Virtual Human Bodies with Clothing and Hair: From Images to Animation

by

Peng Guan

B. S., Fudan University, 2005

Sc. M., Fudan University, 2008

Sc. M., Brown University, 2010

A Dissertation submitted in partial fulfillment of the
requirements for the Degree of Doctor of Philosophy
in the Department of Computer Science at Brown University

Providence, Rhode Island

May 2013

This dissertation by Peng Guan is accepted in its present form
by the Department of Computer Science as satisfying the
dissertation requirement for the degree of Doctor of Philosophy.

Date _____ _____

Michael J. Black, Director

Recommended to the Graduate Council

Date _____ _____

Leonid Sigal, Reader

Disney Research, Pittsburgh

Date _____ _____

Gabriel Taubin, Reader

Division of Engineering, Brown University

Approved by the Graduate Council

Date _____ _____

Peter M. Weber, Dean of the Graduate School

Vitæ

Peng Guan was born on April 3, 1983 in Beijing, China. He is a Ph.D. candidate under the supervision of Michael J. Black at Brown University; he received his B.Sc. degree in Electronic Engineering from Fudan University (2005), his M.Sc. in Electronic Engineering from Fudan University (2008), and his M.Sc. in Computer Science from Brown University (2010). He was awarded "Honored Graduate of Shanghai" in 2005. From Dec 2007 to Apr 2008, he worked as a software development intern at Sony, where he analyzed the bit streams of MPEG4 TV program. From Jun 2011 to Aug 2011, he worked as a research intern at Disney Research Pittsburgh, where he developed multi-linear dynamical hair model for efficient 3D hair animation. Peng Guan's research interests are primarily in computer vision and computer graphics with the focus on data-driven models for 3D human bodies, clothing, and hair. His publications include:

(1) "Multi-linear Data-Driven Dynamic Hair Model with Efficient Hair-Body Collision Handling", Guan, P., Sigal, L., Reznitskaya, V., and Hodgins, J. ACM/Eurographics Symposium on Computer Animation (SCA), 2012.

(2) "DRAPE: DRessing Any PErson", Guan, P., Reiss, L., Hirshberg, D.A., Weiss, A., and Black, M.J. ACM SIGGRAPH, 2012.

(3) "A 2D human body model dressed in eigen clothing", Guan, P., Freifeld, O., and Black, M.J. European Conf. on Computer Vision, ECCV, 2010.

(4) "Estimating human shape and pose from a single image", Guan, P., Weiss, A., Balan, A.O., and Black, M.J. Int. Conf. on Computer Vision, ICCV, 2009.

(5) "3D face recognition based on facial structural angle and local region map", Guan, P., and Zhang, L. Int. Conf. on Multimedia & Expo, ICME, 2008.

(6) "A novel facial feature point localization method on 3D faces", Guan, P., Yu, Y., and Zhang, L. Int. Conf. on Image Processing, ICIP, 2007.

(7) "Discriminant analysis with label constrained graph partition", Guan, P., Yu, Y., and Zhang, L. Int. Symposium on Neural Networks, ISNN, 2007.

(8) "Three dimensional facial feature point localization method based on Bezier surface", Guan, P., and Zhang, L. Journal of Fudan University (in Chinese), Vol 47, No.1, Feb 2008, pp 117-123.

(9) “An automatic railroad detection approach based on image processing", Guan, P., Gu, X., and Zhang, L. Journal of Computer Engineering (in Chinese), Vol 33, No.19, Oct 2007, pp 207-209.

(10) “Extensions of manifold learning algorithms in kernel feature space", Yu, Y., Guan, P., Zhang, L. Int. Symposium on Neural Networks, ISNN, 2007.

Dedicated to my family

Acknowledgements

In the past four years, I have experienced the biggest intellectual and mental transformations by far in my life. I deeply appreciate the help from the people around me and would like to thank them for their support, guidance, and forgiveness of my ignorance.

First and foremost, I want to thank my advisor Michael J. Black who has brought me to Brown, accompanied me to overcome the difficulties along the way, and encouraged me to pursue perfection. Before I came to Brown, Michael told me that the keys to success as a PhD are perseverance, passion, and creativity. On that day, I set them as my goals and I have been striving to achieve them without hesitation. I have experienced countless nights working on my projects, the anxiety before the paper deadlines, and the frustration after the paper being rejected. Whenever I am overwhelmed or need help, Michael is the person who came along and directed me to handle these. The past four years have been the test of my perseverance, passion, and creativity and I am grateful to see the transformations that happen on me. I become a better researcher, critical thinker, and most importantly I am more confident of what I do and what I will do. Michael not only taught me how to identify, formulate, and solve problems, but also instructed me on writing papers, giving good presentations, and impressing the audience. I have no doubt that the skills I have learned from him will stay with me throughout my life and help to maximize the chance of success in my career. Special thanks go to Michael for supporting me during my job searching and career decision. My gratitude towards Michael is beyond words.

I am grateful to my thesis committee: Gabriel Taubin (Brown University) and Leonid Sigal (Disney Research, Pittsburgh). This thesis would not be possible without their critical questions and suggestions particularly during the thesis proposal. I also would like to thank faculty members Erik Sudderth, James Hayes, Tom Doeppner, John Hughes, and many more for constructive suggestions and discussions on various problems.

Leonid Sigal is not only on my thesis committee but also my mentor and friend at Disney Research, Pittsburgh where I spent a wonderful three months in the summer 2011. I am extremely excited about the hair animation project we were working on, and am deeply impressed by Leonid's key insights, deep understanding of the problems, and the broad range of expertise. Leonid is a

delight person to work with and I am sure we will have more interactions and collaborations in the future.

I would like to thank my coauthors and collaborators : Loretta Reiss, Alexander Weiss, Alexandru O. Balan, Deqing Sun, Oren Freifeld, Sivia Zuffi, Matthew Loper, Eric Rachlin, David Hirshberg, Valeria Reznitskaya, Moshe Mahler, Jessica Hodgins and many more for supporting the projects and providing valuable feedback. Among these people, I will specifically thank Loretta Reiss and Alexandru O. Balan. DRAPE is the most important project during my PhD and we spent one and a half year on it. This project would not be as successful as it is now without Loretta's effort on designing the clothing pattern and dressing virtual characters in the simulation software. Loretta is always very helpful and supportive, which made the project going very smoothly and made me focusing on solving technical problems. In some sense, Loretta speeds up my graduation :). Alexandru O. Balan introduced the SCAPE model to me and helped me in a number of ways. It is always difficult to get started in a new environment and make yourself accustomed to the infrastructure and resources. Alexandru is the person who guided me on research and the usage of other department facilities when I first came to Brown in 2008. We also collaborated on ICCV 09 paper about estimating body shape from a single image. His brilliant "height preserving body shape space" is extremely important and valuable to my research.

The excellent courses in Brown computer science department have shaped my knowledge base and trained me to be a well-rounded computer scientist. Brown CS also has the most impressive crew of technical and administrative staffs that I have ever seen. I definitely appreciate the quick responses from technical staffs to solve my computer-related issues. It makes me better utilize my time on research. I would like to thank my fellow graduate students to provide mutual support for each other. We had good time sharing research, having free food at TGIF, and playing foosball at the library. Those sweet memories will be rooted in my heart.

Last but not least, I am grateful to my wife and my family. My wife Jingxin Feng and I have been together for ten years and she is the person who understands me, cares me, and loves me the most. I am indebted to her throughout my life. I am not sure if I can achieve what I have achieved without her whole-hearted support and encouragement. She is the person who takes care of the family; who gives me strength to take on any challenges that are ahead of us; and who always stands by me and makes everything in my life simpler.

Contents

List of Tables

List of Figures

Abstract of "Virtual Human Bodies with Clothing and Hair: From Images to Animation"

by Peng Guan, Ph.D, Brown University, May 2013

Realistic clothing and hair animation are necessary for many applications such as special effects, gaming, and on-line fashion. Thanks to the advances in computer graphics, highly realistic clothing and hair animation is common in recent animated movies by use of Physics Based Simulation (PBS). Clothing and hair modeling/animation are also important to other fields such as computer vision. However, PBS methods create clothing/hair that are specific to a particular body model and there is no good way to invert the PBS process to fit the parameters of the generative model to images. Furthermore, PBS for clothing requires a significant amount of manual work to find the right size for every instance of a body and prepare the clothing for simulation. This makes PBS unsuitable for applications that involve various body shapes, such as virtual fashion and crowd simulation. We believe a new generative clothing model will address these problems.

In this thesis, we describe a complete process from estimating 3D human body form using image evidence to animating the body with data-driven, low-dimensional 3D clothing and hair models. First, we describe a solution to estimating 3D human body shape from a single photograph or painting using multiple image cues including silhouettes, edges, and smooth shading. Second, we explore a 2D clothing model in which the clothing is modeled as a closed contour of points that are offsets from corresponding body contour points. We show the increased accuracy of 2D body shape estimation and clothing type classification using such a model. Third, we focus on modeling the appearance of the 3D body and propose a complete system for simulating realistic clothing on 3D bodies of any shape and pose without manual intervention. Fourth, we also present a 3D hair model that performs hair animation in real-time, preserves the key dynamic properties of physical simulation, and gives the animator continuous interactive control over hair styles (e.g. length and softness) and external phenomena (e.g. wind).

The result is a 3D human body model that can be estimated from images and then animated with realistic clothing, hair, and body movement.

CHAPTER 1

Introduction

1. Thesis Statement

Machine learning can be used to construct data-driven 3D human bodies, clothing, and hair that 1) can be estimated from sensor data; 2) produce realistic animations; 3) are low-dimensional enough to be computationally practical; 4) may be applied to broader computer vision tasks or real-time applications.

2. Introduction

Virtual human bodies, clothing, and hair are widely used in a number of scenarios such as 3D animated movies, gaming, and online fashion. In computer vision, a variety of human body models with different levels of specificity have been utilized to improve the performance of human action analysis, human motion capture, and body shape & pose estimation from images or videos. In the entertainment industry, animators use computer graphics techniques to generate vivid 3D virtual characters and put them in virtual contents such as animated movies and games. Virtual bodies may also be used in medical applications where the body shape and weight of the patients can be tracked over time. In augmented reality, virtual people are overlayed on the real visual contents to provide new user experiences. Besides human body modeling, how to realistically represent clothing and hair is also on the top of the research agenda. The recent animated movie “Brave" (by Disney Animation Studio, 2012) demonstrates state of the art clothing and hair simulation for animated movies. Games typically have a very tight time budget for clothing and hair simulation. However, there is no doubt that the realism of clothing and hairs are among the key indicators of high quality visual effects.

The research on human body shapes and clothing/hair modeling are closely related in a number of applications even though they have been largely treated as independent research topics in the vision and graphics communities. Human beings come in a variety of body shapes. Any *virtual try on* applications are meaningful when they have the real body shapes as input, which can be directly provided by users or estimated from various kinds of evidence. Body shape and pose estimation will be much more accurate when clothing is modeled because clothing obscures body shape. Unfortunately, the effect of clothing is ignored in most of recent work on body shape estimation. Online

FIGURE 1.1. **Physics based simulation for clothing.** Figures are obtained from a recent paper **[50]** ("Efficient Simulation of Inextensible Cloth" in SIGGRAPH 2007.) The cloth contains 8325 (top) and 10688 (bottom) vertices, with average simulation time per frame of 5.2 and 7.8 seconds, respectively.

apparel shopping is another promising application, which involves clothing models that can adapt to different body shapes. Real-time hair models are essential for gaming and accurate body height estimation in forensics.

The standard techniques for clothing and hair modeling employ physics based simulation (PBS) **[29, 62, 80, 112, 15, 123]**, which has the advantage of producing realistic results with typically

FIGURE 1.2. **Physics based simulation for hair.** Renderings are generated using GeForce GTX 480 with hardware tessellation engine. Over 18,000 hairs are interpolated from a few hundred hair strand guides. (Game Developer Conference 2010, youtube link: `http://www.youtube.com/watch?v=YF8CUSiPDJ0`.)

high computational cost. See Figure 1.1 and 1.2. Furthermore, the results are specific to a particular body model. Each character requires a new simulation with typically manual initialization. Hair animation faces similar challenges, except that body pose/motion instead of intrinsic body shape plays a much more important role. These limitations make PBS suitable to animated movies that have an abundant time budget and a limited number of characters, but not for applications such as internet-scale virtual fashion or retail clothing try-on.

We explore clothing and hair models that have the following properties: 1) They should look realistic. They may not have the same level of realism as PBS, but they need to maintain fine details, such as the wrinkles and folding for clothing and plausible dynamics for both clothing and hair. 2) They should be low-dimensional for computational efficiency, which usually implies a parametric model such that the appearance is determined by a relatively small set of parameters. 3) They have the potential to be used in a broader range of applications. For instance, the clothing model should be generic so that it adapts to different body shapes. The hair model should allow real-time simulation for gaming, and preferably gives users interactive hair style control for virtual hair try on.

We start by providing solutions to 3D body shape estimation (with minimal clothing) from a single image (Figure 1.3). We then focus on a simple 2D contour model of clothing that facilitates the

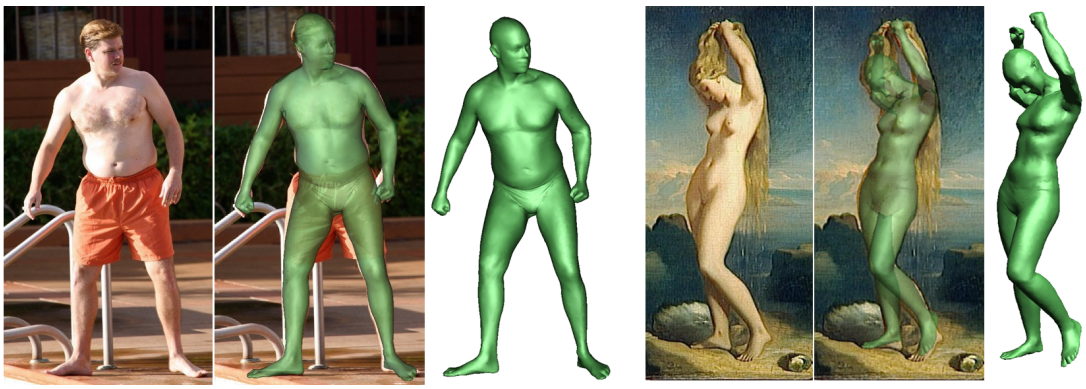


FIGURE 1.3. **3D body shape estimation from a single image or painting.** We estimate the 3D body shape and pose from a single, un-calibrated image. The result is a posable 3D body model.

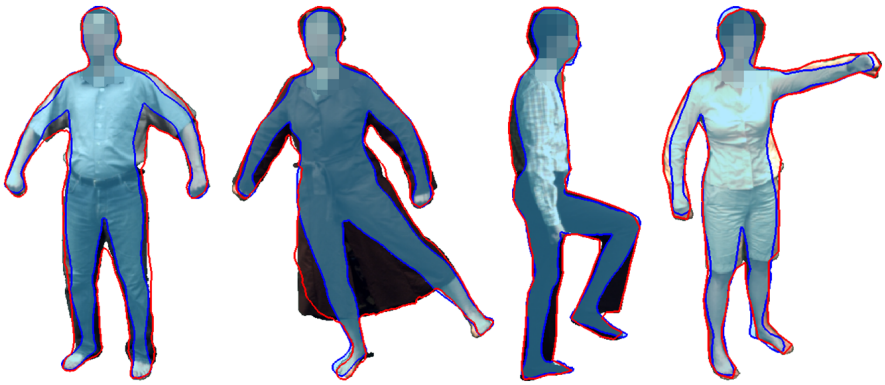


FIGURE 1.4. **2D body shape under clothing.** We build a 2D eigen clothing model to accurately estimate the body shape under clothing. The blue contour represents estimated body shape. The red contour is the clothing outline.

2D body shape estimation under clothing (Figure 1.4). We further build a 3D data-driven clothing model that produces realistic clothing efficiently and adapts to different body shapes (Figure 1.5). This model is fully automatic at run time and it allows clothing animation on 3D bodies of different shapes. Finally, a real-time data-driven hair model is also presented to animate with the 3D body (Figure 1.6). Such models are useful to computer graphics as well as computer vision.

2.1. Human Body Shape and Pose Estimation. There are a large number of research articles that aim to infer 2D or 3D *human body shape and pose* from regular or depth images [**42, 61, 63, 101, 78, 60, 95, 98, 126, 116, 5, 54, 51**]. The estimation results are important for human activity recognition, human intention reasoning, forensics, online apparel shopping, etc. The human body is commonly modeled as a kinematic tree (similar to a puppet) rooted at the pelvis. Body parts such as limbs and the torso are connected through joints. The goal of pose estimation

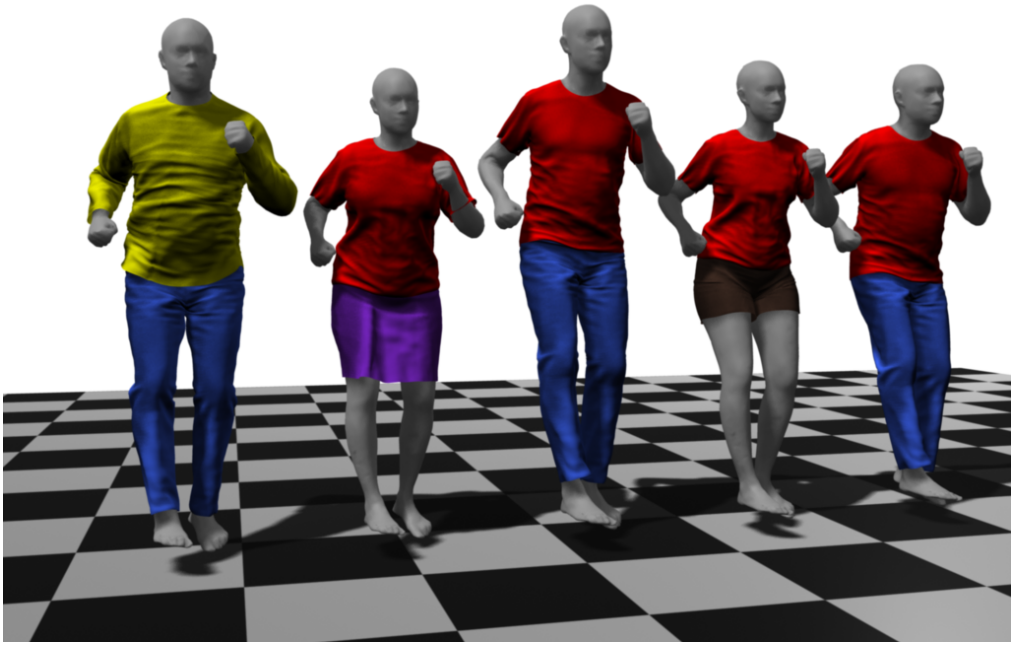


FIGURE 1.5. **Dressing people in different body shape.** Our "DRPAE" model not only produces realistic clothing, but also adapts to different body shapes.

is to infer a set of relative joint rotations that determine the positioning of body limbs and parts in the images. *Human Shape* typically refers to intrinsic pose-independent shape such as height, weight, chest size, waist size and so on. A small set of coefficients for a parametric 3D body model is conveniently used to represent the human shape. We can obtain different human shapes by varying the coefficients.

In this thesis, we provide a solution to 3D shape and pose estimation under the most challenging situation where only a *single* image is available and the image is captured in a natural environment with unknown camera calibration. We also demonstrate that a simplified 2D clothing model helps to increase the accuracy of 2D body shape estimation significantly.

2.2. Clothing Animation. Clothing animation is important for all kinds of digital applications that involve dressed virtual characters. Because of the complex nature of cloth dynamics, many applications (especially in games) rely on texture mapping on the body geometry or very coarse triangulated cloth meshes. On the other hand, clothing simulation in animated movies typically has very high quality because of the abundant computational resources. We focus on developing a data-driven clothing model that is realistic enough to present fine details and wrinkles, but is also efficient, low-dimensional, and can adapt to different body shapes [**52**]. Given the estimated 3D body model

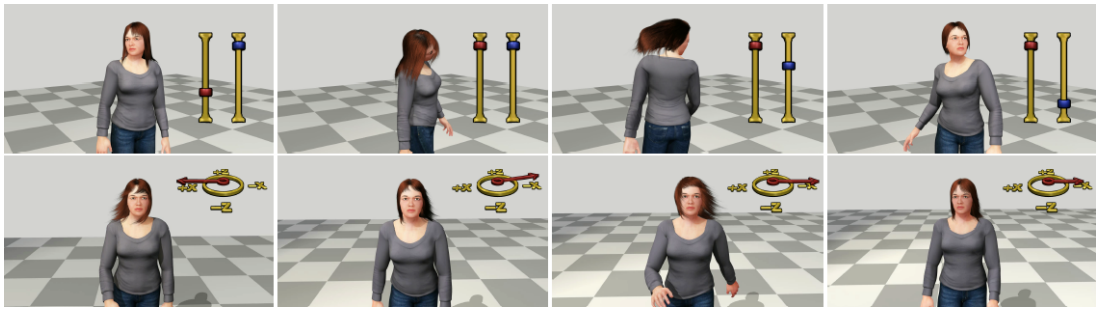


FIGURE 1.6. **Multi-linear dynamic hair model.** Our multi-linear dynamic hair model allows real-time animation of 900 guides of hair with interactive control over the hair softness (red slider, the higher the softer) and length (blue slider, the higher the longer); bottom row shows interactive control of wind strength and direction.

regardless of its intrinsic shape, we are able to properly fit the estimated body with our clothing model. We envision such models to be appropriate for many other computer vision applications as well.

2.3. Hair Animation. Hair animation is a difficult task, primarily due to the large volume of hairs that need to be considered (a typical human head consists 100,000 hair strands). We avoid dealing with every single hair strand by building a low-dimensional model that 1) preserves the key dynamic properties of physical simulation at a fraction of the computational cost, 2) allows user specifiable hair styles (length, softness) and external phenomena (wind) **[53]**. We envision this model to be used in real-time applications such as gaming.

3. Challenges

Body Shape and Pose Estimation from a Single Images. An image of a human body in a natural environment arises from the composite effect of numerous factors including lighting, occlusion, human pose and shape, camera view, clothing, and so forth. *Body shape and pose estimation from images* is very difficult mainly because there are so many uncertainties that will affect what is perceived. We briefly review the challenges of body form estimation from a single image. **Image Capture.** An image is a projection of 3D world to the 2D image plane, during which the depth information is lost and not directly recoverable with only one image. For instance, it is hard to tell whether a person has a flat or bulging belly if he/she is directly facing the camera view. **Occlusion.** External occlusion or self-occlusion (i.e the limb is occluding the torso) can occur in the image, which severely hurts pose estimation. **Lighting.** Lighting is generally considered to have a negative impact because changes in illumination cause changes in how an object appears in the image. However, we show that exactly this property of appearance variation with lighting provides

additional cues for shape estimation. **Background Unknown.** Without a known background, it is difficult to extract an accurate foreground silhouette which is often used for pose and shape estimation. **Clothing.** Clothing obscures body shape. There is an ambiguity between a thin person dressed in loose clothing and a fat person dressed in tight clothing. We show that a 2D clothing model helps to reduce such ambiguity and achieve much better body shape estimation results.

Clothing Animation. The traditional PBS methods for clothing animation have the following challenges. **Computation.** They achieve realistic simulations at high computational cost, which is why they are often used in the off-line scenarios such as movie making. Note that, there exist hybrid methods that use PBS to simulate a low resolution mesh and use data-driven methods to learn the mapping between simulated coarse meshes and highly wrinkled detailed meshes from training set [**111, 70**]. These methods can achieve real-time performance, but their results are not comparable to pure PBS methods. **Manual Treatment.** Choi et al. [**29**] summarize the fundamental problem confronting garment designers to be the "nontrivial task of choosing clothing sizes and initializing clothing simulation on 3D characters". Both of them are currently done manually. **Adaptation to New Body Shapes.** Each distinct body shape requires a separate simulation, which involves a significant amount of manual work, including manually choosing the appropriate cloth size for each character and placing the cloth at proper initial positions. This makes it inappropriate for internet-scale clothing simulation in applications such as online fashion.

Hair Animation. The major challenge of hair animation is to develop a compact, computationally efficient model, that is at the same time expressive enough to convey the dynamic behaviors seen in high-resolution simulations. Current methods are either too slow for real-time applications or too coarse to represent complex hair dynamics.

4. Contribution of the Thesis

In this thesis, we provide a complete pipeline from getting the 3D body shape and pose from image evidence to animating the body with clothing and hair. This makes internet-scale customized clothing animation possible. We also show the potential of modeling clothing in computer vision tasks.

(1) We describe a solution to the challenging problem of estimating human body shape from a single photograph or painting. Our approach computes shape and pose parameters of a parametric 3D human body model directly from multiple monocular image cues including silhouette, edges, and smooth shading. One of the key contributions is the formulation of *parametric human shape from shading*. We estimate the body pose, shape and reflectance as well as the scene lighting that produces a synthesized body that robustly matches the image evidence. To deal with ambiguity in

a monocular image, we learn a low-dimensional linear model of human shape in which variations due to height are concentrated along a single dimension, enabling height-constrained estimation of body shape.

(2) We propose a fully generative 2D eigen clothing model that is based on an underlying naked model with clothing deformation. This model significantly improves the inference of 2D body shape under clothing. Clothing deformation from the body is one-directional (clothing only makes the contour larger), therefore we model the skewed statistics of the eigen-clothing coefficients. This work is also the first to address the shape-based recognition of clothing categories on dressed humans. The preliminary work shows the potential of modeling clothing in computer vision applications.

(3) We propose a 3D clothing model that is able to automatically dress synthetic bodies of any shape in any pose at run time. It provides a factored model of clothing shape so that pose-dependent wrinkles are modeled separately from body shape. Interpenetration is efficiently handled by solving a linear system of equations and this approach is significantly faster than physical simulation. The method is ideal for applications where the body shape is not known in advance such as retail clothing applications where users create different 3D bodies or estimate 3D body shapes. It is also useful for animating many bodies of different shapes because it removes the labor involved in either creating or finding the appropriately fitting garment.

(4) We introduce a multi-linear reduced-space dynamical model for modeling hair. It is explicitly parameterized by a number of real-valued factors (e.g., hair length, hair softness, wind direction/strength, etc.) that make it easy to adjust the groom and motion of hair interactively at test time. We formulate our model using tensor algebra and illustrate how dynamics can be incorporated within this framework. Furthermore, we explicitly address the issue of hair-body collisions by a very efficient optimization procedure formulated directly in the reduced space and solved using a form of iterative least squares. Our formulation goes substantially beyond current reduced-space dynamical models (e.g., [**37**]); in fact, [**37**] can be interpreted of as a special case of our model.

5. Thesis Outline

Chapter 1. Introduction. Thesis statement, background, challenges, and contributions.

Chapter 2. State of the art. We briefly review the state of the art of human body models, human shape and pose estimation, clothing animation, and hair animation.

Chapter 3. Estimating 3D Body Shape and Pose from a Single Image. We describe the solution to estimation 3D human body shape and pose from a single image.

Chapter 4. 2D Eigen-Clothing Model. We propose a 2D eigen-clothing model that improves the accuracy of 2D body shape estimation under clothing.

Chapter 5. DRAPE : DRessing Any PErson. In this chapter, we “dress" the estimated 3D human body with synthetic clothing. The “DRAPE" model generate realistic clothing meshes that automatically adapt to different body shapes.

Chapter 6. Multi-linear Dynamic Hair Model. We model hair guides in the reduced space and use a multi-linear model to interactively control the hair appearances such as hair length, softness, and wind directions/strengh. The model runs in real-time.

Chapter 7. Conclusions and Future Work. We summarize the contributions of the thesis as well as future research directions.

6. List of Published Papers

The content of this thesis is built upon the following original publications.

- Peng Guan, Alexander Weiss, Alexandru O. Balan and Michael J. Black. “Estimating human shape and pose from a single image." IEEE International Conference on Computer Vision (ICCV), pages 1381–1388, 2009.
- Peng Guan, Oren Freifeld and Michael J. Black. “A 2D human body model dressed in eigen clothing." IEEE European Conference on Computer Vision (ECCV), pages 285–298, 2010.
- Peng Guan, Loretta Reiss, David Hirshberg, Alexander Weiss and Michael J.Black. “DRAPE : DRessing Any PErson." ACM Trans. Graph. (Proc. SIGGRAPH), 31(4) pages 35 :1–35 :10, July 2012..
- Peng Guan, Leonid Sigal, Valeria Reznitskaya and Jessica Hodgins. “Multi-linear Dynamic Hair Model with Efficient Collision Handling." ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA), 2012.

CHAPTER 2

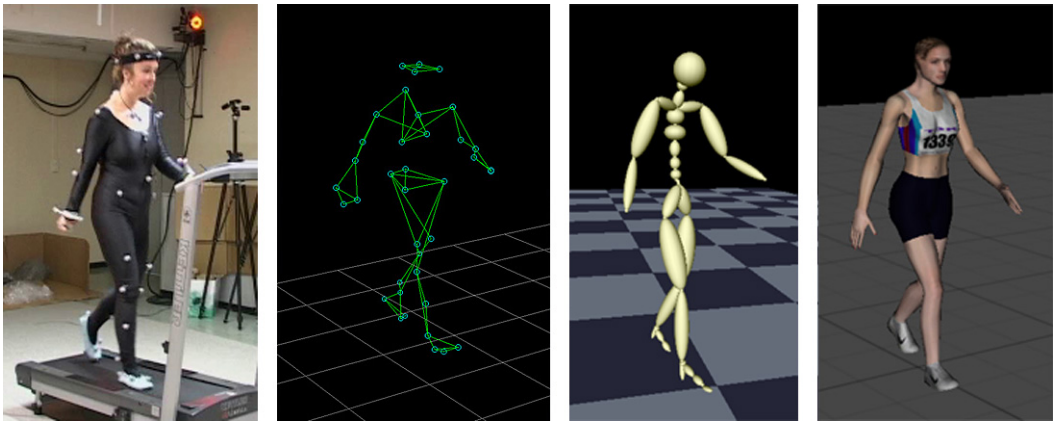
State of the Art: A Brief Review

In the first part of this chapter, we review two kinds of methods for capturing 3D human pose and shape: 1) marker based active sensing and 2) marker-less capturing. Then we will focus on marker-less capturing techniques and introduce a wide variety of 3D body models that make marker-less methods possible. The second part of the chapter focuses on providing the state of the art for animating the 3D body with realistic clothing and hairs.

1. Motion Capture and Human Shape

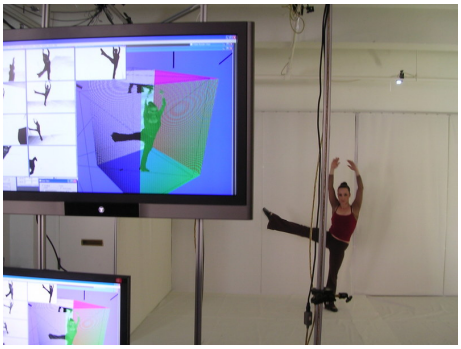
"Motion capture is the process of recording people's movement. It has proved very useful in entertainment, sports, medicine, computer vision, and robotics." [**118**]. In the movie industry, human motion is captured and transferred to cartoon or monster characters. A successful motion capture system can save animators an enormous amount of time, because they do not need to manually specify the key frame motion of the body. Similar techniques include "facial expression retargeting" [**119**] where the facial expressions on real people are captured and are transferred to virtual characters. In sports, motion capture is used to extract the motions of professional athlete for training purposes or for creating vivid motions in the sports games.

1.1. Marker Based Active Sensing. Most successful commercial motion capture systems (MoCap) use markers of some form (e.g. Vicon, Motion Analysis, Meta Motion, Qualisys). See Figure 2.1. These systems primarily focus on capturing human pose. By attaching active (electromagnetic, accelerometers) or passive (reflective markers) devices to the human body parts, human motions can be captured with minimal error by recording the 3D location and movement of the markers. Marker based active sensing has two important advantages: 1) Markers have their own unique digital signatures so that the markers are instantly and uniquely recognized the moment a camera sees them. Because the correspondences are obtained trivially, it requires minimal post-processing and manual effort. 2) Active sensing MoCap can provide sub-millimeter precision at high frame-rate which makes it suitable for military, medical, and sports applications. The downside of marker based Mocap is that it typically happens in a controlled environment; the subject needs to be fully cooperative to wear the cumbersome devices; and the system is expensive. That is why marker-less motion capture is preferred under many circumstances because it requires less cooperation from



(a) Marker-based Motion Capture

(b) Motion Capture in Avatar (2009)

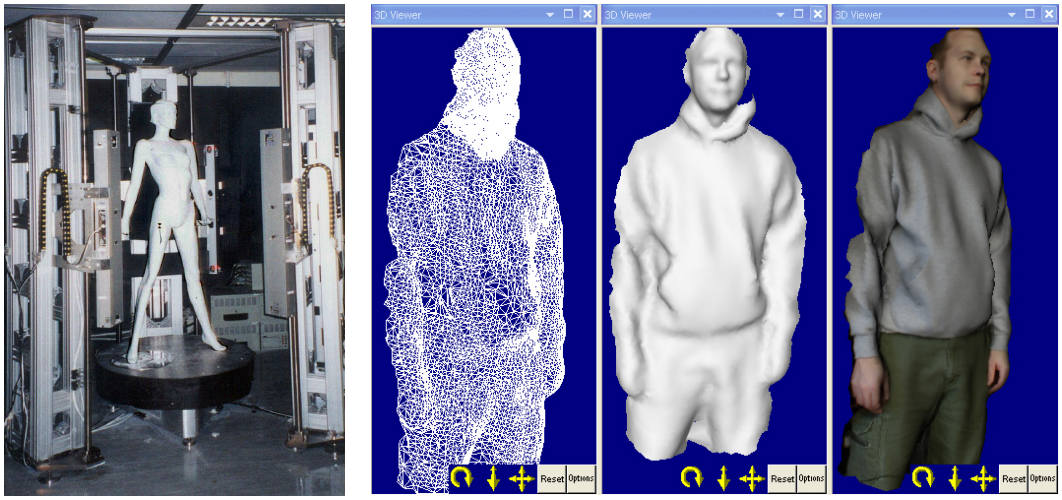


(c) Marker-less Motion Capture

FIGURE 2.1. **Motion Capture Technologies.** (a) The process of motion capture. Human motion can be extracted by placing markers on the body. The motion is then transferred to virtual characters. (`http://lukebeech.wordpress.com/motion-capture/`) (b) Motion capture was used in the movie Avatar (2009). (`http://lukebeech.wordpress.com/motion-capture/`) (c) Marker-less motion capture techniques by Organic Motion. (`http://www.organicmotion.com/`)

the subjects and the environment is less constrained. However, marker-less motion capture (e.g. Organic Motion) is significantly more difficult because we need to deal with noisy information in one or more images.

1.2. Shape Capture. Active 3D scanners (NextEngine, LaserDesign, etc) are commonly used to capture the shape of objects including human bodies. They emit radiation (laser rays, structured light pattern) and collect the depth information of the surfaces of interest. Then, a 3D point cloud, corresponding to 2D points in the image, can be reconstructed, and therefore the shape can be inferred. Just like normal cameras, the shape can only be captured if it is not obscured in the view.

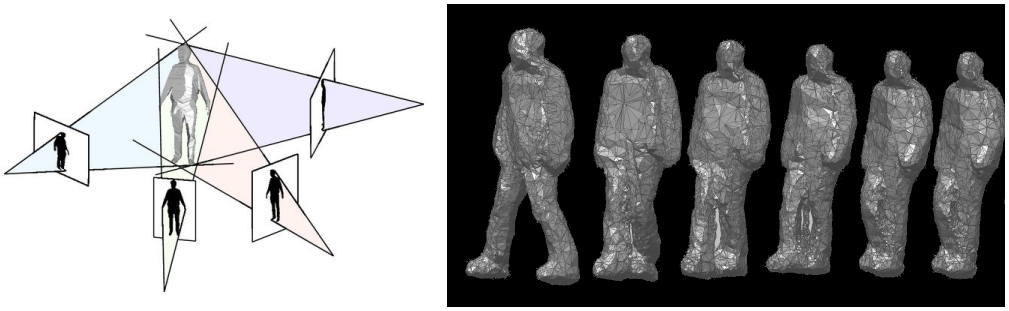


(a) Full Body Scanner (b) Body Scan with Texture

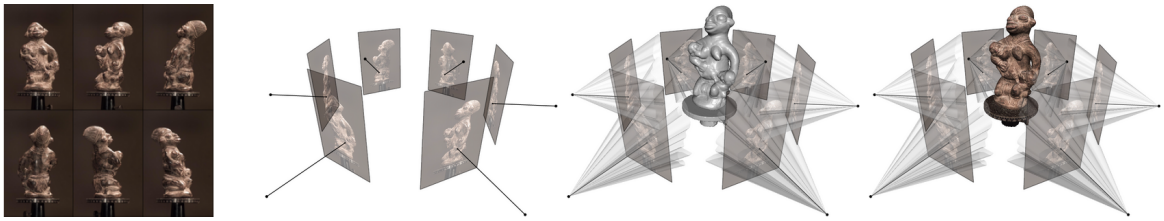
FIGURE 2.2. **3D Body Scans.** (a) A snapshot of a full body scanner. (`http://www.dmsf.ust.hk/fullbodyscanner/fullbodyscanner.htm`) (b) Mesh reconstructed from a partial view with texture map. (`http://www.photomodeler.com`)

This means that a full scan of an object typically requires multiple 3D scanners, with each covering a partial view. 3D scanners first produce a 3D point cloud and these points are triangulated to produce a 3D mesh. Even though 3D scanners are widely used in material production, industrial engineering, and entertainment applications, they do have some weaknesses. 1) Scanning an object in motion is very difficult because the frame rate is not high enough (typically 12 seconds per scan). 2) The cost of these systems are typically high. 3) The scanners do not provide temporal surface point correspondences for the meshes, which makes shape editing, texture-mapping, and shape deformation difficult.

1.3. Vision-based Techniques. In computer vision, people use different ways to infer object shape. It is called *Shape from X*, where *X* can be silhouettes, photometric stereo, depth, motion, shading [**124**] etc. *Shape from Silhouettes* was original introduced by Laurentini [**74**]. Given multiple calibrated cameras, the idea is to maximize the 3D volume of the object such that the projection of the 3D volume to each camera view completely falls inside the foreground silhouette of that view. The 3D volume can be obtained by voxel carving or intersection of generalized cones. To obtain an accurate reconstruction, a lot of calibrated cameras are needed (typically more than 16). *Shape from Stereo* algorithms use texture to establish pixel correspondences between multiple camera views. Suppose the camera calibration and pixel correspondences are known, the 3D location of a point can be computed through “triangulation". However, accurate detection of pixel correspondences is



(a) Shape from Silhouettes

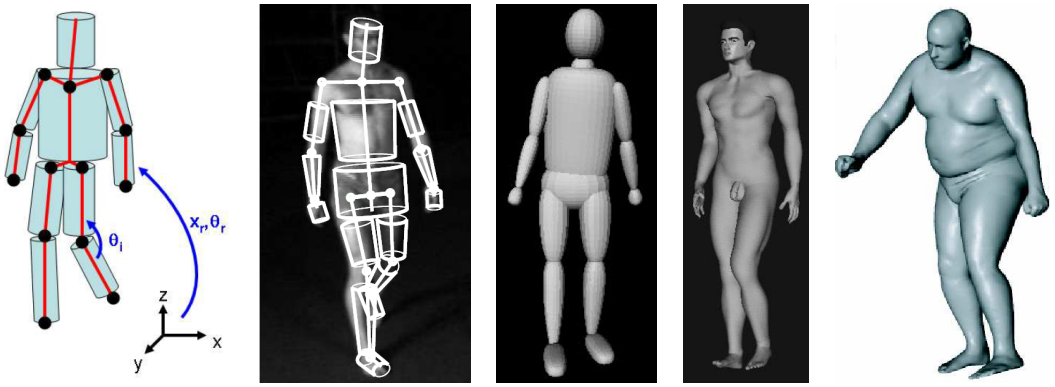


(b) Shape from Stereo

FIGURE 2.3. **Shape from Silhouette and Stereo.** (a) The results of visual hull body reconstruction for a sequence of frames. (`http://www.cgal.org/UserWorkshop/`) (b) Shape reconstruction from stereo images. (`http://carlos-hernandez.org`)

still an active research problem and post-processing is required to fill holes in the mesh. *Shape from Motion* or *Structure from Motion* is similar to *Shape from Stereo* except that the camera is moving and the camera calibration information has to be inferred from the corresponding pixels in different frames. With the advent of Kinect technologies, *Shape from Depth* is becoming cheaper and more flexible. The recent work Kinect Fusion [**64**] is a very good example of combining the depth maps of a moving camera to reconstruct a complex 3D scene. Kinect Fusion is similar to *Structure from Motion* techniques except that it replaces a regular camera with depth camera.

1.4. Model-based Approaches. Once we know the object of interest (e.g. human body), another commonly used approach is to build a parametric *prior* model of it and use it to constrain the reconstruction result. The problem is formulated as an optimization of an objective function over the model parameters [**10, 9, 7, 57, 54**]. Such objective functions typically minimize the differences between the image features extracted from the observations and the projection of the hypothesized object. The projection of the hypothesized object is determined by a set of parameters such as human pose, human shape, and camera projection matrices. Having a *prior* model with a limited number of parameters makes the estimation easier. However, as the number of parameters increases,



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

FIGURE 2.4. **Summary of Body Models.** (a) Kinematic tree body model. x_r and θ_r are the global position and orientation of the root. θ_i is the relative part rotation. (Reprinted from [**93**]). (b) Cylinders body model. (Reprinted from [**41**]). (c) Super-quadrics body model. (Reprinted from [**96**]). (d) Meta-ball body model. (Reprinted from [**84**]). (e) Data-driven statistical body model. (Reprinted from [**84**])

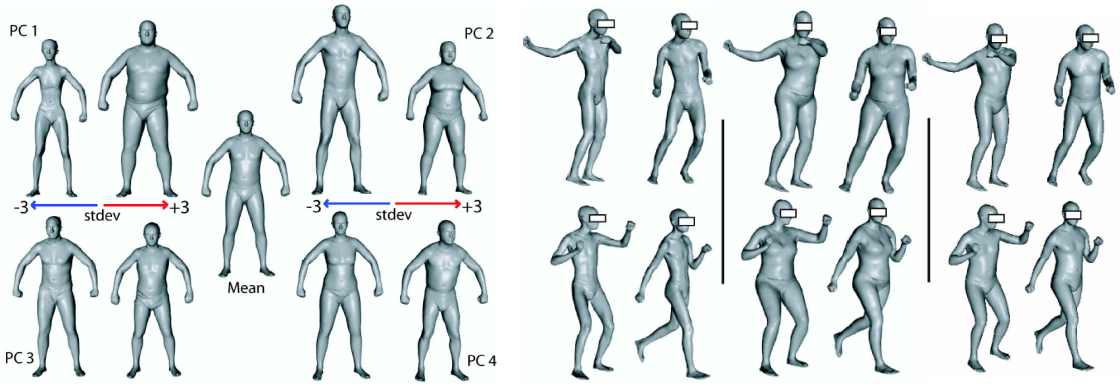
.

the optimization becomes more difficult and it can be trapped in the local optimum. Model-based approaches require the use of a parametric model of the human body. We need to be able to change the articulated pose as well as pose-independent shape or dimensions. The choice of human body model is crucial.

2. Human Body Models

Human body models (Figure 2.4) need to include the kinematic aspect (skeleton, bones) and shape aspect (soft tissue, flesh, muscle, and even clothing) of the human being. Kinematic trees are commonly used to model articulated pose. The body is segmented into parts, which are linked by joints. The parameter space for the kinematic tree include: 1) the position and orientation of the root joint in the world coordinate, and 2) the relative orientation of each joint with respect to its parent joint. In principle, each joint has 3 degrees of freedom (DOFs) for rotation. However, some joints have fewer than 3 DOFs because of the movement constraints of the body joint (e.g. knees). A kinematic tree model typically has 25-60 DOFs in total.

2D Models. Existing shape models are much more diverse, ranging from 2D to 3D models. 2D shape models are mainly based on quadrilateral/elliptical [**3, 44, 68**] patches or contours [**47**]. These 2D models are computationally efficient but they are view dependent and it is impossible



(a) (b)

FIGURE 2.5. **SCAPE Body Model.** (a) The first four principle components in the space of body shape deformation. (b) SCAPE deformations (three subjects, each in four different poses). (Reprinted from [**4**]).

.

to recover the volumetric shape measurements. 3D models, on the other hand, are usually more complex, however they are view independent and the detailed body shape can be recovered.

Geometric Primitives. Simple part based 3D models represent each part as a rigid shape which is connected by the kinematic tree. These models are easy to manipulate and good for articulated pose tracking. However, the rigid parts are commonly represented by simple geometric primitives such as cylinders, truncated cones, and ellipsoids, which are too crude for detailed shape and pose estimation.

Skinning Models. Skeleton-driven surface-based models are very popular in animated movies and gaming. A graphics modeler first needs to create the bone structure and the skin of the 3D character using software such as Maya or 3DMax. Then, in the *rigging* or *skinning* process, each skin vertex is associated with one or more bones that affect the movement of this vertex. The location changes of the vertices are determined by the weighted interpolation of the rigid bone transformations. Such linear blending *skinning* techniques are widely used for character modeling in computer graphics. The weakness of skinning models is that there might be artifacts or non-physical deformations at the joints, because human bodies are highly non-linear structures and sometimes the vertex deformation can not be simply modeled as a linear blending of bone transformations. However, there is no doubt that skinning models are very popular body models and they provide the good tradeoff in terms of realism and ease of use.

Data-driven Statistical Body Models. As 3D body scanners become more accessible, data-driven body models have drawn a lot more attention than before. Data-driven models need a large

database of 3D scans of people who are in different shapes and poses. The deformation of body surface is learned from real 3D scans of people and subtle deformations such as muscle bulging or stretching can be represented using this model. In the training stage, 3D scans of bodies are acquired, post-processed, and aligned. The body deformations, either due to pose changes or intrinsic shapes, are learned from these scans. In the deformation stage, a template body can be deformed to a "new person" in some pose using the learned deformation model. The downside of this type of model is that there is a lot of upfront effort (e.g. 3D scan data post-processing, mesh alignment) before the model can be trained and used. Also, the generalization of the model is largely dependent on the selection of training examples. Ideally, the training set should cover variations of different intrinsic body shapes and body poses, but it requires experience to identify those "good" training examples. As an example, the SCAPE model [**4**] (Figure 2.5) which is learned from real 3D scans of people, is able to represent the novel body shapes not seen in the training set, the articulated body pose deformations, and the non-rigid pose-dependent deformations.

3. 3D Body Shape Estimation from Images

Body *shape* is a pose-independent representation that characterizes the fixed skeletal structure (length of the bones) and the distribution of soft tissue (muscle and fat). There are several methods for representing body shape with varying levels of specificity: 1) non-parametric models such as visual hulls, point clouds and voxel representations (not considered further here); 2) part-based models using generic shape primitives such as cylinders or cones [**41**], superquadrics [**69, 96**] or "metaballs" [**84**]; 3) humanoid models controlled by a set of pre-specified parameters such as limb lengths that are used to vary shape [**35, 58, 76**]; 4) data driven models where human body shape variation is learned from a training set of 3D body shapes [**4, 10, 92, 94**]. Detailed parametric models allow body *reshaping* [**4, 65, 125**] to create new body shapes. This is an interesting research direction and a full review is beyond the scope of this thesis.

Machine vision algorithms for estimating body shape typically rely on structured light, photometric stereo, or multiple calibrated camera views in carefully controlled settings where the use of low specificity models such as visual hulls is possible. As the image evidence decreases, more human-specific models are needed to recover shape. Several methods fit a humanoid model to multiple video frames, or multiple snapshots from a single camera [**35, 96**]. These methods estimate limited aspects of body shape such as scaling parameters or joint locations yet fail to capture the range of natural body shapes.

More realism is possible with data-driven methods that encode the statistics of human body shape. Seo *et al.* [**92**] use a learned deformable body model for estimating body shape from multiple

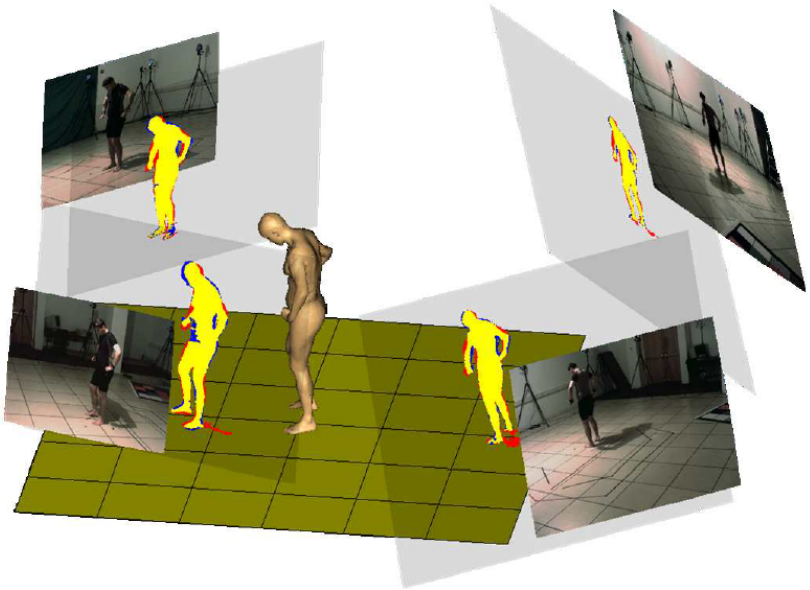


FIGURE 2.6. **Body Shape Estimation from 4 Camera Views.** 3D body shape and pose is estimated from 4 image silhouettes (red). The pose and shape of the body model is optimized such that the model projection in each camera (blue) best matches (yellow) the image silhouette. (Reprinted from **[8]**).

.

photos in a controlled environment with the subject seen in a predefined pose. To estimate a consistent body shape in arbitrary pose it is desirable to have a body model that factors changes in shape due to pose and identity. Balan *et al.* **[10, 9]** (Figure 2.6) show that such a model allows for the shape and pose to be estimated directly from multi-camera silhouettes.

A single monocular image presents challenges beyond the capabilities of all the methods above. Sigal *et al.* **[94]** directly estimate body shape from a single image by training a mixture of experts model to predict 3D body pose and shape directly from various 2D shape features computed from an image silhouette. They estimate body shape in photos taken from the Internet, but require manual foreground segmentation and do not accurately estimate pose. Chen *et al.* **[26, 25]** combine *prior* knowledge of object class and Gaussian Process Latent Variable Models (GPLVM) to infer 3D shapes from a single view. While silhouettes constrain the surface normals at the object boundary, non-rigid deformation, articulation and self occlusion make the silhouette boundary insufficient to recover accurate shape from a single view.

3.1. Body Pose and Shape Under Clothing. Very little work in computer vision has focused on modeling humans in clothing. What work there is focuses on modeling 3D human shape *under clothing* without actually *modeling* the clothing itself. Balan and Black **[7]** (Figure 2.7) present

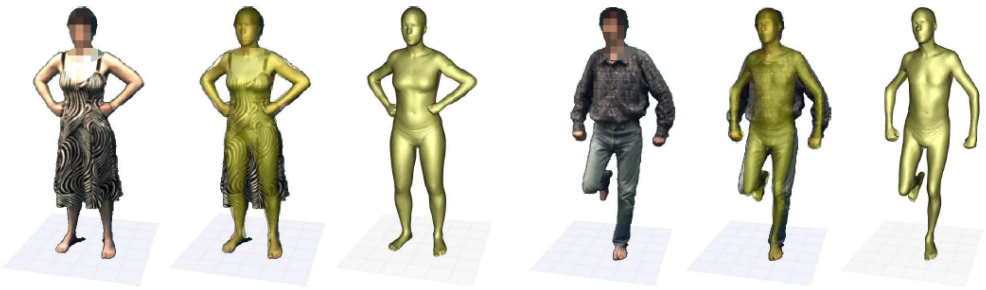


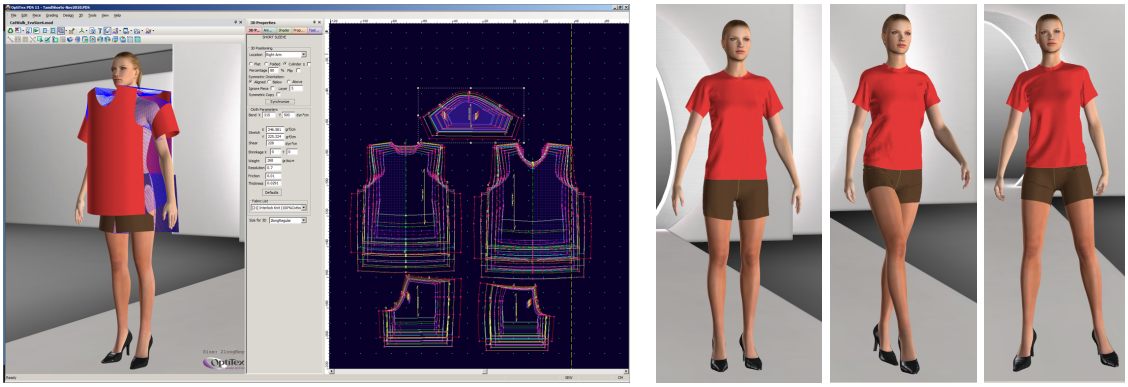
FIGURE 2.7. **The Naked Truth.** Estimating 3D detailed body shape under clothing. (Reprinted from [**7**]).

a system based on the 3D SCAPE [**4**] body model that uses multiple camera views to infer the body shape. They make the assumption that the estimated body shape belongs to a parametric family of 3D shapes that are learned from training bodies. They fit the body to image silhouettes and penalize estimated body shapes that extend beyond the silhouette more heavily than those that are fully inside. This models the assumption that body shape should lie inside the visual hull defined by the clothed body. In essence their method attempts to be robust to clothing by ignoring it. Hasler *et al.* [**57**] take a similar approach to fitting a 3D body to laser range scans of dressed humans.

Almost all work on 2D person detection and pose estimation implicitly assumes the people are clothed. Despite this, few authors have looked at using clothing in the process [**99**] or at actually using a model of the clothing. Recent work by Bourdev and Malik [**19**] learns body part detectors that include upper and lower clothing regions. They do not model the clothing shape or body shape underneath and do not actually recognize different types of clothing.

4. Clothing Animation

When 3D body shapes are estimated, many applications require realistic clothing animation on the bodies. For virtual fashion, the quality of clothing animation is particularly important. In this section, we briefly review clothing animation techniques. The extensive literature on cloth simulation focuses on modeling the physical properties of cloth and developing stable methods that can deal with cloth collisions, friction, and wrinkle buckling [**12, 21, 22, 28, 50**]; see [**29, 62, 80**] for surveys. These methods produce realistic simulations, but at high computation cost. Games and retail clothing applications, however, require efficient solutions because of their interactive nature. Efficient approaches include the Verlet integration scheme [**66**] and GPU acceleration [**18**]. Another important research direction is to automate the clothing fitting procedure, which includes predicting the correct size for 2D patterns or 3D draped forms for a given human body [**52, 77**].



(a) Pattern Design (b) Clothing Simulation Results

FIGURE 2.8. **Clothing Simulation Pipeline.** (a) Design a series of different cloth sizes, and manually initialize the cloth pieces for simulation. (b) Simulation results. (Snapshots generated using OptiTex Inc clothing design and simulation software.)

.

Real Cloth Capture. Structured light, stereo, optical flow, special patterns and multi-camera systems can be used to capture cloth undergoing natural motion **[20, 38, 86, 90, 100, 117]**. These techniques do not immediately provide a way to re-purpose the garments to new sizes and poses but could be used to provide training data to a method like ours. There has been recent interest in using such real cloth motions to learn models of cloth deformation and, in particular, wrinkles **[32, 85]**. Our clothing model requires aligned clothing meshes of different sizes; these are difficult to obtain from scanned garments. Instead, we simulate training clothing using PBS, giving a known alignment between all training instances (cf. **[111]**).

From 2D Patterns to 3D Fitting. Choi et al. **[29]** summarize the major challenges in cloth simulation and the "non-intuitive task of clothing a 3D character with a garment constructed from 2D patterns". There has been relatively little work to address this issue. Cordier et al. **[33]** describe a web application that allows users to interactively adjust a 3D mannequin according to a shopper's body measurements and then resize and fit a garment to the body. Decaudin et al. **[40]** describe a system in which the users draw 2D sketches of contours and lines on a virtual mannequin, and then the system converts these to 3D surfaces. In **[110]**, the system allows users to design clothing directly in 3D space and later flattens the 3D pieces to 2D. Fuhrmann et al. **[48]** define an automated method for positioning 2D pieces around the body to initialize a traditional physical simulation. Umetani et al. **[107]** propose an interactive tool for bidirectional editing between a 2D clothing pattern and a 3D draped form. Our approach is different in that all our effort is up front;

(a) (b)

FIGURE 2.9. **Hair Simulation.** (a) Hair simulation results from **[91]**. (b) Hair simulation in the recent animated movie Brave (2011) by Disney Animation Studio.

.

once a garment is designed, we automate the process of converting it to an infinitely resizable 3D model. Fitting to a new body is then fully automated.

Modeling Wrinkles. Wrinkles are important for producing realistic visual effects, therefore numerous wrinkle generation algorithms have been proposed. One class of methods deforms and blends wrinkles drawn on top of a smooth garment in a set of static poses to synthesize wrinkles as the garment deforms **[34, 55]**. Another approach separates the coarse clothing shape from the fine wrinkle details **[31, 32]**. The coarse shape is obtained by running PBS on a low-resolution version of the mesh. The fine details are synthesized using an example-based method to find appropriate fine wrinkles in a pre-computed wrinkle database or by using linear prediction (regression) from a coarse mesh. Our approach shares ideas with these methods but goes beyond previous work to address how wrinkles vary with body *shape*.

Clothing in Reduced Space. Instead of dealing with the mesh at the triangle level, related work models complex deformations in a lower-dimensional linear subspace **[37, 67, 71]**. This achieves a huge speed up but with reduced realism. These subspace methods replace cloth simulation with a learned dynamical system, where the input is a 3D body mesh and perhaps joint angles of the underlying skeleton, and the output is a clothing mesh. Clothing pose and shape are integrated in this model and no separate control is provided. We take a similar, learning-based, approach but extend this idea to include wrinkles that also depend on body shape.

5. Hair Animation

A large body of work exists on hair modeling, simulation, and rendering. We refer the reader to a survey [**112**] and prior SIGGRAPH course notes [**15, 123**] for an overview. Here we focus on the most relevant methods, mainly those pertaining to real-time-capable approaches.

Hair simulation approaches can loosely be organized into two classes of methods: those that model hair as a continuous medium and those that model it as a set of disjoint, possibly interacting, groups [**112**]. In both cases the rationale is that the number of strands is too large to model each strand individually.

Continuous medium models model hair as a continuum and model complex interactions between strands using fluid dynamics (smooth particle hydrodynamics) [**11, 56**]. Such methods, however, are slow and do not capture clustering effects observed in longer hair.

Disjoint models typically model hair using a sparse set of hair guides, hair strips, or wisps. Hair guides are *representative* strands that are simulated; the dense hair model is then created by interpolating the position of remaining strands from a sparse set of hair guides [**24**]. This approximation allows nearly real-time performance with a moderate number of guides (a GPU implementation with 166 simulated strands can run at 15 FPS [**103**]). Hair strips model hair using thin flat patches (NURBS surfaces) [**72**]. Using a strip to represent tens or hundreds of individual strands leads to significant efficiencies, including in collision handling, resulting in real-time performance; consequently this approach is often used in games. However, strips are unable to represent complex hair styles or motions. Wisps model bundles of hair strands as volumetric primitives [**27, 115, 121**]. These approaches are particularly good at modeling hair styles with well-defined clusters; however, they are typically computationally expensive (e.g., requiring seconds per frame to compute [**27**]). Another promising approach uses the hair mesh structure for modeling the hair volume; topological constraints allow an automatic and unique way to trace the path of individual hair strands through this volume [**122**]. With a coarse resolution mesh this approach is able to simulate hair at 92 FPS. However, the coarse resolution of the mesh does not allow for fine movement of individual strands.

Our method exploits the hair guide formulation, but further reduces computational complexity by modeling hair guides in the reduced space. We also show that by treating *all* hair strands as guide curves in our framework, we can forego the interpolation step as our model learns to incorporate "interpolation" as part of the mapping from the low-dimensional reduced hair space to the full-dimensional hair representation.

CHAPTER 3

Estimating Human Shape and Pose from a Single Image

This chapter is built upon the original work [54]. In this chapter, we describe a solution to the challenging problem of estimating human body *shape* from a single photograph or painting. Our approach computes shape and pose parameters of a 3D human body model directly from monocular image cues and advances the state of the art in several directions. First, given a user-supplied estimate of the subject's height and a few clicked points on the body we estimate an initial 3D articulated body pose and shape. Second, using this initial guess we generate a tri-map of regions inside, outside and on the boundary of the human, which is used to segment the image using graph cuts. Third, we learn a low-dimensional linear model of human shape in which variations due to height are concentrated along a single dimension, enabling height-constrained estimation of body shape. Fourth, we formulate the problem of *parametric human shape from shading.* We estimate the body pose, shape and reflectance as well as the scene lighting that produces a synthesized body that robustly matches the image evidence. Quantitative experiments demonstrate how smooth shading provides powerful constraints on human shape. We further demonstrate a novel application in which we extract 3D human models from archival photographs and paintings.

1. Introduction

While the estimation of 3D human pose in uncalibrated monocular imagery has received a great deal of attention, there has been almost no research on estimating human body shape. The articulated and non-rigid nature of the human form makes shape estimation challenging yet its recovery has many applications ranging from graphics to surveillance. Here we describe the first complete solution to the problem of human shape estimation from monocular imagery. In contrast to the standard multi-camera setting, we observe that a single image silhouette is generally insufficient to constrain 3D body shape. To address this we propose the use of additional monocular cues including smooth shading. Given an initial guess of the body pose, we optimize the pose, shape and reflectance properties of a 3D body model such that it robustly matches image measurements. The resulting body model can be measured, posed, animated, and texture-mapped for a variety of applications. The method is summarized in Figure 4.1.

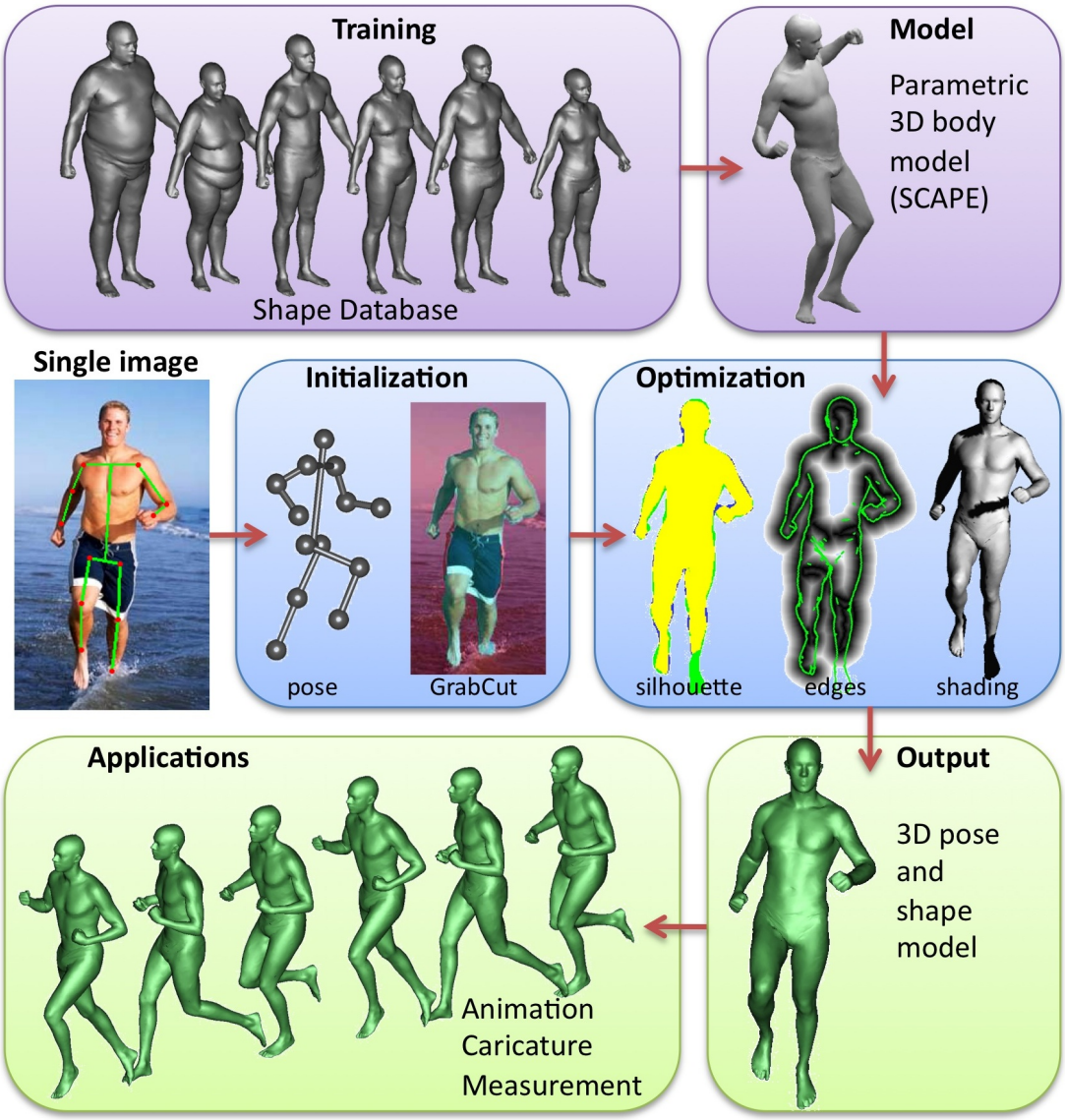


FIGURE 3.1. **Overview.** Given a single image and minimal user input, we compute an initial pose, light direction, shape and segmentation. Our method optimizes 3D body shape using a variety of image cues including silhouette overlap, edge distance, and smooth shading. The recovered body model can be used in many ways; animation using motion capture data is illustrated.

Most work on human pose or shape estimation assumes the existence of a known background to enable the extraction of an accurate foreground silhouette. With a monocular image, however, no known background can be assumed. Still, the outline of the body provides a strong constraint on body shape. Given an initial pose, obtained by manual clicking on a few image locations corresponding to the major joints of the body [**75, 104**], and the mean body shape, we create an initial

foreground region from which we derive a tri-map for GrabCut segmentation [**88**]. This produces an accurate foreground region (cf. [**46**]).

Our parametric representation of the body is based on the SCAPE model [**4**] and our optimization follows that of Balan *et al.* [**9, 10**] but extends it to monocular imagery. Body pose in a monocular image is inherently ambiguous and its estimation from a single silhouette is poorly constrained. If the body limb lengths are not known, then multiple poses can equally well explain the same foreground silhouette [**97**]. To deal with this we constrain the height of the subject during optimization. Previous SCAPE formulations, however, represent variation in human shapes using linear shape deformation bases computed with principal component analysis. Since height is correlated with other shape variations, height variation is spread across many bases. We address this by rotating the learned SCAPE basis so that height variation is concentrated along a single shape basis direction. The height can then be held fixed during optimization, significantly improving monocular shape and pose estimation.

One of the key contributions of this work is the formulation of *body shape from shading*. Unlike the generic shape from shading problem, our goal is to estimate the body shape parameters, the pose of the body, its reflectance properties and the lighting that best explains the shading and shadows observed in the image (similar in spirit to [**16**] but with a more complex model). We assume a single point light source but our experiments suggest that the method is quite robust to violations of this assumption. Since skin has a significant specular component, we approximate its reflectance with a Blinn-Phong model [**17**] and an assumption of piecewise smoothness. Given a body shape, body pose, light direction and skin reflectance we robustly match a synthesized image of the person with the observed image. Note that exploiting shading cues requires accurate surface normals, which are provided by our learned body shape model. Shading information provides strong constraints on surface shape that improve the estimated body shape when combined with other cues.

Shape from shading has a long history in computer vision (see [**124**] for a review) yet typically focuses on recovering the shape of unknown surfaces. Here we have a different problem in which we know that the object is a human but the pose and shape are unknown. For a given set of body shape and pose parameters we can compute the surface normals at each point on the body mesh. We then formulate and optimize a robust *shape from shading* objective function in which the normals are a function of the shape parameters. Similar to this is the work of Samaras and Metaxas [**89**], which constrains a 3D shape using shading information. We go beyond their work to deal with a learned shape deformation model and articulation.

The majority of work related to shading and the human body focuses on carefully calibrated laboratory environments. Theobalt *et al.* [**105**] recover human body shape and detailed reflectance

properties but do so in a multi-camera calibrated environment with careful lighting. While shading has been explored for recovering human face shape (e.g. [**16**]) we know of no work using it to recover human body shape. Balan *et al.* [**9**] recover the albedo of the body using multiple known poses and a Lambertian reflectance model but do not use this to estimate shape. These methods are not applicable to the monocular, uncalibrated case studied here.

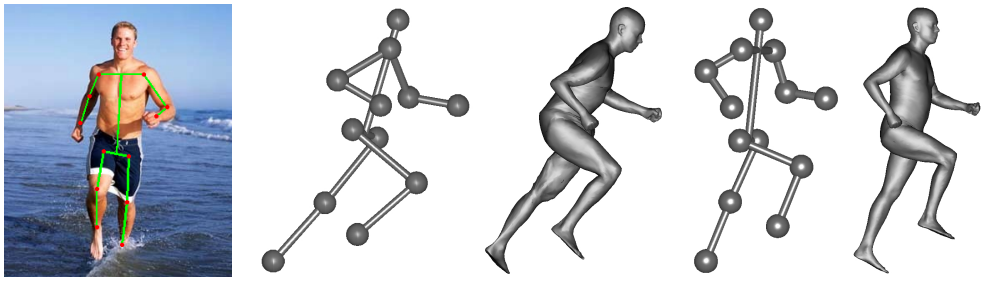
In recent work, de La Gorce *et al.* [**39**] use an accurate hand shape model and shading information for monocular tracking. Given a fixed hand shape, they estimate the albedo of the hand and the illumination in each frame and use these to constrain pose during tracking. The hand shape is controlled by 51 scaling parameters that are estimated in an initialization step where the hand is known to be parallel to the image plane, the background is known and the albedo is assumed to be constant. Our work goes beyond this to estimate a parametric shape model for the whole body in arbitrary poses with piecewise smooth albedo and unknown background.

We quantitatively evaluate the method in a laboratory environment with ground truth 3D shape. We also show a novel application where we compute 3D human shape from photographs and paintings. Here our assumptions about the illumination are only approximate, yet the method is able to recover plausible models of the human body. These models can be texture-mapped (with either texture from the scene or some other source), animated in new poses or deformed to create caricatures.

2. Body Model and Fitting

SCAPE is a deformable, triangulated mesh model of the human body that accounts for different body shapes, different poses, and non-rigid deformations due to articulation [**4**]. For vision applications, it offers realism while remaining relatively low dimensional. We use a mesh with $m = 12,500$ vertices [**10**].

Articulated pose is parametrized by a set of rigid body part rotations $\vec{\theta}$, while changes in body shape between individuals are captured by shape deformations gradients $\vec{d}$ between a reference mesh and a new mesh in the same pose. A low-dimensional statistical model of body shape deformations is learned using principal component analysis (PCA). We learn two gender-specific models from laser scans of over 1000 men and 1000 women, respectively. For a given mesh, the shape deformation gradients are concatenated into a single column vector and approximated as $\vec{d} = \mathbf{U}\vec{\beta} + \vec{\mu}$ where $\vec{\mu}$ is the mean body shape, $\mathbf{U}$ are the first n eigenvectors given by PCA and $\vec{\beta}$ is a vector of linear coefficients that characterizes a given shape; $n = 20$ in our experiments. In Section 3 we extend this formulation to model deformations that preserve height.



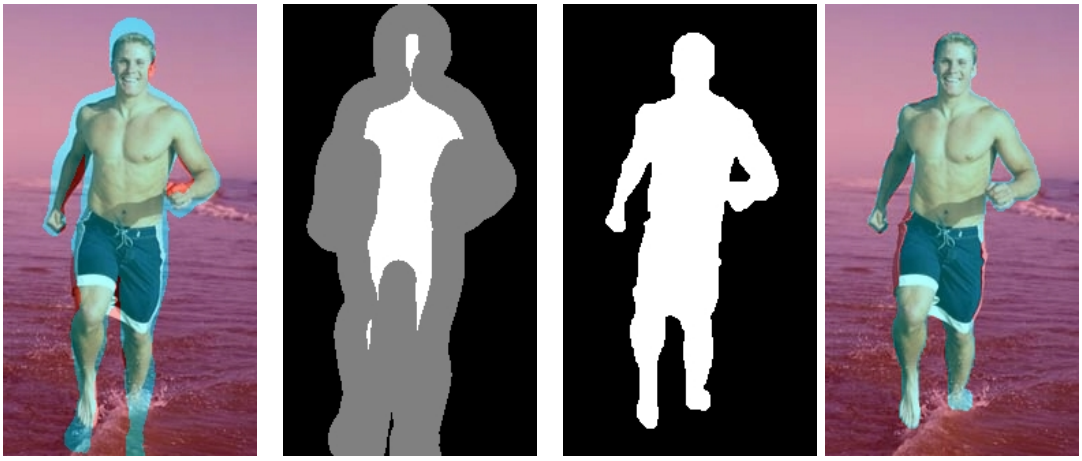
(a) Clicked Points (b) Taylor's Method (c) Perspective

FIGURE 3.2. **Initialization using clicked points on the input image.** Pose estimated with orthographic (b) and perspective (c) camera models, shown from an alternate view. Mean body shape (male) is shown transformed into the pose of the initialized models.

Given a monocular image, our goal is to estimate the shape parameters $\vec{\beta}$ and pose parameters $\vec{\theta}$ that best explain the image evidence. The model parameters are used to produce a 3D mesh, $Y(\vec{\beta}, \vec{\theta})$, that is projected onto the image plane to obtain silhouettes, edges, or shaded appearance images (Fig. 4.1). We denote the body parameters by $\Theta_B = [\vec{\beta}, \vec{\theta}]$. We use standard distance functions for silhouettes, $E_{\text{Si}}(\Theta_B)$, **[10, 96]** and edges, $E_{\text{Eg}}(\Theta_B)$, **[41]** and introduce a novel shading term. The objective function, which also includes an inter-penetration penalty, $E_{\text{Pn}}(\Theta_B)$, is minimized using a gradient-free direct search simplex method. In the monocular case a reasonably good initial estimate of the pose parameters is necessary, as well as a silhouette extracted without a background image. We also combine segmentation and model fitting to seek the pose and shape that best segments the image into foreground and background regions. Both these problems are addressed in the following section.

2.1. Pose Initialization. 3D body pose is initialized in the camera coordinate system using clicked 2D points corresponding to the major joints (Fig. 3.2) **[75, 104]**. We find that the orthographic method of Taylor **[104]** (Fig. 3.2b) produces poses that are inaccurate compared with the perspective method of **[75]** (Fig. 3.2c). The perspective method requires an estimate of focal length which we extract from EXIF metadata when available or which we obtain from user input. We further find that even an approximate focal length produces better initial poses than the orthographic assumption.

Unlike the orthographic case, perspective projection requires a way to position the root joint in 3D. First, the limb most parallel to the image plane is automatically identified as the one that minimizes the ratio between its image length and its actual length. If the limb is parallel to the



(a) Segmentation (b) Tri-map (c) Segmentation Result

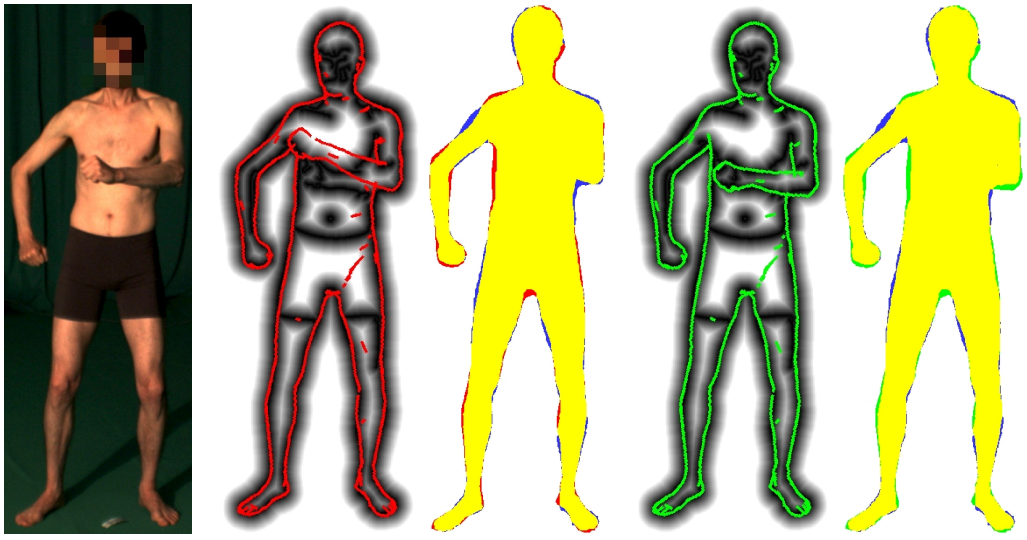
FIGURE 3.3. **Segmentation.** (a) Silhouette corresponding to initial pose and average shape projected into image. (b) Tri-map extracted from initial silhouette by erosion and dilation. (c) GrabCut segmentation result (silhouette and its overlay in the image).

image plane, the depth is uniquely determined using the ratio of similar triangles. If not we use a foreshortening factor similar to the scale parameter in [**104**].

In contrast to [**104**], we do not explicitly require limb lengths as input. Rather, we predict these from a database of over 2400 subjects based on user specified height and gender. We use this database to build a height constrained shape space as described in Section 3, allowing us to deform the mesh to match the mean person of the specified gender and height, and then extract limb lengths from linear combinations of specific vertices.

To find such a mesh we use a rotation of scape bases that will be fully detailed in Section 3 such that height can be considered independently from the remainder of shape. We extend the previous methods to also initialize head pose by solving for the neck rotation that minimizes the distance between several user-clicked 2D face feature points and the corresponding 3D vertex positions on the mesh projected into the image.

2.2. Region-based Segmentation. We use GrabCut [**88**] to perform image segmentation, leveraging the pose initialization to seed GrabCut in an automated way as illustrated by Figure 3.3. Given the initial mesh (Fig. 3.2c), we render its silhouette into the image. This provides an initial guess for foreground segmentation (Fig. 3.3a). Specifically, we construct a tri-map, defining each pixel as foreground, background, or uncertain by eroding and dilating the initial region by 5% of the image width (Fig. 3.3b). The resulting tri-map is used to initialize GrabCut [**88**] which is used



(a) (b) Result without edges (c) Result with edges

FIGURE 3.4. **Internal edges.** (a) Laboratory image with self occlusion. (b) Pose estimation with only the silhouette term cannot estimate the arm pose. Edges (red) of optimized model projected into the edge cost image. Yellow shows the overlap of the model and image silhouettes, blue/red represent unmatched image/model silhouette regions. (c) The estimated pose (green) with the edge term matches the true pose. Note how well the model edges align with the edge cost image.

to segment the foreground. This process is similar to that of Ferrari *et al.* [**46**] but with a 3D body model used for initialization.

2.3. Preventing Limb Inter-penetration. With a monocular view many possible poses cannot be ruled out based on image evidence and some of these involve interpenetration of body parts. To prevent these impossible solutions we develop a fast method to detect and penalize inter-penetrations during optimization. We approximate the shape of a limb by its convex hull and determine if a point is inside by computing the dot products between the triangle normals and the rays from the point to the center of the triangles. Because of the convexity assumption, the point must lie outside the convex hull if any of the dot products are negative. We check each of the 12 extremity parts (thigh, calf, foot, etc.) against the remaining point cloud and compute fraction of vertices that are contained within the hulls of the other parts. We compute a robust penalty function of this quantity summed over all parts and this becomes a cost of interpenetration, $E_{Pn}(\vec{\theta}, \vec{\beta})$, that is included in the objective function.

2.4. Internal Edges. It is well known that silhouettes do not provide pose constraints in regions where one body part occludes another (e.g. Fig. 3.4b). Numerous authors have dealt with

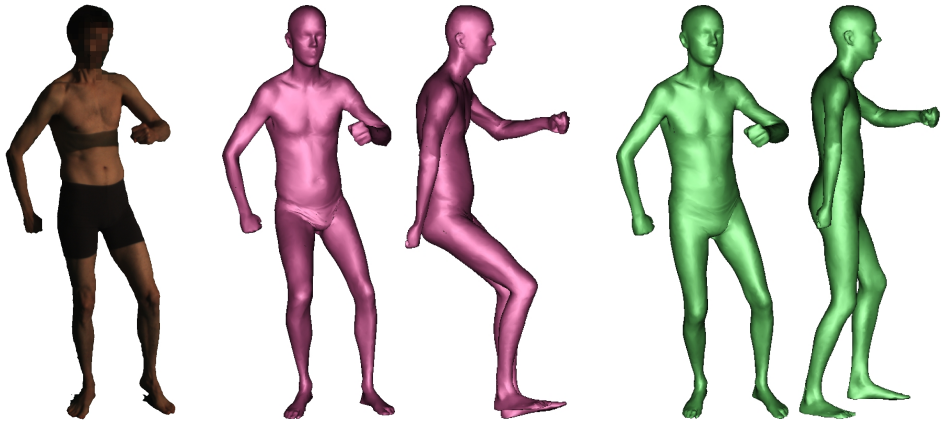


FIGURE 3.5. **Height Constrained Optimization.** Two different body shapes and poses can explain the same image silhouette. Pose and shape estimated without constraining height (magenta). When turned sideways we see it is wrong. Constraining the height during estimation produces a realistic pose (green).

this by combining edge information with silhouettes. We do so as well but, with the SCAPE body model, these edges provide a better fit to the image evidence than do previous models.

We detect image edges using a standard edge detector and apply a thresholded distance transform to define an edge cost map normalized to $[0, 1]$. Model edges are detected by examining each triangle edge in the mesh, taking the dot product of the ray from the camera to the midpoint of the edge with the triangle normals on either side of the edge. A change in the sign of the dot product across the triangle edge indicates a possible model edge. If either triangle is visible from the camera (by ray triangle intersection) then the edge is a visible model edge.

We use the "trapezoid rule" to evaluate the line integral of the set of all visible model edges over the edge cost image. This defines an edge cost, $E_{\text{Eg}}(\Theta_B)$, that is included in the objective function, improving the accuracy of the fit (Fig. 3.4c).

3. Attribute Preserving Shape Spaces

The ambiguities present in inferring 3D pose and shape from a single image mean that we must constrain the search space as much as possible. In a monocular setting we find it useful to be able to constrain the search space based on prior knowledge. Figure 3.5 illustrates one such ambiguity where the wrong body shape can be compensated for by a change in pose. Viewed monocularly, both models explain the image silhouette equally well. Additional information such as the height of the person can remove some of the ambiguity. Unfortunately, the SCAPE eigen-shape representation does not provide any direct control parameters corresponding to intuitive attributes like gender, height or weight that can be specified by a user. If these can be derived as functions of the linear

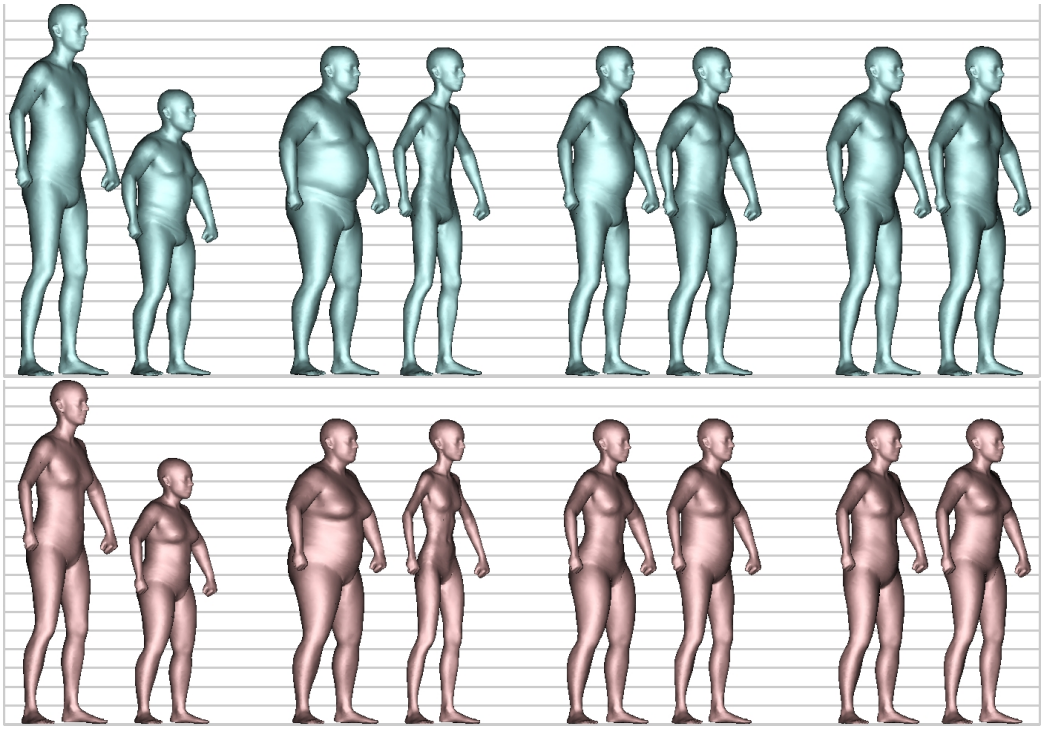


FIGURE 3.6. **Height preserving body shape space.** First pair on each row (men above, women below) shows variation ($\pm$ 3 std) along the height-variation axis. The other pairs show variation ($\pm$ 3 std) along the first three height-invariant axes. Note that shape varies along these axes but height varies by less than $3mm$ for each pair.

coefficients, then they can be included as constraints during body shape estimation. In general, enforcing soft-constraints on attributes is possible using a constrained minimization algorithm, but it makes it more susceptible to getting stuck in local minima. Instead we take a more direct approach and construct a rotation of the original eigen-shape space such that height variation is removed from all but one of the bases. This allows us to optimize over body shapes without varying height.

In previous work, Blanz and Vetter [**16**] compute a direction in shape coefficient space such that any movement along this axis manipulates a certain attribute the most while keeping all the other attributes as constant as possible. This is not equivalent to saying that any movement orthogonal to this axis preserves the attribute, which is what we want. In fact, their axis is not optimized for and fails to preserve an attribute value along orthogonal directions.

Allen *et al.* [**1**] learn a linear mapping from a fixed set of attributes to shape parameters. One could optimize body shape using these parameters instead of PCA coefficients. Preserving an attribute can then be achieved by simply keeping it fixed, but this approach reduces the modes of shape variation to the set of attributes considered.

In contrast, our approach explicitly searches for attribute-preserving directions in the eigen-space and re-orients the bases along these directions. While we focus on constraining height, our method applies to any other geometric attribute that can be measured directly from the mesh (volume, geodesic distances, etc.). Body height $H(\vec{\beta})$ can be measured by reconstructing a mesh $Y(\vec{\beta}, \vec{\theta}^H)$ in a predefined standing pose $\vec{\theta}^H$. Let $\mathbf{G}_1 = \mathbf{I}_n = [\vec{e}_1, \ldots, \vec{e}_n]$ be the identity basis for the shape coefficients ($\vec{d} = \mathbf{U}\mathbf{G}_1\vec{\beta} + \mu$). We seek a new orthonormal basis $\mathbf{G}$ such that none of its bases account for height except one, which becomes the height axis. $\mathbf{G}$ should also preserve the representational power of the original bases: the sub-space spanned by the first j bases is the same after the change of bases, absent the height axis. Our solution works in an incremental fashion and maintains orthogonality at all times by rotating pairs of bases so that one of the bases preserves height while the other moves towards the height axis. First, we start by selecting a candidate basis $\vec{e}_k$ for the height axis as the one that maximizes the absolute correlations between height and shape coefficients of the training examples. Second, we iterate over the remaining bases $\vec{e}_j$ and rotate the current $(\vec{e}_j, \vec{e}_k)$ plane to make $\vec{e}_j$ height preserving. Third, the rotation matrix is used to update, at iteration j, the orthonormal basis $\mathbf{G}_j =$

$$\mathbf{G}_{j-1}\mathbf{R}_{jk}\left(\arg\min_{\varphi}\left(H(\vec{0}_n) - H(\mathbf{G}_{j-1}\mathbf{R}_{jk}(\varphi)\vec{e}_j)\right)^2\right),$$

where $\mathbf{R}_{jk}(\varphi)$ is a $n \times n$ rotation in the $(\vec{e}_j, \vec{e}_k)$ plane:

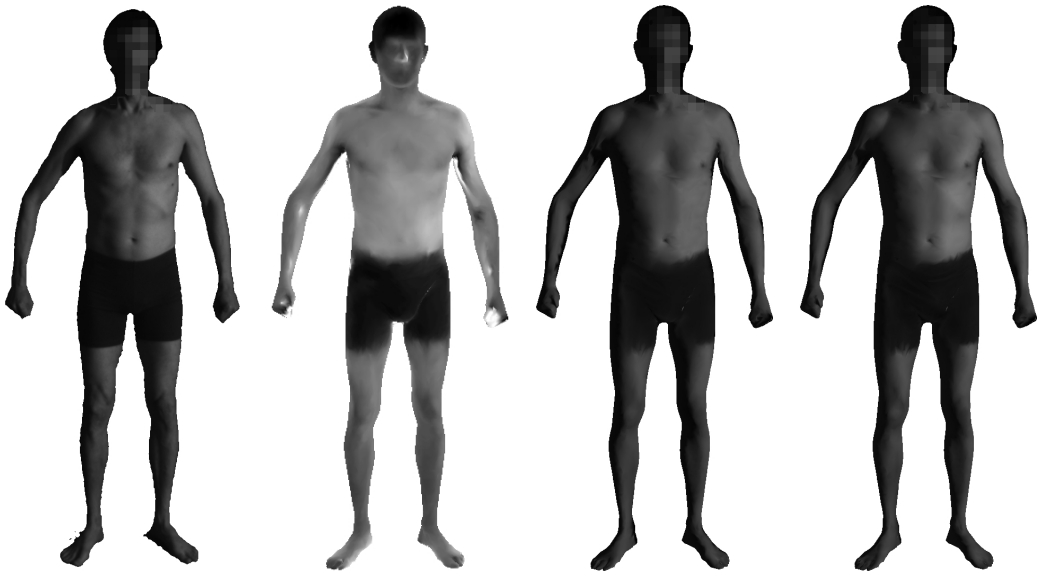
$$\mathbf{R}_{jk}(\varphi) = \begin{array}{c} \\ j \\ \\ k \\ \end{array}\!\!\begin{array}{c} \begin{array}{cccc} j & & k & \end{array} \\ \begin{pmatrix} \mathbf{I} & & & 0 \\ & \cos(\varphi) & -\sin(\varphi) & \\ & & \mathbf{I} & \\ & \sin(\varphi) & \cos(\varphi) & \\ 0 & & & \mathbf{I} \end{pmatrix} \end{array}.$$

The body shape in the new height-preserving shape space can be expressed as $\vec{d} = (\mathbf{U}\mathbf{G}_n)\vec{\beta}' + \vec{\mu}$, where $\vec{\beta}' = (\mathbf{G}_n)^{-1}\vec{\beta}$. By convention, we compute the variance along the new bases and order them in decreasing order following the height axis. Figure 3.6 shows deviations from the mean shape in the male and female height-preserving shape spaces.

For many subjects (e.g. celebrities), height may be known. When unknown (e.g. in paintings) we use the mean height for each gender (women=$1.65m$, men=$1.82m$).

4. Body Shape from Shading

We approximate the body's reflectance using a Blinn-Phong model with diffuse and specular components [**17**]. We assume a single light source and ambient illumination. Let $X(\Theta_B) =$



(a) Input (b) Albedo (c) Lambertian (d) Blinn-Phong

FIGURE 3.7. **Estimated reflectance.** Blinn-Phong model captures specular highlights and is more accurate than the Lambertian model. Note robust spatial term captures discontinuous albedo.

$[\vec{x}_1, \vec{x}_2, ..., \vec{x}_m]$ be the positions of the m vertices of a body mesh, and $N(\Theta_B) = [\vec{n}_1, \vec{n}_2, ..., \vec{n}_m]$ be the associated unit length normals. Notice that both X and N are functions of the pose and shape parameters, allowing us to formulate a parametric *shape from shading* problem. Let $\vec{a} = [a_1, a_2, ..., a_m]$ be the albedo of each vertex and $\vec{s} = [s_1, s_2, ..., s_m]$ be the specularity of each vertex. The shading value of each surface point i is approximated by:

$$\hat{r}_i = b + a_i(\vec{\ell}_i \cdot \vec{n}_i)l + s_i(\vec{h}_i \cdot \vec{n}_i)^\alpha l \tag{1}$$

where $\vec{\ell}_i$ is the direction from vertex $\vec{x}_i$ toward the light source, $\vec{h}_i$ is the halfway vector between $\vec{\ell}_i$ and the direction from vertex i toward the camera, b is ambient illumination, l is light intensity, and α is the specular exponent.

For a distant directional light source (outdoor scene) l is constant for every vertex, while for a point light source (indoor scene) we use a quadratic attenuation function for light intensity with distance from the source (as in [**9**]).

Optimization. The body is placed at the origin of a spherical coordinate system and the light position is parametrized as $\Theta_L = [\gamma, \phi, z]$ with respect to the body center, where γ and ϕ are azimuth and elevation respectively and z is the distance between the light source and the body. The parameters $\vec{\ell}_i$, $\vec{h}_i$ and l in Eq. 1 depend on Θ_L. We denote the reflectance parameters $\Theta_R = [\vec{a}, \vec{s}, b, \alpha]$. Suppose r_i is the linearly interpolated intensity in the input image where vertex i is projected, our

goal is to minimize the energy function $E_{\mathrm{Sh}}(\Theta_B, \Theta_R, \Theta_L) \propto$

$$\sum_{i \in visible} \Big\{ \rho_{\eta_1}(\hat{r}_i(\Theta_B, \Theta_R, \Theta_L) - r_i) + \lambda_1 \sum_{j \in \mathcal{N}(i)} \frac{\rho_{\eta_2}(a_j - a_i)}{d_{j,i}} + \lambda_2 \sum_{j \in \mathcal{N}(i)} \frac{\rho_{\eta_3}(s_j - s_i)}{d_{j,i}} \Big\} \tag{2}$$

where $\mathcal{N}(i)$ are the vertices connected to vertex i, $d_{j,i}$ is $|\vec{x}_j - \vec{x}_i|$, and $\rho_\eta(x) = \frac{x^2}{\eta^2 + x^2}$ is a robust error function **[49]** used to deal with outliers. The robust error function treats whatever samples that satisfy $|x| > \frac{\eta}{\sqrt{3}}$ as outliers and pays constant penalty for them. Note that $\hat{r}_v$ is actually a function of Θ_B, Θ_R which we omit for notational simplicity.

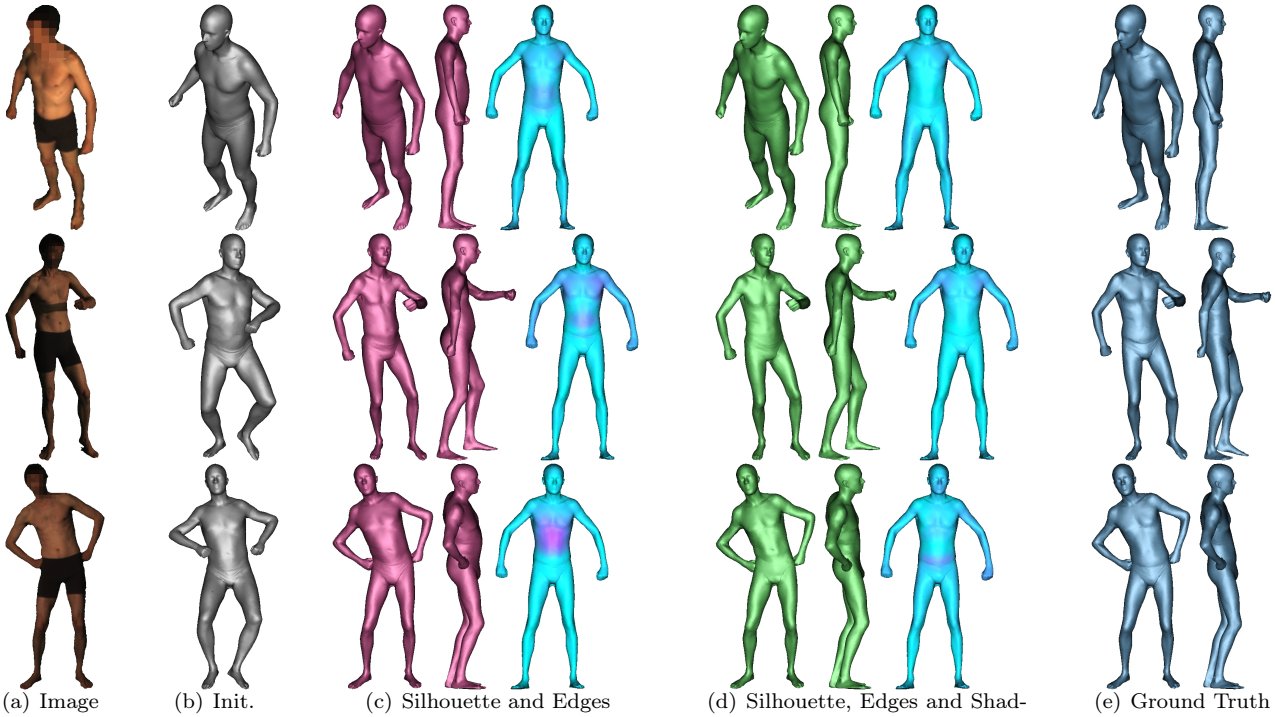
The first term in Eq. 2 penalizes the difference between measured intensities in the observed image, r_i, and the predicted brightness of corresponding model vertices, $\hat{r}_i(\cdot)$. The second term ensures that neighboring vertices have similar albedo. The robust formulation provides a piecewise smooth prior on albedo that allows spatial variations due to clothing, hair, variation in skin color, etc. The third term provides a piecewise smooth prior over specularity. λ_1 and λ_2 weight the relative faithfulness to the observed data and the spatial smoothness assumptions.

The user coarsely initializes Θ_L and then the energy function is minimized in an alternating fashion. First, Θ_L is optimized given fixed Θ_B and Θ_R. (Note that in the first iteration, Θ_B is the initial guess of pose and shape; the albedo and specularity in Θ_R are considered uniform.) Second, we optimize Θ_R with fixed Θ_L and Θ_B. Given the robust formulation in Eq. 2 no closed form solution is possible so we minimize using gradient descent. Third, we fix Θ_L, Θ_R and optimize Θ_B but here the optimization is more difficult since changing Θ_B affects the predicted brightness through changes in the vertex normals. Consequently a gradient-free simplex method is employed to solve step 3. We alternate between the three steps until a convergence criterion is met. We vary the λ values during optimization, starting with larger values and gradually decreasing them, so that the shape is forced to change in order to make the predicted brightness match the image observations. We find that initial pose needs to be fairly accurate, but illumination direction is relatively insensitive to the initialization. Figure 3.7 shows the estimated reflectance for one input image.

5. Results

For quantitative analysis, we captured the pose and shape of a subject using eight synchronized and calibrated cameras with a single "point light source" and a green screen background. We fit the SCAPE model to the eight silhouettes and treat the resulting shape as ground truth.

We then quantify the extent to which shading cues improve monocular shape estimation by comparing the shape estimated with two formulations. In the "Silhouettes and Edges" (***SE***) formulation,



(a) Image (b) Init. (c) Silhouette and Edges (d) Silhouette, Edges and Shading (e) Ground Truth

FIGURE 3.8. **Comparison between *SE* (red) and *SES* (green).** Comparisons are performed on three different poses taken from different viewing angles. The initialization (b) is shown in the camera view. Results and ground truth are shown in both the camera view and in profile. For each result we also show an error map in a canonical pose, indicating per vertex displacement from ground truth; blue indicates low error, while purple indicates high error. Note the lower error for the ***SES*** model.

we fit the pose and shape of the SCAPE model in the height preserving space by optimizing the cost function $E_1 = E_{\text{Si}}(\Theta_B) + E_{\text{Eg}}(\Theta_B) + E_{\text{Pn}}(\Theta_B)$. The "Silhouettes, Edges, and Shading" (***SES***) formulation extends the first by incorporating shading cues; that is, $E_2 = E_1 + E_{\text{Sh}}(\Theta_B, \Theta_R, \Theta_L)$.

Figure 3.8 illustrates how smooth shading improves shape recovery. Silhouettes, even with internal edges, are not sufficient to capture accurate body shape from monocular images. Incorrect estimates happen in areas where surface normals are oriented towards the camera, such as the abdomen in frontal images. In these regions shading provides a strong cue that constrains the body shape.

Anthropometric measurements of chest size, waist size, and weight are provided in Table 3.1. Waist and chest circumference are computed by transforming the body to a canonical pose, slicing the mesh on a fixed plane and computing the convex hull of the contour. Weight is estimated

	Chest Size (cm)			Waist Size (cm)			Body Weight (kg)		
	SE	***SES***	***GT***	***SE***	***SES***	***GT***	***SE***	***SES***	***GT***
Pose 1	95.7 (+3.1)	92.7 (+0.1)	92.6	86.4 (+6.2)	79.6 (-0.6)	80.2	72.0 (+8.2)	65.4 (+1.6)	63.8
Pose 2	84.3 (-7.3)	87.1 (-4.5)	91.6	83.7 (+4.3)	78.5 (-0.9)	79.4	62.5 (-0.7)	62.4 (-0.8)	63.2
Pose 3	95.4 (+4.0)	91.9 (+0.5)	91.4	88.0 (+7.7)	76.9 (-3.4)	80.3	70.8 (+8.2)	63.5 (+0.9)	62.6

TABLE 3.1. **Anthropometric Measurements.** ***GT*** stands for ground truth size. The value in the parenthesis is the deviation from GT size. (Note that the ground truth sizes for each frame vary a little bit, since non-rigid deformations caused by articulations of body will result in variations of shape details.)

FIGURE 3.9. **Applications.** Shape and pose recovered from a single image; texture-mapped in new pose; caricature.

from the body volume by assuming it has the constant density of water. ***SES*** shows substantial improvement over ***SE***.

Figure 3.9 shows an image from the Internet with recovered pose and shape. Note that reflections off the water clearly violate our simple lighting model. Despite that the shape is well recovered. We animate the figure by preserving shape and generating meshes in novel poses from motion capture data. The model can be texture mapped with the image texture or some new texture. Large pose changes may require the texture synthesis of missing data. We can also vary the recovered shape to produce a caricature (Fig. 3.9 right). We do so by finding the shape coefficient with the most significant deviation from the mean and exaggerate it, moving the shape further from the mean in that direction. Here it produces a more muscular physique.

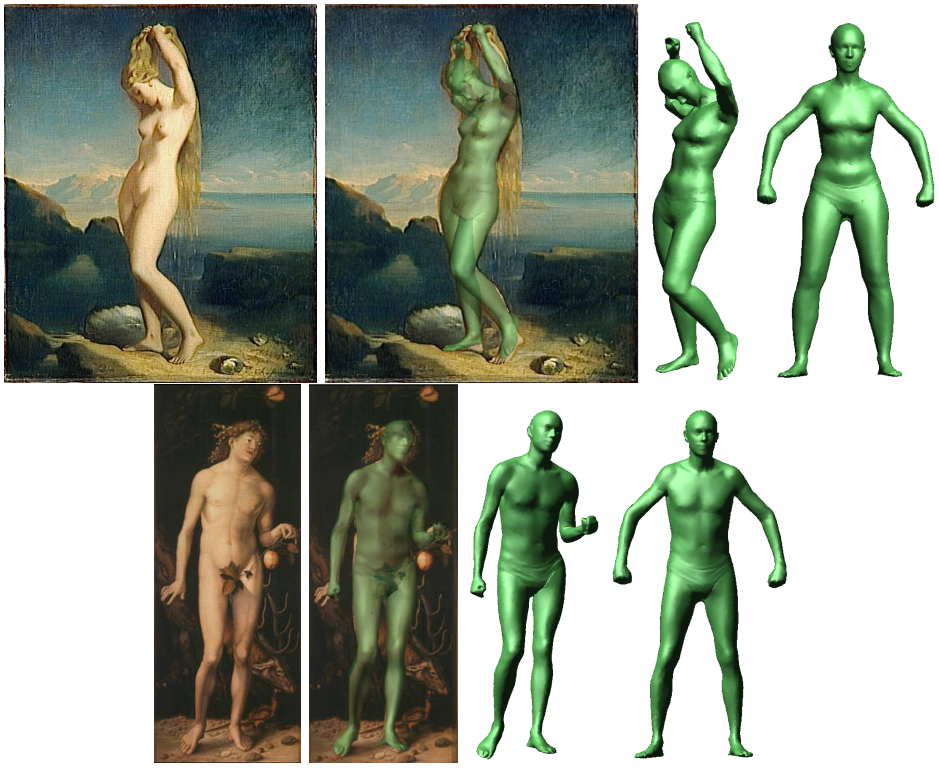


FIGURE 3.10. **Body shape and pose from paintings.** (up) Venus Anadyomene, Théodore Chasseriau, 1838. (down) Adam and Eve (detail), Hans Baldung Grien,1507. Images (left to right): painting, model overlay, recovered shape and pose, shape in new pose.

Although paintings rarely conform to a physical lighting model, we find that shading cues are often significant. Using the same robust formulation as for photographs we recover body pose and shape from two paintings in Fig. 3.10.

6. Discussion

We have described a complete solution for reconstructing a model of the human body from a single image with only minimal user intervention. The main insight is that even a single image contains a range of cues that can constrain the interpretation of 3D body shape. While the bounding contour of the body alone is not sufficient, smooth shading can provide a powerful additional cue. Consequently we developed a new robust method for computing parametric body shape from shading. We also developed a new linear model of body shape deformation in which height variation is removed. The ability to extract body shape from a single image makes several new applications possible. For example, a character from a painting or photograph can be "brought to life" and animated in new poses.

The method as described has several limitations. We assume a single point light source and a simplified model of surface reflectance. None of our experiments actually conform to this model, and yet it still provides a useful approximation. Future work should consider expanding this to more general lighting conditions. We also plan to study more qualitative models of shading. Even in art which is not physically realistic, there are still strong local cues that we should be able to exploit to constrain body shape.

Our experiments have focused on naked or minimally clothed people. Previous work has shown that body shape can be recovered even when people are wearing clothing if multiple poses and camera views are available [**7**]. Extending this to the monocular case is challenging as shading cues would need to be extended to model the complex shading variation caused by clothing.

Other future work will consider automating the initialization stage using a bottom-up 2D person detector and integrating body segmentation with the 3D model fitting process. Since our body shape representation is independent of pose we can also combine constraints from multiple snapshots of the same person. Each image may contain only weak cues but together they could constrain body shape.

CHAPTER 4

2D Eigen Clothing Model for Body Shape Estimation

This chapter is built upon the original work [51]. Detection, tracking, segmentation and pose estimation of people in monocular images are widely studied. Two-dimensional models of the human body are extensively used, however, they are typically fairly crude, representing the body either as a rough outline or in terms of articulated geometric primitives. We describe a new 2D model of the human body contour that combines an underlying naked body with a low-dimensional clothing model. The naked body is represented as a Contour Person that can take on a wide variety of poses and body shapes. Clothing is represented as a deformation from the underlying body contour. This deformation is learned from training examples using principal component analysis to produce *eigen clothing*. We find that the statistics of clothing deformations are skewed and we model the *a priori* probability of these deformations using a Beta distribution. The resulting generative model captures realistic human forms in monocular images and is used to infer 2D body shape and pose under clothing. We also use the coefficients of the eigen clothing to recognize different categories of clothing on dressed people. The method is evaluated quantitatively on synthetic and real images and achieves better accuracy than previous methods for estimating body shape under clothing.

1. Introduction

Two-dimensional models of the human body are widely used in computer vision tasks such as pose estimation, segmentation, pedestrian detection and tracking. Such 2D models offer representational and computational simplicity and are often preferred over 3D models for applications involving monocular images and video. These models typically represent the shape of the human body coarsely, for example as a collection of articulated rectangular patches **[43, 59, 68, 79]**. None of these methods explicitly models how clothing influences human shape. Here we propose a new fully generative 2D model that decomposes human body shape into two components: 1) the shape of the naked body and 2) the shape of clothing relative to the underlying body. The naked body shape is represented by a 2D articulated Contour Person (CP) **[47]** model that is learned from examples. The CP model realistically represents the human form but does not model clothing. Given training examples of people in clothing with known 2D body shape, we compute how clothing deviates from the naked body to learn a low-dimensional model of this deformation. We call the resulting

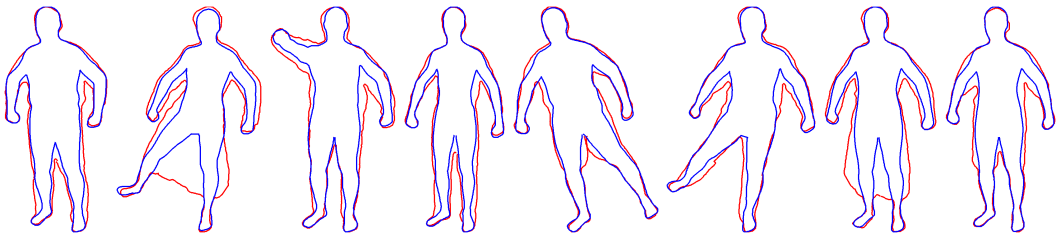


FIGURE 4.1. **Samples from the Dressed Contour Person model.** Different body shapes and poses (blue) are dressed in different types of eigen clothing (red).

generative model the *Dressed Contour Person* (DCP) and samples from this model are shown in Fig. 4.1.

The DCP model can be used just like previous models for person detection, tracking, etc. yet it has several benefits. The key idea is to separate the modeling of the underlying body from its clothed appearance. By explicitly modeling clothing we infer the most likely naked body shape from images of clothed people. We also solve for the pose of the underlying body, which is useful for applications in human motion understanding. The learned model accurately captures the contours of clothed people making it more appropriate for tracking and segmentation. Finally, the model supports new applications such as the recognition of different types of clothing from images of dressed people. Recently, Yamaguchi *et al.* parse clothing types in fashion photos [**120**].

There are several novel properties of the DCP model. First we define *eigen clothing* to model deformation from an underlying 2D body contour. Given training samples of clothed body contours, where the naked shape of the person is known, we align the naked contour with the clothing contour to compute the deformation. The eigen-clothing model is learned using principal component analysis (PCA) applied to these deformations. A given CP model is then "clothed" by defining a set of linear coefficients that produce a deformation from the naked contour. This is illustrated in Fig. 4.1.

There is one problem, however, with this approach. As others have noted, clothing generally makes the body larger [**7, 57**]. A standard eigen-model of clothing could generate "negative clothing" by varying the linear coefficients outside the range of the training samples. While non-negative matrix factorization could be used to learn the clothing model, we show that a simple prior on the eigen coefficients addresses the issue. In particular, we show that the eigen coefficients describing clothing deformations are not Gaussian and we model them using Beta distributions that capture their asymmetric nature.

We also demonstrate the estimation of a person's 2D body shape under clothing from a single image. Previous work on estimating body shape under clothing has either used multiple images [**7**] or laser range scan data [**57**]. These previous approaches also did not actually model clothing but rather tried to ignore it. Both of the above methods try to fit a naked body that lies inside the

measurements (images or range scans) while strongly penalizing shapes that are "larger" than the observations. We show that there is a real advantage to a principled statistical model of clothing. Specifically we show accuracy in estimating naked body shape that exceeds that of Balan and Black [**7**], while only using one uncalibrated image as opposed to four calibrated views.

Finally we introduce a new problem of clothing category recognition. We show that the eigen coefficients of clothing deformations are distinctive and can be used to recognize different categories of clothing such as long pants, skirts, short pants, sleeveless tops, etc. Clothing category recognition could be useful for person identification, image search and various retail clothing applications.

There are several major challenges: 1) Clothing obscures the true body shape which typically makes shape estimation biased to bigger size. 2) Most 2D body models are cardboard-like "puppet" models which limits the realism of 2D body representation. We address the first challenge by explicitly learning an eigen clothing deformation model, which captures statistical distribution of deformations that clothing can apply on top of body. For the second challenge, we employ a recently proposed 2D articulated Contour Person(CP) [**47**] model to realistically represent underlying 2D body.

Detailed estimation of body shape has numerous applications particularly in tracking, surveillance and forensic video analysis. Although in 99% cases people wear clothes, most of the works still focus on minimal clothed people due to the difficulty of dealing with clothing. Recently several works on shape estimation under clothing have been proposed and almost all of them fall into 3D domain [**7**][**57**]. They resolve the ambiguity from clothing by assuming a 3D body model and make the body estimation conform to the parametric shape space.

Since clothing deformation only applies on the underlying body we also employ a realistic 2D body model to represent body underneath. The recently proposed CP [**47**] model factors deformations due to shape, part rotation, and viewpoint change and also provide nice segmentation of parts. Based on CP model and eigen clothing deformation model, we can define a two layers deformation process, in which the first layer being the transformation from a body template to a plausible body under clothing(achieved by CP), and the second layer being the transformation from a plausible body to clothing(achieved by eigen clothing deformation). We finally solve an inference problem after which we get both body deformation and clothing deformation parameters. The body deformation parameters enable us to reconstruct the underlying naked body which is our main focus. The clothing deformation coefficients can surprisingly serve as clothing type indicator which allows us to do clothing type classification.

In summary, the key contributions of this chapter include: 1) the first model of 2D eigen clothing; 2) a full generative 2D model of dressed body shape that is based on an underlying naked model

with clothing deformation; 3) the inference of 2D body shape under clothing that uses an explicit model of clothing; 4) shape under clothing in a single image; 5) avoiding "negative clothing" by modeling the skewed statistics of the eigen-clothing coefficients; 6) the first shape-based recognition of clothing categories on dressed humans.

2. The Contour Person Model

We start with a Contour Person (CP) model [**47**], which is a low-dimensional, realistic, parameterized generative model of 2D human shape and pose. The CP model is learned from examples created by 2D projections of multiple shapes and poses generated from a 3D body model such as SCAPE [**4**]. The CP model is based on a template, T, corresponding to a reference contour that can be deformed into a new pose and shape. This deformation is parameterized and factors the changes of a person's 2D shape due to pose, body shape, and the parameters of the viewing camera. This factorization allows different causes of the shape change to be modeled separately. Let $B_T(\Theta) = (x_1, y_1, x_2, y_2, \ldots x_N, y_N)^T$ denote the parametric form of the CP, where N is the number of contour points and Θ is a vector of parameters that controls the deformation with respect to T. The CP model represents a wide range of 2D body shapes and poses, but only does so for naked bodies. Examples of such body contours, $B_T(\Theta)$, are shown in blue in Fig. 4.1. See Freifeld *et al.* [**47**] for mathematical details.

The CP model may contain internal or occluded portions of the body contour. However, here our clothing training data consists only of silhouettes with no internal structure. Consequently, we restrict the poses we consider and define $B_T(\Theta)$ to be a CP model corresponding to a bounding body contour without holes. In future work, we will generalize the DCP model to take advantage of the ability of the CP to accommodate self occlusions.

3. Clothing Model

We directly model the deformation from a naked body contour to a clothed body by virtually "dressing" the naked contour with clothing. The generative nature of this paradigm not only simulates the process in the real world but also allows us to synthesize novel clothed bodies. We start with a training set (described below) of clothing outlines and corresponding naked body outlines underneath. The CP model is first fit to the naked body outline to obtain a CP representation. For each point on the CP, we compute the corresponding point on the clothing outline (described below) and learn a point displacement model using PCA just like Active Shape Model [**30**]. We further learn a prior over the PCA coefficients using a Beta distribution to prevent infeasible displacements (i.e. "negative clothing").

The DCP model can be thought of as having two "layers" that decouple the modeling of body pose and shape from the modeling of clothing. The first layer generates a naked body deformation from the template contour and the second layer models clothing deformation from this deformed naked contour. The first layer is the CP model, which is compositional in nature and based on deformations of line segments (see [**47**]). The second layer, described here, is simpler and is based directly on displacements of contour points. This is possible since being the last layer of deformation, the compositional structure is less important. The layered representation is desirable because it allows constraints to be imposed independently on the body and the clothing. For example, in tracking applications, one may assume the body shape is constant while the pose and clothing shape changes.

3.1. Data sets. Our method requires training contours of people in clothing for which we know the true underlying naked body shape. We describe two such training sets below.

Synthetic data set. Synthetic data provides ground truth body shapes that enable accurate quantitative evaluation. We use 3D body meshes generated from the CAESAR database [**87**] (SAE International) of laser range scans and dress these bodies in simulated clothing (Fig. 4.2). We used 60 male and 100 female bodies spanning a variety of heights and weights and use commercial software (OptiTex International, Israel) to generate realistic virtual clothing. The clothing simulation uses a physical model to drape the clothing on the body producing realistic effects corresponding to seams, gravity, pose and different materials. The clothing simulation produces a secondary 3D mesh that lies on top of the underlying body mesh by construction. Given a particular camera view, we project the body mesh into the image to extract the body outline and do the same for the combined body and clothing meshes. This provides a pair of training outlines.

For the synthetic dataset we restrict the clothing to a single type (Army Physical Training Uniforms) but in different sizes, as appropriate for the body model. While narrow, this dataset provides nearly perfect training data and ground truth for evaluation. The next dataset expands the range of clothing and pose but with a smaller sample of bodies and real imagery.

Real data set. To model real people in real clothing we use the dataset described by Balan and Black in [**7**] (Fig. 4.2) which contains images of 6 subjects (3 males, 3 females) captured by 4 cameras in two conditions: 1) the "naked condition" in which the subjects wear tight fitting clothing; 2) the "clothed condition" in which they wear different types of "street" clothing. The dataset contains four synchronously captured images of each subject, in each condition, in a fixed set of 11 postures. For each posture the subjects are dressed in 6-10 different sets of clothing (trials). Overall there are 47 trials with a total of 235 unique combinations of people, clothing and poses.

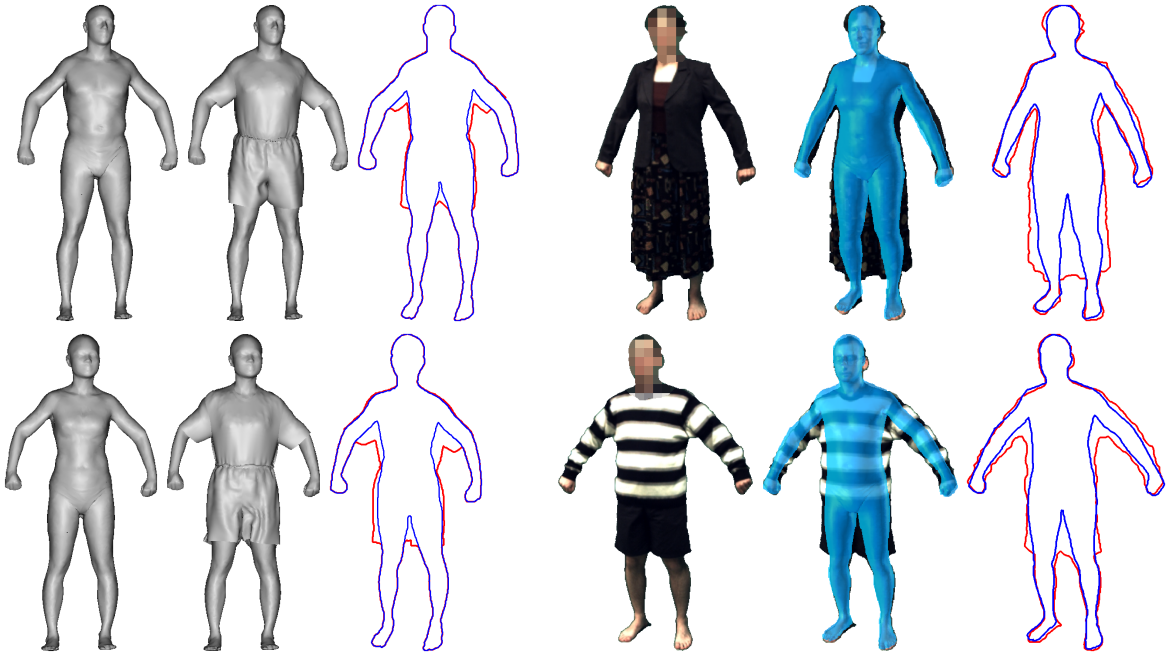


FIGURE 4.2. **Example training data.** Left: Pairs of synthetic 3D bodies, unclothed and clothed. Projecting the silhouette contours of these pairs produces training contours. Right: Training contours derived from multi-camera data (see text); the estimated ground truth 3D body is shown as a translucent overlay.

For each image of a dressed person, we use standard background subtraction [**7**] to estimate the clothed body silhouette and extract the outline. To obtain the underlying naked body contours, we fit a 3D parametric body model using the 4 camera views in the naked condition [**7**]. We take this estimated 3D body shape to be the true body shape. We then hold this body shape fixed while estimating the 3D pose of the body in every clothing trial using the method of [**7**] which is robust to clothing and uses 4 camera views.

The process produces a 3D body of the "true" shape, in the correct pose, for every trial. We project the outline of this 3D body into a selected camera view to produce a training 2D body contour. We then pair this with the segmented clothed body in that view. Note that the fitting of the 3D body to the image data is not perfect and, in some cases, the body contour actually lies outside the clothing contour. This does not cause significant problems and this dataset provides a level of realism and variability not found in the synthetic dataset.

3.2. Correspondence. Given the naked and clothed outlines defined above, we need to know the correspondence between them. Defining the correspondence between the naked outline and the clothing outline is nontrivial and how it is done is important. Baumberg and Hogg, for example, model the outline of pedestrians (in clothing) using PCA [**13**]. In their work, correspondence is

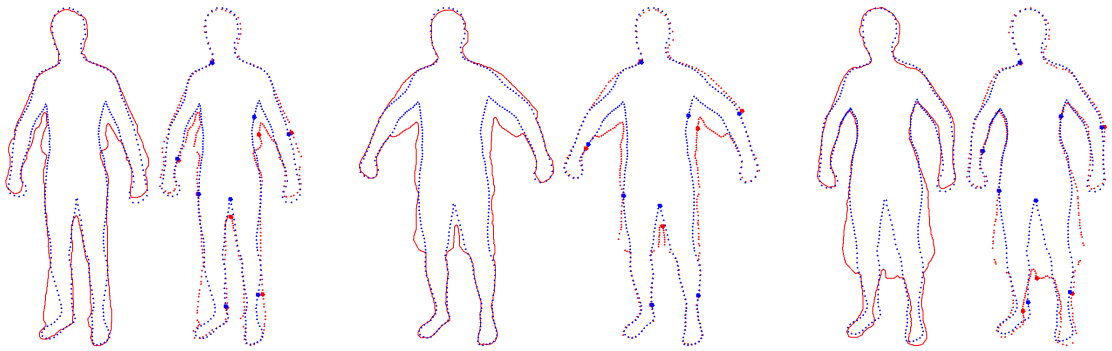
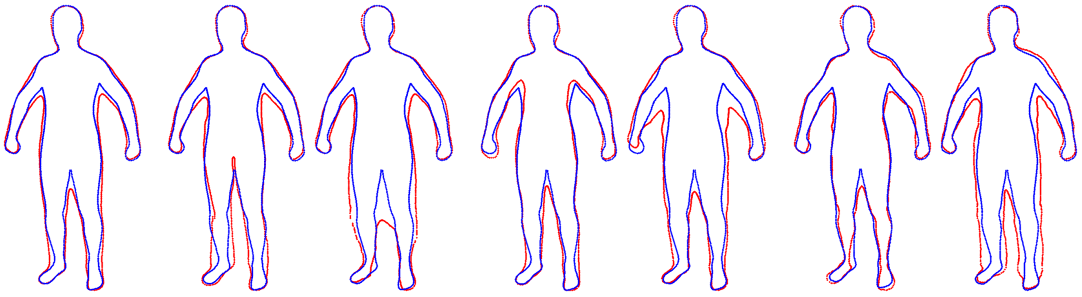


FIGURE 4.3. **Correspondence between body and clothing contours.** In each pair: the left image shows the sample points of the body contour in blue and the densely sampled clothing contour in red. The right image shows the final sub-sampled clothing contour with a few matching points highlighted as larger dots. Nearby dots illustrate corresponding points (in some cases they are on top of each other).

simply computed by parameterizing all training contours with a fixed number of evenly sampled points. Incorrect correspondence (i.e. sliding of points along the contour) results in eigen shapes that are not representative of the true deformations of the contours.

Instead, we start with the trained parametric CP representation $B_T(\Theta)$ and optimize it to fit the 2D naked body that minimizes the difference between the CP silhouette and the naked body silhouette. This gives a CP representation of the naked body that consists of $N = 1120$ points. We then densely sample M points on clothing outline, where $M >> N$ and select the N clothing contour points that best correspond to the CP points. During matching, the relative order of the points is maintained to guarantee the coherence of the match. Let the CP contour be represented by a list of points $P = \{p_1, p_2, ..., p_N\}$ and let the sampled clothing outline be represented by $Q = \{q_1, q_2, ..., q_M\}$. We pick a subset of N points $G = \{q_{k_1}, q_{k_2}, ..., q_{k_N}\}$ from Q that minimizes $\sum_{i=1}^{N} \|p_i - q_{k_i}\|^2$ over the indices k_i such that the ordering, $k_r < k_s$, is preserved for $1 \leq r < s \leq N$. We use the dynamic programming method proposed in **[82]**. Example alignments are shown in Fig. 4.3.

3.3. Point displacement model. We use a vector $\hat{G} = (x_1, y_1, \ldots, x_N, y_N)^T$ to represent the point list G and now we have $B_T(\Theta)$ for the naked body contour and $\hat{G}$ for clothing contour, both of which have N corresponding points. Since $B_T(\Theta)$ has known part segmentation and G is obtained in such a way that no crossed matches are allowed, we can weakly assume that points on G also have rough part association as shown in Fig.4.3. The clothing displacement for a particular training example, i is then defined by $\delta_i = \hat{G}_i - B_T(\Theta_i)$. We collect the training displacements into



(a) mean (b) PC1 (c) PC2 (d) PC4

FIGURE 4.4. **Eigen clothing.** The blue contour is always the same naked shape. The red contour shows the mean clothing contour (a) and ±3 std from the mean for several principal components (b)-(d).

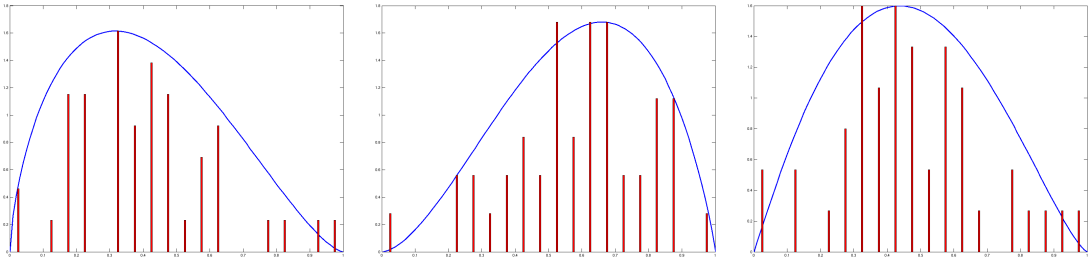
a matrix and perform PCA. We take the first 8 principal components accounting for around 90% of the variance to define the eigen-clothing model. Figure 4.4 shows the mean and first few clothing eigenvectors for the real data set. This illustrates how the principal components can account for various garments such as long pants, skirts, baggy shirts, etc. Note that simply varying the principal components can produce "negative clothing" that extends inside the blue body contour. We address this in the following section.

Using this model we generate new body shapes in new types of clothing by first sampling CP parameters Θ to create a naked body contour $B_T(\Theta)$ and then using the following equation to generate a clothed body

$$C(\Theta, \eta) = B_T(\Theta) + \Delta_{mean} + \sum_{i=1}^{N_\eta} \eta_i \cdot \Delta_i \tag{3}$$

where N_η is the number of eigenvectors used, the η_i's are coefficients, Δ_{mean} is the mean clothing displacement, and Δ_i is the i^{th} eigen-clothing vector.

3.4. Prior on point displacement. Although the PCA model captures clothing deformation, it allows point displacements in both inward and outward directions, which violates our assumption that clothing only makes the body appear bigger. This assumption is confirmed by examining the statistics of the linear eigen coefficients in the training data. Figure 4.5 shows several such distributions, which may be skewed or symmetric. In particular we find that coefficients for the principal components that capture the most variance are typically positively or negatively skewed while coefficients for the lower-variance components tend to be more normally distributed. The first few eigenvectors capture the gross clothing displacements, which are always away from the body. Of course clothing also exhibits many fine details and folds and these are captured by the lower



(a) PC1 (b) PC2 (c) PC8

FIGURE 4.5. **The statistics of clothing displacements.** Example histograms and Beta distribution fits to linear eigen-clothing coefficients. Note the skew that results from the fact that clothing generally makes the body appear larger.

variance eigenvectors. These "detail" eigenvectors modify the main clothing contour both positively and negatively (e.g. out and in) and hence tend to have more symmetric statistics.

Based on the observation of natural clothing statistics, we learn a prior on the PCA coefficients to penalize infeasible clothing displacements. We make the assumption that the eigenvectors are independent (not necessarily true since the data is not Gaussian) and independently model a prior on each coefficient using a Beta distribution. The Beta distribution is defined on $[0, 1]$ and is characterized by two parameters α and β that can be varied to capture a range of distributions including positively skewed, negatively skewed and symmetric shapes:

$$\textbf{Beta}(x; \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}. \tag{4}$$

Given L training body/clothing pairs, and the associated clothing displacements, we project each displacement onto the PCA space to obtain coefficient η_m^l for instance l, ($l \in [1, L]$), on eigenvector m. We normalize $\eta_m^1, \eta_m^2, ..., \eta_m^L$ to $[0, 1]$ to obtain $\tilde{\eta}_m^1, \tilde{\eta}_m^2, ..., \tilde{\eta}_m^L$ and fit these with the Beta distribution. The probability of observing a normalized coefficient $\tilde{x}_m$ for the m^{th} eigenvector is given by $\textbf{Beta}(\tilde{x}_m, \alpha_m, \beta_m)$, where α_m and β_m are the estimated parameters of the Beta distribution. If we observe a coefficient during testing that is out of the scope of the training coefficients, we threshold it to be between the minimal and maximal value in the training set and normalize it to compute its prior probability. If thresholded, however, we still use the original value to reconstruct the shape. Figure 4.5 shows how the Beta function can represent a variety of differently shaped distributions of clothing displacement coefficients.

3.5. Inference. The inference problem is to estimate the latent variables Θ and η by only observing a single image of a person in clothing. We define a likelihood function in terms of silhouette overlap. We adopt a generative approach in which $C(\Theta, \eta)$, the clothed body (Eq. 3), defines an estimated silhouette, $S^e(C(\Theta, \eta))$, and compare it with the observed image silhouette S^o. We follow

[7] and define the asymmetric distance between silhouettes S^r and S^t as $d(S^r, S^t) = \frac{\sum_{i,j} S^r_{i,j} H_{i,j}(S^t)}{\sum S^r_{i,j}}$, where $S^r_{i,j}$ is a pixel inside silhouette S^r and $H_{i,j}(S^t)$ is a distance function which is zero if pixel (i, j) is inside S^t and is the distance to the closest point on the boundary of S^t if it is outside.

We then define the data term as the following symmetric data error function

$$E_{data}(\Theta, \eta) = d(S^e(C(\Theta, \eta)), S^o) + d(S^o, S^e(C(\Theta, \eta))). \tag{5}$$

The first part of Eq. 5 penalizes the regions of the synthesized clothing instance $S^e(C(\Theta, \eta))$ that fall outside the observed clothing silhouette S^o, and the second part makes $S^e(C(\Theta, \eta))$ explain S^o as much as possible.

E_{data} alone is not sufficient to estimate Θ and η correctly, because there are ambiguities in estimating smaller bodies with larger clothing and larger bodies with smaller clothing. As was mentioned in Sec. 3.4, we use the Beta prior to penalize unlikely displacements. Recall that $\tilde{\eta}_m$ represents the normalized coefficient for the m^{th} basis. The prior term is defined as

$$E_{prior}(\eta) = -\sum_m log(\mathbf{Beta}(\tilde{\eta}_m, \alpha_m, \beta_m)). \tag{6}$$

The final energy function we minimize is

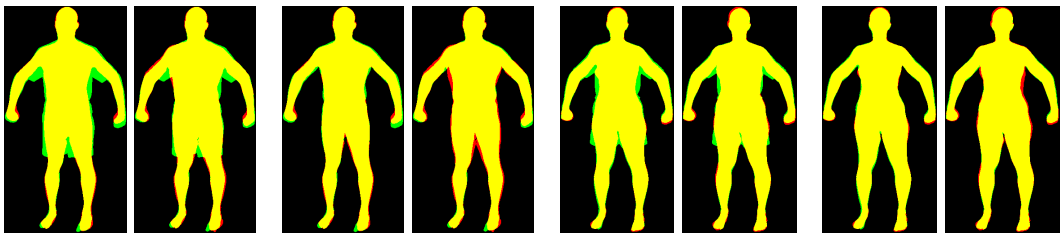
$$E(\Theta, \eta) = E_{data}(\Theta, \eta) + \lambda E_{prior}(\eta) \tag{7}$$

where λ indicates the importance of the prior. Problems with "negative clothing" and clothing that is unusually large are avoided due to the prior. Optimization is performed using MATLAB's `fminsearch` function.

4. Results

We consider two novel applications of the proposed method. The first is the estimation of 2D body shape under clothing given a single image of a clothed person. The second is the recognition of different clothing categories by classifying the estimated clothing deformation parameters. We evaluate our model on three tasks: body shape estimation from synthetic data, body shape estimation from real data, and clothing type classification from real data. We compare the results of the first two tasks with approaches that do not explicitly model clothing deformation.

Body estimation under clothing from synthetic data. We use the synthetic dataset of 60 males and 100 females, in and out of synthetic clothing, as described above. We randomly select 30 males and 50 females as the training set and the remaining 80 bodies as the test set. A gender-specific CP model is learned for males and females separately while a gender-neutral eigen model is learned for clothing deformations. We estimate the underlying bodies for the test samples using the



(a) male (b) compared to GT (c) female (d) compared to GT

FIGURE 4.6. **Synthetic data results.** For each pair of images, the DCP result is on the left and NM result is on the right. The first pair shows an estimated body silhouette (red) overlaid on the clothing silhouette (green); overlapped regions are yellow. The second pair shows the estimated body (red) overlaid on the ground truth (GT) body (green). The third and fourth pairs show the same but for a female. NM typically overestimates the size of the body.

Dressed Contour Person (DCP) and measure the estimation error as

$$err(S^{EST}, S^{GT}) = \frac{\sum_{i,j} |S^{EST}_{i,j} - S^{GT}_{i,j}|}{2 \sum_{i,j} S^{GT}_{i,j}} \tag{8}$$

where S^{EST} is a silhouette corresponding to the estimated naked body contour and S^{GT} is the ground truth underlying naked body silhouette. The results of DCP are also compared with a naive method (NM) in which we simply fit the CP model to the image observations of clothed people. As in [**7**], the NM attempts account for clothing by penalizing contours more if the estimated body silhouette falls outside of the clothing observation than if it does not fully explain the clothing observation. The average estimation errors obtained with NM for males and females are 0.0456 and 0.0472 respectively while DCP achieves 0.0316 and 0.0308. Our DCP model improves accuracies over NM by 30% (male) and 35% (female) relatively. While the synthetic dataset has only one clothing type, the bodies span a wide range of shapes. The results show a principled advantage to modeling clothing deformation compared with ignoring clothing. Figure 4.6 shows some representative results from the test set.

Body estimation under clothing from real data. Figure 4.7 shows 8 different poses from the real dataset (Sec. 3.1). For each pose there are 47 examples having unique combinations of subjects and clothing types. Since the number of body/clothing pairs is limited in each pose, we use a leave-one-out strategy where we estimate the body of instance i using the eigen-clothing model learned from all remaining 46 instances excluding i. We use DCP to estimate the underlying body shape for a total of $47 * 8 = 376$ instances (Fig. 4.7) and compare the results with two other methods: 1) NM described in the previous experiment; and 2) "Naked People estimation in 3D"(NP3D) proposed

TABLE 4.1. Comparison on real data: DCP, NM, and NP3D methods (see text).

Method, AEE	Pose1	Pose2	Pose3	Pose4	Pose5	Pose6	Pose7	Pose8	Average
DCP	0.0372	0.0525	0.0508	0.0437	0.0433	0.0451	0.0503	0.0668	0.0487
NP3D	0.0411	0.0628	0.0562	0.0484	0.0494	0.046	0.0472	0.0723	0.0529
NM	0.0865	0.0912	0.0846	0.0835	0.0877	0.0921	0.0902	0.1184	0.0918
Significance (p-value)									
DCP vs NP3D	0.38	0.13	0.34	0.46	0.36	0.89	0.66	0.54	0.07
DCP vs NM	6.4e-7	4.9e-4	2.1e-4	2.1e-4	6.7e-8	1.0e-5	1.0e-6	2.3e-4	9.9e-17

in [**7**]. Since DCP and NM are 2D methods using a 2D CP model, they only use one camera view. NP3D, however, estimates a 3D body model from four camera views [**7**]. To compare with NP3D we project the estimated body from NP3D into the image corresponding to the camera view used by our method.

Table 4.1 shows the Average Estimation Error (AEE) computed by averaging $err(\cdot, \cdot)$ (Eq. 8) over the 47 instances for each pose (or over all poses in the last column). Figure 4.8 shows details of the fitting results. We find that DCP has lower error than both NM and NP3D. In the case of NM these differences are statistically significant (paired t-test, $p < 0.05$) for all poses and in the aggregate. While DCP has lower error than NP3D in all but one pose, and lower error overall, the differences are not significant at the $p < 0.05$ level. Recall that NP3D is using significantly more information. These results suggest that using a learned statistical model of clothing is preferable to simply trying to ignore clothing [**7**].

Clothing category recognition. We now ask whether the clothing deformation coefficients contain enough information about clothing shape to allow the classification of different types of clothing. Note that this task involves recognizing clothing *on the body* as it is worn by real people. We separate upper clothing and lower clothing and define 7 different categories (as color coded in Fig. 4.9).

We use a simple nearest neighbor (NN) classifier with Euclidean distances computed from the coefficients along the first 8 principal components. Since we have a limited number of clothing instances (47) for each pose, we use a leave-one-out strategy and assume that we know the categories of all the instances except the one we are testing. Each instance is then assigned a category for both upper clothing and lower clothing based on its nearest neighbor. Classification results are shown in Fig. 4.9 along with chance performance for this task.

FIGURE 4.7. **Sample DCP results of estimated body shape overlaid on clothing.** The estimated body contour and synthesized clothing contour are depicted by blue and red outlines respectively. Body shape is the transparent region encompassed by the body contour. Results are shown for a variety of poses (left to right: 1-8) and viewing directions.

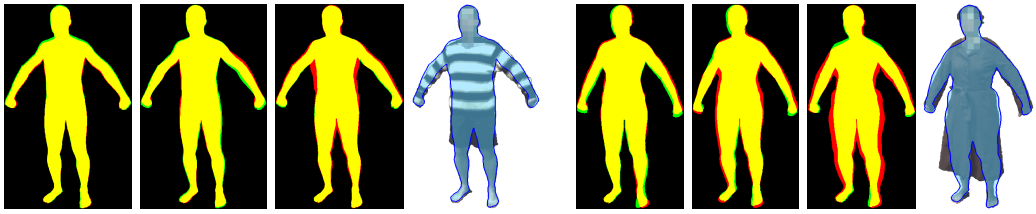


FIGURE 4.8. **Comparisons of DCP, NM, and NP3D.** For each group of images: the first 3 images (left to right) show overlap of the estimated silhouette (red) and the ground truth silhouette (green) for DCP, NP3D, and NM (yellow is overlap); the 4th image shows the body estimated by NM overlaid on a clothing image. NM overestimates body shape as expected.

5. Discussion

We have presented a new generative model of the 2D human body that combines an underlying Contour Person representation of the naked body and layers on top of this a clothing deformation

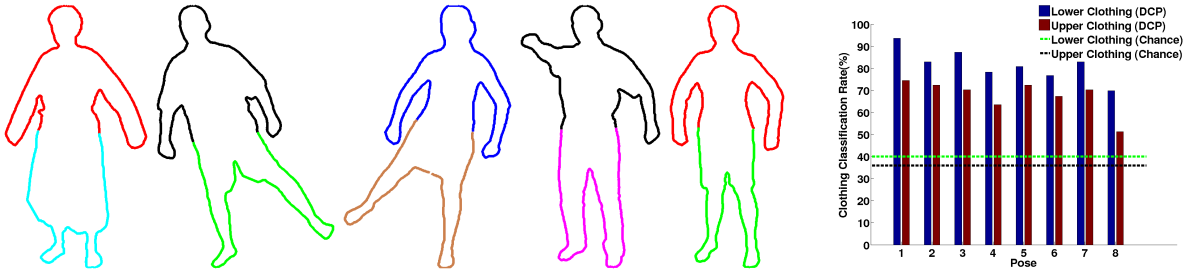


FIGURE 4.9. **Color coded clothing type.** We consider three types of upper clothing: long sleeves (red), short sleeves (black) and sleeveless tops (blue) and four types of lower clothing: short pants (green), long pants (magenta), short skirts (coffee), and long skirts (cyan). Classification results for the 7 clothing types in all 8 poses are shown in the right figure compared to "Chance".

model. This goes beyond previous work to learn an eigen model of clothing deformation from examples and defines a prior over possible deformations to prevent "negative clothing". While previous work has examined 3D body models captured with multiple cameras or laser range scanners, we argue that many computer vision applications use 2D body models and that these applications will benefit from a more realistic generative model of clothed body shape. By modeling clothing deformations we estimate 2D body shape more accurately and even out-perform previous multi-camera systems on estimating shape under clothing. Finally we define a new problem of clothing category recognition on the human body and show how the coefficients of the estimated eigen clothing can be used for this purpose. This new dressed person model is low dimensional and expressive, making it applicable to many problems including 2D human pose estimation, tracking, detection and segmentation.

Our method does have some limitations. The method assumes there is a correspondence between body contour points and clothing contour points. When there is significant limb self occlusion, the clothing silhouette may not contain features that *correspond to* that limb. Dealing with significant self occlusion is future work. Also, here we assume that the rough viewing direction (frontal or side) and rough pose are known.

There are several directions for future work. First, we plan to model clothing deformation as a function of human movement. This may require a model more like the original CP model in which deformations are defined as scaled rotations of contour line segments [**47**]. This representation allows the factoring of contour changes into different deformations that can be composed. Second, we will explore what we call "eigen separates"; that is, separate eigen models for tops and bottoms as well as for hair/hats and shoes. Having separate eigen spaces reduces the amount of training data required to capture a wide range of variations. Finally we plan to extend these methods to model 3D clothing

deformations from a 3D body model. Again data acquisition for 3D clothed and unclothed training data is very difficult, and we plan to use realistic physics simulation of clothing.

CHAPTER 5

DRAPE: Dressing Any Person

This chapter is built upon the original work [52]. We describe a complete system for animating realistic clothing on synthetic bodies of any shape and pose without manual intervention. The key component of the method is a model of clothing called DRAPE (DRessing Any PErson) that is learned from a physics-based simulation of clothing on bodies of different shapes and poses. The DRAPE model has the desirable property of "factoring" clothing deformations due to body shape from those due to pose variation. This factorization provides an approximation to the physical clothing deformation and greatly simplifies clothing synthesis. Given a parameterized model of the human body with known shape and pose parameters, we describe an algorithm that dresses the body with a garment that is customized to fit and possesses realistic wrinkles. DRAPE can be used to dress static bodies or animated sequences with a learned model of the cloth dynamics. Since the method is fully automated, it is appropriate for dressing large numbers of virtual characters of varying shape. The method is significantly more efficient than physical simulation with the major run-time cost being the solution of linear systems.

1. Introduction

Clothed virtual characters in varied sizes and shapes are necessary for film, gaming, and on-line fashion applications. As with real people, virtual characters come in huge variety of sizes. Dressing such characters is a significant bottleneck, requiring manual effort to design clothing, position it on the body, and simulate its physical deformation. DRAPE handles the problem of automatically dressing realistic human body shapes in clothing that fits, drapes realistically, and moves naturally. Recent work models clothing shape and dynamics **[37, 45, 111]** but has not focused on the problem of fitting clothes to different body shapes.

Physics Based Simulation (PBS) **[12, 28, 21]** is widely used to model the complex behavior of cloth and can produce highly realistic clothing simulations. An extensive survey of cloth simulation can be found in **[29]**. PBS, however, requires high resolution meshes to represent details (folds and wrinkles), complicated non-linear functions to capture the physical properties of fabric, and time-consuming collision handling to achieve realism **[37, 29]**. For acceptable visual effects, the clothing mesh and the body mesh may contain tens of thousands of triangles, making PBS computationally

expensive. Moreover, the results of physical clothing simulation are specific to a particular body model. and do not readily generalize to new body shapes. Dressing bodies of different shapes requires a separate simulation for every body shape. Additionally, a fundamental problem confronting garment designers is the nontrivial task of choosing clothing sizes and initializing clothing simulation on 3D characters [**29**]. Pattern makers may need to redesign the 2D patterns based on the anthropometric measurements for characters of different shapes. An improperly chosen size can also lead to failure of the simulation. This poor reusability is one of the factors that hinders richer clothing animation [**29**].

Our method learns a deformable clothing model that automatically adapts to new bodies. Once the DRAPE model is learned for a particular type of clothing, we can dress any body in that clothing. Unlike the PBS methods, users do not need to choose proper sizes and initial positions of cloth pieces before clothing fitting. The model will reshape the garment to fit the body and "drape" it automatically. This is done by learning a linear regression the from underlying body shape parameters to the clothing model parameters. assuming that the garments are reasonably tight fitting. Pattern design is completely separated from the process of dressing bodies and can be done by professional pattern makers before training the model. Therefore, users do not need to know about pattern design, enabling much broader applications of clothing animation.

Here we use SCAPE [**4**] to represent human bodies of different shapes in different poses. We learn separate SCAPE models for men and women using approximately 2000 aligned laser range scans of different people in a single pose [**87**] and additional scans of one individual in roughly 70 poses. This result is an expressive 3D model with parameters controlling a wide range of body shapes ($\vec{\beta}$) and poses ($\vec{\theta}$). The model is sufficiently expressive to represent a wide range of body shapes and poses. using independent parameters, allowing us to repose a single body and put bodies of different shapes in the same pose. We assume that we have SCAPE parameters for all bodies used for training and for dressing. We don't lose much generality here, as we can optimize the SCAPE parameters to represent most poses and body shapes accurately.

For this study, we designed and graded patterns for five common clothing types: T-shirts, shorts, skirts, long sleeved shirts, and long pants [**14**]. These patterns are graded to allow us to generate multiple sizes of each type which cover the most common garments that people tend to wear. The T-shirt is used for illustration in Figure 5.1. The complete system (Figure 5.1) has three components:

1. Training data: The *shape training set* consists of SCAPE bodies with different shapes in the same pose. The *pose training set* contains a single body shape moving through sequences of poses. For each training body shape, we manually choose a size for each garment and dress the body using PBS (Figure 5.1, row 2); this becomes our training data.

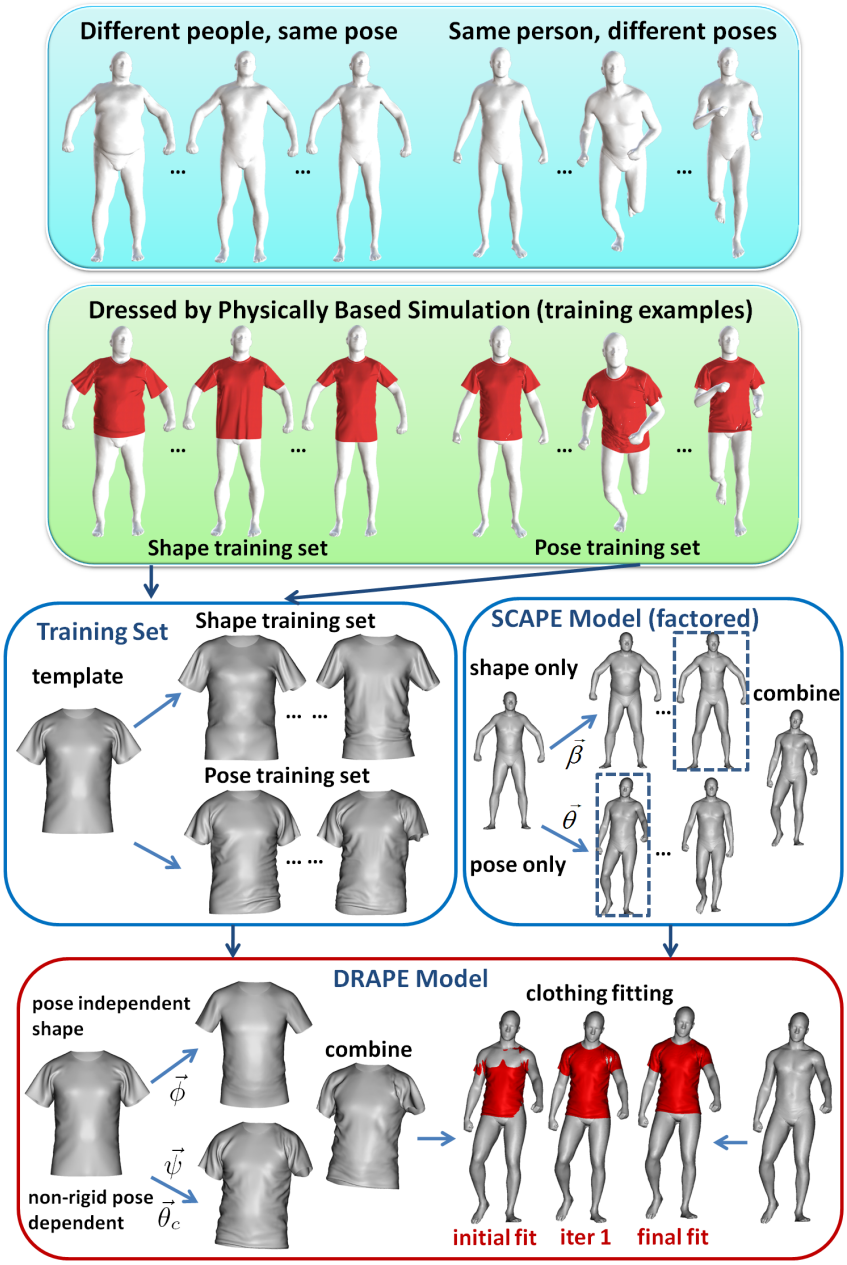


FIGURE 5.1. **Overview.** We dress the bodies in the shape and pose training sets using PBS to generate clothing examples for learning. DRAPE factors rigid pose, pose-independent shape variation, and pose-dependent wrinkle deformation. The SCAPE model is used to represent the underlying naked body. Given an input body, an appropriate clothing configuration is generated according to the body pose and shape. The clothing fitting process eliminates cloth-body interpenetration to create realistic animations.

2. Learned clothing deformation model: For each garment, we learn a *factored clothing model* that represents: *i*) rigid rotation, $\vec{\theta}_c$, of cloth pieces, e.g. the rotation of a sleeve w. r. t. the torso; *ii*) pose-independent clothing shape variations, $\vec{\phi}$, that are linearly predicted from the underlying body shape, $\vec{\beta}$, (learned from the shape training set); and *iii*) pose-dependent non-rigid deformations, $\vec{\psi}$, that are linearly predicted from a short history of body poses and clothing shapes (learned from the pose training set).

3. Virtual fitting: First, we map body shape parameters, $\vec{\beta}$, to clothing shape parameters, $\vec{\phi}$, to obtain a custom shaped garment for a given body. Clothing parts are associated with body parts and the pose of the body is applied to the garment parts by rigid rotation. The learned model of pose-dependent wrinkles is then applied. The custom garment is automatically aligned with the body and interpenetration between the garment and the body is removed by efficiently solving a system of linear equations.

Our model factors the change of clothing shape due to rigid limb rotation, pose-independent body shape, and pose-dependent deformations. As with the original SCAPE model, this allows us to combine deformations induced by different causes. The factored model can be learned from far less data than a model that simultaneously models clothing shape based on pose and body shape. In contrast, training a non-factored model (with pose, shape, and pose-dependent shape intermingled) would require a huge training set with many body shapes performing many motions. The factored model is an approximation that is sufficient for many applications and separates modeling body shape from pose-dependent shape. The method is ideal for applications where the body shape is not known in advance such as on-line virtual clothing try-on where every user has a unique 3D shape or where many different people must be animated (e.g. crowd scenes).

In summary, DRAPE makes the following contributions: 1) Synthetic bodies of any shape are automatically dressed in any pose. 2) A factored model of clothing shape models pose-dependent wrinkles separately from body shape. 3) The method dresses bodies completely automatically at run time. 4) Interpenetration is efficiently handled by solving a linear system of equations.

2. Simulating Clothing for Training

DRAPE models how clothing shape varies with underlying body shape and pose. Learning a DRAPE model requires a shape training set of clothing meshes fit to different body shapes and a pose training set with a single template body in multiple poses. Each type of clothing is represented as a triangulated mesh with a learned model of how that mesh deforms. For a given type of clothing, all training meshes must be in alignment regardless of pose or size. This makes the use of range scans of real clothing difficult, since the number of vertices will vary and one would have to compute

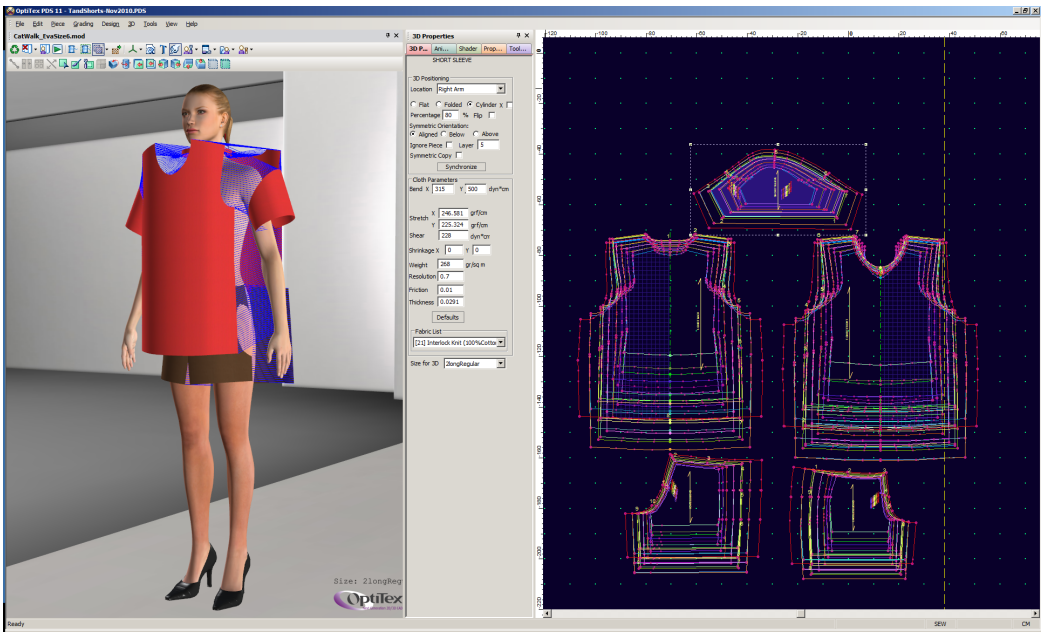


FIGURE 5.2. **Pattern design.** This screen-shot from a commercial pattern design software system (OptiTex) shows graded patterns for the T-shirt and shorts, with the grade points highlighted as purple dots. Some pieces, such as the sleeves, may be reused for both left and right sides. The center panel controls the parameters of the cloth simulation. On the left, the initial cloth placement is shown with the blue lines indicating the points to be stitched during simulation.

the alignment of the training samples which is not necessarily well defined. Consequently we use simulated clothing to generate the DRAPE training data. To prepare the training sets, we created 2D graded patterns for T-shirts, shorts, long-sleeved shirts, long pants, and skirts using a commercial design and 3D simulation software (OptiTex International, Israel). Without loss of generality we will use a T-shirt to illustrate the procedures for data generation. Figure 5.2 illustrates this standard commercial design process while Figure 5.3 shows examples of the training garments.

First, a garment expert drafts 2D patterns for the T-shirt using a commercial design and 3D simulation software (OptiTex International, Israel), which include the major pieces such as the front, back, and sleeves (Figure 5.2). See Figure 5.2. These 2D patterns share the same "grading rules". A garment is defined in 2D by a number of "grading points" (purple points in Figure 5.2), which model different sizes [**14**]. The grading points can be thought of as special boundary points that characterize the sizes and curves of the 2D patterns. *Different sizes of the same garment are not simply scaled versions of each other.* Simulation of the garment first requires manually selecting the appropriate size pattern and positioning the clothing pieces in 3D. The commercial software then stitches the garment and performs a physics-based simulation. PBS is done here with

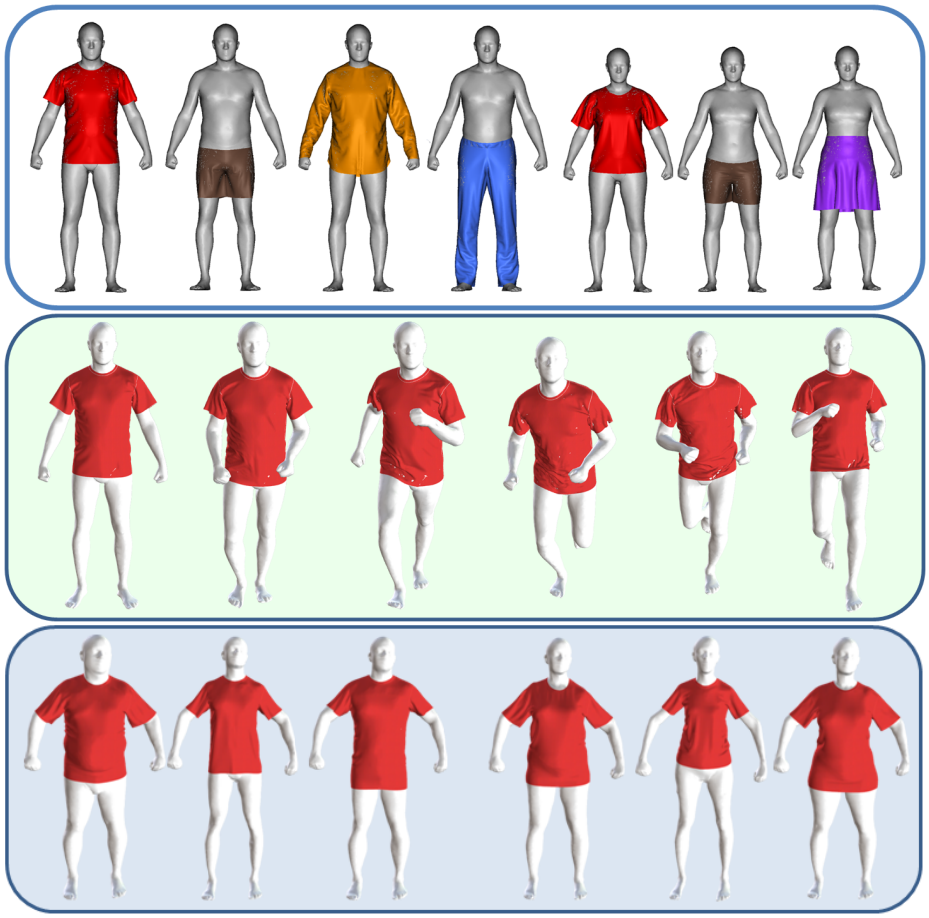


FIGURE 5.3. **Examples of training data.** (top) Clothing types used here (T-shirt, shorts, long-sleeved shirt, long pants, and skirt). (middle) Example T-shirts in the **pose** training set generated from a representative motion sequence. (bottom) Training examples of T-shirts on representative male and female body **shapes**.

OptiTex, but any simulation method could be used. For each size, OptiTex generates meshes with different numbers of vertices. Note, we select one 2D size as the template pattern and align all other 2D sizes to this with the help of the grading points (See AppendixA). The alignment procedure is straightforward because the grading points are in correspondence. After 2D alignment, all 3D meshes for each type of clothing are in full correspondence.

Defining a graded garment for a full range of sizes is a time consuming process requiring domain expertise. Note, however, that such patterns already exist for nearly every manufactured garment today. Our method does not provide automatic grading. Instead, we take care of the tedious process of pattern design and learn a parametric clothing model to provide the users with realistic automatic fitting with appropriate clothing size. That's why we want the *pattern design* to be done once and the users had better not deal with choosing the appropriate cloth sizes.

The training set for *pose-dependent clothing deformation* uses a single male and a single female avatar represented as SCAPE body models [**4**]; we use the average male and average female body shapes in the North American CAESAR data set [**87**]. Using 23 different motion capture sequences we animate the SCAPE avatars and use OptiTex to simulate the clothing in each frame (Figure 5.3 (middle)). These motion sequences capture a wide range of body poses and include walking, running, jumping, kicking, turning, bending the legs, and so on. For each sequence we simulate different clothing types: T-shirt, shorts, and skirt for the female and T-shirt, shorts, long sleeves, and long pants for the male. The clothing pose training sets consist of more than 3500 different poses with 4 male garments and 3 female garments, for a total of $3500 \times 7 = 24,500$ clothing instances. The model for each clothing type is learned separately. The learned DRAPE model is able to combine upper and lower-body clothing models to produce combinations not seen in training.

Finally, for the *shape training sets*, we used the SCAPE body model to generate 60 males and 60 females that span a wide variety of body shapes. Each body is in the same canonical "template" pose shown in Figure 5.3 (bottom). Similar to the pose training set, we simulated 4 male garments and 3 female garments resulting in $60 \times 7 = 420$ clothing instances in the shape training set.

3. DRAPE Model

DRAPE is trained using a set of aligned 3D clothing meshes, with T triangles and V vertices. The set contains a template mesh $\mathcal{X}$, a set of *pose* examples $\mathcal{Y} \in \mathbf{P}$, and a set of *shape* example meshes $\mathcal{Y} \in \mathbf{S}$. $\mathcal{X}$ is obtained by dressing an average body in the template pose. $\mathcal{Y} \in \mathbf{P}$ are obtained by running clothing simulation on one animated body. $\mathcal{Y} \in \mathbf{S}$ are obtained by running clothing simulation on different bodies with the same pose as the template. We consider males and females separately.

It is important to choose an appropriate representation for deformations between example meshes. Simple choices based on vertex coordinates or vertex displacements from the template mesh are problematic for separating deformations induced by different causes. Since factorization is a crucial property of the model, we use *shape deformation gradients* [**102, 4**] to represent deformations between meshes. This allows DRAPE to separate deformations induced by pose and shape and then combine the deformations together. We follow the formulation of SCAPE and present the notation here as it will be needed later. We refer the reader to the above referenced papers for details.

Deformation gradients are linear transformations that align corresponding triangles between a source mesh $\mathcal{X}$ and target mesh $\mathcal{Y}$ with the same topology. Suppose the vertices of a given triangle t in $\mathcal{X}$ are $(\vec{x}_{t,1}, \vec{x}_{t,2}, \vec{x}_{t,3})$ and the corresponding triangle in $\mathcal{Y}$ has the vertices $(\vec{y}_{t,1}, \vec{y}_{t,2}, \vec{y}_{t,3})$. We

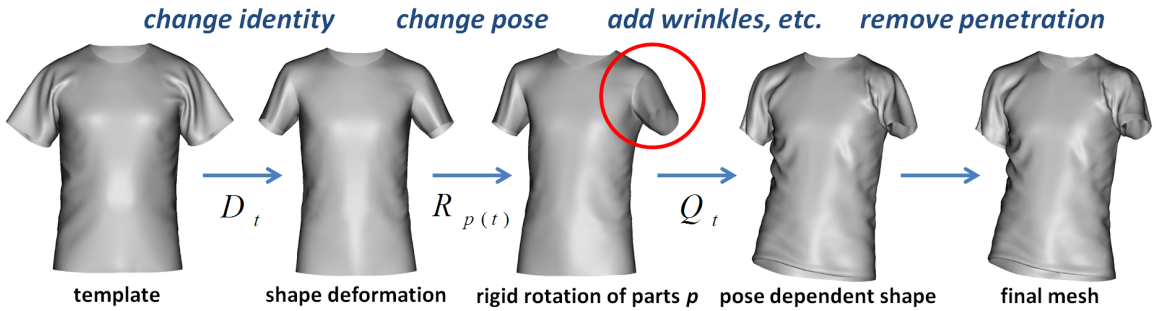


FIGURE 5.4. **DRAPE clothing deformation process.** (1) The template mesh is deformed to fit a new body shape. (2) The pose of the underlying body is used to apply a rotation to clothing parts. (3) Pose-dependent non-rigid deformation produces wrinkles learned from examples. (4) Vertices are moved locally to remove interpenetration with the underlying body mesh.

solve for a 3 by 3 linear transformation A_t such that

$$A_t[\Delta\vec{x}_{t,2}, \Delta\vec{x}_{t,3}, \Delta\vec{x}_{t,4}] = [\Delta\vec{y}_{t,2}, \Delta\vec{y}_{t,3}, \Delta\vec{y}_{t,4}], \tag{9}$$

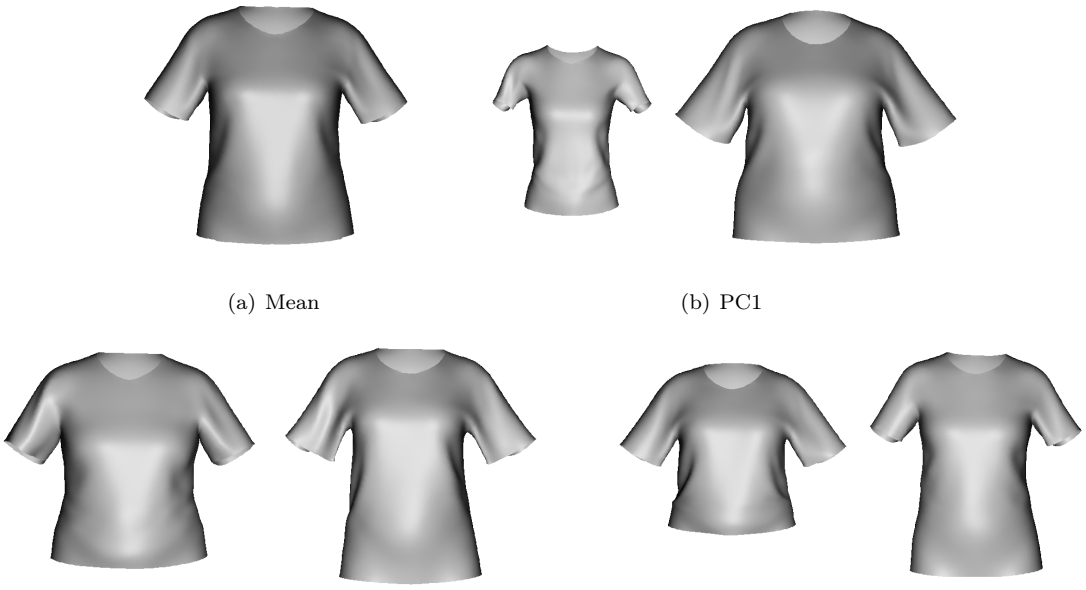
where $\Delta\vec{x}_{t,k} = \vec{x}_{t,k} - \vec{x}_{t,1}$ for $k = 2, 3$ and $\Delta\vec{x}_{t,4} = \Delta\vec{x}_{t,2} \times \Delta\vec{x}_{t,3}$. Since A_t is applied to *edge vectors*, it is translation invariant; it encodes the scale, orientation, and skew of triangle t. Following **[102]** the virtual edge, $\Delta\vec{x}_{t,4}$, makes the problem well constrained so that we can solve for A_t.

The key idea of a factored model is that it expresses the deformations, A_t, as a series of linear transformations, each corresponding to different aspects of the model. We factor A_t into pose-dependent deformation, rigid part rotation, and body shape deformation:

$$A_t = Q_t R_{p(t)} D_t. \tag{10}$$

D_t represents variations in clothing shape on different people and is triangle specific. $R_{p(t)}$ is the rigid rotation applied to clothing part p containing triangle t. Q_t is the triangle-specific non-rigid pose-dependent deformation of the garment. This pose-dependent term captures wrinkles resulting from bending and twisting. The order of the factoring matters. D_t is learned from a shape training set where all the bodies are in a template pose, thus D_t is applied first, when clothing is still in the template pose. We then rotate each of the parts and finally apply wrinkle deformations on top of the previous deformations.

DRAPE models different clothing meshes by applying different transformations D_t, $R_{p(t)}$, and Q_t to the template mesh. The deformations, however, are applied to triangles independently and do not guarantee a consistent mesh. Reconstructing the final mesh involves solving for the vertex



(a) Mean (b) PC1

(c) PC2 (d) PC3

FIGURE 5.5. **Shape model.** Deviations from the template shape: (a) template deformed by the mean deformation to create a "mean template"; (b-d) mean template deformed along the first three principal component directions ($\pm$ 3 standard deviations).

coordinates, $\vec{y}_i \in \mathcal{Y}$, that best match the deformed triangles in a least squares sense

$$\underset{\vec{y}_1,\ldots,\vec{y}_V}{\operatorname{argmin}} \sum_{t=1}^{T} \sum_{k=2,3} ||Q_t R_{p(t)} D_t \Delta\vec{x}_{t,k} - \Delta\vec{y}_{t,k}||^2. \tag{11}$$

Figure 5.4 illustrates each of the DRAPE deformations applied in order. Below we describe them in detail and, in particular, how we learn Q_t and D_t.

3.1. Deformations Due to Body Shape. The shape deformations D_t are learned from $\mathcal{X}$ and $\mathbf{S}$. Recall that the examples in $\mathbf{S}$ have the same pose as $\mathcal{X}$. We solve for the A_t's for each pair of $\mathcal{X}$ and $\mathcal{Y}^j \in \mathbf{S}$ using Equation (9). These deformations are induced by changes in clothing shape that result only from the clothing being draped over different body shapes, so $Q_t R_{p(t)}$ in Equation (10) is the identity and, for a given mesh $\mathcal{Y}^j \in \mathbf{S}$, we can write $A_t^j = D_t^j$. The clothing shape deformations D_t^j for all triangles $t = 1 \ldots T$ are concatenated into a single column vector $\vec{d}^j \in \mathbb{R}^{3\cdot 3\cdot T\times 1}$. These are collected into a matrix of deformations $S = [..., \vec{d}^j, ...]$. Principal component analysis (PCA) is used to find a low dimensional subspace, such that $\vec{d}^j$ can be approximated by $U_d\vec{\phi}^j + \vec{\mu}_d$, where U_d is a matrix of the first few principal components of the shape deformation space, and $\vec{\mu}_d$ represents the mean deformation from the template $\mathcal{X}$. Figure 5.5 illustrates the mean and first three principal components for a female T-shirt.

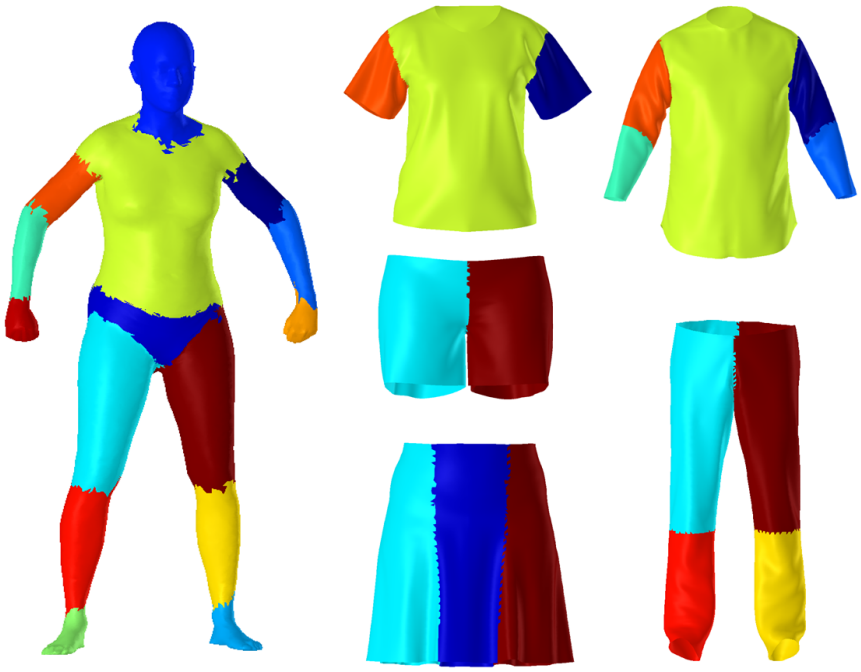


FIGURE 5.6. **Color-coded body and clothing.** The colors show the part correspondences between bodies and clothing. During training, the rigid rotation for each clothing part is the same as the rotation for the corresponding body part. This allows us to transfer a new body pose to the clothing during clothing fitting.

A new clothing shape is represented by a new set of shape coefficients $\vec{\phi}^*$. These define the shape deformation from the template, $\vec{d}^* = U_d\vec{\phi}^* + \vec{\mu}_d$. This is converted into the appropriate 3×3 deformation matrices, D_t^*, which are applied to the template as illustrated in Figure 5.4.

The key idea behind automatically dressing a new body is that we can predict the clothing shape parameters, $\vec{\phi}^*$, from a SCAPE body with shape parameters, $\vec{\beta}$ (refer to Figure 5.1). Given 60 body and clothing training pairs in $\mathbf{S}$, we learn a linear mapping, W, between these vectors using L2-regularized least squares with the weight of the regularized term being 0.2. We then predict clothing parameters for an input body shape $\vec{\beta}$ using the linear equation

$$\vec{\phi}^* = W \cdot \begin{bmatrix} \vec{\beta} \\ 1 \end{bmatrix}. \tag{12}$$

Since clothing shape deformations are a function of body shape, we write $\hat{D}_t(\vec{\beta})$ to represent the deformation matrix for a triangle t predicted from the body shape given by $\vec{\beta}$. In our work, $\vec{\beta}$ is 20 dimensional and $\vec{\phi}^*$ has only 5 dimensions because we expect the shape model to only contain low frequency deformations.

FIGURE 5.7. **Learned pose-dependent deformation model.** For each pair, the left piece of clothing shows the physically-simulated example from the pose training set, and the right piece shows the synthesized deformation patterns predicted by our model.

3.2. Deformations Due to Rigid Part Rotation. The SCAPE body model is composed of parts, which are color coded in Figure 5.6. Clothing is also naturally composed of parts during its design or can be easily segmented into parts. Each clothing part is associated with a single body part as shown by the color coding in Figure 5.6. The part correspondences for each garment are defined manually as part of the pattern creation process.

The SCAPE pose is given by the parameters $\vec{\theta}$ (refer to Figure 5.1); these are relative part rotations along a kinematic tree rooted at the pelvis. These parameters represent rigid 3×3 rotation matrices, R_p for each part p; these are applied to all the triangles in the respective body part. The DRAPE model simply applies these rotations to the corresponding clothing part as defined in Figure 5.6. For a given garment, all the part rotation parameters relevant to that garment are collected into a clothing pose vector $\vec{\theta}_c$. The part-based rotation for a clothing mesh triangle is denoted as $R_{p(t)}(\vec{\theta}_c)$.

3.3. Deformations Due to Body Pose. We use the pose training set $\mathbf{P}$ to learn a non-rigid pose-dependent clothing deformation model; this captures effects such as wrinkles. Since every $\mathcal{Y}^i \in \mathbf{P}$ corresponds to the same SCAPE body shape, all clothing deformations result from pose changes. This means D_t is the identity in Equation (10) and we write the deformations for each mesh $\mathcal{Y}^i$ and each triangle as $A_t^i = Q_t^i R_{p(t)}(\vec{\theta}_c^i)$, where Q_t^i is the residual triangle deformation after accounting for the part-based rigid rotation $R_{p(t)}(\vec{\theta}_c^i)$ given by the training body pose. Since all the clothing meshes in $\mathbf{P}$ are in correspondence, it is trivial to solve for A_t^i and consequently the non-rigid deformations, Q_t^i.

As with the shape deformation model, the clothing pose deformations, Q_t^i, for all the triangles are concatenated into a single column vector, $\vec{q}^i \in \mathbb{R}^{3 \cdot 3 \cdot T \times 1}$. We collect every example $\mathcal{Y}^i$ in $\mathbf{P}$ to form a matrix $P = [..., \vec{q}^i, ...]$. We use PCA to represent a dimensionality-reduced subspace of pose

deformation and $\vec{q}^i$ is approximated by $U_q\vec{\psi}^i + \vec{\mu}_q$. Depending on the complexity of the clothing type, $\vec{\psi}^i$ is chosen to have $30-50$ dimensions capturing 90% of the variance.

While PCA captures the space of possible deformations, to animate clothing we must relate these deformations to body pose. Cloth exhibits complex dynamical phenomenon w.r.t. the movement of underlying human body. To realistically capture how cloth moves and wrinkles, we learn a second order dynamics model for pose-dependent wrinkle deformation using the method described in **[37]**; refer to that paper for a detailed explanation. The second-order model is important to capture smooth wrinkle transitions with pose variation and fine wrinkle details.

As an example, consider the T-shirt, where $\vec{\theta}_c$ contains three relative part rotations: torso w. r. t. the pelvis, left upper arm w. r. t. the torso, and right upper arm w. r. t. the torso. Each of the part rotations are represented by a 3×1 Rodrigues vector. Therefore, $\vec{\theta}_c$ is 9 dimensional in this case.

The key idea is to write the pose deformation coefficients , $\vec{\psi}^f$, of the current frame, f, as a function of the pose, history of pose-dependent deformations, and body state changes; i.e. $\vec{\psi}^f =$

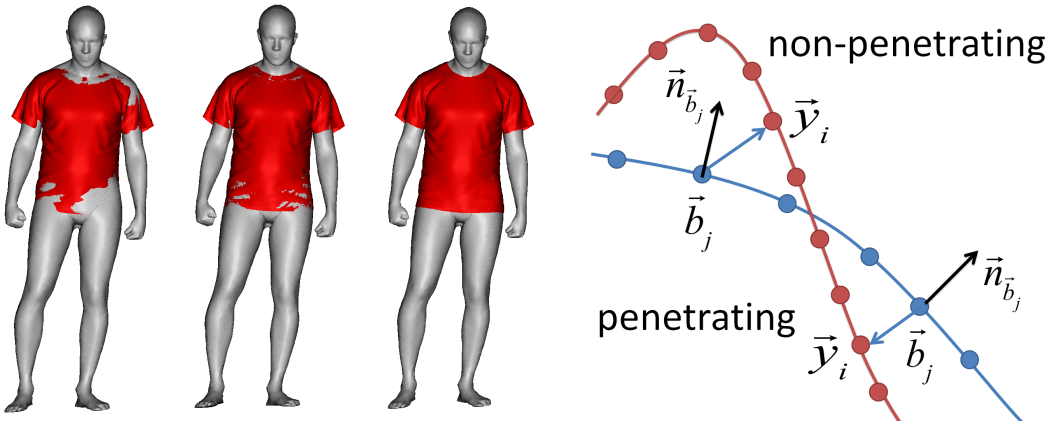
$$M_1\vec{\theta}_c^f + M_2\vec{\psi}^{f-1} + M_3\vec{\psi}^{f-2} + M_4\vec{z}^{f,f-2} + M_5\vec{z}^{f-1,f-2}, \tag{13}$$

where $M_1..M_5$ are the matrices of the dynamics coefficients to be learned, $\vec{\theta}_c^f$ is a vector of the relevant clothing part rotations at frame f, $\vec{\psi}^{f-1}$ and $\vec{\psi}^{f-2}$ are the previous two frames of pose deformation coefficients. $\vec{z}^{j,k} = \begin{bmatrix} {\Gamma^k}^{-1} \cdot (\vec{\tau}^j - \vec{\tau}^k) \\ \vec{\theta}_c^j - \vec{\theta}_c^k \end{bmatrix}$ encodes the relative body translation (normalized by the global rotation at frame k) and rotation change of frame j with respect to frame k, where Γ^k is the global (i. e., pelvis) rotation of the body at frame k, $\vec{\tau}^j$ and $\vec{\tau}^k$ are the global translations of the body. We normalize the position change between two frames so that the model generalizes better to unseen body movement directions. Note that $\vec{\theta}_c^j, \vec{\theta}_c^k$ are relative part rotations, so that they do not need to be normalized by Γ^k.

We learn a gender-specific dynamics model for each type of clothing. Given the training poses, $\mathbf{P}$, the dynamics coefficients $M_1..M_5$ are learned by solving the following least squares problem constructed from the pose training set:

$$\underset{M_1,M_2,M_3,M_4,M_5}{\operatorname{argmin}} \sum_{f=1}^{|\mathbf{P}|} ||\vec{\psi}^f - \begin{bmatrix} M_1^T \\ M_2^T \\ M_3^T \\ M_4^T \\ M_5^T \end{bmatrix}^T \begin{pmatrix} \vec{\theta}_c^f \\ \vec{\psi}^{f-1} \\ \vec{\psi}^{f-2} \\ \vec{z}^{f,f-2} \\ \vec{z}^{f-1,f_2} \end{pmatrix} ||^2. \tag{14}$$

Once $\vec{\psi}^f$ for frame f is predicted from the learned dynamics model using Equation (13), the concatenated pose-dependent deformations will be $\vec{q} = U_q\vec{\psi}^f + \vec{\mu}_q$. Again, this is converted into the



(a) initial, after 1st iter, after 3rd iter (b)

FIGURE 5.8. **Removing interpenetration.** (a) The left, middle, and right figures show the initial clothing prediction, the result after the first iteration of optimization, and the final result respectively. (b) Details of the interpenetration term. The blue dots and red dots represent body and clothing vertices respectively (see text).

appropriate 3×3 deformation matrices. Let $\hat{Q}_t(\vec{\psi}^f)$ represent the deformation matrix for a triangle t. We show in Figure 5.7 and the supplementary video that our model produces visually plausible clothing wrinkles.

3.4. Predicting New Clothing. Putting everything together, we create a new instance of a garment by solving for the vertex coordinates of $\mathcal{Y}$ such that

$$\underset{\vec{y}_1,\ldots,\vec{y}_V}{\operatorname{argmin}} \sum_{t=1}^{T} \sum_{k=2,3} ||\hat{Q}_t(\vec{\psi}^f) R_{p(t)}(\vec{\theta}_c^f) \hat{D}_t(\vec{\beta}) \Delta\vec{x}_{t,k} - \Delta\vec{y}_{t,k}||^2.$$

Computationally, the entire process described in this section involves several matrix multiplications and the solution of a sparse linear least squares problem.

4. Refining the Fit

Given a body shape and pose, DRAPE predicts a plausible clothing mesh. However, when the predicted clothing mesh is overlaid on the body (Figure 5.8(a)), there can be interpenetration between the clothing and the body. Consequently, the prediction step is followed by an efficient refinement step that warps the garment so that it lies entirely outside the body. This is achieved by minimizing a measure of cloth-body interpenetration with respect to the vertices of the garment, regularizing to make the cloth deform plausibly. Our iterative strategy alternates between computing the cloth vertices that penetrate the body, $\mathcal{P}$, and updating the clothing shape. The objective function comprises the following terms:

Cloth-body interpenetration. Given a penetrating vertex on the clothing in $\mathcal{P}$, we compute the nearest vertex on the body and its associated surface normal (Figure 5.8). We seek a clothing mesh such that all such vertices are pushed outside the body mesh. To that end, we define a penalty

$$p_{\mathcal{C}}(\mathcal{Y}) = \sum_{(i,j)\in\mathcal{C}\wedge i\in\mathcal{P}} ||\epsilon + \vec{n}_{\vec{b}_j}^T(\vec{y}_i - \vec{b}_j)||^2$$

where $\mathcal{C}$ is the set of correspondences between each clothing vertex, $\vec{y}_i$, and its closest body vertex, $\vec{b}_j$. Additionally $\vec{n}_{\vec{b}_j}$ is the normal for body vertex $\vec{b}_j$. The term $\epsilon = -0.3cm$ ensures that clothing vertices lie sufficiently outside the body. This equation has many solutions. To make the cloth deform plausibly, we regularize the solution with two additional terms and one optional term:

Smooth warping. We prefer solutions where the warping of the cloth vertices varies smoothly over the surface of the garment; i.e. we seek to minimize

$$s(\mathcal{Y}) = \sum_{i\in\mathbf{V}} ||(\vec{y}_i - \tilde{\vec{y}}_i) - \frac{1}{|\mathbf{N}_i|}\sum_{j\in\mathbf{N}_i}(\vec{y}_j - \tilde{\vec{y}}_j)||^2$$

where $\mathbf{V}$ is the set of vertices in the garment, $\vec{y}$ are vertices of the warped garment, $\tilde{\vec{y}}$ are vertices of the garment before this iteration, and $\mathbf{N}_i$ is the set of vertices adjacent to vertex i. This term prefers a deformation of a vertex that is similar to the average deformation of its neighbors.

Damping. We prefer solutions where the warped vertices keep their original locations as much as possible; i.e. we seek to minimize

$$d(\mathcal{Y}) = \sum_{i\in\mathbf{V}} ||\vec{y}_i - \tilde{\vec{y}}_i||^2.$$

Tightness (optional). There are several clothing types such as shorts, skirts, and long pants that have a waistband that needs to be in contact with the body. The "tightness" term models this and here we use it only for lower-body clothing:

$$t_{\mathcal{C}}(\mathcal{Y}) = \sum_{(i,j)\in\mathcal{C}\wedge i\in\mathcal{T}} ||\vec{y}_i - \vec{b}_j||^2$$

where $\mathcal{T}$ is a set of vertices corresponding to the clothing waist band as defined by the pattern designer. This term specifies that every waist band vertex should be close to its nearest neighbor, $\vec{b}_j$, on the body. Note that this term could be used to model tight cuffs or any clothing region that fits snugly to the body.

Our goal is to efficiently compute the mesh that minimizes

$$E(\mathcal{Y}) = p_{\mathcal{C}}(\mathcal{Y}) + \lambda_s s(\mathcal{Y}) + \lambda_d d(\mathcal{Y}) + \lambda_t t_{\mathcal{C}}(\mathcal{Y}).$$

$E(\mathcal{Y})$ is a sum of squares of linear functions of the vertices, so we can find its solution efficiently using a linear least squares solver. However, because we only consider the "currently penetrating" vertices, $\mathcal{P}$, we need to solve the least squares problem iteratively so that we do not introduce new

penetrating vertices that did not penetrate previously. At each iteration, we update $\mathcal{P}$, construct the sparse least squares problem, solve it, and update the clothing mesh. In our experiments we find that 3 iterations are sufficient to remove most collisions. The entire collision handling step is:

Given a body and a clothing mesh, compute corresponding vertices, $\mathcal{C}$, and only do this once.

$iter = 0$

repeat

$iter = iter + 1$

Determine the penetration vertex set $\mathcal{P}$.

Construct a linear system and solve:

$$\underset{\vec{y}_1,\ldots,\vec{y}_V}{\operatorname{argmin}}\{p_{\mathcal{C}}(\mathcal{Y}) + \lambda_s s(\mathcal{Y}) + \lambda_d d(\mathcal{Y}) + \lambda_t t_{\mathcal{C}}(\mathcal{Y})\}$$

until $iter = 3$

The weights decrease with iterations: $\lambda_s = 4, 2, 1$ and $\lambda_d = 0.8, 0.6, 0.4$. For lower clothing with tight waist bands $\lambda_t = 0.2$.

Details. The clothing deformation model is translation invariant, so the three dimensional global translation of the garment must be determined. Note that the global rotation is already defined by the global rotation of the pelvis. During garment creation, we define several anchor points on the garment and the roughly corresponding points on the 3D body mesh. During fitting we compute the translation by minimizing the difference between the clothing and body anchor points.

To solve for the translation we use several anchor points that are roughly corresponding points on the body and the clothing. For upper-clothing (T-shirts and long sleeves), we use 20 points on the shoulders of the body (each shoulder gets 10 points). Since the clothing meshes are aligned, it is easy to define 20 points on the shoulders of the clothing as well. For lower-clothing types (shorts, skirt, and long pants), we again use 20 points around the waist of the body and the waist of clothing as the anchor points, and translate the clothing to make them overlap.

To layer multiple pieces of clothing, we independently predict and position all pieces then refine from the inside out. For example, we predict the positions of pants and a T-shirt, refine the pants to be outside the body, and then refine the T-shirt to be outside the combined vertex set of the pants and body (here, the nearest neighbors $\mathcal{C}$ are computed between the upper-clothing and the combined vertex set). Combining the body and lower clothing is done efficiently by segmenting the body at the waist vertices and taking the union of the remaining upper body vertices and the lower clothing vertices.

Even though there is a standard method to compute the union of two meshes, the “union" step itself is computationally expensive. Here we use an extremely efficient way to compute the union

in our application. We can model the body mesh as a graph. If the inner layer is lower-clothing, we can cut the edges that are associated with the body waist vertices to make the body graph unconnected (a connected component above the waist and a connected component below the waist). Then we can run bread first search starting from any vertex on the head. The union of the body and lower-clothing will be the vertices that are reachable from the head vertex plus all the lower-clothing vertices. Since we predefine the body waist vertices and the body graph is fixed, this step is done once without additional cost at run time.

5. Experimental Results

We evaluate the performance of the DRAPE model on different clothing types, body shapes, and motion sequences.

Qualitative Evaluation. To illustrate the behavior of the model, we synthesize clothing on 2 test sequences present in the pose training set and 10 novel test sequences not present in the training set. For each test motion sequence, we synthesize multiple bodies with different random shapes using the SCAPE body model. We then dress these bodies with different combinations of upper and lower clothing types. Here, we use 20 body shape coefficients ($\vec{\beta} \in \mathbb{R}^{20\times 1}$), 5 clothing shape coefficients ($\vec{\phi} \in \mathbb{R}^{5\times 1}$), and 50 pose-dependent clothing deformation coefficients ($\vec{\psi} \in \mathbb{R}^{50\times 1}$). The choices of dimensions for $\vec{\beta}$ and $\vec{\phi}$ are discussed later. Figures 4.1 and 5.9 and the accompanying video illustrate that the method synthesizes clothing with detailed wrinkles and generalizes well to body shapes and poses not present in the training set.

We also visually compare the results of our method (with and without dynamics) to PBS. Figure 5.10 illustrates the results with two poses: 1) a male model rotating his torso, and 2) a female model in the middle of a jump. Figure 5.10 shows that the OptiTex simulations (a) contain more high frequency wrinkles than our method with dynamics (b). This is to be expected as our approach is an approximation to the physically-simulated clothing used for training. However, the strength of our method is being able to produce infinitely variable clothing sizes for different body shapes (Figure 5.11). Figure 5.10 (c) shows the results of our method without modeling dynamics; i. e., a zero order model that only uses $\vec{\theta}_c^f$ in Equation (13) to predict pose-dependent deformation. Comparing Figure 5.10 (b) and (c), we see that modeling dynamics is important for maintaining fine wrinkles, especially for fast motions.

Quantitative Evaluation. We take a male *T-shirt* as the representative clothing type for all quantitative experiments. The results are similar for other clothing types.

First, we verify the assumption that the pose-dependent non-rigid wrinkle deformations can be learned by linear regression. We expect the synthesized meshes produced by DRAPE to be

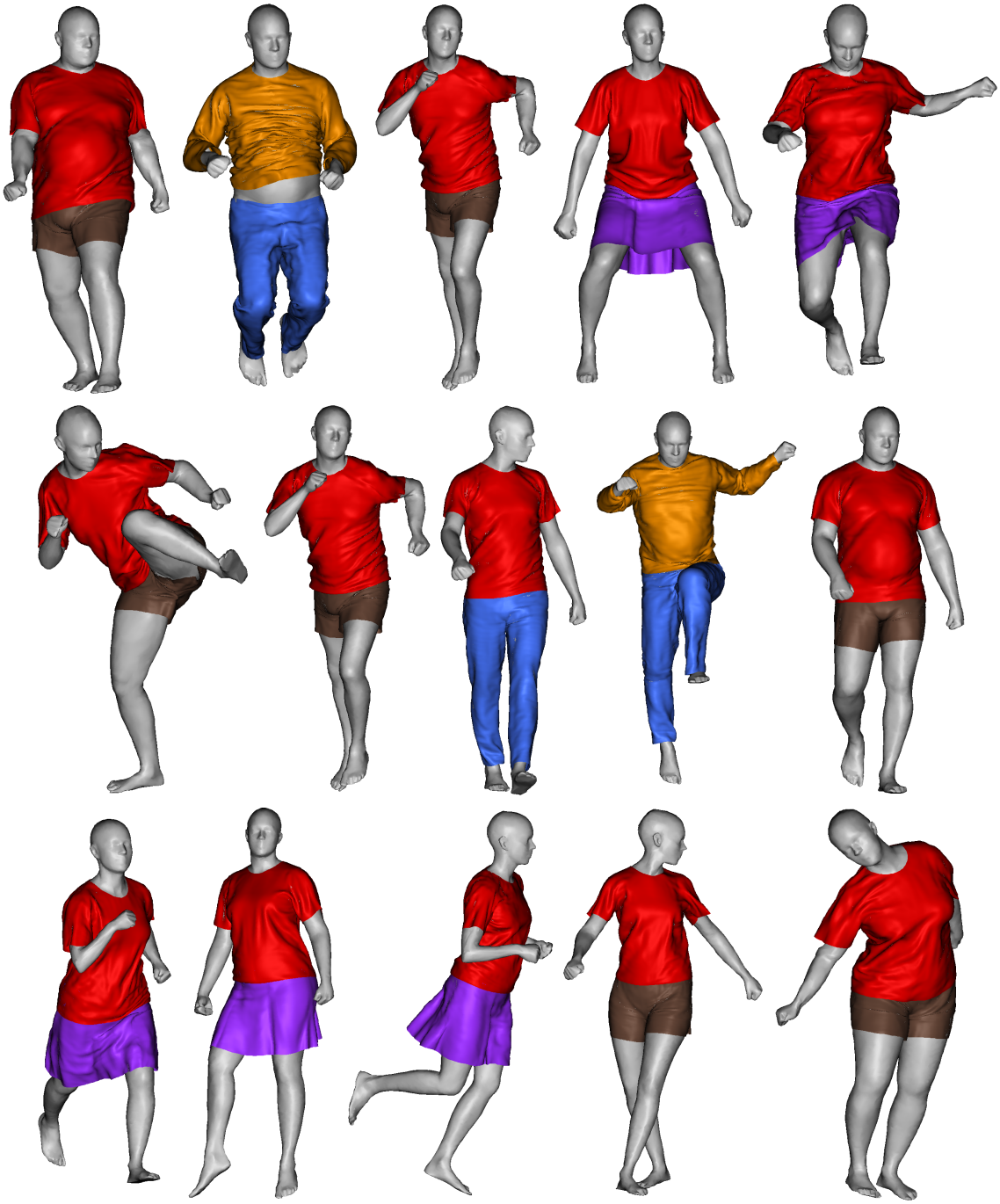
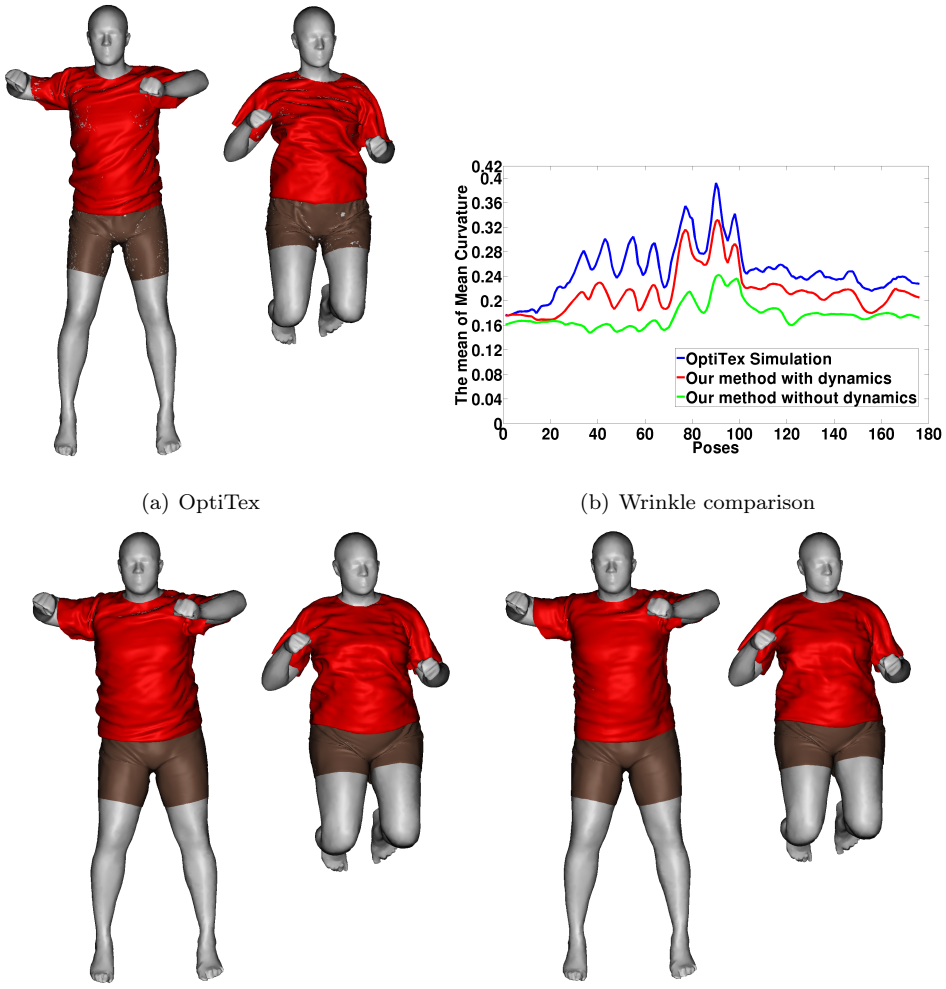


FIGURE 5.9. **More DRAPE results (test sequences not present in training set).** We randomly combine upper clothing type, lower clothing type, pose, and body shape to generate synthetically clothed people. See accompanying **video** for more results.

smoother than the ground truth PBS meshes because the linear model is an approximation of the "real" wrinkle patterns. The effect of this smoothing is shown in Figure 5.10 (d) for a representative 176 frame test sequence (including running, jumping, and stopping) simulated using OptiTex (blue)



(a) OptiTex

(b) Wrinkle comparison

(c) DRAPE with dynamics

(d) DRAPE without dynamics

FIGURE 5.10. **Wrinkles.** Comparison between OptiTex simulation on mean bodies (a) and the DRAPE model with (c) and without (d) dynamics on novel bodies. (b) Measures how "wrinkled" the garment is in terms of the mean of the mean curvature. One test sequence with a motion not appearing in the training set is shown (176 frames). The DRAPE model (with dynamics) captures the wrinkles well while the model without dynamics over smooths the clothing.

and animated by DRAPE with dynamics (red) or DRAPE without dynamics (green). We compute the mean curvature at each vertex and then take the mean of this over all vertices in the garment; this provides an objective measure of the overall amount of wrinkles in the cloth. The plot shows that 1) we lose approximately $5-15\%$ of the high frequency wrinkles due to the linear regression approximation and 2) modeling dynamics greatly helps to maintain fine wrinkles.

Second, we explore the performance of clothing shape prediction, $\vec{\phi}$, as a function of the dimensionality of the SCAPE body coefficients $\vec{\beta}$ (refer to Equation (12)). We use the average Euclidean

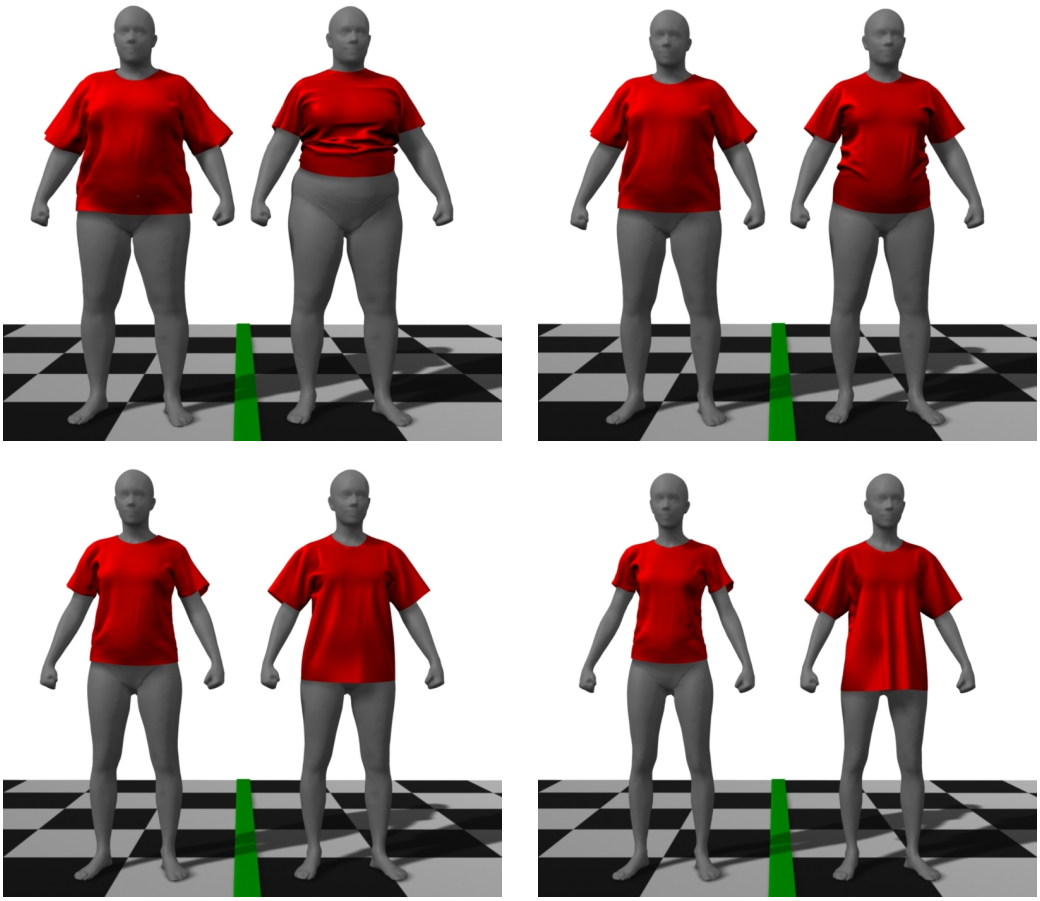
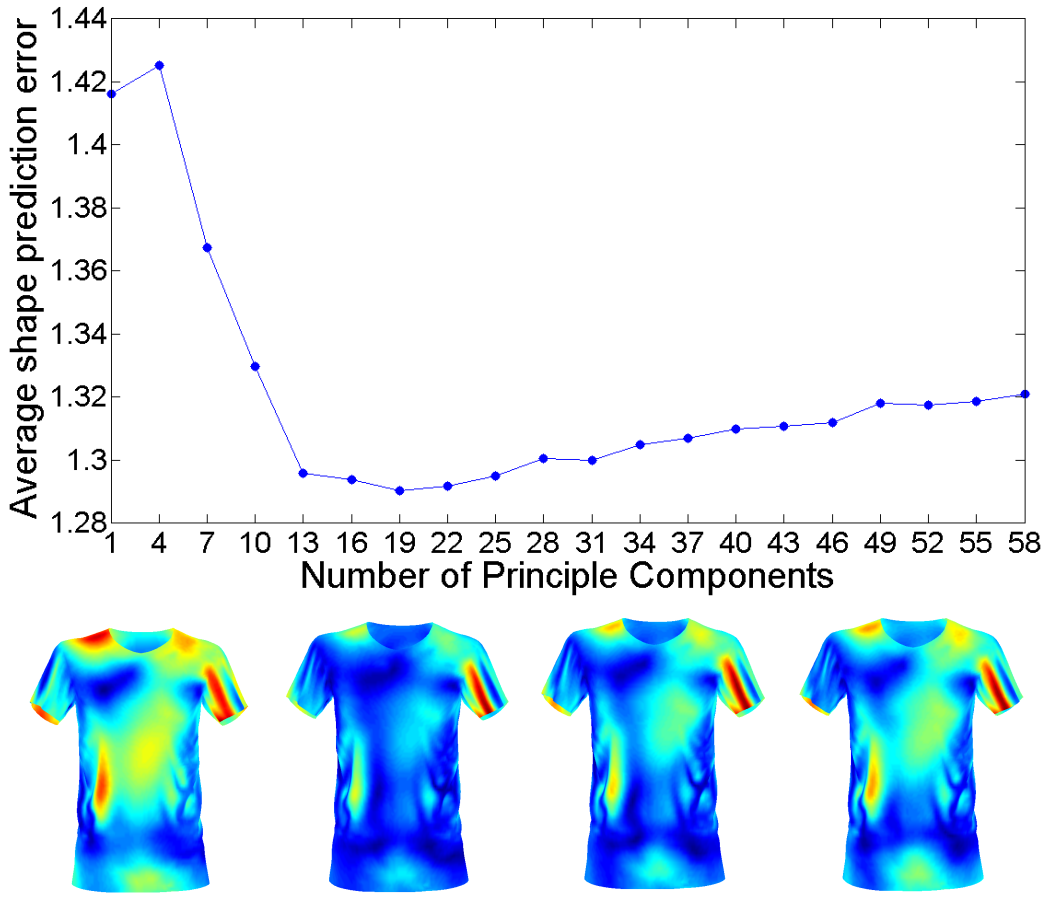


FIGURE 5.11. **Importance of fit.** We compare bodies of different shapes clothed using DRAPE (left) and OptiTex simulation (right). The OptiTex simulation uses a fixed size T-shirt, emphasizing how the quality of the simulation depends heavily on choosing the right sized garment. In contrast, DRAPE automatically predict the appropriate, infinitely-sized, clothing for every body.

vertex distance to measure shape prediction error. We use leave-one-out cross validation to predict the i^{th} clothing instance using the PCA model and the linear shape predictor learned from all the remaining 59 instances excluding i. Figure 5.12 shows average shape prediction error over the 60 examples as a function of the dimensionality of the SCAPE body shape coefficients $\vec{\beta}$. If too many principal components are used, the model tends to over-fit the wrinkles and produce higher errors. The best generalization performance is achieved with approximately 20 PCA dimensions; this might increase with more shape training data. Thus, use use 20 body shape parameters, $\vec{\beta}$, in our experiments.

Speed and Memory. The run time performance for different garments and mesh resolutions is shown in Table 6.2. Our method is implemented using Matlab (single threaded) without special



(a) 10-Dim (b) **20-Dim** (c) 30-Dim (d) 40-Dim

FIGURE 5.12. **Shape prediction accuracy versus subspace dimension.** The shape prediction error (in cm) does not decrease monotonically with the number of principal components. Over fitting occurs with more than 20 dimensions. These errors are illustrated on one of the ground truth clothing meshes, with hot/cold colors representing large/small errors.

optimization such as GPU acceleration. The OptiTex run time does not include manually choosing the appropriate clothing size and placing the cloth pieces in appropriate initial positions.

For a single frame simulation, our method is much faster ($40 - 160$X) than the commercial physical simulation. If we run cloth simulation on a motion sequence, the amortized run time per-frame for OptiTex improves a lot, but is still around 15X slower than our method. This is because OptiTex makes use of temporal coherence. Our method fits clothing to each pose individually, therefore the per-frame run time for an animation is the same as for a single pose.

All timings were obtained with a 32 bit desktop machine with a 3.2 GHz AMD Phenom$^{\text{TM}}$II processor, 4.0 GB of memory, and an NVIDIA GeForce 8600 GT video card. Our method is not

	Mesh Res		Run time (sec/frame)				
			DRAPE			OptiTex	
Garment	#Vert	#Tri	Syn	Fit	Total	Single	Animation
T	18903	37446	0.1	0.8	0.9	46	12.1
Sh	10028	19686	0.06	0.4	0.46	20	5.3
Sk	8933	17582	0.06	0.4	0.46	35	7.2
LS	17136	34026	0.1	0.7	0.8	75	17.9
LP	15980	31746	0.09	0.6	0.69	62	15.7
T+Sh	28931	57132	0.16	1.2	1.4	122	28.0
LS+LP	33116	65772	0.19	1.3	1.6	308	37.6

TABLE 5.1. **Run time performance.** Comparison of the run time performance of our method and the OptiTex package for various garments and resolutions. “T", “Sh", “Sk", “LS", “LP" stand for T-shirt, Shorts, Skirt, Long Sleeves, Long Pants respectively. “Syn" stands for clothing mesh synthesis while “Fit" represents the time for solving body-cloth interpenetration and preparation time. OptiTex-Single shows the run time for a single frame simulation and OptiTex-Animation shows the amortized run time per frame in an animation.

memory intensive. Consider a clothing mesh with 25000 triangles and a body model with 25000 triangles. Using floats for the vertices and normals, we need 450KB in total for the body and clothing to fit into memory. The shape PCA bases take 18MB (20 dimensions). The pose PCA bases take 27MB (30 dimensions). Representing the linear systems for computing the clothing deformation and clothing refinement takes approximately 400KB and 750KB respectively. This easily fits in the memory of a smart phone.

In addition to the run-time cost, there is an up-front cost of creating the training set for learning. The garment design process is completely standard and graded patterns like those used here exist for any mass produced garment already. Preparing the shape training set involves dressing each of the 60 training bodies once using the PBS system. The pose training set requires dressing the template body and simulating the motion sequences. Once the training data is created, learning the shape and pose-dependent models is very fast (minutes). Our advantage can be summarized as “simulate once, use often."

6. Discussion and Limitations

While DRAPE generates realistic clothing for different body shapes and poses, it has several limitations. First, the learned shape and pose deformation models are independent and, when they

are composed during synthesis, unnatural wrinkle patterns may be generated. Here we do not claim a physically realistic model of wrinkles, but rather demonstrate that often the simple factored model produces visually appealing results in practice. To minimize the occurrence of unnatural combinations, while retaining realism, we use a fairly smooth shape model and a higher frequency pose model (cf. [**111**]). The lower frequency shape model is naturally obtained by using fewer principal components for the clothing shape coefficients $\vec{\phi}$. The assumption is that low frequency wrinkles are related to body shape while high frequency wrinkles are largely determined by the body motion. While DRAPE handles interpenetration between the body and the clothing and between upper clothing and lower clothing, it does not model cloth self-penetration in the same clothing item.

It should be noted that the learned model is only as good as the input it is trained from. As shown here, the model is an approximation and DRAPE garments are smoother than the simulations. Here we used a particular commercial package for simulation but higher quality clothing simulations, or real cloth capture, would produce a more realistic DRAPE model. While there is some loss of fidelity compared with the training data, the advantages of the method are that the fitting is automatic, the model generalizes to different body shapes and it is computationally efficient. For many applications, particularly involving dressing many unknown body shapes, the trade off of automation for fidelity may be appropriate.

CHAPTER 6

Multi-linear Dynamic Hair Model

This chapter is built upon the original work [53]. We present a data-driven method for learning hair models that enables the creation and animation of many interactive virtual characters in real-time (for gaming, character pre-visualization and design). Our model has a number of properties that make it appealing for interactive applications: (i) it preserves the key dynamic properties of physical simulation at a fraction of the computational cost, (ii) it gives the user continuous interactive control over the hair styles (e.g., lengths) and dynamics (e.g., softness) without requiring re-styling or re-simulation, (iii) it deals with hair-body collisions explicitly using optimization in the low-dimensional reduced space, (iv) it allows modeling of external phenomena (e.g., wind). Our method builds on the recent success of reduced models for clothing and fluid simulation, but extends them in a number of significant ways. We model motion of hair in a conditional reduced sub-space, where the hair basis vectors, which encode dynamics, are linear functions of user-specified hair parameters. We formulate collision handling as an optimization in this reduced sub-space using fast iterative least squares. We demonstrate our method by building dynamic, user-controlled models of hair styles.

1. Introduction

Hair animation is a difficult task, primarily due to the large volume of hairs that need to be considered (a typical human head consists of 100,000 hair strands) and the complex hair motions and interactions. Despite this, there has been enormous success in model acquisition [**83**], simulation [**91, 36**] and rendering of hair (e.g., Rapunzel's hair in Tangled [**114**]). Such high-quality simulations, however, are expensive and require off-line processing. The approach of Daviet and colleagues [**36**] simulates 25 seconds of video in 48 hours (using 2,000 rods) and that of Salle and colleagues [**91**] simulates 1 frame in 4-38 minutes. Real-time applications, such as prototyping and games, have more stringent computational budgets, and hence often rely on less realistic models which are either entirely procedural [**23**], topologically constrained [**122**], or approximate simulation using low-resolution (e.g., guide curves or strips [**73**]) or level of detail models [**113**].

Rather than attempt to create a fast physically accurate simulation, our goal is to learn a flexible low-dimensional representation of dynamic hair motion that is compact and fast, but at the same time expressive enough to convey the dynamic behaviors seen in high-resolution simulations. Our

FIGURE 6.1. Real-time animation of 900 guide multi-linear hair model, with interactive control over the hair softness (red slider, the higher the softer) and length (blue slider, the higher the longer); bottom row shows interactive control of wind strength (arrow length) and direction (arrow orientation).

data-driven approach generates motion that models the dynamics of hair and provides a level of accuracy comparable to the input data. Our method builds on the recent success of reduced models for clothing [**37**] and fluid simulation [**106**], but extends them in a number of significant ways. It is a general method that is also applicable in other domains, for example for modeling clothing. The approach of de Aguiar and colleagues [**37**] is a special case of our model. Here we focus primarily on hair animation.

We leverage hair simulations produced by a standard simulation package (Shave and a Haircut [**2**]) to build a highly efficient multi-linear model of hair motion as a function of several user-controlled parameters (hair length, softness and wind direction). To build this model, we make two basic assumptions: (i) characters in interactive domains typically exist in finite configuration spaces, where, for example, the user has control over the transitions between a finite set of motions (e.g., as in motion graphs); or has limited dynamic control over the raw character motion (e.g., as with most interactive controllers); and (ii) there exists a continuous manifold space of hair models parameterized by geometric, dynamic, and external factors acting on the hair. The second assumption is motivated by hair grooming and simulation tools that typically provide continuous control over similar parameters but off-line.

Our method takes, as input, multiple sets of hair motions produced by a simulator under various perturbations in the parameters of interest, and learns a reduced multi-linear dynamical model approximating the behavior of hair exhibited across all sets. As a consequence, one can think of the conditional dynamic base vectors, modeling hair evolution, as being functions of real-valued factors that can be specified by the user at test time. Thus using a discrete set of simulations, we are able to build a continuous and intuitive space of dynamic hair models. Because our learning method is statistical in nature, the raw results from the multi-linear model can only *approximately* resolve body-hair contacts. This limitation can cause unwanted hair-body penetrations. To explicitly handle this problem in our model, we propose an optimization step that resolves collisions by optimizing the reduced space representation directly. This process is efficient because we only need to optimize a small set of hair parameters, instead of raw hair strand vertex positions.

Unlike prior real-time hair-simulation methods that typically rely on low-resolution models (with a handful of strips or wisps), our model is considerably more efficient and can deal with up to 4,000 guide hair strands at a small fraction of the computational cost. In contrast to most model reduction approaches [**106**], we assume no specific form for the dynamics. In contrast to data-driven methods [**37**], we do not learn a single linear dynamical model, but rather a family of models parameterized by semantic user-specifiable parameters (including external factors like the wind); we also explicitly and efficiently deal with hair-body collisions, which was a limitation of [**37**].

The ability to realistically animate hair for a large number of characters in real-time has many potential applications. Virtual worlds increasingly rely on physical simulation, and our approach offers the opportunity to incorporate realistic hair models currently lacking in most games and interactive media applications.

Contributions: We introduce a data-driven multi-linear reduced-space dynamical model for modeling hair. It is explicitly parameterized by a number of real-valued factors (e.g., hair length, hair

softness, wind direction/strength, etc.) that make it easy to adjust the groom and motion of hair interactively at test time. We formulate our model using tensor algebra and illustrate how dynamics can be incorporated within this framework. Further, we explicitly address the issue of hair-body collisions by a very efficient optimization procedure formulated directly in the reduced space and solved using a form of iterative least squares. Our formulation goes substantially beyond current reduced-space dynamical models (e.g., [**37**]).

2. Representation

We use a physically based hair simulation software (Shave and a Haircut [**2**]) to simulate a large number of hair guides, each guide being the proxy for a bundle of hair strands. Our method operates on hair guides as this is a typical representation for hair simulators. However, unlike other methods (e.g., [**103**]) that use few (up to 200) hair guides to reduce simulation complexity, we utilize up to 4,000 hair guides with our model[1]. The hair guides are simulated on the head of the virtual character, animated and skinned using a set of 35 motion capture sequences. We adopt a standard approach to interpolate between hair guides to obtain a full set of hair strands [**81**].

Hair: We use N_g guides per-frame and every guide $\mathbf{g}_k (1 \leq k \leq N_g)$ is a curve represented by $N_m = 15$ points in 3D (see Figure 6.2). We generate three different datasets with N_g being 198, 962, and 3980 respectively. Let $\mathbf{g}_{k,1}, \mathbf{g}_{k,2}, ..., \mathbf{g}_{k,N_m} \in \mathbb{R}^3$ be the points on guide k. We concatenate the x, y, z coordinates of points from the guide and obtain $\mathbf{g}_k = [\mathbf{g}_{k,1}, \mathbf{g}_{k,2}, ..., \mathbf{g}_{k,N_m}] = [g_{k,1,x}, g_{k,1,y}, g_{k,1,z}, ..., g_{k,N_m,x}, g_{k,N_m,y}, g_{k,N_m,z}] \in \mathbb{R}^{3N_m}$. We put together all the guides and use a tall vector $\mathbf{h} = [\mathbf{g}_1, \mathbf{g}_2, ..., \mathbf{g}_{N_g}]^T \in \mathbb{R}^{N_h}$ to represent one frame of hairs, where $N_h = 3N_m N_g$.

Body: Similarly we represent the body using a set of vertices of the triangular mesh (see Figure 6.2). For the purposes of our model we only need to consider the head and the shoulders (the *bust*) of the mesh with which hair can potentially come in contact. Assuming that there are N_n vertices in the bust and that each vertex is represented as $\mathbf{b}_i = [b_{i,x}, b_{i,y}, b_{i,z}] \in \mathbb{R}^3$, at a single frame the body is represented using $\mathbf{b} = [\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_{N_n}]^T \in \mathbb{R}^{N_b}$, where $N_b = 3N_n$.

2.1. Dimensionality Reduction in Canonical Space. Given the correlation among the hair guides (and body vertices) and the constrained topology of hair points, the underlying number of degrees of freedom (DOF) is much less than N_h (or N_b in the case of the body). Hence, we adopt Principal Component Analysis (PCA) to reduce the dimensionality of the two representations. We are able to capture most of the variation in the geometric hair appearance using a much lower

[1] In effect we show that by treating *all* hair strands as guide curves in our framework, we can forego the interpolation step as our model learns to incorporate "interpolation" as part of the mapping from the reduced space to the full-dimensional hair representation.

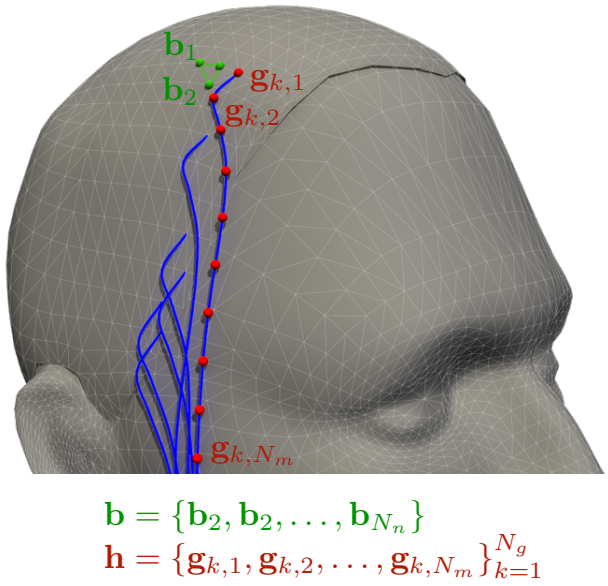


FIGURE 6.2. **Hair and body parametrization.** We represent hair using control points on a sparse set of guides and body using vertices making up the triangular mesh of bust.

dimensional space (typically 50 to 100); as the configuration of the bust is much more constrained, we use only 10 dimensions to represent it. The choice of the space in which the hair and bust are represented is also an important practical issue. Representation in the original world space hinders generalization [**37**]. Therefore, we model the motion in a canonical space of the bust.

We assume that the hair motion is only determined by the motion of the *bust*. We do not consider hair-hand interaction in this work. To normalize hairs and bust at frame t, we transform all the hair points and the bust vertices into a canonical space by: (1) subtracting the average position of the bust vertices, $\mathbf{p}_t = \frac{1}{N_n}\Sigma_{i=1}^{N_n}\mathbf{b}_{i,t} \in \mathbb{R}^3$ and (2) rotating the bust (and hairs) around the Y-axis, $r_t \in \mathbb{R}^1$ such that the head is facing towards the positive Z-axis; the negative Y-axis is the gravity direction. PCA is applied on the normalized data.

As a result, the hair at frame t, $\mathbf{h}_t \in \mathbb{R}^{N_h}$ can be written as:

$$\mathbf{h}_t = R_y(r_t)[\mathbf{Q}^h\mathbf{y}_t + \mu^h] + \mathbf{p}_t, \tag{15}$$

where $R_y(r_t)$ is a 3×3 rotation matrix around the Y-axis that rotates the hairs from a canonical space back to world space, $\mathbf{Q}^h \in \mathbb{R}^{N_h \times d_h}$ are the eigenvectors learned by the hair PCA, d_h is the dimension we choose to represent the hair, μ^h is the mean location of hairs in the canonical space, and $\mathbf{y}_t$ is a vector of hair PCA coefficients for frame t.

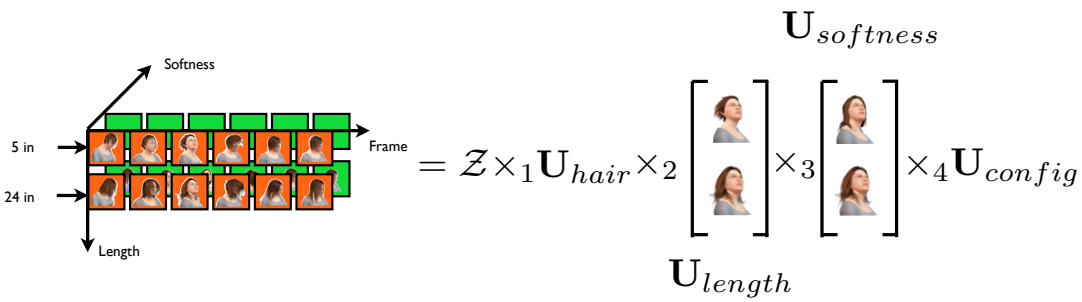


FIGURE 6.3. **Multi-linear hair model.** The representation of the hair tensor $\mathcal{D}$ (left) as a core tensor and mode matrices (right).

The bust vertices are represented in a similar way:

$$\mathbf{b}_t = R_y(r_t)[\mathbf{Q}^b \mathbf{x}_t + \mu^b] + \mathbf{p}_t, \tag{16}$$

where $\mathbf{Q}^b \in \mathbb{R}^{N_b \times d_b}$ are the eigenvectors learned by the bust PCA, d_b is the dimension we choose to represent the bust, μ^b is the mean location of bust vertices learned from training data, and $\mathbf{x}_t$ is a vector of bust PCA coefficients for frame t.

3. Multi-linear Hair Framework

The appearance of hair is a composite effect of many factors, such as length, softness, head pose and motion. We explicitly parameterize our hair model using these real-valued factors. By changing the values of any of these factors, we are able to synthesize hair with different appearance, configuration and motion. To simplify the formulation, we first introduce a generative multi-linear model for hair appearance in a given frame and then illustrate how that model can be extended to incorporate dynamics for synthesis.

Multi-linear algebra provides us with a mathematical framework to factorize hair appearance. The simulated hair exemplars, parameterized by reduced representation, are built into a data tensor $\mathcal{D}$ that is later decomposed in order to separate and represent each constituent factor. We use the Matlab Tensor Toolbox [**6**] to perform tensor operations. Hair data is built into a $N > 2$ tensor or N-way array $\mathcal{D}$, and N-mode singular value decomposition (N-SVD) orthogonalizes N spaces and decomposes the tensor into the $mode - N$ product [**108, 109**]:

$$\mathcal{D} = \mathcal{Z} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 ... \times_i \mathbf{U}_i ... \times_N \mathbf{U}_N. \tag{17}$$

The core tensor $\mathcal{Z}$ governs the interaction between the mode matrices $\mathbf{U}_1$,...,$\mathbf{U}_N$, and each mode matrix $\mathbf{U}_i$ is obtained by $mode - i$ flattening of $\mathcal{D}$ [**6**].

We introduce the formulation in terms of a simple model with two factors, but build and discuss a variety of other models of this form in the results section. We prepare the training dataset such

that we have $N_l = 2$ different hair lengths (*short* and *long*) and $N_s = 2$ different hair softnesses (*soft* and *stiff*). Note that all hair models in our dataset are in correspondence, i.e., contain the same number of hair strands, the same number of points per strand and the same scalp attachment points. Each hair length and softness combination corresponds to approximately $N_f = 12000$ frames of different head poses from 35 training sequences (animated using motion capture data). The total size of the training set is $N_l \times N_s \times N_f$ frames. We now show how we can represent hair, $\mathbf{y} \in \mathbb{R}^{d_h}$, using a multi-linear generative model.

For the simple case of the two factors of length and softness, our hair data tensor $\mathcal{D}$ is a $d_h \times N_l \times N_s \times N_f$ array, which is decomposed to:

$$\mathcal{D} = \mathcal{Z} \times_1 \mathbf{U}_{hair} \times_2 \mathbf{U}_{length} \times_3 \mathbf{U}_{softness} \times_4 \mathbf{U}_{config}. \tag{18}$$

$\mathcal{Z} \in \mathbb{R}^{d_h \times N_l \times N_s \times N_f^*}$, with $N_f^* = min(N_f, d_h \cdot N_l \cdot N_s) = d_h \cdot N_l \cdot N_s$, is the core tensor and $\mathbf{U}_{hair}$ is the hair mode matrix which will be projected out (see Figure 6.3). The $N_l \times N_l$ mode matrix $\mathbf{U}_{length}$ spans the space of hair length parameters, each row of which corresponds to a different hair length in our dataset. Similarly, the $N_s \times N_s$ mode matrix $\mathbf{U}_{softness}$ spans the space of hair softness parameters, each row of which corresponds to a different hair softness in our dataset. $\mathbf{U}_{config}$ spans the space of hair configurations that encode variations in hair appearance as the body moves. This model characterizes how hair length, softness and configuration interact and multiplicatively modulate the appearance of hair.

We can synthesize novel hair length and softness by interpolating between the rows in $\mathbf{U}_{length}$ ($\mathbf{U}_{softness}$). This interpolation corresponds to convex combination of bases, using barycentric coordinates, and can be extended to a dataset with $N_l > 2$ and/or $N_s > 2$. Let $\mathbf{v}_{length} \in \mathbb{R}^{N_l}$ ($\mathbf{v}_{softness} \in \mathbb{R}^{N_s}$) be a vector of coefficients that interpolates between the rows of $\mathbf{U}_{length}$ ($\mathbf{U}_{softness}$). Note that for our simple dataset, where $N_l = 2$, $\mathbf{v}_{length} = \mathbf{U}_{length}^T \cdot [\alpha, (1-\alpha)]^T$, where $\alpha \in (0, 1)$.

We can *generate* hair coefficients, $\mathbf{y}$, by specifying all the constituent factors (length, softness, and configuration):

$$\mathbf{y} = \mathcal{Z} \times_1 \mathbf{U}_{hair} \times_2 \mathbf{v}_{length} \times_3 \mathbf{v}_{softness} \times_4 \mathbf{v}_{config}. \tag{19}$$

Eq. 19 allows us to generate hair with different appearance using only a few matrix multiplications. In fact, to synthesize hairs with fixed style (length and softness), we can pre-compute $\mathcal{M} \in \mathbb{R}^{d_h \times (N_l \cdot N_s \cdot N_f^*)}$

$$\mathcal{M} = \mathcal{Z} \times_1 \mathbf{U}_{hair} \times_2 \mathbf{v}_{length} \times_3 \mathbf{v}_{softness}, \tag{20}$$

which corresponds to a linear space that spans the hair PCA coefficients. Only one matrix multiplication is needed to obtain $\mathbf{y} = \mathcal{M} \cdot \mathbf{v}_{config}$, where $\mathbf{v}_{config}$ is the set of coefficients that encode

hair configuration. However, for a given frame we do not have explicit knowledge of $\mathbf{v}_{config}$ *a priori*, instead in the next section we show how we can solve for $\mathbf{v}_{config}$ by conditioning the model on the bust pose and previous hair configurations; conditioning on previous hair configurations allows us to model dynamics.

4. Dynamics

The simple formulation above is unable to model dynamics and there is no intuitive way to condition the model to obtain $\mathbf{v}_{config}$ for a given frame. To address the first limitation, we build a generative model over a short (3-frame) temporal window of hair and bust configurations. This allows us to model the relationship between the (presumably unknown) hair configuration at the current frame and the (presumably known) body as well as (presumably known) hair configurations at the past frames. To address the second limitation, we show how this model can then be conditioned to predict/simulate the configuration of the hair at the current frame. More specifically, we assume a second order dynamical model on the hair (consistent with a second order ODE governing the true dynamics and empirical observations in [**37**]). We also assume a control signal $\mathbf{x}_t$, in the form of a bust at time t, that governs the motion of the hair and (later) collision detection.

Dynamic Multi-linear Hair Model: We learn a multi-linear model as in Section 3, but with augmented vectors $\mathbf{w}_t = [\mathbf{x}_t; \mathbf{y}_{t-2}; \mathbf{y}_{t-1}; \mathbf{z}_{t,t-2}; \mathbf{z}_{t-1,t-2}; \mathbf{y}_t] \in \mathbb{R}^{d_a}$, where $d_a = d_b + 3d_h + 10$, and $\mathbf{z}_{t,j} \in \mathbb{R}^5$ encodes the relative global bust translation and rotation at frame t with respect to frame j:

$$\mathbf{z}_{t,j} = \begin{bmatrix} R_y(-r_j)(\mathbf{p}_t - \mathbf{p}_j) \\ sin(r_t - r_j) \\ cos(r_t - r_j) \end{bmatrix}. \tag{21}$$

Note that we need to add $\mathbf{z}_{t,t-2}$ and $\mathbf{z}_{t-1,t-2}$ because the body and hair are normalized into a canonical space, so the incremental global motion is lost and needs to be added back (in the form of these auxiliary variables). The resulting hair tensor is $\mathcal{D} \in \mathbb{R}^{d_a \times N_l \times N_s \times N_f^*}$, where $N_f^* = d_a \cdot N_l \cdot N_s$. We also experimented with a complete generative model over the 3-frame temporal window (by adding $\mathbf{x}_{t-1}$ and $\mathbf{x}_{t-2}$ to the augmented vector $\mathbf{w}_t$) as well as with longer temporal windows but longer windows did not result in better performance, often led to over-fitting, and resulted in higher dimensional (more expensive) model inference.

Simulation as Inference: For every time instant, we need to estimate $\mathbf{y}_t$ to animate the hair. To do so, we treat $\mathbf{y}_t$ as missing data and infer it using the generative multi-linear model above operating on the augmented representation $\mathbf{w}_t$; we do so by conditioning on the part of the vector $\mathbf{w}_t^o$ that is observed at a given time instance. In the general case ($t \geq 3$), $\mathbf{w}_t^o = [\mathbf{x}_t; \mathbf{y}_{t-2}; \mathbf{y}_{t-1}; \mathbf{z}_{t,t-2}; \mathbf{z}_{t-1,t-2}] \in$

$\mathbb{R}^{d_a - d_h}$. For every time instance, we condition our model on the observed part, $\mathbf{w}_t^o$, and infer/predict the missing part, $\mathbf{y}_t \in \mathbb{R}^{d_h}$ (i.e., hair coefficients for the current frame). For a given hair style (fixed hair length and softness), our pre-computed matrix $\mathcal{M} = [\mathcal{M}^o; \mathcal{M}^{\mathbf{y}}]$ computed using Equation 20, can be decomposed into two parts, consisting of bases for reconstruction of observed variables, $\mathcal{M}^o$, and $\mathbf{y}_t$ itself.

From Section 3, we know that $\mathbf{w}_t = [\mathbf{w}_t^o; \mathbf{y}_t] = \mathcal{M} \cdot \mathbf{v}_{config,t}$. Hence, we can solve for the linearly *optimal* $\mathbf{v}_{config,t}$ for the current frame t by doing a linear sub-space solve, $\mathbf{v}_{config,t} = (\mathcal{M}^o)^\dagger \cdot \mathbf{w}_t^o$, where $\dagger$ is the pseudo inverse. We can then reconstruct $\mathbf{y}_t$ from $\mathbf{v}_{config,t}$, resulting in a very efficient and compact iterative simulation equation,

$$\mathbf{y}_t = \mathcal{M}^{\mathbf{y}} \cdot (\mathcal{M}^o)^\dagger \cdot \mathbf{w}_t^o. \tag{22}$$

Note, that if we want to change the hair style anywhere (or continuously) within a sequence, we just need to re-compute $\mathcal{M}^{\mathbf{y}} \cdot (\mathcal{M}^o)^\dagger$. For a general case we then have,

$$\mathbf{y}_t = \mathcal{M}^{\mathbf{y}} \cdot (\mathcal{M}^o)^\dagger \cdot [\mathbf{x}_t; \mathbf{y}_{t-2}; \mathbf{y}_{t-1}; \mathbf{z}_{t,t-2}; \mathbf{z}_{t-1,t-2}]. \tag{23}$$

For a given set of factors, the model can be interpreted as a second order conditional linear dynamical system, similar to the one proposed in **[37]**, i.e.,

$$\mathbf{y}_t = \mathbf{A}\mathbf{x}_t + \mathbf{B}_1\mathbf{y}_{t-2} + \mathbf{B}_2\mathbf{y}_{t-1} + \mathbf{C}_1\mathbf{z}_{t,t-2} + \mathbf{C}_2\mathbf{z}_{t-1,t-2}, \tag{24}$$

where

$$\mathcal{M}^{\mathbf{y}} \cdot (\mathcal{M}^o)^\dagger = [\mathbf{A}\ \mathbf{B}_1\ \mathbf{B}_2\ \mathbf{C}_1\ \mathbf{C}_2]. \tag{25}$$

Therefore, the model proposed by de Aguiar and colleagues is a special case of our more general formulation.

For the cases where $t = 1, 2$, the process is very similar except that $\mathbf{w}_t^o = [\mathbf{x}_t]$, and the missing part becomes $[\mathbf{y}_{t-2}; \mathbf{y}_{t-1}; \mathbf{z}_{t,t-2}; \mathbf{z}_{t-1,t-2}; \mathbf{y}_t]$.

4.1. Stability of dynamics. Similarly to **[37]**, we can measure the stability of the learned model by looking at the largest eigenvalue, λ_{max}, of linear dynamics matrix of the dynamical system, namely:

$$\begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_2 \\ \mathbf{I}_{d_h \times d_h} & \mathbf{0}_{d_h \times d_h} \end{bmatrix}. \tag{26}$$

The key difference between our approach and **[37]** is that $\mathbf{B}_1$ and $\mathbf{B}_2$ are both functions of the factors, $\mathbf{v}_{length}$ and $\mathbf{v}_{softness}$, in the multi-linear model. Hence, to prove stability we need to ensure

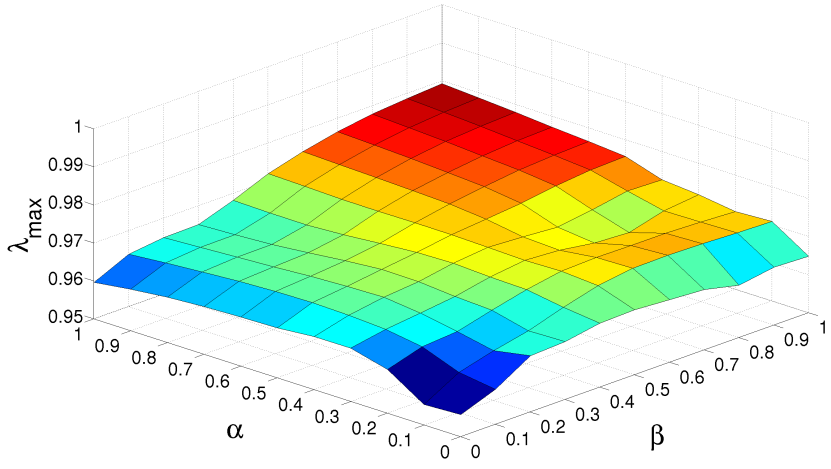


FIGURE 6.4. **Stability of dynamics.** We finely sample α and β using 0.1 interval, and interpolate the λ's in between. $\alpha = 1$ being the long hairs and $\beta = 1$ being the softest hairs. The largest λ is 0.9792 while the smallest is 0.9558.

that the largest eigenvalue λ_{max} is ≤ 1 for *any* value of factors in our model, i.e., we need to show that:

$$\lambda_{max} = \arg \max_{\alpha \in [0,1], \beta \in [0,1]} \lambda_{max}(\alpha, \beta) \leq 1. \tag{27}$$

where α and β are parameters interpolating between the bases of $\mathbf{U}_{length}$ and $\mathbf{U}_{softness}$ respectively. Taking arg max is difficult in practice, therefore we resort to an approximation obtained by evaluating arg max using a set of discrete samples (by uniformly and finely sampling α and β in the range of 0 to 1) and assuming eigenvalues are locally smooth as a function of α and β. We report the λ_{max} for several hair configurations in Table 6.1. We observe that all trained models are stable with λ_{max} consistently < 1. The plot of λ_{max} as a function of α and β is in Figure 6.4. We can see that λ_{max} is consistently < 1, leading to an overall stable model.

5. Collision Handling

The reconstructed hairs $\mathbf{h}_t$, which are a function of predicted hair coefficients $\mathbf{y}_t$, may penetrate the bust. We propose a simple and efficient method to resolve collisions. This method is based on minimizing hair-bust penetration while keeping the predicted hair coefficients unchanged as much as possible. Collision handling is done in the normalized coordinates in the reduced PCA space (for efficiency).

Our measurement of collision is based on a simple approximation of the signed distance to the body mesh. For a hair point $\mathbf{h}_i(\mathbf{y})$, we find its nearest neighbor vertex on the bust $\mathbf{b}_j$. (We drop the

temporal subscript for clarity.) Then the dot product of $\mathbf{b}_j$'s surface normal vector and the offset vector $\mathbf{h}_i(\mathbf{y}) - \mathbf{b}_j$ locally approximates the signed distance to the body mesh for $\mathbf{h}_i(\mathbf{y})$.

$$p_{\mathbf{C}}(\mathbf{y}) = \sum_{(i,j)\in\mathbf{C}} \rho\left(\mathbf{n}_{\mathbf{b}_j}^T \cdot (\mathbf{h}_i(\mathbf{y}) - \mathbf{b}_j)\right), \tag{28}$$

where $\mathbf{C}$ is a set of correspondences between hair guide point $\mathbf{h}_i$ and its closest bust vertex $\mathbf{b}_j$, $\rho(x) = \begin{cases} 0 & x \geq 0 \\ x^2/(\sigma^2 + x^2) & x < 0 \end{cases}$ is a robust error function which only penalizes negative signed distance (i.e., penetrating guide points), $\mathbf{n}_{\mathbf{b}_j}$ is the normal for bust vertex $\mathbf{b}_j$.

Method $\mathcal{A}$: A straightforward way to remove collisions is to minimize the energy function

$$E_{\mathbf{C}}(\mathbf{y}) = \pi_1 p_{\mathbf{C}}(\mathbf{y}) + \pi_2 d_{\mathbf{C}}(\mathbf{y}) + \pi_3 s_{\mathbf{C}}(\mathbf{y}) \tag{29}$$

with respect to the hair PCA coefficients $\mathbf{y}$. The first term, defined in Eq. (28), minimizes penetration. The second term,

$$d_{\mathbf{C}}(\mathbf{y}) = ||\mathbf{y} - \mathbf{y}_0||^2, \tag{30}$$

ensures that the resulting hair coefficients are close to the prediction from the model (to preserve dynamics); where $\mathbf{y}_0$ are the predicted hair PCA coefficients from the multi-linear dynamical model. The third term,

$$s_{\mathbf{C}}(\mathbf{y}) = \sum_{k\in[1,N_g]} ||\mathbf{g}_{k,1}(\mathbf{y}) - \tilde{\mathbf{g}}_{k,1}||^2 \tag{31}$$

ensures that the hair roots are at correct positions on the scalp; where $\tilde{\mathbf{g}}_{k,1}$ is the true hair root position on the scalp for the k-th guide and $\mathbf{g}_{k,1}(\mathbf{y})$ is the model position. π_1, π_2 and π_3 are the relative weights for each of the terms.

Assuming, $\mathbf{y}_t^* = \arg\min E_{\mathbf{C}}(\mathbf{y}_t)$ are the optimized hair coefficients for frame t, the final hair guides in the world space are obtained by: $\mathbf{h}_t^* = R_y(r_t)[\mathbf{Q}^h \mathbf{y}_t^* + \mu^h] + \mathbf{p}_t$. For efficiency, the nearest neighbor correspondences $\mathbf{C}$ are pre-computed, at each frame, based on the model prediction before we use gradient decent optimization on Eq. (29).

Method $\mathcal{B}$: Method $\mathcal{A}$ is fast but still involves a relatively expensive gradient optimization. We propose an approximation scheme which is around 50X faster than Method $\mathcal{A}$ while producing very similar results. The key idea is to reformulate the optimization in Method $\mathcal{A}$ in terms of a series of linear least squares (LLS) problems that can be solved extremely efficiently in closed form. $d_{\mathbf{C}}(\mathbf{y})$ and $s_{\mathbf{C}}(\mathbf{y})$ in Eq. (29) already have a convenient quadratic form and require no special treatment. The first term in Eq. (29), $p_{\mathbf{C}}(\mathbf{y})$, however, is an asymmetric error function and requires approximation.

FIGURE 6.5. **Sub-sampling factor.** Illustrated are sub-sampling factors of 1 (top), 10 (middle), and 15 (bottom) on the 3,980 hair guide dataset. There is almost no visual difference among the hairs corresponding to different sub-sampling factors.

We approximate $p_{\mathbf{C}}(\mathbf{y})$ by taking into account only the set of hair points that currently penetrate $\mathcal{P}$:

$$p_{\mathbf{C}}(\mathbf{y}) \approx \sum_{(i,j)\in\mathbf{C}\bigcap\, i\in\mathcal{P}} ||\mathbf{n}_{\mathbf{b}_j}^T \cdot (\mathbf{h}_i(\mathbf{y}) - \mathbf{b}_j)||^2 \tag{32}$$

With this approximation, every term in Eq. (29) takes quadratic form and all the variables are linear functions of unknowns $\mathbf{y}$, resulting in a standard LLS problem. Because the approximation in Eq. (32) is instantaneous and only deals with the current penetrating guide vertices, new penetrations may be introduced in the solution. To address this, we iteratively solve the optimization in Eq. (29), and for each iteration, re-compute Eq. (32), including the current set of penetrating points. However, we only compute hair-body correspondences $\mathbf{C}$ once at the beginning of the optimization and use it throughout the iterations (three to five iterations are sufficient in practice).

Sub-sampling: Method $\mathcal{B}$ allows real-time hair collision handling when the number of hair guides N_g is moderate, but is still expensive for large number of strands. In this scenario, the computational bottleneck of Method $\mathcal{B}$ becomes computing the nearest neighbor correspondences $\mathcal{C}$. To address this, we sub-sample the hair guide strands and only perform collision handling on selected guides. The intuition is that because we are modeling hair in the PCA sub-space, the hair

guides are correlated and guides within some neighborhood will generally move together. Assuming this is the case, resolving collisions for some hair guides will implicitly help resolve collisions for nearby hair guides. To achieve this goal we re-write Eq. (32) once again, resulting in the final form for $p_{\mathbf{C}}(\mathbf{y})$:

$$p_{\mathbf{C}}(\mathbf{y}) \approx \tau \sum_{(i,j)\in \mathbf{C} \bigcap i \in \mathcal{P} \bigcap i \in \mathcal{S}_\tau} ||\mathbf{n}_{\mathbf{b}_j}^T \cdot (\mathbf{h}_i(\mathbf{y}) - \mathbf{b}_j)||^2, \tag{33}$$

where τ is the sub-sample factor (e.g., $\tau = 2$ will choose every other hair guide for collision handling), and S_τ is the selected subset of hair strands corresponding to τ.

6. Experiments

We generate three datasets with different numbers of hair guides N_g: a *sparse* hair dataset with $N_g = 198$, a *main* hair dataset with $N_g = 962$, and a *dense* hair dataset with $N_g = 3980$. For the *sparse* hair dataset, we synthesize four sets of hair simulations (long soft, long stiff, short soft, and short stiff) to learn a two factor model. The *main* hair dataset is separated into two parts. The first part has the same four styles as the *sparse* dataset. The second part consists of long soft hairstyle (i) without wind and with wind directions of (ii) +z, (iii) +x, and (iv) -x. We use these four simulation datasets to learn a multilinear model with external wind strength and directions as constituent factors. The *dense* hair dataset has only one style (long soft) because it is expensive to generate training data due to the memory constraints and computing resources. We use the *dense* hair dataset to demonstrate the sub-sampling strategy for collision handling. Each dataset consists of 35 different training body motions from which we learn our multi-linear dynamic hair model and 7 test motions on which we perform our experiments; our test and training sets are disjoint. We choose a dimensionality of $d_h = 100$ for hair coefficients, which represents around 98% energy of the PCA subspace. We set $\pi_1 = 0.08$, $\pi_3 = 1.5$ in Equation 29 for all the experiments.

Model Highlights: A key property of our model is that users are able to interactively change the style of the hair, including softness and length, or apply external forces such as wind (Figure 6.1). We show side-by-side comparison of different hair lengths in Figure 6.6 (a)-(d), where (a)-(c) show the interpolated hair lengths with hair length factor being 0 (shortest), 0.5 (interpolated median), and 1 (longest), while (d) shows an extrapolation example where the length factor is 1.35. We can generate additional hair styles, not part of our training sets, by mixing the long and short hair styles we model. Figure 6.6 (e)(f) show two examples. This functionality opens a possibility for interactive hairstyle design. The comparison to the raw simulation can be found in the accompanying video and shows that our data-driven method can adequately approximate dynamic behaviors of hair (sometimes with fewer body-hair collisions as compared to the original).

FIGURE 6.6. **Creating different grooms.** (a) short, (b) interpolated medium, (c) long, (d) extrapolated long, (e) and (f) new hair grooms created by segmenting hair guides into segments and mixing long and short lengths (in (a) and (c)) for each segment.

Collision Handling: We show the performance of collision handling algorithms on the *sparse* hair dataset ($N_g = 198$), but also find similar trends in all other datasets. We define the following measurements for quantitative evaluation: (1) Penetration rate: the ratio of penetrating hair points to the total hair points. Penetration is defined by Equation 28. (2) The mean of maximal penetration amount over all frames in a sequence. The maximal penetration amount for each frame is defined as $\max |\mathbf{n}_{\mathbf{b}_j}^T \cdot (\mathbf{h}_i - \mathbf{b}_j)|$, where $\mathbf{h}_i$ is a penetrating hair point (see Equation 28). "penetration rate" is the most straightforward measurement while the "maximal penetration amount" provides an upper-bound of how deep a hair point penetrates. These two quantities are informative but not necessarily perceptual; we can arbitrarily decrease π_2 in Equation 29 to achieve better collision handling. Therefore, we use the third measurement: (3) deviation from the hair coefficients prediction: $||\tilde{\lambda}^T(\mathbf{y}^* - \mathbf{y}_0)||/||\mathbf{y}_0||$, where $\mathbf{y}_0$ is the model prediction, $\mathbf{y}^*$ are the hair coefficients after collision handling, and $\tilde{\lambda} = [\lambda_1, \lambda_2, ..., \lambda_{d_h}]^T / \sum_{i=1}^{d_h} \lambda_i$ are the normalized eigenvalues of the hair PCA subspace. We weight the hair coefficients deviation $\mathbf{y}^* - \mathbf{y}_0$ according to the importance of the principal directions.

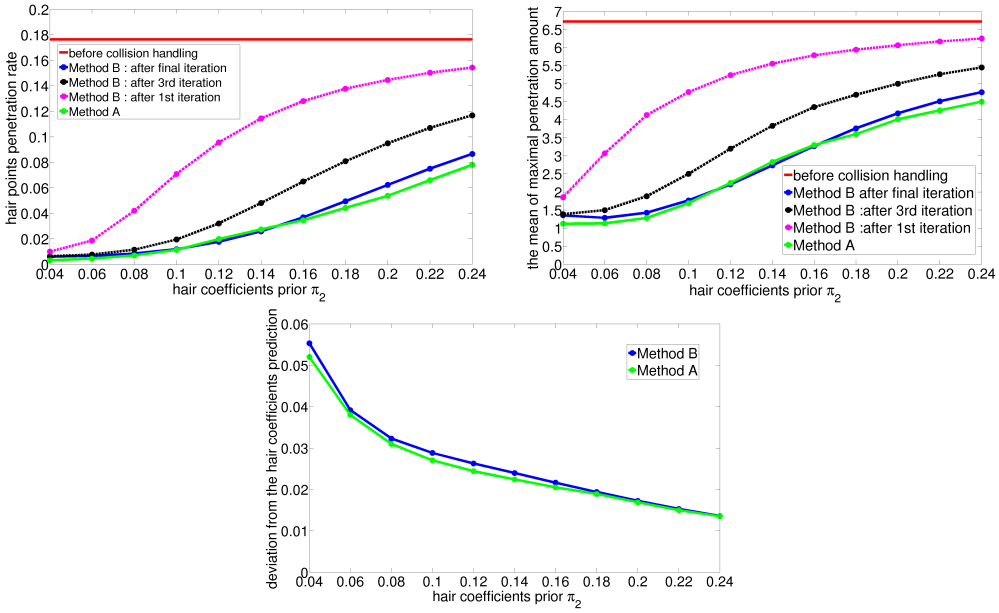


FIGURE 6.7. **Collision handling measurements versus hair coefficients prior.** We show the comparison results for method $\mathcal{A}$, two intermediate steps of method $\mathcal{B}$, final results of method $\mathcal{B}$, and before collision handling.

In Figure 6.7, we show the above-mentioned three measures versus different hair coefficients prior weight π_2. The plot is based on a representative *test* sequence with long soft hairs which includes complex motions. Note that the ground truth hair simulations from Shave and a Haircut in themselves have a non-negligent penetration rate of 1.9% and the mean of maximal penetration amount of 3.8 mm. Our collision handling algorithms (both Method $\mathcal{A}$ and $\mathcal{B}$) significantly reduce the penetration. The final penetration rates and amount of Method $\mathcal{B}$ are very similar to $\mathcal{A}$ which indicates that the iterative least squares does approximate the asymmetric error function in $\mathcal{A}$ well. As we can see from Figure 6.7 (right), the deviation from the original hair coefficients prediction varies between 1.3% and 5.5%, which visually corresponds to very natural dynamics. See the accompanying video for the visual results. Based on these three curves, our selection of π_2 is 0.13 for all the experiments.

Sub-sampling for collision handling: When we model the *dense* hair dataset (3980 hair guides and 59700 hair points), the cost of Method $\mathcal{B}$ is dominated by determining nearest neighbor correspondences $\mathbf{C}$ and penetration set $\mathbf{P}$. Therefore, we show in Figure 6.8 that we can efficiently sub-sample the hair guides to perform collision handling while still achieving almost the same results. When the sub-sample factor τ increases (see Equation 33), the curve of the penetration rate is almost flat, which indicates that we can sub-sample the hairs significantly without sacrificing the collision handling performance, because the dense hair guides are highly correlated. Figure 6.5

Sparse	L-soft	L-stiff	S-soft	S-stiff	λ_{max}
Training	3.39	2.09	1.73	1.23	0.9792
Testing	3.61	1.93	1.91	1.14	
Main	L-soft	L-stiff	S-soft	S-stiff	
Training	2.85	1.66	1.20	0.84	0.9646
Testing	2.93	1.57	1.22	0.78	
Main	L-soft	L-wind+z	L-wind+x	L-wind-x	
Training	2.97	4.23	4.50	4.32	0.9663
Testing	3.12	4.27	4.47	4.21	
Dense	L-soft				
Training	2.76				0.9621
Testing	2.71				

TABLE 6.1. **Average vertex error and stability.** Average vertex error for all the datasets (using Euclidean distance measured in *cm*) and the stability measurement λ_{max} computed over 35 training and 7 testing sequences. "L" and "S" represent long and short hair styles respectively.

visually shows the collision handling results using a sub-sample factor of 1 and 15. There is almost no visual difference between two cases. There is, however, a large computational gain; we achieve a 12X speed up by using a sub-sample factor of 15. We plot the time cost of the collision handling procedure (from the predicted hair coefficients $\mathbf{y}_0$ to final hair coefficients $\mathbf{y}^*$) versus the sub-sample factor τ. As τ increases, the time cost drops significantly. The sub-sampling strategy makes it possible for our method to potentially deal with even more hair strands in real time.

Quantitative Evaluation: We show the hair vertex location differences between the Shave and a Haircut simulation and our end results in Table 6.1. Stiff hairs have much lower errors compared to soft hairs, because the motion of the stiff hairs are more constrained. The long soft hairs with wind have high errors, because wind leads to less predictable hair behavior. The fact that training and testing sequences get similar errors (in some cases the errors of the testing sequences are lower) indicates the generalization power of our method. The stability measurement λ_{max} for each dataset is also shown in the table. These values are all below 1, which shows that our models are stable.

Performance: The speed of our method and the Shave and a Haircut (Shave) simulation package are shown in Table 6.2. Maya and Shave were run on an Intel Core 2 Extreme X9650, 3.67 GHz processors with 4 GB of RAM. Our model was implemented on a GPU and run on a comparable AMD Phenom(tm) 2 X4 965 processor, 3.4GHz with 8 GB of RAM. The amount of RAM is irrelevant as the model is very compact and easily fits in memory. Note that most of our model (everything

Hair		Runtime (FPS) Our Method				Shave
#Strands	#Vert	Syn	Col	Recon	Total	
198	2970	1429	74.6	5000	70	7.7
962	14430	1429	52.9	2500	50	5.5
3980	59700	1429	49	909	45	1.8

TABLE 6.2. **Runtime performance.** Speed comparison (frames per second) between our method and Shave software package. We divide our method into "Syn" (computing the initial hair coefficients from the multi-linear dynamic model), "Col" (remove hair-body penetration), and "Recon" (reconstruction of the hair vertices from hair coefficients). We choose the sub-sample factor $\tau = 1, 5, 15$ for hair strands $= 198, 962, 3980$ respectively.

except collision detection) is perfectly parallelizable; collision detection requires LLS implementation which could also be made efficient on parallel architectures. Our method easily achieves *real-time* (45-70 FPS) and it is $9 - 25X$ faster than Shave. We report the results based on long soft hairs, which tend to have the most collisions. As the number of hair guides increases, our method becomes comparatively much more efficient, due to the benefit of a low-dimensional model that can capture correlations. These results show the potential of our method to deal with large number of hair guides; in doing so it also alleviates the need for additional hair interpolation for rendering. Further speedups are possible using level of detail models and/or culling for parts of the groom that are not visible.

The memory requirement for the our model is quite low. A hair tensor model which has 2 factors with hair dimension of 100 takes approximately 15mb. Adding another factor will double the size, however, too many factors are unlikely. The key advantage of modeling hair in the reduced space is that dimensionality of the hair coefficients is a function of the underlying complexity of the hair motion, and is highly sub-linear with respect to the number of hair strands and the number of vertices per strand. This property also makes the model scalable with the complexity of the hair.

Collision handling for cloth: Our collision handling approach is not limited to hair; it can be used, for example, to resolve body-cloth collisions for clothing. We test our approach on the results of **[37]** (see supplemental video). In **[37]**, the collisions were not resolved explicitly. In contrast, we resolve all the interpenetrations without resorting to a custom rendering pipeline.

We apply our collision handling approach on the data provided to us by the authors of **[37]** (see Figure 6.9 (bottom)) and effectively and efficiently resolve all the interpenetrations without resorting to a custom rendering pipeline.

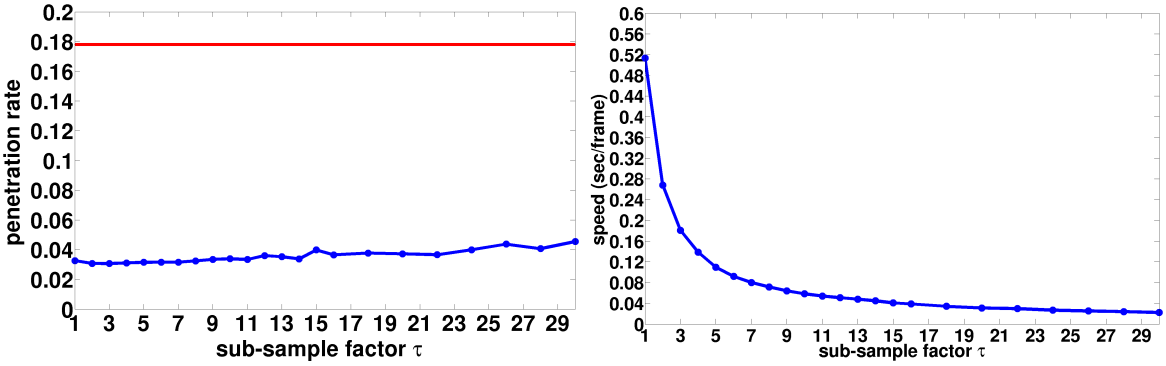


FIGURE 6.8. **Dense hair sub-sampling.** Left: penetration rates comparisons between "before collision handling" (red) and various sub-sample factors on a representative sequence (blue). The penetration rates are computed on the full hairs. Right: Time cost of collision handling procedure versus various sub-sample factors.

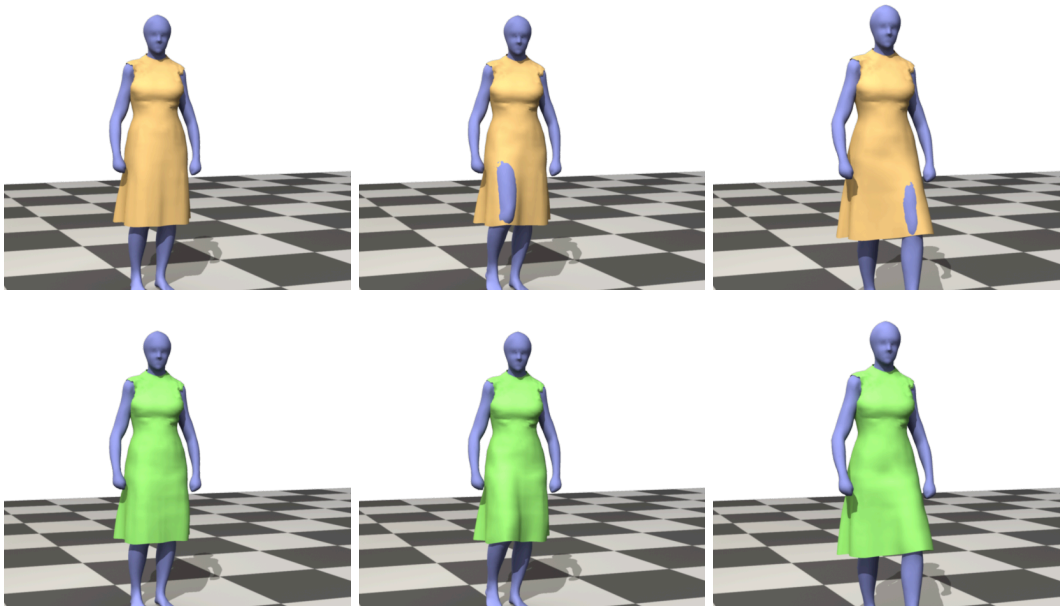


FIGURE 6.9. **Collision handling for cloth.** Our collision detection method applied to the results of "*Stable Spaces for Real-time Clothing*". We thank the authors for providing us with data from the original paper for this experiment.

7. Discussion and Limitation

We present a method for data-driven animation of hair. The multi-linear nature of our model allows us to control the appearance and motion of hair in real-time. Our method efficiently deals with collisions by formulating collision handling as an iterative least squares optimization in the reduced space. While we illustrate our model on hair, the formulation is general and would work for other physical simulations such as clothing and fur.

One of the issues we encountered when building our models is that the results from off-the-shelf hair simulation packages are not free of collision. In particular, they often contain visible penetrations of the hair into the shoulders for longer hair or the scalp for highly dynamic motions. The software package we use has a fixed number of control vertices per hair guide. This constraint works well for short hair where the control vertices are tightly spaced, but results in collisions for longer hairstyles where interactions with the neck, shoulders, and back are more complex. This limitation prevented us from modeling very long hair as we needed training data where the penetrations did not fundamentally change the motion. Also, presumably the model would produce hair motion with fewer collisions if the training data was collision free.

Because our model lives in a low-dimensional subspace we are not able to resolve hair-to-hair collisions *explicitly*, as the motion of individual hair strands is not independent. That said, our model is able to resolve hair-to-hair collisions *implicitly* by learning how hair strands should interact based on the input simulations. However, because there is no commercially available simulator that is able to produce such effects, we cannot show any results with hair-to-hair collisions.

We construct two types of reduced models with a number of parameters: short vs. long, stiff vs. soft, and with/without wind as an external force. Naturally there are many other parameters that we might want to include in the hair model: curly vs. straight, dry vs. wet, greasy vs. clean as well as other external forces such as tugs, barrettes, and headbands. In our experiments, the existing model was robust to the modeled parameter space, with no combination of parameters within the modeled ranges (or slightly outside) producing unnatural results. The tensor algebra is general enough to extend to more factors. However, with more factors being used, the size of the training examples grows exponentially. In theory, if we want to model x factors simultaneously, we need 2^x sets of training data to represent all the combinations of all factors. It would be challenging to keep all the data in memory for training (requiring out of core SVD algorithm). In practice, the animators can choose what factors they want to model to avoid the explosion problem.

In conclusion, the approach to creating a reduced space model presented here appears quite powerful. We expect that implementing this approach for other physical simulations such as clothing and fur would be easy and would result in approximate dynamic models that could be computed and rendered many times faster than real time. This functionality would be useful in visualization of character motion during the animation process as well as allowing rich secondary motion to be added to real-time applications such as games.

CHAPTER 7

Conclusion

We have presented a complete pipeline from estimating 3D human body shapes to animating the bodies with clothing and hair. This chapter summarizes our contribution, limitations, and future research extensions.

1. Contribution

Detailed 3D body shape estimation from images is a difficult problem. What is even more difficult is to estimate detailed 3D body shape from a single image. Even though we have shown that the reconstruction of a generic 3D object can be done using multiple silhouettes or photometric stereo images, those methods do not apply when image evidence is scarce. The philosophy behind our work is that as image cues become fewer, more human specific *prior* information needs to be used. We use the SCAPE body model, but other realistic parametric models will also work. We show that *smooth shading* helps to refine the body shape estimation and our key contribution is the formulation of *parametric body shape from shading*. At the same time, we exhaustively explore the usage of other image cues such as silhouettes and edges. We do not claim that a single image is sufficient to estimate detailed body shape, instead we propose a solution to solve a problem which has been considered almost impossible. In order to make the smooth shading formulation work, we assume that the surface albedos and specularities are piece-wise constant. The assumption limits our method to be used in the cases where the person is close to naked or is wearing uniform colored clothing. With the advent of depth camera technologies, it is becoming increasingly popular to use a depth sensor in addition to a regular camera sensor. In the near future, we will see more applications with depth cameras.

The 2D eigen-clothing model is an attempt to deal with the *clothing* issue in body pose and shape estimation. Most work ignores the effect of clothing, which is fine if the model itself is coarse (e.g. simple rectangular body parts). However, there are many applications (e.g. surveillance) that require us to obtain a relatively detailed body shape estimation. We take the simple idea of modeling the clothing deformation as an additional layer of the body deformation and explore the statistical properties of the clothing deformation. The clothing model greatly improves the 2D body shape estimation from a single image.

After we estimate the 3D body shapes, we would like to automatically dress these bodies with perfect fitting clothing. This is very important for virtual fashion and online clothing retail. DRAPE is a complete solution for dressing people in a variety of shapes, poses, and clothing types. DRAPE is learned from standard 2D clothing designs simulated on 3D avatars with varying shape and pose. Once learned, DRAPE adapts to different body shapes and poses without the redesign of clothing patterns; this effectively creates infinitely-sized clothing. A key contribution is that the method is automatic. In particular, animators do not need to place the cloth pieces in appropriate positions to dress an avatar. Traditionally, clothing simulation is widely used in animated movies and gaming so that realism and speed are the main focuses of prior literature. *Automation* has not drawn attention from researchers until recently. We envision internet-based virtual try-on where hundreds and millions of users should be dressed with appropriate clothing sizes. The process clearly needs to be automatic. The DRAPE model simultaneously achieves automation, realism, and speed, which is unique among clothing simulation methods. By factoring the clothing shape and pose-dependent deformations, we can train the DRAPE model with a reasonably small training set and later the clothing fitting can be done fully automatically. We have no intention to replace physics based simulation (PBS), because PBS produces the most realistic results. PBS is appropriate for animated movies without a doubt. We hope DRAPE model can open a whole new perspective and stimulate more research along the line of *automation*.

Finally, hair simulation is crucial for animated movies and gaming. The most important factor for hair simulation is speed. It is hard to develop a real-time hair simulator because the number of hairs on the real person is huge. Even if we use hair guides to represent a bunch of hairs, it is still computationally expensive. We believe that the movements of hairs are highly correlated, therefore we can represent hairs in a much lower dimensional space without sacrificing the simulated dynamics and realism of hairs. Our system could not be made real-time without modeling hairs in the reduced space. We believe that this is an interesting direction and would like to see more followup works in the near future.

2. Future Work

There has been a lot of work on estimating human pose from images but human shape estimation is less explored. Detailed human shape estimation is limited by the realism of body model and the resolution of image observations. We have shown that we can achieve reliable estimation of human shapes by leveraging a data-driven detailed human body model. Because of the non-rigid nature of human body, the appearance of the body is affected by both articulated pose and shape. This inevitably leads to a high dimensional model and no known realistic body model can avoid

this. Future work should focus on speeding up the optimization process of such high dimensional models. For an optimization in the high dimensional space, the quality of initialization is important. Therefore, a lot of work uses manual assistance to initialize the body model (including ours), which limits the usage of body shape estimation. We should pay attention to getting a good initialized pose automatically.

The downside of using data-driven models for clothing animation is that they are not as realistic as PBS. In future work we will generate pose training data for different body shapes and learn a multi-linear or non-linear model that couples pose and shape. This will likely require significantly more training data and will trade computational efficiency for wrinkle detail. The pose training set may include bodies with different shapes so that we can get diversified wrinkle patterns. We may use multi-linear or non-linear methods to learn the pose deformation to preserve more high frequency wrinkles. Future work should also explore a wider range of garment and fabric types. We will also learn models of "tucked in" clothing and more complex garments with pleats, cuffs, collars, and closures (buttons and zippers). Finally, clothing fit is not just about body shape but also involves individual preference. By training the model with different fit preferences (e.g. loose and tight) we should be able to add a "comfort" axis to the PCA shape basis that can be independently controlled.

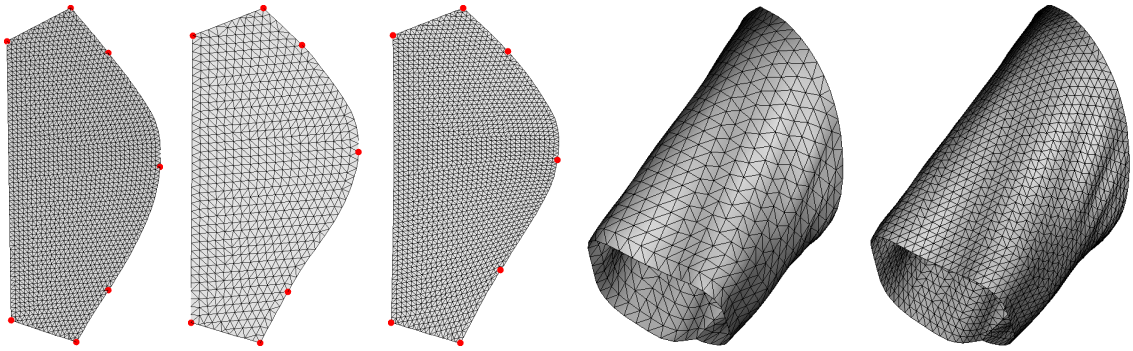
One challenge of hair simulation is how to use it in gaming. There are many things going on in gaming including AI, physical contact, rendering, etc. Future research should focus on employing better techniques to parallelize the hair simulation. Low dimensional models of clothing and hair are very promising because they significantly reduce the computation. We envision more work in this direction in the next few years.

APPENDIX A

Aligning Training Clothing Meshes

OptiTex takes the 2D patterns for each clothing part and triangulates them to produce flat (or cylindrical) meshes of different sizes. These pieces deform as they are stitched and draped during simulation. When we simulate people of different sizes the 3D meshes generated may have different numbers of vertices and will not be in full alignment. We align each piece (e.g., sleeve, front piece, back piece) individually so that when they are stitched, we get an aligned clothing mesh (e.g. T-shirt).

Figure A.1(a) shows a template sleeve piece laid flat and (b) shows a smaller size target flat piece. (Here, we have zoomed in on the target flat piece for better visualization. In reality, (b) has the same triangle resolution as (a), but is smaller in actual size.) The red dots indicate the grading points which are shared between the pieces (i.e. these are in correspondence). As we can see from the figure, the piece in (b) has many fewer triangles than (a).



(a) m^t (b) m^r (c) m (d) M^r (e) M

FIGURE A.1. **Cloth piece alignment.** The goal is to warp the template flat piece m^t to a target aligned flat piece m, such that m keeps the outline of target flat piece m^r (shape preserving) and shares the triangulation of m^t (perfect alignment). Once we have aligned these flat pieces, we can propagate the alignment to draped meshes. M^r is the target draped piece from cloth simulation and M is the final aligned version of M^r which is what we want. The red dots highlight some of the grading points.

We describe a method that aligns each individual flat piece in the plane and propagates the alignment to draped meshes. Let the template flat piece (e.g. the sleeve) be denoted by $m^t \in \mathbb{R}^{3N_t}$ which has N_t vertices $\in \mathbb{R}^3$. A target flat piece is represented by $m^r \in \mathbb{R}^{3N_r}$, where $N_r \neq N_t$. The draped version of m^r is M^r. First, we want to compute a target aligned flat piece m, such that m preserves the shape outline of m^r, but shares the same triangulation as m^t. As a result, m is aligned with m^t because they share the same triangulation. m also represents the shape of m^r because they have the same shape outline.

Then, we propagate the alignment to draped mesh by using m as a bridge to compute M. Here, M is an unknown draped version of m. Since m is aligned with the template flat piece, we know that M will also be aligned. Therefore, if we stitch different pieces, M_i, we will get an aligned draped clothing mesh.

Let the vertices of m^t, m^r, and m be $\mathbf{c}^t$, $\mathbf{c}^r$, and $\mathbf{c}$, where $|\mathbf{c}^t| = |\mathbf{c}|$. Since the same grading rule is applied to m^t and m^r, we denote the set of grading points as $\mathbf{u}$. These grading points are all on the boundary, representing corners and curves. Each vertex $c_i^t \in \mathbb{R}^3$ in m^t (including boundary vertices) can be represented by a linear combination of its neighbors $c_i^t = \sum_{j \in \mathcal{N}(c_i^t)} \alpha_{ij}^t \cdot c_j^t$, where $\sum_{j \in \mathcal{N}(c_i^t)} \alpha_{ij}^t = 1, 0 \leq \alpha_{ij}^t \leq 1$. Remember that m^t is known and actually sits in a 2D plane, so the α_{ij}^t's are easy to compute. We assume that these linear combination coefficients encode the interior structure of m^t. We want to make m and m^t share the same interior structure (i.e. the same neighborhood information and α values), and to make m and m^r share the same grade points and outlines. We construct a least squares problem and solve for $\mathbf{c}$ using the α coefficients computed from m^t:

$$\underset{\mathbf{c}}{\operatorname{argmin}} \left\{ \sum_{i \notin \mathbf{u}} ||c_i - \sum_{j \in \mathcal{N}(c_i^t)} \alpha_{ij}^t c_j||^2 + \sum_{i \in \mathbf{u}} ||c_i - c_i^r||^2 \right\} \tag{34}$$

The first term ensures the identical internal structures of m and m^t while the second term makes the grading point positions the same as m^r.

Once we solve m, the next question is how to compute M given m^r,M^r, and m. We know that m and m^r almost have the same boundary and cover the same region if we consider them in 2D. We overlay m and m^r, and rewrite each vertex c_i of m in terms of some vertices in m^r. This can be done by finding the triangle in m^r where c_i falls into, and write c_i as a linear combination of the vertices of this triangle. Then we propagate the same linear relationship to draped meshes, and write each vertex c_i of M in terms of the vertices of that triangle in M^r. This propagation is valid because m^r and M^r are essentially the same mesh before and after cloth simulation. This gives us a way to *warp* M^r to an aligned mesh M. Recall that M is the draped version of a particular piece.

When all these pieces are in alignment, the stitched mesh will be in alignment. See Figure A.1 (e) for an example of the final mesh.

Bibliography

[1] B. Allen, B. Curless, and Z. Popović. The space of human body shapes: reconstruction and parameterization from range scans. *ACM Trans. Graph.*, 22(3):587–594, July 2003.

[2] J. Alter. Shave and a haircut. *http://www.joealter.com/*, 2006.

[3] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1014–1021, Jun 2009.

[4] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. *ACM Trans. Graph.*, 24(3):408–416, Jul 2005.

[5] A. Baak, M. Müller, G. Bharaj, H.-P. Seidel, and C. Theobalt. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *IEEE International Conference on Computer Vision, ICCV*, pages 1092–1099, Nov 2011.

[6] B. W. Bader and T. G. Kolda. Algorithm 862: Matlab tensor classes for fast algorithm prototyping. *ACM Trans. Math. Softw.*, 32(4):635–653, Dec. 2006.

[7] A. Balan and M. J. Black. The naked truth: Estimating body shape under clothing. In *European Conference on Computer Vision, ECCV*, volume LNCS 5303, pages 15–29, 2008.

[8] A. O. Balan. Detailed human shape and pose from images. *Doctoral Dissertation, Brown University*, May 2010.

[9] A. O. Balan, M. J. Black, H. W. Haussecker, and L. Sigal. Shining a light on human pose: On shadows, shading and the estimation of pose and shape. In *IEEE International Conference on Computer Vision, ICCV*, pages 1–8, Nov 2007.

[10] A. O. Balan, L. Sigal, M. J. Black, J. E. Davis, and H. W. Haussecker. Detailed human shape and pose from images. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1–8, Jun 2007.

[11] Y. Bando, B.-Y. Chen, and T. Nishita. Animating hair with loosely connected particles. *Computer Graphics Forum*, 22(3):411–418, 2003.

[12] D. Baraff and A. Witkin. Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 43–54, New York, NY, USA, 1998. ACM.

[13] A. Baumberg and D. Hogg. Learning flexible models from image sequences. In *European Conference on Computer Vision, ECCV*, pages 299–308, 1994.

[14] A. Beazley and T. Bond. *Computer-Aided Pattern Design and Product Development*. Wiley-Blackwell, 2003.

[15] F. Bertails, S. Hadap, M.-P. Cani, M. C. Lin, K. Ward, S. R. Marschner, T.-Y. Kim, and Z. Kacic-Alesic. Realistic hair simulation: Animation and rendering. In *ACM SIGGRAPH Class Notes, August, 2008*, pages 89:1–89:154, Los Angeles, Etats-Unis, 2008. ACM.

[16] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '99, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[17] J. F. Blinn. Models of light reflection for computer synthesized pictures. In *Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '77, pages 192–198, New York, NY, USA, 1977. ACM.

[18] J. Bordes, M. Maher, and M. Sechrest. Nvidia apex: High definition physics with clothing and vegetation (presentation). In *Game Developers Conference*, 2009.

[19] L. D. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *IEEE International Conference on Computer Vision, ICCV*, pages 1365–1372, Oct 2009.

[20] D. Bradley, T. Popa, A. Sheffer, W. Heidrich, and T. Boubekeur. Markerless garment capture. *ACM Trans. Graph.*, 27(3):99:1–99:9, Aug. 2008.

[21] R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph.*, 21(3):594–603, July 2002.

[22] R. Bridson, S. Marino, and R. Fedkiw. Simulation of clothing with folds and wrinkles. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '03, pages 28–36, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[23] V. Cassol, F. Marson, and S. Musse. Procedural hair generation. In *Brazilian Symposium on Games and Digital Entertainment (SBGAMES) VIII*, pages 185–190, Oct 2009.

[24] J. T. Chang, J. Jin, and Y. Yu. A practical model for hair mutual interactions. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '02, pages 73–80, New York, NY, USA, 2002. ACM.

[25] Y. Chen and R. Cipolla. Single and sparse view 3d reconstruction by learning shape priors. *Computer Vision and Image Understanding*, 115(5):586–602, May 2011.

[26] Y. Chen, T.-K. Kim, and R. Cipolla. Inferring 3d shapes and deformations from single views. In *European Conference on Computer Vision, ECCV*, pages 300–313, 2010.

[27] B. Choe, M. G. Choi, and H.-S. Ko. Simulating complex hair with robust collision handling. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '05, pages 153–160, New York, NY, USA, 2005. ACM.

[28] K.-J. Choi and H.-S. Ko. Stable but responsive cloth. *ACM Trans. Graph.*, 21(3):604–611, July 2002.

[29] K.-J. Choi and H.-S. Ko. Research problems in clothing simulation. *Comput. Aided Des.*, 37(6):585–592, May 2005.

[30] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, Jan 1995.

[31] F. Cordier and N. Magnenat-Thalmann. Real-time animation of dressed virtual humans. *Computer Graphics Forum*, 21(3):327–335, May 2002.

[32] F. Cordier and N. Magnenat-Thalmann. A data-driven approach for real-time clothes simulation. In *Proceedings of the Computer Graphics and Applications, 12th Pacific Conference*, PG '04, pages 257–266, Washington, DC, USA, 2004. IEEE Computer Society.

[33] F. Cordier, H. Seo, and N. Magnenat-Thalmann. Made-to-measure technologies for an online clothing store. *IEEE Comput. Graph. Appl.*, 23(1):38–48, Jan. 2003.

[34] L. D. Cutler, R. Gershbein, X. C. Wang, C. Curtis, E. Maigret, L. Prasso, and P. Farson. An art-directed wrinkle system for cg character clothing. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '05, pages 117–125, New York, NY, USA, 2005. ACM.

[35] D. H. Daniel Grest and R. Koch. Human model fitting from monocular posture images. In *Proc. of VMV*, pages 665–1344. MIT Press, 2005.

[36] G. Daviet, F. Bertails-Descoubes, and L. Boissieux. A hybrid iterative solver for robustly capturing coulomb friction in hair dynamics. *ACM Trans. Graph.*, 30(6):139:1–139:12, Dec. 2011.

[37] E. de Aguiar, L. Sigal, A. Treuille, and J. K. Hodgins. Stable spaces for real-time clothing. *ACM Trans. Graph.*, 29(4):106:1–106:9, Jul 2010.

[38] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. *ACM Trans. Graph.*, 27(3):98:1–98:10, Aug. 2008.

[39] M. de La Gorce, N. Paragios, and D. Fleet. Model-based hand tracking with texture, shading and self-occlusions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1–8, Jun 2008.

[40] P. Decaudin, D. Julius, J. Wither, L. Boissieux, A. Sheffer, and M.-P. Cani. Virtual garments: A fully geometric approach for clothing design. *Computer Graphics Forum*, 25(3):625–634, 2006.

[41] J. Deutscher and I. Reid. Articulated body motion capture by stochastic search. *International Journal of Computer Vision*, 61(2):185–205, Feb 2005.

[42] M. Eichner, M. Marin-Jimenez, A. Zisserman, and V. Ferrari. 2d articulated human pose estimation and retrieval in (almost) unconstrained still images. *International Journal of Computer Vision*, 99(2):190–214, Sept. 2012.

[43] P. Felzenszwalb. Representation and detection of deformable shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2):208–220, Feb 2005.

[44] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, Jan 2005.

[45] W.-W. Feng, Y. Yu, and B.-U. Kim. A deformation transformer for real-time cloth animation. *ACM Trans. Graph.*, 29(4):108:1–108:9, July 2010.

[46] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1–8, Jun 2008.

[47] O. Freifeld, A. Weiss, S. Zuffi, and M. J. Black. Contour people: A parameterized model of 2d articulated human shape. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 639–646, Jun 2010.

[48] A. Fuhrmann, C. Groß, V. Luckas, and A. Weber. Interaction free dressing of virtual humans. *Computers and Graphics*, 27(1):71–82, Feb 2003.

[49] S. Geman and D. McClure. Statistical methods for tomographic image reconstruction. *Bulletin Int. Statistical Institute*, pages 5–21, 1987.

[50] R. Goldenthal, D. Harmon, R. Fattal, M. Bercovier, and E. Grinspun. Efficient simulation of inextensible cloth. *ACM Trans. Graph.*, 26(3), July 2007.

[51] P. Guan, O. Freifeld, and M. J. Black. A 2D human body model dressed in eigen clothing. In *European Conference on Computer Vision, ECCV*, pages 285–298, 2010.

[52] P. Guan, L. Reiss, D. A. Hirshberg, A. Weiss, and M. J. Black. Drape: Dressing any person. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 31(4):35:1–35:10, July 2012.

[53] P. Guan, L. Sigal, V. Reznitskaya, and J. K. Hodgins. Multi-linear data-driven dynamic hair model with efficient hair-body collision handling. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '12, pages 295–304. ACM, 2012.

[54] P. Guan, A. Weiss, A. Balan, and M. J. Black. Estimating human shape and pose from a single image. In *IEEE International Conference on Computer Vision, ICCV*, pages 1381–1388, Oct 2009.

[55] S. Hadap, E. Bangerter, P. Volino, and N. Magnenat-Thalmann. Animating wrinkles on clothes. In *Proceedings of the conference on Visualization '99: celebrating ten years*, VIS '99, pages 175–182, Los Alamitos, CA, USA, 1999. IEEE Computer Society Press.

[56] S. Hadap and N. Magnenat-Thalmann. Modeling dynamic hair as a continuum. *Computer Graphics Forum*, 20(3):329–338, 2001.

[57] N. Hasler, C. Stoll, B. Rosenhahn, T. Thormählen, and H.-P. Seidel. Technical section: Estimating body shape of dressed humans. *Comput. Graph.*, 33(3):211–216, Jun 2009.

[58] A. Hilton, D. Beresford, T. Gentils, R. S. Smith, W. Sun, and J. Illingworth. Whole-body modelling of people from multiview images to populate virtual worlds. *The Visual Computer*, 16(7):411–436, 2000.

[59] G. Hinton. Using relaxation to find a puppet. In *Proc. of the A.I.S.B. Summer Conference*, pages 148–157, 1976.

[60] M. Hofmann and D. M. Gavrila. Multi-view 3d human pose estimation in complex environment. *International Journal of Computer Vision*, 96(1):103–124, Jan. 2012.

[61] M. Hofmanna and D. M. Gavrila. 3d human model adaptation by frame selection and shape-texture optimization. *Computer Vision and Image Understanding*, 115(11):1559–1570, Nov 2011.

[62] D. House and D. Breen. *Cloth Modeling and Animation*. AK Peters, Ltd., 2000.

[63] X. Huang, I. D. Walker, and S. Birchfield. Occlusion-aware reconstruction and manipulation of 3d articulated objects. In *IEEE International Conference on Robotics and Automation*, pages 1365–1371. IEEE, 2012.

[64] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, pages 559–568, New York, NY, USA, 2011. ACM.

[65] A. Jain, T. Thormählen, H.-P. Seidel, and C. Theobalt. Moviereshape: tracking and reshaping of humans in videos. *ACM Trans. Graph.*, 29(6):148:1–148:10, Dec. 2010.

[66] T. Jakobsen. Advanced character physics. In *Game Developers Conference*, pages 383–401, 2001.

[67] D. L. James and C. D. Twigg. Skinning mesh animations. *ACM Trans. Graph.*, 24(3):399–407, Jul 2005.

[68] S. Ju, M. J. Black, and Y. Yacoob. Cardboard people: a parameterized model of articulated image motion. In *Automatic Face and Gesture Recognition, Proceedings of the Second International Conference on Date of Conference*, pages 38–44, Jun 1996.

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

[70] L. Kavan, D. Gerszewski, A. W. Bargteil, and P.-P. Sloan. Physics-inspired upsampling for cloth simulation in games. *ACM Trans. Graph.*, 30(4):93:1–93:10, July 2011.

[71] L. Kavan, P.-P. Sloan, and C. O'Sullivan. Fast and efficient skinning of animated meshes. *Computer Graphics Forum*, 29(2):327–336, 2010.

[72] C. K. Koh and Z. Huang. A simple physics model to animate human hair modeled in 2d strips in real time. In *Proceedings of the Eurographic workshop on Computer animation and simulation*, pages 127–138, New York, NY, USA, 2001. Springer-Verlag New York, Inc.

[73] M. Koster, J. Haber, and H.-P. Seidel. Real-time rendering of human hair using programmable graphics hardware. In *Proceedings of the Computer Graphics International*, CGI '04, pages 248–256, Washington, DC, USA, 2004. IEEE Computer Society.

[74] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, Feb 1994.

[75] H. J. Lee and Z. Chen. Determination of 3d human body postures from a single view. *Computer Vision Graphics and Image Processing*, 30(2):148–168, 1985.

[76] W.-S. Lee, J. Gu, and N. Magnenat-Thalmann. Generating animatable 3d virtual humans from photographs. *Comput. Graph. Forum*, 19(3):1–10, 2000.

[77] S. Miller, M. Fritz, T. Darrell, and P. Abbeel. Parametrized shape models for clothing. In *IEEE International Conference on Robotics and Automation*, pages 4861–4868, Shanghai, China, 2011. IEEE.

[78] F. Moreno-Nogueer and J. Porta. Probabilistic simultaneous pose and non-rigid shape recovery. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1289–1296, Jun 2011.

[79] G. Mori, X. Ren, A. Efros, and J. Malik. Finding and tracking people from the bottom up. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 467–474, Jun 2003.

[80] A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson. Physically based deformable models in computer graphics. *Computer Graphics Forum*, 25(4):809–836, 2006.

[81] H. Nguyen and W. Donnelly. Hair animation and rendering in the nalu demo. *GPU Gems 2*, 2005.

[82] F. Oliveira and J. Tavares. Algorithm of dynamic programming for optimization of the global matching between two contours defined by ordered points. *Computer Modeling in Eng. & Sciences*, 31(1):1–12, 2008.

[83] S. Paris, W. Chang, O. I. Kozhushnyan, W. Jarosz, W. Matusik, M. Zwicker, and F. Durand. Hair photobooth: geometric and photometric acquisition of real hairstyles. *ACM Trans. Graph.*, 27(3):30:1–30:9, Aug. 2008.

[84] R. Plankers and P. Fua. Articulated soft objects for multi-view shape and motion capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):63–83, 2003.

[85] T. Popa, Q. Zhou, D. Bradley, V. Kraevoy, H. Fu, A. Sheffer, and W. Heidrich. Wrinkling captured garments using space-time data-driven deformation. *Computer Graphics Forum*, 28(2):427–435, 2009.

[86] D. Pritchard and W. Heidrich. Cloth motion capture. *Computer Graphics Forum*, 22(3):263–271, 2003.

[87] K. Robinette, S. Blackwell, H. Daanen, M. Boehmer, S. Fleming, T. Brill, D. Hoeferlin, and D. Burnsides. Civilian American and European Surface Anthropometry Resource (CAESAR) final report. Technical Report AFRL-HE-WP-TR-2002-0169, US Air Force Research Laboratory, 2002.

[88] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. on Graphics.*, 23(3):309–314, 2004.

[89] D. Samaras and D. Metaxas. Incorporating illumination constraints in deformable models for shape from shading and light direction estimation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 322–329, Jun 1998.

[90] V. Scholz, T. Stich, M. Keckeisen, M. Wacker, and M. Magnor. Garment motion capture using color-coded patterns. *Computer Graphics Forum*, 24(3):439–447, 2005.

[91] A. Selle, M. Lentine, and R. Fedkiw. A mass spring model for hair simulation. *ACM Trans. Graph.*, 27(3):64:1–64:11, Aug. 2008.

[92] H. Seo, Y. I. Yeo, and K. Wohn. 3d body reconstruction from photos based on range scan. In *Proceedings of the First international conference on Technologies for E-Learning and Digital Entertainment*, Edutainment'06, pages 849–860, Berlin, Heidelberg, 2006. Springer-Verlag.

[93] L. Sigal. Continuous-state graphical models for object localization, pose estimation and tracking. *Doctoral Dissertation, Brown University*, May 2008.

[94] L. Sigal, A. Balan, , and M. J. Black. Combined discriminative and generative articulated pose and non-rigid shape estimation. In *Advances in Neural Information Processing Systems*, pages 1337–1344. MIT Press, 2008.

[95] L. Sigal, M. Isard, H. Haussecker, and M. J. Black. Loose-limbed people: Estimating 3d human pose and motion using non-parametric belief propagation. *International Journal of Computer Vision*, 98(1):15–48, May 2012.

[96] C. Sminchisescu and A. Telea. Human pose estimation from silhouettes. A consistent approach using distance level sets. *J. WSCG*, 10(1-3):413–421, Feb 2002. Special Issue: International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG 2002.

[97] C. Sminchisescu and B. Triggs. Building roadmaps of local minima of visual models. In *European Conference on Computer Vision, ECCV*, pages 566–582, 2002.

[98] H. Souto and S. Musse. Automatic detection of 2d human postures based on single images. In *Proceedings of the 2011 24th SIBGRAPI Conference on Graphics, Patterns and Images*, SIBGRAPI '11, pages 48–55, Washington, DC, USA, 2011. IEEE Computer Society.

[99] N. Sprague and J. Luo. Clothed people detection in still images. In *International Conf. on Pattern Recognition*, pages 585–589, 2002.

[100] J. Stark and A. Hilton. Surface capture for performance-based animation. *Computer Graphics and Applications*, 27(3):21–31, 2007.

[101] M. Straka, S. Hauswiesner, M. Rüther, and H. Bischof. Skeletal graph based human pose estimation in real-time. In *Proceedings of the British Machine Vision Conference*, pages 69.1–69.12, 2011.

[102] R. W. Sumner and J. Popović. Deformation transfer for triangle meshes. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 23(2):399–405, July 2004.

[103] S. Tariq and L. Bavoil. Real time hair simulation and rendering on the gpu. In *ACM SIGGRAPH 2008 talks*, SIGGRAPH '08, pages 37:1–37:1, New York, NY, USA, 2008. ACM.

[104] C. J. Taylor. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. *Computer Vision and Image Understanding*, 80(10):349–363, Oct 2000.

[105] C. Theobalt, N. Ahmed, H. Lensch, M. Magnor, and H.-P. Seidel. Seeing people in different light – joint shape, motion, and reflectance capture. *IEEE Trans. Visual. Comp. Graph*, 13(4):663–674, 2007.

[106] A. Treuille, A. Lewis, and Z. Popović. Model reduction for real-time fluids. *ACM Trans. Graph.*, 25(3):826–834, July 2006.

[107] N. Umetani, D. M. Kaufman, T. Igarashi, and E. Grinspun. Sensitive couture for interactive garment modeling and editing. *ACM Trans. Graph.*, 30(4):90:1–90:12, Jul 2011.

[108] M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *European Conference on Computer Vision, ECCV*, pages 447–460, 2002.

[109] M. A. O. Vasilescu and D. Terzopoulos. Multilinear image analysis for facial recognition. In *International Conf. on Pattern Recognition*, pages 511–514, 2002.

[110] C. C. L. Wang, Y. Wang, and M. M. F. Yuen. Feature based 3D garment design through 2D sketches. *Computer-Aided Design*, 35(7):659–672, June 2003.

[111] H. Wang, F. Hecht, R. Ramamoorthi, and J. O'Brien. Example-based wrinkle synthesis for clothing animation. *ACM Trans. Graph.*, 29(4):107:1–107:8, July 2010.

[112] K. Ward, F. Bertails, T.-Y. Kim, S. Marschner, M.-P. Cani, and M. Lin. A survey on hair modeling: Styling, simulation, and rendering. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):213–234, 2007.

[113] K. Ward, M. C. Lin, J. Lee, S. Fisher, and D. Macri. Modeling hair using level-of-detail representations. In *Proceedings of the 16th Int Conf on Computer Animation and Social Agents*, CASA '03, pages 153–160, 2005.

[114] K. Ward, M. Simmons, A. Milne, H. Yosumi, and X. Zhao. Simulating rapunzel's hair in disney's tangled. In *ACM SIGGRAPH 2010 Talks*, SIGGRAPH '10, pages 22:1–22:1, New York, NY, USA, 2010. ACM.

[115] Y. Watanabe and Y. Suenaga. A trigonal prism-based method for hair image generation. *Computer Graphics and Applications, IEEE*, 12(1):47–53, 1992.

[116] A. Weiss, D. Hirshberg, and M. Black. Home 3d body scans from noisy image and range data. In *IEEE International Conference on Computer Vision, ICCV*, pages 1951–1958, 2011.

[117] R. White, K. Crane, and D. A. Forsyth. Capturing and animating occluded cloth. *ACM Trans. Graph.*, 26(3), Jul 2007.

[118] Wikipedia. Motion capture. *http://en.wikipedia.org/wiki/Motion_ capture/*.

[119] J. xiang Chai, J. Xiao, and J. Hodgins. Vision-based control of 3d facial animation. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 193–206, Jul 2003.

[120] K. Yamaguchi, H. Kiapour, L. E. Ortiz, and T. L. Berg. Parsing clothing in fashion photographs. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 3570–3577, 2012.

[121] X. D. Yang, Z. Xu, T. Wang, and J. Yang. The cluster hair model. *Graphic Models*, 62(2):85–103, Mar 2000.

[122] C. Yuksel, S. Schaefer, and J. Keyser. Hair meshes. *ACM Trans. Graph.*, 28(5):166:1–166:7, Dec. 2009.

[123] C. Yuksel and S. Tariq. Advanced techniques in real-time hair rendering and simulation. In *ACM SIGGRAPH 2010 Courses*, SIGGRAPH '10, pages 1:1–1:168, New York, NY, USA, 2010. ACM.

[124] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah. Shape from shading: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):690–706, Aug 1999.

[125] S. Zhou, H. Fu, L. Liu, D. Cohen-Or, and X. Han. Parametric reshaping of human bodies in images. *ACM Trans. Graph.*, 29(4):126:1–126:10, Jul 2010.

[126] S. Zuffi, O. Freifeld, and M. Black. From pictorial structures to deformable structures. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 3546–3553, 2012.