-life images capturing a person wearing a T-shirt, segmented and cut-out garments and 3D estimations of the shape.*

[Wu et al., 2015; Tatarchenko et al., 2016] and the method from Section 3.4. This requires constructing a database that covers the subspace of possible data points while staying practical in terms of its size, and a careful modeling of the input/output spaces and the associated learning algorithm.

In this chapter, we present a data-driven technique that can recover the 3D shape of a garment from a single image by utilizing a database of purely synthesized clothing. We construct our database by simulating garments on virtual characters for different poses and shapes, and under different lighting conditions and camera views. We then propose a convolutional neural network based architecture that is trained with our data set. The key idea is to learn the deformation (simply represented as the vertex displacement) from a reference mesh (either a garment mesh or a body mesh, depending on the application) with respect to image observations, using a CNN. As the data contains physically simulated garments, our technique is able to capture dynamic clothes in motion for various scene conditions and poses. Our goal is to obtain the correct global 3D shape, possibly with plausible high-frequency deformations (such as wrinkles and folds), ready to be used in applications such as virtual avatars. This is a challenging problem for a single view due to the occlusions and loss of 3D information in real images. We illustrate that even very challenging cases can be handled with the proposed technique with garment specific data-driven priors.

4.1 Synthetic Garment Data Generation

We require to have a database of pairs of renderings and the corresponding 3D garment shapes, in order to apply supervised learning with the proposed CNN in Section 4.2. Unfortunately, there exist no such datasets, mostly due to the difficulty of capturing garments under various settings. Hence, a very important step in our technique is synthesizing data that capture garment deformations. Figure 4.2 shows our data generation pipeline consisting of the following steps:

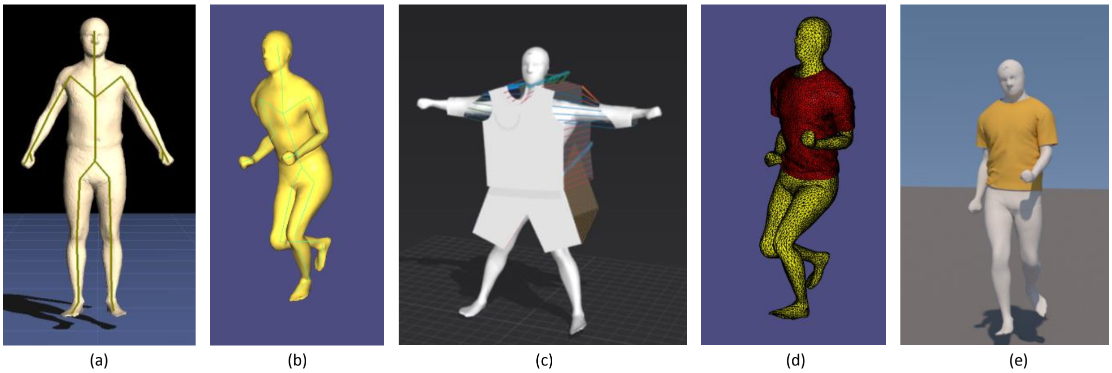


Figure 4.2: *Data generation pipeline. A human model is automatically rigged (a) and animated (b). Then a desired garment is designed (c), simulated under some motion (d), and realistically rendered (e). A mask of the rendered garment is used as a training input.*

Human Model Creation and Animation. The first step in obtaining an accurate garment geometry is to create a dataset of naked human meshes and animate them in a natural manner. We picked 10 meshes, from a dataset of 1500 male meshes [Pishchulin et al., 2015] (Figure 4.2 (a)), generated by a statistical human shape model [Anguelov et al., 2005], covering major variations in body types. For animation, we utilize varying motions such as walking, running, jumping, turning, dancing, and boxing sequences, represented as 3D skeletons and extracted from an available motion capture dataset [CMU,], adding up to 20 minutes of motions. We attach the skeletons to the human shapes with an automatic method [Baran and Popović, 2007b] that computes skinning weights (Figure 4.2 (a)), by augmenting its implementation with our motion capture skeleton. Each motion pose is then represented as a transformation relative to a T-pose, scaled to the size of the corresponding auto-rigged bones and mesh. The meshes are animated applying Dual Quaternion Skinning [Kavan et al., 2008], as in Figure 4.2 (b).

Garment Design. In order to design the clothing and then dress the character with it, we use Marvelous Designer [MD,], a commercial software that allows to select clothing type, material properties, and set tightness of the cloth onto a normalized body posture (T-pose), as in Figure 4.2 (c). This is a tedious manual process, and without loss of generality we design men t-shirts, as well as a woman's dress, representing semi-tight and loose clothing.

Garment Simulation. We animate the characters dressed in the designed garments with the motion capture dataset and simulate cloth material behavior utilizing ARCSim [Narain et al., 2012; Narain et al., 2013] for physi-

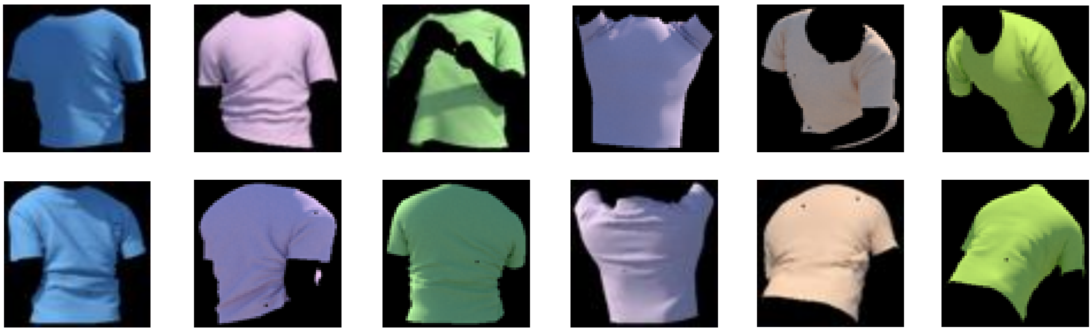


Figure 4.3: *Samples of masked and downscaled renderings of a garment for a front (top) and back (bottom) view.*

cally based simulation. This software has the advantage of cluster deployment, due to the extensive use of OpenMP, which benefits our data generation process. After extending it to support non-rigid object animations and deformations, we run our simulations at 30 FPS, which results in approximately 15000 shapes per character and per garment (with the resolution of 6500 vertices, and 12800 triangles, Figure 4.2 (d)). In order to align the generated meshes, we remove the global translation and rotation, as computed by the translation and rotation element of the root joint from the articulated skeleton of the corresponding animation frame.

Rendering. In order to realistically render the simulated geometry accounting for phenomena such as soft shadows, global illumination, or realistic light sources, we utilize Mitsuba [Jakob, 2010], which is also easily deployable to a cluster. We create a scene with a simple, planar, diffuse surface, serving as the ground, with a gray albedo as the scene. The garment material is set to be diffuse due to Lambertian assumptions and the color is randomly sampled. The whole scene is lit by a realistic sun-sky emitter varying its position, to approximate natural lighting as accurately as possible. The camera pose is also varied to capture view-point changes. We show an example rendering in Figure 4.2 (e). We also render a mask, which is then cropped with a padding of 5 pixels around the boundaries of the masked area while setting the non-garment pixels to zero. Then, the image is downscaled to the size of 64x64 pixels, as in Figure 4.3. In this work we utilize single and two-view models. Therefore, we render views from the front and back. The camera is placed on the normal direction of the pelvis, with a variation of $\pm$ 30 degrees, as also shown in Figures 4.3, 4.7, 4.8. Hence, at test time, the system can cope with large variations in view and pose.

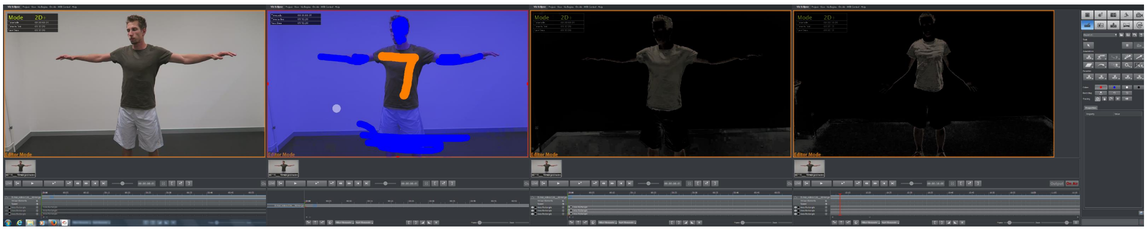


Figure 4.4: *Video segmentation pipeline with our software. From left to right : First video frame, foreground and background scribbles, segmentation result on the frame, segmentation automatically propagated to another frame.*

4.2 3D Garment Shape Estimation Method

Our method aims at estimating 3D garment shape or shape deformations from a single image capturing dynamic motion. Below we explain each step and the neural network architectures we developed to tackle this problem. Given an input image, our system masks the garment, and feeds it as an input to a specialized CNN, trained end-to-end to regress to 3D garment vertex deformations or offsets from a template mesh or human body. The method accurately captures global (low-frequency) deformations, and for data similar to the training set, it is even capable of recovering high frequency details. A better recovery of higher frequency details (such as wrinkles or folds) can be further enforced using a specialized loss layer, computed over vertex positions and normals simultaneously. As a final step, in order to avoid interpenetration between the estimated garment and the body, we minimize an energy term on the vertex displacements.

4.2.1 Preprocessing

As our system is trained to regress with CNN-s, including background information would add noise, and in order not to bias the regressor towards backgrounds, one would have to generate a variety of them for the same training sequences, increasing the training time and data space drastically. Hence, we assume to have a mask for segmenting out the garment as input to our technique.

An accurate segmentation can be obtained by assuming a background of uniform color, utilizing Gaussian Mixture Models to learn a background model, and finally segmenting with graphcuts [Boykov and Funka-Lea, 2006]. We could also apply learning based techniques like the recent background subtraction CNN-s [Badrinarayanan et al., 2015], or cloth specific segmentation methods [Yamaguchi et al., 2013] similar to previous works [Yang et al., 2016]. The masks used for segmentation can also be propagated if a video is used as the input. When assuming a background of uniform color, we use an automatic method as elaborated above [Boykov and Funka-Lea, 2006]. Otherwise, we have an interactive segmentation pipeline based on scribbles as illustrated in Figure 4.4.

4.2.2 Mesh Deformation Representation

We have two different representations for the garment deformation. Both are based on the idea of vertex displacements, with different reference meshes. We will see in the next sections that each representation has its own advantages for different applications of our technique.

Garment-from-Garment Shape Representation. The output of our method is a 3D mesh represented as follows : Let $\mathcal{S}_{ref} = (\mathbf{V}_{ref}, \mathbf{F}_{ref})$ be a reference garment mesh which is dressed on a character in a T-pose, with $\mathbf{V} \in \mathbb{R}^{n \times 3}$ as the matrix storing the 3D coordinates of each vertex in each row, and $\mathbf{F}_{i,j} \in \{1, \ldots, n\}$ containing vertex indices, with each row defining one triangle of the mesh. We encode deformations of $\mathcal{S}_{ref}$ with difference vectors from the reference vertices. We thus encode a mesh $\mathcal{S}_k$ that was created by deforming $\mathcal{S}_{ref}$ with the matrix $\mathbf{V}'_k = \mathbf{V}_k - \mathbf{V}_{ref}$. In order to organize the dataset more conveniently, $\mathbf{V}'_k$ is flattened into $\mathbf{v}'_k$, where $\mathbf{v}'_k \in \mathbb{R}^{3n}$, and these vectors for all meshes in the database are then stacked into a large matrix $\mathbf{Y} \in \mathbb{R}^{M \times 3n}$, where M is the number of deformed shapes (or samples) in the database. Not all of the $3n$ degrees of freedom are necessary to represent our shape deformation set. We compress the matrix $\mathbf{Y}$ by performing a principal component analysis (PCA) to get $\mathbf{U} \in \mathbb{R}^{(N \times l)}, l << 3n$, where $\mathbf{U} = PCA_l(\mathbf{Y})$, and $l = 1000$, still achieving almost perfect reconstruction while reducing the dimensionality by a factor of 20. We thus set $\mathbf{Y} = \mathbf{U}$ for this case.

Garment-from-Body Shape Representation. Depending on the intended application, an alternative representation can be opted. The above representation does not guarantee that any estimated garment shape will fit the body mesh of our choice. Hence, if the intended application is to dress human meshes, we can represent the garment mesh as an offset from a body. For a given pose, we thus first associate each vertex of the garment mesh to its closest vertex on the body mesh, and compute the difference between those to get $\mathbf{V}'_k$ as above. The advantage of this alternative is that one can vary the body mesh and the garment dressed on that body will vary accordingly, avoiding major interpenetration artifacts. The downside is that without a specific body shape, the garment shape cannot be reconstructed. Hence the selection of the representation depends on the choice of the application, which is either garment shape estimation or body dressing estimation.

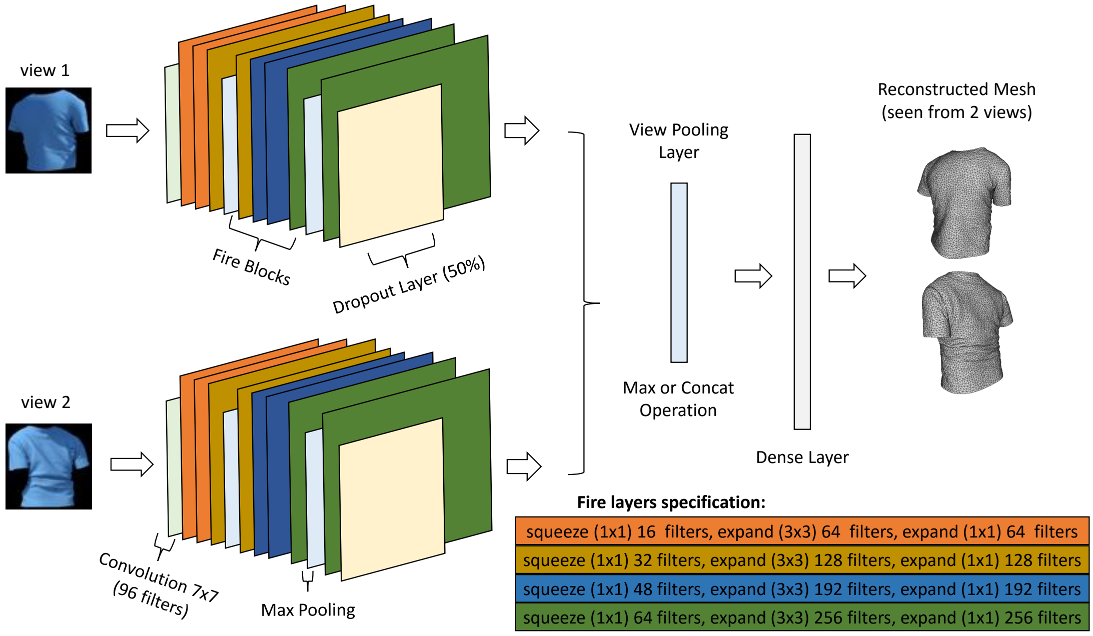
4.2.3 Single-View Architecture

We formulate the 3D garment shape estimation as an end-to-end regression performed by a Convolutional Neural Network (CNN), from an image depicting a person wearing the garment to the 3D shape of the garment. The network learns a representation space of image features that are able to encode the visual appearance of the possibly wrinkled clothing pattern.

Our dataset can be described with the input and output pairs $\mathcal{D} = (\mathcal{X}, \mathcal{Y})$. Let $\mathcal{X}$ be the set of our observations and $x_i \in \mathcal{X}$ a sample from our observation set, where $i \in \{1 \dots N\}$, and N is the number of samples in the dataset. The input x_i may consist of one or more images corresponding to different views. In our experiments, we only use one or two camera views, specifically frontal view and/or back view. Our images are masked and resized to 64×64 pixels hence the full input becomes $64 \times 64 \times 3$ dimensional. The $y_i \in \mathcal{Y}$ is the ground truth output data, which is obtained with either of the 3D garment mesh representations as described above, corresponding to the observed input x_i. The $\mathcal{Y}$ can be either the PCA-reduced output denoted as $\mathcal{Y}_{PCA}$, or the full-space dataset denoted as $\mathcal{Y}_{full}$ (Section 4.2.2).

The regression can then be written as the map $y = CNN(x)$, where CNN is the convolutional neural network model we consider. We have experimented with various CNN architectures including the ones presented in the previous chapter and more recent and advanced ones, whose macro-architectural pattern is inspired by Alexnet [Krizhevsky et al., 2012b]. The input to the network is a 64x64-shaped RGB image. The convolutional part of the network can either contain a sequence of simple convolutional layers or other more advanced convolutional architectural patterns. The convolutional part is followed by a flattening layer, after which one or more dense layers are employed. The final dense layer yields the output of the regression. The activation function we use is always the rectified linear unit (ReLU). To avoid overfitting, dropout is added after the convolutional part of the net. We have considered the following architecture:

SqueezeNet, introduced by [Iandola et al., 2016b], achieves AlexNet performance but needs significantly less parameters and therefore is much faster to train, thanks to its novel "Fire" Layers. The benefit of this architecture is its short training time while maintaining a high degree of quality, which makes it a great candidate for heavy experimentation.



Fire layers specification:

squeeze (1x1) 16 filters, expand (3x3) 64 filters, expand (1x1) 64 filters

squeeze (1x1) 32 filters, expand (3x3) 128 filters, expand (1x1) 128 filters

squeeze (1x1) 48 filters, expand (3x3) 192 filters, expand (1x1) 192 filters

squeeze (1x1) 64 filters, expand (3x3) 256 filters, expand (1x1) 256 filters

Figure 4.5: *Our SqueezeNet incarnation for the two-view case. The single-view is similar, except that only one convolutional block is utilized and there is no view-pooling layer. The input is one or two images of a masked garment, and the output is the garment mesh. For a more detailed description of the networks and a further discussion about the architecture please see Section 4.3.4.*

We base our network architecture on SqueezeNet, and adapt it to our problem. Figure 4.5 demonstrates the two view architecture described in Section 4.2.5. For the single view case, the network consists of only one convolutional block and no view-pooling layer.

4.2.4 Loss Layer

The choice of the loss function plays an important role in the training of neural networks. The ideal way to measure the performance of our neural net model would be to first reconstruct the garment based on the network output and then use it to render an image with the same configuration as the input image. A pixelwise distance between the rendered and the groundtruth image would suffice to measure the performance and backpropagate the loss. However, this is impractical due to the high rendering times that would significantly slow down the learning process. Therefore, we recur to a loss function that measures the error between the parameters of the estimated and the ground truth meshes. One possible way to do that, would be to compute the mean squared error over vertex positions

$$L_{full}(\mathbf{Y}^P, \mathbf{Y}^{GT}) = \frac{1}{n}\sum_{i=1}^{n}\left\|\mathbf{Y}_i^P - \mathbf{Y}_i^{GT}\right\|^2, \tag{4.1}$$

where $\mathbf{Y}^P$ is the predicted output, and $\mathbf{Y}^{GT}$ is the corresponding ground truth, and $\mathbf{Y}_i$ denotes the i-th row of $\mathbf{Y}$. If we regress to PCA component coefficients instead of vertices, we use the following weighted mean squared error function:

$$L_{PCA}(\mathbf{Y}^P, \mathbf{Y}^{GT}) = \frac{1}{l}\sum_{i=1}^{l} w_i \left|\mathbf{Y}_i^P - \mathbf{Y}_i^{GT}\right|, \tag{4.2}$$

Here, w_i is the PCA variance ratio corresponding to the i-th principal component, and l is the number of components. In order to capture the curvature better and in turn the folds and wrinkles, we extend Eq.4.1 by integrating normal estimations through an additional term in the loss. At each training iteration we compute the normals of the estimated vertices and compare them to the ground truth normals computed on the ground truth garment meshes. The final loss becomes:

$$L^*(\mathbf{Y}^P, \mathbf{Y}^{GT}) = \frac{1}{n}\sum_{i=1}^{n}\left\|\mathbf{Y}_i^P - \mathbf{Y}_i^{GT}\right\|^2 - \lambda\left[k\left(\mathbf{N}_i^P\right)^T \mathbf{N}_i^{GT}\right]^3, \tag{4.3}$$

where the matrices $\mathbf{N}$ are the normals of the corresponding vertices, λ is a weighting term set by the user (throughout our experiments set to 1000) that controls the influence of the normals as opposed to the vertex positions, and k a stretching term (set to 3) of the dot product, which when combined with the cubic exponential, gives more weight to the penalization for estimated normals that form a large angle with respect to the ground truth. This new loss function not only fixes some of the global deformation rotations, but also stresses the high frequency wrinkle patters, as demonstrated in Section 4.3.

4.2.5 Two-View Architecture

We additionally tackle the problem of simultaneously predicting the garment mesh when more evidence is included through a second view (e.g. a front and back view image of the garment). It turns out that simply concatenating the images along the channel dimension and then passing them through an architecture similar to the single-view one described above performs worse than the networks trained on single-view input only. One reason for this would be that including a complementary view at the early stages of the network, for the same amount of training data, might inflict noise in the system, as it was also observed in Section 3.4. Hence, we decided to combine information coming from multiple views at a later stage by separately training two similar CNN-s on each view, and then concatenating the outputs of the last convolutional layer of each CNN through a view-pooling layer that performs either a max or a concatenation operation, as shown in Figure 4.5. This architecture is capable of using the additional information from the multi-view input to produce more accurate results. The disadvantage of this architecture is that is has almost twice as many parameters and therefore doubles the training time and the memory needed.

4.2.6 Interpenetration Handling

Despite the fact that the garment shapes estimated from the CNN give small training and testing error, it can still happen that the estimated mesh does not fit the body perfectly but some vertices may be placed inside it, especially in cases where the input pose or body shape is very different from the shapes and poses that we consider during our training stage. Therefore, we employ a least squares energy minimization similar to [Guan et al., 2012] to push the interpenetrating vertices out of the body mesh. The energy (Eq.4.4) consists of multiple terms :

$$E_{\mathcal{B}}\left(\mathcal{Y}\right) = p_{\mathcal{C}}\left(\mathcal{Y}\right) + \lambda_s s\left(\mathcal{Y}\right) + \lambda_d d\left(\mathcal{Y}\right), \tag{4.4}$$

where $p_{\mathcal{C}}\left(\mathcal{Y}\right)$ stands for the interpenetration term, $s\left(\mathcal{Y}\right)$ for the smoothness term and $d\left(\mathcal{Y}\right)$ for the damping term. Parameters λ_s and λ_d are used to weight the importance of the individual terms.

Garment-body interpenetration is the most important term of the objective function. It takes care of pushing the interpenetrating vertices out of the body mesh. Let $\mathcal{C}$ be a set of correspondences between each garment vertex $\vec{v}_i$ and its closest body mesh vertex $\vec{b}_j$. Let $\mathcal{P}$ be a set of vertices that are currently located inside the body. A garment vertex $\vec{v}_i$ is located inside the body if $\vec{n_{b_j}}^T\left(\vec{v}_i - \vec{b}_j\right) < 0$, where $\vec{n_{b_j}}$ is the normal of the body vertex $\vec{b}_j$. Hence we have:

$$p_{\mathcal{C}}\left(\mathcal{Y}\right) = \sum_{(i,j)\in\mathcal{C}\wedge i\in\mathcal{P}} \left\|\epsilon + \vec{n_{b_j}}^T\left(\vec{v}_i - \vec{b}_j\right)\right\|^2 \tag{4.5}$$

where ϵ is set to a small negative number to ensure that the garment vertices are moved safely out of the body. This equation is underdetermined and has infinitely many solutions, therefore two additional terms are added to regularize the system.

The Smoothness term is added to make sure that the vertices are being moved smoothly with respect to their neighbors. This prevents the final solution from having undesirable spikes in place of the interpenetrating vertices which are being moved out of the body:

$$s\left(\mathcal{Y}\right) = \sum_{i \in \mathbf{V}} \left\| \left(\vec{v}_i - \widetilde{\vec{v}}_i\right) - \frac{1}{|\mathbf{B}_i|} \sum_{j \in \mathbf{B}_i} \left(\vec{v}_j - \widetilde{\vec{v}}_j\right) \right\|^2 \tag{4.6}$$

where $\widetilde{\vec{v}}_i$ is the current position of vertex i, $\mathbf{V}$ the set of vertices and $\mathbf{B}_i$ the list of neighboring vertices of vertex $\vec{v}_i$.

The Damping term is added to favor solutions in which the positions of the vertices have not changed very much from the input mesh:

$$d\left(\mathcal{Y}\right) = \sum_{i \in \mathbf{V}} \left\| \left(\vec{v}_i - \widetilde{\vec{v}}_i\right) \right\|^2, \tag{4.7}$$

where λ_s and λ_d are tunable parameters we can set to control the impact of individual terms. In our experiments, we set $\lambda_s = 1.5$ and $\lambda_d = 0.8$.

The Interpenetration Algorithm. The mere solution of the objective function minimization might not guarantee the removal of interpenetration for all vertices at once. Therefore we iterate over the described process multiple times to get rid of the interpenetration entirely, as described in Algorithm 1.

4.3 Experiments and Results

In order to assess the performance of our method, we evaluate it on synthetic and real images and videos, both qualitatively and quantitatively. The experiments consist of tight and loose clothing of male and female models with t-shirts and dresses simulated using physically based simulations on mocap data, and rendered under varying camera poses and lighting conditions. We demonstrate garment capture results on single and two-view images.

4.3.1 Datasets

Utilizing the pipeline described in Section 4.1, we simulated around $100,000$ T-shirt meshes on 7 male bodies of various shapes. A geometry dataset was

```
Data: body mesh $\mathcal{B}$ and garment mesh $\mathcal{Y}_0$
Result: impenetrated garment mesh $\mathcal{Y}$
iter=0;
while iter ¡ maxIter do
    Find garment to body vertex correspondences $\mathcal{C}$ ;
    Find penetrating vertices $\mathcal{P} = \left\{ \vec{v}_i \,\middle|\, \forall i : \vec{n_{b_j}}^T \left( \vec{v}_i - \vec{b}_j \right) < 0 \right\}$ ;
    if $empty(\mathcal{P})$ then
        return $\mathcal{Y}_{iter}$ ;
    end
    Solve for: $\mathcal{Y}_{iter+1} = \underset{v_i \in \mathbf{V}_{\mathcal{Y}}}{arg\,min}\, E_{\mathcal{B}}(\mathcal{Y}_{iter})$ ;
    iter = iter + 1;
end
return $\mathcal{Y}_{iter-1}$ ;
```

Algorithm 1: Interpenetration removal algorithm

created, stemming from the T-shirt mesh depicted in Figure 4.6 (Left). Likewise, we simulated around $15,000$ dress meshes on a female character (Figure 4.6 (Right)). We then constructed the final dataset consisting of geometry and corresponding images under different lighting conditions and from front and back views, as explained in Section 4.1. We separate the samples into a training dataset, containing 90% of the images and the corresponding geometries, and a testing dataset consisting of the rest. We would like to stress that our dataset consists of purely synthetic images, hence the training has never seen a real image, but it is still able to capture plausible low-frequency deformations on real data.

4.3.2 Qualitative Evaluation

We firstly assess the estimation quality from the visual perspective and we encourage the reader to view this section electronically.

Synthetic Data. We show results for our ″Garment-from-Body″ mesh representation on the T-shirt dataset in Figure 4.7 and the dress dataset in Figure 4.8, achieving accurate reconstructions. It needs to be noted that the captured wrinkles lack some of the fine details, not present in the images passed to the network, due to the relatively small resolution of of the latter (64×64), as shown in the figures. Nevertheless, we get realistic deformations with dynamic details at different scales preserved for all cases. The algorithm can recover the overall shape and deformation of the garments, as

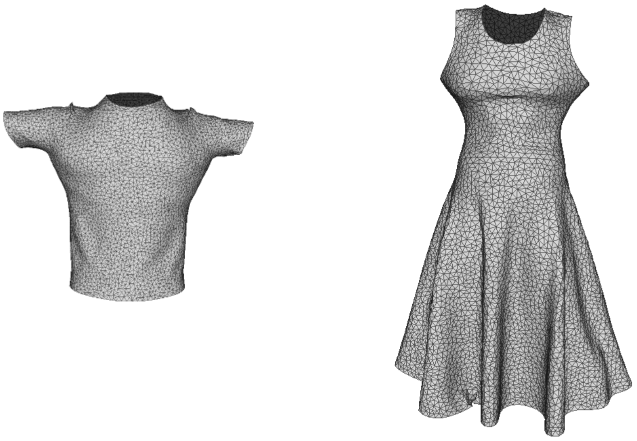


Figure 4.6: *The garment meshes used for simulation*

well as finer wrinkles and folds. One main advantage of our single image-based geometry estimation method, is that we can capture deformations due to a dynamic motion, as opposed to methods that would simulate the garment assuming a known body shape and pose in a statically stable physical state. This is illustrated in the following figures, and can be more clearly seen in Figure 4.8.

As we mentioned in Section 4.1, our generation conditions contain multiple degrees of freedom (DOF), such as camera position, illumination and body pose change. In Figure 4.9, 4.10 and 4.11, we illustrate that we get consistent estimations under different poses, lighting changes, and views, respectively. Incorporating these degrees of freedom into the database thus provides robust results under such changes, which is essential for a practical garment capture system.

Real Data. We evaluate the models trained on the "T-shirt" dataset on real data, that we captured in an uncontrolled environment with a smartphone camera. Figure 4.12 shows the estimation on single-view inputs and Figure 4.13 on two-view inputs, utilizing the respective CNN architectures as explained in Section 4.2. The major deformations in shape and pose are captured accurately and look plausible. This is despite the fact that the material of the captured garment is quite different than the one we have in the database, and the input images to the network are very small. Hence, despite our inability to capture small-scale wrinkle details, we nevertheless obtain quite faithful garment shapes. This generality also allows us to use images depicting textured garments, as we show in Figure 4.13.

Furthermore, we evaluate our dress models on our "dress" dataset (Figure 4.14). Please note, that this is a much more challenging problem, as dresses

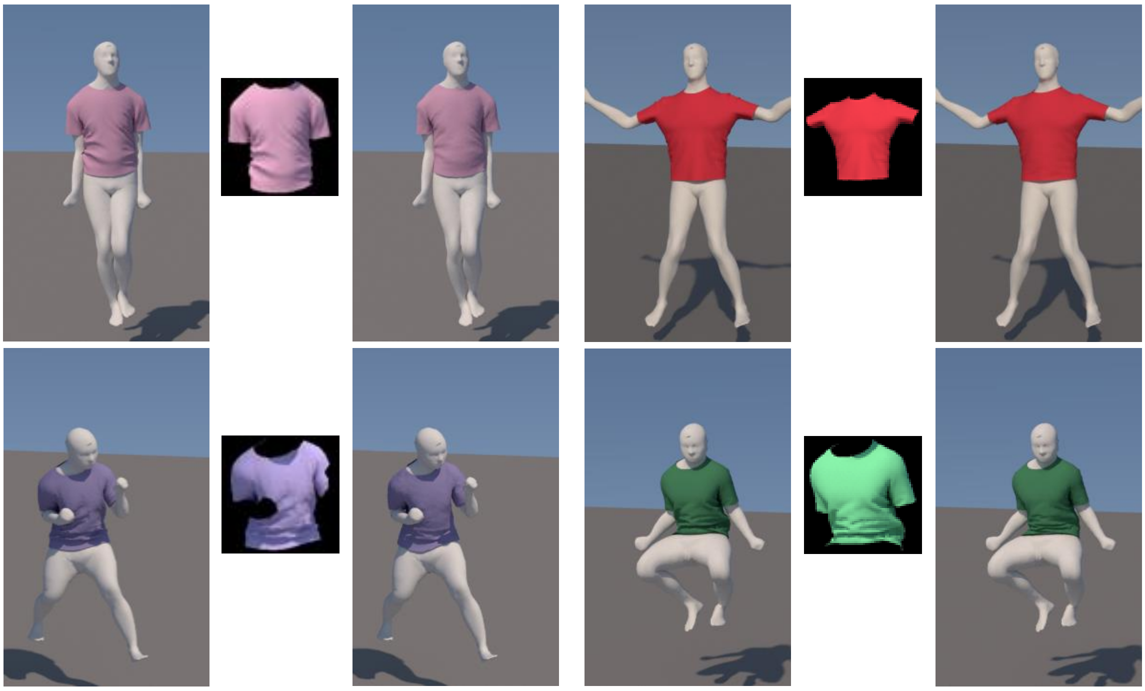


Figure 4.7: *Recovered garment shapes with the "Garment-from-Body" representation. From left to right: initial rendering, segmented T-shirt, and rendering of the same scene using the estimated mesh.*

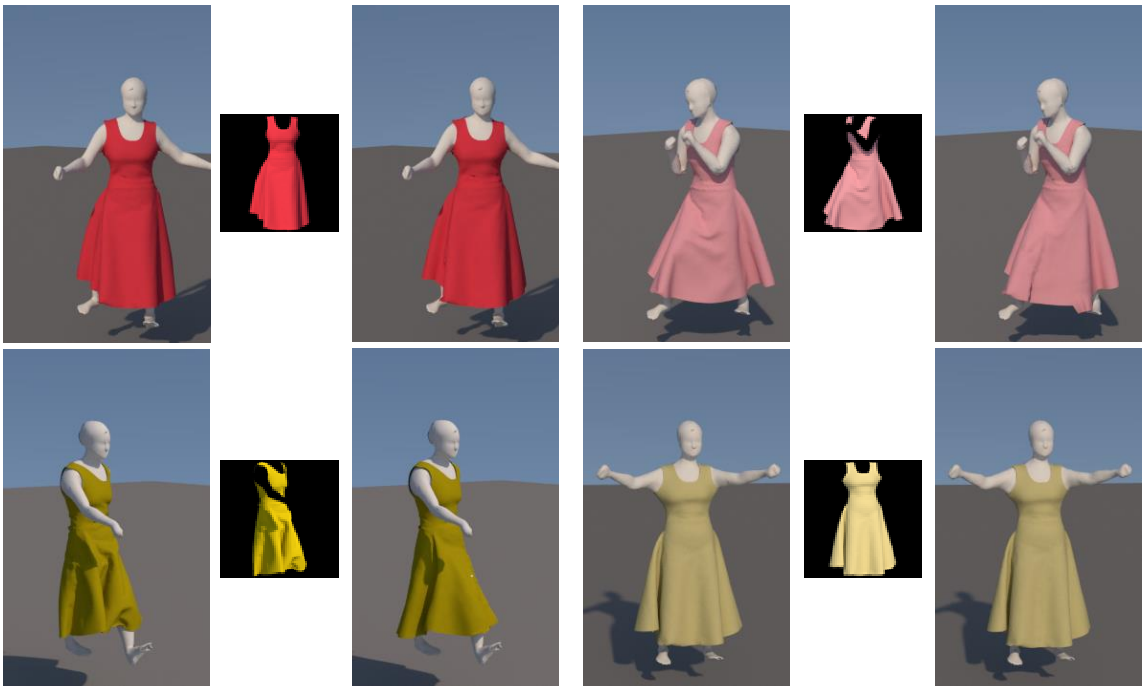


Figure 4.8: *Recovered garment shapes with the "Garment-from-Body" representation. From left to right: initial rendering, segmented dress and rendering of the same scene using the estimated mesh.*

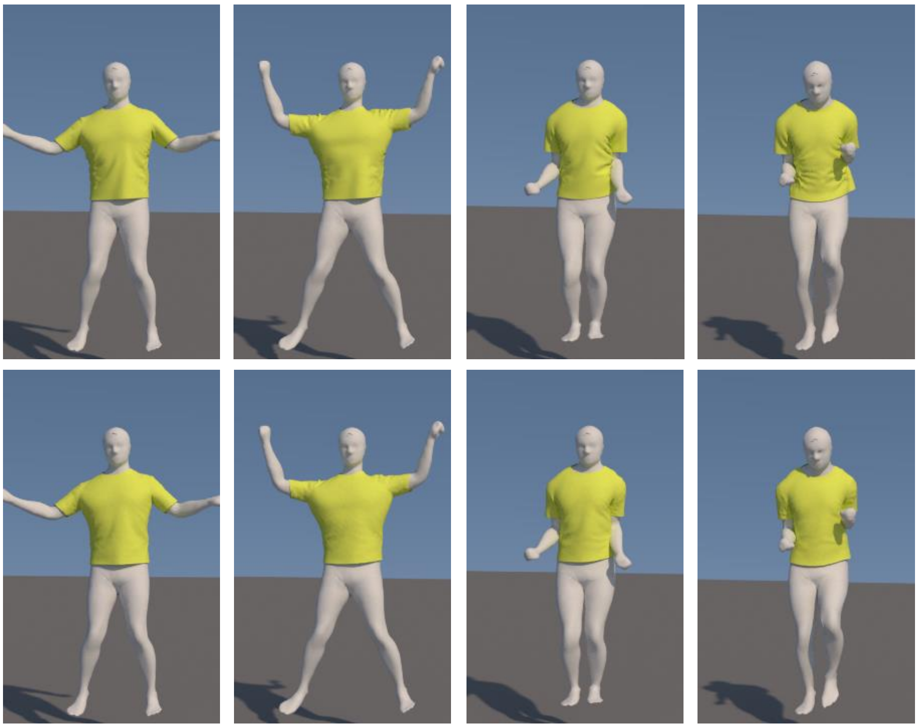


Figure 4.9: *Pose changes: input images (first row) and the estimations (second row).*

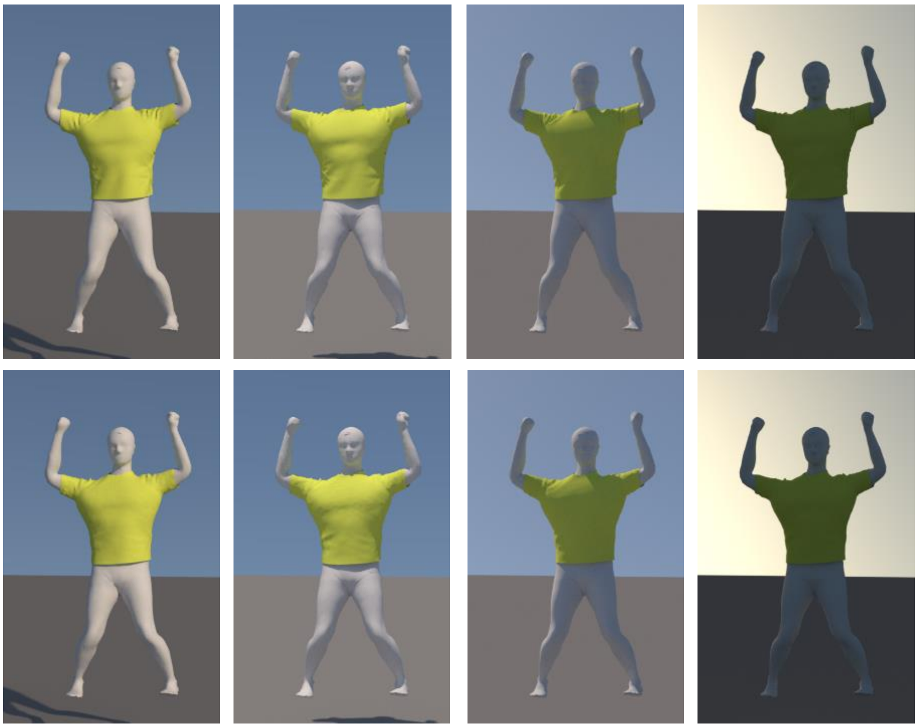


Figure 4.10: *Illumination changes: input images (first row) and the estimations (second row).*

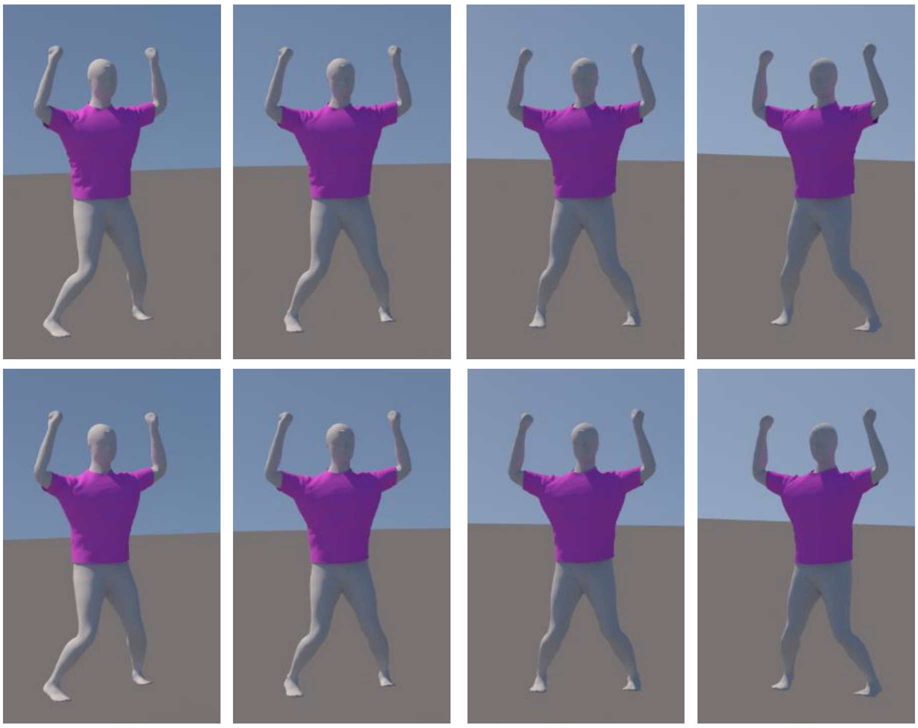


Figure 4.11: *View changes: input images (first row) and the estimations (second row).*

usually have much more variety in both intrinsic shape and material. Despite that, our technique can still recover the global shape.

4.3.3 Quantitative Evaluation

Due to the fact that there was no real-life garment image dataset that would contain accurate ground truth geometry up to this work, we quantitatively evaluate our method on synthetic datasets. For every quantitative experiment, we report the average of the mean squared error of the vertex positions from the ground truth mesh over the entire training set, and the mean cosine similarity of the face normals of the estimated and ground truth meshes, given by $\mathbf{n}^T\mathbf{n}'$ for the ground truth $\mathbf{n}$ and estimated $\mathbf{n}'$ normals.

The results are reported in Tables 4.1 and 4.2. While the absolute errors for the vertex positions per-se do not explicitly inform us on the quality of the individual reconstructions, they serve as a mean to assess and compare the generalization error over the various experiments that we consider, as we elaborate below.

Learned Shape Representation. As mentioned in Section 4.2, we consider two mesh representation formulations. One outputs the PCA coefficients,

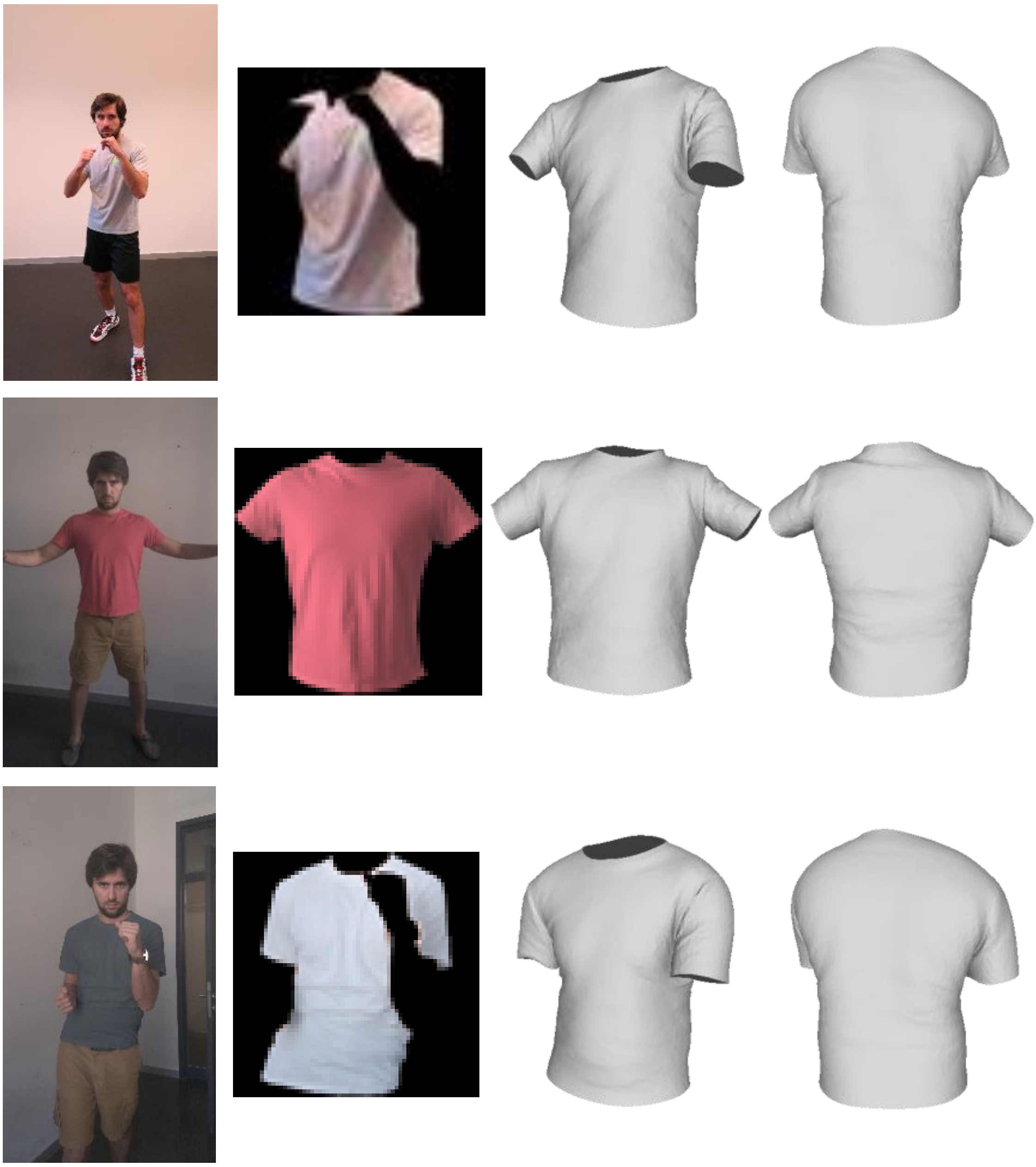


Figure 4.12: *Estimated T-shirt shapes from captured images. From left to right: original image, image after segmentation (input image), view of the estimated mesh from the front and back.*

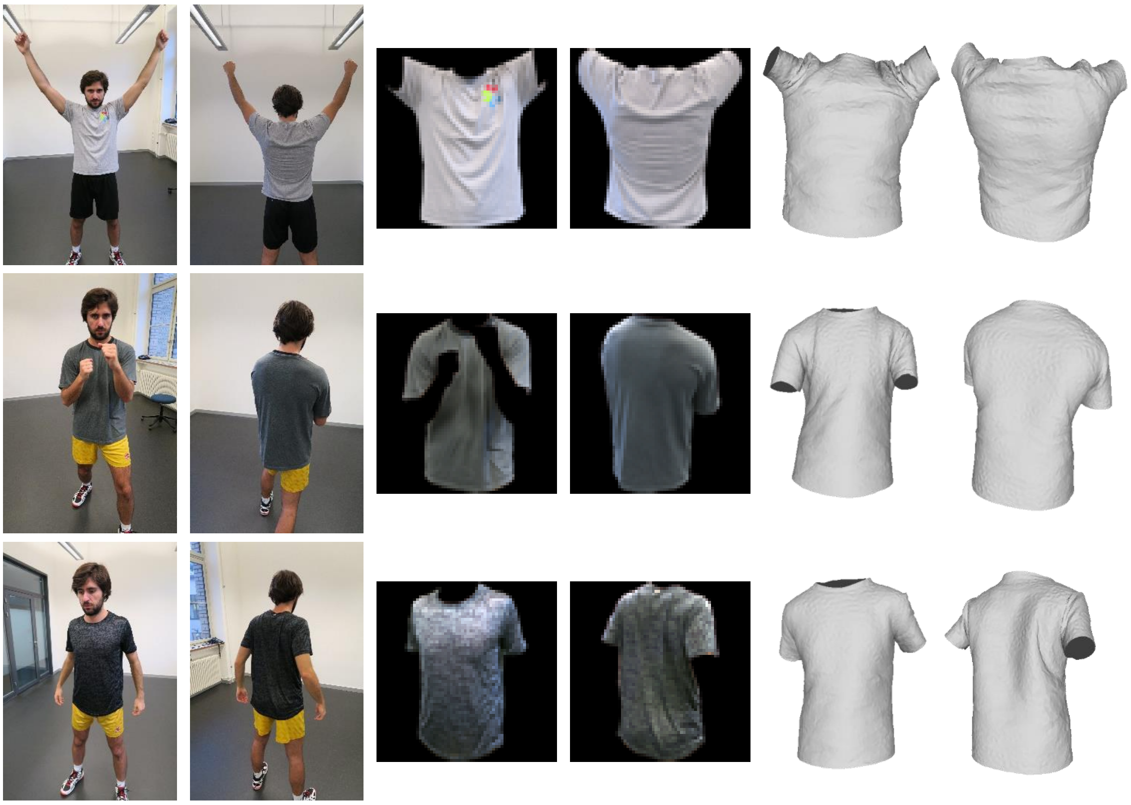


Figure 4.13: *Estimated garment shapes from two views. From left to right: original image from the front and back, image after segmentation (input image) from the front and back, view of the estimated meshes from the front and back.*

which can be used to obtain the per-vertex deformations and the other performs the estimation on the full space, directly outputting the displacement for every vertex. The experiments have shown, that the models using PCA get outperformed significantly, having greater mean squared error and standard deviation. This points to the fact that the deep neural nets do a much better job in creating an internal representation of the data from the training samples than simple PCA. One potentially big disadvantage of the non-PCA model is its size. Because the output layer has over $18,000$ units, the size of the model grows quickly. For instance, while the PCA model of SqueezeNet occupies around 42MB in memory, the full non-PCA model occupies around 660MB.

Garment-from-Garment vs Garment-from-Body. In this experiment, we compare the performance of the two formulations of the regression that we introduced earlier. The "Garment-from-Body" formulation achieves lower reconstruction errors as reported in the tables. This happens because the off-

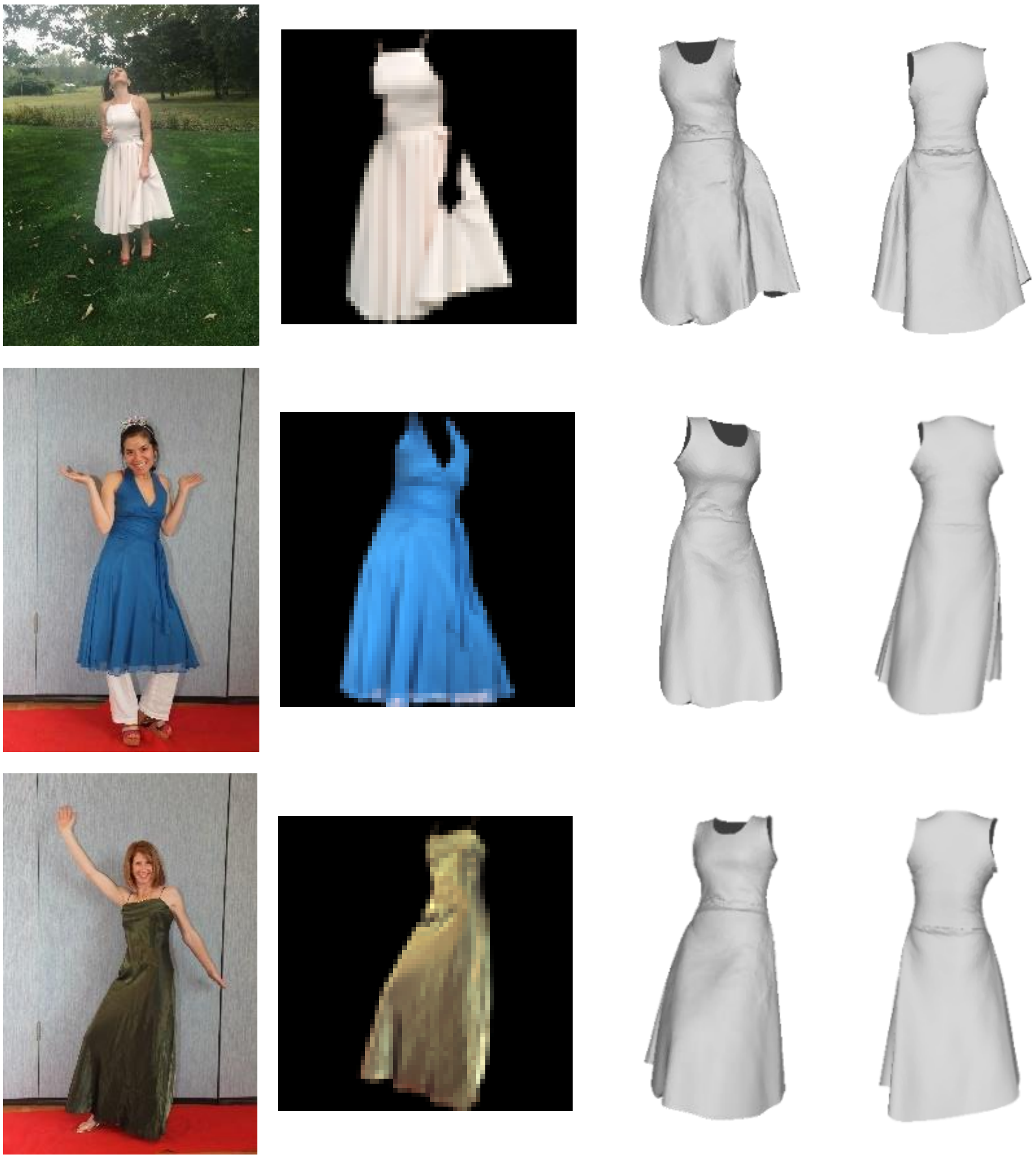


Figure 4.14: *Dress shapes estimated from real images. From left to right: original image, image after segmentation (input image), view of the estimated mesh from the front and back. Please note that none of the dresses match exactly our test dress in neither shape nor material stiffness or reflectance. Despite that, we are able to capture the overall shape even for more challenging images (such as the first image, where the actress grabs the side of her dress).*

Model	MSE	NCS
GfG-PCA space frontal view	507.14±781.390	0.903±0.049
GfG-MSE full space frontal view	342.164±522.742	0.906±0.04
GfG-MSE full space frontal view silhouettes	496.308±765.161	0.901±0.047
GfG-MSE+normals full space frontal view	331.327±557.942	0.916±0.044
GfG-MSE+normalsExp full space frontal view	345.163±607.051	0.921±0.046
GfG-MSE+normals-viewMaxPool full space two views	323.168±472.058	0.917±0.04
GfB-MSE full space frontal view	81.037±205.640	0.908±0.048
GfB-MSE+normals full space frontal view	95.299±194.844	0.900±0.052

Table 4.1: *The performance of models trained on the "T-shirt" dataset. "GfG" stands for the "Garment from Garment" representation. "GfB" stands for the "Garment from Body" representation. Each entry also contains information on which architecture and loss function was used. MSE stands for vertex mean squared error and NCS for mean cosine similarity of the face normals.*

Model	MSE	NCS
GfG-MSE full space frontal view	294.487±303.214	0.937±0.043
GfG-MSE full space frontal view silhouettes	376.824±387.903	0.925±0.052
GfG-MSE+normals full space frontal view	297.833±318.009	0.946±0.04
GfG-MSE+normals-viewConcat full space two views	185.926±222.316	0.965±0.026

Table 4.2: *The performance of models trained on the "Dress" dataset. "GfG" stands for the "Garment-from-Garment" representation, and "GfB" for the "Garment-from-Body" representation. Each entry also contains information on which architecture and loss function was used. MSE stands for vertex mean squared error and NCS stands for mean cosine similarity of the face normals.*

sets from the body tend to be much smaller than those from the reference garment mesh in the T-pose. The scale of the estimated values is smaller and therefore the scale of the error is also smaller. This representation might, however, create a problem if we want to use the estimated garment to dress a body mesh, as the displacement is often too big for the interpenetration solver to work properly without distorting the mesh too much. For this reason, the "Garment-from-Body" formulation is more desirable for dressing characters and "Garment-from-Garment" is more suitable for reconstructing the garment only.

The Importance of Silhouettes. A 2D silhouette of an object is one of the most important visual cues, as it restricts the space of shapes the object could possible have in 3D. For this reason, the following experiment has been conducted: We have trained two models of our SqueezeNet incarnations. One was trained on the image dataset we have, the other was trained only on the silhouettes of the garments, loosing shading and color information. As it can be observed in Table 4.1, the silhouette model performs well, but is outperformed by the model trained on RGB images. This is an important result, as it proves that CNNs can in fact learn the shading cues, due to the wrinkling patterns and further enhance the quality of the estimation. This is also demonstrated in the next chapter. The silhouette though still remains an important cue that the network automatically learns.

Comparison of Single-View and Multi-View Nets. In this experiment, we compare the performance of single and multi-view CNN-s. The multi-view architectures achieve superior performance in comparison to the single view models on both datasets. Hence, our method can benefit from multi-view input to achieve a more accurate estimation. The difference is particularly visible on the dress dataset, as loose clothing has naturally more ambiguities of shape in the occluded parts. Please refer to Figure 4.15 for a visual comparison.

Loss Function with Normals. We compare the performance of the models that are optimized for MSE and those which try to account for the curvature of the surface of the estimated mesh, by using the customized loss function in Eq. 4.3, as described in Section 4.2. The results in Tables 4.1 and 4.2 show that these models usually have slightly worse MSE, but their normals are aligned more accurately.

Failure Cases. In Figure 4.16, we present interesting failure cases for garments with shapes or poses that were not present in the training set. Similarly, sometimes the input image is imperfect and contains noise (such as segmentation artifacts).

4.3.4 Full Specification of the Architecture and Performance

Here, we specify the full details for both single and two-view architectures, including the expansion on Fire layers (as proposed in [Iandola et al., 2016b]). The architectures are captured in Figure 4.17. Here we expand on the choice of architecture. Initially, we experimented with much bigger CNNs, such as Deep Residual Nets. However, due to the size of such networks and the heavy experimentation needed, we decided to opt for SqueezeNet. It delivers comparable performance to the former, it is much faster to train and in our experience it converged more reliably. For this reason, we encourage the use of SqueezeNet for similar problems, as we also do for the Hand Pose estimation task in Section 5.3.2.2. Despite that, it is definitely possible to optimize the results further with deeper architectures.

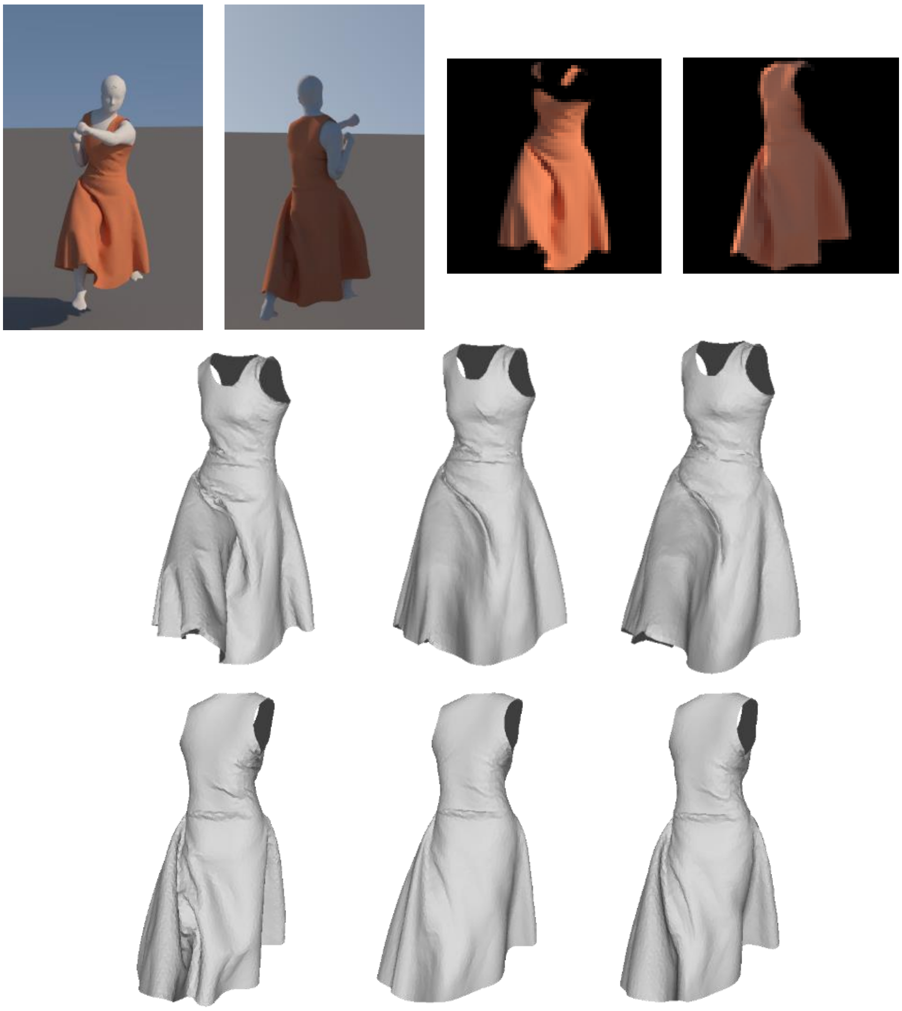


Figure 4.15: *Demonstration of superiority of multi-view models over single-view models. (top) From left to right: initial rendering frontal and back view, the segmented and rescaled garment, frontal and back view. It can be observed that the mesh estimated by the multi-view model is much more accurate than that estimated by the single-view model, especially on the back side. Front (middle) and back (bottom) views of (from left to right): ground truth geometry, geometry estimated by the single-view SqueezeNet model trained on frontal views, and geometry estimated by the multi-view SqueezeNet model.*

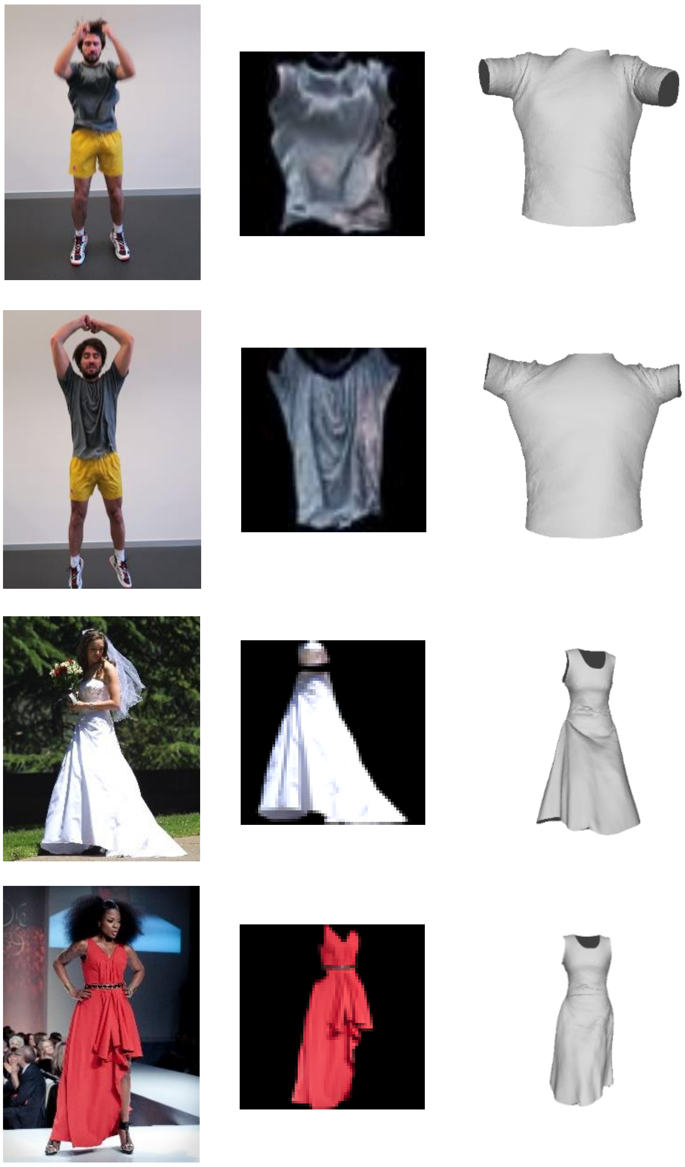


Figure 4.16: *Failure cases. From left to right: original image, image after segmentation (input image), view of the estimated mesh from the front.*
First row: *A very hard image. Similar silhouettes are not present in the training set and the wrinkling patterns are also completely new. Such cases can cause the estimation to fail.*
Second row: *Here, the estimation is hindered by the imperfect segmentation at the top of the image as well as unseen wrinkling patterns.*
Third row: *The specific dress captured slightly from the side is also not part of the training set. However, please observe how our technique tried to generate a very close shape (the wrinkles on the lower left part of the dress).*
Fourth row: *The "intrinsic" shape of this dress is dramatically different. Please observe how our model compensated for lack of clothing on the right with a more plausible inwards deformation.*

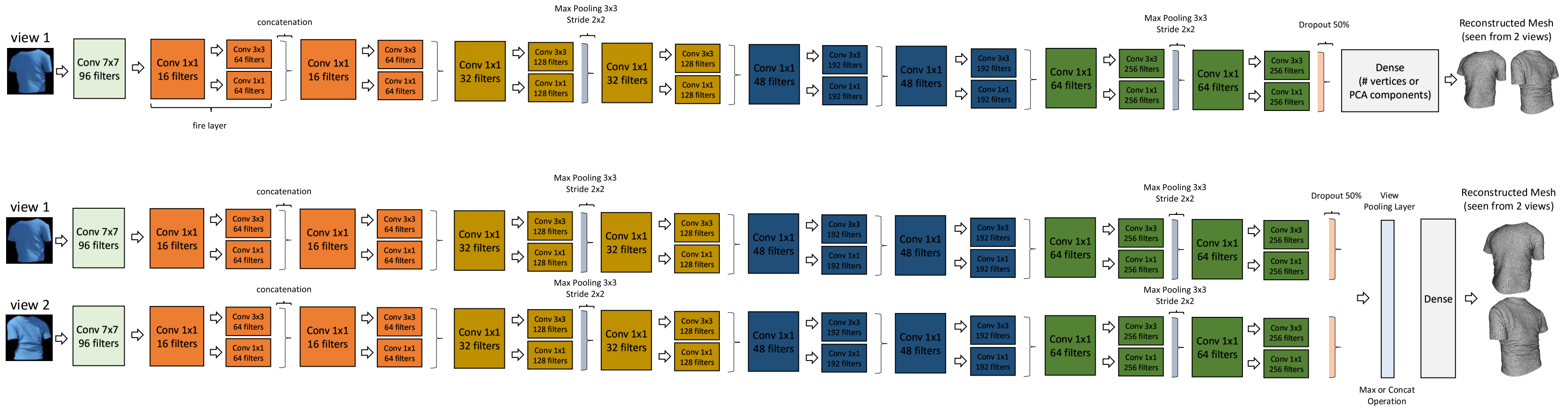


Figure 4.17: ***Top:*** *Single-view architecture.* ***Bottom:*** *Two-view architecture.*

Training Time. Deep neural nets are known for their lengthy training times. However, this is not the case for our architectures. Thanks to the compressed nature of the SqueezeNet architecture, we are able to train our incarnations in less than 8 hours. This corresponds to less than 100 epochs over $100,000$ samples till convergence. Hence, our method scales well to bigger garment databases as well.

Testing Time. Since the overall estimation is a neural net inference followed by estimation of the offsets, the total inference time is a few milliseconds. This makes our method practical even for real-time applications, in contrast to many other 3D shape estimation techniques, or garment simulations.

Data Generation. Our data generation pipeline is not optimized for speed. The generation of our dataset which we utilized for training, with the help of our pipeline described in Section 4.1, took approximately 10 days on a cluster. However, please note that both simulation and rendering is done on a CPU and the process could most certainly be optimized to run orders of magnitude faster.

4.3.5 Additional Experiments

In this section, we show a potential application of our method as a fast approximation of physically based clothing simulation. Furthermore, we perform a simple quantitative experiment on real data.

Fast Garment Simulations. A potential application of our method is the speeding-up of physically based cloth simulation (PBS). PBS is known to be computationally expensive. Therefore, one could simulate very coarse clothing (low resolution mesh), render them with our pipeline, and utilize the resulting rendered images as input for our trained CNN which outputs a higher resolution mesh. As long as the garments used for training and testing are similar in terms of their geometries and materials, this image-based method works well without having to go through any scene-level 3D information, or requiring correspondences between the underlying body, skeleton, or garment meshes. The performance gain is dependent on the resolution of the high-res training and the low-res input meshes. In our experiments, we opted for a low-resolution mesh of 398 vertices instead of the original 6065, resulting in a speed-up of about 8x, allowing the simulation to run in real time. We demonstrate the quality of the resulting garments in Figure 4.18.

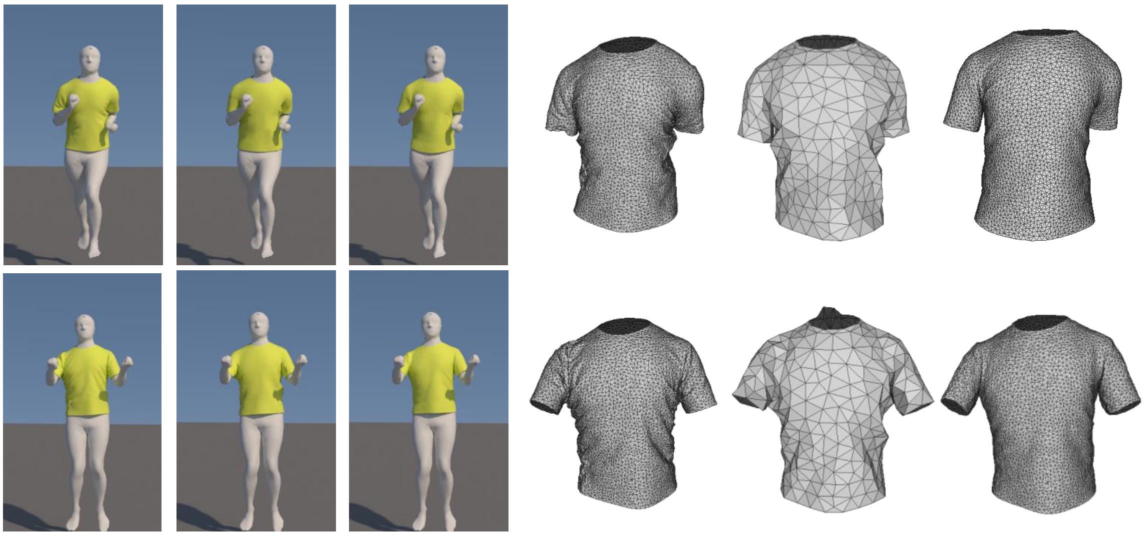


Figure 4.18: *From left to right: simulation with a high resolution mesh, with the corresponding low resolution mesh, and the estimated high resolution mesh with our method. For each case, we also show the raw mesh. The input to our method is the rendering of the simulation with the low resolution mesh.*

4.3.6 Pixel Overlap of the Reconstruction

Performing any meaningful quantitative evaluation on real-life data is not easy because there are no suitable datasets available and creating one is a very challenging task. To the best of our knowledge, such an accurate dynamic clothing capture system is not available. The consumer-level depth cameras are insufficient, as they cannot capture high frequency details.

Therefore, to provide at least some evaluation on the real data we perform the following experiment. We render the reconstructed mesh of the garment in the correct orientation. Then, we compare the image masks, which are the mask of the input image fed to our neural net model and the mask of the rendering of the estimated mesh in the correct orientation. Please note that the results of this experiment should be taken only as an informative lower bound, as our rendering of the final estimation does not contain any occluders such as arms. Furthermore, the experiment does not measure the quality of the reconstructed 3D mesh as a whole. Wrinkles and other finer deformations are not considered by this metric. We report the average pixel overlap for single view and two view T-shirt dataset and also for the single view dress dataset. The averages are 87.9%, 89.4% and 91.4%.

4.3.7 Further Discussion

A novel approach to clothing shape estimation was presented above, however bringing along limitations too, which we will discuss next. We aim to prove our idea correct instead of deploying a ready-to-be-used, production-level solution for a specific purpose. However, we believe our study to be a pioneering one, with the potential to spark a fruitful line of research in this area. The number of use cases and the simplicity of the method has a very big potential for a vast amount of applications. Nevertheless, in order to be broadly usable much more research has to be put into this direction. The work explained in this chapter shall provide an initial setup, which can be explored much further in the future.

Garment Generality. One of the limitations of our technique is that it has to be trained for every type of clothing, as each reference garment mesh may vary dramatically in resolution, depending on its shape complexity.

Impact of the Dataset Generation on Performance. Since our approach is data driven, the performance is only as good as the training dataset. There are many factors that come into play when creating such a dataset, such as the artistic quality of the reference garment design, the resolution of the mesh, the versatility of the motion capture database and the characters animated by it, the correct behavior of the physically based simulation and last but not least, the realism of the renderings (illumination, reflectance, texture, occluders etc.). Please note, that our data generation pipeline is not perfect and as such does not create a perfect dataset. Despite that, we are able to estimate the rough shape of the garments quite accurately. The closer the input image is to the training dataset, the better estimations we get and with very similar images (the synthetic test set), we show that we are even capable of recovering the high frequency detail. We believe that most of the flaws in the estimation could be lifted by creating an ultimate dataset. The creation of such a dataset, however, is another challenging engineering task. Another interesting line of research would be applying Generative Adversarial Nets [Goodfellow et al., 2014] which could help to create bigger variance in our training data. These relatively new approaches of generative and discriminative nets have only recently been used for clothing in lower resolution images [Lassner et al., 2017a].

Dependency on Segmentation. It is true that our technique requires quality segmentation of the garments on the input. Utilizing the recent deep learning based works on image segmentation, such as [Long et al., 2014], could eventually alleviate the need for supervised segmentation in an uncontrolled environment altogether.

4.4 Conclusion

In this chapter, we proposed an end-to-end 3D garment shape estimation algorithm, that captures plausible garment deformations at interactive rates, from an uncontrolled single view setup depicting a dynamic garment state. Additionally, we showed how to achieve this by training a CNN based regressor with statistical priors and specialized loss functions, that utilizes purely synthetically generated data for training, demonstrating its scalability for such a supervised learning task. Finally, we showed how to leverage information simultaneously present in multi-view setups in order to increase the predictive performance of the network and in turn the accuracy of the estimations.

We rely on a data generation pipeline, where the deformation of the human body shape and thus the garment is given by a standard skinning model with automatically computed weights and simple blending of bone transformations. We also use an off-the-shelf physical simulator for cloth deformation, and default parameters. These steps introduce certain artifacts. Better and tailored databases can be obtained by improving these steps, and considering accurate cloth parameters.

Our method relies on segmented garments from an image, although it tolerates a certain amount of noise and inaccuracy, as illustrated in Figure 4.4, and moderately textured garments. By training on more data that cover a larger variety of cases, the method can be extended to handle complex textures and unsegmented images, for non-garment deforming objects as well.

For practical purposes, such as hardware or time constraints, the networks are trained with relatively small input images of 64×64 pixels. This prevents our method from capturing some of the wrinkles. With the current progress in the compactness of CNN representations, one could envisage higher-frequency details to be captured as well.

CHAPTER 5

Hand Pose Estimation

In the previous chapters, we focused on the digital human body as a whole, with an extra layer of clothing added to it. In this chapter, we zoom-in and concentrate on a very dexterous body part - the hand. On a higher level, we slowly shift from a semi pose-invariant, pure shape estimation of the human body and a slight coupling of shape and pose estimation of the garment, to a pure pose estimation of the free human hand and a nuance of internal hand shape estimation to help this process. Looking at it from a machine learning point of view, as opposed to the previous chapters where a supervised training with synthetic datasets was mostly a necessity, partly due to the difficulty of data acquisition, here we explore unsupervised and semi-supervised learning with real unlabeled data. In order to explore the latter, we initially relax our RGB input requirement to a Depth one, and show how a pre-trained convolutional model, can be refined on unseen and unlabeled depth images. Afterwards, we demonstrate it‘s adaptation to the RGB input case. The choice of initially exploring Depth was also influenced by the abundance of existing methods and to a smaller extent respective datasets, as opposed to the RGB case.

5.1 3D Hand Pose Estimation from Depth Data

Recent approaches in 3D hand pose estimation from a single depth image are predominantly based on convolutional neural network architectures [Tompson et al., 2014; Oberweger et al., 2015a; Zhou et al., 2016; Ge et al., 2016; Sinha et al., 2016; Ye et al., 2016], typically requiring labeled data for training. While the accuracy of such methods has been disputed on a limited number of available datasets that are applicable to learning-based approaches, such as [Tompson et al., 2014; Danhang Tang, 2014; Sun et al., 2015], the main problem seems to shift to a large degree towards scarcity of data labeling (e.g. 3D joint positions). This has been particularly demonstrated in [Yuan et al., 2017a], where simply having a bigger and more complete labeled dataset yields much better estimation results, but also in [Supancic et al., 2015], where it is shown that just using nearest-neighbor search methods in the pose data space can already outperform many of the existing, CNN-based methods. Multiple ways of creating labeled data have been presented in the past, usually on the expense of additional set-up environments and man-work. For instance, labels have been generated via optimization [Tompson et al., 2014], utilizing multiple cameras, integrating special sensors [Yuan et al., 2017a] or in a semi-supervised way [Oberweger et al., 2016; Tang et al., 2013]. This is also reflected by the limited amount of public datasets available.

Next to convolutional data-driven approaches, there have been several generative, model-driven ones that perform iterative optimization. For instance, [Melax et al., 2013; Schröder et al., 2014; Tagliasacchi et al., 2015; Taylor et al., 2016; Tkach et al., 2016] optimize for point cloud correspondences while [Oikonomidis et al., 2011; Sharp et al., 2015; Tompson et al., 2014; Qian et al., 2014] attempt to find a good pose, by iteratively rendering many synthetic depth images and comparing them to the input image. Such approaches usually perform better on unseen poses, as compared to data-driven ones, when applied to poses quite dissimilar from the ones in training datasets. Another advantage of these methods is their independence from a big labeled training set. Nevertheless, they usually require temporal information and a good initialization, which typically classifies them as tracking methods, and are computationally more expensive.

Inspired by such optimization approaches and haunted by the problem of creating labeled data, in this section, a novel method is proposed, that bypasses the expensive effort of labeling ground truth data, while still leveraging from the speed of purely data-driven approaches. This combination achieves accurate 3D hand pose predictions, through the help of pre-trained convolutional models that can be refined to unseen and unlabeled depth images. In our case, the pre-training is made purely on a synthetic dataset generated by us. This allows to boost existing data-driven methods, which are mainly optimized for the small amount of available training datasets, to real-world scenarios where labeled data is hard to obtain.

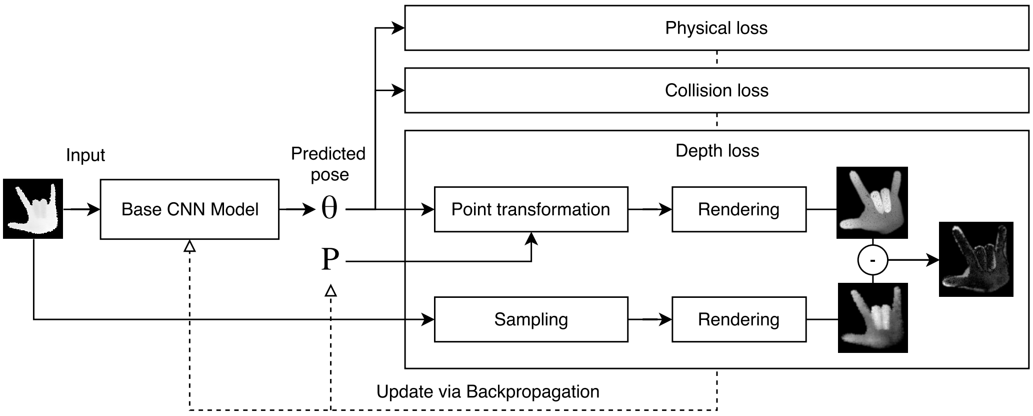


Figure 5.1: *Overview of the training pipeline. Given a depth image as input, a base CNN model predicts the hand pose θ. Given θ, we calculate a loss consisting of a collision, physical and depth component. During training, we update the weights of the base model, as well as P, a point cloud that represents the hand shape and gets iteratively updated to the real one. Since we can calculate the loss using only the input image, θ and P, our model can be trained without labeled data.*

5.1.1 DepthNet Overview

The overview of our method is depicted in Figure 5.1. The main goal is to estimate the 3D hand pose, given a single depth image, utilizing an end-to-end CNN. Specifically we attempt to tackle cases, where the depth input data space does not necessarily represent the training space, due to e.g. variation in hand shape, pose space and sensor noise. We propose to achieve this by refining a base convolutional neural network on unlabeled depth images. The base model is an AlexNet-like [Krizhevsky et al., 2012b] architecture (Section 5.1.2), pre-trained purely on synthetic data to provide a (rough) pose estimate θ. To train on unlabeled data, we propose a combined loss function, containing a depth (Section 5.1.3), collision (Section 5.1.4) and a physical component (Section 5.1.5), as shown in Eq. 5.1 :

$$\mathcal{L} = \mathcal{L}^{depth}(\theta, P) + \mathcal{L}^{\text{coll}}(\theta) + \mathcal{L}^{\text{phys}}(\theta) \tag{5.1}$$

where P is a point cloud representation and estimate of the hand shape in a neutral pose, that gets iteratively adapted to the real one during training. The depth loss adapts the base model to reduce the $L1$ error norm between a synthesized depth image, generated through applying the prediction θ to an updated pointcloud P of a rigged hand model, and a second synthesized

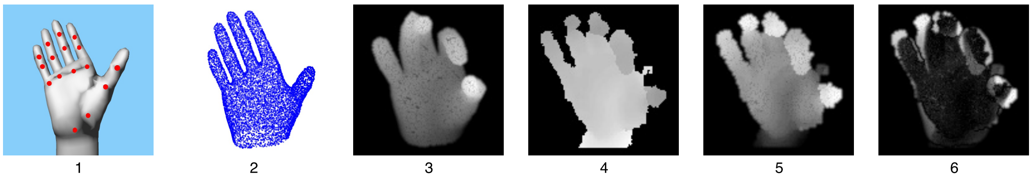


Figure 5.2: *Overview of the different kinds of formats and hand images we process. From left to right (1) our rigged hand model with its 16 joints (2) the uniformly sampled point cloud P of the rigged mesh (3) a rendering after transforming P using the render component (Section 5.1.3.1) (4) a typical noisy depth image input to the base CNN model. (5) a rendering of points sampled from (4). (6) the absolute difference between (3) and (5)*

depth image based only on the input depth image. The collision and physical loss can be thought as regularizers that penalize unnatural looking poses.

5.1.2 Base CNN Model

We start by training a CNN model which can predict a pose θ from a depth image D. To represent θ, we adopt quaternions, however euler angles, rotation matrices or similar structures could possibly be utilized too. Without loss of generality, we chose our base network to be based on the AlexNet architecture [Krizhevsky et al., 2012b].

In order to initially train the network, we generate a lot of synthetic training data, consisting of pairs of depth images and poses in our format of θ. The data is generated from a rigged 3D hand model, with 16 control joints, as depicted in Figure 5.2 (1). Given the depth images of the synthetic training data, we train the base model to minimize the mean squared error between the pose from our dataset and the predicted pose.

Our trained CNN based model could be replaced by any other model that can predict a pose given a depth image. The only constraint in this case, is that θ must be informative enough to calculate a forward kinematic chain, yielding the exact information on how each joint transforms to the predicted pose, as explained in Section 5.1.3.1. The base network is only supposed to give a rough initial prediction. The additional components, explained below, enable it to get refined by training with unlabeled data.

5.1.3 Depth Component

In order to assess the prediction accuracy on unlabeled data, we opt at comparing the input depth image D to a synthesized depth image from our predicted pose θ and a pointcloud P sampled from the hand model. Hence rendering and synthesis of depth images given θ becomes a necessity, which we achieve by imitating the calculations of a common render application and utilizing linear blend skinning (LBS) [Lewis et al., 2000] to transform the points according to θ.

5.1.3.1 Point Transformation

Forward Kinematics. The point transformation behaves similar to a render application. This includes the forward kinematic chain, which yields for each joint the transformation matrix, transforming from the model space of a base pose into the model space of the skinned pose θ. This step is differentiable, as it only consists of matrix multiplications and trigonometric functions. We denote with $M = [M_1, \dots, M_m]$ those transformation matrices, where m is the number of joints used.

Linear Blend Skinning. In contrast to [Zhou et al., 2016] though, we do not just transform each joint position to its final position, but a bigger set of points $P = [p_1, \dots, p_n]$ representing the whole hand, where each point is associated with one or more joints. Let $w_{i,j}$ be the weight which defines how much the point p_i is bound to the joint j. Linear blend skinning (LBS) [Lewis et al., 2000], $f^{\text{skin}}(P, M)$, transforms each point by a linear combination of the matrices M_j according to its weights:

$$\hat{p}_i := f_i^{\text{skin}}(P, M) = \sum_{j=1}^{m} w_{i,j} M_j p_i \tag{5.2}$$

It can be noted that this method is not just differentiable with respect to M, which is a very important property that allows backpropagation to the base model, but also with respect to P. This allows us to relax the static hand model to a dynamic one, that gets updated during training to automatically adapt to the hand shape. In order to give an intuitive advantage of this approach, imagine a personalized adaption to a different real person's hand shape, starting from a non-parametric 3D hand model. This becomes important, since in reality, not only the poses change but also the hand shapes.

5.1.3.2 Depth Rendering

The 3D hand model shape and pose can be adapted to the real hand shape and pose by iteratively minimizing a difference in depth projections ($\mathcal{L}^{depth}$), of points P and P_D, sampled from the hand model and the input depth image D, respectively. Instead of utilizing triangles as primitives, that involve a difficult rasterization step and produce an image which is not differentiable, we make use of a method based on [Roveri, 2018], to render a point cloud in a differentiable way. Instead of transforming the vertices of the model, we actually transform a point cloud P. P and its weights W are once uniformly sampled from the hand model, which acts therefore only as a hand shape prior.

In order to render P in a differentiable way, we select only the points with the lowest z-value, which are the ones closest to the camera, for each of the image coordinates ($D_{i,j}$), and weight the z-value of each point with a 2D basis function ϕ around its position. This weighting (smoothing) step is important since otherwise, only picking a depth value at each widely spaced sampled point would make the method non-differentiable. Let $p_i = [p_{i,x}, p_{i,y}, p_{i,z}] \in PCL$. The rendered depth image approximation is defined as:

$$f_{i,j}^{\text{depth}}(P) = \max_k(\text{depth}_{i,j}(p_k)) \tag{5.3}$$

where we assume the points to be in the $[0, 1]$ range and the z-values to represent the depth with respect to the camera:

$$\text{depth}_{i,j}(p) = (1 - p_z)\phi_{i,j}(p) \tag{5.4}$$

We choose $\phi \in C^1$ to have finite spatial support of a circle with radius r. Let $\text{dist}_{i,j}^2(p) = (j - p_x)^2 + (i - p_y)^2$. We can define ϕ as:

$$\phi_{i,j}(p) = \left(1 - \left(\frac{\text{dist}_{i,j}(p)}{r}\right)^2\right)^2 \mathbb{1}_{\text{dist}_{i,j}^2(p) < r^2} \tag{5.5}$$

Even though the rendered images look like depth images, there is still a visible disparity between the synthesized and real images, as it can be seen in between Figure 5.2 (3) and (4). Therefore we also sample a point cloud P_D from the real depth image D and render it using f^{depth}, as in Figure 5.2 (5). The actual loss taken in the end is the L1 norm of the difference between both synthesized images, Figure 5.2 (3) and (5):

$$\mathcal{L}^{\text{depth}} = \sum_{i,j} |f_{i,j}^{\text{depth}}(f^{\text{skin}}(P, M)) - f_{i,j}^{\text{depth}}(P_D)| \tag{5.6}$$

5.1.4 Collision Component

Inspired by [Tagliasacchi et al., 2015] and [Melax et al., 2013], we also attempt to avoid finger interpenetration by penalizing over a self-collision approximation of the hand mesh. We approximate the hand with cylinders and check for each cylinder if a joint position is inside. Let $B = [b_1, \dots b_m]$ denote the joint positions calculated using the joint transformation matrices M (see 5.1.3.1). Let $C \subset \mathbb{N} \times \mathbb{N}$ be all joint indices paired with their parent. Hence, each pair $(i, j) \in C$ describes a bone. We define the loss as:

$$\mathcal{L}^{\text{coll}} = \frac{1}{m} \sum_{(i,j) \in C} \sum_{k=1}^{m} \mathbb{1}_{i \neq k, j \neq k} f^{\text{coll}}(b_i, b_j, b_k) \tag{5.7}$$

where $f^{\text{coll}}(a, b, p)$ is the penetration depth of a point p into a cylinder with the endpoints a and b and a fixed radius. The radius could be determined using the point cloud P of our hand shape but we obtained reasonable results by choosing a fixed value.

5.1.5 Physical Component

The physical loss is defined similarly to [Zhou et al., 2016] and [Tagliasacchi et al., 2015]. We first transform our pose θ (which we represent in quaternions) to euler angles using the function $\varphi(\theta)$. We can specify a valid range $[\underline{\varphi}, \bar{\varphi}]$ for each angle, in euler angles. The bounds are determined manually by looking at the model while varying the pose. The loss penalizes poses outside the specified range:

$$\mathcal{L}^{\text{phys}}(\theta) = \max(\underline{\varphi} - \varphi(\theta), 0) + \max(\varphi(\theta) - \bar{\varphi}, 0) \tag{5.8}$$

In Table 5.1, we provide extra detail on the bounds for the valid range $[\underline{\varphi}, \bar{\varphi}]$ of the euler angles for each finger part/joint of the rigged 3D hand model, depicted in Figure 5.2 and downloaded online from Turbosquid [1]. An angle of zero represents the joint/finger in its neutral pose, which could be defined arbitrarily, but in our case is the open palm. There are three rotation axes. The first one has the same axis as the one from the finger. The second axis rotates the finger to the side (e.g. to account for the separation between the fingers). The third axis rotates the finger towards the palm. We first rotate around the first axis, then around the third axis and lastly around the second one.

[1] https://www.turbosquid.com/3d-models/rigged-male-hand-max/786338

Finger Part	*1st Rotation Axis*	*2nd Rotation Axis*	*3rd Rotation Axis*
1. Index Finger Lower	[-1.4, 0.1]	[-0.1, 0.4]	[-0.1, 0.6]
2. Index Finger Middle	[0, 0]	[-0.1, 0.1]	[-0.1, 1.2]
3. Index Finger Upper	[0, 0]	[0, 0]	[-1.0, 1.0]
4. Middle Finger Lower	[0, 0]	[-0.8, 0.2]	[-0.1, 1.9]
5. Middle Finger Middle	[0, 0]	[0, 0]	[-0.7, 1.2]
6. Middle Finger Upper	[0, 0]	[0, 0]	[-0.3, 1.6]
7. Ring Finger Lower	[0, 0]	[-0.7, 0.2]	[-0.1, 1.8]
8. Ring Finger Middle	[0, 0]	[0, 0]	[-0.7, 1.5]
9. Ring Finger Upper	[0, 0]	[0, 0]	[-0.3, 1.6]
10. Little Finger Lower	[0, 0]	[-0.5, 0.2]	[-0.1, 1.8]
11. Little Finger Middle	[0, 0]	[0, 0]	[-0.5,1.5]]
12. Little Finger Upper	[0, 0]	[0, 0]	[-0.3, 1.6]
13. Thumb Lower	[0, 0]	[-0.4, 0.4]	[-0.2, 1.8]
14. Thumb Middle	[0, 0]	[0, 0]	[-0.5, 1.5]
15. Thumb Upper	[0, 0]	[0, 0]	[-0.3, 1.6]

Table 5.1: *Allowed euler angle bounds for each joint or finger part. Values are in radians.*

5.1.6 Derivation and Implementation

The linear blend skinning, the depth renderer and the collision component are implemented as custom operations in Tensorflow using CUDA kernels. To implement a custom operation $f : A \mapsto B$ in Tensorflow, the explicit gradient of $g(f)$ has to be given, where $g : B \mapsto \mathbb{R}$ is an arbitrary loss function.

5.1.6.1 Linear blend skinning

Let m be the number of joints. Let $M = [M_1, \dots, M_m]$, where $M_i \in \mathbb{R}^{4\times4}$ is the matrix, that transforms from joint space of joint i to view space for the hand pose θ (calculated with the forward kinematic chain). Let $P = [p_1, \dots, p_n]$ with $p_i \in \mathbb{R}^4$ all point positions in homogeneous coordinates and $w_{i,j} \in W \in \mathbb{R}^{n\times m}$ the weight, that defines how much point p_i is bound to the joint $j \in [m]$.

Derivative w.r.t M. We write f^{skin} in terms of a scalar values with $c \in [4]$ as coordinate axis:

$$f^{\text{skin}}_{i,c}(P, M) = \sum_{d=1}^{4} \sum_{j=1}^{m} w_{i,j} M_{j,c,d} p_{i,d} \tag{5.9}$$

The derivative of $f := f^{\text{skin}}$ is:

$$\frac{\partial f_{i,c}(P,M)}{\partial M_{k,l,m}} = \mathbb{1}_{\{l=c\}} w_{i,k} p_{i,m} \tag{5.10}$$

The loss function $g(f)$ can be differentiated using the chain rule:

$$\begin{aligned}\frac{\partial}{\partial M_{k,l,m}} g(f(M,P)) &= \sum_{i=1}^{n} \sum_{c=1}^{4} \frac{\partial g(f)}{\partial f_{i,c}} \frac{\partial f_{i,c}(P,M)}{\partial M_{k,l,m}} &(5.11)\\ &= \sum_{i=1}^{n} \sum_{c=1}^{4} \frac{\partial g(f)}{\partial f_{i,c}} \mathbb{1}_{\{l=c\}} w_{i,k} p_{i,m} &(5.12)\\ &= \sum_{i=1}^{n} \frac{\partial g(f)}{\partial f_{i,l}} w_{i,k} p_{i,m} &(5.13)\end{aligned}$$

Derivative w.r.t P. To update P, we also need to differentiate f^{skin} w.r.t. P. In the equations above, we left out the batch dimension of f^{skin}, since all the operations can be done in parallel and independently for each batch element. However, our points P are the same for each batch element, hence more precision is required here. Let $f^{\text{skin}} \in \mathbb{R}^{n \times 4} \times \mathbb{R}^{N \times n \times 4} \mapsto \mathbb{R}^{N \times n \times 4 \times 4}$ where n is the number of points and N the size of the batch. Note that also M exists for each batch element and therefore has now four dimensions. For a batch index b, point index i and coordinate index c we rewrite Eq. 5.9 as:

$$f^{\text{skin}}_{b,i,c}(P,M) = \sum_{d=1}^{4} \sum_{j=1}^{m} w_{i,j} M_{b,j,c,d} p_{i,d} \tag{5.14}$$

Taking the derivative of $f := f^{\text{skin}}$ gives:

$$\frac{\partial f_{b,i,c}(P,M)}{\partial p_{k,l}} = \mathbb{1}_{\{k=i\}} \sum_{j=1}^{m} w_{i,j} M_{b,j,c,l} \tag{5.15}$$

And the derivative of $g(f)$ is given by:

$$\begin{aligned}\frac{\partial}{\partial p_{k,l}} g(f(M,P)) &= \sum_{b=1}^{N} \sum_{i=1}^{n} \sum_{c=1}^{4} \frac{\partial g(f)}{\partial f_{b,i,c}} \frac{\partial f_{b,i,c}(P,M)}{\partial p_{k,l}}\\ &= \sum_{b=1}^{N} \sum_{i=1}^{n} \sum_{c=1}^{4} \frac{\partial g(f)}{\partial f_{b,i,c}} \mathbb{1}_{\{k=i\}} \sum_{j=1}^{m} w_{i,j} M_{b,j,c,l}\\ &= \sum_{b=1}^{N} \sum_{c=1}^{4} \frac{\partial g(f)}{\partial f_{b,k,c}} \sum_{j=1}^{m} w_{i,j} M_{b,j,c,l} &(5.16)\end{aligned}$$

5.1.6.2 Depth Renderer

Let $f := f^{\text{depth}}$. The derivative of a loss function $g(f)$ for $c \in \{x, y, z\}$ is given by:

$$\frac{\partial}{\partial p_{l,c}} g(f(P)) = \sum_{i=1}^{120} \sum_{j=1}^{120} \frac{\partial g(f_{i,j})}{\partial f_{i,j}} \frac{\partial f_{i,j}(P)}{\partial p_{l,c}} \tag{5.17}$$

with

$$\frac{\partial f_{i,j}(P)}{\partial p_{l,x}} = h(l, P)(j - p_{l,x}) \tag{5.18}$$

$$\frac{\partial f_{i,j}(P)}{\partial p_{l,y}} = h(l, P)(i - p_{l,y}) \tag{5.19}$$

$$\frac{\partial f_{i,j}(P)}{\partial p_{l,z}} = -\mathbb{1}_{\{l = \text{argmax}_k(\text{depth}_{i,j}(p_k))\}} \phi(p_l) \tag{5.20}$$

where

$$h(l, P) = \mathbb{1}_{\{l = \text{argmax}_k(\text{depth}_{i,j}(p_k)) \wedge \phi(p_l) > 0\}} \frac{4(1 - p_{l,z})}{r^2} \left(1 - \frac{(j - p_{l,x})^2 + (i - p_{l,y})^2}{r^2}\right) \tag{5.21}$$

To implement the loops over the image from Eq. 5.17 efficiently, we can make use of the finite spatial support of ϕ and loop only over a range of $\{\lfloor p_{l,x} - r \rfloor, \ldots, \lceil p_{l,y} + r \rceil\}$ for each point p_l, which is for some indices i, j with $1 \leq i, j \leq 120$ fulfilling $l = \text{argmax}_k(\text{depth}_{i,j}(p_k))$.

5.1.6.3 Collision Component

Let $B := [b_1, \ldots, b_m] \in \mathbb{R}^{m \times 3}$ the joint positions, $C \subset \mathbb{N} \times \mathbb{N}$ be all joint indices paired with their parent. Let $\mathcal{L} := \mathcal{L}^{\text{coll}}$ and $f := f^{\text{coll}}$. We define f by:

$$f(a, b, p) = \max(0, \min(f_1(a, b, p), f_2(a, b, p))) \tag{5.22}$$

with

$$f_1(a, b, p) = (0.5h)^2 - \text{dist}_h(a, b, p) \tag{5.23}$$

and

$$f_2(a, b, p) = r^2 - \text{dist}_r(a, b, p) \tag{5.24}$$

where $\text{dist}_h(a, b, p)$ and $\text{dist}_r(a, b, p)$ are the squared distances of a point p from $\mu = \frac{a+b}{2}$ in cylindrical coordinates of a cylinder with endpoints a and b

and height h:

$$\text{dist}_h(a,b,p) = \left(\frac{2}{h}(b-\mu)^T(p-\mu)\right)^2 \tag{5.25}$$

$$\text{dist}_r(a,b,p) = \|p-\mu\|^2 - \text{dist}_h(a,b,p) \tag{5.26}$$

The derivative of $g(\mathcal{L})$ is given by:

$$\frac{\partial}{\partial b_l} g(\mathcal{L}(B)) = \frac{g'(\mathcal{L})}{m} \sum_{(i,j)\in C} \sum_{k=1}^{m} \mathbb{1}_{i\neq k, j\neq k} \frac{\partial f(b_i, b_j, b_k)}{\partial b_l} \tag{5.27}$$

with

$$\frac{\partial f(b_i, b_j, b_j)}{\partial b_l} = -\mathbb{1}_{\{\min(f_1,f_2)>0\}} (\mathbb{1}_{\{f_1<f_2\}} \frac{\partial\, \text{dist}_h}{\partial b_l} + \mathbb{1}_{\{f_1>f_2\}} \frac{\partial\, \text{dist}_r}{\partial b_l}) \tag{5.28}$$

and

$$\frac{\partial\, \text{dist}_h(b_i, b_j, b_k)}{\partial b_l} = \frac{4}{h}\sqrt{\text{dist}_h} \cdot \begin{cases} b_i - b_k, & \text{if } l = i \\ b_k - b_j, & \text{if } l = j \\ b_j - b_i, & \text{if } l = k \end{cases} \tag{5.29}$$

$$\frac{\partial\, \text{dist}_r(b_i, b_j, b_k)}{\partial b_l} = \frac{\partial\, \text{dist}_h}{\partial b_l} + \begin{cases} 2b_k - b_i - b_j, & \text{if } l = i \\ 0.5(b_i + b_j) - p_k, & \text{if } l = j \\ 0.5(b_i + b_j) - p_k, & \text{if } l = k \end{cases} \tag{5.30}$$

Please note that we left out the arguments (b_i, b_j, b_j) for clarity in Eq. 5.28, 5.29 and 5.30.

5.2 Experiments and Results

5.2.1 Architecture and Training Details

As mentioned in Section 5.1.2, we utilize a slightly modified AlexNet [Krizhevsky et al., 2012b] architecture as our candidate base model.

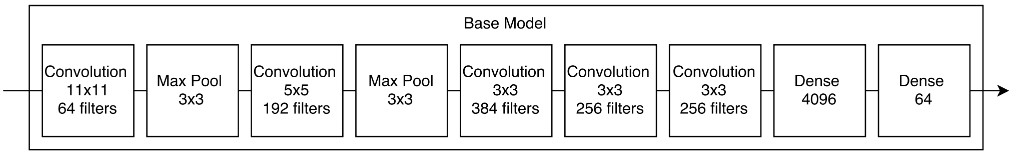

Base Model
Convolution 11x11 64 filters
Max Pool 3x3
Convolution 5x5 192 filters
Max Pool 3x3
Convolution 3x3 384 filters
Convolution 3x3 256 filters
Convolution 3x3 256 filters
Dense 4096
Dense 64

Figure 5.3: *Architecture of the base CNN model*

Similar to [Zhou et al., 2016] and the method from Section 3.4, we adopt it for regression, however we experienced that removing one of the two fully connected layers achieves a faster learning with similar performance. Hence, we only have one fully connected layer with 4096 neurons using a ReLU activation function. Before linearly regressing to θ, we add a dropout layer with (keep) probability 0.75. For details on the architecture please check Figure 5.3.

As input we expect a batch of 120×120 pixels depth images, with the depth values scaled in a $[0, 1]$ range. The hand is cropped and centered, by padding on the sides when necessary such that the aspect ratio is preserved. We choose to regress to quaternions and therefore $\theta \in \mathbb{R}^{16\times 4}$, since there are 16 joints.

The network is pre-trained on synthetic depth images with randomly generated poses. The poses are sampled from a feasible angle range and collisions are avoided utilizing a similar approximation, as in Section 5.1.4. We train the base model on synthetic data until convergence, with the Adam Optimizer [Kingma and Ba, 2014] and a learning rate of 10^{-3}. The complete network is trained on unlabeled data with a learning rate of 10^{-5}, for 10 epochs for each training set. We choose a batch size of 200 and 1000 for training on the labeled synthetic and unlabeled real data respectively.

Method Speed. We conducted our experiments on an Intel i7 860 (from 2010), 8 GB of RAM with an Nvidia Geforce GTX 1060. A forward pass through the network for predicting a single input image takes 3.5 ms, which is practically real-time. Training unsupervised for 10 epochs, as we did, takes around 30 minutes.

5.2.2 Datasets

We evaluate our method on two public datasets (NYU and ICVL) and a separate one created by us. The NYU [Tompson et al., 2014] dataset is recorded using a Microsoft Kinect sensor and provides therefore, in comparison to the other dataset, very noisy images. The training set has 72757 images from one person and three simultaneous views. The test set is a sequence of 8252 images from two persons. The ground truth annotations are fitted with an offline PSO approach and consist of 36 3D spatial hand features per frame. Similar to previous works [Zhou et al., 2016], we also use only a subset of 16 3D positions for evaluation.

The ICVL dataset [Danhang Tang, 2014] is less noisy due to the usage of the Intel Creative Interactive Gesture Camera. Two test sequences from two different persons with a total of 1596 frames are provided for testing and about 180K frames for training from several persons. The ground truth of 16 3D bone center locations is obtained utilizing the tracking method proposed by [Melax et al., 2013]. We segment the images of NYU and ICVL by cutting out a padded block around the 3D annotations. Furthermore, it is important to mention that our 3D joint positions, that are dependent on our fixed 3D model skeleton, deviate a lot from the ones used in both datasets, which is important for a fair comparison.

Our own dataset is created using the Intel RealSense Camera and consists of 2000 depth images for testing and 50000 depth images for training, from only one person wearing a black wristband. This allows for a simple brightness based segmentation to cut out the wrist, which makes it easy to separate the foreground from the background. We also capture a color image for each frame for qualitative comparison. Note that we do not provide any annotations. We compare to different methods by computing ROC curves, that denote the fraction of frames below a maximum 3D join prediction error.

5.2.3 Feasibility of Learning Hand Pose and Shape

It has been shown by [Loper and Black, 2014] how to estimate the human body shape from depth (and color) image differences using a gradient based method, but except for the concurrent optimization based approaches on differentiable offline [Remelli et al., 2017] and online [Tkach et al., 2017] calibration for hands, to the best of our knowledge, there exist no CNN based works in the field of hand pose estimation. In methods utilizing PSO [Oikonomidis et al., 2011; Tompson et al., 2014; Qian et al., 2014; Sharp et al., 2015], we have seen that an error metric on depth images is meaningful enough to intelligently sample, compare and prune candidate poses, however the gradient idea has not been exploited. Encouraged by such results, we show in two experiments that we can optimize for the hand pose and shape.

Hand Pose Optimization. In our first experiment we attempt to overfit our model on single images, in order to show that reasonable results are ob-

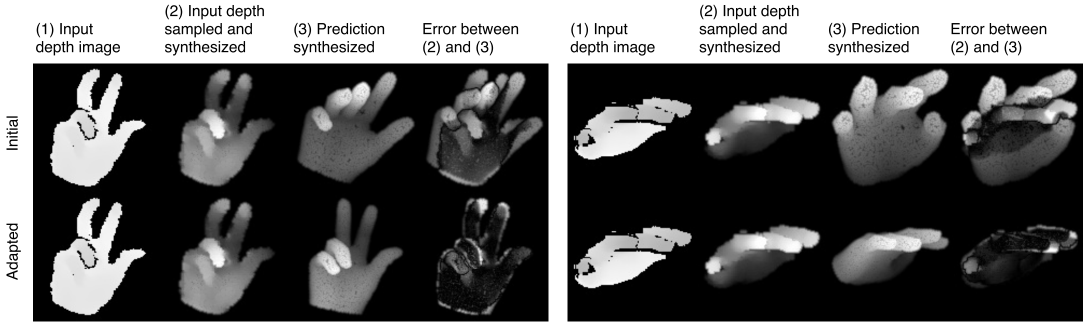

(1) Input depth image
(2) Input depth sampled and synthesized
(3) Prediction synthesized
Error between (2) and (3)
(1) Input depth image
(2) Input depth sampled and synthesized
(3) Prediction synthesized
Error between (2) and (3)
Initial
Adapted

Figure 5.4: *Firstly, we overfit our model on single depth images to give evidence that learning the pose without any annotations is possible.*

tainable despite our loss function being highly non-convex. We take images from our training set and train the network for 70 epochs (update steps). In Figure 5.4, for each block, we show the initial prediction of our base model (on top) and the prediction after training (at the bottom). We typically obtain good results, where the error between the depth images is minimized, except for cases when the initial prediction is too far from the actual pose, impeding convergence to a desired minimum.

Hand Shape Optimization. We also explore whether the point cloud P is adequately adapted to the hand shape. For that, we perform the same experiment as before, however we keep the weights of the model fixed, such that only P gets updated. As shown in Figure 5.5, all our test runs converge slowly to an optimum which almost completely vanishes the loss over the synthesized depth images.

5.2.4 Self Comparison

Given the previous results on single images, it is important to demonstrate that our model can also adapt to complete datasets. Our attempt is not to overfit to any dataset, but generalize to similar inputs by adapting to the sensor noise and hand shape in the training set. To show this, we start with a **quantitative** self comparison on the NYU [Tompson et al., 2014] and ICVL [Danhang Tang, 2014] datasets. We train our base model twice, unsupervised for each of the training sets, where first we use the depth component only and then all losses altogether.
The results on the validation set are shown in Figure 5.6. We can see a significant improvement in both datasets by training with the depth component only. Incorporating the physical and collision component gives only a very small improvement on ICVL, but a second big improvement on NYU. This

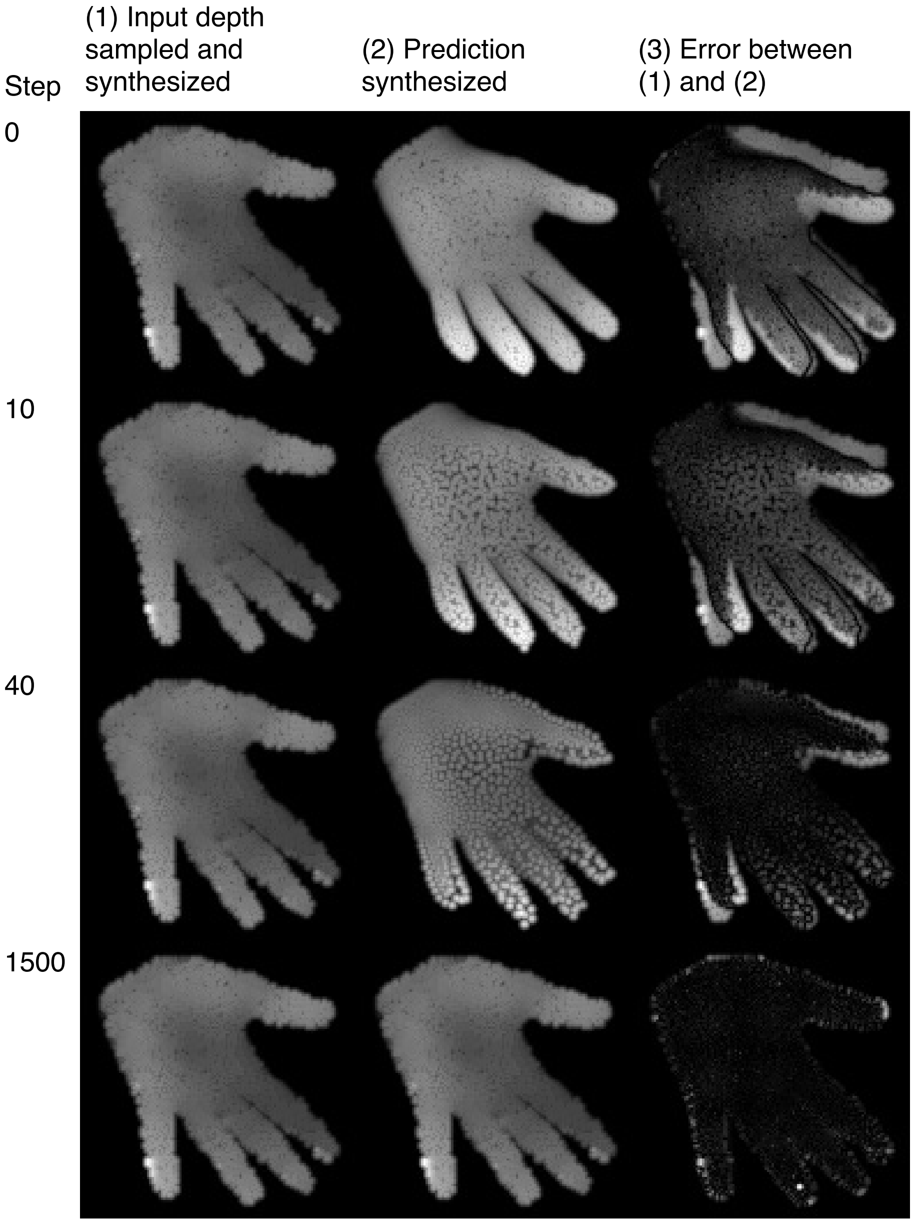

(1) Input depth sampled and synthesized
(2) Prediction synthesized
(3) Error between (1) and (2)
Step
0
10
40
1500

Figure 5.5: *In a second experiment, we optimize for the hand point cloud P only and demonstrate how the updates converge close to a global optimum.*

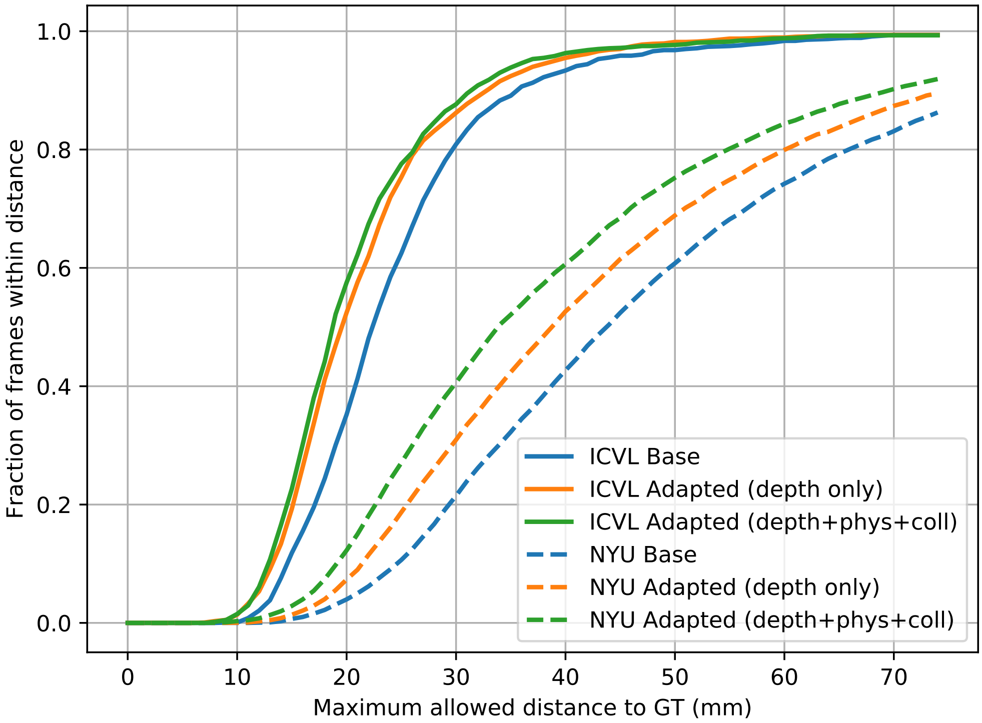


Figure 5.6: *ROC curves for the self comparison experiment on the NYU and ICVL dataset.*

illustrates that when learning from noisy data, we are more dependent on prior information, e.g. by enforcing non-self-intersection and physical constraints. The images of ICVL, however, are mostly of higher quality, allowing to infer this information already from the depth image.
In Figure 5.7, we show a **qualitative** self comparison on two random pose predictions from our dataset, before and after training. A more accurate 3D pose of the real hand is observed in the latter case. To give quantitative evidence, we train our model on our own training data for 10 epochs and show in Figure 5.8, that the model generalizes well to the validation set. We also notice that the variance drops from 0.0035 to 0.0026, indicating a more stable estimation as also backed up by visual inspection, where the jitter in video sequences gets reduced.

5.2.5 Comparison to State-of-the-art

Despite the fact that our target is to refine an initial CNN based model and adapt it to unlabeled depth data, we also compare to state-of-the-art meth-

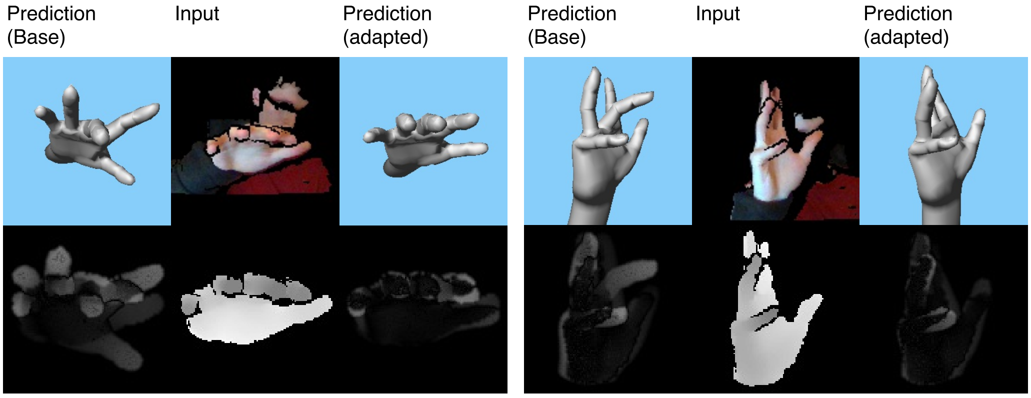


Figure 5.7: *Two qualitative examples from our validation set are shown, after training with* 50*k images from our training set without annotations. For each block, top row shows a 3D rendering of our hand model in the predicted pose before (left) and after training (right). For visual comparison, we demonstrate the RGB input image (center), which is not utilized by this method. The bottom row shows the absolute depth errors of the poses from above (left and right) (see Figure 5.2 for details) and the input depth image (center).*

ods that attempt to estimate the 3D pose from a single depth image on standardized datasets.

To start, we give some more context about the methods we compare to and claim that a direct comparison is quite difficult. We compare to REN [Guo et al., 2017], DeepPrior [Oberweger et al., 2015a] and DeepModel [Zhou et al., 2016] on the NYU and ICVL dataset. Additionally we compare to Feedback [Oberweger et al., 2015b] and LRF [Danhang Tang, 2014] on the NYU and ICVL dataset respectively.

All these methods utilize the ground truth annotations of the training data to refine their models, whereas we deliberately do not make use of them. Furthermore, except for [Oberweger et al., 2015a], the rest of the methods attempt to minimize the difference between the given annotations and their predictions. The feedback loop, proposed by [Oberweger et al., 2015a], minimizes a loss based on the depth images, as we do. Instead of a point transformation and rendering architecture though, they synthesize depth images (given a pose) via a generative CNN. This has the advantage of not needing an explicit model, as we do, but on the other hand they show a high dependency on good annotations, whereas we are completely independent of them.

Since we do not optimize for joint positions and use a different model than the ones that have been used for creating the annotations in the NYU and ICVL dataset, we observed an error between our joint position prediction

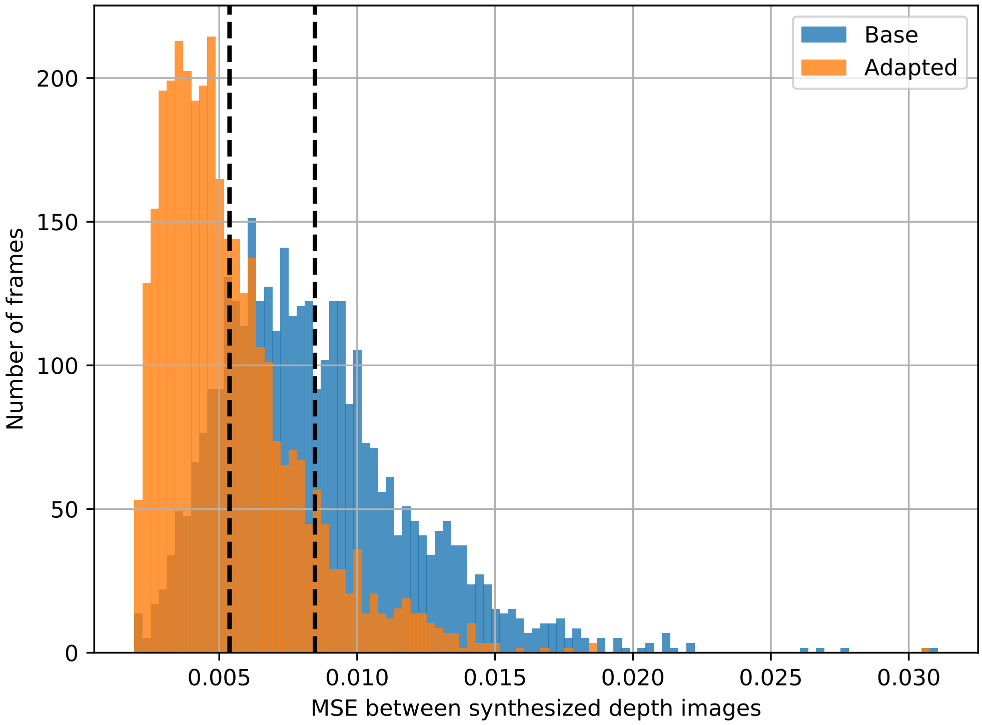


Figure 5.8: *Distribution of the MSE between the synthesized depth images (see (3) and (5) in Fig 5.2) over the frames of our validation set, before and after training on our training set. The vertical lines show the mean values before (right) and after prediction (left).*

and the ground truth, even for accurate predictions, as evaluated by the depth image differences. Therefore, a bias can be assumed in our joint position prediction. We estimate this bias as the minimum error for each joint, over the whole training set, between our joint position predictions and the ground truth. The evaluations on the validation sets are plotted in Figure 5.9, where a second curve for our method is added, showing the error of the same prediction with the bias subtracted. Since this bias might be too optimistic, we believe that our real joint prediction error should be somewhere between the two curves, showing that our method compares closely to several state-of-the-art methods on both datasets.

5.3 3D Hand Pose Estimation from RGB Images

3D hand pose estimation from monocular RGB images and video is more challenging than its depth-based counterpart and it has only recently been

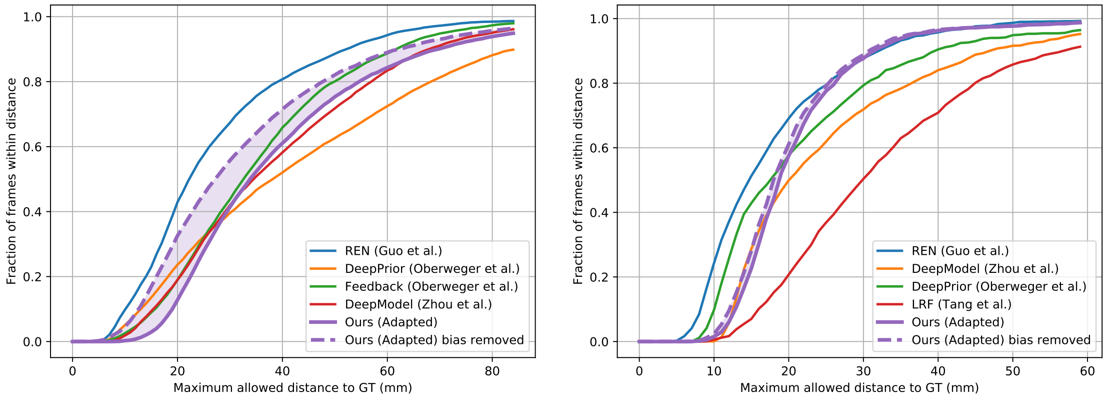


Figure 5.9: *Comparison to state-of-the-art methods on the NYU and ICVL dataset. For one of our curves (dashed), we remove the bias introduced from the mismatch between our hand model and the ones used as groundtruths for each dataset.*

explored [Zimmermann and Brox, 2017; Panteleris et al., 2017; Mueller et al., 2017; Spurr et al., 2018]. We need new network architectures, and new real ground truth (GT) datasets to tackle this highly ambiguous problem. While the former is easier to achieve and also compare to, unfortunately on very limited monocular datasets captured [Zhang et al., 2016], the latter is quite hard to obtain, and based on the hunger of CNN-s for real data, it seems to also explain the bottleneck behind limited accuracy of various architectures on such monocular RGB based tasks, as opposed to their depth counterparts.

In this section, we propose an extension of the previously introduced architecture. adapted to the RGB case and a high quality dataset to improve the accuracy of 3D hand pose estimation from a single RGB image. Our squeezenet [Iandola et al., 2016b] based architecture attempts to map a single RGB hand image directly to a 3D hand representation. We continue to utilize angle differences from a reference neutral pose, as in the Section 5.1, without the necessity to lift from 2D to 3D as in previous works [Zimmermann and Brox, 2017; Zhao et al., 2016; Tompson et al., 2014]. The network is trained on our new, large, realistically rendered hand dataset, consisting of around 3 Million RGB images with respective 3D annotations. By construction, such a model allows to refine itself on real-data in a semi-supervised fashion, showing improved performance on gesture classification tasks, which we demonstrate in Section 5.4.

A crucial part of our technique is refining the network in an unsupervised way on real unseen monocular data, given that a depth image is provided or extracted, by leveraging the technique presented in Section 5.1.3. We demon-

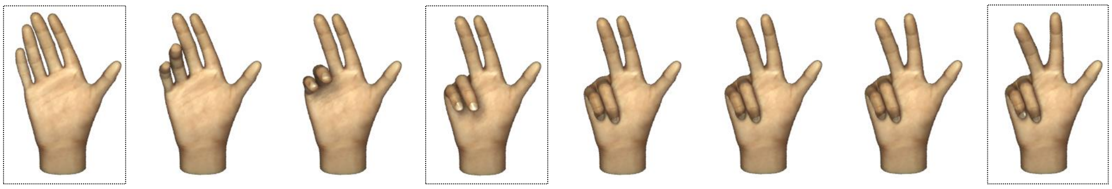


Figure 5.10: *Three base poses (in boxes) with linear interpolation on the parameter space in between.*

strate through various experiments that we can obtain a performance boost as compared to training with purely synthetic or limited monocular ground truth data, unlocking further applications that work with RGB monocular data. When compared to previous works based on monocular RGB images, an increased performance can be observed for a variety of tasks (3D pose estimation, hand gesture recognition and 2D fingertip detection), while being on par with methods that require depth as input. Below, we initially introduce the RGB synthetic dataset and then we delve into the method which utilizes it.

5.3.1 Synthetic Dataset Generation

In the absence of monocular RGB labeled datasets, in order to capture the space of pose variability already at training time, we create a new, large, realistically rendered, available free-hand dataset.

Hand Model. We opted for a commercial rigged and textured hand model[2] for Maya®[3]. The skeleton consists of 21 bones with 51 degrees of freedom (DoF), see Figure 5.13 (Left). Since not all the DoF are feasible for a human skeleton, we restrict our method to 4 DoF per finger and 3 for the rotation of the wrist. A real human hand has more than these 23 DoF [Lee and Kunii, 1995], however, the additional DoF are often ignored to simplify the problem [Tagliasacchi et al., 2015].

Synthetic Dataset. Inspired by [Xu et al., 2016], we decided to use a combination of manual and automatic sampling. We first create some base poses. Then we linearly interpolate over the parameters between each pair of base poses to generate new poses, as in Figure 5.10, detecting intersections. This procedure allows to easily adapt the dataset to a desired purpose by crafting suitable base poses and then automatically generating the linear span

[2] https://www.turbosquid.com/3d-models/rigged-male-hand-max/786338

[3] www.autodesk.com/products/maya

between them, as explained below. We end up with 399 such poses. In addition to the varying poses, for each view (we consider 5 views - front, back, both sides and top, Figure 5.13 (Middle)) we apply 5 random rotations (45 degrees for each DoF of the wrist joint) and illumination changes to each image. We also vary the texture and shape.

Base Poses. We enumerate the thumb poses separately and firstly focus on the other four fingers. We fix three possible opening states of a finger: fully open, partially closed (metacarpophalangeal joint still stretched) and fully closed. Furthermore, we assume that if some fingers are not fully stretched, they are closed the same way (either all of them partially closed or fully closed). The side-movements are combined with the opening state of a finger, e.g. they are ignored when the fingers are partially or fully closed, due to the human hand limitations [Lee and Kunii, 1995]. A further simplifying assumption we made was to enumerate side-movements by counting the gaps between the fingers as explained below:

- If all four fingers are fully open, there are three gaps in between. Each gap can be either open or closed, giving a total of $2^3 = 8$ combinations.
- If three fingers are fully open, the number of gaps depends on the (partially) closed finger, being either two or one. Since the closed finger can be either fully or partially closed, we get a total of $2(2^2 + 2^1 + 2^1 + 2^2) = 24$ poses.
- If two fingers are fully open, we get a total of $2(2^1 + 2^0 + 2^1 + 2^0 + 2^0 + 2^1) = 18$ poses with the same considerations as in the previous case.
- If only one finger is fully open, we do not have side-movements with our assumptions, giving a total of $2(2^0 + 2^0 + 2^0 + 2^0) = 8$ poses.
- If all fingers are closed, they can be either partially of fully closed, giving a total of 2 poses.

Having in mind the interpolation to be performed, we fixed six different poses of the thumb, giving a total of $6 \times 60 = 360$ base poses. Additionally, we added four poses where the thumb touches one of the remaining fingers each, since the above setting of separating the thumb from the remaining fingers does not handle this. We also added 6 poses with crossed fingers, because the focus on the gap between the fingers does not capture this. In the end, 29 base poses were created inspired by the *HGR* [Kawulok et al., 2014; Nalepa and Kawulok, 2014; Grzejszczak et al., 2016] dataset, giving a total of 399 base poses.

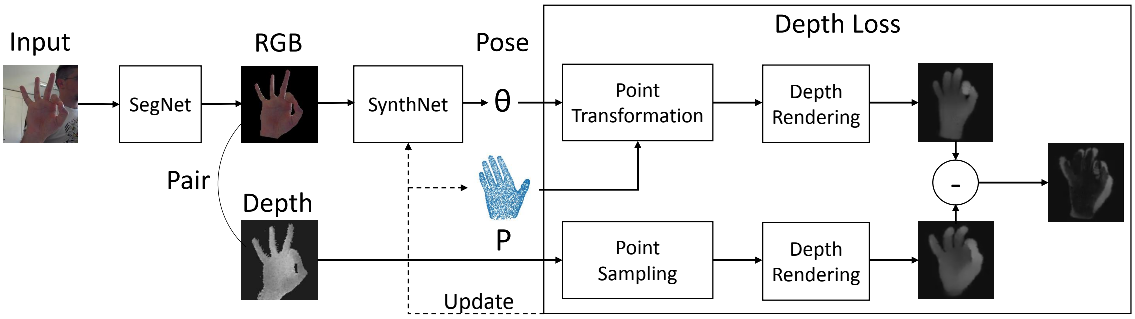


Figure 5.11: *Overview of the extended training pipeline for the RGB input case. Given a monocular RGB image as input, a SegNet based network first segments out the background, the result of which is input into SynthNet, a CNN model trained purely on synthetic data (Section 5.3.1) that predicts the hand pose in terms of angles θ. In order to fine-tune the network to real monocular data, provided that a corresponding depth image is given, we augment the initial base network with a depth loss component. We refer to this combination during training time as RefNet. Given θ as well as P, a point cloud that initially represents our hand model and gets iteratively updated to the input one, the weights of SynthNet can be updated without the need of labeled data. At test time, a forward pass through SegNet and SynthNet estimates the desired pose.*

Collision Avoidance. Since a linear interpolation within the hand pose space can lead to self-intersection, the automatic generation of new poses contains an intersection detection which rejects such undesired poses. In order to detect intersections, we loop over all finger vertices to find the nearest (other) finger neighbor. By projecting the vertices difference vector onto the other finger surface normal, it can be computed whether the vertex is inside the foreign mesh or not. An intersection occurrence is detected when an "inside" threshold is passed. In order to simulate flesh interaction between fingers, we relax the threshold allowing very little intersection. Due to interpolation with collision avoidance we end up with 122106 different poses.

Un-natural Poses. The linear interpolation preserves many constraints applied to the base poses, e.g. maximal angle-range and fixed ratio between certain angles. Thus, it suffices to create the base poses with the desired constraints to make sure that the same holds within the complete dataset.

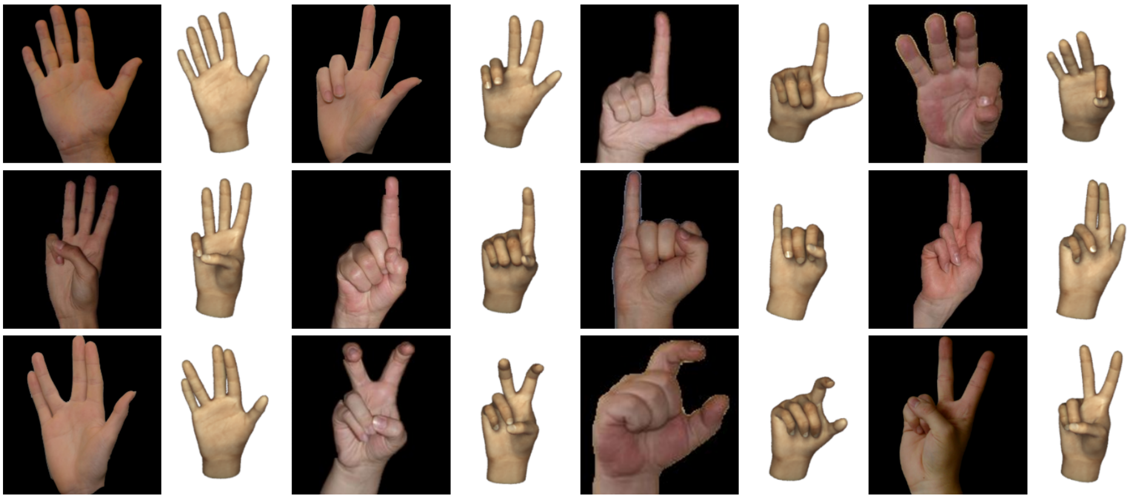


Figure 5.12: *Real predictions on the HGR dataset.*

5.3.2 Method Overview

The overview of our method is depicted in Figure 5.11. We attempt to achieve two main goals: 1. estimate the 3D hand pose, given a single monocular RGB image, and 2. enable a refinement of our method predictions on unseen real images in an unsupervised way. Due to the lack of real RGB ground truth datasets, we tackle the first goal, by training a CNN (*SynthNet* Section 5.3.2.2) that minimizes an angle loss ($\mathcal{L}^{angle}$) in a supervised manner. We train purely on our newly presented (Section 5.3.1) large synthetic dataset, consisting of masked-out renderings of hands in various poses, shapes, illuminations and textures and their respective 3D annotations. At test time, we first segment a raw RGB image in order to obtain only the hand part, by passing it through a segmentation CNN (Section 5.3.2.1), trained on a combination of real and our own synthetic data to minimize a categorical cross-entropy loss ($\mathcal{L}^{mask}$). This first part captures priors on the variability of possible free hand poses already at training time and achieves results on-par or even better than state-of-the-art works on real datasets for a variety of tasks (Section 5.4).

We tackle the second goal of real data based refinement, by extending our *SynthNet* with a component based on a depth loss ($\mathcal{L}^{depth}$), as in Section 5.1.3, which allows it to get fine-tuned on unseen unlabeled real RGB data, provided that an analogue unlabeled depth image (registered or unregistered), is present at training time. We refer to this combination during training time as $RefNet$, which can be considered as a differentiable renderer. The weights of *SynthNet* are adapted to real data in an unsupervised

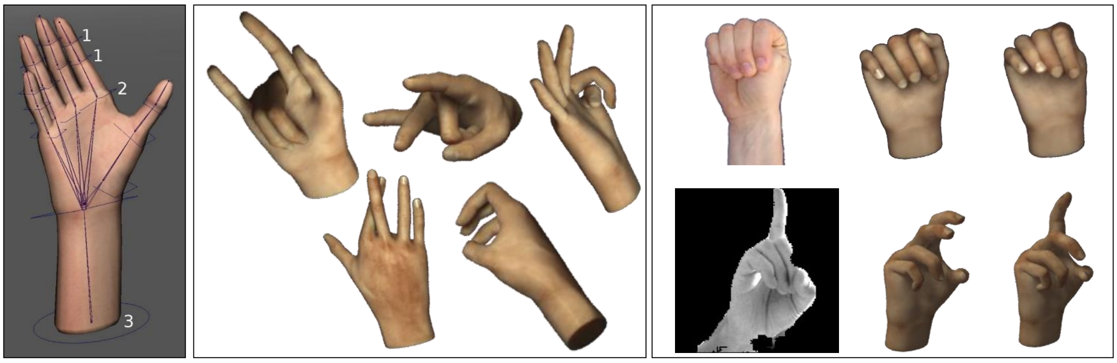


Figure 5.13: *(Left) Rigged hand model with max* 51 *DOF (Middle) 5 samples from our dataset in 5 different orientations (Right) Two semi-supervised refinement examples from our own dataset (top) and Senz3D (bottom) - from left to right: input, SynthNet unrefined and refined prediction.*

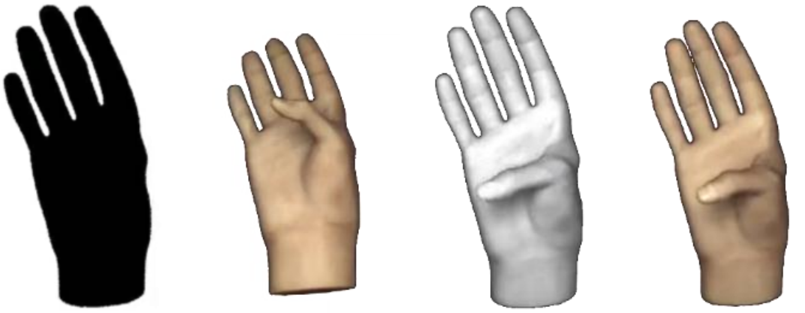


Figure 5.14: *Predictions with a CNN trained on silhouette (left) and RGB (converted to grayscale, right) input respectively.*

manner. During test time, a forward pass through it allows to estimate the 3D pose. This second part is very important, because of the known discrepancy between real and synthetic data due to different hand shapes, poses, sensors, and environment conditions, which is even more enhanced when the expected input is an RGB image. This refinement leads to significant improvements over the network trained purely on synthetic data, which we show through experiments in Section 3.5.5.

5.3.2.1 Hand Segmentation Nets

It might be disputable whether in applications such as object shape or pose inference from a 2D image, splitting the task into an object segmentation first, and then an inference from such a segmentation, is the optimal approach. However, decoupling segmentation from inference splits the problem into two easier to solve sub-parts, each requiring less training data, and above all simpler and more lightweight CNN-s, reducing computational costs. As explained in Chapter 3, learning based works show great promise in the human body shape and pose estimation from silhouette tasks, however this certainly does not suffice, due to the inherent silhouette ambiguity and the more pronounced self-occlusions that a hand has, as compared to human body. In Chapter 4, we additionally showed that garment shape estimation is feasible, provided masked-out realistic synthetic renderings of garments, as opposed to just silhouettes. We provide further attempts to prove this point in a synthetic experiment, depicted in Figure 5.14, where the inside pointing thumb is better estimated in the latter case.

Before segmentation, the hand needs to be localized in the image . This is not the focus of this work but a necessary pre-processing step. Many recent works tackle the task of hand detection and segmentation with neural networks [Vodopivec et al., 2016; Bambach et al., 2015; Sharp et al., 2015]. Inspired by [He et al., 2017] that compute object detection and segmentation operating in two stages with Faster R-CNN [Ren et al., 2015], we adopted SegNet [Badrinarayanan et al., 2015] to first propose the hand region and then compute a pixel-wise mask of the hand. The detection is also performed via segmentation, producing a rough mask to localize the hand and crop around it, which in turn is utilized to produce a more refined hand mask. In order to decrease training and inference time, without affecting accuracy, we removed some layers from both the encoder (two convolutions and one max-pooling) and decoder (8 convolutions and one up-sampling). We call this architecture *OurSegNet* and provide details Section 5.4.1. Segmentation is a necessary preprocessing step of our pipeline, and not a contribution of this work, hence we analyze both it‘s performance and that of *HandSegNet* from [Zimmermann and Brox, 2017] in Section 5.4. The expected input RGB image and segmented output are 256 $\times$ 256 pixels each. The latter serves as input for the next stage.

5.3.2.2 Synthetic RGB CNN Model (SynthNet)

Inspired by the SqueezeNet [Iandola et al., 2016b] adaptation in Chapter 4, which is trained purely on realistically rendered masked-out synthetic garment images to map directly to 3D garment vertex meshes, here, we also pose our problem as finding a mapping from masked-out images of hands to the 3D hand pose, through a SqueezeNet adaptation. For us, the most important insight from the previous chapter, is that despite the fact that high frequencies (wrinkles) are hard to capture from the data, due to the difficulty of representing various cloth fabrics in a synthetic dataset, low frequency

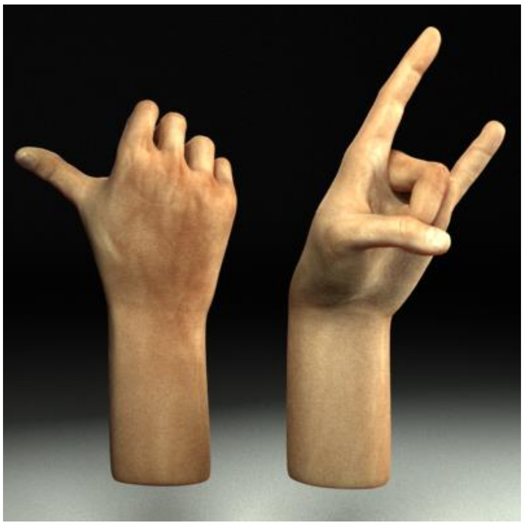


Figure 5.15: *Realistically rendered hand model.*

features, such as shape and especially pose of the garment are captured quite accurately from real garment images. Since hands can be assumed to be of the same "fabric", exhibiting similar skin properties, it becomes even more feasible to render a realistic synthetic hand dataset (Section 5.3.1), e.g. as in Figure 5.15.

We start by training a SqueezeNet model (*SynthNet*) adapted to regression, purely on our synthetically generated dataset (Section 5.3.1), which directly predicts, as in [Zhou et al., 2016; Mehta et al., 2016], a 3D pose θ from a (masked-out) RGB image I (Section 3.5.5). Our 3D pose θ is represented in euler angles, similar to [Zhou et al., 2016], however quaternions or rotation matrices can be utilized too, with the constraint, as in Section 5.1.2, that θ must be informative enough to calculate a forward kinematic chain, yielding the exact information on how each joint transforms to the predicted pose (Section 5.3.2.3). This is made possible by our rigged hand model (Figure 5.13 (Left)). More specifically, θ is given as an angle difference for each of the hand joints from the joint angles of a hand in a neutral pose (open palm). Given the RGB images of the synthetic training data, we train our SynthNet from scratch to minimize the mean squared error ($\mathcal{L}^{angle}$) between the pose from our dataset and the predicted pose. We noticed that by first converting the input images to grayscale and then applying histogram normalization, with one and 99 percentile as borders to remove pixel outliers, not only made the network converge faster, but also helped with skin-color invariance. Since during training, all the hand masks are centered, at test time, we also center and scale the hands to a square image of 225×225 pixels (similar to the SqueezeNet input), when necessary padded at the borders.

5.3.2.3 Semi- and Unsupervised Refinement from RGB and Depth Images

Semi-Supervised Refinement on Real RGB Images. One advantage of utilizing angles instead of joint positions, is that they can be easily be restricted to the allowed Degrees of Freedom, reducing the large space of infeasible poses, and constraining the latent space [Choi et al., 2017]. Given a skeleton, angles can easily be converted to joints and hence fully determine a pose. This might penalize accuracy on exact 3D joint estimation tasks, under fixed hand skeleton model assumptions, however it can be quite attractive for other tasks where the hand skeleton constellation is more important than the exact joint position, e.g. hand gesture recognition/classification. Another advantage of utilizing angles, is that it allows any pre-trained fully supervised network (regardless whether real or synthetic data is used), to refine itself on easily obtainable real unlabeled RGB images. Real images of hands in various shapes, skin colors, lighting conditions and rotations can be easily captured with cheap RGB sensors, under the constraint that users perform pre-specified gestures, as in [Memo et al., 2015; Memo and Zanuttigh, 2017]. These gestures can be easily modeled, given a synthetic hand model, obtaining the ground truth (angles) without additional manual effort. Angles are advantageous here, as various user poses would map to the same ground-truth, regardless of the exact hand position and rotation in the image. In this way, the input space is enriched with multiple real images that map to the same angles, which in turn helps to fine-tune synthetic networks and improve the gesture recognition predictions. Details on this are provided in Figure 5.13 with the respective discussion in Section 5.4.4.

Unsupervised Refinement with Depth Images. *SynthNet* alone gives good initial predictions on various real data ((Section 5.4), Figure 5.12 and Figure 5.23), however a discrepancy between synthetic and real datasets is known in literature and practice. Following the method presented in Section 5.1.3, we extend our network with a component that enables *SynthNet* to get refined unsupervised, trained to minimize a depth loss ($\mathcal{L}^{depth}$) on unlabeled depth data, that have one-to-one correspondences to the input real RGB images. Unlike, the above we do not make use of the Physical and Collision component. Let's assume we have pairs of RGB and Depth images (I, D). Acquisition of such pairs is very cheap with today's RGBD sensors (Section 5.4.2). We compare the input depth image D to a synthesized depth image D_I, which is computed from *SynthNet* predicted pose θ, given I as input, and a pointcloud P sampled from the hand mesh model, in order to predict the accuracy on unlabeled data. We transform the P points according to θ, applying Linear Blend Skinning (LBS) [Lewis et al., 2000], and subsequently render them to obtain a synthetic depth image.

5.4 Experiments and Results

In Figure 5.16, we demonstrate qualitative results of poses inferred from *SynthNet*, with input images from the *HGR* dataset [Kawulok et al., 2014; Nalepa and Kawulok, 2014; Grzejszczak et al., 2016] and one additional individual performing various poses in Figure 5.17.

5.4.1 Training Details and Architectures

The architectures for the hand detection, segmentation and pose inference model are illustrated in Figure 5.18, Figure 5.19 and Figure 5.20 respectively in more detail. We train *HandSegNet* with Adam optimizer [Kingma and Ba, 2014] and an initial learning rate of $10^{-}5$ with decay $5 \times 10^{-}4$, changed to $10^{-}6$ and $10^{-}7$ over 10 epochs, and *OurSegNet* with Stochastic Gradient Descent and the same initial learning rate for 30 epochs. *SynthNet* was trained utilizing Adam optimizer, with an initial learning rate of $10^{-}4$ over 10 epochs with a batch size of 100, while *RefNet* with Adam optimizer and learning rate of $4 \times 10^{-}3$ for over 40 epochs with a batch size of 1000. As input we expect a batch of 120×120 pixels depth images, with the depth values scaled in a $[0, 1]$ range.

Heat Map Visualizations for SynthNet. In order to investigate further the network's learning capacity, we visualize features learned by our network, by relating image positions to error contributions. We adopt a technique introduced by [Zeiler and Fergus, 2014] to regression. A black box is moved over different poses. For each position, the increase of error compared to the original image is measured and finally visualized as a heatmap. We calculate the mean squared error over different sets of parameters. Figure 5.21 shows heat-map examples for a variety of poses. As it can be noticed, most of the fingers demonstrate a high error throughout the whole finger when that part is missing, as we would expect.

Method Speed. We conducted our experiments on an Intel i7 860 (from 2010), 8 GB of RAM with an Nvidia Geforce GTX 1060. A forward pass through the network for predicting a single input image takes 3.5 ms, which is practically real-time. Training unsupervised for 10 epochs, as we did, takes around 30 minutes.

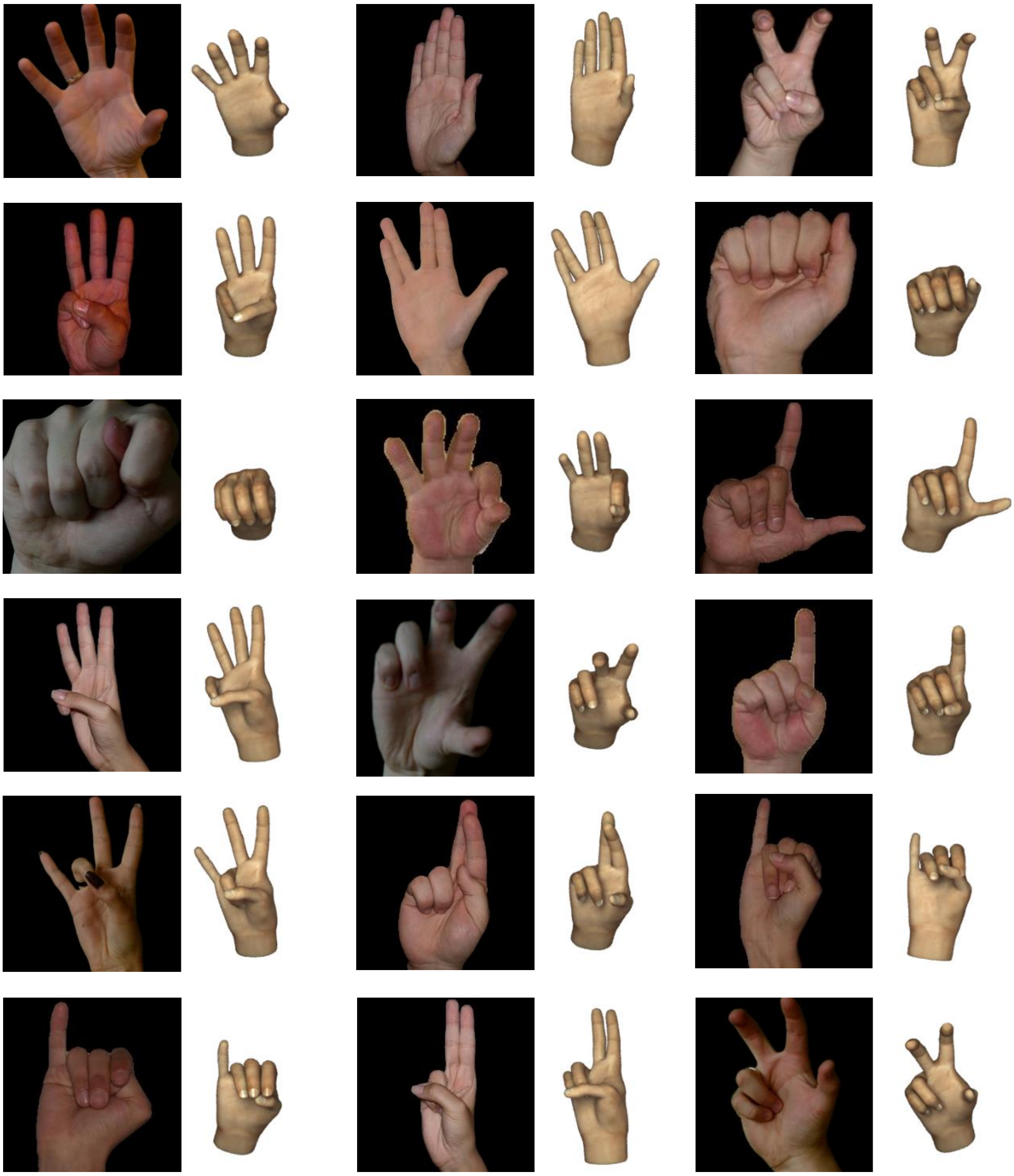


Figure 5.16: *Qualitative results on various hand poses, shapes and color.*

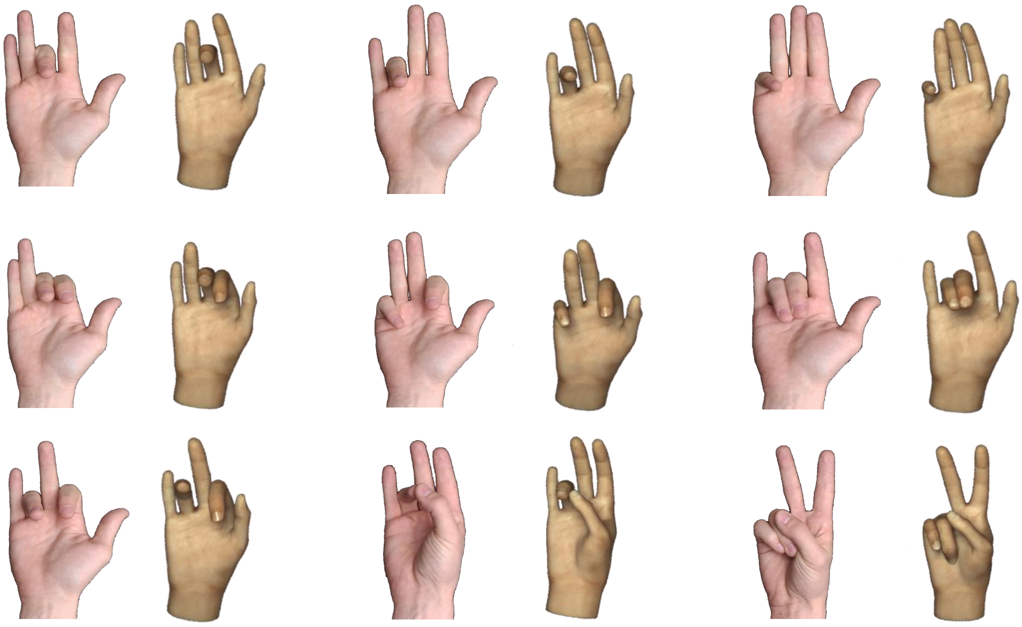


Figure 5.17: *Qualitative results on one individual in various hand poses.*

5.4.2 Training and Test Datasets

Detection and Segmentation Datasets. We utilize the method and dataset from [Zimmermann and Brox, 2017], for hand bounding box detection. On the other hand, for segmentation we use both real and synthetic data. The real hand dataset contains 19000 images, 6000 of which come from the Hand Gesture Recognition (HGR) dataset [Kawulok et al., 2014; Nalepa and Kawulok, 2014; Grzejszczak et al., 2016], which is an augmentation of the initial 1500 raw images (consisting of 33 individuals and 70 gestures), that we segment, add various backgrounds and perform in-plane rotations of the hand. The remaining 13000 belong to three individuals, captured performing various poses in front of a green screen, which is replaced with a random background. The synthetic images are in the 100*K* range and come from our synthetic dataset.

Pose Inference Datasets. Many publicly available datasets are shot with depth cameras, e.g. the recently introduced BigHand2.2M Dataset [Yuan et al., 2017b]. There is a lack of proper RGB datasets. The NYU Hand Pose Dataset [Tompson et al., 2014] e.g. contains holes in the RGB images if no depth data is available, while the Dexter RGBD dataset [Sridhar et al., 2016] has incomplete hand annotation (fingertips) [Zimmermann and Brox, 2017]. We make use of the Stereo Hand Tracking Dataset [Zhang et

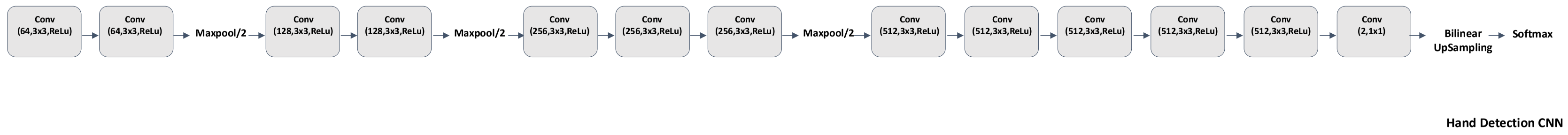


Figure 5.18: *CNN architecture of the detection part.*

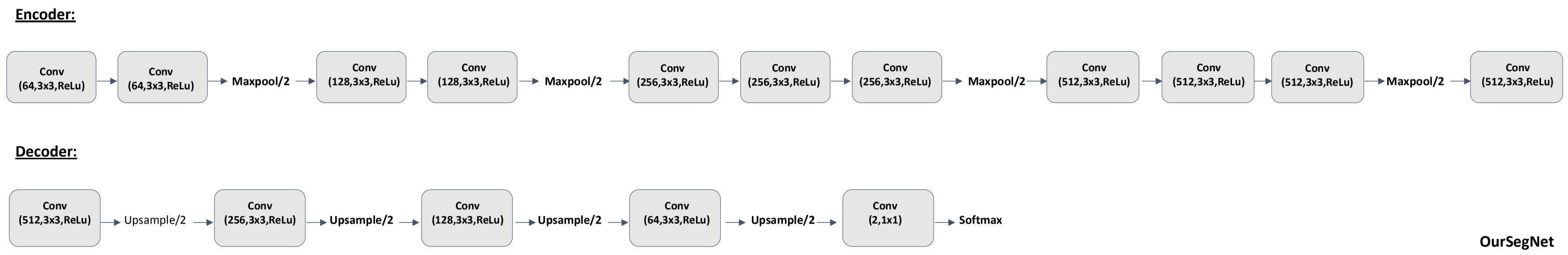


Figure 5.19: *CNN architecture of OurSegNet.*

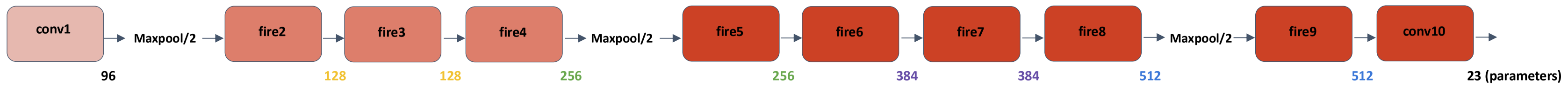


Figure 5.20: *CNN architecture for pose inference SynthNet.*

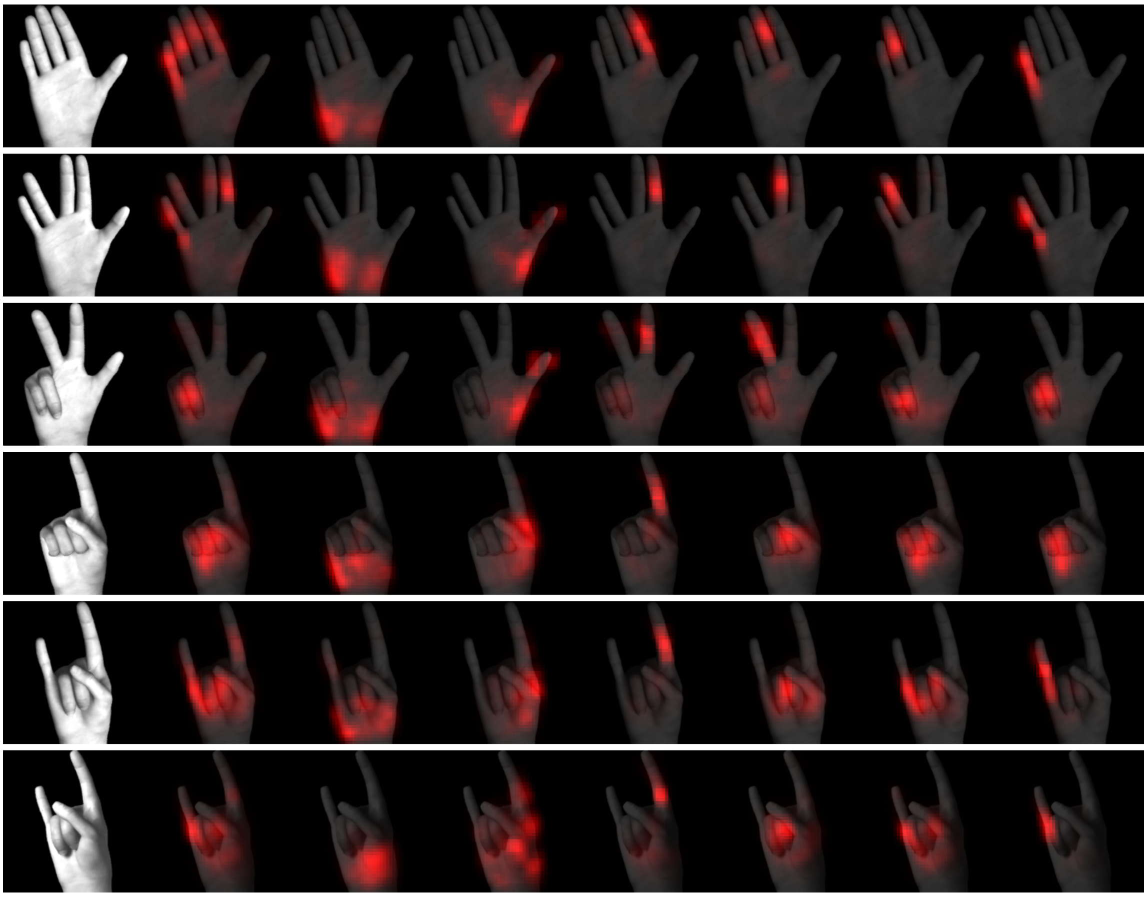


Figure 5.21: *Error Contribution Heatmap. Shown are six images from our dataset. For each input image, we visualize the contribution to the mean squared error of the complete pose, arm (wrist), thumb, pointer, middle, index and pinky (from left to right).*

al., 2016] (StereoDS), which contains twelve motion sequences in front of various backgrounds (B1 through B6, and for each set, a count and random sequence of 1500 images each), which provides RGB and Depth images together with the 3D joint positions. Another area having a rich variety of RGB datasets is hand gesture recognition, where the ground truth is a class label. We utilize the German Fingerspelling Database (RWTH) [Dreuw et al., 2006], that provides the classes of 35 gestures from the German sign language, for 20 people, HGR [Kawulok et al., 2014; Nalepa and Kawulok, 2014; Grzejszczak et al., 2016], which in addition to the class provides visible 2D fingertip locations and Senz3D [Memo et al., 2015; Memo and Zanuttigh, 2017], containing 11 gestures performed by 4 different people repeated 30 times each. Additionally, to demonstrate unsupervised refinement on real data, we capture our own dataset (*IntelDS*) utilizing the Intel RealSense Camera. It consists of 1000 pairs of registered RGB and depth images for

Dataset	*HandSegNet*	*HandSegNet+Synth*	*OSN*	*OSN+Synth*
B1 Random	91.5	97.7	91	95.5
B1 Count	92	98	92	96
RWTH	93.34	93.37	92.9	93,1

Table 5.2: *Segmentation accuracy in % for HandSegNet and OurSegNet (OSN) trained with and without our synthetic dataset.*

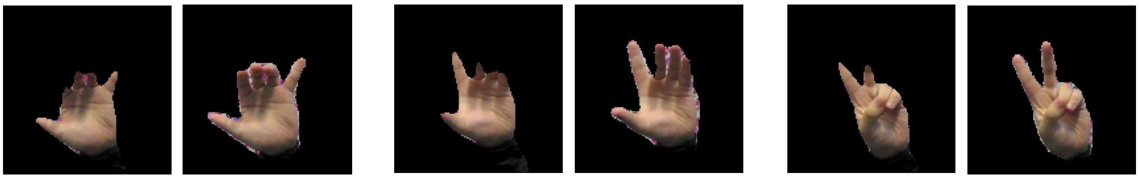


Figure 5.22: *Three examples of segmentation improvement on StereoDS before (left) and after (right) adding our synthetic training data.*

testing and 30,000 for training (in the size of 120 × 120 pixels and without GT annotations), from one individual wearing a black wristband, that allows for a simple intensity based segmentation.

5.4.3 Segmentation Accuracy Improvement

We evaluate the segmentation accuracy for both *HandSegNet* [Zimmermann and Brox, 2017] and *OurSegNet*, when training is performed with and without adding our synthetic dataset to the available real ones. We evaluate on B1 random and count (150 images each) of *StereoDS* and the complete *RWTH*, observing an accuracy increase in the latter case (Table 5.2). Figure 5.22 depicts visual results on three examples where the synthetic data helps in segmenting complete fingers.

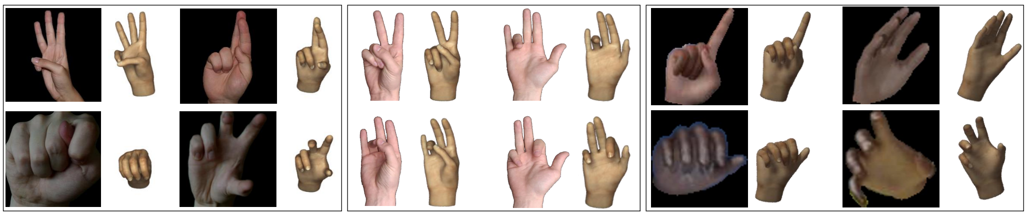


Figure 5.23: *SynthNet predictions on (left) HGR dataset (middle) one individual hand (right) synthetic dataset from [Zimmermann and Brox, 2017].*

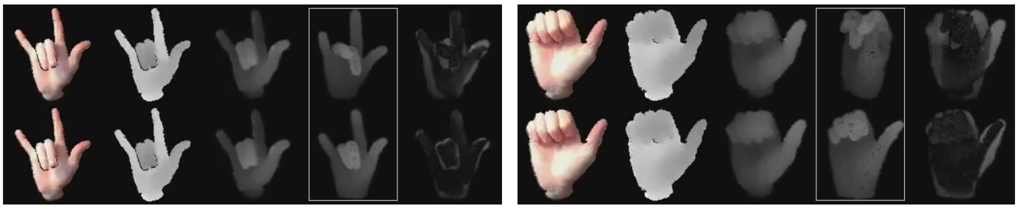


Figure 5.24: *Two examples from our validation set $IntelDS$. SynthNet predictions before (top) and after refinement (bottom). From left to right : RGB Input (I), Input Depth (D) , Synthesized Input Depth (D_S), Prediction (D_I) and Error in depth prediction.*

5.4.4 Refinement with Unlabeled Data

Semi-Supervised on Real RGB Images. As a proof-of-concept, we utilize the *Senz3D* dataset [Memo et al., 2015; Memo and Zanuttigh, 2017], to fine-tune our *SynthNet* on real RGB images, by splitting the dataset in half (300 each) for training and testing for a gesture classification task on 10 of the classes. We first manually craft a synthetic pose for each of the classes, in order to obtain approximate GT labels (angles) for each training image. Then, we learn a mapping from angles to classes, similar to [Zimmermann and Brox, 2017]. We measure the accuracy utilizing a 10-fold cross validation, and notice an increase from 94 to 96.7%, which is enabled by representing the 3D pose in terms of angles as opposed to 3D joints (Section 5.3.2.2). Figure 5.13 (Right) visualizes this improvement for samples extracted from two datasets..

Unsupervised on Pairs of RGB and Depth Images. We utilize the $IntelDS$ to refine our $RefNet$ in an unsupervised way, utilizing pairs of RGB and depth data, and compare it to the results of *SynthNet* before refinement. We visualize the results before and after refinement in Figure 5.24, also through ROC curves in Figure 5.25 and Figure 5.26, demonstrating a clear improvement after the refinement. By computing MSE between the two synthesized images which are utilized to compute the depth loss, we notice that the error halves in the latter case. Applying our method to videos, without smoothing or per-frame interpolation, not only enhances the pose prediction quality, but also removes jitter significantly.

5.4.5 Refinement and Intersection Handling

Our method has limitations too. Firstly, since it does not have an explicit intersection handler, at prediction time, some intersections occur. Secondly,

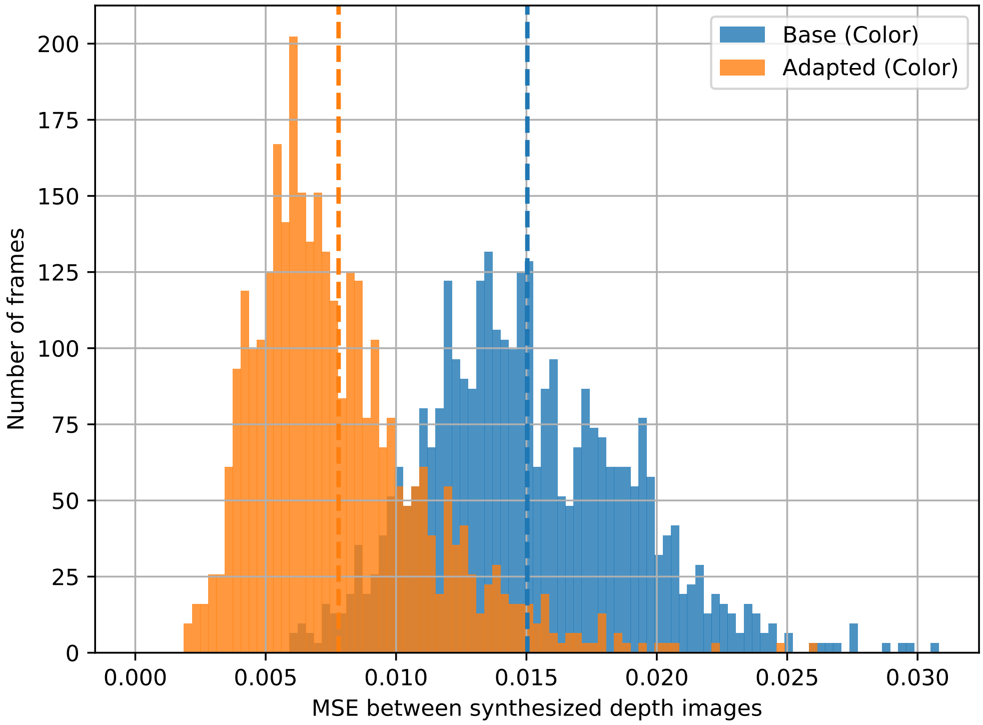


Figure 5.25: *MSE histogram computed over pixel (depth) image difference for the network before and after refining unsupervised.*

due to the thumb discrepancy between real and synthetic data, there is a difficulty of estimating closed fists (with the thumb in) directly from synthetic data, as well as poses like in Figure 5.16 (1*st* Column, 3*rd* row).

Kinematic Constraints at Run-time. Due to our database construction the cases where fingers intersect are minimized, however they exist. One way to tackle this could be to add an intersection handling term as part of the loss function. This however would help only at training time, during a potential unsupervised refinement. In order to make sure that no intersection happens during prediction, we, for completeness and comparison (Figure 5.27), provide a small extension of the pose prediction pipeline, by calculating a new pose ϕ through minimizing the energy function in equation 5.31. The first norm penalizes large deviations from the pose $\overline{\phi}$ predicted by the CNN. A cylindrical model is utilized to penalize intersections in the second term, where each finger consists of three cylinders. A cylinder p is determined by a radius r_p (obtained from our hand model) and a segment $s_{p1}^{p2}(\phi)$ serving as axis (computed from the rotation determined from ϕ). We denote with d the distance function between two segments.

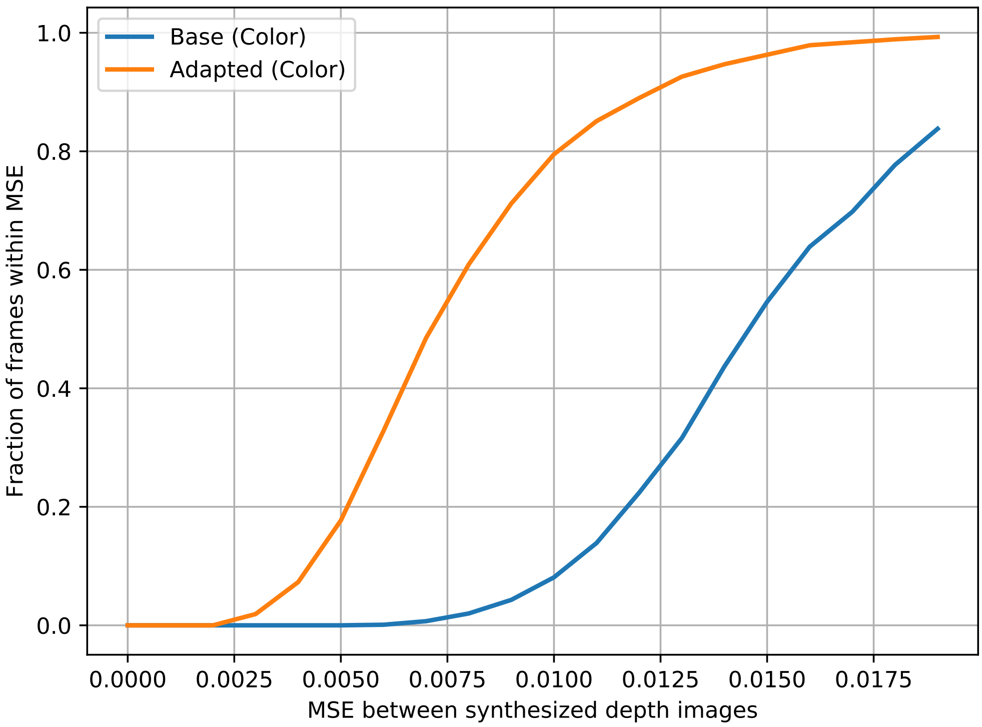


Figure 5.26: *ROC curve corresponding to Figure 5.25.*

$$E_1(\phi) = \phi - \overline{\phi}^2 + \lambda \left[\sum_{(p,q)\in I} max\{0, r_p + r_q - d(s_{p1}^{p2}, s_{q1}^{q2})\}^2 \right]^2 \qquad (5.31)$$

To solve the optimization problem imposed by equation 5.31, we utilize the Ceres Solver [Agarwal et al.,]. Figure 5.28 shows qualitative examples of refinements with $\lambda = 10$. The additional step helps to correct small failures. A bit of intersection is still allowed, in order to simulate flesh interaction with our model.

5.4.6 Comparison to State-of-the-art

We compare to related methods working on RGB or depth input images, and investigate generalization on various dataset, for three main tasks : gesture recognition, 2D fingertip estimation and 3D pose estimation. Qualitative results on predictions are depicted in Figure 5.12, Figure 5.16, Figure 5.17 and Figure 5.23.

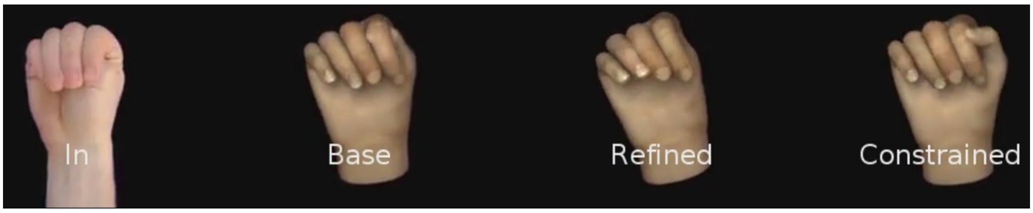


Figure 5.27: *From left to right : input image, SynthNet prediction, SynthNet refined semi-supervised. SynthNet with interpenetration constraint handling prediction.*

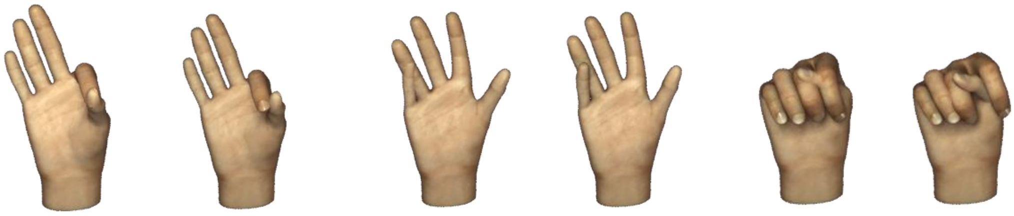


Figure 5.28: *Kinematic Constraints Demonstration. The left image of each pair shows an initial prediction, the right image a refined version using kinematic constraints.*

Classification on Spelling Dataset. Like [Zimmermann and Brox, 2017], we evaluate our system on *RWTH* on all the 30 static gestures, by first predicting the poses and then applying a pose classifier to the respective class. Unlike [Zimmermann and Brox, 2017], we do not utilize images from this dataset to refine on and we first segment the images utilizing *OurSegNet*. We utilize 10-fold cross validation to estimate the accuracy since no split specification was given by [Zimmermann and Brox, 2017]. Training was done with one hidden layer of 500 neurons with ReLu activation and dropout probability of 0.5. We achieve superior performance compared to [Zimmermann and Brox, 2017] and [Dreuw et al., 2006] as shown in Table 5.3. We repeat the same experiment, however now on *Senz3D* over 10 classes, also achieving a better performance than [Zimmermann and Brox, 2017].

Fingertip Detection Comparison. We evaluate *SynthNet* predictions on the

Method	*RWTH*	*Senz3D*
[Dreuw et al., 2006] on subset (from [Zimmermann and Brox, 2017])	63.44	-
[Zimmermann and Brox, 2017]	66.8	77
Ours	**73.6**	**94**

Table 5.3: *Classification accuracy comparison, in % of correctly classified poses, on the RWTH and Senz3D.*

Method	*Error*
[Zimmermann and Brox, 2017] (their segmentation)	804.23 px^2
[Zimmermann and Brox, 2017] (oracle segmentation)	483.28 px^2
Ours (oracle segmentation)	**361.47** px^2

Table 5.4: *Fingertip accuracy on the HGR Dataset computed as MSE over pixel errors, with image size* 225 × 225 *pixels.*

Evaluated for \Trained on	*Joint positions*	*Joint angles*
Joint Position MSE	**0.199**	0.397
Joint Angle MSE (deg)	42.829	**12.763**

Table 5.5: *Joint Angles vs Positions MSE on our synthetic dataset.*

HGR dataset, which contains hands from multiple people, assuming an oracle segmentation (ground truth segmented by us). Figure 5.12 and Figure 5.23 (left) shows a qualitative assessment of our results, where the predicted pose seems quite accurate, despite training only on synthetic data. To quantitatively compare to [Zimmermann and Brox, 2017], we measure the accuracy of predicting 2D (visible) joint positions, by computing the MSE on pixels for all front facing images (since back facing ones have almost no visible fingertip). [Zimmermann and Brox, 2017] provide 3D joints directly, while we apply the kinematic chain on angles θ to retrieve the 3D joints. These 3D fingertips are then projected into 2D, by solving a least-squares system to best fit to the groundtruth labels (since no camera info is given). Table 5.4 depicts these results, with [Zimmermann and Brox, 2017] evaluated with their and the oracle segmentation (since we train *OurSegNet* on *HGR* we only evaluate on oracle segmentation), where our method achieves higher accuracy.

ROC Angle and 3D Joint Curves. We evaluate accuracy on 3D pose prediction for different methods by computing ROC curves, that denote the fraction of frames below a maximum 3D joint (or angle) prediction error, on the B1 set of *StereoDS*. We compare to [Zimmermann and Brox, 2017], that assume an RGB input as we do, and four other depth-based methods. Such methods are trained to directly predict 3D joint positions, unlike ours that predicts angles (Section 5.3.2.2), and hence minimizes a different quantity (e.g. a slight wrist angle miss-calculation would bring a larger error on 3D joints prediction, even if the rest of the angles are correctly predicted). Thus, we argue that a direct comparison on this dataset is not possible, also due to the discrepancy between the GT skeleton in *StereoDS* and our hand model skeleton, from which we compute 3D joints from angles. In order to back this

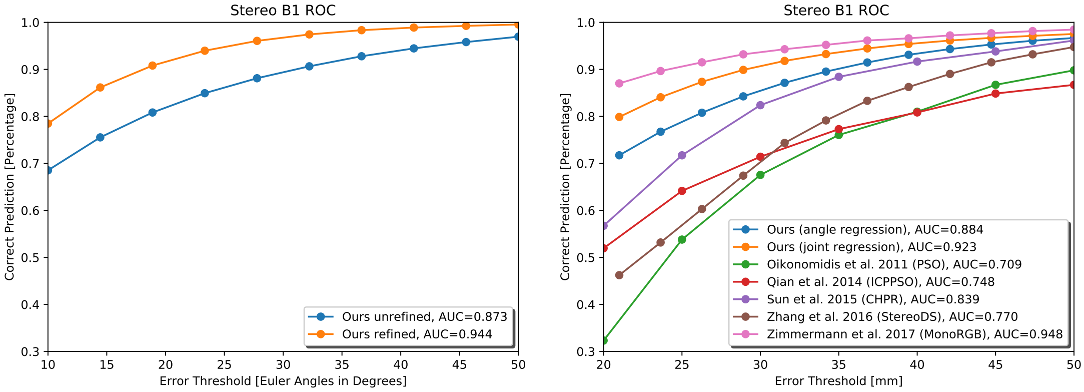


Figure 5.29: *Accuracy on the StereoDS dataset. (Left) Improvement in euler angles due to refinement (Right) Comparison to state-of-the-art methods trained to map onto 3D joints. We show our ROC curve trained on angles along with a version trained on joints.*

up, we performed an experiment, on 300 unseen samples from our synthetic dataset, where we once trained for 3D joint positions and once for angles, and computed the MSE for both cases. As it can be noticed in Table 5.5, training for the respective task always achieves a smaller error. Nevertheless, for completeness we compare on this dataset and report ROC curves for both angles and joints, in Figure 5.29. Due to the lack of GT segmentation we first apply *OurSegNet* to obtain the masked-out RGB images. The methods we compare to, refine on sets B2-B6 consisting of $15,000$ images. We can not directly fine-tune on such datasets unfortunately, however we apply the following procedure : we compute the GT angles over B3-B5 (note from a different skeleton) and utilize this as our GT for refinement on the training set. Due to inaccurate segmentation we do not make use of B2 and B6. We then apply forward kinematics to obtain the 3D joints from angles, and learn a linear mapping from our skeleton predicted 3D joints to those of the *StereoDS* GT, in order to minimize the bias between both skeletons. At test time, we first predict the angles on B1, then compute joints and apply the mapping. The results are depicted in Figure 5.29 (Right) with [Zimmermann and Brox, 2017] achieving (as expected) a higher Area Under Curve (AUC). Nevertheless, computing the ROC for euler angle errors, as in Figure 5.29 (Left), we notice that the AUC for our method after refinement is almost the same as that of [Zimmermann and Brox, 2017]. In order to quantitatively prove our claim for the discrepancy between training for different tasks, we additionally train a network to predict 3D joints instead of angles, utilizing only our synthetic data and refining on B3-B5. We already notice a boost in the predictions, with the new curve, Figure 5.29 ((Right) Ours (joint

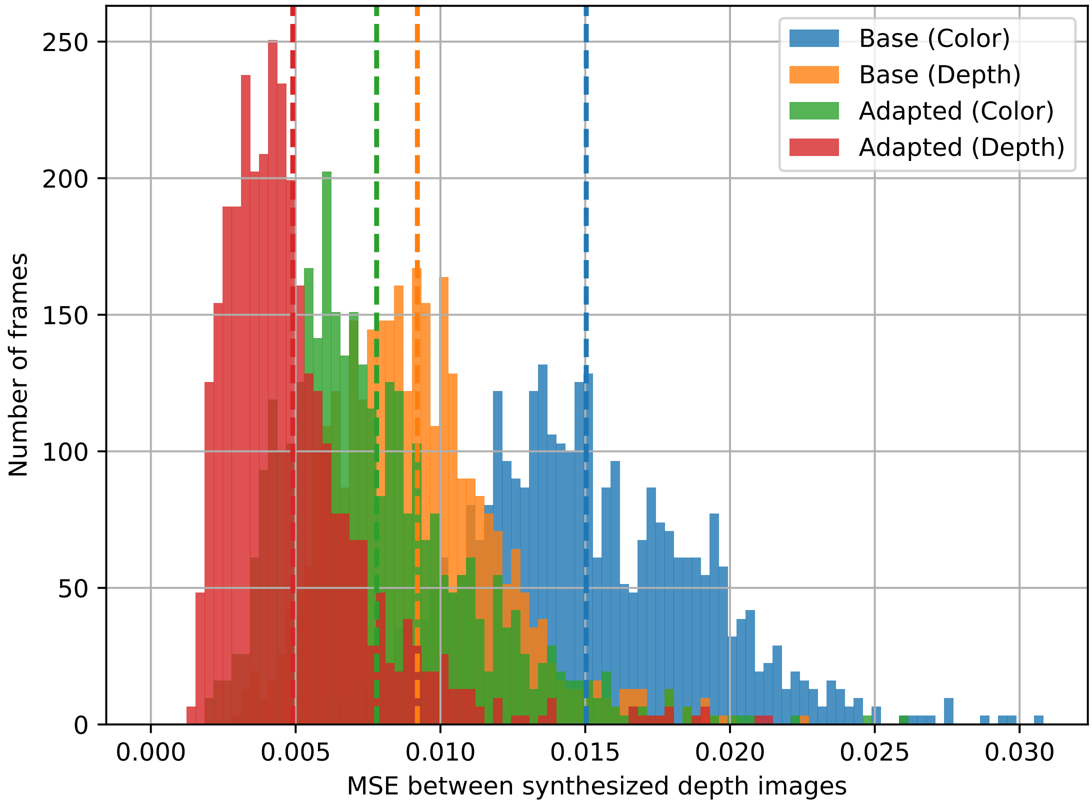


Figure 5.30: *MSE histogram of our (Color) and method from Section 5.1 (Depth) both before (Base) and after (Adapted) after unsupervised refinement.*

regression)), reaching similar accuracy to that of [Zimmermann and Brox, 2017]. We think that the difference between the curves can be due to our refinement only on a part of the complete training set that [Zimmermann and Brox, 2017] was refined on.

Lastly, we compare to our method based on depth images only, and we notice a similar prediction accuracy. For the reader courtesy, we additionally provide the per-frame MSE over 1000 frames, the MSE histogram and ROC curve in Figure 5.32, Figure 5.30 and Figure 5.31. Please note that what is important to be compared from the graphs, except for the increased accuracy due to adaptation, is Base (Depth) vs Adapted (Color). Base (Depth) can be thought of as a CNN method from the literature trained on depth images while Adapted (Color) is our method that has been trained on our synthetic RGB images, and has only seen depth images and utilized them in an unsupervised manner, at little capturing cost. We believe that this example shows

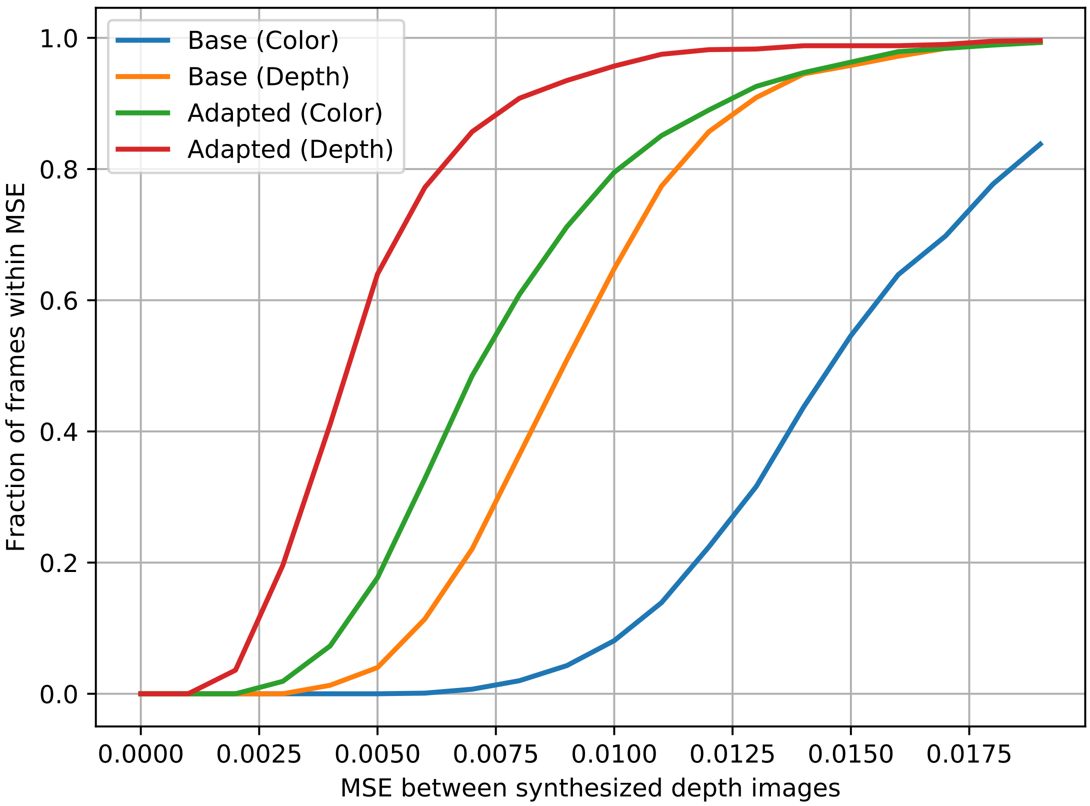


Figure 5.31: *ROC curve corresponding to Figure 5.30.*

the potential of CNN RGB based methods to work on par with Depth-based ones.

5.5 Discussion and Conclusion

Depth Based. We showed that utilizing our depth based method, a base CNN model, trained purely on synthetic data, can be automatically refined to new unlabeled depth images. This method could be utilized both as an extension to previous data-driven methods (under minimal constraints), as well as a stand alone method for 3D pose estimation. The ability of the network to adapt to new poses and shapes, while running real-time on CPU, unlocks further applications, such as personalized gesture recognition or hand-tracking, which could be integrated into smart-phones. Even though we tackle only single depth estimation, it can also be applied to tracking in videos (under minimal jittery).

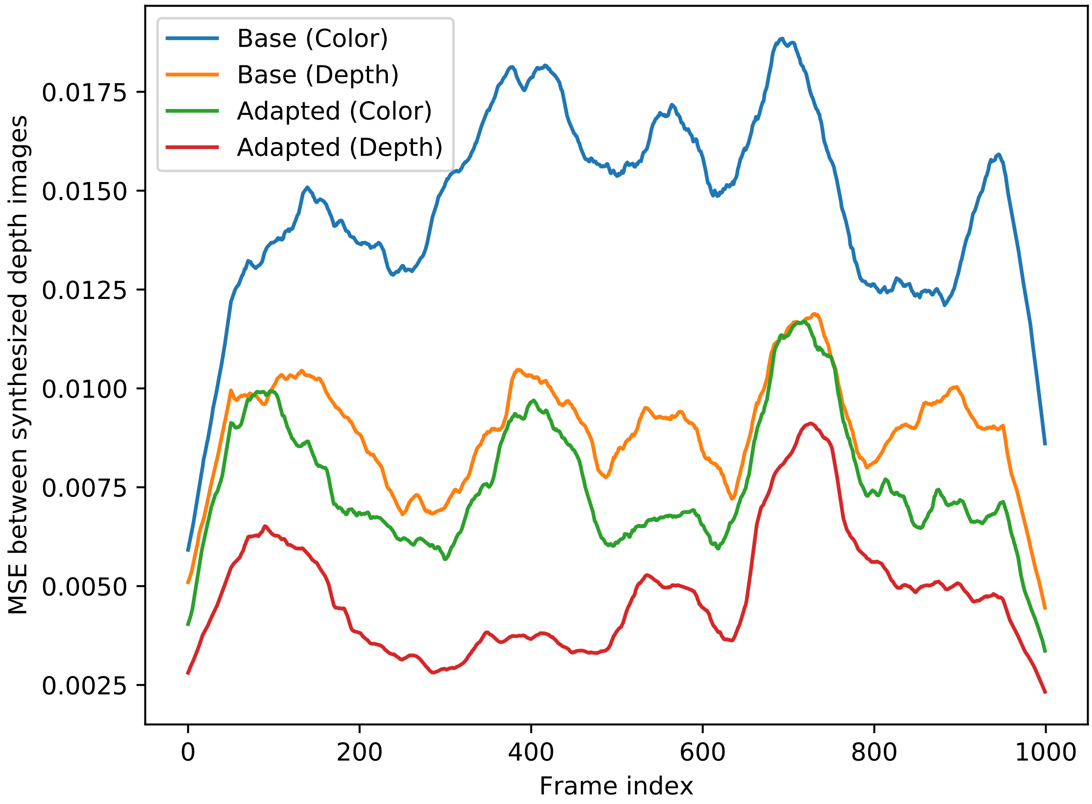


Figure 5.32: *MSE variation per frame of our (Color) and the method from Section 5.1 (Depth) both before (Base) and after (Adapted) after unsupervised refinement.*

We assume that we adapt the base CNN to a single hand shape only. For optimal performance, we require therefore a consistent hand shape and also a good hand segmentation. In order to cope with that, we could potentially extend our model to predict the hand shape for each input image, similar to what we do for the pose estimation. This is possible since our current model internally adapts to a hand shape, in order to help the pose refinement. We also believe that retraining the network with images of a new user is a possible option, if a personalized hand tracker is desired, since training with 50K images takes only about 30 minutes, when trained from scratch.

We require our base CNN to make reasonable predictions, however we have shown that training a CNN merely on synthetic depth data yields sufficient initial estimations. Even with some of the assumptions violated (e.g. non-consistent segmentation when using ICVL or NYU, label mismatching), we could show comparable results to state-of-the-art on two public datasets.

RGB Based. When we switched from Depth to RGB input, we showed, through quantitative and qualitative evaluations, that lightweight CNN-s, trained purely on our newly proposed synthetic dataset, can achieve accurate pose inference for a variety of tasks, strongly competing with and even outperforming existing state-of-the-art. We additionally showed that by extending its construction with a depth loss component, coupled with our pose representation, the accuracy is further improved via semi-supervised and unsupervised training with real unlabeled images. At the moment, we utilize training data generated from a single shape hand model. Despite the fact that we could show generalization on multiple real hands, and good accuracy especially on classification tasks, there is still room for improvement, e.g. experimenting with adding a second shape improved prediction on *HGR* by 10%. Our current optimization model allows an internal adaptation to a hand shape, as in the Depth based case. Coupling our method with recent and more powerful hand shape models such as [Tkach et al., 2016] and [Romero et al., 2017]'s has the potential to improve and personalize hand pose estimation for a variety of human hand shapes.

Even though we showed improvements in segmentation, based on the synthetic dataset, most of it is due to the real GT training data we annotated. As also backed up by our refinement experiments, further real GT datasets with segmentation and pose annotations are very important. Our technique can also be seen as an economic and automatic way of creating a ground truth labeled dataset and we believe will be instrumental in creating new datasets as well.

Lastly, we envisage that both presented methods could be applied, without loss of generality, to human pose estimation tasks under minimal changes to the underlying 3D model representation and architecture.

CHAPTER 6

Conclusion

In this thesis, we investigated the problem of reconstructing the 3D virtual human from monocular imagery, mainly coming from an RGB sensor. Instead of following a holistic approach, we separately considered three constituting parts of the human avatar: the naked body, clothing and the human hand, considering that the human face has received more attention from the community. We mainly focused on the estimation of the 3D shape and pose from 2D images, potentially taken from a smart-phone, and throughout the thesis we utilized discriminative methods to find these mappings, focusing on CNNs, with the intention of preserving low run-times. We leveraged from existing and realistically synthesized datasets to learn important statistics and data-driven priors that can generalize well and provide accurate reconstructions on unseen real input data. Through this process, we did not only base on single views and annotated groundtruth data for supervised learning. We also showed how to utilize multiple views simultaneously, and more importantly how to leverage from multiple views during training time, in order to boost performance achieved from a single view at inference time. On top of that, we demonstrated that learning and refining unsupervised with unlabeled real data is possible, by integrating lightweight differentiable renderers into CNNs.

With respect to the naked body, our aim was to estimate the intrinsic body shape, regardless of the adopted pose, with applications in mind such as shape from selfies, health monitoring and garment fitting. For this, we assumed that the human is depicted in a picture in uniform background, making it possible for a reliable silhouette extraction to be achievable with standard methods, allowing poses under minimal self-occlusion. We tackled this problem with three different approaches: one based on handcrafted features in combination with CCA and random forest regressors, a second one based on simple standard CNNs, and a third one based on more involved CNNs with generative and cross-modal components. We showed robustness to pose changes, silhouette noise and state-of-the-art performance on existing datasets, outperforming also optimization based methods.

We then, tackled the estimation of garment shape from one or two images. Without loss of generalization, we assumed a t-shirt and a dress as our representative clothing whose shape is to be estimated. The images were segmented with standard techniques, under uniform background assumptions. We provided two possible estimations of the garment shape, one that gets deformed from a template garment and another one that gets deformed from the underlying body, providing empirical evidence of the advantages in using one versus the other. We utilized lightweight CNNs in combination with a new realistically rendered garment dataset synthesized under physically correct assumptions, also due to dynamics, to tackle this very difficult problem. Despite training only on synthetic data, to the best of our knowledge, we were the first to show that garment shape estimation also from real images is possible through CNNs.

Lastly, we looked into the problem of inferring a 3D hand pose from an RGB or Depth image. To this end, we proposed an end-to-end CNN system that leveraged from our newly proposed realistically rendered free hand dataset, consisting of 3 Million samples of hands in various poses, orientations, textures and illuminations. This dataset proved to be helpful not only for pose inference tasks, but it also improved hand segmentation. We did not confine ourselves to a fully supervised training with only synthetic data. Instead, we introduced network components based on differentiable renderers that enabled us to train and refine our networks with unlabeled real images in an unsupervised fashion, showing clear improvements. Maintaining simplicity, we could show on-par and improved performance over state-of-the-art methods for two input modalities, under various tasks varying from 3D pose estimation to gesture recognition.

6.1 Limitations and Outlook

Some of the limitations and potential future directions of the methods presented in this thesis were already mentioned in the ending sections of each chapter. Here, we repeat and extend them by presenting some general and more specific future directions that tackle the current limitations that our methods have.

Body Shape Estimation. Throughout this work we made use of one of the first human body parametric models based on SCAPE [Anguelov et al., 2005]. While this served our purpose, a trivial extension would be to incorporate, more efficient, compact and faster models [Loper et al., 2015] that have been developed since then and have been applied to similar tasks [Bogo et al., 2016a]. Additionally, our template model was based on the mean mesh stemming from both female and male population. Learning two separate models would definitely improve inference for the respective tasks.

We decoupled intrinsic shape from shape deformations due to pose changes. While this suffices for the intended applications that we presented, people come in a great variety of poses in everyday life, hence looking into these two problems simultaneously is advantageous. Another point rooting for this approach is that most of the works focusing on 3D human pose estimation, represent poses in terms of simple 3D joints and stick figures, which could in reality result in unrealistic human body shapes or even body interpenetration, as also motivated by [Bogo et al., 2016a]. A combination of these and our methods in a hybrid fashion or even completely through CNNs is an interesting area to explore.

We considered here a binary silhouette as our input from where the body shape is inferred. In order to simplify the problem, we deliberately did not consider very important cues such as shading and texture. This decision was based on a couple of factors: a) a silhouette is the most representative cue or feature of the human body, b) there existed only naked available scans of human bodies, despite the fact that humans are generally depicted with clothing in images. Hence, no ground truth correspondences could be established between images and shapes. Under tight clothing assumptions, a silhouette alleviated this problem. With current advances in datasets, capturing and modeling, it is now possible to look at the very same problem, however considering the full RGB image. Utilizing the methods from the first chapter, coupled with unsupervised techniques from the third chapter, annotated datasets [Lassner et al., 2017b; Varol et al., 2017], differentiable rendering [Loper and Black, 2014] and generative models [Lassner et al., 2017a], is a direction worth exploring. Based on previous [Bălan and Black, 2008] and current [Zhang et al., 2017] works that try to estimate the body shape under clothing from multiple cameras or scanners, we could exploit human body shape priors and trained networks also for monocular imagery. Furthermore, similar to recent work on faces [Tewari et al., 2017], one could think of applying autoencoders and discriminator components to estimate pose, shape, reflectance, lighting and camera parameters simultaneously.

Despite the necessity of estimating shape from a single image, for more consistent shape estimations, we believe that multiple instances of the same person need to be taken into account at various time stamps, and potentially by considering more poses. This brings us towards videos, where not only temporal consistency could be exploited, but also dynamics [Pons-Moll et al., 2015], which we did not explore here. Lastly, in our attempt to utilize data-driven techniques, which are typically based on sensor data (image inputs or body scans), we discarded real-world priors. We could leverage from a well understood knowledge, gathered in the last century, about the inner workings of the human body. Hence, as a continuation, instead of utilizing surface or volumetric based meodels, we are exploring anatomically correct body models, e.g. [Kadlecek et al., 2016], which are useful not only for the task of anatomically correct human shape estimation from RGB images, but also for extrapolation of secondary motion and interaction with external objects and forces.

Garment Shape Estimation. For the task of garment shape estimation, we relied on a data generation pipeline where the deformation of the human body shape and thus the garment is given by a standard skinning model with automatically computed weights and simple blending of bone transformations. These steps introduce certain artifacts. In order to avoid skinning problems, a very reasonable choice would be to use a more compact human body model [Loper et al., 2015] that allows baking it with ready made motion capture systems. We also used an off-the-shelf physical simulator for cloth deformation, with default parameters, few material properties and targeted to one clothing type. Better and tailored databases can be obtained by improving these steps, and considering accurate cloth parameters. The performance of our data-driven approach can only be as good as the training dataset. The more realistic and general the dataset becomes the more details can be captured.

Our method relies on segmented garments from an image, although it tolerates a certain amount of noise and moderately textured garments. By training on more data covering a larger variety of cases and coupling it with techniques based on GANs [Lassner et al., 2017a], the method can be extended to handle complex textures and unsegmented images. Additionally, the networks were trained with relatively small input images, which prevented us from capturing high-frequency details, e.g. wrinkles. Our current method could be augmented with local wrinkle regressors similar to the work by [Cao et al., 2015], under the assumption of a known camera calibration. With the current progress in the compactness of CNN representations and utilizing recently captured cloth datasets from scans [Pons-Moll et al., 2017], we believe that most of the above difficulties can be resolved.

Lastly, as compared to the human body, garment deformations are more noticeable through dynamics. We envisage the extension of our technique to sequences, as in [Guan et al., 2012] and videos, by considering temporal constraints, multiple frames at once and incorporating recurrent network architectures, which we leave as an interesting future direction.

Hand Pose Estimation. We utilize training data generated from a single shape hand model. Despite the fact that we could show generalization on multiple real hands, and good accuracy especially on classification tasks, there is still room for improvement, e.g. experimenting with adding a second shape improved the prediction accuracy by 10% on one of the datasets. Our current optimization model allows an internal adaptation to a hand shape, however this is only useful when a personalized tracker is desired. If we would like to generalize to a variety of human hand shapes, better and larger datasets would be needed, allowing us to estimate not only pose parameters, but also shape parameters. One way to tackle this, would be to couple our method with recent, more powerful hand shape models such as [Tkach et al., 2016] and [Romero et al., 2017]'s, with the latter being an extension of the SMPL model from [Loper et al., 2015]. Following the argument previously made on human body pose, utilizing a hand model versus pure 3D joint and stick-figure or skeleton hand representations, would help to eliminate potential interpenetration between fingers, that are present in the current models.

Except for a few fixed static gestures, human communication is based on sequences and dynamics, hence proper hand tracking is necessary. In this work we focused on tracking by detection, on a per-frame basis, and we could show smooth transitions between frames. Leveraging from works on pose estimation from videos focused on hands [Song et al., 2015] and bodies [Song et al., 2017], we can think of extending our current method such that the predictions become more robust.

We could show that a very crucial component of our system was the generated dataset. This, of course, could be augmented with more textures, shapes, backgrounds and with objects interacting with the hands. There exist however other possibilities that can be taken into account to improve the generalization of the networks trained on such datasets. One option would be to utilize the concept of cross-modal learning as in [Spurr et al., 2018], in order to learn latent spaces from which various modalities (e.g. depth or RGB) could be separately or jointly utilized for pose inferring tasks. Another option, which we are currently exploring, would be to enrich our current synthetically generated dataset through GANs, by learning mappings from synthetic to real images that do not have one-to-one correspondences. In this

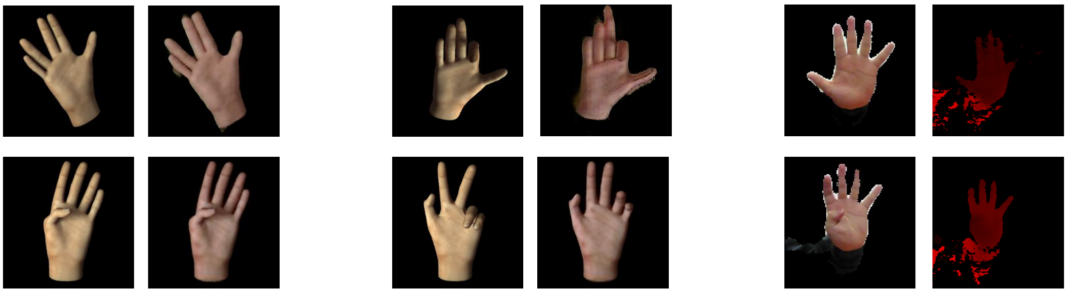


Figure 6.1: *First two columns: RGB hand images generated with GANs having a synthetic RGB hand image as an input. Third column: Depth hand images generated with GANs having an RGB hand image as an input.*

way, we would get a free groundtruth annotation of real-looking hand samples. In addition to obtaining real images from synthetic ones, this could be applicable also to generate depth images as in Figure 6.1. Given a depth image, we could utilize depth based methods, that currently are more accurate than RGB based ones.

Where are we headed? With current hardware getting faster, data acquisition cheaper, and CNN based techniques improving, from a technical perspective, the current hybrid methods, that leverage from good initializations of discriminative methods and accurate refinements of generative methods will probably be slowly replaced by pure weakly supervised or fully unsupervised discriminative methods.

Looking at it from an application view-point, as capturing costs are being reduced and manual tedious work is being replaced by automatization, every nuance of shape and appearance, whether with or without clothes, static or in motion will be obtainable. People will be finally able to recover their 3D virtual doubles, which will make their experience in the virtual world richer and more realistic.

References

[Agarwal et al.,] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. `http://ceres-solver.org`.

[Alexandros Neophytou, 2014] Adrian Hilton Alexandros Neophytou. A layered model of human body and garment deformation. 2014.

[Amberg et al., 2007] Brian Amberg, Sami Romdhani, and Thomas Vetter. Optimal step nonrigid ICP algorithms for surface registration. In *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA*, 2007.

[Andrew et al., 2013] Galen Andrew, Raman Arora, Jeff A. Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 1247–1255, 2013.

[Anguelov et al., 2005] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: Shape completion and animation of people. In *SIGGRAPH*, 2005.

[Athitsos and Sclaroff, 2003] Vassilis Athitsos and Stan Sclaroff. Estimating 3d hand pose from a cluttered image. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), 16-22 June 2003, Madison, WI, USA*, pages 432–442, 2003.

[Aubry et al., 2011] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1626–1633. IEEE, 2011.

[Badrinarayanan et al., 2015] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *CoRR*, abs/1511.00561, 2015.

[Balan et al., 2007] Alexandru O. Balan, Leonid Sigal, Michael J. Black, James E. Davis, and Horst W. Haussecker. Detailed human shape and pose from images. In *CVPR*, 2007.

[Bambach et al., 2015] Sven Bambach, Stefan Lee, David J. Crandall, and Chen Yu. Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[Baran and Popovic, 2007a] Ilya Baran and Jovan Popovic. Automatic rigging and animation of 3d characters. *ACM Trans. Graph.*, 2007.

[Baran and Popović, 2007b] Ilya Baran and Jovan Popović. Automatic rigging and animation of 3d characters. In *ACM SIGGRAPH 2007 Papers*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.

[Beeler et al., 2010] Thabo Beeler, Bernd Bickel, Paul A. Beardsley, Bob Sumner, and Markus H. Gross. High-quality single-shot capture of facial geometry. *ACM Trans. Graph.*, 29(4):40:1–40:9, 2010.

[Bérard et al., 2014] Pascal Bérard, Derek Bradley, Maurizio Nitti, Thabo Beeler, and Markus Gross. High-quality capture of eyes. *ACM Trans. Graph.*, 33(6), 2014.

[Bérard et al., 2016] Pascal Bérard, Derek Bradley, Markus Gross, and Thabo Beeler. Lightweight eye capture using a parametric model. *ACM Trans. Graph.*, 35(4), 2016.

[Bogo et al., 2016a] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *European Conference on Computer Vision*, pages 561–578. Springer, 2016.

[Bogo et al., 2016b] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter V. Gehler, Javier Romero, and Michael J. Black. Keep it SMPL: automatic estimation of 3d human pose and shape from a single image. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V*, pages 561–578, 2016.

[Boisvert et al., 2013] Jonathan Boisvert, Chang Shu, Stefanie Wuhrer, and Pengcheng Xi. Three-dimensional human shape inference from silhouettes: reconstruction and validation. *Mach. Vis. Appl.*, 24(1):145–157, 2013.

[Boscaini et al., 2016a] Davide Boscaini, Jonathan Masci, Emanuele Rodolà, and Michael M. Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. Technical Report arXiv:1605.06437, 2016.

[Boscaini et al., 2016b] Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Michael M. Bronstein, and Daniel Cremers. Anisotropic diffusion descriptors. volume 35, pages 431–441, 2016.

[Boykov and Funka-Lea, 2006] Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient n-d image segmentation. *Int. J. Comput. Vision*, 70(2):109–131, November 2006.

[Boykov and Jolly, 2001] Yuri Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *ICCV*, pages 105–112, 2001.

[Bradley et al., 2008] Derek Bradley, Tiberiu Popa, Alla Sheffer, Wolfgang Heidrich, and Tamy Boubekeur. Markerless garment capture. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 27(3):99, 2008.

[Breiman, 2001] Leo Breiman. Random forests. *Machine Learning*, 2001.

[Bronstein et al., 2010] Alexander M Bronstein, Michael M Bronstein, Ron Kimmel, Mona Mahmoudi, and Guillermo Sapiro. A gromov-hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching. *International Journal of Computer Vision*, 89:266–286, 2010.

[Bălan and Black, 2008] Alexandru O. Bălan and Michael J. Black. The naked truth: Estimating body shape under clothing. In *ECCV*, 2008.

[Cao et al., 2015] Chen Cao, Derek Bradley, Kun Zhou, and Thabo Beeler. Real-time high-fidelity facial performance capture. *ACM Trans. Graph.*, 34(4):46:1–46:9, July 2015.

[Cao et al., 2017] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Real-time multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[Carranza et al., 2003] Joel Carranza, Christian Theobalt, Marcus A. Magnor, and Hans-Peter Seidel. Free-viewpoint video of human actors. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, pages 569–577, New York, NY, USA, 2003. ACM.

[Casas et al., 2014] Dan Casas, Marco Volino, John Collomosse, and Adrian Hilton. 4d video textures for interactive character appearance. *Comp. Graph. Forum (Proc. Eurographics)*, 2014.

[Chandar et al., 2016] Sarath Chandar, Mitesh M. Khapra, Hugo Larochelle, and Balaraman Ravindran. Correlational neural networks. *Neural Computation*, 28(2):257–285, 2016.

[Chen and Cipolla, 2009] Yu Chen and Roberto Cipolla. Learning shape priors for single view reconstruction. In *ICCV Workshops*, 2009.

[Chen et al., 2010] Yu Chen, Tae-Kyun Kim, and Roberto Cipolla. Inferring 3d shapes and deformations from single views. In *ECCV*, 2010.

[Chen et al., 2011] Yu Chen, Tae-Kyun Kim, and Roberto Cipolla. Silhouette-based object phenotype recognition using 3d shape priors. In *ICCV*, 2011.

[Chen et al., 2013] Xiaowu Chen, Yu Guo, Bin Zhou, and Qinping Zhao. Deformable model for estimating clothed and naked human shapes from a single image. *The Visual Computer*, 2013.

[Chen et al., 2015] Wenlin Chen, James T. Wilson, Stephen Tyree, Kilian Q. Weinberger, and Yixin Chen. Compressing convolutional neural networks. *CoRR*, abs/1506.04449, 2015.

[Cheng et al., 2016] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1335–1344, 2016.

[Choi et al., 2017] Chiho Choi, Sangpil Kim, and Karthik Ramani. Learning hand articulations by hallucinating heat distribution. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 3123–3132, 2017.

[CMU,] CMU. Carnegie-Mellon Mocap Database.

[Danhang Tang, 2014] Alykhan Tejani T-K. Kim Danhang Tang, Hyung Jin Chang. Latent regression forest: Structured estimation of 3d articulated hand posture. *CVPR*, 2014.

[de Aguiar et al., 2008] Edilson de Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. Performance capture from sparse multi-view video. In *SIGGRAPH*, 2008.

[de La Gorce et al., 2011] M. de La Gorce, D. J. Fleet, and N. Paragios. Model-based 3d hand pose estimation from monocular video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1793–1805, Sept 2011.

[Delamarre and Faugeras, 1999] Q. Delamarre and O. Faugeras. 3d articulated models and multi-view tracking with silhouettes. In *ICCV*, 1999.

[Deng et al., 2017] Xiaoming Deng, Shuo Yang, Yinda Zhang, Ping Tan, Liang Chang, and Hongan Wang. Hand3d: Hand pose estimation using 3d neural network. *CoRR*, abs/1704.02224, 2017.

[Dreuw et al., 2006] Philippe Dreuw, Thomas Deselaers, Daniel Keysers, and Hermann Ney. Modeling image variability in appearance-based gesture recognition. In *ECCV Workshop on Statistical Methods in Multi-Image and Video Processing*, pages 7–18, Graz, Austria, May 2006.

[Duchi et al., 2011] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[Eisemann et al., 2008] Martin Eisemann, Bert de Decker, Marcus A. Magnor, Philippe Bekaert, Edilson de Aguiar, Naveed Ahmed, Christian Theobalt, and Anita Sellent. Floating textures. *Comput. Graph. Forum*, 27(2):409–418, 2008.

[Erol et al., 2007] Ali Erol, George Bebis, Mircea Nicolescu, Richard D. Boyle, and Xander Twombly. Vision-based hand pose estimation: A review. *Comput. Vis. Image Underst.*, 108(1-2):52–73, October 2007.

[Fang et al., 2015] Yi Fang, Jin Xie, Guoxian Dai, Meng Wang, Fan Zhu, Tiantian Xu, and Edward Wong. 3d deep shape descriptor. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[Feng et al., 2015] Andrew Feng, Dan Casas, and Ari Shapiro. Avatar reshaping and automatic rigging using a deformable model. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, pages 57–64, Paris, France, November 2015. ACM Press.

[Fischer et al., 2015] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. *CoRR*, abs/1504.06852, 2015.

[Fleming et al., 2017] Reuben Fleming, Betty J. Mohler, Javier Romero, Michael J. Black, and Martin Breidt. *Appealing Avatars from 3D Body Scans: Perceptual Effects of Stylization*, pages 175–196. Springer International Publishing, 2017.

[Gall et al., 2009] Juergen Gall, Carsten Stoll, Edilson de Aguiar, Christian Theobalt, Bodo Rosenhahn, and Hans-Peter Seidel. Motion capture using joint skeleton tracking and surface estimation. In *CVPR*, pages 1746–1753, 2009.

[Ge et al., 2016] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns. In *Proc. CVPR*, 2016.

[Girshick et al., 2014] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[Glorot and Bengio, 2010] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS 10). Society for Artificial Intelligence and Statistics*, 2010.

[Goodfellow et al., 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[Gordon et al., 1989] C. C. Gordon, T. Churchill, C. E. Clauser, B. Bradtmiller, and J. T. McConville. Anthropometric survey of US Army personnel: Summary statistics, interim report for 1988. Technical report, DTIC Document, 1989.

[Grzejszczak et al., 2016] Tomasz Grzejszczak, Michal Kawulok, and Adam Galuszka. Hand landmarks detection and localization in color images. *Multimedia Tools and Applications*, 75(23):16363–16387, 2016.

[Guan et al., 2008] Li Guan, Jean-Sébastien Franco, and Marc Pollefeys. Multi-object shape estimation and tracking from silhouette cues. In *CVPR*, 2008.

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

[Guan et al., 2010] Peng Guan, Oren Freifeld, and Michael J. Black. A 2d human body model dressed in eigen clothing. In *Proceedings of the 11th European Conference on Computer Vision: Part I*, ECCV'10, pages 285–298, Berlin, Heidelberg, 2010. Springer-Verlag.

[Guan et al., 2012] Peng Guan, Loretta Reiss, David A. Hirshberg, Alexander Weiss, and Michael J. Black. Drape: Dressing any person. *ACM Trans. Graph.*, 2012.

[Guo et al., 2017] Hengkai Guo, Guijin Wang, Xinghao Chen, Cairong Zhang, Fei Qiao, and Huazhong Yang. Region ensemble network: Improving convolutional network for hand pose estimation. *CoRR*, abs/1702.02447, 2017.

[Guskov et al., 2003] Igor Guskov, Sergey Klibanov, and Benjamin Bryant. Trackable surfaces. In *SCA*, 2003.

[Hahn et al., 2014] Fabian Hahn, Bernhard Thomaszewski, Stelian Coros, Robert W. Sumner, Forrester Cole, Mark Meyer, Tony DeRose, and Markus Gross. Subspace clothing simulation using adaptive bases. *ACM Trans. Graph.*, 33(4):105:1–105:9, July 2014.

[Han et al., 2015] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. *CoRR, abs/1510.00149*, 2, 2015.

[Hardoon et al., 2004] David R. Hardoon, Sandor R. Szedmak, and John R. Shawe-taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Comput.*, 2004.

[Hardoon et al., 2007] David R. Hardoon, Janaina Mourão Miranda, Michael Brammer, and John Shawe-Taylor. Unsupervised analysis of fmri data using kernel canonical correlation. *NeuroImage*, 2007.

[Hasler et al., 2009] Nils Hasler, Carsten Stoll, Martin Sunkel, Bodo Rosenhahn, and Hans-Peter Seidel. A statistical model of human pose and body shape. *Comput. Graph. Forum*, 2009.

[Hasler et al., 2010] Nils Hasler, Hanno Ackermann, Bodo Rosenhahn, Thorsten Thormählen, and Hans-Peter Seidel. Multilinear pose and body shape estimation of dressed subjects from image sets. In *CVPR*, 2010.

[He et al., 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[He et al., 2017] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. *arXiv preprint arXiv:1703.06870*, 2017.

[Helten et al., 2013] Thomas Helten, Andreas Baak, Gaurav Bharaj, Meinard Müller, Hans-Peter Seidel, and Christian Theobalt. Personalization and evaluation of a real-time depth-based full body tracker. In *3DV*, 2013.

[Hilliges et al., 2012] Otmar Hilliges, David Kim, Shahram Izadi, Malte Weiss, and Andrew Wilson. Holodesk: Direct 3d interactions with a situated see-through display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2421–2430, New York, NY, USA, 2012. ACM.

[Hirshberg et al., 2012] D. Hirshberg, M. Loper, E. Rachlin, and M.J. Black. Coregistration: Simultaneous alignment and modeling of articulated 3D shape. In *European Conf. on Computer Vision (ECCV)*, LNCS 7577, Part IV, pages 242–255. Springer-Verlag, October 2012.

[Hotelling, 1936] Harold Hotelling. Relations between two sets of variates. *Biometrika*, 1936.

[Hu et al., 2015] Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. Single-view hair modeling using a hairstyle database. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2015)*, 34(4), July 2015.

[Hu et al., 2017] Liwen Hu, Shunsuke Saito, Lingyu Wei, Koki Nagano, Jaewoo Seo, Jens Fursund, Iman Sadeghi, Carrie Sun, Yen-Chun Chen, and Hao Li. Avatar digitization from a single image for real-time rendering. *ACM Trans. Graph.*, 36(6):195:1–195:14, 2017.

[Iandola et al., 2016a] Forrest N Iandola, Matthew W Moskewicz, Khalid Ashraf, Song Han, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 1mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[Iandola et al., 2016b] Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. *CoRR*, abs/1602.07360, 2016.

[Ilg et al., 2017] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1647–1655, 2017.

[Ioffe and Szegedy, 2015] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 448–456, 2015.

[Iyer et al., 2005] Natraj Iyer, Subramaniam Jayanti, Kuiyang Lou, Yagnanarayanan Kalyanaraman, and Karthik Ramani. Three-dimensional shape searching: state-of-the-art review and future trends. *Computer-aided Design*, 37(5):509–530, 2005.

[Jain et al., 2010] Arjun Jain, Thorsten Thormählen, Hans-Peter Seidel, and Christian Theobalt. Moviereshape: Tracking and reshaping of humans in videos. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 2010.

[Jakob, 2010] Wenzel Jakob. Mitsuba renderer, 2010. http://www.mitsuba-renderer.org.

[Jeong et al., 2015] Moon-Hwan Jeong, Dong-Hoon Han, and Hyeong-Seok Ko. Garment capture from a photograph. *Comput. Animat. Virtual Worlds*, 26(3-4):291–300, May 2015.

[Kadlecek et al., 2016] Petr Kadlecek, Alexandru Eugen Ichim, Tiantian Liu, Jaroslav Krivánek, and Ladislav Kavan. Reconstructing personalized anatomical models for physics-based body animation. *ACM Trans. Graph.*, 35(6):213:1–213:13, 2016.

[Kakade and Foster, 2007] Sham M. Kakade and Dean P. Foster. Multi-view regression via canonical correlation analysis. In *COLT*, 2007.

[Kakadiaris and Metaxas, 1998] Ioannis A. Kakadiaris and Dimitri Metaxas. Three-dimensional human body model acquisition from multiple views. *International Journal of Computer Vision*, 30(3):191–218, 1998.

[Kavan et al., 2008] Ladislav Kavan, Steven Collins, Jiří Žára, and Carol O'Sullivan. Geometric skinning with approximate dual quaternion blending. *ACM Trans. Graph.*, 27(4):105:1–105:23, November 2008.

[Kawulok et al., 2014] Michal Kawulok, Jolanta Kawulok, Jakub Nalepa, and Bogdan Smolka. Self-adaptive algorithm for segmenting skin regions. *EURASIP Journal on Advances in Signal Processing*, 2014(170):1–22, 2014.

[Kendall et al., 2015] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2938–2946, 2015.

[Keskin et al., 2012] Cem Keskin, Furkan Kıraç, Yunus Emre Kara, and Lale Akarun. *Hand Pose Estimation and Hand Shape Classification Using Multi-layered Randomized Decision Forests*, pages 852–863. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[Kim et al., 2007] Tae-Kyun Kim, Shu-Fai Wong, and Roberto Cipolla. Tensor canonical correlation analysis for action classification. In *CVPR*, 2007.

[Kim et al., 2017] Hyeongwoo Kim, Michael Zollhöfer, Ayush Tewari, Justus Thies, Christian Richardt, and Christian Theobalt. Inversefacenet: Deep single-shot inverse face rendering from A single image. *CoRR*, abs/1703.10956, 2017.

[Kingma and Ba, 2014] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[Krizhevsky et al., 2012a] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[Krizhevsky et al., 2012b] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[Lahner et al., 2016] Zorah Lahner, Emanuele Rodola, Frank R. Schmidt, Michael M. Bronstein, and Daniel Cremers. Efficient globally optimal 2d-to-3d deformable shape matching. In *CVPR*, 2016.

[Lassner et al., 2017a] Christoph Lassner, Gerard Pons-Moll, and Peter V. Gehler. A generative model of people in clothing. In *International Conference on Computer Vision (ICCV)*, 2017.

[Lassner et al., 2017b] Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J. Black, and Peter V. Gehler. Unite the people: Closing the loop between 3d and 2d human representations. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*, Piscataway, NJ, USA, July 2017. IEEE.

[Laurentini, 1994] A. Laurentini. The visual hull concept for silhouette-based image understanding. *PAMI*, 1994.

[Lee and Kunii, 1995] Jintae Lee and Tosiyasu L. Kunii. Model-based analysis of hand posture. *IEEE Comput. Graph. Appl.*, 15(5):77–86, September 1995.

[Lewis et al., 2000] J. P. Lewis, Matt Cordner, and Nickson Fong. Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 165–172, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[Ling and Jacobs, 2007] Haibin Ling and David W. Jacobs. Shape classification using the inner-distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(2):286–299, February 2007.

[Litman and Bronstein, 2014] Roee Litman and Alexander M. Bronstein. Learning spectral descriptors for deformable shape correspondence. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(1):171–180, 2014.

[Liu et al., 2016] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[Long et al., 2014] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014.

[Long et al., 2015] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[Loper and Black, 2014] Matthew M. Loper and Michael J. Black. OpenDR: An approximate differentiable renderer. In *Computer Vision – ECCV 2014*, volume

8695 of *Lecture Notes in Computer Science*, pages 154–169. Springer International Publishing, September 2014.

[Loper et al., 2015] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: a skinned multi-person linear model. *ACM Trans. Graph.*, 34(6):248:1–248:16, 2015.

[Maaten and Hinton, 2008] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[Masci et al., 2015a] Jonathan Masci, Davide Boscaini, Michael M. Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proc. of the IEEE International Conference on Computer Vision (ICCV) Workshops*, pages 37–45, 2015.

[Masci et al., 2015b] Jonathan Masci, Davide Boscaini, Michael M. Bronstein, and Pierre Vandergheynst. Shapenet: Convolutional neural networks on non-euclidean manifolds. *CoRR*, abs/1501.06297, 2015.

[McWilliams et al., 2013] Brian McWilliams, David Balduzzi, and Joachim M Buhmann. Correlated random features for fast semi-supervised learning. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 440–448. Curran Associates, Inc., 2013.

[MD,] MD. Marvelous designer.

[Mehta et al., 2016] Dushyant Mehta, Helge Rhodin, Dan Casas, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation using transfer learning and improved CNN supervision. *CoRR*, abs/1611.09813, 2016.

[Mehta et al., 2017] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: real-time 3d human pose estimation with a single RGB camera. *ACM Trans. Graph.*, 36(4):44:1–44:14, 2017.

[Melax et al., 2013] Stan Melax, Leonid Keselman, and Sterling Orsten. Dynamics based 3d skeletal hand tracking. In *Proceedings of Graphics Interface 2013*, GI '13, pages 63–70, Toronto, Ont., Canada, Canada, 2013. Canadian Information Processing Society.

[Memo and Zanuttigh, 2017] Alvise Memo and Pietro Zanuttigh. Head-mounted gesture controlled interface for human-computer interaction. In *Multimedia Tools and Applications*, 2017.

[Memo et al., 2015] Alvise Memo, Ludovico Minto, and Pietro Zanuttigh. Exploiting silhouette descriptors and synthetic data for hand gesture recognition. In *Smart Tools and Apps for Graphics - Eurographics Italian Chapter Conference, Verona, Italy, October 15-16 2015.*, pages 15–23, 2015.

[Mikic et al., 2003] Ivana Mikic, Mohan Trivedi, Edward Hunter, and Pamela Cosman. Human body model acquisition and tracking using voxel data. *International Journal of Computer Vision*, 53(3):199–223, 2003.

[Mölbert et al., 2017] Simone Claire Mölbert, Anne Thaler, Betty J. Mohler, Stephan Streuber, Javier Romero, Michael J. Black, Stephan Zipfel, Hans-Otto Karnath, and Katrin Elisabeth Giel. Assessing body image in anorexia nervosa using biometric self-avatars in virtual reality: Attitudinal components rather than visual body size estimation are distorted. *Psychological Medicine*, 26:1–12, July 2017.

[Mueller et al., 2017] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. Ganerated hands for real-time 3d hand tracking from monocular RGB. *CoRR*, abs/1712.01057, 2017.

[Nalepa and Kawulok, 2014] Jakub Nalepa and Michal Kawulok. Fast and accurate hand shape classification. In Stanislaw Kozielski, Dariusz Mrozek, Pawel Kasprowski, Bozena Malysiak-Mrozek, and Daniel Kostrzewa, editors, *Beyond Databases, Architectures, and Structures*, volume 424 of *Communications in Computer and Information Science*, pages 364–373. Springer, 2014.

[Narain et al., 2012] Rahul Narain, Armin Samii, and James F. O'Brien. Adaptive anisotropic remeshing for cloth simulation. *ACM Transactions on Graphics*, 31(6):147:1–10, November 2012. Proceedings of ACM SIGGRAPH Asia 2012, Singapore.

[Narain et al., 2013] Rahul Narain, Tobias Pfaff, and James F. O'Brien. Folding and crumpling adaptive sheets. *ACM Transactions on Graphics*, 32(4):51:1–8, July 2013. Proceedings of ACM SIGGRAPH 2013, Anaheim.

[Neophytou and Hilton, 2013] Alexandros Neophytou and Adrian Hilton. Shape and pose space deformation for subject specific animation. In *3DV*, 2013.

[Ngiam et al., 2011] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. Ng. Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 689–696, 2011.

[Oberweger et al., 2015a] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Hands deep in deep learning for hand pose estimation. *CoRR*, abs/1502.06807, 2015.

[Oberweger et al., 2015b] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Training a feedback loop for hand pose estimation. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 3316–3324, Washington, DC, USA, 2015. IEEE Computer Society.

[Oberweger et al., 2016] Markus Oberweger, Gernot Riegler, Paul Wohlhart, and Vincent Lepetit. Efficiently creating 3d training data for fine hand pose estimation. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4957–4965, 2016.

[Oikonomidis et al., 2011] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *Proceedings of the British Machine Vision Conference*, pages 101.1–101.11, 2011.

[Panteleris et al., 2017] Paschalis Panteleris, Iason Oikonomidis, and Antonis A. Argyros. Using a single rgb frame for real time 3d hand pose estimation in the wild. *CoRR*, abs/1712.03866, 2017.

[Perbet et al., 2014] Frank Perbet, Sam Johnson, Minh-Tri Pham, and Björn Stenger. Human body shape estimation using a multi-resolution manifold forest. In *CVPR*, 2014.

[Pickup et al., 2014] D. Pickup, X. Sun, P. L. Rosin, R. R. Martin, Z. Cheng, Z. Lian, M. Aono, A. Ben Hamza, A. Bronstein, M. Bronstein, S. Bu, U. Castellani, S. Cheng, V. Garro, A. Giachetti, A. Godil, J. Han, H. Johan, L. Lai, B. Li, C. Li, H. Li, R. Litman, X. Liu, Z. Liu, Y. Lu, A. Tatsuma, and J. Ye. Shape retrieval of non-rigid 3d human models. In *Proceedings of the 7th Eurographics Workshop on 3D Object Retrieval*, 3DOR '15, pages 101–110, Aire-la-Ville, Switzerland, Switzerland, 2014. Eurographics Association.

[Piryankova et al., 2014] I. Piryankova, J. Stefanucci, J. Romero, S. de la Rosa, M. Black, and B. Mohler. Can i recognize my body's weight? the influence of shape and texture on the perception of self. *ACM Transactions on Applied Perception for the Symposium on Applied Perception*, 11(3):13:1–13:18, September 2014.

[Pishchulin et al., 2015] Leonid Pishchulin, Stefanie Wuhrer, Thomas Helten, Christian Theobalt, and Bernt Schiele. Building statistical shape spaces for 3d human modeling. *CoRR*, 2015.

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

ACM Transactions on Graphics, (Proc. SIGGRAPH), 34(4):120:1–120:14, August 2015.

[Pons-Moll et al., 2017] Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael Black. Clothcap: Seamless 4D clothing capture and retargeting. *ACM Transactions on Graphics, (Proc. SIGGRAPH)*, 36(4), 2017. Two first authors contributed equally.

[Popa et al., 2009] Tiberiu Popa, Qingnan Zhou, Derek Bradley, Vladislav Kraevoy, Hongbo Fu, Alla Sheffer, and Wolfgang Heidrich. Wrinkling captured garments using space-time data-driven deformation. *Computer Graphics Forum (Proc. Eurographics)*, 28(2):427–435, 2009.

[Pritchard and Heidrich, 2003] D. Pritchard and W. Heidrich. Cloth motion capture, 2003.

[Qian et al., 2014] Chen Qian, Xiao Sun, Yichen Wei, Xiaoou Tang, and Jian Sun. Realtime and robust hand tracking from depth. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1106–1113, 2014.

[Remelli et al., 2017] Edoardo Remelli, Anastasia Tkach, Andrea Tagliasacchi, and Mark Pauly. Low-dimensionality calibration through local anisotropic scaling for robust hand model personalization. In *Proceedings of the International Conference on Computer Vision*, 2017.

[Ren et al., 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[Rhodin et al., 2016] Helge Rhodin, Nadia Robertini, Dan Casas, Christian Richardt, Hans-Peter Seidel, and Christian Theobalt. General automatic human shape and motion capture using volumetric contour cues. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V*, pages 509–526, 2016.

[Robinette and Daanen, 1999] Kathleen M. Robinette and Hein A. M. Daanen. The caesar project: A 3-d surface anthropometry survey. In *3DIM*, 1999.

[Robson et al., 2011] C. Robson, R. Maharik, A. Sheffer, and N. Carr. Context-aware garment modeling from sketches. *Computers and Graphics (Proc. SMI 2011)*, pages 604–613, 2011.

[Rogge et al., 2014] Lorenz Rogge, Felix Klose, Michael Stengel, Martin Eisemann, and Marcus Magnor. Garment replacement in monocular video sequences. *ACM Trans. Graph.*, 2014.

[Rohit Girdhar, 2016] Mikel Rodriguez A K Gupta Rohit Girdhar, David F Fouhey. Learning a predictable and generative vector representation for objects. *European Conference on Computer Vision*, 2016.

[Romero et al., 2009] J. Romero, H. Kjellström, and D. Kragic. Monocular real-time 3d articulated hand pose estimation. In *2009 9th IEEE-RAS International Conference on Humanoid Robots*, pages 87–92, Dec 2009.

[Romero et al., 2017] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), November 2017. (*) Two first authors contributed equally.

[Roveri, 2018] Riccardo Roveri. Tech report: Projection of unordered point sets and generation of distance field images with gaussian interpolation. Technical report, 2018.

[Rustamov, 2007] Raif M Rustamov. Laplace-beltrami eigenfunctions for deformation invariant shape representation. 2007.

[Sarafianos et al., 2016] Nikolaos Sarafianos, Bogdan Boteanu, Bogdan Ionescu, and Ioannis A. Kakadiaris. 3d human pose estimation: A review of the literature and analysis of covariates. *Computer Vision and Image Understanding*, 152:1–20, 2016.

[Sargin et al., 2007] Mehmet Emre Sargin, Yücel Yemez, Engin Erzin, and A. Murat Tekalp. Audiovisual synchronization and fusion using canonical correlation analysis. *Trans. Multimedia*, 2007.

[Savva et al.,] M Savva, F Yu, Hao Su, M Aono, B Chen, D Cohen-Or, W Deng, Hang Su, S Bai, X Bai, et al. Shrec16 track large-scale 3d shape retrieval from shapenet core55.

[Schmidt et al., 2007] Frank R. Schmidt, Dirk Farin, and Daniel Cremers. Fast matching of planar shapes in sub-cubic runtime. In *ICCV*, 2007.

[Schmidt et al., 2009] Frank R. Schmidt, Eno Töppe, and Daniel Cremers. Efficient planar graph cuts with applications in computer vision. In *CVPR*, 2009.

[Scholz and Magnor, 2004] Volker Scholz and Marcus A. Magnor. Cloth motion from optical flow. In *VMV*, 2004.

[Scholz et al., 2005] Volker Scholz, Timo Stich, Michael Keckeisen, Markus Wacker, and Marcus Magnor. Garment motion capture using color-coded patterns. *Computer Graphics forum*, 24(3):439–448, September 2005. Conference Issue: 26th annual Conference Eurographics 2005, Dublin, Ireland, August 29th - September 2nd, 2005.

[Schröder et al., 2014] Matthias Schröder, Jonathan Maycock, Helge Ritter, and Mario Botsch. Real-time hand tracking using synergistic inverse kinematics. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[Sekine et al., 2014] M. Sekine, K. Sugita, F. Perbet, B. Stenger, and M. Nishiyama. Virtual fitting by single-shot body shape estimation. In *Int. Conf. on 3D Body Scanning Technologies*, pages 406–413, October 2014.

[Sermanet et al., 2013] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2013.

[Shapira et al., 2008] Lior Shapira, Ariel Shamir, and Daniel Cohen-Or. Consistent mesh partitioning and skeletonisation using the shape diameter function. *Visual Comput.*, 2008.

[Sharma et al., 2012] Abhishek Sharma, Abhishek Kumar, Hal Daumé III, and David W. Jacobs. Generalized multiview analysis: A discriminative latent space. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 2160–2167, 2012.

[Sharp et al., 2015] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, Daniel Freedman, Pushmeet Kohli, Eyal Krupka, Andrew Fitzgibbon, and Shahram Izadi. Accurate, robust, and flexible real-time hand tracking. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 3633–3642, New York, NY, USA, 2015. ACM.

[Shen et al., 2011] Jianbing Shen, Xiaoshan Yang, Yunde Jia, and Xuelong Li. Intrinsic images using optimization. In *CVPR*, pages 3481–3487. IEEE Computer Society, 2011.

[Sigal et al., 2007] Leonid Sigal, Alexandru O. Balan, and Michael J. Black. Combined discriminative and generative articulated pose and non-rigid shape estimation. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pages 1337–1344, 2007.

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

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[Sinha et al., 2016] Ayan Sinha, Chiho Choi, and Karthik Ramani. Deephand: Robust hand pose estimation by completing a matrix imputed with deep features. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[Slama et al., 2013] Rim Slama, Hazem Wannous, and Mohamed Daoudi. Extremal human curves: A new human body shape and pose descriptor. In *FG*, 2013.

[Song et al., 2015] Jie Song, Fabrizio Pece, Gábor Sörös, Marion Koelle, and Otmar Hilliges. Joint estimation of 3d hand position and gestures from monocular video for mobile interaction. In *ACM Conference on Human Factors in Computing Systems (CHI)*, CHI '15, pages 3657–3660, New York, NY, USA, 2015. ACM.

[Song et al., 2017] Jie Song, Limin Wang, Luc Van Gool, and Otmar Hilliges. Thin-slicing network: A deep structured model for pose estimation in videos. In *CVPR*, 2017.

[Spurr et al., 2018] Adrian Spurr, Jie Song, Seonwook Park, and Otmar Hilliges. Cross-modal deep variational hand pose estimation. In *CVPR*, 2018.

[Sridhar et al., 2013] Srinath Sridhar, Antti Oulasvirta, and Christian Theobalt. Interactive markerless articulated hand motion tracking using rgb and depth data. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*, ICCV '13, pages 2456–2463, Washington, DC, USA, 2013. IEEE Computer Society.

[Sridhar et al., 2014] Srinath Sridhar, Helge Rhodin, Hans-Peter Seidel, Antti Oulasvirta, and Christian Theobalt. Real-time hand tracking using a sum of anisotropic gaussians model. In *Proceedings of the International Conference on 3D Vision (3DV)*, December 2014.

[Sridhar et al., 2016] Srinath Sridhar, Franziska Mueller, Michael Zollhöfer, Dan Casas, Antti Oulasvirta, and Christian Theobalt. Real-time joint tracking of a hand manipulating an object from RGB-D input. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II*, pages 294–310, 2016.

[Starck et al., 2005] J. Starck, G. Miller, and A. Hilton. Video-based character animation. In *ACM SIGGRAPH Eurographics SCA*, 2005.

[Stoll et al., 2010] Carsten Stoll, Juergen Gall, Edilson de Aguiar, Sebastian Thrun, and Christian Theobalt. Video-based reconstruction of animatable human characters. In *SIGGRAPH Asia*, 2010.

[Su et al., 2015] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 945–953, 2015.

[Sugano et al., 2014] Yusuke Sugano, Yasuyuki Matsushita, and Yoichi Sato. Learning-by-synthesis for appearance-based 3d gaze estimation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 1821–1828, 2014.

[Sun et al., 2009] Jian Sun, Maks Ovsjanikov, and Leonidas J Guibas. A concise and provably informative multi-scale signature based on heat diffusion. 28(5):1383–1392, 2009.

[Sun et al., 2015] Xiao Sun, Yichen Wei, Shuang Liang, Xiaoou Tang, and Jian Sun. Cascaded hand pose regression. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[Supancic et al., 2015] James Steven Supancic, Grégory Rogez, Yi Yang, Jamie Shotton, and Deva Ramanan. Depth-based hand pose estimation: methods, data, and challenges. In *IEEE International Conference on Computer Vision, ICCV*, 2015.

[Szegedy et al., 2015] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[Tagliasacchi et al., 2015] Andrea Tagliasacchi, Matthias Schröder, Anastasia Tkach, Sofien Bouaziz, Mario Botsch, and Mark Pauly. Robust articulated-icp for real-time hand tracking. *Computer Graphics Forum (Symposium on Geometry Processing)*, 34(5), 2015.

[Tang et al., 2013] D. Tang, T. H. Yu, and T. K. Kim. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *2013 IEEE International Conference on Computer Vision*, pages 3224–3231, Dec 2013.

[Tang et al., 2017] Danhang Tang, Hyung Jin Chang, Alykhan Tejani, and Tae-Kyun Kim. Latent regression forest: Structured estimation of 3d hand poses. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(7):1374–1387, 2017.

[Tangelder and Veltkamp, 2008] Johan W H Tangelder and Remco C Veltkamp. A survey of content based 3d shape retrieval methods. *Multimedia Tools and Applications*, 39(3):441, 2008.

[Tatarchenko et al., 2016] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Multi-view 3d models from single images with a convolutional network. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VII*, pages 322–337, 2016.

[Taylor et al., 2016] Jonathan Taylor, Lucas Bordeaux, Thomas Cashman, Bob Corish, Cem Keskin, Toby Sharp, Eduardo Soto, David Sweeney, Julien Valentin, Benjamin Luff, Arran Topalian, Erroll Wood, Sameh Khamis, Pushmeet Kohli, Shahram Izadi, Richard Banks, Andrew Fitzgibbon, and Jamie Shotton. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *ACM Trans. Graph.*, 35(4):143:1–143:12, July 2016.

[Tewari et al., 2017] Ayush Tewari, Michael Zollhöfer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Pérez, and Christian Theobalt. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 3735–3744, 2017.

[Thaler et al., 2018] Anne Thaler, Michael N. Geuss, Simone C. Mölbert, Katrin E. Giel, Stephan Streuber, Javier Romero, Michael J. Black, and Betty J. Mohler. Body size estimation of self and others in females varying in BMI. *PLoS ONE*, 13(2), February 2018.

[Tkach et al., 2016] Anastasia Tkach, Mark Pauly, and Andrea Tagliasacchi. Sphere-meshes for real-time hand modeling and tracking. *ACM Transaction on Graphics (Proc. SIGGRAPH Asia)*, 2016.

[Tkach et al., 2017] Anastasia Tkach, Andrea Tagliasacchi, Edoardo Remelli, Mark Pauly, and Andrew Fitzgibbon. Online generative model personalization for hand tracking. *ACM Transaction on Graphics (Proc. SIGGRAPH Asia)*, 2017.

[Tomè et al., 2017] Denis Tomè, Chris Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. *CoRR*, abs/1701.00295, 2017.

[Tompson et al., 2014] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics*, 33, August 2014.

[Toshev and Szegedy, 2013] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. *CoRR*, abs/1312.4659, 2013.

[Toshev and Szegedy, 2014] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1653–1660, 2014.

[Varol et al., 2017] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*, Piscataway, NJ, USA, July 2017. IEEE.

[Vodopivec et al., 2016] Tadej Vodopivec, Vincent Lepetit, and Peter Peer. Fine hand segmentation using convolutional neural networks. *arXiv preprint arXiv:1608.07454*, 2016.

[Vranic et al., 2001] Dejan V Vranic, Dietmar Saupe, and J Richter. Tools for 3d-object retrieval: Karhunen-loeve transform and spherical harmonics. 2001.

[Wan et al., 2017] Chengde Wan, Thomas Probst, Luc J. Van Gool, and Angela Yao. Crossing nets: Dual generative models with a shared latent space for hand pose estimation. *CoRR*, abs/1702.03431, 2017.

[Wang et al., 2015] Fang Wang, Le Kang, and Yi Li. Sketch-based 3d shape retrieval using convolutional neural networks. *CoRR*, abs/1504.03504, 2015.

[Wei et al., 2016] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4724–4732, 2016.

[Weiss et al., 2011] Alexander Weiss, David A. Hirshberg, and Michael J. Black. Home 3d body scans from noisy image and range data. In *ICCV*, 2011.

[White et al., 2007] Ryan White, Keenan Crane, and David Forsyth. Capturing and animating occluded cloth. In *ACM Transactions on Graphics (SIGGRAPH)*, 2007.

[Wu et al., 2015] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[Wuhrer et al., 2014] Stefanie Wuhrer, Leonid Pishchulin, Alan Brunton, Chang Shu, and Jochen Lang. Estimation of human body shape and posture under clothing. *CVIU*, 2014.

[Xi et al., 2007] Pengcheng Xi, Won-Sook Lee, and Chang Shu. A data-driven approach to human-body cloning using a segmented body database. In *Pacific Graphics*, 2007.

[Xie et al., 2016] Jin Xie, Meng Wang, and Yi Fang. Learned binary spectral shape descriptor for 3d shape correspondence. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[Xu et al., 2011] Feng Xu, Yebin Liu, Carsten Stoll, James Tompkin, Gaurav Bharaj, Qionghai Dai, Hans-Peter Seidel, Jan Kautz, and Christian Theobalt. Video-based characters: Creating new human performances from a multi-view video database. In *SIGGRAPH*, 2011.

[Xu et al., 2016] Chi Xu, Ashwin Nanjappa, Xiaowei Zhang, and Li Cheng. Estimate hand poses efficiently from single depth images. *Int. J. Comput. Vision*, 116(1):21–45, January 2016.

[Yamaguchi et al., 2013] Kota Yamaguchi, M. Hadi Kiapour, and Tamara L. Berg. Paper doll parsing: Retrieving similar styles to parse clothing items. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[Yan and Mikolajczyk, 2015] Fei Yan and Krystian Mikolajczyk. Deep correlation for matching images and text. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[Yang et al., 2014] Yipin Yang, Yao Yu, Yu Zhou, Sidan Du, James Davis, and Ruigang Yang. Semantic parametric reshaping of human body models. In *3DV*, 2014.

[Yang et al., 2016] Shan Yang, Tanya Ambert, Zherong Pan, Ke Wang, Licheng Yu, Tamara L. Berg, and Ming C. Lin. Detailed garment recovery from a single-view image. *CoRR*, abs/1608.01250, 2016.

[Ye and Yang, 2014] Mao Ye and Ruigang Yang. Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera. In *CVPR*, 2014.

[Ye et al., 2013] Mao Ye, Qing Zhang, Liang Wang, Jiejie Zhu, Ruigang Yang, and Juergen Gall. *A Survey on Human Motion Analysis from Depth Data*, pages 149–187. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[Ye et al., 2016] Qi Ye, Shanxin Yuan, and Tae-Kyun Kim. Spatial attention deep net with partial PSO for hierarchical hybrid hand pose estimation. *CoRR*, abs/1604.03334, 2016.

[Yi et al., 2016] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. LIFT: learned invariant feature transform. *CoRR*, abs/1603.09114, 2016.

[Yuan et al., 2017a] Shanxin Yuan, Qi Ye, Björn Stenger, Siddhand Jain, and Tae-Kyun Kim. Bighand2.2m benchmark: Hand pose dataset and state of the art analysis. *CoRR*, abs/1704.02612, 2017.

[Yuan et al., 2017b] Shanxin Yuan, Qi Ye, Björn Stenger, Siddhand Jain, and Tae-Kyun Kim. Bighand2.2m benchmark: Hand pose dataset and state of the art analysis. *CoRR*, abs/1704.02612, 2017.

[Zeiler and Fergus, 2014] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, pages 818–833, 2014.

[Zeiler, 2012] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[Zhang et al., 2015] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Appearance-based gaze estimation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 4511–4520, 2015.

[Zhang et al., 2016] Jiawei Zhang, Jianbo Jiao, Mingliang Chen, Liangqiong Qu, Xiaobin Xu, and Qingxiong Yang. 3d hand pose tracking and estimation using stereo matching. *CoRR*, abs/1610.07214, 2016.

[Zhang et al., 2017] Chao Zhang, Sergi Pujades, Michael Black, and Gerard Pons-Moll. Detailed, accurate, human shape estimation from clothed 3D scan sequences. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*, Piscataway, NJ, USA, July 2017. IEEE. Spotlight.

[Zhao et al., 2016] Ruiqi Zhao, Yan Wang, and Aleix M. Martínez. A simple, fast and highly-accurate algorithm to recover 3d shape from 2d landmarks on a single image. *CoRR*, abs/1609.09058, 2016.

[Zhou et al., 2010] Shizhe Zhou, Hongbo Fu, Ligang Liu, Daniel Cohen-Or, and Xiaoguang Han. Parametric reshaping of human bodies in images. *ACM Trans. Graph.*, 29(4), 2010.

[Zhou et al., 2013] Bin Zhou, Xiaowu Chen, Qiang Fu, Kan Guo, and Ping Tan. Garment modeling from a single image. *Pacific Graphics*, 2013.

[Zhou et al., 2016] Xingyi Zhou, Qingfu Wan, Wei Zhang, Xiangyang Xue, and Yichen Wei. Model-based deep hand pose estimation. In *IJCAI*, 2016.

[Zimmermann and Brox, 2017] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single RGB images. *CoRR*, abs/1705.01389, 2017.