ABSTRACT

Title of dissertation: Single-View 3D Reconstruction of Animals

Angjoo Kanazawa, Doctor of Philosophy, 2017

Dissertation directed by: David Jacobs
Department of Computer Science

Humans have a remarkable ability to infer the 3D shape of objects from just a single image. Even for complex and non-rigid objects like people and animals, from just a single picture we can say much about its 3D shape, configuration and even the viewpoint that the photo was taken from. Today, the same cannot be said for computers – the existing solutions are limited, particularly for highly articulated and deformable objects. Hence, the purpose of this thesis is to develop methods for single-view 3D reconstruction of non-rigid objects, specifically for people and animals. Our goal is to recover a full 3D surface model of these objects from a single unconstrained image. The ability to do so, even with some user interaction, will have a profound impact in AR/VR and the entertainment industry. Immediate applications are virtual avatars and pets, virtual clothes fitting, immersive games, as well as applications in biology, neuroscience, ecology, and farming. However, this is a challenging problem because these objects can appear in many different forms.

This thesis begins by providing the first fully automatic solution for recovering a 3D mesh of a human body from a single image. Our solution follows the classical paradigm of bottom-up estimation followed by top-down verification. The key is to solve for the mostly

likely 3D model that explains the image observations by using powerful priors. The rest of the thesis explores how to extend a similar approach for other animals. Doing so reveals novel challenges whose common thread is the lack of specialized data. For solving the bottom-up estimation problem well, current methods rely on the availability of human supervision in the form of 2D part annotations. However, these annotations do not exist in the same scale for animals. We deal with this problem by means of data synthesis for the case of fine-grained categories such as bird species. There is also little work that systematically addresses the 3D scanning of animals, which almost all prior works require for learning a deformable 3D model. We propose a solution to learn a 3D deformable model from a set of annotated 2D images with a template 3D mesh and from a few set of 3D toy figurine scans. We show results on birds, house cats, horses, cows, dogs, big cats, and even hippos. This thesis makes steps towards a fully automatic system for single-view 3D reconstruction of animals. We hope this work inspires more future research in this direction.

Single-view 3D Reconstruction of Animals

by

Angjoo Kanazawa

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2017

Advisory Committee:

Professor David Jacobs, Chair/Advisor

Professor Rama Chellappa

Professor Hal Daumé III

Professor Larry Davis

Professor Thomas Goldstein

Professor Matthias Zwicker

To my loving parents.

Acknowledgments

This thesis is the result of positive energy and encouragement of all of my mentors, collaborators, peers, friends and family, without whom I would not have been able to complete. I thank everyone I came across during and before this journey!

I cannot thank my advisor David Jacobs enough for all of his support, knowledge and patience over the years I spent at University of Maryland. Through thick and thin David's door was always open. With his east facing office, his open door cast a lot of light onto the corridor of AVW building and also onto my path as a researcher and a person.

I am so grateful to have had consistent access to David's acute insight, uncanny ability to reduce the problem to its simplest form and point out counter examples early on. He taught me how to internalize and think about problems intuitively. His classes are also excellent, every student at UMD should attend his lectures particularly the ones on Fourier transforms and dimensionality reduction. I am grateful for the amount of intellectual freedom he provided, guiding yet letting me discover the path that I wanted to take. I am indebted for his patience and support when I wanted to change the topic I've been working on and barged into the office saying I wanted to reconstruct cats instead. His emphasis for quality over quantity helped me stay sane and navigate the ever fast changing academic world. David also always put me first and was very generous in allowing me to explore a lot of opportunities outside his lab. In these occasions I was really lucky to have met other close mentors that helped me shape my thesis.

I am thankful to Manmohan Chandraker for trusting me with an ambitious project at NEC during my internship. Manmohan is an excellent role model for me in his uncompromising pursuit of high quality research while striking a balance between the fundamental and the practical. Manmohan also taught me practical skills in designing experiments and managing the paper writing process.

To Michael Black, I am so grateful for picking me up from the pile of arXiv papers. His support came at the most necessary time when I was feeling dejected, when the paper that I was finally proud of got rejected at a conference (this later won the best paper award). Despite his busy schedule, Michael provided mentorship and in-depth research discussions with a sparkling alacrity. I learned a lot, in particular, how to tell a good story. Michael's energy is contagious and his child-like excitement towards new projects and ideas has been inspirational. The environment he fostered in his lab was lively and I treasure the friendship I discovered at MPI.

I like to thank Ronen Basri for his intuitive explanations of equations and casual willingness to consult my math problems. I thank Peter Belheumeur for guiding me in my early years, teaching me the valuable skills of being organized and keeping track of research progress. A big thanks to Rob Fergus, my undergraduate advisor at NYU for rescuing me from Wall Street and introducing research and David Jacobs to me.

I thank my committee members: Rama Chellappa, Hal Daumé III, Larry Davis, Thomas Goldstein, and Matthias Zwicker for insightful comments, suggestions and questions. Special thanks to Rama for teaching me image processing and Hal for his super amazing Machine Learning class.

At all the places I visited, there were wonderful, wonderful people that made my life fun and I learned a lot from.

Jiongxing Liu and Thomas Berg were my first research officemates – they were great company, I had a lot of fun exchanging ideas and getting lunch around Columbia. I thank Shahar Kovalsky, my optimization guru that I'm so fortunate to have worked with.

Thanks to all the members of the Jacobs lab: Carlos Castillo, Daozheng Chen, Anne Jorstad, Joao Soares, Arijit Biswas, Abhishek Sharma, Jin Sun, Soumyadip Sengupta, Hao

Zhou, Abhay Kumar. As well as other members of UMD: Raviteja Vemulapalli, Kota Hara, Mohammad Rastegari, Ejaz Ahmed, Jonghyun Choi, Sameh Khamis, Aleksandrs Ecins, Francisco Barranco, Gregory Kramida, Kotaro Hara, Ioana Bercea. I fondly remember our CVSS seminars with the bad-but-free pizza, taking classes with you all, random research discussions in the office and after deadline celebrations.

Special thanks to Vassilios Lekakis who taught me how to Pomodoro, probably the most important skill I learned during my PhD.

Of course there are the wonderful staffs of UMIACS/UMD: Janice Perrone, Arlene E Schenk, Jennifer Story and Fatima Bangura amongst others without whom I would've been in big trouble.

I like to thank Wanyen Lo, Abhijit Ogale, Andreas Wendel, Yuandong Tian, Dave Ferguson, and the rest of the Magic team for their guidance and friendship at Google X. Chia-yin Tsai for all the positive energy and helping me not take myself so seriously. Thanks to Luis, Batu, Ranjay and Kia for keeping it fun. Thanks Chris Choy and Yu Xiang for fun lunches and providing feedback at NEC.

Oh wonderful members of MPI: Federica Bogo, Javier Romero, Laura Sevilla, Fatma Güney, Christoph Lassner, Thomas Nestmeyer, Varun Jampani, Sergey Prokudin, Sergi Pujades, Peter Gheler, Andreas Geiger, Gerard Pons-Moll, Naureen Mahmood, Alejandra Quiros-Ramirez, Jonas Wulff, Osman Ulusoy, Naejin Kong, Judith Butepage, Maren Mahsereci, Silvia Zuffi, Melanie, Nicole and Roco! Also thanks to Gabbie Ays for providing me with such a beautiful German home. I had an amazing time in Tübingen thanks to you all.

There are also other mentors I would like to thank over my years at UMD and NYU: Howard Elman for his advice and mentorship. Margaret Wright, a role model and also one of the first to introduce research to me. Joel Spencer for teaching me the joy of math.

Thanks to all my other friends and family who have supported me and got here where I am: members of the ACM and WinC at NYU, founders of Diaspora, Team Awesome + adlott, Lisa-Burt-Ryusuke-Stephen, Kazu san, Agnes and Mikey, Cheriko, Wara-geppu, Sha-geppu, and All-mighty Endo for all of the laughter, friendship, and love. Thanks to all other Marist families, Kobo and Ranni family, and Mr. DeCouto my first Math teacher! David and Cindy, thanks for taking care of me in the end and coming to my defense, your support really meant a lot.

I sincerely thank my parents, who raised me with so much love and support, who told me that I can be anything I want to be.

Lastly but not least, thank you Austin, my best friend and love who shared every moment of struggle and excitement with me on this long journey. Thank you!!

Contents

List of Tables

List of Figures

Chapter 1

Introduction

"Nothing is so fleeting as form; yet never does it quite deny itself."

— Ralph Waldo Emmerson

1.1 Objective

Although there has been a significant progress in the field of 3D reconstruction, most of the research assumes the availability of multiple views or range sensors and much less attention has been given to the problem of 3D reconstruction from a single image. This is partly due to the fact that geometrically, recovering 3D points from a monocular camera is an ill-posed problem. However, humans have the ability to see the "unseen" from just a single picture. That is, for a familiar object we can make a good guess of what it would look like from another viewpoint. Even for complex, non-rigid objects like people and animals, from just a single picture we can say much about its 3D shape, configuration, and even the viewpoint that the photo was taken from. Today, the solutions we have for this problem is limited, and not much has been explored particularly for highly articulated and deformable objects. Hence, the purpose of this thesis is to develop methods for single-view 3D reconstruction of

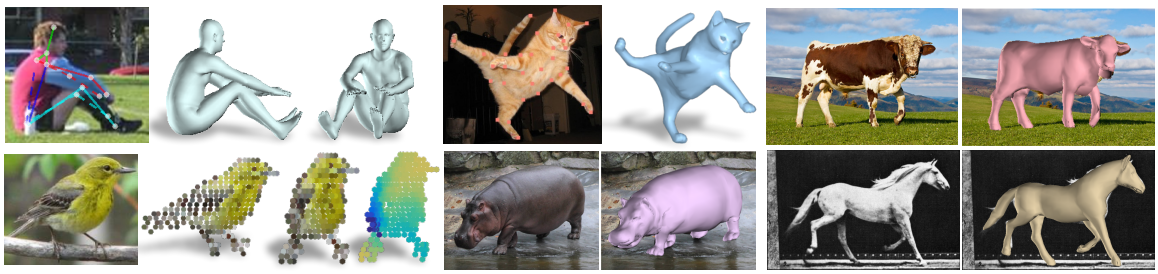


Figure 1-1: **Sample results of this thesis.**

non-rigid objects, specifically of people and animals.

The ultimate goal is to recover the full 3D surfaces of objects from a single unconstrained image "in-the-wild" in a fully automatic manner. The object category may be known, but no other information such as camera calibration, scene, or lighting is known. The kinds of 3D representations explored in this thesis are a set of 3D point clouds or a 3D mesh with interpretable parameters such as joint angles or axes of shape variations. This thesis develops the first fully automatic solution for recovering 3D mesh of a human body from a single image. Extending a similar approach to animals exposes novel challenges and methods to overcome them are explored. See Figure 1-1 for examples of results that are obtained in this thesis.

1.2 Motivation

To see why this is a problem worth solving, let us first consider what it means to recover a 3D representation from a single image. In the traditional multi-view setting, 3D reconstruction is analogous to solving the correspondence between images of the same object taken from different views. But in the single-view case, this is analogous to solving the correspondence between images of *different* instances of the same category. Because in essence, the 3D shape is inferred by relating the observed image with images seen in the past or with a model

encapsulating the knowledge of the 3D shape of the object category. Since different object instances have different underlying 3D shapes, the established correspondences between them are the same in the semantic sense. Such correspondences are one of the key ingredients for many vision tasks such as recognition, segmentation, image search, image morphing, synthesis, *etc*. So in the single-view case, the process by which 3D is recovered is as important as the final output. For this reason, the resulting 3D representation is a lot more meaningful than the precise 3D representation of the object that can be geometrically recovered from multiple views.

This thesis focuses on reconstructing people and animals, leaving man-made objects and scenes to future work. People and animals are arguably the most interesting subjects; they occupy a significant portion of the visual data on the Internet. People take approximately 93 million selfies per day from android devices alone [184] and a study shows that people are twice as likely to upload a picture of cats instead of selfies [153]. Also, they are governed by a single topology within each class, making the semantic correspondence across its surface well defined compared to man-made objects like chairs whose topology may vary or categories that do not have a unified 3D form such as scenes.

There are many practical applications of single-view 3D reconstruction. To begin with, most of the image data on the Internet are captured from a single camera. Given the advancements of applications that use 3D models like 3D printing, augmented and virtual reality (AR/VR), the ability to obtain 3D models from the real world is more relevant than ever, even if it requires some user assistance. Immediate applications of people and animals in VR are virtual avatars and pets, virtual clothes fitting, immersive games and much more. Consumers are also keen on getting their pets 3D printed. Monocular markerless motion capture is also valuable for the film/entertainment industry. The recovered 3D models can

be textured and rendered from novel viewpoints for image synthesis and manipulation in Computer Graphics. Further, analysis of humans and animals in 3D has many application in biology, neuroscience, ecology, farming, *etc*.

1.3 Challenges

One way to solve the single-view 3D reconstruction problem is by applying the classical paradigm of bottom-up estimation followed by top-down verification. Bottom-up estimation detects salient image features or key parts by solving some form of a correspondence problem. The top-down verification takes a deformable 3D model of the object category and fits the model to the bottom-up observations. The seminal work of Blanz and Vetter [32] began this direction by learning a low-dimensional 3D model of a face from 3D scans and fitting it to images using texture cues. However, it's not straight forward to adapt this approach for humans and animals because they exhibit a lot more articulation, appearance and shape variability, making both the correspondence and model fitting problem more challenging. The few works that attempt 3D mesh reconstruction of human bodies from a single image require perfect silhouettes and manually annotated correspondences. Even then, the results have only been shown for simple poses. One reason the fitting is hard is because articulated limbs exacerbate the inherent depth ambiguity problem in monocular reconstruction. Many configurations of the body satisfy the same 2D constraints, and not many configurations produce solutions that are physically plausible or realistic. These problems are even more challenging for animals where it's not clear how to even learn a 3D deformable model due to the lack of 3D scans.

1.4 A Solution for Human Bodies

This thesis begins by focusing on the case with human bodies, a subject of much interest and progress in the past years. By combining the recent developments in automatic 2D joint detection and 3D human modeling, we provide the first fully automatic solution that produces a 3D mesh of a human body from a single image. Our success owes much to the availability of deep learning based accurate 2D joint detectors and a high quality, fully differentiable generative 3D model that models articulation with a kinematic skeleton and the 3D surface with a statistical shape model. We further endow the model with a pose prior learned from 3D data and the ability to reason about interpenetration, avoiding impossible poses. Having a strong model helps reduce ambiguity, making the problem easier. This result demonstrates that the bottom up and top down paradigm is an effective solution for articulated and deformable objects. The rest of the thesis explores how to extend these components to other animals such as birds, house cats, horses, cows, dogs, big cats, and even hippos.

1.5 Extension to Animals

1.5.1 Weakly-supervised Solution for Correspondences

We first explore the challenges in solving the bottom-up correspondence problem for animals within the scope of fine-grained categories, taking bird species as an example. Because semantic correspondences are difficult to find based on appearance similarities alone, many methods rely on the availability of human supervision in the form of part or keypoint annotations. These annotations are used to learn part detectors or used as a bootstrap to establish dense correspondences. While these annotations are readily available for faces and

humans, they do not exist in the same scale, if at all, for animals. Collecting them is not only labor-intensive but also ambiguous and difficult. Thus we present a framework for matching images of objects with some degree of non-rigidity and articulation across sub-categories and pose variations without requiring supervised annotations. The key idea is to take advantage of the structure in fine-grained categories to create synthetic data, from which a bird-specific 2D deformation model is learned using a convolutional neural network (CNN). The learned model is used as a spatial prior, which significantly improves the matching quality.

1.5.2 Learning a Deformable 3D Model of Animals

The rest of the thesis explores 3D model building for animals. The best practices of humans tell us that 3D deformation is well modeled by separating it into two factors, one that models changes due to pose and one that models changes due to shape differences between individuals [17, 93, 143]. However, most prior work learns such models from a large set of registered 3D scans of people in various shapes and poses. Such 3D scans are hard to acquire for animals, because it is impractical to bring a large number of animals into a lab environment for scanning, especially when the animals are wild and rare. Furthermore, unlike humans, animals are not cooperative – we cannot ask them to stay in a certain pose while we scan them. Artists may create 3D models of animals, but this is also expensive and may lack realism.

On the other hand, animal photographs are much easier to acquire. We propose a method for learning a 3D deformation model for changes due to pose using a set of user-annotated 2D images and a template 3D mesh. We depart from the kinematic skeleton to model pose since designing such a structure requires a priori knowledge of how the animal deforms (*i.e.* how many bones to use). Instead, we model articulation using a continuous stiffness field

that governs the amount of deformation allowed for each local region. The key intuition is that highly deformable regions are sparse. We learn this from multiple images at once by forcing sparsity on the stiffness field while the template is deformed to fit each image. We demonstrate this on cats and horses.

We then learn a shape model by scanning a few dozen realistic toy figurines of a range of quadruped animals. Since there are not enough scans for each animal species to learn a class-specific shape model, we learn a multi-species shape model by sharing information that is common across quadrupeds. Learning a statistical shape model requires that all the 3D data must be in correspondence. This involves registering a common template mesh to every scan, but this is challenging since the shape variation across quadruped species far exceeds the kind of variation seen between humans. Moreover, these toys have different shape *and* pose, while the human data for learning shape consists of various individuals scanned in a common neutral pose. We propose a multi-stage registration process where a novel analytical shape model is used to roughly align the scans to kick off the process. During fitting we use silhouettes to obtain more accurate shape fits. Despite being trained on toy scans, our model generalizes to images of real animals, capturing their shape well.

1.6 Thesis outline

The structure of the thesis is as follows: Chapter 2 goes over relevant prior art. Chapter 3 proposes the fully automatic solution for human bodies. The rest of the chapters explore challenges in extending a similar approach to animals and methods to overcome them. Chapter 4 explores the challenges in solving the bottom-up correspondence problem for a particular subset of animals like birds. Chapter 5 explores how a model that captures 3D deformations due to pose changes may be learned from a set of 2D images and a template

3D mesh. Chapter 6 explores how to model the shape variability across quadrupeds from a few dozen set of scanned 3D toy figurines. Finally, we conclude and discuss future directions in Chapter 7.

Chapter 2

Background

2.1 A Brief History of Point Correspondences

To do any kind of 3D reconstruction, we need to solve the correspondence problem between image pairs, this problem is also referred to as image matching. Initial efforts in matching an image pair started in stereo [88] and optical flow [100, 148], which used pixel intensity values to find correspondences based on the brightness constancy assumption. Since then, much of the research have been devoted to detecting and designing robust, reliable low and mid-level appearance features to match, starting from SSD [15], corner detectors [90] filter banks [113], shape context for silhouettes [29], SIFT [147], HOG[61], DAISY [209], VLAD [19] and more until the early 2010s. Most of these works focus on matching the same object instance from different viewpoints or subsequent frames in video.

We focus our discussion on methods that solve for correspondence between images of *different* object instances of the same class below. These methods can be divided into supervised and unsupervised methods, where in this context supervision refers to the use of human annotated point correspondences.

2.1.1 Supervised Methods

The point supervision acquired from humans for objects are, for reasons of efficiency and ease of annotation, 2D locations of salient object landmarks such as parts and body joints, generally referred to as "keypoints". This kind of supervision developed from success in object detection, which led to the problem of object *part* detection, in which the definition of parts progressed from rough bounding boxes to finer-scale points. As such, many methods refer to this semantic correspondence problem as keypoint localization. The first instantiation of this kind of annotation was for faces, in which from several face parts or "fiducial points" were labeled for aligning faces and improving face recognition [28, 46, 242, 175]. Many datasets of fiducial points were proposed such as BioID [110], labeled face parts in-the-wild [28], and annotated faces in-the-wild [242]. Research in human detection also progressed from simple bounding box representations [61] to rough parts [174], to appearance and configuration specific parts or poselets [38], to 2D joint detection [226, 112, 187, 80]. Features extracted at localized keypoints were shown to be particularly important for fine-grained classification [138, 72, 232], which motivated keypoint annotation of animal species such as dogs [138] and birds [217]. There is a large literature in each of these fields and a comprehensive summary is out of the scope of this thesis. Please see surveys [219] for fiducial point detection, [161] for human 2D pose detection, and [234] for fine-grained recognition.

Recent approaches use deep learning based methods to predict these keypoints from images. Initial works used CNNs to directly regress 2D locations for human pose [211] and fiducial points [200]. More recent works regress keypoint confidence maps [210] from which the final location is obtained using some form of a spatial prior (possibly another network) to remove outliers [107, 163, 164, 210, 140, 212, 166, 54] or by iterative refinement [47]. For human 2D joint detection, the most recent approaches have developed an effective architecture

that can directly output very accurate confidence maps by making use of multiple stages [220] or the hourglass structure that allows bottom-up, top-down inference [154]. Particularly for 2D human pose, these deep learning based approaches can obtain very reliable results for unconstrained images in-the-wild. But these approaches require a large set of annotated training data, which is expensive to obtain and even and ambiguous to label for other categories such as animals.

2.1.2 Unsupervised Methods

The problem of computing dense alignment between images of different scenes was first explored by the SIFT-flow algorithm [137], which use optical flow methods to match images with SIFT features. This was followed by Deformable Spatial Pyramid Matching [123], which uses a pyramid graph to regularize match consistency at multiple spatial extents. The PatchMatch algorithm [24, 25] uses a randomized search technique for efficiently finding approximate nearest-neighbors points. Bristow *et al*. solves for semantic correspondences by training an exemplar Linear Discriminant Analysis classifier [89] for each pixel; they show results on animals and humans but objects are in a relatively similar pose [41]. These approaches use purely geometric spatial priors for regularizing the matches and do not use category-specific semantic priors. Zhou *et al*. makes use of the class specific information by jointly solving correspondences across an image collection of an object category via enforcing cycle consistency [239]. Their approach is complementary with the method presented in Chapter 4, which learns a category-specific spatial prior also without any keypoint supervision.

2.2 3D Deformable Model of People and Animals

Blanz and Vetter [32] began the direction of building a statistical model of 3D faces by aligning 3D scans of faces and computing a low-dimensional shape model. The variability among faces, however, is much lower than human bodies or among animal species, making alignment of the training data much simpler. Additionally faces have much less articulation, again simplifying the modeling problem.

2.2.1 Modeling Articulated Bodies

Many works assume a model of articulation is provided by users or artists in the form of kinematic skeletons [84, 9, 203, 22, 228, 203] or painted stiffness [170]. Since obtaining such priors from users is expensive, many methods learn deformable models automatically from 3D data [32, 18, 17, 56, 63, 131, 183]. Anguelov *et al*. [18] use a set of registered 3D range scans of human bodies in a variety of configurations to construct skeletons using graphical models. Hasler *et al*. [94] proposes a method to automatically learn the skeletal structure for both pose and shape from example 3D models in various poses. Popa *et al*. [170] learn the material stiffness of animal meshes by analyzing a set of vertex-aligned 3D meshes in various poses. All methods use 3D data for modeling articulation. In Chapter 5, we propose a method that can learn a model of articulation from a set of annotated 2D images and a template 3D mesh.

2.2.2 Modeling Humans

Unlike faces, recovering a 3D model for the human body requires solving for both *pose* – the articulated posture of the limbs, and *shape* – the pose-invariant surface of the 3D human body. There is a long history of learning 3D shape and pose models of humans

[14, 17, 55, 93, 143]. See [143] for a more comprehensive overview. Note that pose and shape are not fully independent; certain poses change the shape of the 3D surface and joint locations are dependent on the individual body shape. Initial works either focus on modeling how the 3D surfaces changes with pose [12, 108], or modeling just the space of human shape variation with principal component analysis [13, 185]. SCAPE [17] is the first model that captures both body shape variation and pose-dependent shape changes in terms of triangle deformations. In this thesis we use the most recent SMPL [143] model, which also models both shape and pose-dependent shape but in terms of vertex displacements. SMPL combines a low-dimensional shape space with an articulated blend-skinned model, where the parameters are the coefficients of the shape space and the set of 3D rotations for each of the 23 joints. SMPL is learned from 3D scans of 4000 people in a common pose and another 1800 scans of 60 people in a wide variety of poses. A nice feature of SMPL is that all the model parameters are linear in its inputs, making it easy to differentiate and optimize.

2.2.3 Modeling Animals

There is little work that systematically addresses the 3D scanning [4] and modeling of animals. The range of sizes and shapes, together with the difficulty of handling live animals and dealing with their movement, makes traditional scanning difficult. Previous 3D shape datasets like TOSCA [42] have a limited set of 3D animals that are artist-designed and with limited realism. Chen et al. [56] model sharks by registering 11 different 3D shark models from the Internet and learning a shape space on it. Cashman *et al.* [50] learn a morphable model of dolphin shapes from 2D images. Ntouskos et al. [156] take multiple views of different animals from the same class, manually segment the parts in each view, and then fit geometric primitives to segmented parts. Favreau et al. [75] animate an artist created rigged 3D model

of an animal given a 2D video sequence. Reinert et al. [176] take a video sequence of an animal and extract a textured 3D model by using an interactive sketching and tracking approach. The 3D shape is obtained by fitting generalized cylinders to each sketched stroke over multiple frames. No methods try to learn a 3D shape space spanning multiple animal species. We do this in Chapter 6.

Related are mesh deformation techniques in Computer Graphics [37, 236, 36, 198, 195]. These methods take an existing 3D mesh and deform it to fit some user-supplied 3D positional constraints. Common objectives are minimization of the elastic energy [205] or preservation of local differential properties [136]. The solution can be constrained to lie in the space of natural deformations, by learning from a few set of artist created exemplar meshes [197, 199, 183, 66, 151, 170]. Often these experiments are carried out on animals and other non-rigid objects. These methods do not always separate pose and shape (for some non-rigid animals like octopus this is an advantage), but mostly end up modeling shape changes due to pose since the number of example meshes are limited. [37] offers an excellent survey on linear surface deformation methods.

2.3 Methods for Single-view Reconstruction

There are several methods for single-view reconstruction, and we divide the discussion into three different techniques: model-based, Shape from X, and those that solve for 3D using image collections.

Related are methods that directly estimate the depth value for general scenes or man-made objects [231, 98, 182, 86, 68, 223, 58], but we focus our discussion to methods that recover a 3D representation for non-rigid objects.

2.3.1 Model-based methods

Attempts to fit a parametric 3D model to a single image dates all the way back, in fact, to the first Computer Vision paper of Roberts in 1963 [177], where parameters and viewpoints of simple rectangular blocks were solved to reconstruct a 2D line image. The idea started getting traction in the early 80s [145, 167], with models like superquadrics [20], well into the 90's [146, 104]. Model-based methods were also prominent for 2D image understanding, where the models progressed from general geometric primitives [230] to class-specific deformable models that can fit to novel images in ways consistent with the training set, *i.e.* Active Shape Models [60]. These works inspired the seminal work of Blanz and Vetter [32], which built a high-resolution morphable model of a 3D face mesh and its texture from 3D scans and fit it to a single image. From a user provided initial alignment of the mean 3D face to the image, the algorithm solved for the parameters of the 3D shape and texture that minimized the residual differences between the rendered model and the image in an analysis-by-synthesis loop. Below we discuss recent approaches for faces, human skeletons, human 3D models, and other object categories. See [181] for a more in depth discussion on deformable surface 3D reconstruction from monocular images.

Faces Note that there is a long list of work that recover 3D face models from an interactive video stream where temporal cues such as optical flow [70] help make automatic point tracking more reliable [70, 165, 216, 51, 45, 180]. But there are only few methods that only use a single-view image or a collection of single-view images.

Kemelmacher-Shlizerman and Basri [118] use shape from shading cues to morph a single 3D reference mesh of a face to a target image. Hassner and Basri [96] solve for the depth from examples by referencing a 3D database based on image patch similarity. A 3D mesh

is recovered from the depth estimate and results are shown for segmented images of faces, humans, hands, and fish. The availability of accurate fiducial point detectors [28, 46] allows extension of these approaches to less-constrained, in-the-wild Internet images in a fully automatic manner [95, 119].

3D human pose Most 3D human reconstruction methods formulate the problem as finding a 3D *skeleton* such that its 3D joints project to known or estimated 2D joints. Note that the previous work often refers to this skeleton in a particular posture as a "shape.", but in this thesis we take shape to mean the pose-invariant surface of the human body in 3D and distinguish this from pose, which is the articulated posture of the limbs.

Initial methods make different assumptions about the statistics of limb-length variation. Lee and Chen [132] assume known limb lengths of a stick figure while Taylor [202] assumes the ratios of limb lengths are known. Parameswaran and Chellappa [160] assume that limb lengths are isometric across people, varying only in global scaling. Barron and Kakadiaris [26] build a statistical model of shape variation from extremes taken from anthropometric tables. Jiang [111] takes a non-parametric approach, treating poses in the CMU dataset [5] as exemplars.

Recent methods typically use the CMU dataset and learn a statistical model of pose. The formulation of these methods is similar to that of non-rigid structure from motion [40], except that a 3D basis is learned a priori from the CMU dataset. Both [173, 71] learn a dictionary of poses and use a fairly weak anthropometric model on limb lengths to resolve ambiguities. Akhter and Black [9] take a similar approach, but add a novel pose prior that captures pose-dependent joint angle limits. Zhou et al. [240] also learn a dictionary but they create a sparse basis that also captures how these poses will appear from different camera

views. They show that the resulting optimization problem is easier to solve. Pons-Moll et al. [168] take a different approach to model pose. They estimate qualitative information (posebits) from mocap and then relate these to 3D pose.

All these previous methods work hard to deal with the fact that the problem is ambiguous, pose is non-linear, and that optimization is hard. They all use weak models of the body and the statistics of limb lengths. In several cases they normalize the problem so that limb lengths do not appear in the formulation. In contrast, in Chapter 3, we argue that using a much stronger 3D generative model of body shape , learned from thousands of people, can capture the anthropometric constraints of the population. Having a strong model helps reduce ambiguity, making the problem easier. Also, the availability of 3D surface allows modeling of interpenetration, avoiding impossible poses. The result is that our optimization problem is simpler to formulate and relies on fewer assumptions.

None of the methods above are fully automatic from a single image, most assume known correspondences, and some involve significant manual intervention. There are, however, a few methods that try to solve the entire problem of inferring 3D pose from a single image.

Simo-Serra et al. [189, 190] take into account that 2D part detections are unreliable and formulate a probabilistic model that estimates the 3D pose and 2D joint detections jointly. Wang et al. [218] use a weak model of limb lengths [132] but exploit automatically detected joints in the image and match to them robustly using an L1 distance. Zhou et al. [235] run a 2D pose detector [225] and then optimize 3D pose, automatically rejecting outliers. Akhter and Black [9] run a different 2D detector [122] and show results for their method on a few images. Both methods are only evaluated qualitatively. Yasin et al. [227] take a non-parametric approach in which the detected 2D joints are used to look up the nearest 3D poses in a mocap dataset, which serves as the prior for 2D and 3D joints. Recent work [241]

uses a CNN to estimate 2D joint locations. They then fit 3D pose to this using a monocular video sequence. They do not show results for single images. Similarly [204] also uses a CNN but directly output the 3D joint locations from video sequences.

None of these automated methods estimate 3D body shape. In Chapter 3 we demonstrate a complete system that uses 2D joint detections and fits pose and shape to them from a single image.

3D Human Pose and Shape Here we refer to methods that output a 3D model of human bodies, by "3D model" we mean a dense 3D surface, such as a 3D mesh. Early works on humans fit coarse human models consisting of primitive geometric shapes related by a kinematic skeleton to silhouettes [191, 82, 7]. Several methods fit body shape and pose to multi-camera images or sequences: Balan *et al*. [21] fit SCAPE to multi-camera silhouettes. Jain *et al*. [106] fit a body to multiple frames with manual intervention. We focus our discussion on single-image methods.

Sigal *et al*. [188] assume that silhouettes are given, compute shape features from them, and then use a mixture of experts to predict 3D body pose and shape from the features. They use the SCAPE model to fit to the image silhouettes. This is not fully automatic because very accurate silhouettes are required. Guan et al. [84, 83] take manually marked 2D joints and first estimate the 3D pose of a stick figure using classical methods [132, 202]. They use this stick figure to pose the SCAPE model, project the model into the image and use this to segment the image with GrabCut [179]. They then fit the SCAPE shape and pose to a variety of features including the silhouette, image edges, and shading cues. They assume the camera focal length is known or approximated, the lighting is roughly initialized, and that the height of the person is known. They use an interpenetration term that models

each body part by its convex hull. They then check each of the extremities to see how many other body points fall inside it and define a penalty function that penalizes interpenetration. This does not admit easy optimization.

In similar work, Hasler et al. [92] fit a parametric body model to silhouettes. Typically, they require a known segmentation and a few manually provided correspondences. In cases with simple backgrounds, they use four clicked points on the hands and feet to establish a rough fit and then use GrabCut to segment the person. They demonstrate this on one image. Zhou et al. [237] also fit a parametric model of body shape and pose to a cleanly segmented silhouette using significant manual intervention. Chen et al. [56] fit a parametric model of body shape and pose to manually extracted silhouettes; they do not evaluate quantitative accuracy.

Most recently Kulkarni et al. [128] use an articulated 3D mesh model, together with a probabilistic programming framework to estimate body pose from single images. The use hand defined pose priors, deal with visually simple images, and do not evaluate 3D pose accuracy. In related work they estimate object shapes from single images but do this for simple rigid shapes and not human bodies.

To our knowledge, no previous method estimates *3D body shape* and pose directly from only *2D joints*. A priori, it may seem impossible, but we show in Chapter 3 that given a good statistical model, recovering shape works surprisingly well. This is enabled by using SMPL [143], which unlike SCAPE has explicit 3D joints that can be directly projected to 2D joints. SMPL also models how joint locations are related to the 3D surface of the body, enabling inference of shape from joints. Of course this will not be perfect as a person can have the exact same limb lengths with varying weight or muscles causing varying shape. SMPL, however, does not represent anatomical joints, rather it represents them as a function

of the surface vertices. This couples joints and shape during model training.

Other categories There are several works on recovering 3D models of human hands [64, 114]. Most follow a model-based fitting approach with a 3D generative model of hands with kinematic skeletons. Khamis *et al.* [120] learn a 3D shape model of hands can be from Kinect data. However, possibly due to the large space of deformation and self-occlusion, hands only seem to work well with range data [114] or at the least require a monocular video [64].

For animals, Cashman and Fitzgibbon [50] in their seminal paper learn a 3D model of animal shape from 2D images. Specifically, they learn a low-dimensional 3D model of animals such as dolphins from a set of user segmented and annotated images. The formulation is elegant but the approach suffers from an overly smooth shape representation; this is not so problematic for dolphins but for other animals it is. The key limitation, however, is that they do not model articulation. The method proposed in Chapter 5 is complementary to their approach in that the 3D deformation over pose is learned from a set of hand clicked 2D images.

Other methods use a reference 3D mesh and make use of strong image cues such as silhouettes and/or user interaction. Kraevoy *et al.* take a template 3D mesh of an object category and align and deform it to fit a contour drawing with user provided viewpoint initialization. They use a hidden Markov model to solve the correspondences between the 3D vertices and the contour points. They show results on human, dogs, bears, and cups. The results look good but are only tested on simple poses, and require sophisticated contour drawings from users. Their approach is closer to the related problem of sketch-based 3D modeling techniques [59].

Recently, Kholgade et al. introduced an exciting new photo editing tool that allows users to perform 3D manipulation by aligning a stock 3D models to 2D images [121]. The user selects the stock model closest to the target object, provides a mask for the ground and shadow and interactively aligns the object. The shape is deformed according to user provided 3D-to-2D correspondences and they also solve for the texture and lighting. Our approach complements this application, which is only demonstrated for rigid objects.

2.3.2 Single-view Shape from X

Shape from Shading [99] is one of the earliest single-view 3D reconstruction methods. As such, there is a long history and we refer to this survey [233] for details. Other derivatives of this form are Shape from Defocus [73, 74], Shape from Texture [222, 76] and Shape from Specularities [31]. These problems are closely related to that of intrinsic image decomposition, whose goal is to factorize an image into distinct scene properties such as reflectance, illumination and specularities [129, 201]. The most recent work of Barron and Malik unify these problems and provide an elegant solution to solve for the shape, reflectance and illumination from a single image of a masked object [27]. The key is to solve for the mostly likely explanations for all components jointly by using powerful priors. Since their approach is object agnostic, they do not recover the "other side" of the image that is not observed. Thus model-based methods complement these approach well, in fact Kar *et al*. use their approach for fitting a morphable model to images [117].

For general curved surfaces, there is a subset of Shape from Silhouette methods that can recover 3D models just from a single silhouette or contour drawings. This was pioneered by Terzopoulos in the late 80's [207, 206] but only worked for tube-like objects with genus 0. Prasad *et al*. [172, 171] extend these approaches to output a full 3D model for arbitrary

curved surfaces of higher genus using interactive contour labeling. Oswald *et al.* [159] reduce the amount of user interaction to few scribbles and a single volume hyperparameter using convex relaxation techniques. Vicente and Agapito [215] use these approaches to extract a 3D mesh template of animals from a reference image silhouette. The recovered mesh is used to recover the 3D model of novel images of different objects by deforming the mesh to fit a small set of user-annotated keypoints and silhouette. The results are of low resolution when applied to complex shapes.

There is also a line of work called *Shape from Template* that solves single-view reconstruction of a deformable open surface from a single image and a 3D template [85, 162]. Most of these methods focus on reconstructing inextensible surfaces such as a piece of paper or garments, where the length and the area are constant throughout deformation. They also require a template image of the exact same object as the target image. We deal with closed surfaces of articulated objects in this thesis.

2.3.3 3D from Image Collections

A new challenge in computer vision is to reconstruct a target object from a single image, using an image collection of similar objects [214, 117, 48]. Given a large enough image collection of an object category, the idea is to assume that for every target image there are at least a few images of different object instances that have a similar 3D shape, which can be used as surrogate viewpoints to apply traditional structure from motion (SfM) techniques. The seminal work of [214] demonstrates the possibility of a solution, but relies on ground truth part annotations to establish correspondences between the target and surrogate images as well as ground truth silhouettes.

The subsequent works of [117, 48] take a step further in using part annotations only

during training. Kar *et al*. [117] estimates the camera parameters using SfM on the ground truth part annotations, then learns a category-specific 3D shape basis using silhouettes. At test time, these morphable models are fit to predicted silhouettes by initializing the viewpoint using a CNN, and refining the final 3D model using a state-of-the-art intrinsic image decomposition method [27]. Carreira *et al*. [48] solves for dense correspondences between pairs of training images that are close in global pose. The annotated parts are used to regularize outliers during the matching process. These pairwise matches are propagated across the dataset by solving a shortest path problem so correspondences between images with wide-baselines can be obtained. At test time, the target image is matched to images of similar viewpoint, which establishes the correspondence to the rest of the training data. These correspondences are passed to SfM to obtain the final 3D points. In Chapter 4 we propose a method that do not require part annotations at training and test time.

Chapter 3

Automatic Estimation of 3D Human Pose and Shape from a Single Image

3.1 Introduction

Although the estimation of a 3D human body from a single image has been a long-standing problem with many applications, most previous approaches focus only on reconstructing the 3D joints and ignore the 3D human shape. In this chapter we present a *fully automatic solution* for 3D mesh reconstruction of the human body from a single image.

We solve the problem in two steps. First we estimate 2D joints using a recently proposed convolutional neural network (CNN) called DeepCut [166]. So far CNNs have been quite successful at estimating 2D human pose [107, 163, 164, 166, 211], but not 3D pose and shape from one image. Consequently we add a second step, which estimates 3D pose and shape from the 2D joints using a 3D generative model called SMPL [143]. The overall framework,

The contents of this work is in collaboration with Federica Bogo, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black, presented at ECCV 2016 [34]. This was part of my internship at Max Planck Institute for Intelligent Systems, Perceiving Systems.

Figure 3-1: **Example results.** 3D pose and shape estimated by our method for two images from the Leeds Sports Pose Dataset [112]. We show the original image (left), our fitted model (middle), and the 3D model rendered from a different viewpoint (right).

which we call "SMPLify", fits within a classical paradigm of bottom up estimation (CNN) followed by top down verification (generative model). A few examples are shown in Fig. 6-1.

There is a long literature on estimating 3D pose from 2D joints. Unlike previous methods, our approach exploits a high-quality 3D human body model that is trained from thousands of 3D scans and hence captures the statistics of shape variation in the population as well as how people deform with pose. Here we use the SMPL body model [143]. The key insight is that such a model can be fit to very little data because it captures so much information of human body shape.

We define an objective function and optimize pose and shape directly, so that the projected joints of the 3D model are close to the 2D joints estimated by the CNN. Remarkably, fitting only 2D joints produces plausible estimates of 3D body *shape*. We perform a quantitative evaluation using synthetic data and find that 2D joint locations contain a surprising amount of 3D shape information.

In addition to capturing shape statistics, there is a second advantage to using a generative 3D model: it enables us to reason about interpenetration. Most previous work in the area has estimated 3D stick figures from 2D joints. With such models, it is easy to find poses that are impossible because the body parts would intersect in 3D. Such solutions are very

common when inferring 3D from 2D because the loss of depth information makes the solution ambiguous.

Computing interpenetration of a complex, non-convex, articulated object like the body, however, is expensive. Unlike previous work [84, 83], we provide an interpenetration term that is differentiable with respect to body shape and pose. Given a 3D body shape we define a set of "capsules" that approximates the body shape. Crucially, capsule dimensions are linearly regressed from model shape parameters. This representation lets us compute interpenetration efficiently. We show that this term helps to prevent incorrect poses.

SMPL is gender-specific; i.e. it distinguishes the shape space of females and males. To make our method fully automatic, we introduce a gender-neutral model. If we do not know the gender, we fit this model to images. If we know the gender, then we use a gender-specific model for better results.

To deal with pose ambiguity, it is important to have a good pose prior. Many recent methods learn sparse, over-complete dictionaries from the CMU dataset [5] or learn dataset-specific priors. We train a prior over pose from SMPL models that have been fit to the CMU mocap *marker* data [5] using MoSh [142]. This factors shape from pose with pose represented as relative rotations of the body parts. We then learn a generic multi-modal pose prior from this.

We compare the method to recently published methods [9, 173, 240] using the exact same 2D joints as input. We show the robustness of the approach qualitatively on images from the challenging Leeds Sports Pose Dataset (LSP) [112] (Fig. 6-1). We quantitatively compare the method on HumanEva-I [187] and Human3.6M [102], finding that our method is more accurate than previous methods.

In summary our contributions are: 1) the first fully automatic method of estimating 3D

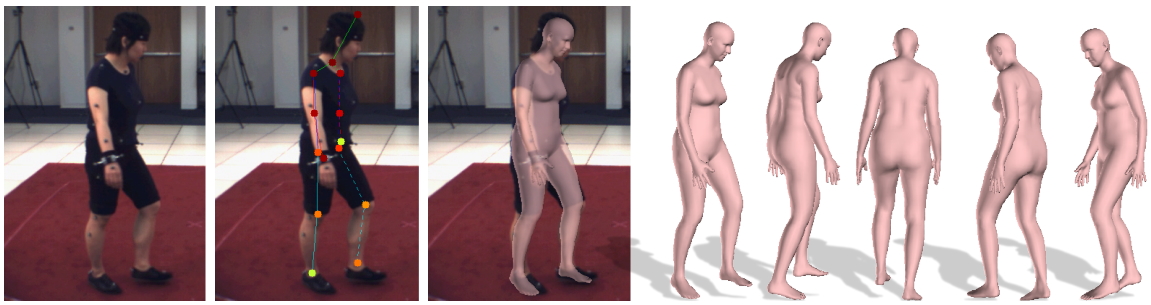


Figure 3-2: **System overview.** Left to right: Given a single image, we use a CNN-based method to predict 2D joint locations (hot colors denote high confidence). We then fit a 3D body model to this, to estimate 3D body shape and pose. Here we show a fit on HumanEva [187], projected into the image and shown from different viewpoints.

body shape and pose from 2D joints; 2) an interpenetration term that is differentiable with respect to shape and pose; 3) a novel objective function that matches a 3D body model to 2D joints; 4) for research purposes, we provide the code, 2D joints, and 3D models for all examples presented [1].

3.2 Method

Figure 4-3 shows an overview of our system. We take a single input image, and use the DeepCut CNN [166] to predict 2D body joints, J_{est}. For each 2D joint i the CNN provides a confidence value, w_i. We then fit a 3D body model such that the projected joints of the model minimize a robust weighted error term. In this work we use a skinned vertex-based model, SMPL [143], and call the system that takes a 2D image and produces a posed 3D mesh, *SMPLify*.

The body model is defined as a function $M(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\gamma})$, parameterized by shape $\boldsymbol{\beta}$, pose $\boldsymbol{\theta}$, and translation $\boldsymbol{\gamma}$. The output of the function is a triangulated surface, $\mathcal{M}$, with 6890 vertices. Shape parameters $\boldsymbol{\beta}$ are coefficients of a low-dimensional shape space, learned from a training set of thousands of registered scans. Here we use one of three shape models: male, female, and gender-neutral. SMPL defines only male and female models. For a fully

automatic method, we trained a new gender-neutral model using the approximately 2000 male and 2000 female body shapes used to train the gendered SMPL models. If the gender is known, we use the appropriate model. The model used is indicated by its color: pink for gender-specific and light blue for gender-neutral.

The pose of the body is defined by a skeleton rig with 23 joints; pose parameters $\boldsymbol{\theta}$ represent the axis-angle representation of the relative rotation between parts. Let $J(\boldsymbol{\beta})$ be the function that predicts 3D skeleton joint locations from body shape. In SMPL, joints are a sparse linear combination of surface vertices or, equivalently, a function of the shape coefficients. Joints can be put in arbitrary poses by applying a global rigid transformation. In the following, we denote posed 3D joints as $R_\theta(J(\boldsymbol{\beta})_i)$, for joint i, where R_θ is the global rigid transformation induced by pose $\boldsymbol{\theta}$. SMPL defines pose-dependent deformations; for the gender-neutral shape model, we use the female deformations, which are general enough in practice. Note that the SMPL model and DeepCut skeleton have slightly different joints. We associate DeepCut joints with the most similar SMPL joints. To project SMPL joints into the image we use a perspective camera model, defined by parameters K.

3.2.1 Approximating Bodies with Capsules

We find that previous methods produce 3D poses that are impossible due to interpenetration between body parts. An advantage of our 3D shape model is that it allows us to detect and prevent this. Computing interpenetration however is expensive for complex, non-convex, surfaces like the body. In graphics it is common to use proxy geometries to compute collisions efficiently [69, 208]. We follow this approach and approximate the body surface as a set of "capsules" (Fig. 3-3). Each capsule has a radius and an axis length.

We train a regressor from model shape parameters to capsule parameters (axis length and

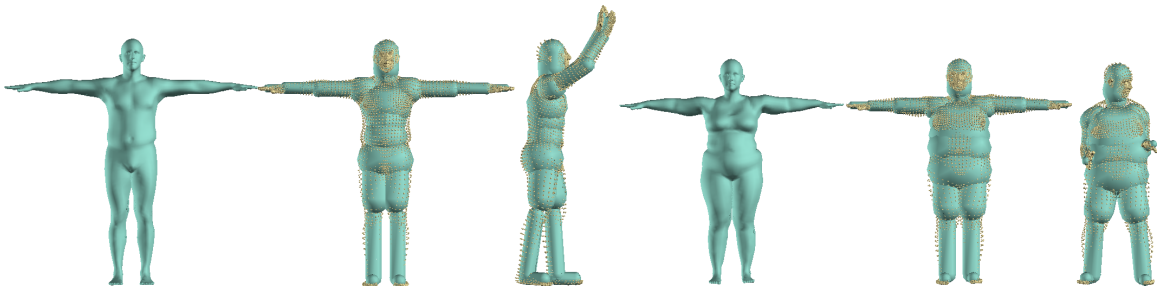


Figure 3-3: **Body shape approximation with capsules.** Shown for two subjects. Left to right: original shape, shape approximated with capsules, capsules reposed. Yellow point clouds represent actual vertices of the model that is approximated.

radius), and pose the capsules according to R_θ, the rotation induced by the kinematic chain. Specifically, we first fit 20 capsules, one per body part, excluding fingers and toes, to the body surface of the unposed training body shapes used to learn SMPL [143]. Starting from capsules manually attached to body joints in the template, we perform gradient-based optimization of their radii and axis lengths to minimize the bidirectional distance between capsules and body surface. We then learn a linear regressor from body shape coefficients, $\boldsymbol{\beta}$, to the capsules' radii and axis lengths using cross-validated ridge regression. Once the regressor is trained, the procedure is iterated once more, initializing the capsules with the regressor output. While previous work uses approximations to detect interpenetrations [169, 192], we believe this regression from shape parameters is novel.

3.2.2 Objective Function

To fit the 3D pose and shape to the CNN-detected 2D joints, we minimize an objective function that is the sum of five error terms: a joint-based data term, three pose priors, and a shape prior; that is $E(\boldsymbol{\beta}, \boldsymbol{\theta}) =$

$$E_J(\boldsymbol{\beta}, \boldsymbol{\theta}; K, J_{\text{est}}) + \lambda_\theta E_\theta(\boldsymbol{\theta}) + \lambda_a E_a(\boldsymbol{\theta}) + \lambda_{sp} E_{sp}(\boldsymbol{\theta}; \boldsymbol{\beta}) + \lambda_\beta E_\beta(\boldsymbol{\beta}) \tag{3.1}$$

where K are camera parameters and λ_θ, λ_a, λ_{sp} λ_β are scalar weights.

Our joint-based data term penalizes the weighted 2D distance between estimated joints, J_{est}, and corresponding projected SMPL joints:

$$E_J(\boldsymbol{\beta}, \boldsymbol{\theta}; K, J_{\text{est}}) = \sum_{\text{joint } i} w_i \rho(\Pi_K(R_\theta(J(\boldsymbol{\beta})_i)) - J_{\text{est},i}) \tag{3.2}$$

where Π_K is the projection from 3D to 2D induced by a camera with parameters K. We weight the contribution of each joint by the confidence of its estimate, w_i, provided by the CNN. For occluded joints, this value is usually low; pose in this case is driven by our pose priors. To deal with noisy estimates, we use a robust differentiable Geman-McClure penalty function, ρ [79].

We introduce a pose prior penalizing elbows and knees that bend unnaturally:

$$E_a(\boldsymbol{\theta}) = \sum_i \exp(\boldsymbol{\theta}_i), \tag{3.3}$$

where i sums over pose parameters (rotations) corresponding to the bending of knees and elbows. The exponential strongly penalizes rotations violating natural constraints (e.g. elbow and knee hyperextending). Note that when the joint is not bent, θ_i is zero. Negative bending is natural and is not penalized heavily while positive bending is unnatural and is penalized more.

Most methods for 3D pose estimation use some sort of pose prior to favor probable poses over improbable ones. Like many previous methods we train our pose prior using the CMU dataset [5]. Given that poses vary significantly, it is important to represent the multi-modal nature of the data, yet also keep the prior computationally tractable. To build a prior, we use poses obtained by fitting SMPL to the CMU marker data using MoSh [142]. We then

fit a mixture of Gaussians to approximately 1 million poses, spanning 100 subjects. Using the mixture model directly in our optimization framework is problematic computationally because we need to optimize the negative logarithm of a sum. As described in [158], we approximate the sum in the mixture of Gaussians by a max operator:

$$E_\theta(\boldsymbol{\theta}) \equiv -\log \sum_j (g_j \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}_{\theta,j}, \Sigma_{\theta,j})) \approx -\log(\max_j (c g_j \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}_{\theta,j}, \Sigma_{\theta,j}))) \tag{3.4}$$

$$= \min_j \left(-\log(c g_j \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}_{\theta,j}, \Sigma_{\theta,j}))\right) \tag{3.5}$$

where g_j are the mixture model weights of $N = 8$ Gaussians, and c a positive constant required by our solver implementation. Although E_θ is not differentiable at points where the mode with minimum energy changes, we approximate its Jacobian by the Jacobian of the mode with minimum energy in the current optimization step.

We define an interpenetration error term that exploits the capsule approximation introduced in Sec. 3.2.1. We relate the error term to the intersection volume between "incompatible" capsules (i.e. capsules that do not intersect in natural poses). Since the volume of capsule intersections is not simple to compute, we further simplify our capsules into spheres with centers $C(\boldsymbol{\theta}, \boldsymbol{\beta})$ along the capsule axis and radius $r(\boldsymbol{\beta})$ corresponding to the capsule radius. Our penalty term is inspired by the mixture of 3D Gaussians model in [196]. We consider a 3D isotropic Gaussian with $\sigma(\boldsymbol{\beta}) = \frac{r(\boldsymbol{\beta})}{3}$ for each sphere, and define the penalty as a scaled version of the integral of the product of Gaussians corresponding to "incompatible" parts

$$E_{sp}(\boldsymbol{\theta}; \boldsymbol{\beta}) = \sum_i \sum_{j \in I(i)} \exp\left(\frac{||C_i(\boldsymbol{\theta}, \boldsymbol{\beta}) - C_j(\boldsymbol{\theta}, \boldsymbol{\beta})||^2}{\sigma_i^2(\boldsymbol{\beta}) + \sigma_j^2(\boldsymbol{\beta})}\right) \tag{3.6}$$

where the summation is over all spheres i and $I(i)$ are the spheres incompatible with i. Note

that the term penalizes, but does not strictly avoid, interpenetrations. As desired, however, this term is differentiable with respect to pose and shape. Note also that we do not use this term in optimizing shape since this would bias the body shape to be thin to avoid interpenetration.

We use a shape prior $E_\beta(\boldsymbol{\beta})$, defined as

$$E_\beta(\boldsymbol{\beta}) = \boldsymbol{\beta}^T \Sigma_\beta^{-1} \boldsymbol{\beta} \tag{3.7}$$

where Σ_β^{-1} is a diagonal matrix with the squared singular values estimated via Principal Component Analysis from the shapes in the SMPL training set. Note that the shape coefficients $\boldsymbol{\beta}$ are zero-mean by construction.

3.2.3 Optimization

We assume that camera translation and body orientation are unknown; we require, however, that the camera focal length or its rough estimate is known. We initialize the camera translation (equivalently $\boldsymbol{\gamma}$) by assuming that the person is standing parallel to the image plane. Specifically, we estimate the depth via the ratio of similar triangles, defined by the torso length of the mean SMPL shape and the predicted 2D joints. Since this assumption is not always true, we further refine this estimate by minimizing E_J over the torso joints alone with respect to camera translation and body orientation; we keep $\boldsymbol{\beta}$ fixed to the mean shape during this optimization. We do not optimize focal length, since the problem is too unconstrained to optimize it together with translation.

After estimating camera translation, we fit our model by minimizing Eq. (3.1) in a staged approach. We observed that starting with a high value for λ_θ and λ_β and gradually decreasing them in the subsequent optimization stages is effective for avoiding local minima.

When the subject is captured in a side view, assessing in which direction the body is facing might be ambiguous. To address this, we try two initializations when the 2D distance between the CNN-estimated 2D shoulder joints is below a threshold: first with body orientation estimated as above and then with that orientation rotated by 180 degrees. Finally we pick the fit with lowest E_J.

We minimize Eq. (3.1) using Powell's dogleg method [155], using OpenDR and Chumpy [3, 141]. Optimization for a single image takes less than 1 minute on a common desktop machine.

3.3 Evaluation

We evaluate the accuracy of both 3D pose and 3D shape estimation. For quantitative evaluation of 3D pose, we use two publicly available datasets: HumanEva-I [187] and Human3.6M [102]. We compare our approach to three state-of-the-art methods [9, 173, 240] and also use these data for an ablation analysis. Both of the ground truth datasets have restricted laboratory environments and limited poses. Consequently, we perform a qualitative analysis on more challenging data from the Leeds Sports Dataset (LSP) [112]. Evaluating shape quantitatively is harder since there are few images with ground truth 3D shape. Therefore, we perform a quantitative evaluation using synthetic data to evaluate how well shape can be recovered from 2D joints corrupted by noise. For all experiments, we use 10 body shape coefficients. We tune the λ_i weights in Eq. (3.1) on the HumanEva training data and use these values for all experiments.

3.3.1 Quantitative Evaluation: Synthetic Data

We sample synthetic bodies from the SMPL shape and pose space and project their joints into the image with a known camera. We generate 1000 images for male shapes and 1000 for

female shapes, at 640×480 resolution.

In the first experiment, we add varying amounts of i.i.d. Gaussian noise (standard deviation (std) from 1 to 5 pixels) to each 2D joint. We solve for pose and shape by minimizing Eq. (3.1), setting the confidence weights for the joints in Eq. (3.2) to 1. Figure 3-4 (left) shows the mean vertex-to-vertex Euclidean error between the estimated and true shape in a canonical pose. Here we fit gender-specific models. The results of shape estimation are more accurate than simply guessing the average shape (red lines in the figure). This shows that joints carry information about body shape that is relatively robust to noise.

In the second experiment, we assume that the pose is known, and try to understand how many joints one needs to accurately estimate body shape. We fit SMPL to ground-truth 2D joints by minimizing Eq. (3.2) with respect to: the full set of 23 SMPL joints; the subset of 12 joints corresponding to torso and limbs (excluding head, spine, hands and feet); and the 4 joints of the torso. As above, we measure the mean Euclidean error between the estimated and true shape in a canonical pose. Results are shown in Figure 3-4 (right). The more joints we have, the better body shape is estimated. To our knowledge, this is the first demonstration of estimating 3D body shape from only 2D joints. Of course some joints may be difficult to estimate reliably; we evaluate on real data below.

3.3.2 Quantitative Evaluation: Real Data

HumanEva-I. We evaluate pose estimation accuracy on single frames from the HumanEva dataset [187]. Following the standard procedure, we evaluate on the Walking and Box sequences of subjects 1, 2, and 3 from the "validation" set [33, 204]. We assume the gender is known and apply the gender-specific SMPL models.

Many methods train sequence-specific pose priors for HumanEva; we do not do this. We

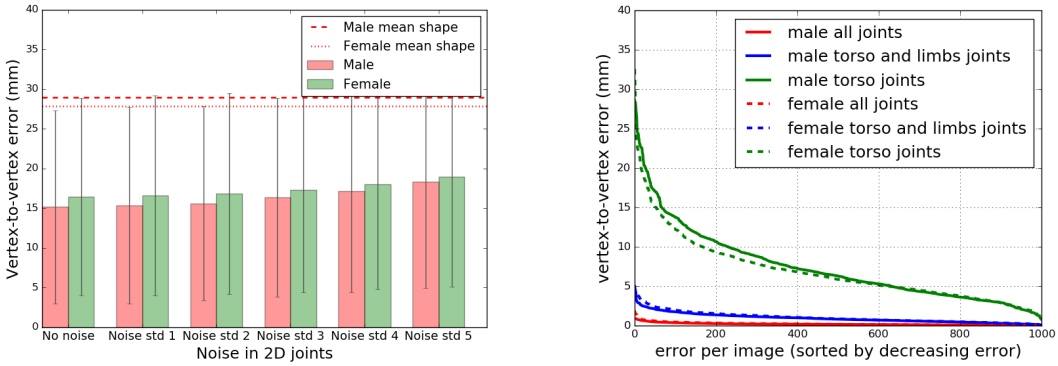


Figure 3-4: **Evaluation on synthetic data.** Left: Mean vertex-to-vertex Euclidean error between the estimated and true shape in a canonical pose, when Gaussian noise is added to 2D joints. Dashed and dotted lines represent the error obtained by guessing the mean shape for males and females, respectively. Right: Error between estimated and true shape when considering only a subset of joints during fitting.

do, however, tune our weights on HumanEva training set and learn a mapping from the SMPL joints to the 3D skeletal representation of HumanEva. To that end we fit the SMPL model to the raw mocap marker data in the training set using MoSh to estimate body shape and pose. We then train a linear regressor from body vertices (equivalently shape parameters $\boldsymbol{\beta}$) to the HumanEva 3D joints. This is done once on training data for all subjects together and kept fixed. We use the regressed 3D joints as our output for evaluation.

We compare our method against three state-of-the-art methods [9, 173, 240], which, like us, predict 3D pose from 2D joints. We report the average Euclidean distance between the ground-truth and predicted 3D joint positions. Before computing the error we apply a similarity transform to align the reconstructed 3D joints to a common frame via the Procrustes analysis on every frame. Input to all methods is the same: 2D joints detected by DeepCut [166]. Recall that DeepCut has not been trained on either dataset used for quantitative evaluation. Note that these approaches have different skeletal structures of 3D joints. We evaluate on the subset of 14 joints that semantically correspond across all representations. For this dataset we use the ground truth focal length.

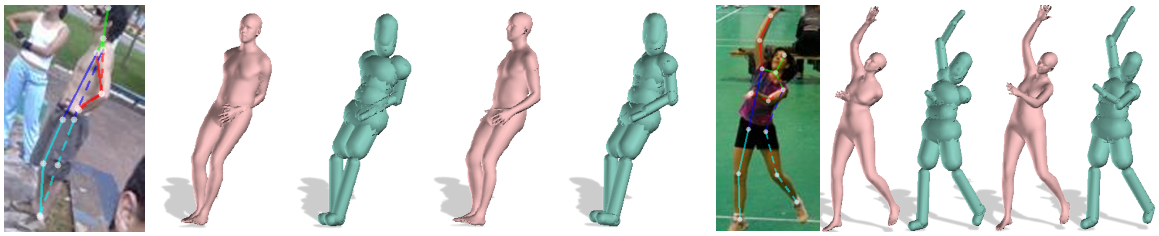


Figure 3-5: **Interpenetration error term.** Examples where the interpenetration term avoids unnatural poses. For each example we show, from left to right, CNN estimated joints, and the result of the optimization *without* and *with* interpenetration error term.

Table 3.1 shows quantitative results where SMPLify achieves the lowest errors on all sequences. While the recent method of Zhou et al. [240] is very good, we argue that our approach is conceptually simpler and more accurate. We simply fit the body model to the 2D data and let the model constrain the solution. Not only does this "lift" the 2D joints to 3D, but SMPLify also produces a skinned vertex-based model that can be immediately used in a variety of applications.

To gain insight about the method, we perform an ablation study (Table 3.2) where we evaluate different pose priors and the interpenetration penalty term. First we replace the mixture-model-based pose prior with $E_{\theta'}$, which uses a single Gaussian trained from the same data. This significantly degrades performance. Next we add the interpenetration term, but this does not have a significant impact on the 3D joint error. However, qualitatively, we find that it makes a difference in more complex datasets with varied poses and viewing angles as illustrated in Fig. 3-5.

	Walking			Boxing			Mean	Median
Method:	S1	S2	S3	S1	S2	S3		
Akhter & Black [9]	186.1	197.8	209.4	165.5	196.5	208.4	194.4	171.2
Ramakrishna et al. [173]	161.8	182.0	188.6	151.0	170.4	158.3	168.4	145.9
Zhou et al. [240]	100.0	98.89	123.1	112.5	118.6	110.0	110.0	98.9
SMPLify	**73.3**	**59.0**	**99.4**	**82.1**	**79.2**	**87.2**	**79.9**	**61.9**

Table 3.1: **HumanEva-I results.** 3D joint errors in mm.

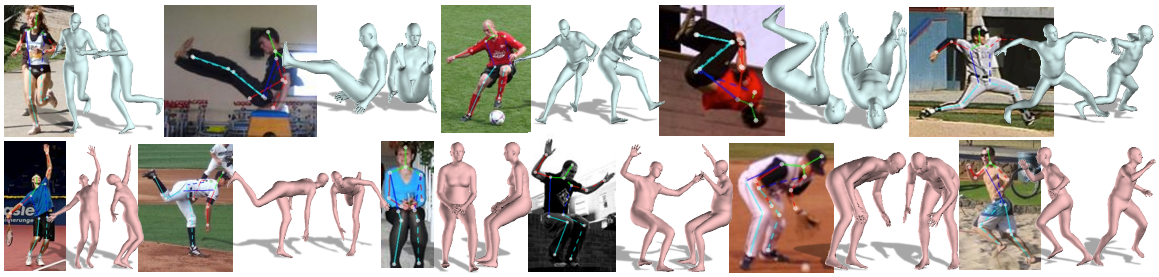


Figure 3-6: **Leeds Sports Dataset.** Each sub-image shows the original image with the 2D joints fit by the CNN. To the right of that is our estimated 3D pose and shape and the model seen from another view. The top row shows examples using the gender-neutral body model; the bottom row show fits using the gender-specific models.

Human3.6M. We perform the same analysis on the Human 3.6M dataset [102], which has a wider range of poses. Following [134, 204, 241], we report results on sequences of subjects S9 and S11. We evaluate on all 15 action sequences captured from the frontal camera ("cam3") from trial 1. These sequences consist of 2000 frames on average and we evaluate on all frames *individually*. As above, we use training mocap and MoSh to train a regressor from the SMPL body shape to the 3D joint representation used in the dataset. Other than this we do not use the training set in any manner. We assume that the focal length as well as the distortion coefficients are known since the subjects are closer to the borders of the image. Evaluation on Human3.6M is shown in Table 3.3 where our method again achieves the lowest average 3D error. While not directly comparable, Ionescu et al. [101] report an error of 92mm on this dataset.

	Walking			Boxing			Mean	Median
Method:	S1	S2	S3	S1	S2	S3		
$E_\beta + E_J + E_{\theta'}$	98.4	79.6	117.8	105.9	98.5	122.5	104.1	82.3
$E_\beta + E_J + E_{\theta'} + E_{sp}$	97.9	79.4	116.0	105.8	98.5	122.3	103.7	82.3
SMPLify	**73.3**	**59.0**	**99.4**	**82.1**	**79.2**	**87.2**	**79.9**	**61.9**

Table 3.2: **HumanEva-I ablation study.** 3D joint errors in mm. The first row drops the interpenetration term and replaces the pose prior with a uni-modal prior. The second row keeps the uni-modal pose prior but adds the interpenetration penalty. The third row shows the proposed SMPLify model.

	Directions	Discussion	Eating	Greeting	Phoning	Photo	Posing	Purchases	Sit
Akhter & Black [9]	199.2	177.6	161.8	197.8	176.2	186.5	195.4	167.3	160.7
Ramakrishna et al. [173]	137.4	149.3	141.6	154.3	157.7	158.9	141.8	158.1	168.6
Zhou et al. [240]	99.7	95.8	87.9	116.8	108.3	107.3	93.5	95.3	109.1
SMPLify	**62.0**	**60.2**	**67.8**	**76.5**	**92.1**	**77.0**	**73.0**	**75.3**	**100.3**
	SitDown	Smoking	Waiting	WalkDog	Walk	WalkTogether		Mean	Median
Akhter & Black [9]	173.7	177.8	181.9	176.2	198.6	192.7		181.1	158.1
Ramakrishna et al. [173]	175.6	160.4	161.7	150.0	174.8	150.2		157.3	136.8
Zhou et al. [240]	137.5	106.0	102.2	106.5	110.4	115.2		106.7	90.0
SMPLify	**137.3**	**83.4**	**77.3**	**79.7**	**86.8**	**81.7**		**82.3**	**69.3**

Table 3.3: **Human 3.6M.** 3D joint errors in mm.

3.3.3 Qualitative Evaluation

Here we apply SMPLify to images from the Leeds Sports Pose (LSP) dataset [112]. These are much more complex in terms of pose, image resolution, clothing, illumination, and background than HumanEva or Human3.6M. The CNN, however, still does a good job of estimating the 2D poses. We only show results on the LSP test set. Figure 3-6 shows several representative examples where the system works well. The figure shows results with both gender-neutral and gender-specific SMPL models; the choice has little visual effect on pose. For the gender-specific models, we manually label the images according to gender.

Figure 3-8 visually compares the results of the different methods on a few images from each of the datasets. The other methods suffer from not having a strong model of how limb lengths are correlated. LSP contains complex poses and these often show the value of the interpenetration term. Figure 3-5 shows two illustrative examples. Figure 3-7 shows a few failure cases on LSP. Some of these result from CNN failures where limbs are mis-detected or are matched with those of other people. Other failures are due to challenging depth ambiguities. See our website [1] for more results.

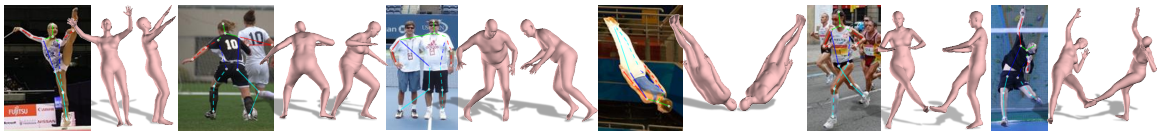


Figure 3-7: **LSP Failure cases.** Some representative failure cases: misplaced limbs, limbs matched with the limbs of other people, depth ambiguities.

Figure 3-8: **Qualitative comparison.** From top to bottom: Input image. Akhter & Black [9]. Ramakrishna et al. [173]. Zhou et al. [240]. SMPLify.

3.4 Conclusions

In this chapter we have presented SMPLify, a fully automated method for estimating 3D body shape and pose from 2D joints in single images. SMPLify uses a CNN to estimate 2D joint locations, and then fits a 3D human body model to these joints. We use the recently proposed SMPL body model [143], which captures correlations in body shape, highly constraining the fitting process. We exploit this to define an objective function and optimize pose and shape directly by minimizing the error between the projected joints of the model and the estimated 2D joints. This gives a simple, yet very effective, solution to estimate 3D pose and approximate shape. The resulting model can be immediately posed and animated. We extensively evaluate our method on various datasets and find that SMPLify outperforms state-of-the-art methods.

Our formulation opens many directions for future work. In particular, body shape and

pose can benefit from other cues such as silhouettes, and in fact in the next chapter we use silhouettes to improve pose and shape estimation of quadruple animals. Our formulation can easily benefit from multiple camera views and multiple frames. Additionally a facial pose detector would improve head pose estimation and automatic gender detection would allow the use of the appropriate gender-specific model. It would be useful to train CNNs to predict more than 2D joints, such as features related directly to 3D shape. Our method provides approximate 3D meshes in correspondence with images, which could be useful for such training. The method can be also be extended to deal with multiple people in an image; having 3D meshes should help with reasoning about occlusion.

Our results demostrate that a model-based fitting approach for single-view 3D reconstruction is an effective solution even for highly articulated objects. In the following chapters we discuss how a similar approach can be extended to animals.

Chapter 4

Learning 2D Deformation Field of Birds

4.1 Introduction

In this chapter we focus on the bottom-up estimation problem for animals. In particular, we focus on how the semantic correspondence problem may be solved without using any human provided keypoint annotations. Instead of a model-based approach, here we follow the recent "3D by image collection" approach discussed in 2.3.3, where 3D is obtained by matching images of objects in a large image collection. However, different object instances exhibit large appearance and shape variations, which cannot be handled by traditional appearance features such as SIFT [147] alone. Thus, prior works rely on supervision in the form of keypoint annotations [48, 117, 214] or 3D CAD models in the case of rigid objects [23, 50] to augment appearance information with shape priors. Such annotations are labor-intensive,

The contents of this work is in collaboration with Manmohan Chandraker and David Jacobs, presented at CVPR 2016 [115]. This was part of my internship at NEC Labs America, also supported by the National Science Foundation under Grant No. 1526234.

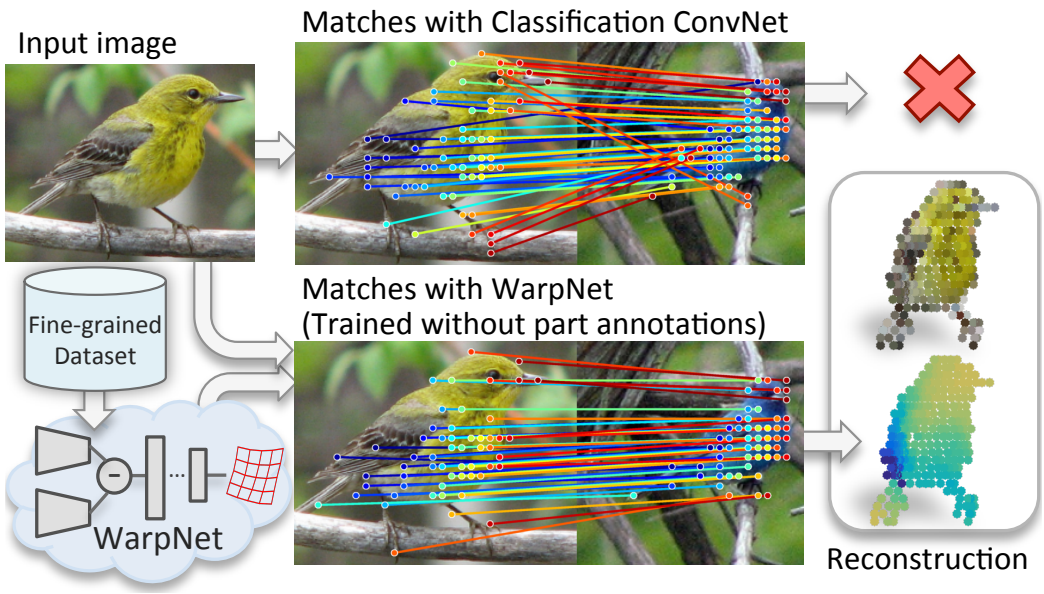


Figure 4-1: **Idea.** We propose a novel deep learning architecture, WarpNet, for learning an image-specific spatial prior for matching two images of different object instances in a fine-grained dataset. WarpNet is trained without using any human-provided part annotations, but significantly improves matching accuracy across variations in appearance, pose and articulation (bottom), which is not possible with appearance features alone (top). Our match quality is high enough to be propagated across images to be used for single-view reconstruction without using any manually annotated keypoints (right).

thus, too sparse for reconstruction and not scalable. Further, it can be quite difficult and impractical to obtain human-labeled annotations for parts that are not nameable. In contrast, this chapter presents a framework to match images of *fine-grained datasets* such as birds, with some degree of non-rigidity and articulation, across sub-category and pose variations, without requiring supervised keypoint annotations. We then present an approach to the challenging novel problem of weakly-supervised single-view object reconstruction.

We postulate that the structure of fine-grained datasets, combined with the power of convolutional neural networks (CNNs), allows matching instances of different sub-categories without human keypoint annotation. Fine-grained datasets for objects such as birds can be analyzed along two dimensions – appearance and global-shape. Instances within the same sub-category that are imaged in different poses can be matched by appearance similarity,

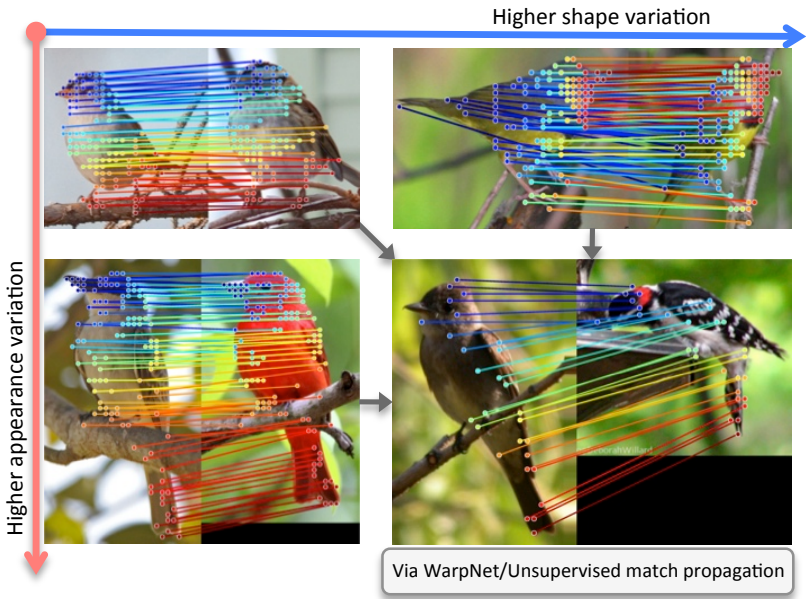


Figure 4-2: **Intuition:** Matching within a category exploits appearance similarity, while matching instances across related categories is possible through global shape similarity. By propagation, one may match across variations in both appearance and shape.

while instances with similar pose or viewpoint from different categories can be matched through similarity in global shape. Instances with both appearance and shape variations may then be matched by propagation (Fig. 4-2). In other words, because sub-categories share a common shape, matches that are difficult via appearance alone can be overcome by using their similarity in global shape. In Section 4.2, we demonstrate a practical realization of this intuition by introducing a deep learning architecture, *WarpNet* that learns a space of 2D deformation fields of fine-grained categories by taking advantage of their similarity in shape. Specifically, WarpNet learns to warp points on one object into corresponding points on another (from a possibly different category or pose) without requiring human keypoint annotations. The predicted deformation between two objects acts as an image-specific shape prior, which encourages matches that are consistent with the shape of the two objects.

WarpNet is a Siamese network that accepts two images as input (Section 4.2.2). To overcome the absence of human annotated keypoints, our training presents an image and a

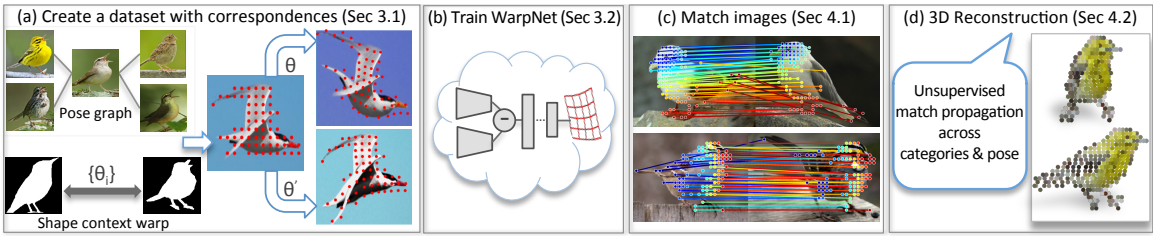


Figure 4-3: **Overview of our framework.** (a) Lacking part annotations, we exploit the fine-grained dataset to create artificial correspondences. (b) These are used to train our novel deep learning architecture that learns to warp one object into another. (c) The output of the network is used as a spatial prior to match across appearance and shape variations. (d) Our high-quality matches can be propagated across the dataset. We use the WarpNet output and the structure of fine-grained categories to perform single-view reconstruction without part annotations.

warped version related by a known thin-plate spline (TPS) transformation, which yields artificial correspondences. We assume the object bounding box and foreground segmentation are known, which can be obtained through state-of-the-art segmentation [53] or co-segmentation methods [126]. We experiment using both ground truth and co-segmentation outputs. In Section 4.2.1, we exploit neighborhood relationships within the dataset through the pose graph of Krause *et al.* [126] to compute exemplar TPS transformations between silhouettes, from which our artificial transformations are sampled. A point transformer layer inspired by [105] is used to compute the warp that aligns keypoints without supervision, which provides a spatial prior for matching (Section 4.3). We show that WarpNet generalizes well to match real images with distinct shapes and appearances at test time. In particular, it achieves matching accuracy over 13.6% higher than a baseline ILSVRC CNN [52].

Establishing matches between a given instance and other objects in the dataset opens the door to a novel problem – weakly supervised reconstruction in fine-grained datasets. Several sub-problems must be solved to achieve this goal, such as match propagation and image subset selection. Prior works such as [48, 214] approach these sub-problems, but the absence of supervised annotations poses new challenges. In Section 4.3.2, we suggest ways to

overcome them through the use of matches from our WarpNet, the pose graph and heuristics that exploit the structure of fine-grained datasets. We demonstrate reconstructions that are nearly as good as those obtained using supervised annotations and better than those from appearance-only CNNs or unsupervised baselines such as deformable spatial pyramids [123].

To summarize, our key contributions are:

- A novel deep learning architecture, WarpNet, that predicts a warp for establishing correspondences between two input images across category and pose variations.
- A novel exemplar-driven mechanism to train WarpNet without requiring supervised keypoint annotations.
- An approach to weakly-supervised single-view object reconstruction that exploits the structure of the fine-grained dataset to yield reconstructions of birds nearly on par with the method that uses supervised part annotations.

4.2 Learning without Part Annotations

We present a deep learning framework, *WarpNet*, that learns the correspondence from one image to another without requiring part annotations. Given two images I_1 and I_2, our network outputs a function that takes points in I_1 to points in I_2. We parameterize this function as a thin-plate spline (TPS) transformation since it can capture shape deformations well [29]. Inspired by Dosovitskiy *et al.* [6], we generate artificial correspondences by applying known transformations to an image. However, our approach is distinct in using the structure afforded by fine-grained datasets and dealing with non-rigidity and articulations. Our network generalizes well to instances of different categories at test time and we use its output as a spatial prior in computing a match between two objects. Figure 4-3 gives an overview of our

approach. We discuss each step in detail below.

4.2.1 Generating Unsupervised Correspondences

Since we do not have annotated point correspondences, we create artificial ones by applying random spatial and chromatic transformations to images. The key requirement is that the spatial transformations applied are complex enough to learn meaningful correspondences, while producing transformed images that are reflective of actual image pairs to match at test time. For instance, affine transformations are not expressive enough to capture non-rigid deformations and articulations in birds. Instead, we use TPS transformations and exploit the fine-grained dataset to generate exemplar warps that span a realistic range of transformations.

We use the pose graph of Krause *et al*. [126], whose edge weights are determined by the cosine distance of the fourth-layer of a pre-trained ILSVRC CNN, which captures abstract concepts such as class-independent shape. We compute shape context TPS warps [29] between the silhouettes of images that are within 3 nearest-neighbors apart on the pose graph. We sort the TPS warps using the mean of their bending and affine energy, retaining only those between the 50th and 90th percentiles to avoid warps that are too trivial or too drastic. We create m transformed versions of every image by sampling from this set of TPS warps. We sample n points uniformly on the foreground, which we use as correspondences. Figure 4-4 shows the effect of transformations sampled from the exemplar-TPS warps. The images on the left are the originals and the ones on the right are transformed versions. Notice how the transformation induces changes in shape and articulations around the head and the tail, which validates the utility of our exemplar TPS warps.

Figure 4-4: Sample exemplar-TPS warped images used for training our WarpNet. Left: original images, right: artificial versions made by applying exemplar TPS warp + chromatic transformation. Notice changes in shape and articulations at the head and the tail.

4.2.2 WarpNet Architecture

Our proposed WarpNet is a Siamese network [57] that takes two images related by an exemplar TPS transformation, I_1 and I_2, along with the corresponding n keypoint locations, as inputs during training (at test time, the input consists only of two images from possibly different categories and poses that must be matched). The main objective of WarpNet is to compute a function that warps points $\boldsymbol{p}_2$ in I_2 to image coordinates in I_1, such that after warping the L2 distance to the corresponding points $\boldsymbol{p}_1$ in I_1 is minimized. Figure 4-5 illustrates the architecture of WarpNet.

First, the input images are passed through convolution layers with tied weights. The extracted features are then combined by element-wise subtraction of the feature maps. We subtract rather than concatenate the feature maps along the channels, since concatenation significantly increases the number of parameters in the network making it unstable to train. The combined feature maps are passed through a point transformer, similar to [105], which

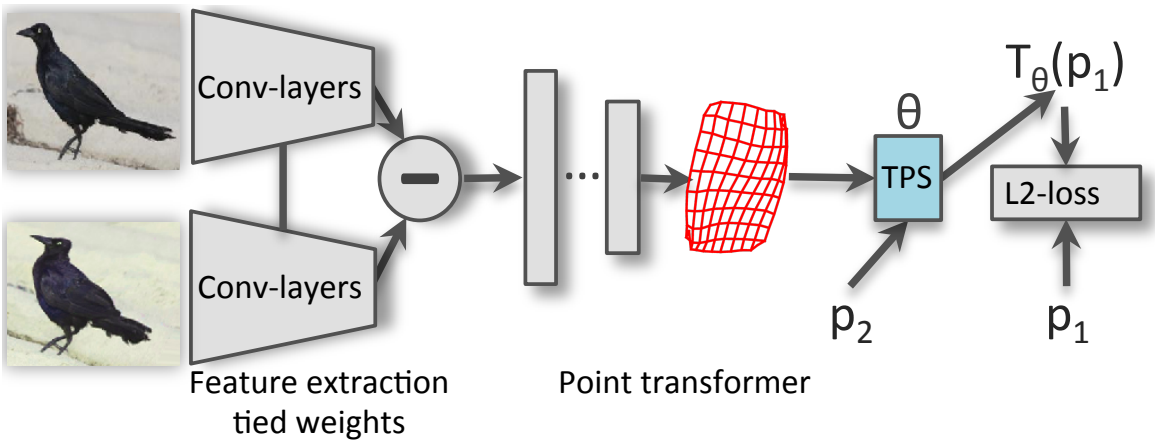


Figure 4-5: WarpNet architecture. Visual features are extracted from two input images using a Siamese CNN. They are combined to predict a deformed grid that parameterizes a TPS transformation. The network objective is to minimize the distance between corresponding points $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ of the image pair after applying the predicted transformation to $\boldsymbol{p}_2$.

regresses on the (x, y) coordinates of a deformed $K \times K$ grid. The output grid, normalized to a range of $[-1, 1] \times [-1, 1]$, acts as the control points for computing a grid-based TPS transformation from I_2 to I_1. This involves solving a system of linear equations, handled by the TPS layer. See Appendix A for details. The predicted TPS transformation is applied to the keypoints of I_2 generating the transformed version $T_\theta(\boldsymbol{p}_2)$, which finally gets sent to the L2 loss layer along with $\boldsymbol{p}_1$. Since every step consist of linear operations, the whole network can be trained with backpropagation.

We implicitly train the warp parameters in terms of distance between corresponding points rather than direct supervision against the TPS warp coefficients. This provides a natural distance between warps, where we can train the network without knowing the exact transformation parameters used.

Figure 4-6 illustrates the output of the trained network given two real images as input, denoted source and target. Despite the fact that the network has never seen objects of different instances, it is able to compute warps between the two objects. Note that WarpNet accounts for variations in shape (fat to skinny, small to large birds), articulation (such as the

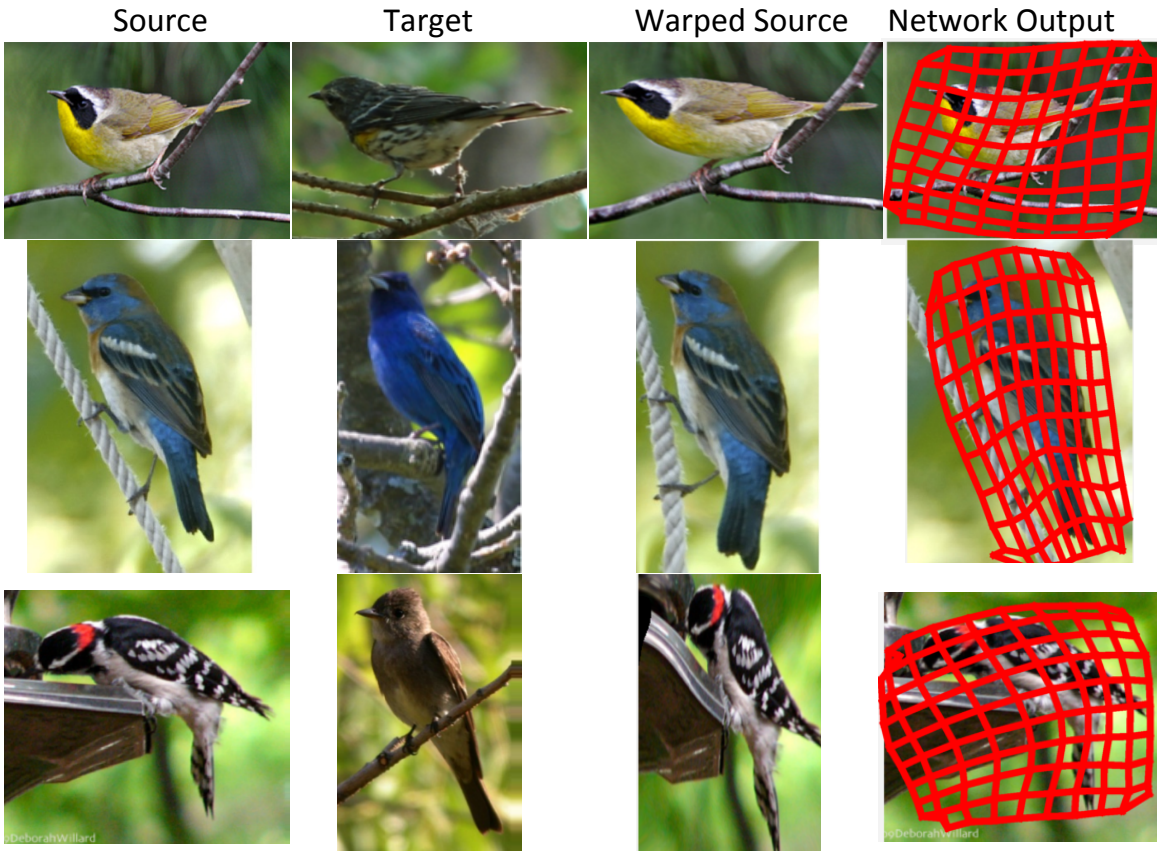


Figure 4-6: Visualizations of the network output. WarpNet takes two images, source and target, as inputs and produces a 10x10 deformed lattice (last column) that defines a TPS warp from target to source. The third column shows the warped source image according to the network output. Notice how the network accounts for articulations at the tail and the head as well as differences in shape of the birds. WarpNet is trained in an unsupervised manner and none of these images were seen by the network during training.

orientation of the head or the tail) and appearance.

4.3 Matching and Reconstruction

4.3.1 Matching with WarpNet

Given two images I_i and I_j, a match for a point u_i in I_i is the most similar point v_j in I_j using the similarity score consisting of an appearance term and a spatial term:

$$s(u_i, v_j) = \exp\left(\frac{-d_f(u_i, v_j)}{\sigma_f}\right) + \lambda \exp\left(\frac{-d_w(u_i, v_j)}{\sigma_w}\right), \tag{4.1}$$

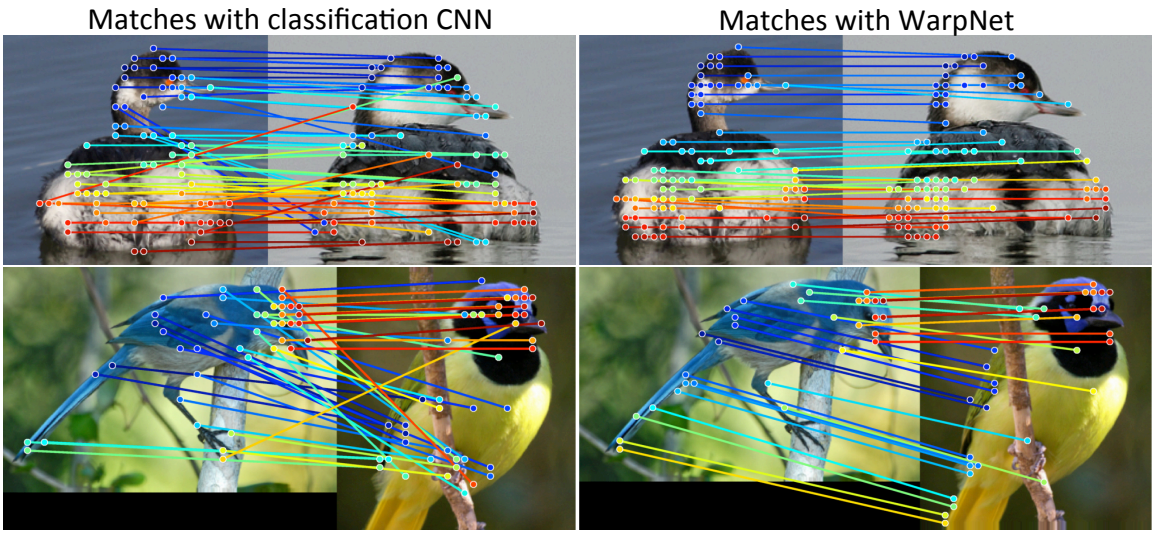


Figure 4-7: Sample matches obtained by ILSVRC trained CNN *vs*. WarpNet. Note WarpNet's relative robustness to variations in appearance, pose and articulation.

where $d_f(u, v)$ is the L2 distance of appearance features extracted at u_i and v_j, while d_w is a symmetric spatial prior:

$$d_w(u, v) = \frac{1}{2}(||\boldsymbol{x}_i^u - T_{\theta_{ij}}(\boldsymbol{x}_j^v)|| + ||\boldsymbol{x}_j^v - T_{\theta_{ji}}(\boldsymbol{x}_i^u)||). \tag{4.2}$$

We use WarpNet to compute $T_{\theta_{\cdot,\cdot}}$ in both directions.

The matches are then ranked by the ratio-test strategy [147]. This simple but powerful heuristic allows discarding points in I_i that are similar to many other points in I_j. Since the keypoints are extracted densely on the foreground, we compute the similarity score ratio between the first and second nearest neighbors that are at least 10 pixels away. Figure 4-7 shows a few qualitative matching results comparing the baseline CNN and WarpNet.

4.3.2 Single-View Object Reconstruction

Obtaining good matches is a critical first step towards 3D reconstruction. While single-view 3D reconstruction methods in the past have relied on expensive supervised inputs such as part annotations or CAD models, our matching enables a first approach towards a challenging

new problem, namely, part annotation free single-view reconstruction. We discuss initial approaches to variants of existing supervised methods or structure from motion (SFM) pipelines that may be used to solve this problem without requiring annotations.

Propagating correspondences In the CUB-200-2011 dataset, there are only 60 images for each category. Moreover, birds are often imaged from preferred viewpoints, but it is critical for reconstruction to obtain matches across a well-distributed set of viewpoints. On the other hand, deformations may be very high even within a category (open wings as opposed to closed), which makes straightforward matching within a category challenging. Inspired by the work of Carreira *et al*. [48], we use a shortest path method to propagate matches across objects of similar shapes in the dataset, in order to obtain a denser set of tracks. However, note that we lack the initial set of point annotations as well as the camera poses obtained through part annotations in [48, 214], who also manually select a subset of keypoints to eliminate articulations. Instead, we determine unsupervised matches purely through our WarpNet and rely on the pose graph to determine nearest neighbors for propagation.

Choosing a subset for reconstruction A key problem we encounter is the choice of images for reconstruction. In previous works on reconstruction within PASCAL VOC [48, 214], it has been possible to use the entire dataset since it contains less than 1000 images for birds. In contrast, CUB-200-2011 contains nearly 12000 images, which poses computational challenges and requires greater vigilance against outliers. Moreover, annotations in [48, 214] preclude the need for algorithmic considerations on baseline or shape variations in choosing the image set. For instance, to reconstruct a sitting bird imaged from a frontal view, we must propagate matches to side views of sitting birds in other categories to ensure a good baseline, while avoiding images of flying birds.

Given a collection of images, several heuristics have been proposed for selecting the right subset or order for multiview rigid-body reconstruction [193, 194]. However, those are not directly applicable for single-view reconstruction of deformable objects. Instead, we propose three heuristics that utilize the structure of fine-grained bird datasets:

- Use images from categories that share a keyword (for example, all "warblers", or all "sparrows").
- Use images from categories that are related by an ornithological taxonomy, as defined by [157].
- Use images from the five nearest neighbor subcategories on a similarity tree of bird species [30].

The above heuristics perform comparably and address the same goal – introduction of matched keypoints from more than one subcategory to ensure good viewpoint coverage.

Reconstruction Given an image of a target object from one particular class, we consider images from several other categories using one of the above heuristics. We compute pairwise matches at 85% precision threshold between all pairs of images whose distance on the pose graph is less than 4. We ignore pairs that have less than 50 surviving matches. We then set up a virtual view network (VVN) [48] to propagate matches across all the selected images by solving a shortest path problem. We use scores from (4.1), bounded between $[0, 1]$, as weights on the graphs connecting the keypoints. After propagation, we discard as spurious any propagated matches with shortest path distance more than 0.4 and remove all images that have less than 30 matches with the target object. We then create the measurement matrix of tracked keypoints of the target object. We only consider keypoints visible in at least 10% of the images as stable enough for reconstruction. We finally send the observation

matrix to the rigid factorization method of [150], which robustly handles missing data, to obtain 3D shape. A rigid factorization suffices to produce reasonable reconstructions since the dataset is large enough, but non-rigid methods alternately could be used.

4.4 Experiments

We perform experiments on the CUB-200-2011 dataset which contains 11788 images of 200 bird categories, with 15 parts annotated [217]. We reconstruct without part annotation, assuming objects are localized within a bounding box. We quantitatively evaluate our matches using and extending the part annotations. Next, we evaluate the effectiveness of WarpNet as a spatial prior and analyze the choice of transformations for creating the artificial training dataset. Finally, we demonstrate the efficacy of our framework with several examples of unsupervised single-view reconstruction.

4.4.1 Experimental Details

We create the pose graph of [126] using the `conv4` feature of AlexNet trained on ILSVRC2012 [127]. For creating the artificial dataset, we only use the training data (∼6000 images) and create $m = 9$ copies of each image using our exemplar-TPS. This results in approximately 120k image pairs, each with $n = 100$ point correspondences.

Network training details We use the VGG-M architecture of [52] until the `pool5` layer as the feature extraction component of WarpNet. The point transformer consists of C512-C256-F1024-D-Op using the notation of [8]. Both convolutional layers use 3x3 kernel, stride 1 with no padding, with ReLU non-linearity. The output layer is a regressor on the grid coordinates, with grid size $K = 10$. The feature extraction weights are initialized with

weights pre-trained on the ILSVRC classification task, following prior state-of-the-art for correspondence [140].

All training images are cropped around the bounding box padded with 32 pixels and resized to 224×224 with pixel-wise mean subtraction computed using the CUB-200-2011 dataset. These images were further augmentated on the fly with these spatial and chromatic parameters:

- mirroring (consistent mirror for image pairs)
- scaling between $[0.8, 1.2]$
- vertical or horizontal translation by a factor within 3% of image size
- rotation within $[-20, 20]$ degrees;
- contrast 1 of [6] with factors within $[0.5, 2]$
- contrast 2 of [6] with saturation multiplication factors within $[0.7, 1.4]$, saturation or hue addition within $[-0.1, 0.1]$ but with no power saturation.

The feature extraction layer weights (up to `pool5`) are initialized with the VGG_M_1024 model of [52]. The learning rates on the pre-trained weights are set to one-tenth of the global learning rate. All other weights are initialized from a Gaussian distribution with a zero mean and variance equal to 0.1.

We train the network with momentum 0.9 and weight decay of 10^{-5}. We tune the weight decay and the learning rate following the feature extraction using a held out set of artificial datasets. The learning rates of the pre-trained feature extraction layers is set to 0.01 of the global learning rate, which is set to a constant value of 0.0001. We train the network with mini-batch size 64, for 45k iterations, when the test error begins to converge.

For the point-transformer architecture (after combining `pool5` features), we experiment

using several fully-connected layers instead of starting with convolution layers. However, starting with convolution layers is clearly the better choice since it yields the lowest test errors while keeping the number of parameters reasonable. We did not further fine-tune the architecture of the point-transformer such as tuning the number of feature maps, the kernel size, stride, or the number of convolution layers.

Matching and Reconstruction For matching and reconstruction, images are resized with aspect ratio intact and the smallest side 224 pixels. We uniformly sample points on the foreground with a stride of 8 as keypoints for matching. For all experiments we use L2-normalized `conv4` features extracted at the keypoints using the hole algorithm [53] for computing the appearance term in (4.1). Hyperparameters used for matching are $\sigma_f = 1.75$, $\sigma_w = 18$, $\lambda = 0.3$, tuned using the artificial dataset.

4.4.2 Match Evaluation

We compare our approach with ILSVRC pre-trained VGG-M `conv4` [52], SIFT at radius 8 [147] and matches from the deformable spatial pyramid (DSP) [123]. Only the appearance term in (4.1) is used for computing matches with VGG-M `conv4` and SIFT. For computing the matches with DSP, we mask out the background prior to extracting SIFT features following [48] and only keep matches of the keypoints. For this experiment, the set of keypoints to match includes the locations of annotated parts.

In order to evaluate WarpNet as a stand-alone learned spatial prior, we compare WarpNet with DSP by replacing the SIFT features in DSP with VGG features. We call this method *VGG+DSP*. We further evaluate WarpNet against the original DSP by using WarpNet as a spatial prior for SIFT matches, where the unary term d_f in (4.1) is computed with SIFT features. We call this method *SIFT+WarpNet*.

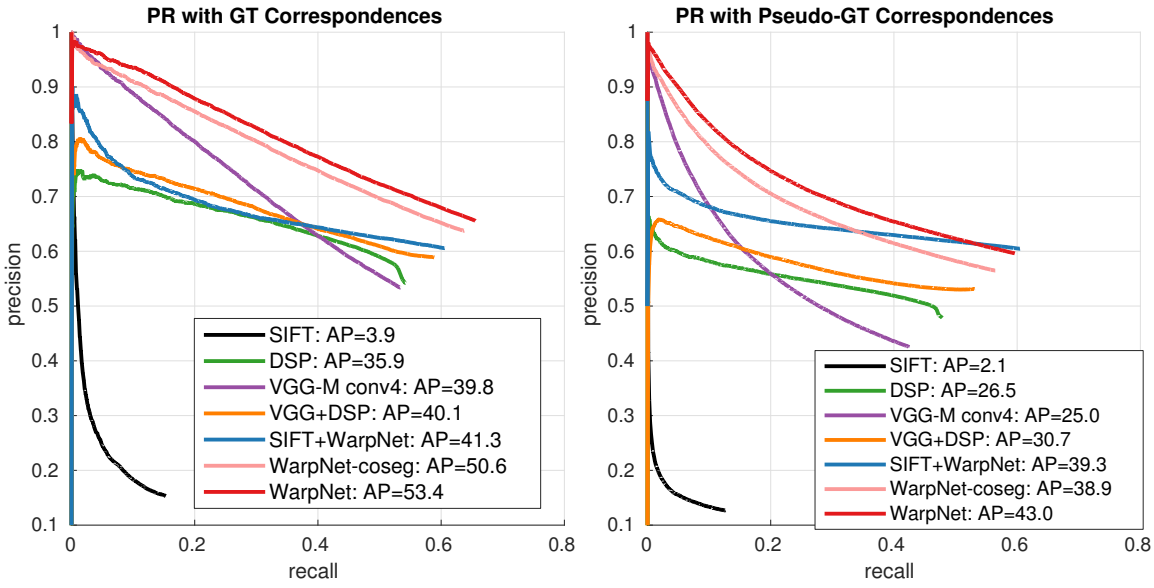
As discussed in Section 4.2.1, the only supervision required in training WarpNet is the segmentation mask to mine exemplar-TPS transformations. We also evaluate the robustness of WarpNet using co-segmentation outputs of [126], called *VGG+coseg*.

Test set We evaluate on 5000 image pairs that are within 3 nearest neighbors apart on the pose graph, comprising more than 50k ground truth matches. [1] Due to the unsupervised nature of the pose graph, these pairs exhibit significant articulation, viewpoint and appearance variations (see Figures 4-1, 4-6). We remove severely occluded pairs with less than 7 parts visible in both images and pairs whose TPS warp computed from part annotations have very high bending energy. None of the test images were used to train WarpNet.

Evaluation metrics We evaluate the accuracy of matches with the percentage of correct keypoints (PCK) metric [226], where a match is considered correct if the predicted point is within $\alpha * L$ of the ground-truth correspondence. Following [8], we chose L to be the mean diagonal length of the two images. We also compute the precision-recall (PR) curve adopting the procedure of [152]. A match is considered a true positive within a radius $\alpha = 0.05$, otherwise it is a false positive. In this setup, a recall of 1 is obtained only if all the matches retrieved are correct, that is, 100% α-PCK. We compute PR curves using the ratio-test values described in Section 4.3.1 for ranking the matches and report AP. For DSP, we use its matching cost for ranking instead of the ratios, since second closest matches are not available.

Results Figure 4-8(a) shows the obtained PR curves. WarpNet achieves an AP of 53.4%, an 13.6% increase over matches using just the appearance feature of VGG-M `conv4`. WarpNet achieves a much higher recall due to its spatial prior, learned without using any part annotations. As a side note, `conv4` features of WarpNet alone achieve very similar performance

[1] Please see supplementary materials for results on a test set with 1-nearest neighbors, where we observe similar trends but with higher PCKs.

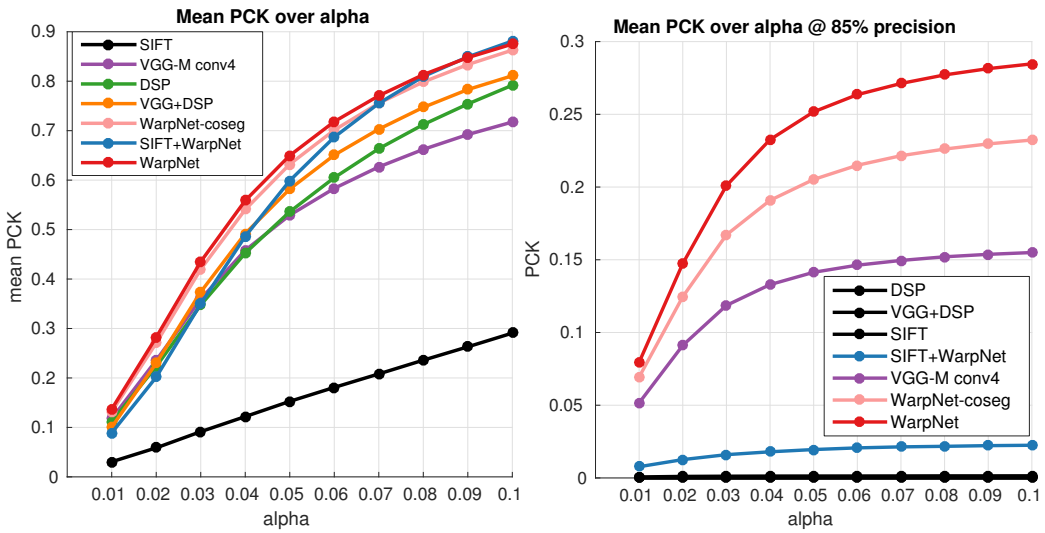


(a) (b)

Figure 4-8: Precision-Recall curves for matching points between neighboring images on the pose graph. We evaluate points with (a) human-annotated correspondences and (b) expanded pseudo-ground-truth correspondences.

to the VGG-M `conv4`. In all cases, WarpNet outperforms DSP as a spatial prior and changing SIFT to VGG features yields around 5% improvement in the final recall. *WarpNet-coseg* still outperforms the baseline VGG-M by 10.8%, showing our approach is applicable even without ground truth segmentations.

Figure 4-9(a) shows the PCK as a function of α, where WarpNet consistently outperforms other methods. We observe that VGG-M `conv4` and DSP perform similarly, showing that while DSP obtains low recall at high precision, its overall match quality is similar to CNN features, an observation in line with [48]. Since only high precision matches are useful for reconstruction where outliers need to be avoided, we show the same curves thresholded at 85% precision in Figure 4-9(b) for VGG-M and our method. Note that some methods in black have zero recall at this precision. The growing gap between WarpNet and VGG-M `conv4` as α increases suggests that, unlike WarpNet, appearance features alone make grossly wrong matches (see Figures 4-1 and 4-7).



(a) (b)

Figure 4-9: PCK (higher the better) over varying definition of correctness α. (a) Mean PCK of all retrieved matches regardless of ratio score. (b) Mean PCK with matches thresholded at 85% precision, which are the matches used for reconstruction.

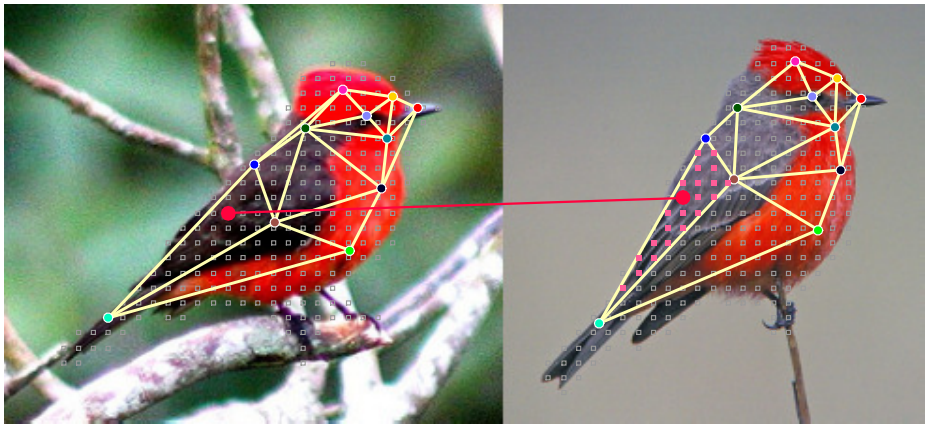
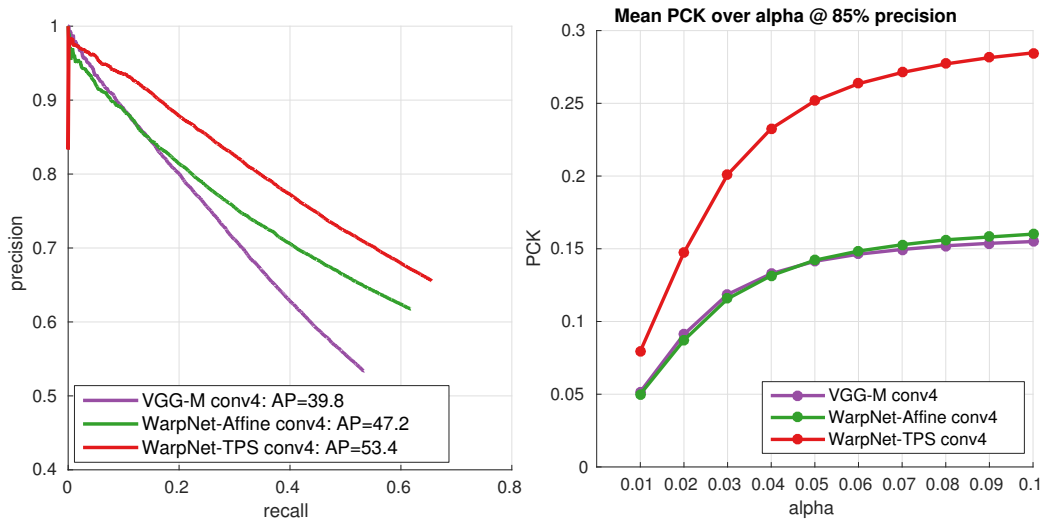


Figure 4-10:]
Illustration of the pseudo-gt correspondences. We triangulate each image using the annotated keypoints (colored points). The match for the big red dot in the left image is found by looking at points within the same triangle (small pink dots) in the right image and picking the closest point in terms of barycentric coordinates.

Expanding the set of part annotations A caveat of the CUB-200-2011 for our task is that part annotations are sparse and concentrated on semantically distinct parts such as eyes and beaks around the head region, with only four points on the bird body that are often not all visible. To investigate matching performance more densely, we carefully expand the ground-truth matches using the annotated parts. This process is illustrated in Figure 4-10.



(a) PR (b) PCK@85%

Figure 4-11: Comparing results for WarpNet trained on artificial data created using affine-spatial transformations with (a) PR curves and (b) PCK over α. WarpNet trained with exemplar-TPS is more effective in terms of recall and precision.

Given a pair of images I_1 and I_2, we Delaunay triangulate each image independently using the parts visible in both as vertices. For a point u within a triangle in I_1, we consider points in I_2 that are within the *same* triangle as possible candidates (shown as pink dots in Figure 4-10), find the point that is closest to u in terms of barycentric coordinates and accept this as a new pseudo ground-truth match if the distance is less than 0.1. Figure 4-8(b) shows the PR curve obtained using the pseudo-ground truth matches (in addition to the annotated parts). We see the same trends as Figure 4-8(a), but with a wider gap between the baselines and our method. This is reasonable given that bird bodies usually consist of flat or repeated textures that are challenging to match with local appearances alone, highlighting the efficacy of WarpNet's spatial prior.

4.4.3 Choice of Transformations

We now analyze the choice of exemplar TPS transformations for creating the artificial dataset. We train another WarpNet under the same settings, but on an artificial dataset created using only affine spatial transformations, which we refer to as AffineNet. Note that AffineNet's

output is still a TPS transformation, thus, it has the same capacity as the original WarpNet. Figure 4-11(a) shows the PR curve of AffineNet in comparison to WarpNet and VGG-M `conv4`. WarpNet outperforms AffineNet in all aspects. While AffineNet has a higher final recall (that is PCK of all matches) than VGG-M `conv4`, its recall at high precision is slightly lower than that of VGG-M `conv4`. This is highlighted in Figure 4-11(b), which shows PCK of matches at 85% precision over α, where AffineNet performs on par with VGG-M `conv4`. This indicates that the warps predicted by AffineNet are helpful in a general sense, but not precise enough to improve the recall at high precision. This experiment shows that using exemplar-TPS transformations for creating the artificial dataset is critical for training a useful WarpNet.

4.4.4 Single-view Object Reconstruction

We compare our method with three other matching methods. One is a supervised matching approach similar to [48], where the network predicted TPS warp T_θ in (4.2) is replaced by the supervised TPS warp computed using the annotated keypoints. We call this approach `supervised` and it is an upper-bound to our method since ground-truth part annotations are used for reconstruction. We also perform reconstructions with VGG-M `conv4` features alone and DSP. We do not include the mirrored image as another viewpoint of the target object, since bilateral symmetry does not hold for articulated objects. For post-processing we use the xy-snapping method proposed in [48], which only uses the z-component from the reconstructed shape, while fixing the x, y coordinates. We do not resample the target objects multiple times prior to factorization since it did not seem to make a difference.

Figure 4-12 shows reconstructions for various types of birds using the four methods from three viewpoints: camera view, 45° azimuth and 45° elevation. The colors indicate depth

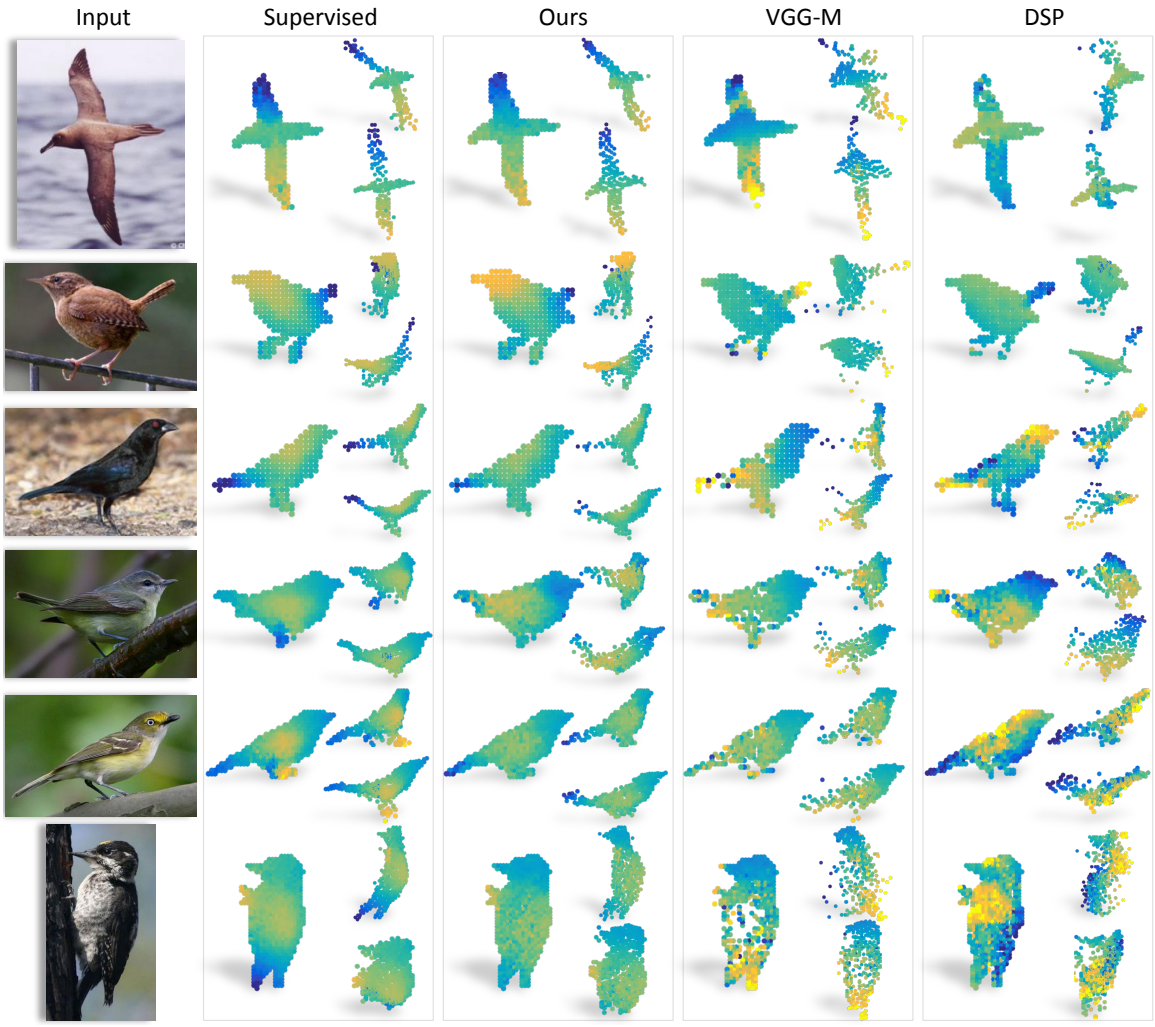


Figure 4-12: Sample reconstructions showing 3 views for each method: The camera viewpoint followed by the 45° azimuth in counter-clockwise direction (top right) and 45° elevation (bottom right). Colors show the depth where yellow is closer and blue is farther. The supervised method uses the spatial prior computed from annotated part correspondences, which can be seen as an upper bound. No part correspondences were used for the last three methods. WarpNet consistently obtains reconstructions most similar to the supervised method.

values (yellow is close, blue is far), with range fixed across all methods. WarpNet produces reconstructions that are most consistent with the `supervised` approach. Reconstructions from VGG-M and DSP are noisy due to errors in matching and often produce extreme outlier points that had to be clipped for ease of visualization. Articulated parts such as tails and wings are particularly challenging to match, where VGG-M and DSP often fail to recover consistent depths. Please see these results in 360° from the supplementary video https://goo.gl/w8DF1m.

4.5 Conclusion

In this chapter, we introduced a framework for improving the quality of matches between objects in fine-grained datasets without using human keypoint annotations. Our target application is single-view object reconstruction where prior works rely on some form of keypoint annotation during the reconstruction process, which is expensive and not scalable. The core of our approach is a novel deep learning architecture that predicts a 2D deformation field between two objects of fine-grained categories, parameteriaed by TPS transformations. We show that our network can be trained without supervised human keypoint annotations by exploiting the shape commonality in fine-grained datasets and use its output as a spatial prior for accurate matching.

Our approach achieves significant improvements over prior state-of-the-art without using part annotations and we show reconstructions of similar quality as supervised methods. Key challenges for future work are to determine optimal subsets of images for reconstruction and a good order for adding images that allows incremental reconstruction with bundle adjustment.

One caveat of modeling the deformation in the 2D space is that it cannot model out-

of-plane rotation. Due to this, WarpNet is capable of hallucinating birds to be of similar pose even when the baseline is wide. This may be avoided by a better choice of image pairs for matching, so that only birds with similar viewpoints are matched or by predicting a "matchable" region mask as in [238]. Another fundamental way of dealing with this is to model the deformation space in 3D. This is the topic of the following chapters, where we explore how to learn a 3D deformable model of animals.

Chapter 5

Learning 3D Deformation of Animals from 2D images

5.1 Introduction

Recent advances in computer vision and graphics have enabled the collection of high-quality 3D models with tools such as multi-view stereo [77] and commercial depth-sensors [103]. However, it is still difficult to obtain models of highly articulated and deformable objects like animals. In November 2015, searching Turbosquid for "chair" returns 24,929 results, while "cat" returns only 164 results. On the other hand, the Internet is brimming with cat pictures. In this chapter, we aim to create new 3D models of an animal by deforming a template 3D model to fit a 2D image. We assume that a sparse (less than 30) set of 2D-to-3D point correspondences are available through user clicks, which serve as positional constraints that guide the template deformation.

The contents of this work is in collaboration with Shahar Z. Kovalsky, Ronen Basri, and David Jacobs, presented at Eurographics 2016 [116]. This material is based upon work supported by the National Science Foundation under grant no. IIS-1526234, the Israel Binational Science Foundation, Grant No. 2010331 and the Israel Science Foundation Grants No. 1265/14

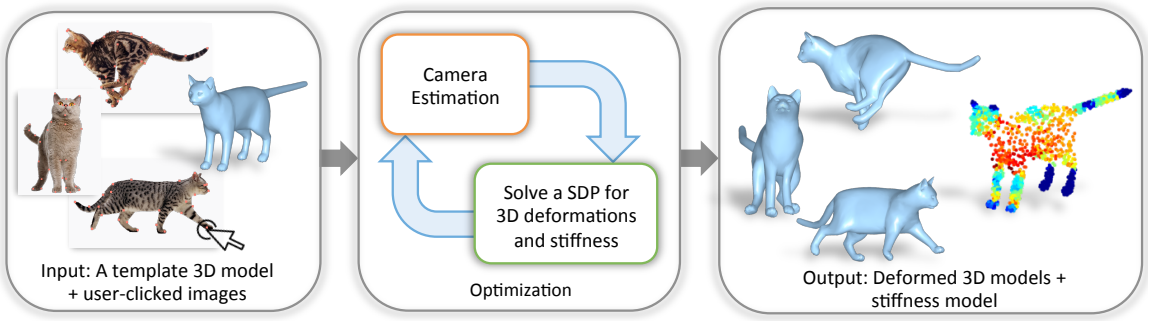


Figure 5-1: **Overview.** Our inputs are a reference 3D model and a set of images with user clicked 3D-to-2D point correspondences. The algorithm then alternates between solving for the camera viewpoint and the 3D deformations for all images. Our novel formulation allows us to solve for the deformation for each image and the stiffness model of the animal jointly in a single semidefinite program (SDP). The outputs of our algorithm are a set of deformed 3D models and the stiffness model, which specifies the rigidity of every local region of the animal (red indicates high deformability and blue indicates rigidity).

However, even with user-provided point correspondences, it is challenging to deform the template in a realistic and plausible manner. On top of the ill-posed nature of recovering 3D from 2D, animals are highly deformable, but not in a uniform way. In order to deform the template realistically, we argue that it is critical to understand how an animal can deform and articulate. For example, looking at many images of cats shows that a cat's body may curl up like a ball or twist and that its limbs articulate, but its skull stays mostly rigid. Hence, when modifying a 3D template model of a cat, we should restrict the amount of deformation allowed around the skull, but allow larger freedom around limb joints and the torso.

Here, we propose a novel deformation framework that aims to learn an animal-specific 3D deformation model from a set of annotated 2D images and a template 3D model. Our framework is inspired by the idea of *local stiffness* field, which specifies the amount of distortion allowed for a local region. The concept of stiffness is used in 3D deformation methods to model natural bending at joints and elastic deformations [170, 36]. In previous methods, the stiffness field is provided by users or learned from a set of vertex-aligned 3D meshes in various poses [170]. Instead, we learn the stiffness field from user-clicked 2D images

using the insight that highly deformable regions are sparse and consistent across multiple images. The idea is that large distortion is only allowed for those regions that require high deformation across many images. To our knowledge, our work is the first to learn stiffness of a 3D model from annotated 2D images.

We depart from the traditional skeleton models in this chapter, which are a set of rigid sticks connected by deformable joints [17, 224, 143]. Skeleton models are an excellent low-dimentional model for articulation as used in chapters 3 and 6. However, they are created by artists or learned from a large set of 3D scans of objects in various poses, and it's not clear how to learn them from 2D images. Creating the skeleton model requires a prior knowledge of how the object articulates (*e.g.* how many bones to use?), and this is the kind of knowledge that we wish to learn from images. Another benefit of our model is that it can represent continuous pose changes, which is essential for representing local deformations.

Figure 5-1 shows an overview of our proposed framework. Given a stock 3D cat mesh and target images of cats, a user provides 3D-to-2D point correspondences by clicking key features in images. These are passed on to the proposed algorithm, which simultaneously deforms the mesh to fit each cat's pose and learns a cat-specific model of 3D deformation. In the end, we obtain new 3D models for each target image and a stiffness model that describes how cats may deform and articulate.

Our primary contribution is a deformation framework that learns an animal-specific model of local stiffness as it deforms the template model to match the user-clicked 2D-to-3D correspondences. Specifically,

- We propose a locally bounded volumetric deformation energy that controls the maximal amount of distortion applied to local regions of the model using the recent optimization techniques of [124]. The bounds act as a local stiffness model of the animal, which we

learn by imposing a L1 sparsity penalty. The final deformation is orientation preserving and has worst-case distortion guarantees.

- We show that both the deformation and the stiffness bounds can be solved jointly as a sequence of convex optimization problems.

- We demonstrate the effectiveness of our framework on cats and horses, which are challenging animals as they exhibit large degrees of deformation and articulation.

5.2 Problem statement and background

We consider the problem of modifying a template 3D mesh of an animal according to a set of user-clicked photographs of the target animal. Our goal is to produce plausible 3D models guided by the annotated images, not necessarily obtaining precise 3D reconstructions of the images. In particular, given a sparse set of 2D-to-3D correspondences obtained from user-clicks, we wish to solve for a set of class-specific 3D deformations that faithfully fit the image annotations.

More formally, we are given a 3D template model, represented by a surface mesh $\mathbf{S} \subset \mathbb{R}^3$ as well as N images of class instances $I^1, \ldots, I^N$. Each image is associated with a sparse set of user prescribed point correspondences to the 3D model; namely, the i'th image I^i comes with pairs $\{(\boldsymbol{x}_k^i, \boldsymbol{p}_k^i)\}$ relating the surface point $\boldsymbol{x}_k^i \in \mathbf{S}$ to a 2D image location $\boldsymbol{p}_k^i \in \mathbb{R}^2$. Our goal is to leverage the N annotated images to learn a deformation model $\mathcal{D}$ capturing the possible deformations and articulations of the object class. In particular, for each image I^i we wish to find a deformation $\Phi^i \in \mathcal{D}$ that maps its 3D landmark points $\{\boldsymbol{x}_k^i\}$ to their

corresponding image points $\{\boldsymbol{p}_k^i\}$ once projected to the image plane; namely, satisfying

$$\begin{bmatrix} \boldsymbol{p}_k^i \\ 1 \end{bmatrix} = \Pi^i \begin{bmatrix} \Phi^i(\boldsymbol{x}_k^i) \\ 1 \end{bmatrix}, \tag{5.1}$$

where $\Pi^i \in \mathbb{R}^{3\times 4}$ is the camera projection matrix for the i'th image. In what follows we assume weak perspective projection, which is an orthographic projection followed by scaling of the x and y coordinates:

$$\Pi = \begin{bmatrix} \alpha_x & & \\ & \alpha_y & \\ & & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}_1 & t_1 \\ \mathbf{r}_2 & t_2 \\ \mathbf{0} & 1 \end{bmatrix}. \tag{5.2}$$

$\mathbf{r}_1$ and $\mathbf{r}_2$ are the first two rows of the object rotation matrix, t_1, t_2 are the first two coordinates of the object translation, and $\frac{\alpha_x}{\alpha_y}$ specifies the camera aspect ratio. Its parameters can be solved in a least squares approach given six or more 3D-to-2D point correspondences. Please see [91] for more information. Note that perspective projection may be similarly handled.

5.2.1 Parameterized deformation model

We parameterize the deformations of the surface model $\mathbf{S}$ by introducing an auxiliary tetrahedral mesh enclosed within the surface, $\mathbf{M} = (\mathbf{V}, \mathbf{T})$, where $\mathbf{V} \in \mathbb{R}^{3\times n}$ is a matrix of n coarse vertex coordinates and $\mathbf{T} = \{t_j\}_{j=1}^m$ is a set of m tetrahedra (tets). Every surface point $\boldsymbol{x} \in \mathbf{S}$ can then be written as a linear combination of the vertices $\mathbf{V}$. In particular, for the landmark points we set $\boldsymbol{x}_k^i = \mathbf{V}\boldsymbol{\alpha}_k^i$, where $\boldsymbol{\alpha}_k^i \in \mathbb{R}^n$ is a coefficient vector computed by linear moving least squares [133]. Figure 5-2 shows the surface and the tetrahedral mesh of a template cat model. The use of a tetrahedral mesh introduces a notion of volume to the

Figure 5-2: A template 3D surface and its auxiliary tetrahedral mesh with surface vertices shown in blue dots.

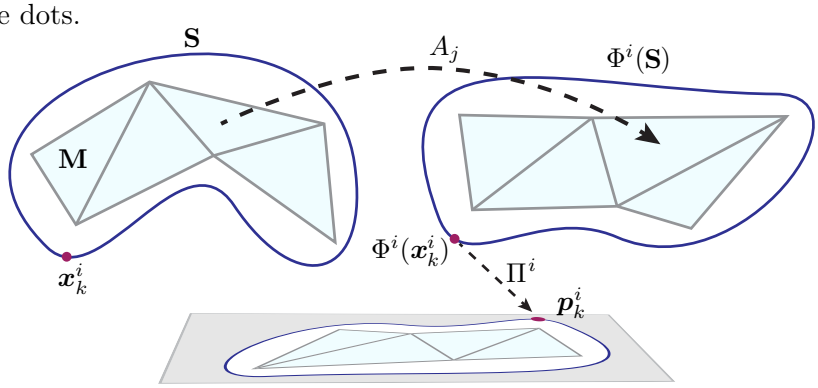


Figure 5-3: Illustration of the deformation model.

model making it more robust at preserving volumetric detail [62, 236].

Deformations of $\mathbf{M}$ thereby induce deformations of the surface $\mathbf{S}$. Specifically, we shall consider continuous piece-wise linear (CPL) maps $\Phi : \mathbf{M} \to \mathbb{R}^3$, whereby the deformation, restricted to the j'th tet, is defined by the affine map $\boldsymbol{v} \mapsto A_j\boldsymbol{v} + \boldsymbol{t}_j$. Φ maps the vertices $\mathbf{V}$ to new locations $\mathbf{U} \in \mathbb{R}^{3\times n}$. In fact, Φ is uniquely determined by the new vertex locations $\mathbf{U}$; for the j'th tet, the following full rank linear system holds

$$\begin{pmatrix} \boldsymbol{u}_{j1} & \boldsymbol{u}_{j2} & \boldsymbol{u}_{j3} & \boldsymbol{u}_{j4} \end{pmatrix} = \begin{bmatrix} A_j & \boldsymbol{t}_j \end{bmatrix} \begin{pmatrix} \boldsymbol{v}_{j1} & \boldsymbol{v}_{j2} & \boldsymbol{v}_{j3} & \boldsymbol{v}_{j4} \\ 1 & 1 & 1 & 1 \end{pmatrix}, \tag{5.3}$$

where $\boldsymbol{v}_{j\cdot}$ and $\boldsymbol{u}_{j\cdot}$ are its four vertices in the original and the deformed mesh respectively.

We denote by $A_j = A_j(\mathbf{U})$ the linear part of each affine transformation, linearly expressed in terms of the new vertex locations $\mathbf{U}$. Lastly, note that subject to a deformation $\Phi = \Phi_{\mathbf{U}}$ the location of the landmark points can be simply expressed as $\hat{\boldsymbol{x}}_k^i = \Phi_{\mathbf{U}}(\boldsymbol{x}_k^i) = \mathbf{U}\boldsymbol{\alpha}_k^i$. This relationship along with the positional constraints are depicted in Figure 5-3.

5.2.2 Landmark-guided 3D deformation

Our goal is to deform the template $\mathbf{S}$ such that (5.1) is satisfied without introducing local distortions to its shape. A popular approach to prevent distortion is minimizing the as-rigid-as-possible (ARAP) functional [10, 195]:

$$f_{\texttt{ARAP}}(\mathbf{U}) = \sum_{j=1}^{m} ||A_j - R_j||_F^2 |t_j|, \tag{5.4}$$

where $R_j \in SO(3)$ is the closest rotation to A_j and $|t_j|$ is the normalized volume of the j'th tet. Intuitively, ARAP tries to keep the local transformations applied to each tet of the mesh as similar as possible to a rigid transformation. Note that while the ARAP functional is non-convex, it is convex-quadratic for fixed rotations R_j.

The ARAP functional minimizes the ℓ_2-norm of a "non-rigidity" measure, which strives to evenly distribute local deviations from rigid transformation. As such, it fails to faithfully represent articulation and local deformations. Moreover, it is not straightforward to adapt this functional alone to benefit from having many annotated image exemplars. In this work, we also use the ARAP functional, but allow non-uniform distribution of distortion by assigning local stiffness as described in the next section.

5.3 Learning stiffness for articulation and deformation

Natural objects do not usually deform in a uniform manner; some parts such as joints deform a lot more while parts such as the limbs and skull stay rigid. In order to model such deformation and articulation, we introduce the notion of local stiffness, which specifies how much distortion is allowed at each tet. We learn local stiffness from data using a sparsity promoting energy, so large deformations are concentrated in regions that require them across many images. In this section we discuss how we simultaneously deform the template $\mathbf{S}$ to match each of the images $I_1, \ldots, I_N$ while learning the stiffness.

5.3.1 Modeling local stiffness

Denote by $\mathbf{U}^i$ the deformation mapping $\mathbf{S}$ to the i'th image I^i, and by $\{A_j^i\}$ the linear transformations associated with its tets. Inspired by [135, 124], we control deformations by explicitly imposing constraints on their linear parts.

First we require that each A_j^i satisfies

$$\det(A_j^i) \geq 0, \tag{5.5}$$

which entails that the mapping is locally injective and orientation preserving; in particular, tets may not flip. Second, we bound the *local isometric distortion* with the constraint

$$\max\left\{\|A_j^i\|_2, \|A_j^{i\,-1}\|_2\right\} \leq 1 + \epsilon + s_j \tag{5.6}$$

where $\|\cdot\|_2$ is the operator (spectral) norm. The small constant $\epsilon \geq 0$ is common for all tets and governs the degree of global non-rigidity. $s_j \geq 0$ is the local *stiffness* for the j'th tet controlling how much this particular tet may deform. Note that ϵ and s_j are not image

specific (i.e. they are independent of i) and encode the class-prior of how an object class can deform and articulate.

Intuitively, $\|A_j^i\|_2$ and $\|{A_j^i}^{-1}\|_2$ quantify the largest change of Euclidean length induced by applying A_j^i to any vector. Therefore, Equation (5.6) bounds local length changes by a factor of $1 + \epsilon + s_j$. If, for example, $\epsilon = s_j = 0$ then A_j^i must be a rotation; looser bounds allow "less locally isometric" deformations. In practice, ϵ is set to a small value and is fixed throughout the experiments.

5.3.2 Optimizing articulation and deformation

Subject to these constraints, we propose minimizing an energy comprising three terms:

$$f = f_{\texttt{DEFORM}} + \lambda f_{\texttt{POS}} + \eta f_{\texttt{STIFFNESS}}. \tag{5.7}$$

$f_{\texttt{DEFORM}}$ is defined via the ARAP deformation energy (5.4) as

$$f_{\texttt{DEFORM}} = \frac{1}{N} \sum_{i=1}^{N} f_{\texttt{ARAP}}(\mathbf{U}^i). \tag{5.8}$$

$f_{\texttt{POS}}$ is defined by

$$f_{\texttt{POS}} = \frac{1}{N} \sum_{i=1}^{N} \sum_{k} \left\| \begin{bmatrix} \boldsymbol{p}_k^i \\ 1 \end{bmatrix} - \Pi^i \begin{bmatrix} \mathbf{U}^\mathbf{i} \boldsymbol{\alpha}_k^i \\ 1 \end{bmatrix} \right\|_2^2, \tag{5.9}$$

which accounts for the user prescribed correspondences and the camera parameters, aiming to satisfy (5.1). Lastly, we set

$$f_{\texttt{STIFFNESS}} = \|\boldsymbol{s}\|_1, \tag{5.10}$$

where $\boldsymbol{s}$ is the vector whose elements are the local stiffness bounds $\{s_j\}$. This L1 regularization encourages most s_i to be 0, so that only those tets that must distort are allowed to do so.

λ is a parameter that controls the trade-off between satisfying the constraints and preserving the original shape of $\mathbf{M}$. η is a parameter that controls the strength of the stiffness regularization. As η increases, it forces most A_j to stay rigid and as η approaches 0 the solution approaches that of the ARAP functional and the positional constraints. See Section 5.4 for parameter settings.

In conclusion, jointly deforming the template $\mathbf{S}$ to match each of the images $I_1, \ldots, I_N$, while estimating the local stiffness boils down to the following optimization problem:

$$\begin{aligned} \min_{\{\mathbf{U}^i\},\{\Pi^i\},\boldsymbol{s}} \quad & f_{\texttt{DEFORM}} + \lambda f_{\texttt{POS}} + \eta f_{\texttt{STIFFNESS}} \\ \text{s.t.} \quad & A_j^i = A_j^i(\mathbf{U}^i), \quad \forall\, j = 1, \ldots, m,\ i = 1, \ldots, N \\ & \det(A_j^i) \geq 0, \\ & \max\left\{ \|A_j^i\|_2, \|{A_j^i}^{-1}\|_2 \right\} \leq 1 + \epsilon + s_j, \\ & s_j \geq 0. \end{aligned} \tag{5.11}$$

Note that usually in prior work, deformations are solved independently for each set of positional constraints, since there is nothing that ties multiple problems together. Introducing a shared stiffness field allows us to leverage information from multiple images and improve the quality of results for all images.

5.3.3 Realizing the optimization

Optimizing (5.11) is not straightforward, as it involves the non-convex constraint (5.6). We realize these constraints in a convex optimization framework based on the construction presented in [124] for optimization subject to bounds on the extremal singular values of matrices.

This previous work makes the observation that the set of matrices whose maximal singular value, $\sigma_{\max}$, is bounded from above by some constant $\Gamma \geq 0$ is convex and can be written as a linear matrix inequality (LMI):

$$\mathcal{C}^{\Gamma} = \left\{ A \in \mathbb{R}^{n \times n} : \begin{pmatrix} \Gamma I & A \\ A^T & \Gamma I \end{pmatrix} \succeq 0 \right\}. \tag{5.12}$$

It is further shown that for any rotation matrix $R \in SO(n)$, the set

$$R\mathcal{C}_{\gamma} = \left\{ RA \in \mathbb{R}^{n \times n} | \frac{A + A^T}{2} \preceq \gamma I \right\}, \tag{5.13}$$

is a maximal convex subset of the non-convex set of matrices with non-negative determinant whose minimal singular value, $\sigma_{\min}$, is bounded from below by some constant $\gamma \geq 0$. This calls for an iterative algorithm in which R is updated in each iteration so as to explore the entire set of matrices with bounded minimum singular value. As suggested by [124], a natural choice for R is the closest rotation to A. This choice, in turn, also minimizes the ARAP functional in Equation (5.4) for a fixed A.

In order to employ the convex optimization framework of [124], we rewrite the constraints (5.5) and (5.6) as

$$1/c_j \leq \sigma_{\min}(A_j^i) \leq \sigma_{\max}(A_j^i) \leq c_j \quad \text{and} \quad \det(A_j^i) \geq 0,$$

with $c_j = 1 + \epsilon + s_j$. This follows by observing that $\|A_j^i\|_2 = \sigma_{\max}(A_j^i)$ and $\|{A_j^i}^{-1}\|_2 = 1/\sigma_{\min}(A_j^i)$. Plugging (5.11) into the framework of [124] then yields the following optimization

problem:

$$
\begin{aligned}
\min \quad & f_{\texttt{DEFORM}} + \lambda f_{\texttt{POS}} + \eta f_{\texttt{STIFFNESS}} \\
\text{s.t.} \quad & A_j^i = A_j^i(\mathbf{U}^i), \quad \forall\, j = 1, \ldots, m,\ i = 1, \ldots, N \\
& A_j^i \in \mathcal{C}^{\Gamma_j^i}, \\
& A_j^i \in R_j^i \mathcal{C}_{\gamma_j^i}, \\
& s_j \geq 0, \\
& \Gamma_j^i \leq (1 + \epsilon + s_j), \\
& \frac{1}{(1 + \epsilon + s_j)} \leq \gamma_j^i,
\end{aligned}
\tag{5.14}
$$

whose optimization variables are $\{\mathbf{U}^i\}$,$\{\Gamma_j^i\}$,$\{\gamma_j^i\}$ and $\boldsymbol{s}$.

Lastly, we note that the last constraint of (5.14) is convex; in fact, following a standard derivation (e.g., see Appendix [11]), it can be equivalently rewritten as a convex second-order cone constraint,

$$
\left\| \begin{pmatrix} 2 \\ (1 + \epsilon + s_j) - \gamma_j^i \end{pmatrix} \right\| \leq (1 + \epsilon + s_j) + \gamma_j^i. \tag{5.15}
$$

See B for details. Therefore, with fixed $\{R_j^i\}$ and $\{\Pi^i\}$, Equation (5.14) is a semidefinite program (SDP) and can be readily solved using any SDP solver. However, note that the entire problem is not convex due to the interaction between $R_j^i, \mathbf{U}^i$, and Π^i. Thus, we take a block-coordinate descent approach where we alternate between two steps: (a) update R_j^i and Π^i fixing $\mathbf{U}^i$, (b) update $\mathbf{U}^i$ fixing R_j^i and Π^i via solving Equation (5.14). As in [124], we find that allowing the surface to deform gradually makes the algorithm less susceptible to local minima. To this end, we initialize the procedure with a large η, which controls the degree of non-rigidity, and slowly reduce its value as the algorithm converges. This algorithm

is outlined in Algorithm 1.

Algorithm 1: Jointly solving for the deformations and the stiffness

Input: Template 3D mesh $\mathbf{S}$, its auxiliary tetrahedral mesh $\mathbf{M} = (\mathbf{V}, \mathbf{T})$, and N 3D-to-2D annotated images $\{I^i\}$

Output: N deformed auxiliary tetrahedral meshes vertices $\{\mathbf{U}^i\}$, the projection matrices $\{\Pi^i\}$, and the stiffness model $\boldsymbol{s}$

$\texttt{maxIter} = 10$;

$\widetilde{\mathbf{U}}^{\mathbf{i}} = \mathbf{V}, \quad i = 1 \dots N$; // initialize

for $\eta \leftarrow \eta_{\max}$ **to** $\eta_{\min}$ **do** // warm start

  $\mathbf{U}^{\mathbf{i}(0)} = \widetilde{\mathbf{U}}^{\mathbf{i}}$;

  $t = 0$;

  repeat

    Compute $\Pi^{i(t)}$ by solving Equation (5.1) with $\mathbf{U}^{i(t)}$;

    Compute the polar decompositions $A_j^{i(t)} = R_j^{i(t)} S_j^{i(t)}$;

    Update $\{\mathbf{U}^{i(t+1)}\}, \boldsymbol{s}^{(t+1)}$ by solving Equation (5.14) with $\Pi^{i(t)}$ and $R_j^{i(t)}$;

    $t = t + 1$;

  until *convergence* or $t > \texttt{maxIter}$

  $\widetilde{\mathbf{U}}^{\mathbf{i}} = \mathbf{U}^{\mathbf{i}(t)}$;

return $\{\mathbf{U}^{i(t)}\}, \{\Pi^{i(t)}\}, \boldsymbol{s}^{(t)}$

5.4 Experimental Detail

We use our approach as described to modify a template 3D mesh according to the user-clicked object pose in 2D images. We first compare our approach with the recent method of Cashman et al. [50], which is the closest work to ours with publicly available source code [49]. We then present an ablation study where key components of our model are removed in order to evaluate their importance and provide qualitative and quantitative evaluations.

We experiment with two object categories, cats and horses. We collected 10 cat and 11 horse images in a wide variety of poses from the Internet. Both of the template 3D meshes were obtained from the Non-rigid World dataset [43]. These templates consist of ∼3000 vertices and ∼6000 faces, which are simplified and converted into tetrahedral meshes of 510, 590 vertices and 1500, 1700 tets for the cat and the horse respectively via a tet generation

software [186]. We manually simplify the mesh in MeshLab until there are around 300 vertices. We found automatic simplification methods over-simplify skinny regions and fine details, leading to a poor volumetric tet-representation. The cat template and its auxiliary tetrahedral mesh are shown in Figure 5-2. The template mesh used for horses can be seen in Figure 5-8. For all experiments we set $\epsilon = 0.01$, and $\lambda = 10$. In order to allow gradually increasing levels of deformation, we use $\eta_{\max} = 0.5$ and $\eta_{\min} = 0.05$ with 10 log-steps in between for all experiments. The values for η and λ were set by hand, but deciding on the values did not require much tuning.

In each iteration, the camera parameters are computed using the 2D-to-3D correspondences. We initialize the parameters using the direct linear transform algorithm and refine it with the sparse bundle adjustment package [91, 144]. In order to obtain annotations, we set up a simple system where the user can click on 2D images and click on the corresponding 3D points in the template mesh. Our system does not require the same vertices to be annotated in every image. The average number of points annotated for each image for both cats and horses was 29 points.

5.5 Results

5.5.1 Comparison with Cashman *et al.* [50]

Cashman *et al.* employ a low resolution control mesh on the order of less than 100 vertices which is then interpolated with Loop subdivision. In order to apply their method to ours, we simplified our template mesh with quadratic decimation until we reach around 150 vertices while retaining the key features of the template mesh as much as possible (shown in inset). Since their method relies on silhouettes, we provide hand-segmented silhouettes to their

algorithm along with the user-clicked points. We transferred the user-clicks from the full mesh to the simplified mesh by finding the closest 3D vertex in the simplified mesh for each labeled vertex in the full resolution mesh. We did not include points that did not have a close enough 3D vertex due to simplification. On average 24 points were labeled for their experiment and we use their default parameters.

Figure 5-5 compares the results obtained with the method of [50] and our model. Two views are shown for each result, one from the estimated camera pose and another from an orthogonal viewpoint. As the authors in [50] point out, their method focuses on modeling shape and is not designed for highly articulated objects such as cats. Consequently, we can see that it has difficulties dealing with the wide range of poses present in our cat dataset. Regions such as limbs and tails especially lose their original shape. Their method is based on surface deformation, which does not have a notion of volume. This causes flattening of the 3D models as can been seen in the orthogonal views. Since we guarantee worst-case distortion and orientation preserving deformation of the auxiliary mesh, our surface reconstructions are well behaved compared to [50]. Recall that silhouettes, along with the user-clicked points, are used to obtain the results for [50].

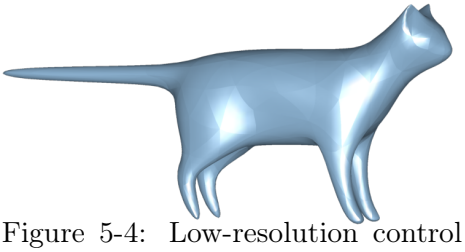


Figure 5-4: Low-resolution control mesh for subdivision surfaces

5.5.2 Qualitative Evaluation

The 3D models in Figure 5-1 were obtained using our proposed framework. We now evaluate the effectiveness of the proposed framework by comparing the results without any distortion bounds (i.e. removing Equation (5.6)) and with constant distortion bounds (i.e. fixing s_j

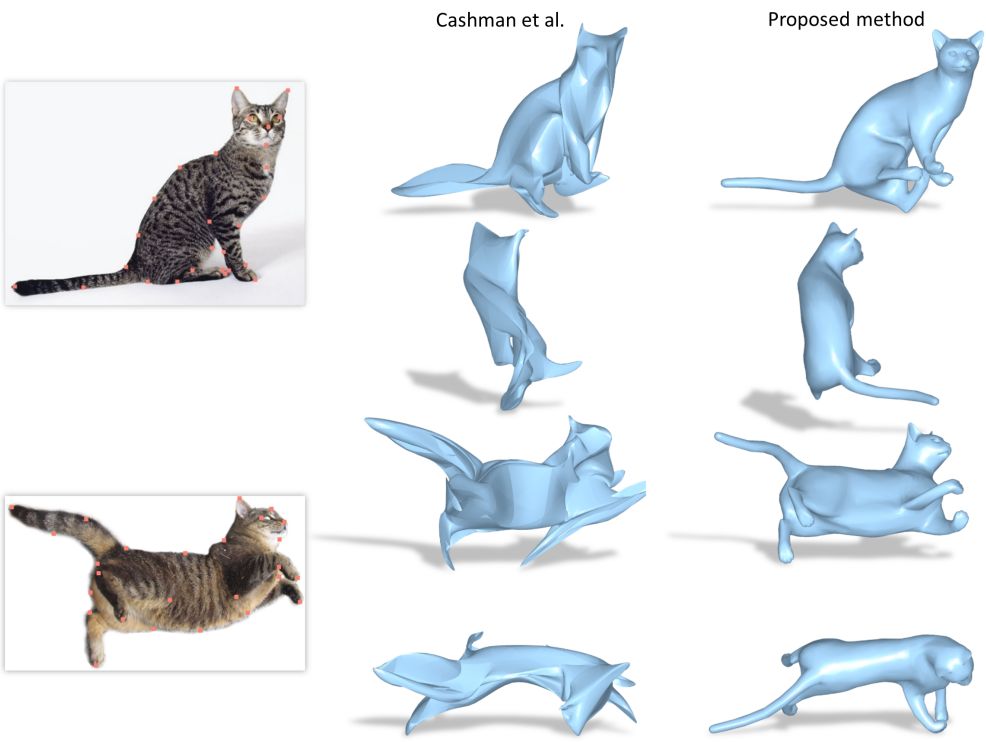


Figure 5-5: **Comparison with Cashman *et al.* [50]** : the first column shows the user-clicked input images, the second column shows the result of [50] and the third column shows the result of our proposed method. Two views are shown for each image, one from the estimated camera and another from an orthogonal view point. Our method is more robust to large deformations and retains the volume of the model. Note that silhouettes along with the user-clicked points are used for [50].

to a constant). Qualitative results of this ablation study are shown in Figure 5-6. The first column shows input images along with their user-clicked points. The second column shows results obtained with no bounds, leaving just the ARAP energy, which we refer to as `Unbounded`. This is similar to the approach used in [121], but with volumetric instead of surface deformation. The third column, `Uniform`, shows results obtained with a uniform bound, where the stiffness $1+\epsilon+s_j$ is replaced with a single constant $c_j = 2$ for all faces. This is the deformation energy used in [124] applied to 2D positional constraints. The constant was slowly increased from 1 to 2 in a manner similar to η in order to allow for increasing levels of deformation. Finally, in the last column we show results obtained with the proposed

framework where the distortions are bounded with local stiffness.

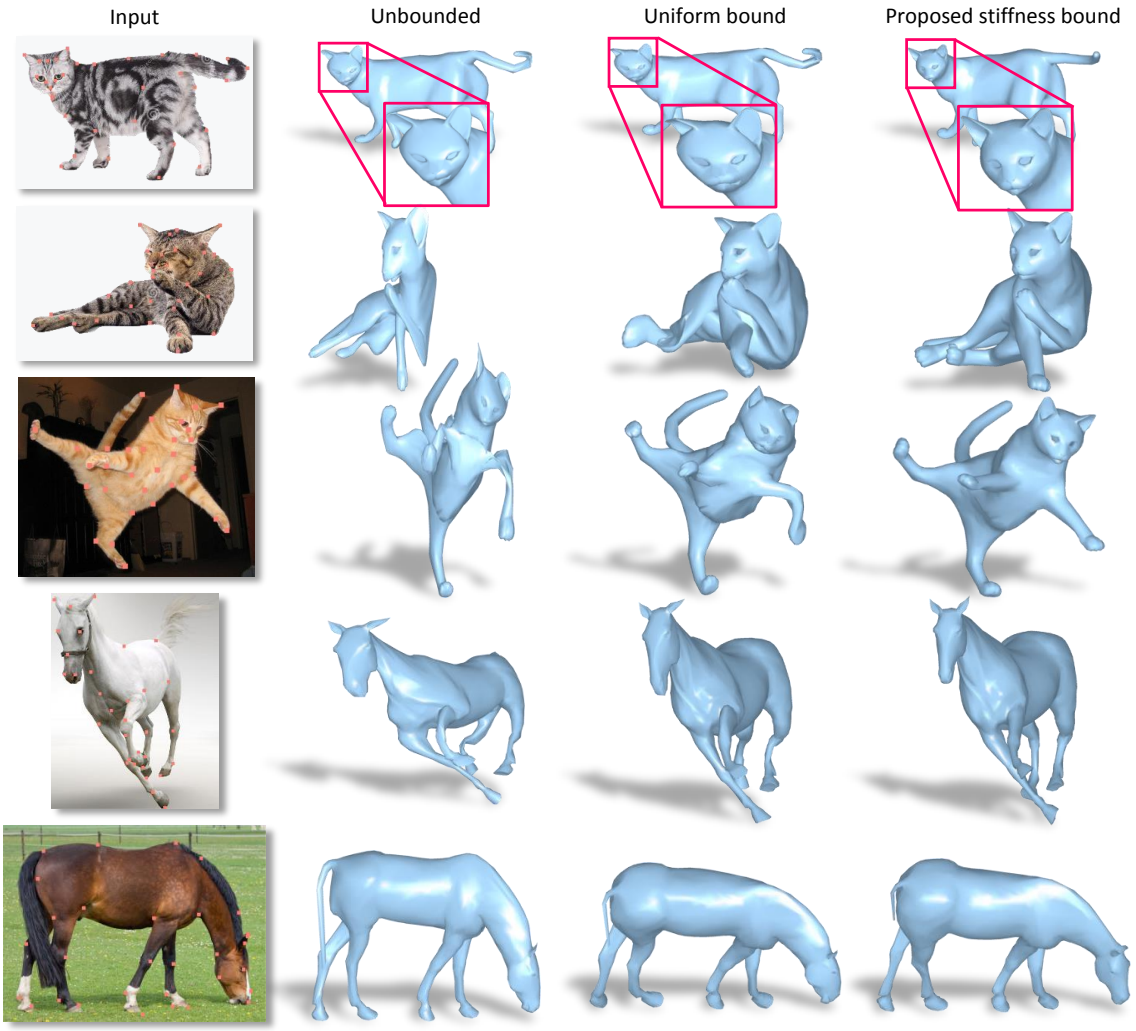


Figure 5-6: **Ablation study.**User-clicked input images (first column). Unbounded (second column) is the model without any bounds on the distortion leaving just the volumetric ARAP energy. Uniform (third column) is when the stiffness bounds (s_j in Equation (5.6)) are replaced by a single constant, which is the approach of [124] applied to 2D positional constraints. The last column shows the results with our complete framework where the stiffness bounds and the deformations are jointly optimized. Without the animal-specific stiffness, distortions either spread out too much or concentrate around the positional constraints.

First, notice the wide range of poses present in the images used; some are particularly challenging requiring large deformation from the template 3D mesh. In general, `Unbounded` concentrates high distortions near positional constraints causing unnatural stretching and deformation around limbs and faces. This is evident with horse legs in row 4 as `Unbounded`

deforms them in an elastic manner. `Uniform` distributes the distortions, however, when the pose change from the template is significant, distortions spread out too much causing unrealistic results as seen in rows 2 and 3. The unnatural distortion of the faces is still a problem with `Uniform`. The proposed framework alleviates problems around the face and the horse limbs as it learns that those regions are more rigid.Please refer to the supplementary materials for comprehensive results of all cat and horse experiments.

We provide visualizations of the learned stiffness model in Figure 5-7 and 5-8. Figure 5-7 visualizes the learned stiffness values for cats and horses in various poses. We show the centroid of tetrahedra faces colored by their stiffness values in log scale. Blue indicates rigid regions while red indicates highly deformable regions. Recall that there is one stiffness model for each animal category. The level of deformations present in the input images are well reflected in the learned stiffness model. For cats, the torso is learned to be highly deformable allowing the animal to twist and curl, while keeping the skull and limbs more rigid. Similarly for horses, the neck, the regions connecting the limbs as well as the joints are learned to be deformable while keeping skull, limbs, and buttocks region rigid. The fact that the buttocks is considered rigid is anatomically consistent, since horses have a rigid spine making them suitable for riding [109].

We also present segmentation results using the learned stiffness values as another form of visualization in Figure 5-8. In order to obtain the segmentations, we first transferred the stiffness values from tetrahedra faces to vertices by taking the mean stiffness of faces a vertex is connected to. Then, we constructed a weighted graph on the vertices based on their connectivity, where the weights are set to be the sum of the Euclidean proximity and the similarity of the stiffness values. We apply normalized cuts to partition this graph and interpolate the result to the surface mesh vertices using the parameterization described in

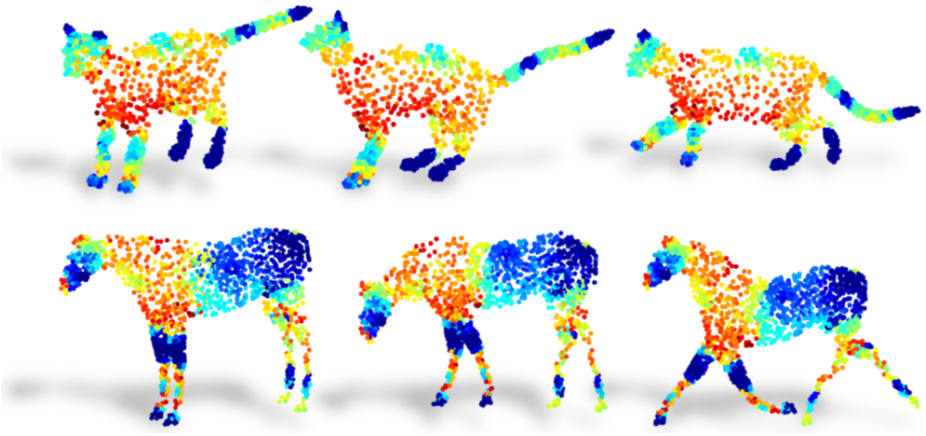


Figure 5-7: **Learned stiffness visualization.** Blue indicates rigid regions while red indicates highly deformable regions.

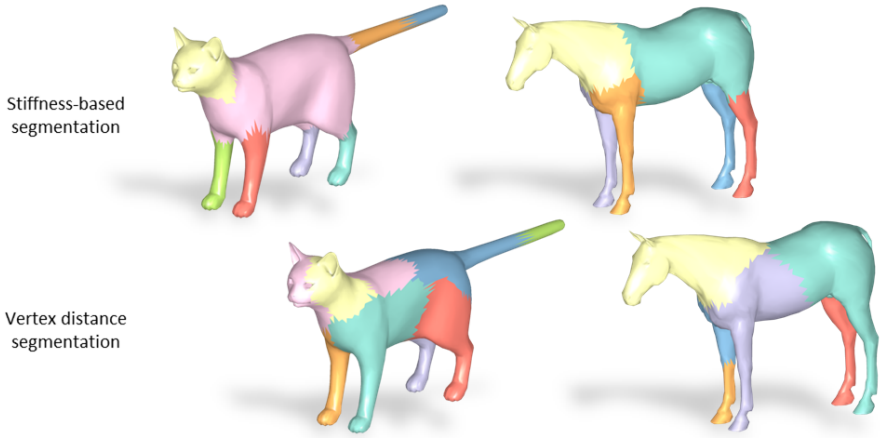


Figure 5-8: **Mesh segmentation.** A visualization of the learned stiffness by means of segmentation. Segmenting the template mesh using stiffness illustrates regions that deform together as learned by our framework. We see that they correspond to semantically reasonable segmentations. We show segmentation results based on vertex distance alone as a comparison.

Section 5.2.1. We also show the segmentation results using just the Euclidean proximity as a comparison. Stiffness-based segmentation illustrates that regions which deform together as learned by our framework correspond to semantically reasonable parts.

The learned stiffness model can be used as a prior to solve for stiffness-aware deformations of new annotated images. Figure 5-9 shows the results of deforming the template to new input images via using the stiffness values learned from the previous experiment, i.e. the new images were not used to learn the stiffness. Similar to other experiments, we do warm

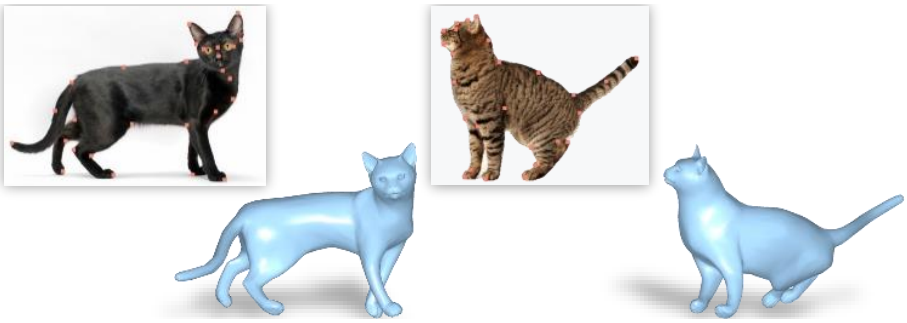


Figure 5-9: **Novel images.** Deformation results using the learned stiffness from 10 cats as a fixed prior for new images.

start where the stiffness bounds are linearly increased from 1.01 to their actual value in 10 steps. The results are visually very similar to the results obtained when the stiffness was learned with those images along with the other 10 cat images. From this perspective, the joint optimization for the stiffness and the deformations using multiple images is the "training" (Figure 5-6), while the single-image optimization with a fixed stiffness prior is the "testing" (Figure 5-9).

5.5.3 Quantitative evaluation

Lastly, we conduct an evaluation against the ground truth by using pictures of a rendered 3D model as the input to our framework. Specifically, we use the TOSCA dataset [44], which has 11 vertex-aligned models of cats in various poses. We take the neutral pose (`cat0`) as the template and randomly project the other 10 models to produce images where the ground truth shape is known. We randomly sample 35 points and use them as the 3D-to-2D correspondences. In order to guarantee that these points are well distributed, we segment the model into 15 components and make sure that there is at least one point from each component. These components correspond to key parts such as the paws, limbs, left and right ears, tail base and tip, etc. We compare the results of the `No Bound`, `Uniform`, and the proposed approach. Using this method, we produce two images from each ground truth model and conduct the experiment with 20 images.

We evaluate our method using several error metrics. First, we measure the distortions between the ground truth and the deformed models, which capture how natural the deformations are. We argue this is the most important measure since obtaining plausible deformations is the main goal of our algorithm. For this we use the stretch, edge-length, area, and angle distortion errors as defined in [229] by comparing the corresponding triangles. Additionally, we report the mean Euclidean distance between the 3D vertices, which measures how close the surface of the deformed models are to the ground truth. While a low Euclidean

Table 5.1: **Quantitative evaluation against ground-truth.** The numbers are lower the better for all metrics.

		Distortion error metric [229]			
Methods	Mean dist	Stretch	Edge	Area	Angle
`Unbounded`	0.291	1.01	0.156	0.216	0.159
`Uniform`	**0.281**	1.01	0.13	0.198	0.13
`Proposed`	0.287	**0.996**	**0.105**	**0.181**	**0.085**

distance is desirable for 3D reconstruction, we do not expect a close match everywhere due to ambiguities arising from a single view and sparse point constraints. In particular, Euclidean distance is not necessarily indicative of visual quality. We report the average error over all 20 input images. Before computing the error metrics, the deformed and ground truth models are aligned by a similarity transform. The results are shown in Table 5.1. As expected, all methods attain comparable mean Euclidean distance to ground truth, while our approach obtains substantially lower errors in distortion metrics. This demonstrates the advantage of learning stiffness from multiple images, yielding a more plausible deformation model.

Implementation details With an unoptimized MATLAB implementation, training with 10 images took 4 hours and testing a single image with a learned stiffness prior took ∼30 minutes. We use YALMIP [139] for the SDP modeling and MOSEK as the solver [16]. Our biggest bottleneck is the SDP optimization due to many LMI constraints. Reducing the number of tets can significantly reduce the run-time.

5.6 Discussion

Limitations of our current approach suggest directions for future work. One failure mode is due to a large pose difference between the template and the target object, which may lead to an erroneous camera parameters estimate (e.g., local minima), as seen in row 5 of Figure 5-6. Here, the head of the horse in the image is lowered for grazing while the head of the horse template is upright causing a poor initialization of the camera estimate. Using a user-supplied estimate of the viewpoint or automatic viewpoint estimation methods like [212] are possible solutions.

Another failure mode is due to the inherent depth ambiguity problem when only a single-view of the object is available, where many 3D shapes project to the same 2D observations.

As such, some of our deformed models do not have the "right" 3D pose when seen from a different view. A case can be seen in the video (https://goo.gl/Xp0QJQ) that shows 360 degree views of the final models. How the left ankle of the horse in row 4 of Figure 5-6 is bent in an physically impossible direction is also attributed to this. One reason is because the current stiffness model is isotropic. An interesting future direction is to make the distortion bounds dependent on the orientation of the transformation. This would allow the framework to learn that certain parts only deform in certain directions.

Our method could also be enhanced to prevent surface intersections or reason about occlusion (e.g. if the point is labeled, it should be visible from the camera). Run-time is also an issue for adapting the stiffness model into a real-time posing application. This may be addressed by recent advancements in efficiently computing mappings with geometric bounds [125].

Another failure mode is due to the oversimplification of skinny regions when computing the auxiliary template mesh. Horse legs particularly suffer from this problem. The flattening of the ankle of one of the horses in the video is due to the fact that only 1 or 2 tets are used to represent that region. This suggests another interesting direction, which is to use the learned stiffness field for class-specific or deformation-aware mesh processing tasks. For example for mesh simplification, it makes sense to use stiffness values to simplify regions that are rigid more aggressively than regions that are highly deformable.

Since our framework is based on a volumetric deformation approach, no explicit factorization between shape and pose exists as used in Chapters 3 and 6. However, enforcing sparse stiffness motivates the framework to model changes in pose. In order to explicitly model shape, one idea is to add another stiffness field specifically to account for shape variation with different constraints. However, prior works [32, 143] suggest that shape variations are

captured by low-dimensional models – this is the focus of the next chapter. The learned stiffness field could also be used for learning a 3D skeleton model (rigging) of the template mesh, from which the method discussed in the next chapter may be applied. Though note that the generality of our formulation is an advantage when the explicit shape and pose factorization is not clear *e.g.* octopus. Cats are also an instance of this because they are very flexible.

5.7 Conclusion

In this chapter we introduced an optimization framework to learn the 3D deformation model of an animal from a template 3D mesh and a set of use-clicked 2D images. We do so by introducing a notion of local stiffness that controls how much each face of the mesh may distort. Our formulation jointly solves for the deformed meshes that fits each image and the stiffness field. The key intuition is that highly deformable regions are sparse and consistent across multiple images. We conceptualize this by adding a sparsity term on the stiffness field and by forcing all images to share a single stiffness field. Our experiments show that learning a class-specific model of 3D deformation is essential for obtaining more plausible 3D deformations.

Chapter 6

3D Menagerie: Modeling the Shape of Quadrupeds

6.1 Introduction

In this chapter, we study how we can extend a similar approach to modeling 3D animals, building on the best practices learned from the modeling of 3D human bodies. We find that modeling animals presents novel challenges and describe how to overcome them. Specifically our goal is to learn a SMPL like generative model of the 3D pose and shape of animals and then fit this model to image observations as illustrated in Fig. 6-1. We focus on a subset of four-legged mammals that all have the same number of "parts" and model members of the families Felidae, Canidae, Equidae, Bovidae, and Hippopotamidae.

Animals, however, differ from humans in several important ways. First, the shape variation across species far exceeds the kinds of variations seen between humans. Even within

The contents of this work is in collaboration with Silvia Zuffi, David Jacobs, and Michael J. Black, presented at CVPR 2017 [244]. This material is based upon work supported by the National Science Foundation under grant no. IIS-1526234. We thank Seyhan Sitti for scanning the toys and Federica Bogo and Javier Romero for their help with the silhouette term.

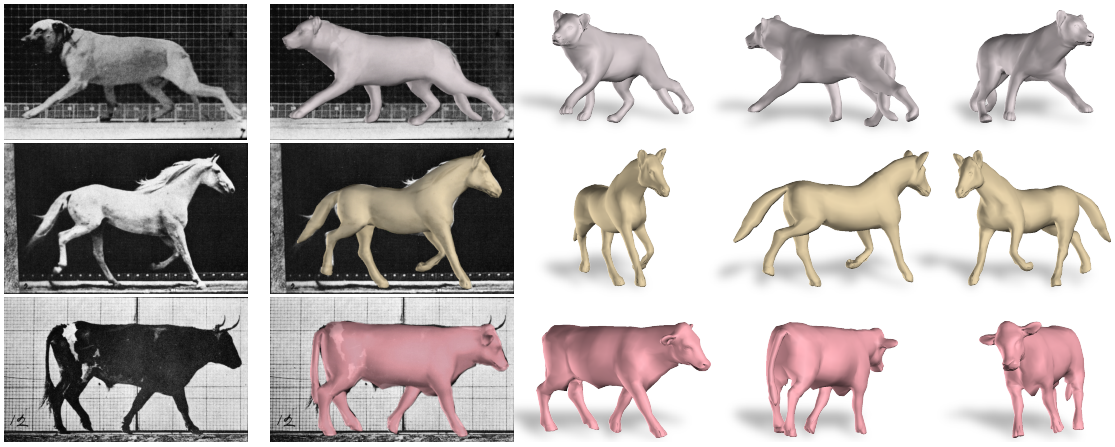


Figure 6-1: **Animals from images.** We learn an articulated, 3D, statistical shape model of animals using very little training data. We fit the shape and pose of the model to 2D image cues showing how it generalizes to previously unseen shapes.

the canine family, there is a huge variability in dog shapes as a result of selective breeding. Second, all these animals have tails, which are highly deformable and obviously not present in human shape models. Third, obtaining 3D data to train a model is much more challenging. SMPL and previous models like it (e.g. SCAPE [17]) rely on a large database of thousands of 3D scans of many people (capturing shape variation in the population) and a wide range of poses (capturing pose variation). Humans are particularly easy and cooperative subjects. It is impractical to bring a large number of wild animals into a lab environment for scanning and it would be difficult to take scanning equipment into the wild to capture animals shapes in nature. Since scanning live animals is impractical we instead scan realistic toy animals to create a dataset of 41 scans of a range of quadrupeds as illustrated in Fig. 6-2. We show that a model learned from toys generalizes to real animals.

The key to building a statistical 3D shape model is that all the 3D data must be in correspondence. This involves registering a common template mesh to every scan. This is a hard problem, which we approach by introducing a novel part-based model and inference scheme that extends the "stitched puppet" (SP) model [243]. Our new *Global-Local Stitched*

Figure 6-2: **Toys.** Example 3D scans of animal figurines used for training our model.

Shape model (*GLoSS*) aligns a template to different shapes, providing a coarse registration between very different animals (Fig. 6-5 left). The GLoSS registrations are somewhat crude but provide a reasonable initialization for a *model-free* refinement, where the template mesh vertices deform towards the scan surface under an As-Rigid-As-Possible (ARAP) constraint [195] (Fig. 6-5 right).

Our template mesh is segmented into parts with blend weights so that it can be reposed using linear blend skinning (LBS). Using this we "pose normalize" the refined registrations to a neutral pose and learn a low-dimensional shape space using principal component analysis (PCA). This is analogous to the SMPL shape space but for multiple species of animals [143] and produces a model where new shapes can be generated and reposed. With the learned shape model, we further refine the registration of the template to the scans using co-registration [97], which regularizes the registration by penalizing deviations from the model fit to the scan. We update the shape space and iterate to convergence.

The final *Skinned Multi-Animal Linear* model (*SMAL*) provides a shape space of animals trained from 41 scans. Because quadrupeds have shape variations in common, the model generalizes to new animals not seen in training. This allows us to fit SMAL to 2D data using manually detected keypoints and segmentations. As shown in Fig. 6-1 and Fig. 6-9, our model can generate realistic animal shapes in a variety of poses.

In summary we describe a method to create a realistic 3D model of animals and fit this model to 2D data. The problem is much harder than modeling humans and we develop new

tools to extend previous methods to learn an animal model. This opens up new directions for research on animal shape and motion capture.

6.2 Dataset

We created a dataset of 3D animals by scanning toy figurines (Fig. 6-2) using an Artec hand-held 3D scanner. We also tried scanning taxidermy animals in a museum but found, surprisingly, that the shapes of the toys looked more realistic. We collected a total of 41 scans from several species: 1 cat, 5 cheetahs, 8 lions, 7 tigers, 2 dogs, 1 fox, 1 wolf, 1 hyena, 1 deer, 1 horse, 6 zebras, 4 cows, 3 hippos. We estimated a scaling factor so animals from different manufacturers were comparable in size. Like previous 3D human datasets [178], and methods that create animals from images [50, 116], we collected a set of 36 hand-clicked keypoints that we use to aid mesh registration.

6.3 Global/Local Stitched Shape Model

The Global/Local Stitched Shape model (GLoSS) is a 3D articulated model where body shape deformations are locally defined for each part and the parts are assembled together by minimizing a stitching cost at the part interfaces. The model is inspired by the SP model [243], but has significant differences from it. In contrast to SP, the shape deformations of each part are analytic, rather than learned. This makes it more approximate but, importantly, allows us to apply it to novel animal shapes, without requiring *a priori* training data. Second, GLoSS

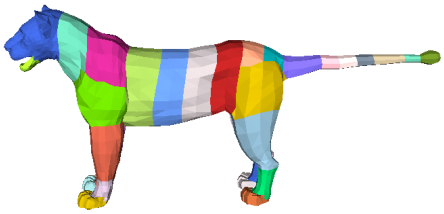


Figure 6-3: **Template mesh** is segmented into 33 parts. Shown here in the neutral pose.

is a globally differentiable model that can be fit to data with gradient-based techniques.

To define a GLoSS model we need the following: a 3D template mesh of an animal with the desired polygon count, its segmentation into parts, skinning weights, and an animation sequence. To define the mesh topology, we use a 3D mesh of a lioness downloaded from the Turbosquid website. The mesh is rigged and skinning weights are defined. We manually segment the mesh into $N = 33$ parts (Fig. 6-3) and make it symmetric along its sagittal plane.

We now summarize the GLoSS parametrization. Let i be a part index, $i \in (1 \cdots N)$. The model variables are: part location $\mathbf{l}_i \in \mathbb{R}^{3\times 1}$, part absolute 3D rotation $\mathbf{r}_i \in \mathbb{R}^{3\times 1}$, expressed as a Rodrigues vector, intrinsic shape variables $\mathbf{s}_i \in \mathbb{R}^{n_s\times 1}$ and pose deformation variables $\mathbf{d}_i \in \mathbb{R}^{n_d\times 1}$. Let $\pi_i = \{\mathbf{l}_i, \mathbf{r}_i, \mathbf{s}_i, \mathbf{d}_i\}$ be the set of variables for part i and $\Pi = \{\mathbf{l}, \mathbf{r}, \mathbf{s}, \mathbf{d}\}$ the set of variables for all parts. The vector of vertex coordinates, $\hat{\mathbf{p}}_i \in \mathbb{R}^{3\times n_i}$, for part i in a global reference frame is computed as:

$$\hat{\mathbf{p}}_i(\pi_i) = R(\mathbf{r}_i)\mathbf{p}_i + \mathbf{l}_i, \tag{6.1}$$

where n_i is the number of vertices in the part, and $R \in SO(3)$ is the rotation matrix obtained from $\mathbf{r}_i$. The $\mathbf{p}_i \in \mathbb{R}^{3\times n_i}$ are points in a local coordinate frame, computed as:

$$\text{vec}(\mathbf{p}_i) = \mathbf{t}_i + \mathbf{m}_{p,i} + B_{s,i}\mathbf{s}_i + B_{p,i}\mathbf{d}_i. \tag{6.2}$$

Here $\mathbf{t}_i \in \mathbb{R}^{3n_i\times 1}$ is the part template, $\mathbf{m}_{p,i} \in \mathbb{R}^{3n_i\times 1}$ is the vector of average pose displacements; $B_{s,i} \in \mathbb{R}^{3n_i\times n_s}$ is a matrix with columns representing a basis of intrinsic shape displacements, and $B_{p,i} \in \mathbb{R}^{3n_i\times n_d}$ is the matrix of pose dependent deformations. These deformation matrices are defined below.

Pose deformation space. We compute the part-based pose deformation space from examples. For this we use an animation of the lioness template using linear blend skinning (LBS). Each frame of the animation is a pose deformation sample. We perform PCA on the vertices of each part in a local coordinate frame, obtaining a vector of average pose deformations $\mathbf{m}_{p,i}$ and the basis matrix $B_{p,i}$.

Shape deformation space. We define a synthetic shape space for each body part. This space includes 7 deformations of the part template, namely scale, scale along x, scale along y, scale along z, and three stretch deformations that are defined as follows. Stretch for x does not modify the x coordinate of the template points, while it scales the y and z coordinates in proportion to the value of x. Similarly we define the stretch for y and z. This defines a simple analytic deformation space for each part. We model the distribution of the shape coefficients as a Gaussian distribution with zero mean and diagonal covariance, where we set the variance of each dimension arbitrarily.

6.4 Initial Registration

The initial registration of the template to the scans is performed in two steps. First, we optimize the GLoSS model with a gradient-based method. This brings the model close to the scan. Then, we perform a model-free registration of the mesh vertices to the scan using As-Rigid-As-Possible (ARAP) regularization [195] to capture the fine details.

6.4.1 GLoSS-based registration

To fit GLoSS to a scan, we minimize the following objective:

$$E(\Pi) = E_m(\mathbf{d}, \mathbf{s}) + E_{stitch}(\Pi) + E_{curv}(\Pi) + E_{data}(\Pi) + E_{pose}(\mathbf{r}), \tag{6.3}$$

where

$$E_m(\mathbf{d}, \mathbf{s}) = k_{sm} E_{sm}(\mathbf{s}) + k_s \sum_{i=1}^{N} E_s(\mathbf{s}_i) + k_d \sum_{i=1}^{N} E_d(\mathbf{d}_i)$$

is a model term, where E_s is the squared Mahalanobis distance from the synthetic shape distribution and E_d is a squared $L2$ norm. The term E_{sm} represents the constraint that symmetric parts should have similar shape deformations. We impose similarity between left and right limbs, front and back paws, and sections of the torso. This last constraint favors sections of the torso to have similar length.

The stitching term E_{stitch} is the sum of squared distances of the corresponding points at the interfaces between parts (cf. [243]). Let C_{ij} be the set of vertex-vertex correspondences between part i and part j. Then $E_{stitch}(\Pi) =$

$$k_{st} \sum_{(i,j) \in \mathcal{C}} \sum_{(k,l) \in C_{ij}} \|\hat{\mathbf{p}}_{i,k}(\pi_i) - \hat{\mathbf{p}}_{j,l}(\pi_j)\|^2, \tag{6.4}$$

where $\mathcal{C}$ is the set of part connections. Minimizing this term favors connected parts.

The data term is defined as: $E_{data}(\Pi) =$

$$k_{kp} E_{kp}(\Pi) + k_{m2s} E_{m2s}(\Pi) + k_{s2m} E_{s2m}(\Pi), \tag{6.5}$$

where E_{m2s} and E_{s2m} are distances from the model to the scan and from the scan to the model, respectively:

$$E_{m2s}(\Pi) = \sum_{i=1}^{N} \sum_{k=1}^{n_i} \rho(\min_{\mathbf{s}\in\mathcal{S}} \|\hat{\mathbf{p}}_{i,k}(\pi_i) - \mathbf{s}\|^2), \tag{6.6}$$

$$E_{s2m}(\Pi) = \sum_{l=1}^{S} \rho(\min_{\hat{\mathbf{p}}} \|\hat{\mathbf{p}}(\Pi) - \mathbf{s}_l\|^2), \tag{6.7}$$

where $\mathcal{S}$ is the set of S scan vertices and ρ is the Geman-McClure robust error function [79]. The term $E_{kp}(\Pi)$ is a term for matching model keypoints with scan keypoints, and is defined as the sum of squared distances between corresponding keypoints. This term is important to enable matching between extremely different animal shapes.

The curvature term favors parts that have a similar pairwise relationship as those in the template; $E_{curv}(\Pi) =$

$$k_c \sum_{(i,j)\in\mathcal{C}} \sum_{(k,l)\in C_{ij}} \left| \|\hat{\mathbf{n}}_{i,k}(\pi_i) - \hat{\mathbf{n}}_{j,l}(\pi_j)\|^2 - \|\hat{\mathbf{n}}^{(t)}_{i,k} - \hat{\mathbf{n}}^{(t)}_{j,l}\|^2 \right|,$$

where $\hat{\mathbf{n}}_i$ and $\hat{\mathbf{n}}_j$ are vectors of vertex normals on part i and part j, respectively. Analogous quantities on the template are denoted with a superscript (t). Lastly, E_{pose} is a pose prior on the tail parts learned from animations of the tail. The values of the energy weights are manually defined and kept constant for all the toys.

We initialize the registration of each scan by aligning the model in neutral pose to the scan based on the median value of their vertices. Given this, we minimize Eq. 6.3 using the Chumpy auto-differentiation package [3]. Doing so aligns the lioness GLoSS model to all the toy scans. Figure 6-4a-c shows an example of fitting of GLoSS (colored) to a scan (white), and Fig. 6-5 (first and second column) shows some of the obtained registrations. To compare the GLoSS-based registration with SP we computed SP registrations for the big cats family.

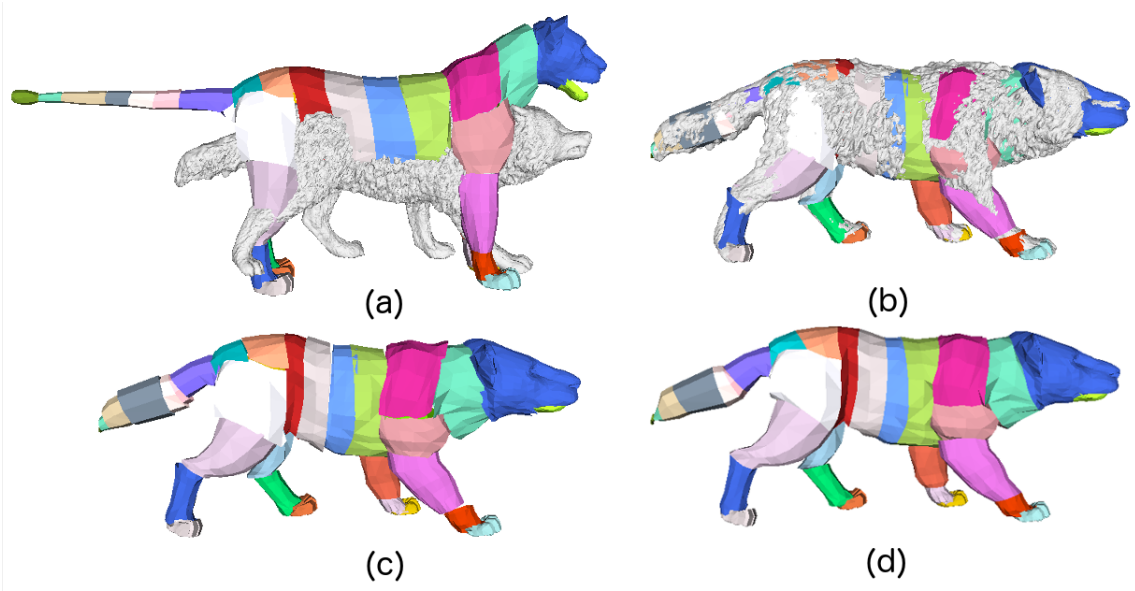


Figure 6-4: **GLoSS fitting.** (a) Initial template and scan. (b) GLoSS fit to scan. (c) GLoSS model showing the parts. (d) Merged mesh with global topology obtained by removing the duplicated vertices at the part interfaces.

We obtain an average scan-to-mesh distance of $4.39(\sigma = 1.66)$ for SP, and $3.22(\sigma = 1.34)$ for GLoSS.

6.4.2 ARAP-based refinement

The GLoSS model gives a good initial registration. Given this, we turn each GLoSS mesh from its part-based topology into a global topology where interface points are not duplicated (Fig. 6-4d). We then further align the vertices $\mathbf{v}$ to the scans by minimizing an energy function defined by a data term equal to Eq. 6.5 and an As-Rigid-As-Possible (ARAP) regularization term [195]:

$$E(\mathbf{v}) = E_{data}(\mathbf{v}) + E_{arap}(\mathbf{v}). \tag{6.8}$$

This model-free optimization brings the mesh vertices closer to the scan and therefore more accurately captures the shape of the animal (see Fig. 6-5).

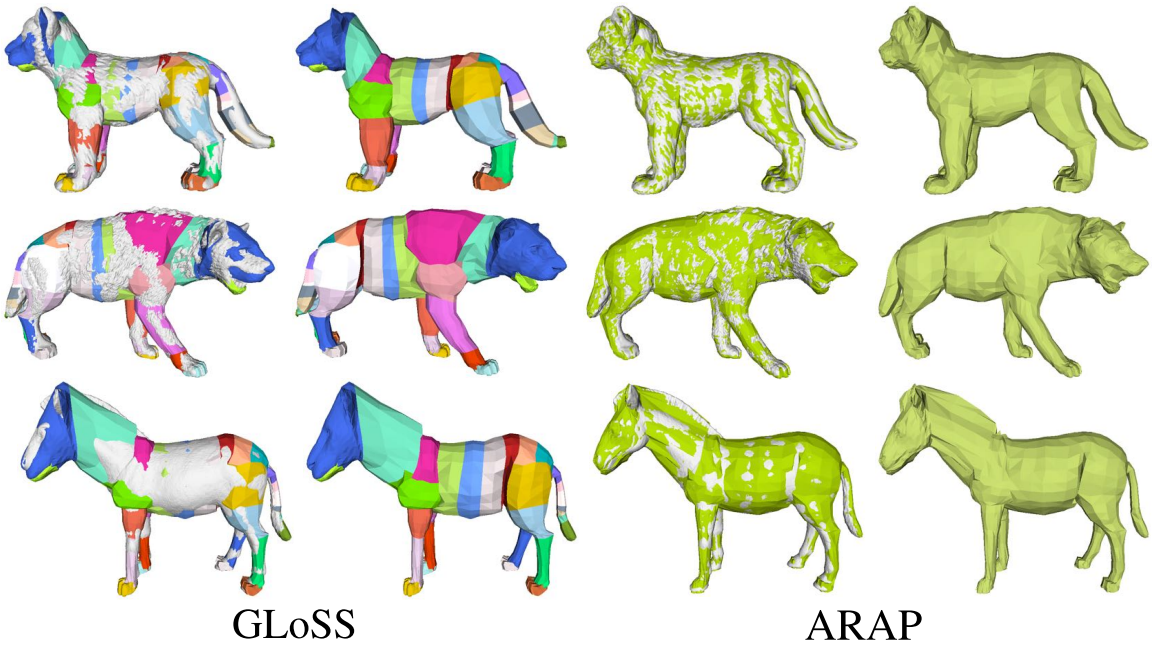


Figure 6-5: **Registration results**. Comparing GLoSS (left) with the ARAP refinement (right). The fit to the scan is much tighter after refinement.

6.5 Skinned Multi-Animal Linear Model

The above registrations are now sufficiently accurate to create a first shape model, which we refine further below to produce the full SMAL model.

Pose normalization Given the pose estimated with GLoSS, we bring all the registered templates into the same neutral pose using LBS. The resulting meshes are not symmetric. This is due to various reasons: inaccurate pose estimation, limitations of linear-blend-skinning, the toys may not be symmetric, and pose differences across sides of the body create different deformations. We do not want to learn this asymmetry. To address this we perform an averaging of the vertices after we have mirrored the mesh to obtain the registrations in the neutral pose (Fig. 6-6). Also, the fact that mouths are sometimes open and other times closed presents a challenge for registration, as inside mouth points are not observed in the scan when the animal has a closed mouth. To address this, palate and tongue points in the

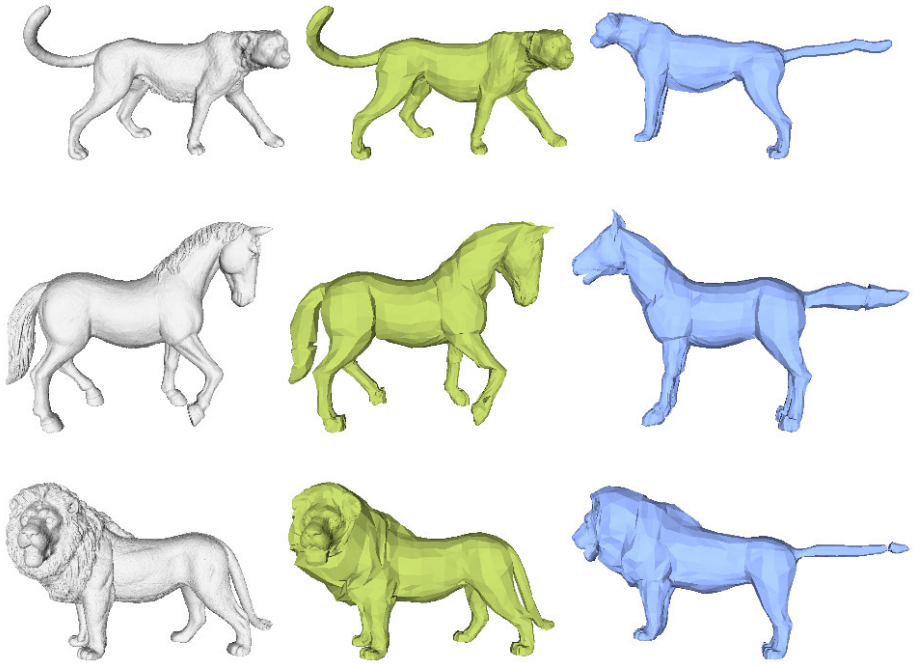


Figure 6-6: **Pose normalization.** We show the 3D scan in gray, ARAP registration in green, and pose-normalized scan in blue.

registration are regressed from the mouth points using a simple linear model learned from the template. Finally we smooth the meshes with Laplacian smoothing.

Shape model Pose normalization removes the non-linear effects of part rotations on the vertices. In the neutral pose we can thus model the statistics of the shape variation in a Euclidean space. We compute the mean shape and the principal components, which capture shape differences between the animals.

SMAL. The SMAL model is a function $M(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\gamma})$ of shape $\boldsymbol{\beta}$, pose $\boldsymbol{\theta}$ and translation $\boldsymbol{\gamma}$. $\boldsymbol{\beta}$ is a vector of the coefficients of the learned PCA shape space, $\boldsymbol{\theta} \in \mathbb{R}^{3N} = \{\mathbf{r}_i\}_{i=1}^{N}$ is the relative rotation of the $N = 33$ joints in the kinematic tree, and $\boldsymbol{\gamma}$ is the global translation applied to the root joint. Analogous to SMPL, the SMAL function returns a 3D mesh, where the template model is shaped by $\boldsymbol{\beta}$, articulated by $\boldsymbol{\theta}$ through LBS, and shifted by $\boldsymbol{\gamma}$.

Fitting To fit SMAL to scans we minimize the objective:

$$E(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\gamma}) = E_{pose}(\boldsymbol{\theta}) + E_s(\boldsymbol{\beta}) + E_{data}(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\gamma}), \tag{6.9}$$

where $E_{pose}(\boldsymbol{\theta})$ and $E_s(\boldsymbol{\beta})$ are squared Mahalanobis distances from prior distributions for pose and shape, respectively. $E_{data}(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\gamma})$ is defined as in Eq. 6.5 but over the SMAL model. For optimization we use Chumpy [3].

Co-registration To refine the registrations and the SMAL model further, we then perform co-registration [97]. The key idea is to first perform a SMAL model optimization to align the current model to the scans, and then run a model-free step where we *couple*, or regularize, the model-free registration to the current SMAL model by adding a coupling term to Eq. 6.8:

$$E_{coup}(\mathbf{v}) = k_o \sum_{i=1}^{V} |\mathbf{v}_i^0 - \mathbf{v}_i|, \tag{6.10}$$

where V is the number of vertices in the template, $\mathbf{v}_i^0$ is vertex i of the model fit to the scan, and the $\mathbf{v}_i$ are the coupled mesh vertices being optimized. During co-registration we use a shape space with 30 dimensions. We perform 4 iterations of registration and model building and observe the registration errors decrease and converge.

With the registrations to the toys in the last iteration we learn the shape space of our final SMAL model.

6.5.1 Animal shape space

After refining with co-registration, the final principal components are visualized in Fig. 6-8. The global SMAL shape space captures the shape variability of animals across different

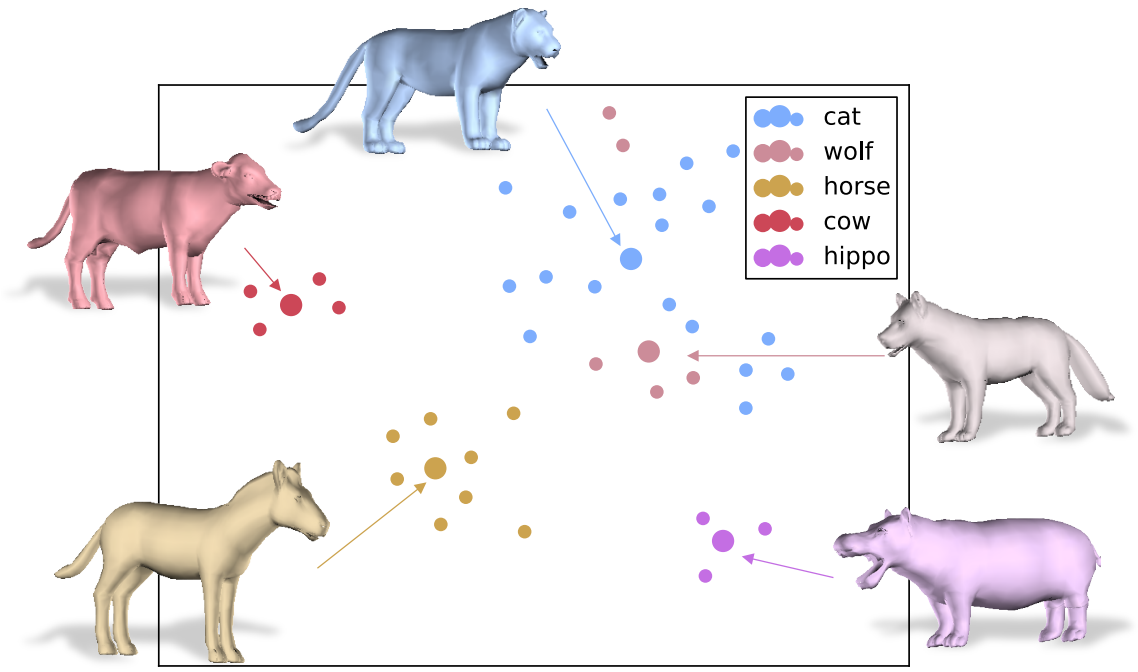


Figure 6-7: Visualization (using t-SNE [149]) of different animal families using 8 PCs. Large dots indicate the mean of the PCA coefficients for each family.

families. The first component captures scale differences; our training set includes adult and young animals. The learned space nicely separates shape characteristics of animal families. This is illustrated in Fig. 6-7 with a t-SNE visualization [149] of the first 8 dimensions of the PCA coefficients in the training set. The meshes correspond to the mean shape for each family. We also define family-specific shape models by computing a Gaussian over the PCA coefficients of the class. We compare generic and family specific models below.

6.6 Fitting Animals to Images

We now fit the SMAL model, $M(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\gamma})$, to image cues by optimizing the shape and pose parameters. We fit the model to a combination of 2D keypoints and 2D silhouettes, both manually extracted, as in previous work [50, 116].

We denote $\Pi(\cdot; f)$ as the perspective camera projection with focal length f, where $\Pi(\mathbf{v}_i; f)$

is the projection of the i'th vertex onto the image plane and $\Pi(M; f) = \hat{S}$ is the projected model silhouette. We assume an identity camera placed at the origin and that the global rotation of the 3D mesh is defined by the rotation of the root joint.

To fit SMAL to an image, we formulate an objective function and minimize it with respect to $\Theta = \{\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\gamma}, f\}$. The function is a sum of the keypoint and silhouette reprojection errors, a shape prior, and two pose priors, $E(\Theta) =$

$$E_{kp}(\Theta; \mathbf{x}) + E_{silh}(\Theta; S) + E_{\boldsymbol{\beta}}(\boldsymbol{\beta}) + E_{\boldsymbol{\theta}}(\boldsymbol{\theta}) + E_{lim}(\boldsymbol{\theta}) \tag{6.11}$$

Each energy term is weighted by a hyper-parameter defining their importance.

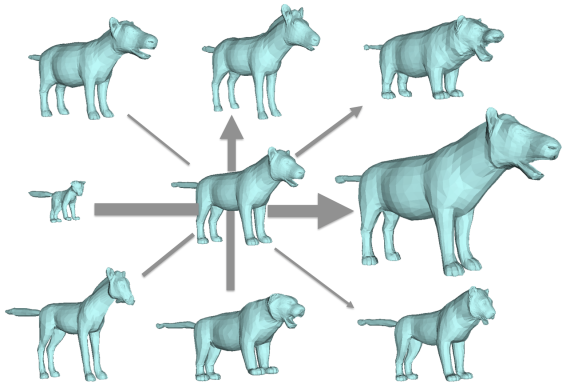


Figure 6-8: **PCA space.** First 4 principal components. Mean shape is in the center. The width of the arrow represents the order of the components. We visualise deviations of ± 2 std.

Keypoint reprojection. See [2] for a definition of keypoints which include surface points and joints. Since keypoints may be ambiguous, we assign a set of up to four vertices to represent each model keypoint and take the average of their projection to match the target 2D keypoint. Specifically for the k'th keypoint, let $\mathbf{x}$ be the labeled 2D keypoint and $\{\mathbf{v}_{k_j}\}_{j=1}^{k_m}$ be the assigned set of vertices, then

$$E_{kp}(\Theta) = \sum_k \rho(||\mathbf{x} - \frac{1}{|k_m|} \sum_{j=1}^{|k_m|} \Pi(\mathbf{v}_{k_j}; \Theta)||_2), \tag{6.12}$$

where ρ is the Geman-McClure robust error function [79].

Silhouette reprojection. We encourage silhouette coverage and consistency similar to [117, 130, 221] using a bi-directional distance:

$$E_{silh}(\Theta) = \sum_{\mathbf{x} \in \hat{S}} \mathcal{D}_{\mathcal{S}}(\mathbf{x}) + \sum_{\mathbf{x} \in S} \rho(\min_{\hat{\mathbf{x}} \in \hat{S}} ||\mathbf{x} - \hat{\mathbf{x}}||_2), \tag{6.13}$$

where S is the ground truth silhouette and $\mathcal{D}_S$ is its L2 distance transform field such that if point $\mathbf{x}$ is inside the silhouette, $\mathcal{D}_S(\mathbf{x}) = 0$. Since the silhouette terms have small basins of attraction we optimize the term over multiple scales in a coarse-to-fine manner.

Shape prior. We encourage $\boldsymbol{\beta}$ to be close to the prior distribution of shape coefficients by defining $E_{\boldsymbol{\beta}}$ to be the squared Mahalanobis distance with zero mean and variance given by the PCA eigenvalues. When the animal family is known, we can make our fits more specific by using the mean and covariance of training samples of the particular family.

Pose priors. $E_{\boldsymbol{\theta}}$ is also defined as the squared Mahalanobis distance using the mean and covariance of the poses across all the training samples and a walking sequence. To make the pose prior symmetric, we double the training data by reflecting the poses along the template's sagittal plane. Since we do not have many examples, we further constrain the pose with limit bounds:

$$E_{lim}(\boldsymbol{\theta}) = \max(\boldsymbol{\theta} - \boldsymbol{\theta}_{\max}, 0) + \max(\boldsymbol{\theta}_{\min} - \boldsymbol{\theta}, 0). \tag{6.14}$$

$\boldsymbol{\theta}_{\max}$ and $\boldsymbol{\theta}_{\min}$ are the maximum and minimum range of values for each dimension of $\boldsymbol{\theta}$ respectively, which we define by hand. We do not limit the global rotation.

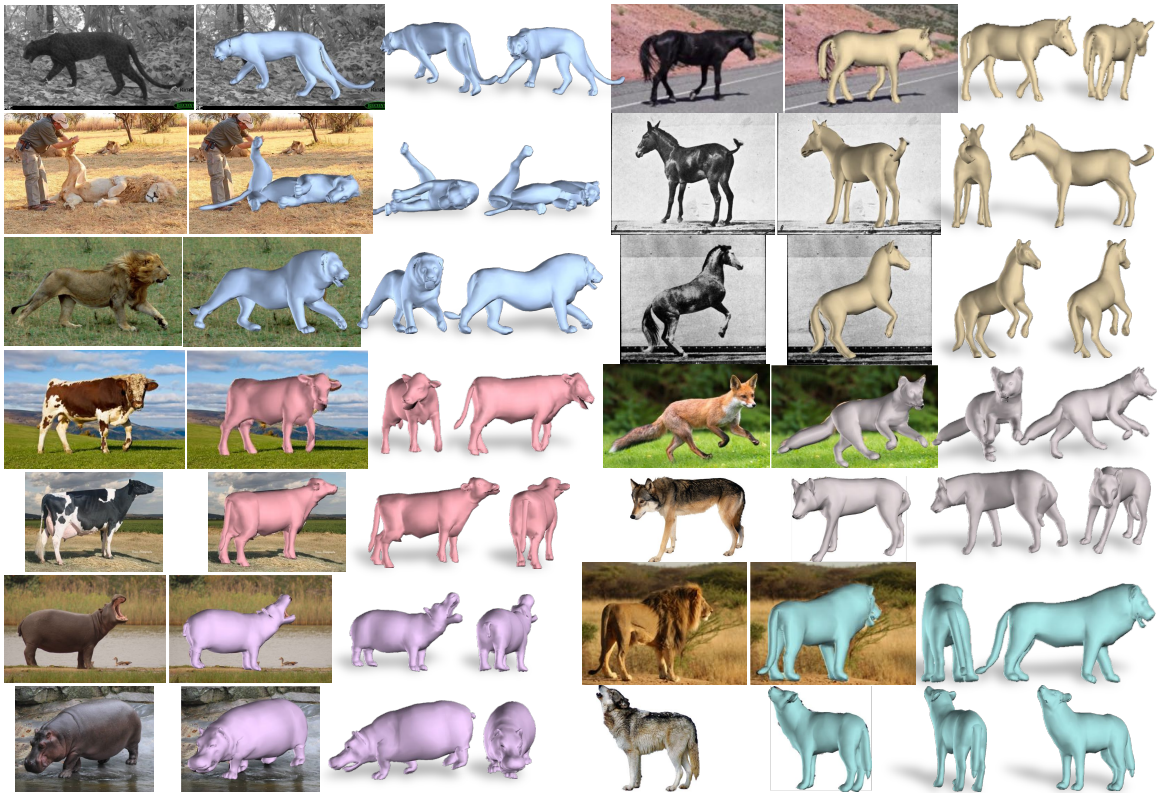


Figure 6-9: Fits to real images using manually obtained 2D points and segmentation. Colors indicate animal family. We show the input image, fit overlaid, views from $-45°$ and $45°$. All results except for those in mint colors use the animal specific shape prior. The SMAL model, learned form toy figurines, generalizes to real animal shapes.

Optimization. Following [34], we first initialize the depth of $\boldsymbol{\gamma}$ using the torso points. Then we solve for the global rotation $\{\boldsymbol{\theta}_i\}_{i=0}^{3}$ and $\boldsymbol{\gamma}$ using E_{kp} over points on the torso. Using these as the initialization, we solve Eq. 6.11 for the entire Θ without E_{silh}. Similar to previous methods [34, 116] we employ a staged approach where the weights on pose and shape priors are gradually lowered over three stages. This helps avoid getting trapped in local optima. We then finally include the E_{silh} term and solve Eq. 6.11 starting from this initialization. Solving for the focal length is important and we regularize f by adding another term that forces $\boldsymbol{\gamma}$ to be close to its initial estimate. The entire optimization is done using OpenDR and Chumpy [3, 141]. Optimization for a single image typically takes less than a minute on a common Linux machine.

6.7 Experiments

We have shown how to learn a SMAL animal model from a small set of toy figurines. Now the question is: does this model capture the shape variation of real animals? Here we test this by fitting the model to annotated images of real animals. We fit using class specific and generic shape models, and show that the shape space generalizes to new animal families not present in training (within reason).

Data. For fitting, we use 19 semantic keypoints of [65] plus an extra point for the tail tip. Note that these keypoints differ from those used in the 3D alignment. We fit frames in the TigDog dataset, reusing their annotation, frames from the Muybridge footage, and images downloaded from the Internet. For images without annotation, we click the same 20 keypoints for all animals, which takes about one minute for each image. We also hand segmented all the images. No images were re-visited to improve their annotations and we found the model to be robust to noise in the exact location of the keypoints. The annotations and the results are accessible at [2].

Results. The model fits to real images of animals are shown in Fig. 6-1 and 6-9. The weights for each term in Eq. 6.11 are tuned by hand and held fixed for fitting *all* images. All results use the animal specific shape space except for those in mint green, which use the generic shape model. Despite being trained on scans of toys, our model generalizes to images of real animals, capturing their shape well. Variability in animal families with extreme shape characteristics (*e.g.* lion manes, skinny horse legs, hippo faces) are modeled well. Both the generic and class-specific models capture the shape of real animals well.

Similar to the case of humans [34], our main failures are due to inherent depth ambiguity, both in global rotation and pose (Fig 6-10). In Fig. 6-11 we show the results of fitting the

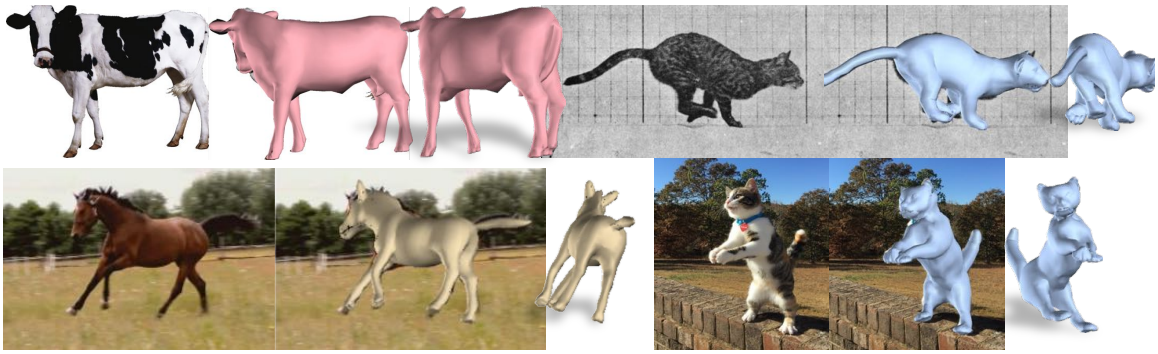


Figure 6-10: Failure examples due to depth ambiguity in pose and global rotation.

generic shape model to classes of animals not seen in the training set: boar, donkey, sheep and pigs. While characteristic shape properties such as the pig snout cannot be exactly captured, these fits suggest that the learned PCA space can generalize to new animals within a range of quadrupeds.

6.8 Conclusions

Human shape modeling has a long history, while animal modeling is in its infancy. We have made small steps towards making the building of animal models practical. We showed that starting with toys, we can learn a model that generalizes to images of real animals as well as to types of animals not seen during training. This gives a procedure for building richer models from more animals and more scans. While we have shown that toys are a good starting point, we would clearly like a much richer model. In particular, the limited set of toys and poses means that it is difficult to learn a rich model of pose-dependent deformation. Each toy is seen in only one pose. This is in contrast to human training data where the same person is observed in many poses, allowing variation with pose to be isolated from identity. The small number of training poses also imply that we cannot learn a multi-modal pose prior used in Chapter 3. For that, a key future direction is to actually exploit 2D image

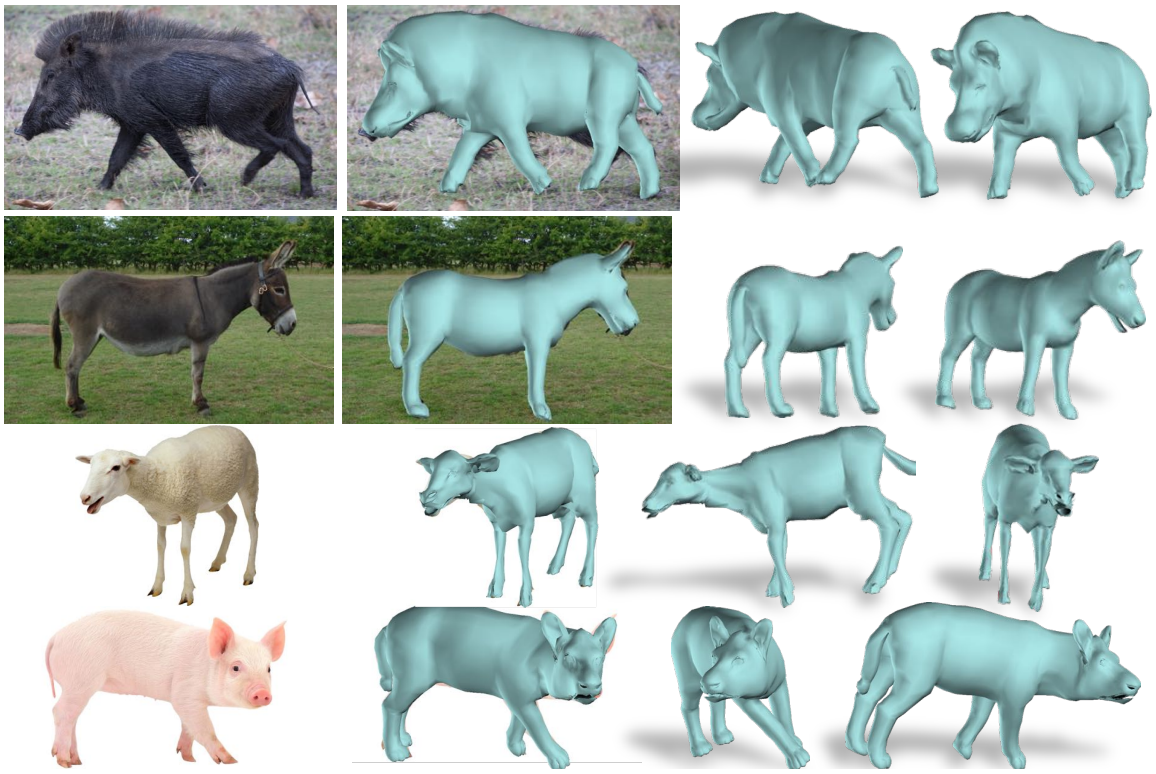


Figure 6-11: Generalization of SMAL to animal species not present in the training set.

information to improve our model. Our fits can provide a starting point from which to learn richer deformations to explain 2D image evidence.

In this work we have focused on a limited set of quadrupeds. A key issue is dealing with varying numbers of parts (e.g. horns, tusks, trunks) and parts of widely different shape (e.g. elephant ears). Moving beyond the class of animals here will involve creating a vocabulary of reusable shape parts and new ways of composing them.

Chapter 7

Concluding Remarks

This thesis develops methods for single-view 3D reconstruction of people and animals. We propose a working solution for recovering a 3D mesh of a human body from just a single image, and explore how to apply similar practices to animals. Doing so reveals novel challenges whose common thread is the lack of specialized data. We propose three different methods to deal with these challenges. We summarize the major contribution of this thesis below:

- We have proposed the first fully automatic method for estimating 3D body pose *and* shape from 2D joints. The code and results are available for research purposes [1].

- We reduce the ambiguity of the problem by introducing a differentiable interpenetration term and pose priors. We test our methods quantitatively over standard datasets and qualitatively on unconstrained images of people on the Internet with complex poses. Our output is realistic and can be immediately animated in standard graphics pipelines.

- We propose a deep learning architecture, WarpNet, that learns a class-specific model of 2D deformation for establishing correspondences between two input images across sub-category and pose variations. A novel exemplar-driven mechanism is introduced to

train the network without requiring any human provided keypoint annotations.

- We propose a method that learns how an animal category deforms using a set of user-annotated 2D images and a reference 3D mesh. To our knowledge, this is the first approach that learns a model of 3D pose deformation from 2D images. Our formulation is based on a novel bounded deformation energy where both the bounds and the deformation can be solved jointly in a sequence of convex optimization problems.

- We describe a method to create *Skinned Multi-Animal Linear* model (SMAL), a realistic 3D model of several quadruped animals that can be fit to 2D image observations. To our knowledge no other methods learn a 3D shape space that spans multiple animals. Despite being trained on toy scans, our model generalizes to images of real animals.

The solutions explored in this thesis are pieces of the big puzzle for getting a system that can do fully automatic single-view 3D reconstruction of animals. Aside from the detailed future directions discussed in each chapter, below are several overarching directions for future work.

Looking at the actual image a.k.a making it end-to-end: One caveat of the presented model-fitting approach is that the algorithm only looks at the supplied 2D keypoints or silhouettes and ignores the rest of the image. A natural next step is to combine the bottom up estimation and the top down inference step inside a single deep learning framework and train it end-to-end. The main challenge is that 3D annotation of an unconstrained image is very difficult to get, even with a lot of resources. Existing methods to obtain ground truth 3D information requires motion capture or scanning in a lab environment, which creates a domain shift problem between lab images and Internet quality images. Possible solutions are image synthesis by rendering realistic images [213, 87] and domain adaptation techniques

[78].

Time and motion: the missing fourth dimension: Since much of the focus was on the ability to see the 3D from a single view, none of the presented work deal with time and motion. When a video of a moving object is available, a nice property is that we can assume that there is only one 3D shape. A temporal smoothness assumption is another constraint that could be added. Also from the 3D modeling perspective, these properties make time an important signal that could be used for learning a more powerful animal 3D models. Future directions also include modeling the dynamics of human action and animal movement.

The emperor's new clothes: Another aspect that we did not mention in the thesis was texturing the 3D models. All of our models are "naked"! A possible direction with textures is to treat the texture map as a latent variable for model fitting. Blanz and Vetter [32] show this for faces, but it's not clear if appearances of humans and animals can be modeled well with a low-dimensional model. Filling in the texture of the "unobserved" region of the 3D model is another interesting problem. This is related to the texture synthesis problem [67]. Recent advances that use generative adversarial networks [81] for synthesizing images suggest an effective solution. This is also related to the problem of clothing and fur. Most human shape models have ignored clothing. Analogously, here we do not model fur. Future work should consider explicit models of fur and how it deforms with pose.

In all, there are a lot of interesting directions and applications in this domain. We hope this thesis motivates more research in 3D reconstruction of people and animals.

Appendices

A Computing Thin-Plate Spline Coefficients

Given a regular grid points $\{\boldsymbol{x}_i\}$ and deformed grid points $\{\boldsymbol{x}'_i\}$, $i = 1, \ldots, K^2$, the thin-plate spline (TPS) transformation from the regular grid coordinate frame to the deformed grid coordinate frame for the x-coordinates is given by:

$$T_{\theta_x}(\boldsymbol{x}) = \sum_{j=0}^{3} a_j^x \phi_j(\boldsymbol{x}) + \sum_{i=1}^{K^2} w_i^x U(||\boldsymbol{x}, \boldsymbol{x}_i||), \tag{1}$$
$$\text{s.t.} \quad \sum_{i=1}^{K^2} w_i^x = 0, \quad \sum_{j=1}^{2} \sum_{i=1}^{K^2} w_i^x x_j = 0,$$

where $\phi_0 = 1$, $\phi_j(\boldsymbol{x}_i) = x_j$, $U(r) = r^2 \log r^2$. A similar transformation may be expressed for the y-coordinate, denoted $T_{\theta_y}(\boldsymbol{x})$, with coefficients $\boldsymbol{w}^y$ and $\boldsymbol{a}^y$. The final transformation is $T_\theta(\boldsymbol{x}) = [T_{\theta_x}(\boldsymbol{x}), T_{\theta_y}(\boldsymbol{x})]$. With the interpolation conditions $T_\theta(\boldsymbol{x}_i) = \boldsymbol{x}'_i$, we can write the TPS coefficients $\theta = \begin{pmatrix} \boldsymbol{w}^x & \boldsymbol{w}^y \\ \boldsymbol{a}^x & \boldsymbol{a}^y \end{pmatrix}$ as the solution to a system of linear equations:

$$L\theta = \begin{pmatrix} \boldsymbol{x}' \\ 0 \end{pmatrix}, \tag{2}$$

where $L = \begin{pmatrix} K & P \\ P^T & 0 \end{pmatrix}$, $K_{ij} = U(||\boldsymbol{x}_i - \boldsymbol{x}_j||)$ and row i of P is $(1, x_i, y_i)$. As discussed in [35], L is non-singular, invertible and only needs to be computed once since the regular grid $\boldsymbol{x}$ is fixed for our application. Thus, computing the TPS coefficients from a deformed grid is a linear operation $\theta = L^{-1}\boldsymbol{x}'_i$ with weights, L^{-1}, computed once in the beginning of the training.

B Writing $\frac{1}{(1+\epsilon+s)} \leq \gamma$ as a Second-Order Cone

The constraint $\frac{1}{(1+\epsilon+s)} \leq \gamma$ can be realized as a second-order cone (SOCP) by using the method of rotated second-order cone [39]. Specifically, as discussed in Alizadeh and Goldfarb [11], a constraint of the form $\mathbf{w}^\top \mathbf{w} \leq xy$, where $x \geq 0$, $y \geq 0$, $\mathbf{w} \in \mathrm{R}^n$, is equivalent to SOCP

$$\left\| \begin{pmatrix} 2\mathbf{w} \\ x - y \end{pmatrix} \right\| \leq x + y. \tag{3}$$

In our formulation, $x = (1 + \epsilon + s)$, $y = \gamma$, and $\mathbf{w} = 1$. This can also be derived using the identity

$$xy = \frac{1}{4}((x + y)^2 - (x - y)^2), \tag{4}$$

where

$$\begin{aligned} 1 &\leq xy \\ 1 &\leq \frac{1}{4}((x + y)^2 - (x - y)^2) \\ 4 + (x - y)^2 &\leq (x + y)^2 \\ \sqrt{4 + (x - y)^2} &\leq (x + y) \\ \left\| \begin{pmatrix} 2 \\ x - y \end{pmatrix} \right\| &\leq (x + y). \end{aligned}$$

Bibliography

[1] http://smplify.is.tue.mpg.de.

[2] http://smal.is.tue.mpg.de.

[3] http://chumpy.org.

[4] Digital life. http://www.digitallife3d.com/, Accessed November 12, 2016.

[5] http://mocap.cs.cmu.edu.

[6] A.Dosovitskiy, J.T.Springenberg, M.Riedmiller, and T.Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *NIPS*, 2014.

[7] Ankur Agarwal and Bill Triggs. Recovering 3d human pose from monocular images. *IEEE transactions on pattern analysis and machine intelligence*, 28(1):44–58, 2006.

[8] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *ICCV*, 2015.

[9] Ijaz Akhter and Michael J. Black. Pose-conditioned joint angle limits for 3D human pose reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1446–1455, 2015.

[10] Marc Alexa, Daniel Cohen-Or, and David Levin. As-rigid-as-possible shape interpolation. In *ACM SIGGRAPH*, 2000.

[11] Farid Alizadeh and Donald Goldfarb. Second-order cone programming. *Mathematical programming*, 95(1):3–51, 2003.

[12] Brett Allen, Brian Curless, and Zoran Popović. Articulated body deformation from range scan data. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 612–619. ACM, 2002.

[13] Brett Allen, Brian Curless, and Zoran Popović. The space of human body shapes: reconstruction and parameterization from range scans. In *ACM transactions on graphics (TOG)*, volume 22, pages 587–594. ACM, 2003.

[14] Brett Allen, Brian Curless, Zoran Popović, and Aaron Hertzmann. Learning a correlated model of identity and pose-dependent body shape variation for real-time synthesis. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '06, pages 147–156. Eurographics Association, 2006.

[15] Padmanabhan Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2(3):283–310, 1989.

[16] Erling D Andersen and Knud D Andersen. The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In *High performance optimization*, pages 197–232. Springer, 2000.

[17] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. SCAPE: Shape completion and animation of people. *ToG*, 24(3):408–416, 2005.

[18] Dragomir Anguelov, Daphne Koller, Hoi-Cheung Pang, Praveen Srinivasan, and Sebastian Thrun. Recovering articulated object models from 3d range data. In *UAI*, pages 18–26, 2004.

[19] Relja Arandjelovic and Andrew Zisserman. All about vlad. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1578–1585, 2013.

[20] Ruzena Bajcsy and Franc Solina. Three dimensional object representation revisited. In *First International Conference on Computer Vision*. The Computer Society of the IEEE, 1987.

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

[22] Luca Ballan, Aparna Taneja, Juergen Gall, Luc Van Gool, and Marc Pollefeys. Motion capture of hands in action using discriminative salient points. In *ECCV*, October 2012.

[23] Sid Ying-Ze Bao, Manmohan Chandraker, Yuanqing Lin, and Silvio Savarese. Dense object reconstruction with semantic priors. In *CVPR*. IEEE, 2013.

[24] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3), August 2009.

[25] Connelly Barnes, Eli Shechtman, Dan Goldman, and Adam Finkelstein. The generalized patchmatch correspondence algorithm. *Computer Vision–ECCV 2010*, pages 29–43, 2010.

[26] C. Barron and I.A. Kakadiaris. Estimating anthropometry and pose from a single uncalibrated image. *Computer Vision and Image Understanding, CVIU*, 81(3):269–284, 2001.

[27] Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *IEEE transactions on pattern analysis and machine intelligence*, 37(8):1670–1687, 2015.

[28] Peter N Belhumeur, David W Jacobs, David J Kriegman, and Narendra Kumar. Localizing parts of faces using a consensus of exemplars. *IEEE Trans. Pattern Anal. Mach. Intell*, 35(12):2930–2940, 2013.

[29] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(4):509–522, 2002.

[30] Thomas Berg and Peter N. Belhumeur. How do you tell a blackbird from a crow? In *ICCV*, 2013.

[31] Andrew Blake and Heinrich Bulthoff. Shape from specularities: Computation and psychophysics. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 331(1260):237–252, 1991.

[32] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *ACM SIGGRAPH*, pages 187–194, 1999.

[33] Liefeng Bo and Cristian Sminchisescu. Twin Gaussian processes for structured prediction. *International Journal of Computer Vision, IJCV*, 87(1-2):28–52, 2010.

[34] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *European Conf. on Computer Vision (ECCV)*, October 2016.

[35] Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, pages 567–585, 1989.

[36] Mario Botsch, Mark Pauly, Markus H Gross, and Leif Kobbelt. Primo: coupled prisms for intuitive surface modeling. In *Symposium on Geometry Processing*, pages 11–20, 2006.

[37] Mario Botsch and Olga Sorkine. On linear variational surface deformation methods. *Visualization and Computer Graphics, IEEE Transactions on*, 14(1):213–230, 2008.

[38] Lubomir Bourdev and Jitendra Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1365–1372. IEEE, 2009.

[39] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[40] Christoph Bregler, Aaron Hertzmann, and Henning Biermann. Recovering non-rigid 3d shape from image streams. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 690–696. IEEE, 2000.

[41] Hilton Bristow, Jack Valmadre, and Simon Lucey. Dense semantic correspondence where every pixel is a classifier. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4024–4031, 2015.

[42] A. Bronstein, M. Bronstein, and R. Kimmel. *Numerical Geometry of Non-Rigid Shapes*. Springer Publishing Company, 2008.

[43] Alexander M. Bronstein, Michael M. Bronstein, and Ron Kimmel. Calculus of nonrigid surfaces for geometry and texture manipulation. *IEEE Transactions on Visualization and Computer Graphics*, 13(5), 2007.

[44] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008.

[45] Chen Cao, Yanlin Weng, Stephen Lin, and Kun Zhou. 3d shape regression for real-time facial animation. *ACM Transactions on Graphics (TOG)*, 32(4):41, 2013.

[46] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. Face alignment by explicit shape regression. In *CVPR 2012*, January 2012.

[47] Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4733–4742, 2016.

[48] João Carreira, Abhishek Kar, Shubham Tulsiani, and Jitendra Malik. Virtual view networks for object reconstruction. In *CVPR*. IEEE, 2015.

[49] Thomas J. Cashman and Andrew W. Fitzgibbon. Forms: Flexible object reconstruction from multiple silhouettes. http://forms.codeplex.com/.

[50] Thomas J. Cashman and Andrew W. Fitzgibbon. What shape are dolphins? building 3D morphable models from 2D images. *IEEE Trans. Pattern Anal. Mach. Intell*, 35(1):232–244, 2013.

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

[52] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014.

[53] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015.

[54] Xianjie Chen and Alan L Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *Advances in Neural Information Processing Systems*, pages 1736–1744, 2014.

[55] Yinpeng Chen, Zicheng Liu, and Zhengyou Zhang. Tensor-based human body modeling. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 105–112, June 2013.

[56] Yu Chen, Tae-Kyun Kim, and Roberto Cipolla. Inferring 3D shapes and deformations from single views. In *European Conference on Computer Vision, ECCV*, pages 300–313, 2010.

[57] S. Chopra, R. Hadsell, and Y. L. Le Cun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, pages I: 539–546, 2005.

[58] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European Conference on Computer Vision*, pages 628–644. Springer, 2016.

[59] Matthew T Cook and Arvin Agah. A survey of sketch-based 3-d modeling techniques. *Interacting with computers*, 21(3):201–211, 2009.

[60] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995.

[61] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[62] Edilson de Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. Performance capture from sparse multi-view video. *ACM SIGGRAPH*, 27(3), 2008.

[63] Edilson de Aguiar, Christian Theobalt, Sebastian Thrun, and Hans-Peter Seidel. Automatic conversion of mesh animations into skeleton-based animations. *Computer Graphics Forum (Proc. Eurographics EG'08)*, 27(2):389–397, 2008.

[64] Martin de La Gorce, Nikos Paragios, and David J Fleet. Model-based hand tracking with texture, shading and self-occlusions. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference On*, pages 1–8. IEEE, 2008.

[65] Luca Del Pero, Susanna Ricco, Rahul Sukthankar, and Vittorio Ferrari. Behavior discovery and alignment of articulated object classes from unstructured video. *International Journal of Computer Vision*, 121(2):303–325, 2017.

[66] Kevin G Der, Robert W Sumner, and Jovan Popović. Inverse kinematics for reduced deformable models. In *ACM Transactions on Graphics*. ACM, 2006.

[67] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1033–1038. IEEE, 1999.

[68] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[69] C. Ericson. *Real-time collision detection*. The Morgan Kaufmann Series in Interactive 3-D Technology, 2004.

[70] Irfan Essa, Sumit Basu, Trevor Darrell, and Alex Pentland. Modeling, tracking and interactive animation of faces and heads//using input from video. In *Computer Animation'96. Proceedings*, pages 68–79. IEEE, 1996.

[71] Xiaochuan Fan, Kang Zheng, Youjie Zhou, and Song Wang. Pose locality constrained representation for 3D human pose reconstruction. In *European Conference on Computer Vision, ECCV*, pages 174–188, 2014.

[72] Ryan Farrell, Om Oza, Ning Zhang, Vlad I Morariu, Trevor Darrell, and Larry S Davis. Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 161–168. IEEE, 2011.

[73] Paolo Favaro and Stefano Soatto. Learning shape from defocus. In *European Conference on Computer Vision*, pages 735–745. Springer, 2002.

[74] Paolo Favaro and Stefano Soatto. A geometric approach to shape from defocus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):406–417, 2005.

[75] Laurent Favreau, Lionel Reveret, Christine Depraz, and Marie-Paule Cani. Animal gaits from video. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 277–286. Eurographics Association, 2004.

[76] David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002.

[77] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Trans. Pattern Anal. Mach. Intell*, 32(8):1362–1376, 2010.

[78] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

[79] S. Geman and D. McClure. Statistical methods for tomographic image reconstruction. *Bulletin of the International Statistical Institute*, 52(4):5–21, 1987.

[80] G. Gkioxari, B. Hariharan, R. Girshick, and J. Malik. Using k-poselets for detecting people and localizing their keypoints. In *CVPR*, 2014.

[81] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[82] D. Grest and R. Koch. Human model fitting from monocular posture images. In *Proc. VMV*, pages 665–1344, 2005.

[83] Peng Guan. *Virtual human bodies with clothing and hair: From images to animation*. PhD thesis, Brown University, Department of Computer Science, December 2012.

[84] Peng Guan, Alexander Weiss, Alexandru O. Balan, and Michael J. Black. Estimating human shape and pose from a single image. In *ICCV*, pages 1381–1388. IEEE, 2009.

[85] Nail Gumerov, Ali Zandifar, Ramani Duraiswami, and Larry Davis. Structure of applicable surfaces from single views. *Computer Vision-ECCV 2004*, pages 482–496, 2004.

[86] Abhinav Gupta, Alexei A. Efros, and Martial Hebert. *Blocks World Revisited: Image Understanding Using Qualitative Geometry and Mechanics*, pages 482–496. Springer, 2010.

[87] Ankur Handa, Viorica Patraucean, Vijay Badrinarayanan, Simon Stent, and Roberto Cipolla. Scenenet: Understanding real world indoor scenes with synthetic data. *arXiv preprint arXiv:1511.07041*, 2015.

[88] Marsha J Hannah. Computer matching of areas in stereo images. Technical report, DTIC Document, 1974.

[89] Bharath Hariharan, Jitendra Malik, and Deva Ramanan. Discriminative decorrelation for clustering and classification. *Computer Vision–ECCV 2012*, pages 459–472, 2012.

[90] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Fourth Alvey Vision Conference*, pages 147–151, 1988.

[91] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, 2004.

[92] N. Hasler, H. Ackermann, B. Rosenhahn, T. ThormÃd'hlen, and H. P. Seidel. Multilinear pose and body shape estimation of dressed subjects from image sets. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1823–1830, 2010.

[93] Nils Hasler, Carsten Stoll, Martin Sunkel, Bodo Rosenhahn, and Hans-Peter Seidel. A statistical model of human pose and body shape. *Comput. Graph. Forum*, 28(2):337–346, 2009.

[94] Nils Hasler, Thorsten Thormählen, Bodo Rosenhahn, and Hans-Peter Seidel. Learning skeletons for shape and pose. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, pages 23–30. ACM, 2010.

[95] Tal Hassner. Viewing real-world faces in 3d. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3607–3614, 2013.

[96] Tal Hassner and Ronen Basri. Example based 3d reconstruction from single 2d images. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*, pages 15–15. IEEE, 2006.

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

[98] Derek Hoiem, Alexei A Efros, and Martial Hebert. Automatic photo pop-up. *ACM transactions on graphics (TOG)*, 24(3):577–584, 2005.

[99] Berthold KP Horn. Shape from shading: A method for obtaining the shape of a smooth opaque object from one view. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA, 1970.

[100] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[101] Catalin Ionescu, Joao Carreira, and Cristian Sminchisescu. Iterated second-order label sensitive pooling for 3D human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1661–1668, 2014.

[102] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence, TPAMI*, 36(7):1325–1339, 2014.

[103] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *UIST*. ACM, 2011.

[104] David W Jacobs. *Recognizing 3-D objects using 2-D images*. PhD thesis, Massachusetts Institute of Technology, 1993.

[105] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *NIPS*, 2015.

[106] Arjun Jain, Thorsten Thormählen, Hans-Peter Seidel, and Christian Theobalt. MovieReshape: Tracking and reshaping of humans in videos. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH*, 29(5):148:1–148:10, 2010.

[107] Arjun Jain, Jonathan Tompson, Yann LeCun, and Christoph Bregler. MoDeep: A deep learning framework using motion features for human pose estimation. In *Asian Conference on Computer Vision, ACCV*, volume 9004, pages 302–315, 2015.

[108] Doug L James and Christopher D Twigg. Skinning mesh animations. In *ACM Transactions on Graphics (TOG)*, pages 399–407. ACM, 2005.

[109] Leo B Jeffcott and G Dalin. Natural rigidity of the horse's backbone. *Equine veterinary journal*, 12(3):101–108, 1980.

[110] Oliver Jesorsky, Klaus J Kirchberg, and Robert W Frischholz. Robust face detection using the hausdorff distance. In *International Conference on Audio-and Video-Based Biometric Person Authentication*, pages 90–95. Springer, 2001.

[111] H. Jiang. 3D human pose reconstruction using millions of exemplars. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1674–1677, 2010.

[112] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Proceedings of the British Machine Vision Conference*, pages 12.1–12.11, 2010.

[113] David G Jones and Jitendra Malik. Computational framework for determining stereo correspondence from a set of linear spatial filters. *Image and Vision Computing*, 10(10):699–708, 1992.

[114] David Joseph Tan, Thomas Cashman, Jonathan Taylor, Andrew Fitzgibbon, Daniel Tarlow, Sameh Khamis, Shahram Izadi, and Jamie Shotton. Fits like a glove: Rapid and reliable hand shape personalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5610–5619, 2016.

[115] Angjoo Kanazawa, David W Jacobs, and Manmohan Chandraker. Warpnet: Weakly supervised matching for single-view reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3253–3261, 2016.

[116] Angjoo Kanazawa, Shahar Kovalsky, Ronen Basri, and David Jacobs. Learning 3d deformation of animals from 2d images. *Comput. Graph. Forum, (Proc. Eurographics EG'16)*, 35(2):365–374, May 2016.

[117] Abhishek Kar, Shubham Tulsiani, João Carreira, and Jitendra Malik. Category-specific object reconstruction from a single image. In *CVPR*. IEEE, 2015.

[118] Ira Kemelmacher-Shlizerman and Ronen Basri. 3d face reconstruction from a single image using a single reference face shape. *IEEE transactions on pattern analysis and machine intelligence*, 33(2):394–405, 2011.

[119] Ira Kemelmacher-Shlizerman and Steven M Seitz. Face reconstruction in the wild. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1746–1753. IEEE, 2011.

[120] Sameh Khamis, Jonathan Taylor, Jamie Shotton, Cem Keskin, Shahram Izadi, and Andrew Fitzgibbon. Learning an efficient model of hand shape variation from depth images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2540–2548, 2015.

[121] Natasha Kholgade, Tomas Simon, Alexei A. Efros, and Yaser Sheikh. 3D object manipulation in a single photograph using stock 3D models. *ACM SIGGRAPH*, 33(4):127, 2014.

[122] Martin Kiefel and Peter Gehler. Human pose estimation with fields of parts. In *European Conference on Computer Vision, ECCV*, volume 8693, pages 331–346, 2014.

[123] Jaechul Kim, Ce Liu, Fei Sha, and Kristen Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *CVPR*. IEEE, 2013.

[124] Shahar Z. Kovalsky, Noam Aigerman, Ronen Basri, and Yaron Lipman. Controlling singular values with semidefinite programming. *ACM SIGGRAPH*, 33(4):68, 2014.

[125] Shahar Z. Kovalsky, Noam Aigerman, Ronen Basri, and Yaron Lipman. Large-scale bounded distortion mappings. *ACM Transactions on Graphics (Special Issue of SIGGRAPH Asia)*, 34(6):191:1–191:10, October 2015.

[126] Jonathan Krause, Hailin Jin, Jianchao Yang, and Fei-Fei Li. Fine-grained recognition without part annotations. In *CVPR*. IEEE, 2015.

[127] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[128] Tejas D. Kulkarni, Pushmeet Kohli, Joshua B. Tenenbaum, and Vikash Mansinghka. Picture: A probabilistic programming language for scene perception. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 4390–4399, 2015.

[129] Edwin H Land and John J McCann. Lightness and retinex theory. *Josa*, 61(1):1–11, 1971.

[130] Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J. Black, and Peter V. Gehler. Unite the people: Closing the loop between 3D and 2D human representations. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition CVPR*, 2017.

[131] Binh Huy Le and Zhigang Deng. Robust and accurate skeletal rigging from mesh sequences. *ACM Transactions on Graphics*, 33(4):84, 2014.

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

[133] David Levin. The approximation power of moving least-squares. *Math. Comput*, 67(224), 1998.

[134] Sijin Li and Antoni B Chan. 3D human pose estimation from monocular images with deep convolutional neural network. In *Asian Conference on Computer Vision, ACCV*, pages 332–347, 2014.

[135] Yaron Lipman. Bounded distortion mapping spaces for triangular meshes. *ACM SIGGRAPH*, 31(4):108, 2012.

[136] Yaron Lipman, Olga Sorkine, Daniel Cohen-Or, David Levin, Claudio Rossi, and Hans-Peter Seidel. Differential coordinates for interactive mesh editing. In *Shape Modeling Applications, 2004. Proceedings*, pages 181–190. IEEE, 2004.

[137] Ce Liu, Jenny Yuen, Antonio Torralba, Josef Sivic, and William Freeman. Sift flow: Dense correspondence across different scenes. *Computer vision–ECCV 2008*, pages 28–42, 2008.

[138] Jiongxin Liu, Angjoo Kanazawa, David Jacobs, and Peter Belhumeur. Dog breed classification using part localization. *Computer Vision–ECCV 2012*, pages 172–185, 2012.

[139] J. Löfberg. YALMIP: A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, 2004.

[140] Jonathan L Long, Ning Zhang, and Trevor Darrell. Do convnets learn correspondence? In *NIPS*, pages 1601–1609, 2014.

[141] M. Loper and M. J. Black. OpenDR: An approximate differentiable renderer. In *European Conference on Computer Vision, ECCV*, pages 154–169, 2014.

[142] Matthew Loper, Naureen Mahmood, and Michael J. Black. MoSh: Motion and shape capture from sparse markers. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH Asia*, 33(6):220:1–220:13, 2014.

[143] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, October 2015.

[144] Manolis IA Lourakis and Antonis A Argyros. Sba: A software package for generic sparse bundle adjustment. *TOMS*, 2009.

[145] David G Lowe. Solving for the parameters of object models from image descriptions. In *ARPA Image Understanding Workshop*, pages 121–127, April 1980.

[146] David G Lowe. Fitting parameterized three-dimensional models to images. *IEEE transactions on pattern analysis and machine intelligence*, 13(5):441–450, 1991.

[147] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[148] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.

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

[150] M. Marques and J. P. Costeira. Estimating 3D shape from degenerate sequences with missing data. *Computer Vision and Image Understanding*, pages 261–272, February 2009.

[151] Sebastian Martin, Bernhard Thomaszewski, Eitan Grinspun, and Markus Gross. Example-based elastic materials. *ACM Transactions on Graphics*, 2011.

[152] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell*, 27(10):1615–1630, 2005.

[153] Jessica Gall Myrick. Emotion regulation, procrastination, and watching cat videos online: Who watches internet cats, why, and to what effect? *Computers in Human Behavior*, 52:168–176, 2015.

[154] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, pages 483–499. Springer, 2016.

[155] J. Nocedal and S. Wright. *Numerical optimization*. Springer, 2006.

[156] Valsamis Ntouskos, Marta Sanzari, Bruno Cafaro, Federico Nardi, Fabrizio Natola, Fiora Pirri, and Manuel Ruiz. Component-wise modeling of articulated objects. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[157] Cornell Laboratory of Ornithology. The birds of north america online (p. rodewald, ed.). http://bna.birds.cornell.edu/BNA/, August 2015.

[158] Edwin Olson and Pratik Agarwal. Inference on networks of mixtures for robust robot mapping. *Int. J. Robotics Research*, 32(7):826–840, 2013.

[159] Martin R Oswald, Eno Töppe, and Daniel Cremers. Fast and globally optimal single view reconstruction of curved objects. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 534–541. IEEE, 2012.

[160] V. Parameswaran and R. Chellappa. View independent human body pose estimation from a single perspective image. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 16–22, 2004.

[161] Xavier Perez-Sala, Sergio Escalera, Cecilio Angulo, and Jordi Gonzalez. A survey on model based approaches for 2d and 3d visual human pose recovery. *Sensors*, 14(3):4189–4210, 2014.

[162] Mathieu Perriollat, Richard Hartley, and Adrien Bartoli. Monocular template-based reconstruction of inextensible surfaces. *International journal of computer vision*, 95(2):124–137, 2011.

[163] T. Pfister, J. Charles, and A. Zisserman. Flowing convnets for human pose estimation in videos. In *IEEE International Conference on Computer Vision, ICCV*, pages 1913–1921, 2015.

[164] T. Pfister, K. Simonyan, J. Charles, and A. Zisserman. Deep convolutional neural networks for efficient pose estimation in gesture videos. In *Asian Conference on Computer Vision, ACCV*, pages 538–552, 2014.

[165] Frédéric Pighin, Richard Szeliski, and David H Salesin. Resynthesizing facial animation through 3d model-based tracking. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 143–150. IEEE, 1999.

[166] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. DeepCut: Joint subset partition and labeling for multi person pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 4929–4937, 2016.

[167] Jean Ponce and Michael Brady. Toward a surface primal sketch. In *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, volume 2, pages 420–425. IEEE, 1985.

[168] G. Pons-Moll, D. Fleet, and B. Rosenhahn. Posebits for monocular human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 2345 – 2352, 2014.

[169] G. Pons-Moll, J. Taylor, J. Shotton, A. Hertzmann, and A. Fitzgibbon. Metric regression forests for correspondence estimation. *International Journal of Computer Vision, IJCV*, 113(3):1–13, 2015.

[170] Tiberiu Popa, Dan Julius, and Alla Sheffer. Material-aware mesh deformations. In *Shape Modeling International*, 2006.

[171] M. Prasad, A. Zisserman, and A. W. Fitzgibbon. Fast and controllable 3D modelling from silhouettes. In *Annual Conference of the European Association for Computer Graphics (Eurographics)*, pages 9–12, 2005.

[172] Mukta Prasad and Andrew Fitzgibbon. Single view reconstruction of curved surfaces. In *Computer Vision and Pattern Regognition (CVPR)*, 2006.

[173] V. Ramakrishna, T. Kanade, and Y. Sheikh. Reconstructing 3D human pose from 2D image landmarks. In *European Conference on Computer Vision, ECCV*, pages 573–586, 2012.

[174] Deva Ramanan. Learning to parse images of articulated bodies. In *Advances in neural information processing systems*, pages 1129–1136, 2007.

[175] Rajeev Ranjan, Vishal M Patel, and Rama Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *arXiv preprint arXiv:1603.01249*, 2016.

[176] Bernhard Reinert, Tobias Ritschel, and Hans-Peter Seidel. Animated 3d creatures from single-view video by skeletal sketching. In *GI '16: Proceedings of the 42st Graphics Interface Conference*, 2016.

[177] Lawrence Gilman Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963.

[178] 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.

[179] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH*, 23(3):309–314, 2004.

[180] A Roy-Chowdhury, Rama Chellappa, and R Gupta. 3d face modeling from monocular video sequences. *Face Processing: Advanced Modeling and Methods*, page 4, 2005.

[181] Mathieu Salzmann and Pascal Fua. Deformable surface 3d reconstruction from monocular images. *Synthesis Lectures on Computer Vision*, 2(1):1–113, 2010.

[182] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):824–840, 2009.

[183] Scott Schaefer and Can Yuksel. Example-based skeleton extraction. In *SGP07: Eurographics Symposium on Geometry Processing*, pages 153–162, 2007.

[184] Theresa M Senft and Nancy K Baym. Selfies introduction˜ what does the selfie say? investigating a global phenomenon. *International Journal of Communication*, 9:19, 2015.

[185] Hyewon Seo, Frederic Cordier, and Nadia Magnenat-Thalmann. Synthesizing animatable body models with parameterized shape modifications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 120–125. Eurographics Association, 2003.

[186] Hang Si. Tetgen, a delaunay-based quality tetrahedral mesh generator. *ACM Trans. Math. Softw*, 41(2):11, 2015.

[187] L. Sigal, A. Balan, and M. J. Black. HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision, IJCV*, 87(1):4–27, 2010.

[188] Leonid Sigal, Alexandru Balan, and Michael J. Black. Combined discriminative and generative articulated pose and non-rigid shape estimation. In *Advances in Neural Information Processing Systems 20, (NIPS)*, pages 1337–1344, 2008.

[189] E. Simo-Serra, A. Quattoni, C. Torras, and F. Moreno-Noguer. A joint model for 2D and 3D pose estimation from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 3634–3641, 2013.

[190] E. Simo-Serra, A. Ramisa, G. Alenya, C. Torras, and F. Moreno-Noguer. Single image 3D human pose estimation from noisy observations. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 2673 – 2680, 2012.

[191] C. Sminchisescu and A.C. Telea. Human pose estimation from silhouettes, a consistent approach using distance level sets. In *WSCG International Conference for Computer Graphics, Visualization and Computer Vision*, pages 413–420, 2002.

[192] C. Sminchisescu and B. Triggs. Covariance scaled sampling for monocular 3D body tracking. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 447–454, 2001.

[193] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from Internet photo collections. *IJCV*, November 2008.

[194] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Skeletal graphs for efficient structure from motion. In *CVPR*. IEEE, 2008.

[195] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry Processing*. Eurographics Association, 2007.

[196] Srinath Sridhar, Franziska Mueller, Antti Oulasvirta, and Christian Theobalt. Fast and robust hand tracking using detection-guided optimization. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 3121–3221, 2015.

[197] Robert W. Sumner and Jovan Popovic. Deformation transfer for triangle meshes. *ACM SIGGRAPH*, 23(3):399–405, 2004.

[198] Robert W. Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. *ACM Transactions on Graphics*, 26(3):80, 2007.

[199] Robert W. Sumner, Matthias Zwicker, Craig Gotsman, and Jovan Popovic. Mesh-based inverse kinematics. *ACM SIGGRAPH*, 24(3):488–495, 2005.

[200] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep convolutional network cascade for facial point detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3476–3483, 2013.

[201] Marshall F Tappen, William T Freeman, and Edward H Adelson. Recovering intrinsic images from a single image. In *Advances in neural information processing systems*, pages 1367–1374, 2003.

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

[203] Jonathan Taylor, Richard V. Stebbing, Varun Ramakrishna, Cem Keskin, Jamie Shotton, Shahram Izadi, Aaron Hertzmann, and Andrew W. Fitzgibbon. User-specific hand modeling from monocular depth sequences. In *Computer Vision and Pattern Regognition (CVPR)*, 2014.

[204] Bugra Tekin, Artem Rozantsev, Vincent Lepetit, and Pascal Fua. Direct prediction of 3D body poses from motion compensated sequences. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 991–1000, 2016.

[205] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. In *ACM SIGGRAPH*, pages 205–214. ACM, 1987.

[206] Demetri Terzopoulos, Andrew Witkin, and Michael Kass. Constraints on deformable models: Recovering 3d shape and nonrigid motion. *Artificial intelligence*, 36(1):91–123, 1988.

[207] Demetri Terzopoulos, Andrew Witkin, and Michael Kass. Symmetry-seeking models and 3d object reconstruction. *International Journal of Computer Vision*, 1(3):211–221, 1988.

[208] Jean-Marc Thiery, Emilie Guy, and Tamy Boubekeur. Sphere-meshes: Shape approximation using spherical quadric error metrics. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH Asia*, 32(6):178:1–178:12, 2013.

[209] Engin Tola, Vincent Lepetit, and Pascal Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE transactions on pattern analysis and machine intelligence*, 32(5):815–830, 2010.

[210] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in Neural Information Processing Systems*, pages 1799–1807, 2014.

[211] Alexander Toshev and Christian Szegedy. DeepPose: Human pose estimation via deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1653–1660, 2014.

[212] Shubham Tulsiani and Jitendra Malik. Viewpoints and keypoints. In *CVPR*. IEEE, 2015.

[213] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. *arXiv preprint arXiv:1701.01370*, 2017.

[214] Sara Vicente, João Carreira, Lourdes de Agapito, and Jorge Batista. Reconstructing PASCAL VOC. In *CVPR*. IEEE, 2014.

[215] Sara Vicente and Lourdes de Agapito. Balloon shapes: Reconstructing and deforming objects with volume from images. In *3DV*. IEEE, 2013.

[216] Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popović. Face transfer with multilinear models. In *ACM Transactions on Graphics (TOG)*, pages 426–433. ACM, 2005.

[217] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[218] C. Wang, Y. Wang, Z. Lin, A. Yuille, and W. Gao. Robust estimation of 3D human poses from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 2369 – 2376, 2014.

[219] Nannan Wang, Xinbo Gao, Dacheng Tao, Heng Yang, and Xuelong Li. Facial feature point detection: A comprehensive survey. *Neurocomputing*, 2017.

[220] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 4724–4732, 2016.

[221] A. Weiss, D. Hirshberg, and M.J. Black. Home 3D body scans from noisy image and range data. In *Int. Conf. on Computer Vision (ICCV)*, Barcelona, November 2011. IEEE.

[222] Andrew P Witkin. Recovering surface shape and orientation from texture. *Artificial intelligence*, 17(1-3):17–45, 1981.

[223] Jiajun Wu, Tianfan Xue, Joseph J Lim, Yuandong Tian, Joshua B Tenenbaum, Antonio Torralba, and William T Freeman. Single image 3d interpreter network. In *European Conference on Computer Vision*, pages 365–382. Springer, 2016.

[224] J. Y. Yan and M. Pollefeys. A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video. *IEEE Trans. Pattern Anal. Mach. Intell*, 2008.

[225] Y. Yang and D. Ramanan. Articulated pose estimation using flexible mixtures of parts. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 3546–3553, 2011.

[226] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, pages 1385–1392. IEEE, 2011.

[227] Hashim Yasin, Umar Iqbal, Björn Krüger, Andreas Weber, and Juergen Gall. A dual-source approach for 3D pose estimation from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 4948–4956, 2016.

[228] Mao Ye and Ruigang Yang. Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera. In *Computer Vision and Pattern Regognition (CVPR)*. IEEE, 2014.

[229] Shin Yoshizawa, Alexander Belyaev, and Hans-Peter Seidel. A fast and simple stretch-minimizing mesh parameterization. In *Shape Modeling Applications, 2004. Proceedings*, pages 200–208. IEEE, 2004.

[230] Alan L Yuille, David S Cohen, and Peter W Hallinan. Feature extraction from faces using deformable templates. In *CVPR*, pages 104–109. IEEE, 1989.

[231] Li Zhang, Guillaume Dugas-Phocion, Jean-Sebastien Samson, and Steven M Seitz. Single-view modelling of free-form scenes. *Computer Animation and Virtual Worlds*, 13(4):225–235, 2002.

[232] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based r-cnns for fine-grained category detection. In *European conference on computer vision*, pages 834–849. Springer, 2014.

[233] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape-from-shading: a survey. *IEEE transactions on pattern analysis and machine intelligence*, 21(8):690–706, 1999.

[234] Bo Zhao, Jiashi Feng, Xiao Wu, and Shuicheng Yan. A survey on deep learning-based fine-grained object classification and semantic segmentation. *International Journal of Automation and Computing*, pages 1–17, 2017.

[235] F. Zhou and F. De la Torre. Spatio-temporal matching for human detection in video. In *European Conference on Computer Vision, ECCV*, pages 62–77. Springer, 2014.

[236] Kun Zhou, Jin Huang, John Snyder, Xinguo Liu, Hujun Bao, Baining Guo, and Heung-Yeung Shum. Large mesh deformation using the volumetric graph laplacian. *ACM Transactions on Graphics*, 24(3):496–503, 2005.

[237] Shizhe Zhou, Hongbo Fu, Ligang Liu, Daniel Cohen-Or, and Xiaoguang Han. Parametric reshaping of human bodies in images. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH*, 29(4):126:1–126:10, 2010.

[238] Tinghui Zhou, Philipp Krähenbühl, Mathieu Aubry, Qixing Huang, and Alexei A. Efros. Learning dense correspondence via 3d-guided cycle consistency. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[239] Tinghui Zhou, Yong Jae Lee, Stella X. Yu, and Alexei A. Efros. Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In *CVPR*. IEEE, 2015.

[240] X. Zhou, M. Zhu, S. Leonardos, K. Derpanis, and K. Daniilidis. Sparse representation for 3D shape estimation: A convex relaxation approach. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 4447–4455, 2015.

[241] Xiaowei Zhou, Menglong Zhu, Spyridon Leonardos, Kosta Derpanis, and Kostas Daniilidis. Sparseness meets deepness: 3D human pose estimation from monocular video. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 4447–4455, 2016.

[242] Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2879–2886. IEEE, 2012.

[243] Silvia Zuffi and Michael J. Black. The stitched puppet: A graphical model of 3D human shape and pose. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2015)*, pages 3537–3546, June 2015.

[244] Silvia Zuffi, Angjoo Kanazawa, David Jacobs, and Michael J. Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017.