3D Animal Reconstruction with Deformable Template Models

Benjamin Biggs

Supervisor: Dr. Andrew Fitzgibbon
Prof. Roberto Cipolla

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Jesus College

March 2021

"Our perfect companions never have fewer than four feet."
Colette (1873 – 1954)

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Benjamin Biggs
March 2021

Acknowledgements

This work would not have been possible without the dedicated support from my PhD supervisors, Andrew Fitzgibbon and Roberto Cipolla. I would also like to thank the dedicated staff at GlaxoSmithKline, in particular the Pharma Supply Chain Tech Digital Innovation group, and particularly my line manager Patrick Hyett and project sponsor Julie Huxley-Jones.

The authors would like to thank Ignas Budvytis and James Charles for technical discussions, Peter Grandi and Raf Czlonka for their impassioned IT support, the Biggs' family for the excellent title pun and Philippa Liggins for proof reading.

The authors would like to thank the GSK AI team for providing access to their GPU cluster, Michael Sutcliffe, Thomas Roddick, Matthew Allen and Peter Fisher for useful technical discussions, and the GSK TDI group for project sponsorship.

The authors would like to thank Richard Turner for useful technical discussions relating to normalizing flows, and Philippa Liggins, Thomas Roddick and Nicholas Biggs for proof reading. This work was entirely funded by Facebook AI Research.

Abstract

TODO. Across many sectors concerned with animal husbandry, there is growing support for a system able to continuously monitor captive animals. Within farmyards, zoos, veterinary centres, animal research facilities and many others, humans typically take responsibility for identifying signs of disease or distress within their animal populations. While this can be effective, a significant challenge is posed when a small number of humans are expected to care for large animal groups.

This report discusses the development of a system to track, monitor and react to signs of poor physiological and psychological health among captive animals. In this work, it is proposed that a useful component of such a system would be the recovery of a detailed per-frame 3D animal reconstruction from an input video sequence. This is achieved through an approach which combines discriminative machine learning with generative model fitting to recover strong shape and pose attributes.

We present a system to recover the 3D shape and motion of a wide variety of quadrupeds from video. The system comprises a machine learning front-end which predicts candidate 2D joint positions, a discrete optimization which finds kinematically plausible joint correspondences, and an energy minimization stage which fits a detailed 3D model to the image. In order to overcome the limited availability of motion capture training data from animals, and the difficulty of generating realistic synthetic training images, the system is designed to work on silhouette data. The joint candidate predictor is trained on synthetically generated silhouette images, and at test time, deep learning methods or standard video segmentation tools are used to extract silhouettes from real data. The system is tested on animal videos from several species, and shows accurate reconstructions of 3D shape and pose.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Motivation

Animal welfare is an important concern for business and society, with an estimated 70 billion animals currently living under human care [34]. Across multiple industries, monitoring and assessing animal health is achieved by measuring individuals' body shape and movement. These measurements should be taken without interfering with the animal's normal activity, and are needed around the clock, under a variety of lighting and weather conditions, perhaps at long range (e.g. in farm fields or wildlife parks).

Of course, employing humans to monitor large animal populations is costly and can lead to data bias. For example, prey animals such as rodents are known to alter their behaviour in the presense of perceived predators. To overcome this, a number of animal monitoring systems have been designed, especially for use in clinical work. Many systems are 'invasive', meaning they require animals to undergo a surgical operation (generally to implant a tracking chip) before monitoring can take place. Although these systems can offer detailed biometric data (such as blood pressure, ECG etc.), the implantation procedure is costly and can cause stress to the animals, leading to welfare concerns and complex behavioural effects.

Since even crudest system can ensure some basic health standard (e.g. to check that *some* activity occurs over a given time period), some systems further attempt to monitor the animals' physical activity.

To overcome this, a number of non-invasive systems are available. The most basic of these can still be surprisingly effective, for example checking that *some* motion has occured during a given time period can offer an important health standard. More advanced systems can also estimate energy and inquisitiveness levels of the target animals by analysing motion patterns over time. Most systems achieve this by either placing floor-level pressure pads [152], or by installing an overhead camera which performs simple visual blob detection via colour

thresholding [133], [102]. One such open source system instructs the user to set a colour tolerance that masks all non-animal pixels and provide expected maximum and minimum animal sizes (in pixels) to help eliminate noise. While this is effective at tracking multiple animals with distinctive colour when placed in an arena with a solid, fixed background, it does not work well in many scenarios, e.g. outdoors. The presence of changing light levels, casting of shadows across tracking targets or moving backgrounds (e.g. foliage) make such thresholds ineffective. Further, this system's ability to distinguish between multiple tracked subjects is hindered when animals cross one another, as two individual blobs temporarily become one, and from then on are difficult to resolve.

Some work has been done in automatic behavioural scoring for rodents, in which up to ten predefined behaviours can be visually recognized. These approaches typically employ machine learning algorithms, which are taught to recognize behaviours present in a video stream by analyzing a large set of pre-collected examples. For 'normal' behaviours (e.g. drinking, eating etc.) this can be a viable approach and by analyzing the changing frequency of such behaviours can indeed offer insights into underlying conditions. However, these systems cannot be readily extended to handle more serious conditions (e.g. animals experiencing a seizure), due to the ethical concerns associated with collecting sufficient examples for training.

In addition, many Of course, even if a behaviour detection system were built for a range of animal species this

Unfortunately, all these approaches suffer as they fail to reconstruct a full 3D mesh.

1.2 Approach

This thesis focuses on developing methods for recovering a 3D model of an animal oa tracking system to enable recovery of a per-frame 3D animal reconstruction from an image or video stream.

The system should apply to a wide range of animal species without significant customization. Success in this endeavour would enable real-time changes of a known skeletal structure to be programmatically analysed to completely model an animal's movements. These behaviour patterns could then be interpreted to form a profile for each animal in a batch, taking into account expected norms for their species as well as their individual personality traits. When animals are first brought into a facility, they are given some time to acclimatize to their new surroundings before a clinical study begins. The application could make use of this period to refine behaviour models to their particular characteristics without being influenced by external factors. The system would then begin monitoring the population,

storing detailed analytics and reacting to any deviations to an animal's unique behaviour profile. As a simple example, should a typically lively and sociable dog suddenly begin exhibiting signs of withdrawal from the group, this would indicate a cause for concern and be stored in that animal's 'virtual log book'. In some cases, an animal may begin to exhibit signals that demand immediate attention, such as a dramatic and sudden energy drop that may indicate pain. The application could handle such events by sending an SMS text message to an on-call veterinary professional, to alert them of the specific problem and thereby enable a rapid response. These real-time diagnostics could then be aggregated and displayed on a dashboard screen, visible to all laboratory technicians. A concept drawing is shown in Figure 1.5.

1.2.1 Problem definition

A major challenge of this work is to develop and adapt methods for resolving the inherent ambiguity associated with recovering a 3D model from 2D input data. This challenge can be overcome by augmenting the input video sequence (Figure 1.1) with strong prior knowledge about the target species class (e.g. quadruped body measurements). This prior knowledge can be divided into two components: a *shape* prior that enforces topological (e.g. order of body parts) and measurement constraints (e.g. length of limbs), and a *pose* prior that defines likely limb configurations and can be used to rule out those which are anatomically impossible.

Fig. 1.1 An example input video sequence.

An example output showing the recovery of a 3D model from an input 2D monocular video is shown in Figure 1.2:

Fig. 1.2 Sample output printed from Deformable Mesh Animation [120].

A distinction should be made between two common tracking techniques: (1) discriminative body part recognizers and joint position predictors, and (2) 3D reconstructions via generative model fitting. Discriminative predictors have become the dominant paradigm in human body tracking to facilitate common use-cases, such as gesture detection or controller-less gameplay. However, recovering 3D models from human subjects is a growing field. Applications are found in fashion to faciliate online 'try-ons' for virtual clothing [69], in animation and visual effects to generate virtual characters from live actor performances [65], and in healthcare for tracking patients' body weight over time [138]. It is hypothesized that recovering a full 3D animal reconstruction is necessary to enable the intended diagnostic purposes of this animal work. In particular, returning only joint positions or body parts may be insufficient to estimate animal weight. If this can be realized, identifying behavioural changes from the reconstruction is expected to be a relatively straightforward machine learning problem.

A typical method for recovering 3D structure from tracking targets is using a *model fitting* approach, in which a 3D object representative of the target class is adapted to recreate the performance of the target. This method involves: (1) parameterizing a representative 3D *template mesh* with terms that represent shape and pose attributes and (2) defining an optimizer to adapt to these per-frame parameter settings to an input video sequence. An example of a template mesh is shown in Figure 1.3.

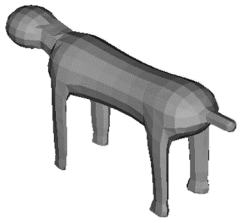


Fig. 1.3 An example prior, in this case a template mesh.

Shape attributes capture variation between different members of the target class and remain constant for a particular individual. For example, shape parameters may be adapted

to vary a model's height and weight. However, pose attributes generally capture limb positions and joint angles, and therefore tend to vary considerably during a capture sequence. Figure 1.4 highlights the difference by keeping pose parameters fixed while shape attributes are varied between the three models [121].

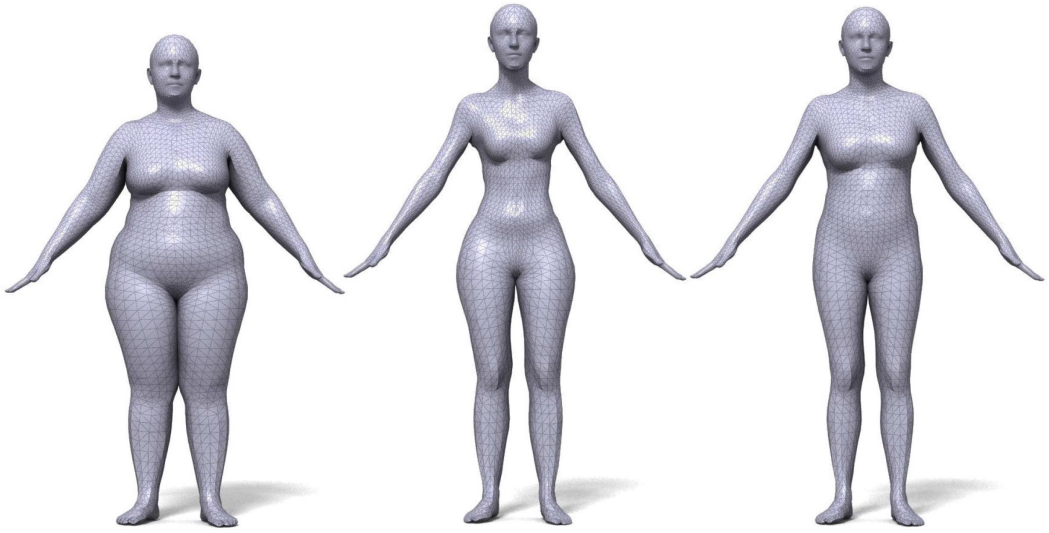


Fig. 1.4 Varying human shape parameters while pose remains fixed. Reprinted from [121].

Shape and pose parameters derived from a video sequence can be applied to the template mesh to generate a digital version of the same activity. If successful, the changing parameters should appear to adapt (or morph) the template mesh to faithfully reconstruct the performance given by the original live animal. In early experimentation, in which tracking targets are restricted to the same species, the template can be chosen to be a close shape fit to the target animal, thereby largely reducing the problem to finding optimal per-frame pose parameters. However, tracking examples are eventually broadened to include a wide range of animal species.

1.2.2 Relation to human reconstruction

Given that human tracking is now an established computer vision subfield, and the growing interest in analysing human behaviour from CCTV camera tracking systems, it is natural to ask whether the techniques used in this work transfer to the animal case. As an emerging research field, animal tracking presents many challenges in common with human gait and pose tracking problems, particularly in accurately monitoring morphable objects which frequently self-occlude. However, notable additional challenges are posed by the large shape and texture variation between animal tracking candidates and also due to the lack of

available training data which could otherwise be employed to train deep neural networks. An advantageous aspect of tracking animals over humans is the simple fact that animals tend not to wear clothing, which in humans causes significant shape and appearance variability.

The previous chapter discussed the primary objective of this work, which is to recover full 3D shape and pose from a live input video sequence exhibiting an animal subject. As explained, the major challenge common to all methods operating on monocular RGB input is to resolve the inherent depth ambiguity associated with recovering a 3D model from 2D input. Competitive methods achieve this by relying on strong motion cues [84] or (if available) by incorporating strong prior knowledge of the tracking target. Strong shape and pose priors (e.g. body part configuration, acceptable body part lengths, likely joint positions etc.) are available for this problem, so this report will focus on analysing these methods in the literature.

All solutions face an important design decision, which is to make a distinction between features of an input sequence the system should aim to model and to which it should remain invariant. For example, nearly all human systems aim to model the angle between a tracking target's upper and lower leg region, but nearly all will attempt to remain invariant to skin colour variation between candidates. The next two sections discuss examples of systems in which this decision has been made differently, generally according to the intended real-world application.

We address this problem using techniques from the recent human body and hand tracking literature, combining machine learning and 3D model fitting. A discriminative front-end uses a deep hourglass network to identify candidate 2D joint positions. These joint positions are then linked into coherent skeletons by solving an optimal joint assignment problem, and the resulting skeletons create an initial estimate for a generative model-fitting back-end to yield detailed shape and pose for each frame of the video.

Although superficially similar to human tracking, animal tracking (AT) has some interesting differences that make it worthy of study:

Variability.

In one sense, AT is simpler than human tracking as animals generally do not wear clothing. However, variations in surface texture are still considerable between individuals, and the variety of shape across and within species is considerably greater. If tracking is specialized to a particular species, then shape variation is smaller, but training data is even harder to obtain.

Training data.

For human tracking, hand labelled sequences of 2D segmentations and joint positions have been collected from a wide variety of sources [5, 68, 52]. Of these two classes of labelling, animal *segmentation* data is available in datasets such as MSCOCO [68], PASCAL VOC [30] and DAVIS [93]. However this data is considerably sparser than human data, and must be "shared" across species, meaning the number of examples for a given animal shape class is considerably fewer than is available for an equivalent variation in human shape. While segmentation data can be supplied by non-specialist human labellers, it is more difficult to obtain *joint position* data. Some joints are easy to label, such as "tip of snout", but others such as the analogue of "right elbow" require training of the operator to correctly identify across species.

Of more concern however, is 3D skeleton data. For humans, motion capture (mocap) can be used to obtain long sequences of skeleton parameters (joint positions and angles) from a wide variety of motions and activities. For animal tracking, this is considerably harder: animals behave differently on treadmills than in their quotidian environments, and although some animals such as horses and dogs have been coaxed into motion capture studios [145], it remains impractical to consider mocap for a family of tigers at play.

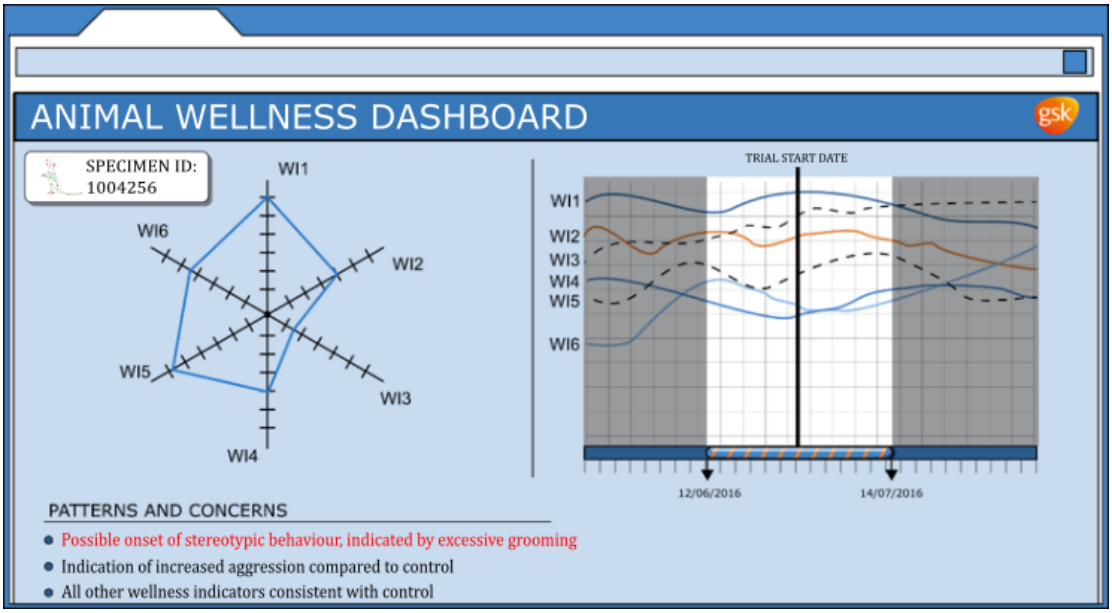


Fig. 1.5 Concept drawing showing an animal health dashboard. Specific wellness markers WI1,...,WI6 have yet to be determined.

1.3 Contributions

In summary, the contributions of this thesis are as follows:

1. We demonstrate a robust framework for 3D animal reconstruction using deformable template models

1.4 Co-Authored Papers

Extracts from this thesis appear in the following co-authored publications and preprints. Chapter 4 contains work from:

1. Benjamin Biggs, Thomas Roddick, Andrew Fitzgibbon and Roberto Cipolla, Creatures Great and SMAL, ACCV 2018 ORAL Presentation

Chapter 5 contains work from:

1. Benjamin Biggs, Oliver Boyne, James Charles, Andrew Fitzgibbon and Roberto Cipolla, Who Left the Dogs Out? 3D Animal Reconstruction with Expectation Maximization In the Loop, ECCV 2020

And Chapter 6 contains work from:

1. Benjamin Biggs, Sebastien Ehrhadt, Hanbyul Joo, Benjamin Graham, Andrea Vedaldi and David Novotny, 3D Multi-bodies: Fitting Sets of Plausible 3D Human Models to Ambiguous Image Data, NeurIPS 2020 SPOTLIGHT Presentation

1.5 Thesis Structure

The following five thesis chapters discuss methods for deriving 3D animal reconstructions from monocular input images and video. The first two chapters cover necessary background and an in-depth literature review covering related methods for animal reconstruction. Chapter 4 discusses an approach for animal reconstruction by learning only from synthetic training data. Chapter 5 describes an end-to-end and real-time technique applied to the challenging dog category. Chapter 6 introduces a method for handling input images with significant ambiguity. The final Chapter 7 summarizes the work and offers some opportunities for future endeavours in the field.

Chapter 2

Related Work

2.1 Introduction

This chapter discusses background and methods related to 3D shape and pose estimation for articulated subjects. We begin with techniques for correspondence prediction – a classical method which discusses work focussing on 3D shape and pose estimation for animals. 3D shape and pose estimation for articulated subjects – primarily animals and humans. We initially focus on skeletal prediction techniques, which output either sets 2D or 3D keypoints. Such techniques are of value to our objective of 3D surface reconstruction, as have been employed in multi-stage pipelines where keypoints are predicted first before a subsequent model fitting stage. The chapter continues with an evaluation of so-called 'model-free' techniques, which operate without an explicit template prior.

2.2 Predicting point correspondences

Before delving into methods for 3D reconstruction, it is first necessary to discuss techniques for identifying *point correspondences*. Point correspondences have a long history in computer vision for associating the same real-world location as it is represented by multiple camera views or on a 3D model surface. In a multi-image scenario, determining reliable correspondences between image pairs can be used to greatly reduce the ambiguity when reconstructing 3D scenes from 2D images. Even with only a single image available, correspondences can be predicted between the image and a representative 3D template mesh. This 3D-to-2D correspondence type is important for constraining the class of model fitting algorithms (discussed in depth later) which operate by aligning a 3D template mesh to a given 2D image. Of course, determining point correspondences is made more difficult in the presence of particular

nuisance factors. In the case of animal imagery, we must associate points on a non-rigid object with independently moving parts (articulated), deal with frequent self-occlusion in which limbs overlap each other from the perspective of the camera, occlusion caused by environmental factors (e.g. trees, fences, humans etc.), varied and unknown backgrounds and a range of complex lighting conditions (including shadows). Throughout this section, the methods highlighted will be appraised against their suitability in this complex setting.

2.2.1 Relating separate views of the same object/scene

The first class of techniques focuses on classical approaches for determining corresponding image points taken of precisely the same (and almost always rigid) object. Early techniques focused on stereo [42] or optical flow [47] imagery, and matched image points based on finding regions with similar pixel intensities. Due to the adverse effects caused by changeable environmental factors (e.g. lighting) would have on the appearance of the real-world location when captured in separate images, attention moved towards designing schemes with improved robustness. Improvements were achieved when matching points based on local *mid-level features* such as edges and corners, which have greater invariance to colour changes caused by lighting effects. Typical pipeliens would first identify *interest points* (typically corners [82, 43, 117] or blobs [76]), from which local image patches could be compared according to either Squared Sum of Intensity Differences (SSD) or a cross-correlation (CC) scheme. Steady improvements were then made through the design of ever-improving feature descriptors, which encode local image information around points and aim for invariance against common transformations (e.g. viewpoint, rotation and scaling). Progress in this field arguably reached maturity with the advent of SIFT [76], which encodes points according to local histograms of graident orientations and was later speed-up by SURF [8] and DAISY [131]. There have been modern attempts to learn sophisticated feature representations using convolutional neural network architectures [151, 41], which are shown to offer still further improvement.

The primary aim of these systems is to derive point correspondences between multiple views of the same object, usually as depicted in stereo images or between successive frames of a video. Unfortunately, by matching points based on local geometric features learnt from few image examples, these techniques do not readily extend to identifying correspondences between different instances of the same category. For example, matching SIFT features is likely to result in poor quality correspondences if tested on two dogs of different breeds due to the differing appearance and body geometry. For similar reasons, this class of techniques tend to deteriorate when tested on articulated objects since the object's structure can change and cause self-occlusion between views. The techniques are also known to suffer in scenarios with significant viewpoint changes (e.g. image of the front/back of an animal), since there

are few correpsonding points available for matching. Finally, these techniques do not directly offer a method for identifying correspondences between an image and a representative 3D mesh. Although some work exists that extends some of the aforementioned feature descriptors (e.g. 3D-SIFT [100]) to 3D, matching typically requires a photorealistic 3D scan of the 2D subject which we cannot assume as input for our problem.

2.2.2 Predicting keypoints with a semantic meaning

This section will explore an alternative class of methods for identifying point correspondences. So far, the approaches described do not detect correspondences with any semantic meaning; in other words, the returned points cannot truly be 'named' and there is no guarantee the same points (or even the same number of points) will be identified in different test images. Instead, this section will focus on techniques which predict a set of keypoint locations which are specified in a pre-defined list (for example: nose, tail tip, toe). In general, data-driven machine learning algorithms are used in order to learn an association between image appearance information and semantic keypoint labels. The techniques fall into two general categories: the former set of *supervised techniques* rely on large image datasets manually annotated with keypoint locations, and the latter set of *unsupervised techniques* learn the association through other means.

Supervised techniques

Early work in the supervised prediction of landmarks began through the refinement of object detection methods to predict fine-grained object part labels and eventually progressed to keypoint locations. Perhaps the earliest techniques in this category made use of face part annotations (referred to as fiducial points) to align target faces to improve the face recongition accuracy. Human detection and pose estimation methods progressed from simple bounding box representations [26], to object part prediction [9, 9], poselets [36] and subsequently 2D keypoint localization [9, 9]. Most commonly, methods aim to predict the location of important 2D human joints (such as the shoulders and wrists) in order to roughly approximate the subject's skeletal pose. For this reason, this task is commonly referred to as *2D human pose estimation*. The earliest techniques represented humans as a graph of parts [83] and fit shape primitives (e.g. cylinders [45]) to detected edges. Tree-based graphical models known as pictorial structures [33] were adopted and later made efficient [32]. Improvements were made with models capable of expressing complex relationships betwen joints, such as flexible part mixtures [150, 53].

Before the popularization of modern deep learning architectures, various methods made use of features computed underneath predicted 2D landmark locations for fine-grained image classification tasks. For this reason, there are limited examples of keypoint datasets for animal categories such as dogs [71] and birds [142]. Chapter 4 of this thesis will discuss StanfordExtra, a new dataset complete with annotated keypoint locations and segmentation masks for 12,000 dog images, encompassing 120 different breeds. At the time of publication, StanfordExtra is the largest annotated animal dataset of its kind.

Recent works in 2D pose estimation typically employ convolutional neural networks (CNNs) due to the complex feature represenations that can be learnt for joints that, when applied discriminatively, enable accurate recongition. An early example [127] learnt a pose embedding space with a CNN, and employed a nearest neighbour search algorithm to regress a pose. Later, deeper CNN models were used to regress facial point [125] and full body [134] landmarks. More recent works improve robustness by regressing keypoint confidence maps [132] rather than 2D keypoints directly, enabling spatial priors to be applied to remove outliers [18, 95, 96, 22, 132, 135, 97]. More recent methods are able to directly produce accurate confidence maps through a multi-stage pipeline [141]. Of particular note are hourglass [85] (relied upon in this thesis Chapter 3) and multi-level [122, 147] structures, which combine global reasoning of full-body attributes and of fine-grained details. A related class of methods [39, 128] focus on *dense* human pose estimation, which relate all 2D image pixels to a representative 3D surface of the human body.

Modern techniques in 2D human pose estimation demonstrate impressive accuracy on in-the-wild datasets, and deal with with parsing multiple subjects in challenging poses and in the presence of various occluders. However, part of what enables these achievements is the prevalance of large 2D keypoint datasets which can be used for training. Further discussion of available 2D keypoint datasets has been left for Chapter 5, in which they are considered in-depth. Further discussion on the history and advances in 2D human pose estimation are comprehensively reviewed in [98, 27].

Unsupervised learning

As this thesis focuses on developing methods for animal reconstruction, it is useful to review techniques which operate without large 2D keypoint training datasets, which are scarce for animal subjects. Note that the methods in this section all describe approaches for determining point correspondences between different scenes. Under consideration are methods based on transfer learning, unsupervised learning and methods based on weak-supervision.

Early correspondence techniques include dense alignment methods including SIFT-flow [70] which employed optical flow methods to match image using SIFT features, and

Bristow et al. [15] who demonstrate a method for learning per-pixel semantic correspondences using geometric priors. They also show examples on various animal categories. Recent unsupervised techniques learn *category-specific* semantic priors by employing deep networks on large image collections.

Zhou et al. [155] demonstrate a method for solving correspondences across an image collection by enforcing cycle consistency. Kanazawa et al. [57] introduce WarpNet which predicts a dense 2D deformation field for bird images by learning from synthetic thin-plate spline warps generated on extracted silhouettes. Thewlis et al. [130] apply a similar trick, by ensuring a consistent mapping of warped facial images to a spherical coordinate frame and show results on human and cats. Jakab et al. [51] show they can estimate 2D human pose without training data data by leverging that between two frames of a simple video sequence, human body shape and texture remains reasonably similar but the pose (including global rotation) varies. They therefore construct an architecture that, given a pair of frames (I, J) defines a network f that given frame I predicts a 2D location vector y. The system then combines this vector y with the second frame J and trains a secondary network g to reconstruct the original frame I. Due to the limited capacity of v, the fact that apart from the pose, most of the information necessary for reconstruction is already available in J, the network eventually learns to encode 2D pose coordinates using v.

Transfer learning describes a family of methods in which a machine learning model is first *pre-trained* to solve a related task (often making use of secondary dataset with may be larger in size) in order to accumulate knowledge which offers an advantage when solving the original task. DeepLabCut [79], LEAP [94] and DeepPoseKit [37] exemplify such techniques, in which existing architectures [97, 85, 48, 107] are first trained to predict 2D human pose (making use of the large available datasets), and are then repurposed to predict 2D animal keypoints using few (generally 100s) training examples. Cao et al [17] demonstrate a cross-domain adaptation technique, which transfers knowledge gained from a modestly-sized animal dataset to unseen animal types. There are also dense estimation techniques, which extend DensePose [39] described above to proximal animal classes [106], such as chimpanzees, by aligning the geometry between the animal category to humans for which data is plentiful.

2.3 Image segmentation and object detection

This section will briefly discuss techniques for extracting binary segmentation masks and object bounding boxes for subjects present in input images.

2.4 Modelling articulated subjects

The design of 3D morphable models (3DMMs) has a significant recent history in computer vision research. A 3DMM is a statistical model designed to represent the structure, deformation and apperance space of for a particular object category. Such a model can be constructed for any object category for which a dense point-to-point correspondence can be established between instances. For example, a 3DMM can be designed to represent medium-sized quadrupeds but perhaps not for general animal categories. How, for instance, would one sensibly determine correspondences between a dog and an octupus? 3DMMs have been used extensively as a strong 3D prior to aid various 3D reconstruction algorithms. They are, however, most influential for problems with the most ambiguity: particularly when dealing with articulated objects (e.g. animals or humans), when only a single monocular RGB image is available or when no paired 3D training data is available.

Blanz and Vetter [12] presented the first 3DMM, which expressed a low-dimensional face space space learnt by aligning various face scans. This work, presented over two decades ago, has been recognized with an impact paper award for the continued applications for the ideas presented. Indeed, the approach introduced has found applications far beyond faces [16, 92, 35], including for cars [109], other human body parts including the hands [60] and ears [25], the human body surface [6, 73] and a restricted set of animal categories [124, 158].

This section will cover methods for modelling articulated subjects, focussing primarily methods for human bodies and animals.

2.4.1 Constructing 3D morphable models

3D deformable models are typically represented by a polygon mesh. A polygon mesh $M = (V, T)$ is a collection of vertices, edges bound by vertex pairs, and polygons bound by sequences of edges and vertices [116]. Although other convex shapes are allowed, this thesis only has need to discuss triangular mesh polygons, which henceforth will be referred to as *triangles*. An example mesh is shown in Figure 2.1.

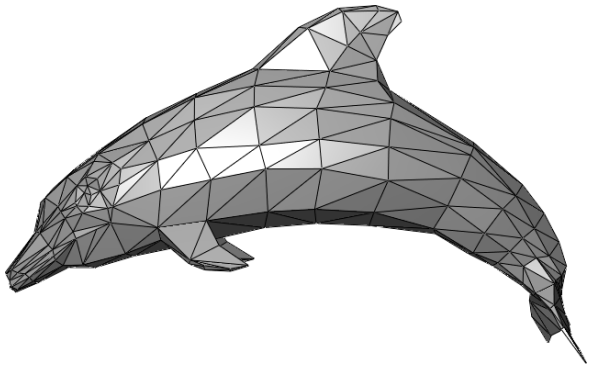


Fig. 2.1 A polygon mesh [143].

A 3D morphable model can then be constructed by deforming a template mesh M on n-vertices to a set of 3D training examples. Generally, this optimization will have a significant degrees of freedom so it is necesssary to employ techniques for regularization. One such regularizer is known as the *As Rigid as Possible* scheme:

Definition 2.4.1 (As Rigid as Possible)**.** As Rigid as Possible (ARAP) surface deformation [119] is a distance function that measures similarity between two meshes with corresponding vertices. For two vertex sets V and W, ARAP minimizes over $N = |V|$ rotation matrices. Note $j \sim i$ indicates vertex indices j adjacent to vertex index i:

$$D(V,W) = \min_{R_{1..N}} \sum_{i=1}^{N} \sum_{j \sim i} ||(V_i - V_j) - R_i(W_i - W_j)||^2 \tag{2.1}$$

This distance function can be incorporated into an energy-based optimizer as a regularization function. By considering how small vertex regions overlap, the function can be used to discourage 'unnatural movement', e.g. shearing effects, over mesh faces. ARAP regularizers are particularly useful in cases in which there is no prior knowledge of the mesh. Figure 2.2 shows an example of a dinosaur mesh undergoing ARAP deformation, obtained by translating the highlighted yellow vertex.

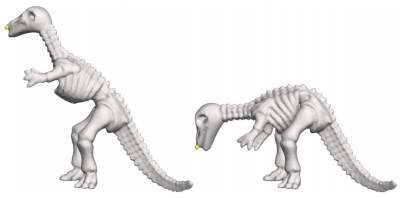


Fig. 2.2 Dinosaur mesh undergoing ARAP deformation, obtained by translating the highlighted yellow vertex. Reprinted from [119].

Once aligned, a d-dimensional *shape space* can then be defined (with $d << n$), where each $w \in \mathbb{R}^d$ gives rise to a vertex configuration in $\mathbb{R}^3$ (with unchanged triangulation). In this way, every plausible 3D example has a parameter vector $w \in \mathbb{R}^d$ that generates it. This construction can then be interpreted as a *generative* model. However, very few selections of $w \in \mathbb{R}^d$ will generate a plausible-looking 3D mesh. This can then be interpreted probabilistically, by defining a density function $f(w)$ that defines the likelihood that a realistic 3D example would be represented by w in shape space.

2.4.2 Modelling shapes (e.g. faces)

The concepts raised above were first introduced in the seminal work of Blanz and Vetter [12]. They define a linear generator function based on principle component analysis (PCA) in order to map d-dimensional parameter vectors to the set of n vertex coordinates. In particular they use the mapping:

$$g(\alpha) = \bar{c} + E\alpha \tag{2.2}$$

where $g : \mathbb{R}^d \mapsto \mathbb{R}^{3n}$ is the generator function, $\bar{c} \in \mathbb{R}^{3n}$ is the mean 3D face in the training dataset and $E \in \mathbb{R}^{3n \times d}$ is a matrix containing the d most dominant eigenvectors computed over shape residuals $\{c_i - \bar{c}_i\}$.

This construction assumes 3D faces in this d-dimensional parameter space follow a multivariate normal distribution (a concept explored further in this thesis Chapter 4). In addition, the function $f(w)$ which defines the likelihood shape space vector α represents a plausible face, is therefore given by the Mahalanbois distance of α to the origin.

Note that this formulation additionally enables the definition of facial expressions. For example, Blanz and Vetter defined an expression (e.g. surprise) according to the difference in shape space between a expressive and neural face of the same subject. This then enabled the formulation above to be factored into identity and expression components:

$$g(\alpha_{idt}, \alpha_{exp}) = \bar{c} + E_{idt}\alpha_{idt} + E_{exp}\alpha_{exp} \tag{2.3}$$

where E_{idt}, E_{exp} are the basis vectors of the identity and expression space and $\alpha_{idt}, \alpha_{exp}$ are the coefficients. As noted by Lewis et al. [9], the basis vectors of the expression space above can be interpreted as a data-driven *blendshape model*: a standard approach in the animation industry for representing facial expressions as a linear combination of target faces. This concept will later reemerge in a section discussing corrective blendshapes uses in SMPL [73].

As identified by Blanz and Vetter, improved modelling of finer details (particularly around the eye or nose regions) can be obtained through local modelling. Various authors [9] began manually segmentating the face into parts and learning individual PCA representations for them. Later, segmentations were automatic and learned based on displacement patterns found in the training dataset. Next, approaches were adopted based on hierarchical, multi-scale frameworks [9, 9]. Possibly the closest to later sections which require a focus on *pose deformation* is the work of Wu et al [9]. who combine a a local shape space model with an anatomical bone structure that helps regularize deformation.

A standard challenge in face modelling is towards reconstructing appearance, typically incorporating albedo and illumination (although frequently these are not factored, in which case apperance is generally referred to as *texture*). Early work modelled shape and texture independently [9, 9], although recent techniques show solving for these factors jointly enable constraints to be applied due to correlations present. Perhaps most interesting are the recent techniques among these [9, 9], who propose methods based on deep convolutional models to jointly model shape and texture.

2.4.3 Modelling articulation (e.g. hands)

3D morphable models have also influenced work in 3D hand tracking and modelling. Human hands servce multiple purposes in everyday life, serving a mechanism to handle tools/objects, expressing emotion and aiding (or even as the primary tool for) communication. As a result, hands (and particularly fingers) exhibit complex articulation patterns which are best characterized as 3D *rotations*. Compared to the previous section, in which face shape variation could be represented as as an abstract linear basis learnt from scans, an advantage to modelling hands is the modes of articulation can be defined in advance.

In particular, human hand motion is controlled by a hierarchical bone structure referred to as a *skeleton*. The point at which two bones meet is referred to as a *joint* and can be used to define acceptable centers of articulation. The direction and magnitude of the articulation can then be neatly expressed as a 3D rotation.

This formulation helps provide insight into why the abstract linear basis (shape space) introduced in the previous section would be poor choice for modelling hands. Deformation is here characterized in terms of 3D rotations, and 3D rotations are non-linear with respect to the input angle. This is easily shown:

Definition 2.4.2 (3D Rotations)**.** The simplest kind of 3D rotation is an *elementary rotation* and involves a rotation around a single axis of a coordinate system. For example, the following matrix represents a rotation by an angle γ around the x axis:

$$R_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{bmatrix} \tag{2.4}$$

One can then apply this matrix to an input point $p \in \mathbb{R}^3$ to compute the new position p' after rotating γ around the x axis:

$$p' = R_x(\theta)p \tag{2.5}$$

This formulation can then be extended to represent any 3D rotation as the composition of elementary rotations. For example, a 3D rotation can be decomposed into a γ rotation around the x-axis (pitch), followed by a β rotation around the y-axis (yaw) and finally by an α rotation around the z-axis (roll).

$$R = R_z(\alpha)R_y(\beta)R_x(\gamma) \tag{2.6}$$

One can see immediately that affecting a 3D rotation (i.e. computing the new position of the points) is a non-linear function of the input angle. It is necessary, therefore to describe an alternative technique for low-dimensional and efficient mesh deformation, which relies on *rigging* and *linear blend skinning*.

Definition 2.4.3 (Skeletal Rigging and Linear Blend Skinning (LBS))**.** In cases that the articulated object is known in advance, it is common to augment a representative 3D mesh model with an internal skeleton that approximates the biological counterpart. This is achieved through a process known as *rigging*.

Formally, a skinned mesh consists of a set of rigged vertices $V \subseteq \mathbb{R}^3 \times \mathbb{R}^{|J|}$, a set of faces $F \subseteq V^3$ and joint transformation matrices $J \subseteq \mathbb{R}^{3\times 4}$. Each vertex $v = (x, s) \in V$ consists of positional coordinate $x \in \mathbb{R}^3$ and a weight vector $s \in \mathbb{R}^{|J|}$ which describes the level of influence each joint $j \in J$ has over its movement. Many approaches exist for assigning weights, but perhaps the simplest is to build a vector with entries corresponding to the distance from the vertex to each joint centre. Skinning weight vectors are normalized such that their entries sum to one, and for computational reasons, the number of non-zero elements is typically limited to 2 or 4. The weakness of such models is that artifacts and other unrealistic deformations can occur around the model joints, particularly for meshes that model non-linear structures such as humans. However, the technique is frequently used in computer graphics and game design when a character's shape is known ahead of time.

Once a mesh has been suitably rigged, Linear Blend Skinning (LBS) can be used to apply the mesh deformation. Typically, a user assigns a transformation (e.g. rotation and

translation) to each 'joint' and LBS computes the an updated position of vertex $v = (x, s)$ where x is the original location and $s = \{s_j\}$ are the skinning weight influence of joints j. The matrix U_j (occasionally referred to as a the *reference pose*) is the mapping from the bone's "default" coordinate system to world coordinates. As a result, it need only be computed (and inverted) once since it is invariant to changing joint angles. In general, a user will supply the matrices D_j to define the mapping from the bone's deformed coordinate system to world coordinates.

The updated position of an input point x can then be calculated by LBS:

$$LBS(D, x; U, s) = \sum_j s_j D_j U_j^{-1} x \tag{2.7}$$

Note this formulation is made slightly more complicated in the case of kinematic trees, since the

Similarly to the approach mentioned above, this formulation can again be seen as a generative 3D model. In particular, an output vertex positions v' can be computed from a vector of 3D joint rotations θ and input vertex position v as follows:

$$v' = LBS(R(\theta), v) \tag{2.8}$$

With the above formulation, it is now possible to give an overview of recent hand tracking literature. In many cases, 3D morphable hands models are built to reflect the 27 biological hand bones (occasionally except the capal bones which join the fingers to the wrists). Of course, while the major source of hand deformation is due to 3D pose, some sources of variation are present due to hand size and finger propotions.

Allen et al [9] handle this variation adapting a 3D surface with displacement maps with various constraints designed to avoid self-intersections before adopting the linear blend skinning formulation defined above. Rhee et al [9] learn a shape deformation space (with a similar technique to that of faces) and user-specific skinning weights for LBS. Albrecht undergo a laborious process to create extremely detailed hand models through laser scanninng plaster casts [9]. A significant advance was presented by Taylor et al [9]. who presented a method to learn a personalized hand model (although not a shape basis) given an input video sequence (with depth) taken of a user slowly articulating their fingers. Ballan et al. [9] follow a similar process by making use of multi-view input.

2.4.4 Modelling the human body surface

However, of all human categories, the works of most relevance to this thesis are those which represent the entire human body surface. It is first important to characterize these two deformation modes which must be overcome with modelling algorithms. Firstly, there is considerable variation in the *shape* characteristics between different human subjects. Humans not only vary in their heights and weights, but also in their body part proportions, muscle density, fat etc. Secondly, humans exhibit significant *pose* variation, characterized by the range of motion of body parts (e.g. arms and legs). In general, pose is likely to change for a individual subject over a sequence.

The earliest deformable 3D models of the human body was presented by Allen at al [9] (although [9] came soon after with similar ideas). Allen et al. learnt a PCA shape space model from 250 registered body scans cound in the CAESAR dataset. The model was articulated through a set of pose parameters, which use linear blend skinning to interpolate rotation matrices assigned to the joint to transform model veertices. Unfortunately, this approach suffers from artefacts around joint locations, due to a loss of volume. For this reason, it is important to note that pose and shape are not entirely independent; in fact, body shape does indeed change due to pose variation. Imagine for example, how a fatty stomach region would deform during a walking sequence. SCAPE [6] improved over this by introducing a model equipped with both body shape variation and pose-dependent shape changes, expressed in terms of triangle deformations (rather than vertex displacements, see [73] for a comprehensive overview). An important advance was made by Hasler et al. [9], who learn two linear blend rigs: one for pose and one for body shape. In this model, shape change was controlled through the introduction of abstract bones that further deform the vertices.

Perhaps the most significant advance however, was the introduction of the Skinned Multi-Person Linear (SMPL) model of Loper et al. [73]. SMPL follows a similar design philosophy to SCAPE by decomposing shape into identity-dependent and pose-dependent components. However, unlike scape, SMPL adopts a vertex-based skinning approach based on corrective blend shapes. The model's shape space is first taught how human beings deform through pose changes using 1786 high-resolution 3D scans of different subjects in a wide variety of poses. Following alignment to a template mesh, a linear model for each biological gender is created from the CAESAR dataset [101] using principal component analysis (PCA). SMPL can then be viewed as a function, which makes use of a shape basis and linear blend skinning to map a set of pose and shape parameters to a set of vertex locations. Precisely, *pose* is given as a set of 3D rotations (per-joint and global) in axis-angle form $\theta \in \mathbb{R}^{24}3$. *Shape* is

then given as coefficients for a learned shape basis $\beta \in \mathbb{R}^{10}$. The SMPL function can then be viewed as:

$$v = \text{SMPL}(\theta, \beta) + t \tag{2.9}$$

where $v \in \mathbb{R}^{6890\times 3}$ and $t \in \mathbb{R}^3$ is a global translation parameter. Further details on SMPL have been left to Chapter 5 of this thesis, which makes use of the model to examine uncertainty when deriving 3D reconstructions of ambiguous input imagery.

More recently, SMPL has been combined with face and hand models to add expressive capabilities [146, 55, 90]. CAPE [77] also shows how to add a clothing parameter to effectively model humans in clothing, a challenge solved by learning a shape prior over freeform vertex deformations Techniques have also been developed to model human clothing – a common challenge generally handled by allowing SMPL model vertices to vary independently to the provided blend shapes. SMPL has also been recently improved with STAR [88] which constructs a part-based shape space (closely related to the local PCA space discussed in the earlier shape section of this literature review). They show this new parameterization is much more efficient (uses approximately 20% of the model parameters of SMPL) and avoids capturing spurious long-range correlations present in the training dataset. They also show a method for learning shape-dependent pose-corrective blendshapes, that better model how individuals with different body shapes deform with motion. Tangential work of Xu et al. [149] train an end-to-end network and learn 3D human body model parameters (including faces and hands) for an input artist model using variational auto-encoders and normalizing flows. This work will be further explored Chapter 5 in which these generative models will be fully examined.

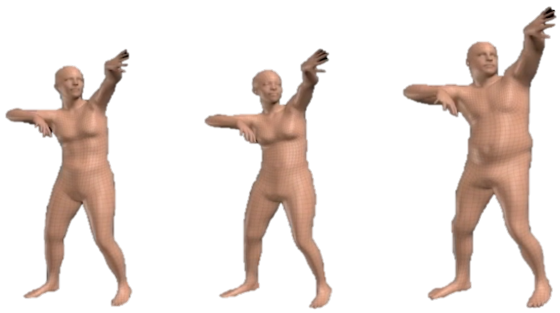


Fig. 2.3 SMPL model showing pose-invariant shape changes, reprinted from [73].

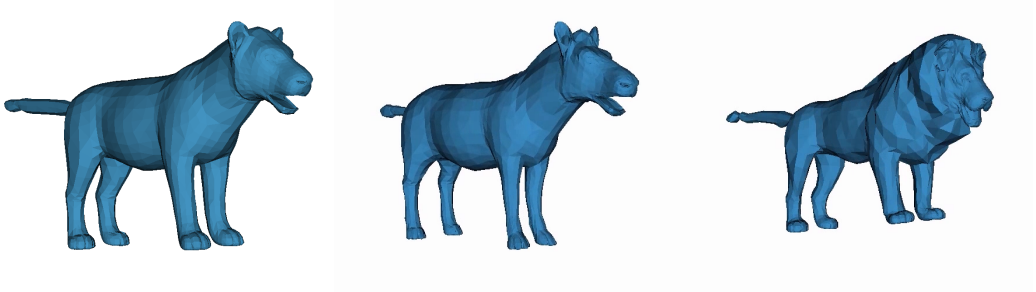
2.4.5 Modelling animals

There is still relevatively little work specifically focusing on the 3D scanning [9] and modelling of animal categories. The variation in animal shape and sizes combined with the practical challenges associated with scanning live animal subjects (particularly in attaching traditional motion capture equipment) make scanning a difficult task. As a result, there is a significant lack of real 3D animal training data available in the public domain which could otherwise have been employed to build 3D deformable models. As with humans, animals deformations can again be factored into shape (e.g. variation mostly due to identity) and pose (variation due to articulated motion). However, the enormous diversity among animal species and even between individual breeds results in a much more complex shape space.

Some early work by Favreau et al [9] describe a method for animating an artist-designed rigged 3D model, by tracking a 2D sequence. Chen et al. [9] learn a shape space by registering 11 3D shark models downloaded from the Internet. Cashman et al. [9] learn a morphable model of dolphin shapes by adapting a representative 3D model to 2D images. Ntouskos et al. [9] fit geometric primitives to manually-segemented animal parts generated from an input collection. Reinert et al. [9] demonstrates an effective method for fitting generalized cylinders to an input video sequence supplied with sketched limb tracks. They demonstrate reconstructed results with 3D texture on a few quadruped sequences. So far, none of these techniques for animal reconstruction explicitly factor shape and pose.

SMAL

A similar technique to that used to build the SMPL model has been recently used to build a Skinned Multi-Animal Linear Model (SMAL) [158], a generative animal model exhibiting realistic 3D shape (see Figure 2.4) and pose (see Figure 2.5). Due to the lack of available motion capture data for animal subjects, the SMAL model is learnt from a set of 41 3D scans of toy figurines in arbitrary poses. The figurines span five quadruped families, and included examples of lions, cats, tigers, dogs, horses, any many more, although notably for this work no rodent toys were included. The paper introduces a new technique to accurately align each toy scan to a common template, allowing the shape space to be learnt.



(a) Default SMAL mesh. (b) SMAL in horse shape. (c) SMAL in lion shape.

Fig. 2.4 SMAL with varying shape parameters.

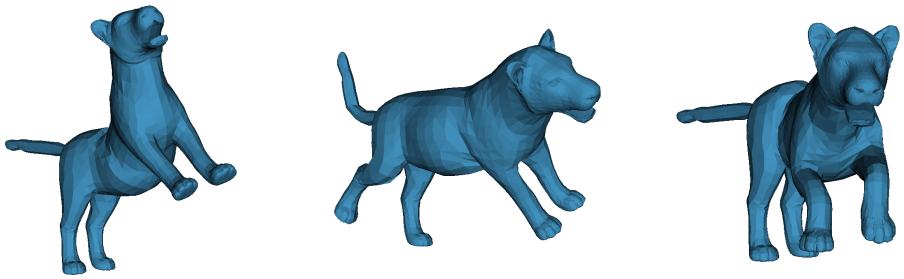


Fig. 2.5 SMAL with varying pose parameters.

From the paper, SMAL is defined as a function SMAL(θ, β) parameterized by pose-invariant shape $\beta \in \mathbb{R}^{41}$ (again, coefficients of a low-dimensional shape space) and pose $\theta \in \mathbb{R}^{32\times 3}$ (including global rotation). There are three pose parameters for each of the 32 body joints and an additional three to express the global rotation. Global translation γ is expressed by a further three parameters. The SMAL function returns a triangulated surface comprising 6890×3 vertex locations. Chapters 3 and 4 of this thesis make use of the SMAL model in order to reconstruct various quadruped categories.

2.5 Methods for monocular reconstruction of articulated subjects

Having discussed methods for modelling articulated subjects, this section will discuss approaches for reconstructing the 3D shape and pose of a subject from a monocular image or video. It is important to note that this task is challenging and fundamentally ill-posed. In common with other challenging 3D reconstruction tasks, input images will typically exhibit variation in camera view, lighting and environmental occlusion. However, 3D reconstruction

pipelines for articulated subjects must also deal with variation due to body shape, body pose, clothing and self-occlusion (body parts obscuring other parts). In addition, the challenge of reconstructing 3D models from 2D images is also inherently ambiguous. As explained by Toshev and Szegedy [134], even if 2D structure can be determined (for example, using 2D keypoint prediction), the subsequent 'lifting' step to recover 3D remains ill-posed, as the space of consistent 3D poses for given 2D landmark locations is infinite. It is for this reason that the history of monocular 3D reconstruction makes extensive use of 3D morphable models (or other geometric, temporal, structural priors), as they provide necessary optimization contraints.

The theme of this section is therefore to discuss how 3D morphable models (3DMMs) can be incoporated into 3D reconstruction pipelines. In general, algorithms take as input an input image or video and predict a set of 3D model parameters α (often factored into shape β and pose θ). Once determined, the output parameters are then supplied to the morphable model's generator function $g : \alpha \mapsto \mathbb{R}^3$ (e.g. SMPL : $(\beta, \theta) \mapsto \mathbb{R}^3$ or SMAL : $(\beta, \theta) \mapsto \mathbb{R}^3$) to produce vertex positions $V \subseteq \mathbb{R}^3$ for the model $M = (V, T)$. Recall that in general the triangulation T of such a model is fixed. For completeness and comparison, this section will make some mention of the small class of 3D reconstruction methods for articulated subjets which operate without an explicit 3D morphable model. Although currently a developing area, current work in this category typically requires either paired 3D training data, employ alternative (and arguably more restrictive) shape priors (e.g. symmetry constraints) or produce results of significantly lower fidelity.

Methods which align a parametric 3D model to monocular input date back as far as 1963, in a seminal paper by Roberts [9]. Roberts presents a method which optimizes parameters for viewpoint and cuboidal shape primitives to reconstruct a 2D line image. Model-based methods have also been applied to understand object structure, starting with fitting of geometric primitives [9] and later with Active Shape Models [9] which learn deformation priors from a provided training set. Perhaps due to the numerous commercial applications, the majority of recent work in 3D shape and pose recovery focuses particuarly on *humans* as a special case. The first example of such an approach is the seminal work of Blanz and Vetter [12] who built the first 3D morphable face model by aligning 3D scans and optimized the parameters to provide a fit to a single image. Since then, the research community has collected a multitude of open source human datasets which provide strong supervisory signals for training deep neural networks. These include include accurate 3D deformable template models [73] generated from real human scans, 3D motion capture datasets [50, 140] and large 2D datasets [68, 52, 5] which provide keypoint and silhouette annotations. The combination of these publically available datasets and their incoporation into deep learning pipelines

have led to impressive reconstruction results when tested on in-the-wild human images and videos. Unfortunately, the diversity among animal subjects and the practical challenges associated with data capture have resulted in few datasets being made available. Despite appearing superficially similar to human tracking, these factors result in specific challenges to animal tracking which must be carefully handled. Perhaps for this reason, the body of related literature for animal tracking is considerably sparser. The remainder of this section will focus on methods for 3D pose estimation, followed by 3D shape and pose reconstruction of human and animal bodies. Further discussion on techniques for body part (e.g. face, hands) reconstruction are deferred to the following survey papers [9, 9].

2.5.1 3D Pose Estimation

Techniques for 3D pose estimation output a set of 3D keypoint locations which can be combined to form a skeletal outline. Apart from basic limb measurements, no other shape detail (e.g. surface definition, object density etc.) is obtained. However, it should be noted that this output form is often perfectly satisfactory depending on the intended application. In particular, this family of techniques have found numerous applications in controllerless gaming (e.g. Microsoft Kinect [111]), motion capture (e.g. for digital character generation [9]), gait analysis (e.g. identifying lameness in cattle [9]) and many more.

The general approach is to recover a 3D skeleton such that the 3D joints project to known or estimated 2D joints subject to anatomical priors. Early approaches in this category fit human stick figures with various constraints, including assumptions of fixed limb lengths [9], length ratios [9] or that limb lengths are isometric across individuals and vary only in global scaling [9]. More advanced techniques built statistical models of shape variation using anthropometric tables or learnt them from motion capture data [7].

A broad category of appraoches for this are methods for *non-rigid structure from motion* [9]. The general formulation is to express a 3D skeleton $S \in \mathbb{R}^{3\times P}$ on P points as a linear combination of basis shapes $S_1, \ldots, S_k$ where $S_i \in \mathbb{R}^{3\times P}$. Precisely:

$$S = \sum_{i=1}^{K} l_i \cdot S_i \quad S, S_i \in \mathbb{R}^{3\times P} \quad l_i \in \mathbb{R} \tag{2.10}$$

Assuming scaled orthographic projection, the following expression represents the projection of P points of S into 2D image coordinates (u_i, v_i):

$$\begin{bmatrix} u_1 & u_2 & \ldots & u_P \\ v_1 & v_2 & \ldots & v_P \end{bmatrix} = R \cdot (\sum_{i=1}^{K} l_i \cdot S_i) + T \tag{2.11}$$

or equivalently:

$$\begin{bmatrix} u_1 & u_2 & \dots & u_P \\ v_1 & v_2 & \dots & v_P \end{bmatrix} = \begin{bmatrix} l_1 R & \dots & l_K R \end{bmatrix} \cdot \begin{bmatrix} S_1 \\ S_2 \\ \dots \\ S_K \end{bmatrix} \tag{2.12}$$

This can then be extended to handle multiple views of the subject taken over a monocular video sequence. Let $(u_i^{(t)}, v_i^{(t)})$ denote the tracked 2D point at timestep t. This gives rise to the following system, taken over N timesteps:

$$\underbrace{\begin{bmatrix} u_i^{(1)} & \dots & u_P^{(1)} \\ v_i^{(1)} & \dots & v_P^{(1)} \\ u_i^{(2)} & \dots & u_P^{(2)} \\ v_i^{(2)} & \dots & v_P^{(2)} \\ & \dots & \\ u_i^{(N)} & \dots & u_P^{(N)} \\ v_i^{(N)} & \dots & v_P^{(N)} \end{bmatrix}}_{W} = \underbrace{\begin{bmatrix} l_1^{(1)} R^{(1)} & \dots & l_K^{(1)} R^{(1)} \\ l_1^{(2)} R^{(2)} & \dots & l_K^{(2)} R^{(2)} \\ & \dots & \\ l_1^{(N)} R^{(N)} & \dots & l_K^{(2)} R^{(N)} \end{bmatrix}}_{Q} \cdot \underbrace{\begin{bmatrix} S_1 \\ S_2 \\ \dots \\ S_K \end{bmatrix}}_{B} \tag{2.13}$$

This shows the tracking matrix W can be factored into 2 matrices: Q which contains the camera pose $R^{(t)}$ and configuration weights $l_1^{(t)}, \dots, l_K^{(t)}$ per frame t. B encodes the K basis shapes S_i. This system can be factored with singular value decomposition to yield the shape basis S_i, per-frame camera rotations R and per-frame configuration weights l. A number of techniques follow this formulation [9, 9, 9], but start with a shape basis learnt from available motion capture datasets (e.g. CMU [9]).

More recent approaches were designed to be fully automatic. Shotton et al. [111] designed a commerically-available system for 3D human skeletal tracking which required a depth sensor. A generative 3D body model was used to synthesize a large training dataset of depth images with corresponding body part labels. Density estimators for each body part are then used in combination to localize body joints with a calculated confidence value. Taylor et al. [128] predict dense correpsondences between image pixels (again, with depth so in $\mathbb{R}^3$) and a representative 3D human body model, again by training on synthetic depth images. Chapter 3 of this thesis demonstrates a technique for predicting keypoints by training on synthetic *silhouette* data, rendered from an animal deformable body model, which overcomes the need for depth imagery at test time.

Automatic monocular approaches often take advantage of 2D keypoint or body part detectors when reasoning about 3D skeletons. Simo-Serra et al. [9, 9] form a probabilistic model that models both 3D pose and 2D keypoints jointly, overcoming noise among 2D

body parts. Other approaches [9, 9] employ a two-stage pipeline; they begin by localizing 2D joint positions on an input image before running a subsequent optimization step that 'lifts' these to a 3D pose. Tangential work [9] takes uses detected 2D joints to perform a nearest neighbour search in a 3D mocap dataset. The most recent two-stage pipelines rely on deep convolutional networks to predict keypoints. Examples of such systems include DeepPose [134], an approach which employs a CNN to reason jointly about 2D landmark detection and 3D pose estimation from single RGB images. Pishchulin et al. [97] later introduced DeepCut which extends DeepPose to the multi-person case.

State-of-the-art techniques now operate as a direct regression to a 3D pose. Most often, paired 3D training data (such as is available from datasets such as Human3.6M [9]) is required which is generally expensive to obtain, particularly for animal categories. One branch of approaches [129] predicts body configuration in terms of angles. Other approaches include Pavlakos et al. [9], who use a 2D joint predictor [9] followed by a deep architecture to regress 3D heatmaps. Moreno-Noguer [9] learn a pairwise distance matrix from 2D-to-3D space in order to allow unlikely 3D predictions to be ruled out with a suitable prior. These techniques were designed under the assumption that neural networks would struggle to learn a 'lifting' function from 2D to 3D pose. This assumption was corrected by Martinez et al. [9] who demonstrate the effectiveness of a simple architecture at regressing accurate 3D keypoints from 2D predictions. This technique was later interpreted probabilistically by The technique was interpreted probabilistically by Li et al. [9], who handled ambiguity in the 2D-to-3D lifting problem with a mixture density network. Related work that predicts a depth segmentation (so not strictly 3D keypoints) is SURREAL [9] who train their network with data generated synthetically with a 3D human body model.

2.5.2 Model-based human shape and pose

This section will discuss methods for reconstructing a full 3D *dense* human from a monocular image or video sequence. Early work in this category fit shape primitives combined into a kinematic tree to silhouettes extracted from the input [9, 9, 9]. The introduction of the 3D deformable human body model known as SCAPE [9] enabled various fitting approaches. Sigal et al [9] compute shape features from manually extracted silhouettes and use a mixture of experts formulation for predicting SCAPE model parameters. Later, Guan et al. [9] fit the SCAPE model to provided keypoints, extracted silhouettes, edges and shading cues. They also define an interpenetration term that penalizes self-intersecting body parts, although this does not lead to easy optimization. Hasler et al. [9], Zhou et al. [9] and Chen et al. [9] present a similar approach, although show optimization only to input keypoints and manually or semi-manually (e.g. GraphCut [9]) extracted silhouettes.

A significant advance was made by the introduction of SMPLify [9], the first fully-automatic method for monocular 3D human pose and shape reconstruction. Many of the concepts presented by SMPLify are used throughout this thesis, making it worthy of study.

Fitting a 3D model to 2D keypoints

SMPLify works by fitting the SMPL [73] model to a set of 2D image locations predicted by DeepCut [9], a deep convolutional neural network. For an input image I, DeepCut predicts a set of image keypoint locations $J_{\text{est}} \in \mathbb{R}^{23\times 2}$ which correspond to locations on the 3D SMPL mesh J_{SMPL}. Precisely $J_{\text{SMPL}} = R_\theta(J(\beta))$ where $J(\beta)$ computes 3D skeleton positions from SMPL shape parameters β, and R_θ is the global rigid transformation effected by SMPL pose parameters θ. A model fitting approach is then used to align the SMPL model to the predicted keypoint positions. This is achieved through optimizing the SMPL parameters (β,θ), global translation t and camera parameters K, subject to priors over pose, shape and limb interpenetration priors.

The key energy term used in the optimization (and indeed throughout this thesis) is given by $E_{\text{J}}(\beta,\theta;K,J_{\text{est}})$ and measures the weighted 2D distance between estimated keypoints J_{est} and the corresponding SMPL joints J_{SMPL}.

$$E_J(\beta,\theta,t;K,J_{\text{est}}) = \sum_{\text{joint},i} w_i \rho(\Pi_K(J_{\text{SMPL,i}} - J_{\text{est,i}})) \tag{2.14}$$

The weighted 2D distance is implemented using the Geman-McClure [9] penalty function ρ which helps deal with noisy DeepCut estimates. SMPLify implements Π_K perspective camera model with known (or roughly initialized) focal length although others opt for orthographic projection. The following definition provides a quick primer for this:

Definition 2.5.1 (Primer on camera geometry)**.** Perspective projection is a function which maps a 3D structure to blah blah.

$$2X = Y \tag{2.15}$$

Orthographic projection is the following:

$$3X = Z \tag{2.16}$$

The full energy formulation is then given as:

$$E(\beta,\theta) = E_J(\beta,\theta;K,J_{\text{est}}) + \lambda_\theta E_\theta(\theta) + \lambda_\alpha E_\alpha(\theta) + \lambda_{\text{sp}} E_{\text{sp}}(\theta;\beta) + \lambda_\beta E_\beta(\beta) \tag{2.17}$$

where the following energy terms are employed, balanced according to the λ scalar weights:

- $E_\theta(\theta)$ is referred to as a *pose prior* which favours more likely poses by assigning large punishment to those that deviate from known poses collected from a large dataset.
- $E_\beta(\beta)$ is referred to as a *shape prior* which favours more likely pose-invariant shape configurations by assigning large punishment to those that deviate from known shapes collected from a large dataset.
- $E_\alpha(\theta)$ is a *joint limit* prior which ensures particular joints remain within acceptable angle limits. For example, a knee joint in a human model should be prohibited from bending more than 5 degrees upwards.
- $E_{sp}(\theta;\beta)$ is an *interpenetration* term, which can only be defined in such shape modelling approaches. Using both shape and pose from the model, it is possible to determine if any limbs are self-intersecting, or intersect other parts of the body and assign appropriate penalty.

SMPLify has recently undergone subsequent variations, including BLAH, BLAH, and an application to 3D quadruped reconstruction discussed later. Chapter 3 of this thesis will introduce a *self-supervised* version of SMPLify that uses synthetic data for training, thereby overcoming the need for a large 2D dataset with manually-labelled keypoints.

An example result can be seen in Figure 2.6:

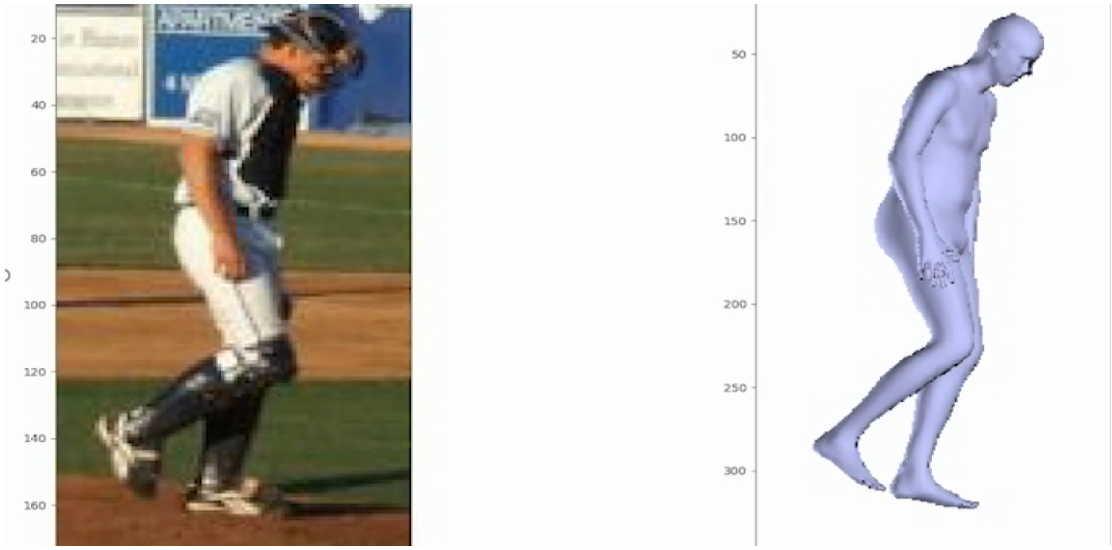


Fig. 2.6 SMPLify: Fitting the SMPL model to the Leeds Sports Dataset.

Direct regression

The most recent, and state-of-the-art approaches employ deep learning techniques to solve the entire optimization problem by directly regressing shape and pose parameters of the template model. Tan et al. [126] present a technique they term *indirect learning* which learns an encoding $f : \mathbb{R}^{H\times W} \mapsto (\theta, \beta, t, K)$ of input images to SMPL pose and shape, translation and camera parameters. Their method makes use of a *silhouette renderer* $R : (V, T) \mapsto \{0,1\}^{H\times W}$ (learnt using synthetic data) capable of producing a binary silhouette from a predicted SMPL mesh. In this way, R allows the network's SMPL predictions to be supervised to ensure generated silhouettes match ground-truth annotations. Kanazawa et al [9] extended this work to with their Human Mesh Recovery [**?**] paper, making use of a learnt differentiable renderer and adding 2D/3D keypoint losses (similar to Eqt 99).

The abundance of available human data has supported the development of successful monocular 3D reconstruction pipelines [64, 56]. Such approaches rely on accurate 3D data to build detailed priors over the distribution of human shapes and poses, and use large 2D keypoints datasets to promote generalization to "in-the-wild" scenarios. Silhouette data has also been shown to assist in accurate reconstruction of clothes, hair and other appearance detail [105, 4]. While the dominant paradigm in human reconstruction is now end-to-end deep learning methods, SPIN [63] show impressive improvement by incorporating an energy minimization process within their training loop to further minimize a 2D reprojection loss subject to fixed pose & shape priors. Inspired by this innovation, we learn an iteratively-improving shape prior by applying expectation maximization during the training process.

Of course, these techniques are typically data hungry, requiring not only 3D morphable models but also 3D supervision per training image and large

Model-free techniques

Talk about PiFu + animals + articulated Nerf.

2.5.3 Model-based animals

We want to do model based reconstruction to impose a prior on the fitting and also automatically recover an interpretable fit.

Table 2.1 summarizes previous work on animal reconstruction. It is interesting to note that while several papers demonstrate reconstruction across species, which *prima facie* is a richer class than just dogs, the test-time requirements (e.g. manually-clicked keypoints/silhouette segmentations, input image quality etc.) are considerably higher for those systems. Thus we claim that the achievement of reconstructing a full range of dog breeds, with variable

Paper	Animal Class	Training requirements	Template Model	Video required	Test Time Annotation	Model Fitting	Test Size
This paper	Dogs	J2, S2, T3, P3	SMAL	No	None	No	1703
3D-Safari [156]	Zebras, horses	M3 (albeit synthetic), J2, S2, P3	SMAL	3-7 frames / animal	None	Yes	200
Lions, Tigers and Bears (SMALR) [157]	MLQ	Not trained	SMAL	3-7 frames / animal	J2, S2	Yes	14
3D Menagerie (SMAL) [158]	MLQ	Not trained	SMAL	No	J2, S2	Yes	48
Creatures Great and SMAL [10]	MLQ	Not trained	SMAL	Yes	S2 (for best results shown)	Yes	9
Category Specific Mesh Reconstructions [58]	Birds	J2, S2	Bird convex hull	No	None	No	2850
What Shape are Dolphins [20]	Dolphins, Pigeons	Not trained	Dolphin Template	25 frames / category	J2, S2	Yes	25
Animated 3D Creatures [**?**]	MLQ	Not trained	Generalized Cylinders	Yes	J2, S2	Yes	15

Table 2.1 Literature summary: Our paper extends large-scale "in-the-wild" reconstruction to the difficult class of diverse breeds of dogs. MLQ: Medium-to-large quadrupeds. J2: 2D Joints. S2: 2D Silhouettes. T3: 3D Template. P3: 3D Priors. M3: 3D Model.

fur length, varying shape and pose of ears, and with considerable occlusion, is a significant contribution.

A major impediment to research in 3D animal reconstruction has been the lack of a strong evaluation benchmark, with most of the above methods showing only qualitative evaluations or providing quantitative results on fewer than 50 examples. To remedy this, we introduce *StanfordExtra*, a new large-scale dataset which we hope will drive further progress in the field.

While animals are often featured in computer vision literature, there are still relatively few works that focus on accurate 3D animal reconstruction.

A primary reason for this is the lack of large scale 3D datasets stemming from the practical challenges associated with 3D motion capture, as well as a lack of 2D data which captures a wide variety of animals. The recent Animal Pose dataset [17] is one such 2D alternative, but contains significantly fewer labelled images than our new StanfordDogs dataset (4,000 compared to 20,580 in). On the other hand, animal silhouette data is plentiful [68, 30, 61].

Learning animal shape from unrelated 2D images

Cashman and Fitzgibbon [20] introduce an optimization technique able to recover a parameterized, morphable 3D model from unrelated 2D images depicting examples of the

target class. The method requires user-supplied 2D object outlines and point constraints for each image, and a single rigid mesh for the entire object class. The authors demonstrate recovering an 8-parameter morphable dolphin model from 32 images obtained from Google. To reduce required user activity, it is reasonable to assume that given sufficient labelled training data, it would be simple to manipulate a convolutional network architecture able to perform foreground / background segmentation and identify human key points (say, joints) for the desired object class. The system achieves impressive results when optimizing over both pose and shape parameters across a range of object classes, but struggles for articulated models such as polar bears.

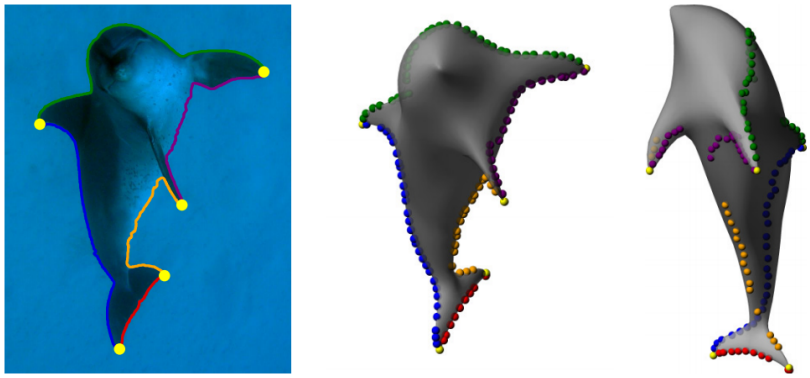
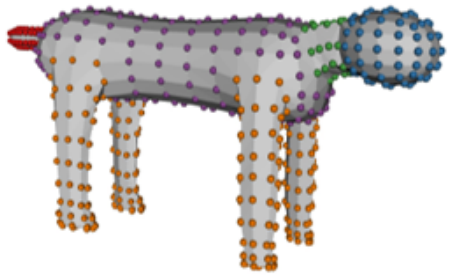


Fig. 2.7 8-parameter dolphin model with annotated contour (left) and contour generators (middle and right).

Fitting to animal video sequences (Stebbing)

Stebbing et al. [120] introduce a technique capable of fitting a template mesh to live video sequences for a range of different animal species. Some user interaction is required in order to segment the animal from the background and to provide sparse 3D-mesh-to-2D-image key point correspondences. This work only operates on input video sequences (rather than single frames), so a number of temporal terms are incorporated that encourage sensible inter-frame model deviation. The system requires an annotated input template mesh representative of the target animal species. Note that this work does not require the template mesh to have an inner skeletal structure. However, the user assists an ARAP-style term by assigning each mesh vertex v_i to one of M groups which share a set of basis rotations B_m.



(a) Template mesh with joint movement constraints.

(b) Example of user supplied point tracks.

Fig. 2.8 User input required for the deformable mesh animation algorithm, reprinted from [120].

Through reasonably accurate pose fitting and by allowing some pose-invariant shape deformation, this work produces smooth meshes which are often a good match to the input video. Moreover, their experimentation shows that ARAP is a useful prior for reconstructing articulated, non-rigid motion in instances that an internal skeleton is a priori unknown. However, the shape attributes for the reconstructed model are not particularly accurate, which results in frequent errors appearing at internal occluding contours. In addition, the large non-convex optimization algorithm is an expensive operation, taking around 1 minute per video frame on a standard Linux workstation.

Results showing this work fitting a crude dog template mesh to a sample video obtained from YouTube are shown previously in Figure 1.2. Figure 2.9 shows another example, which operates on a template impala mesh.

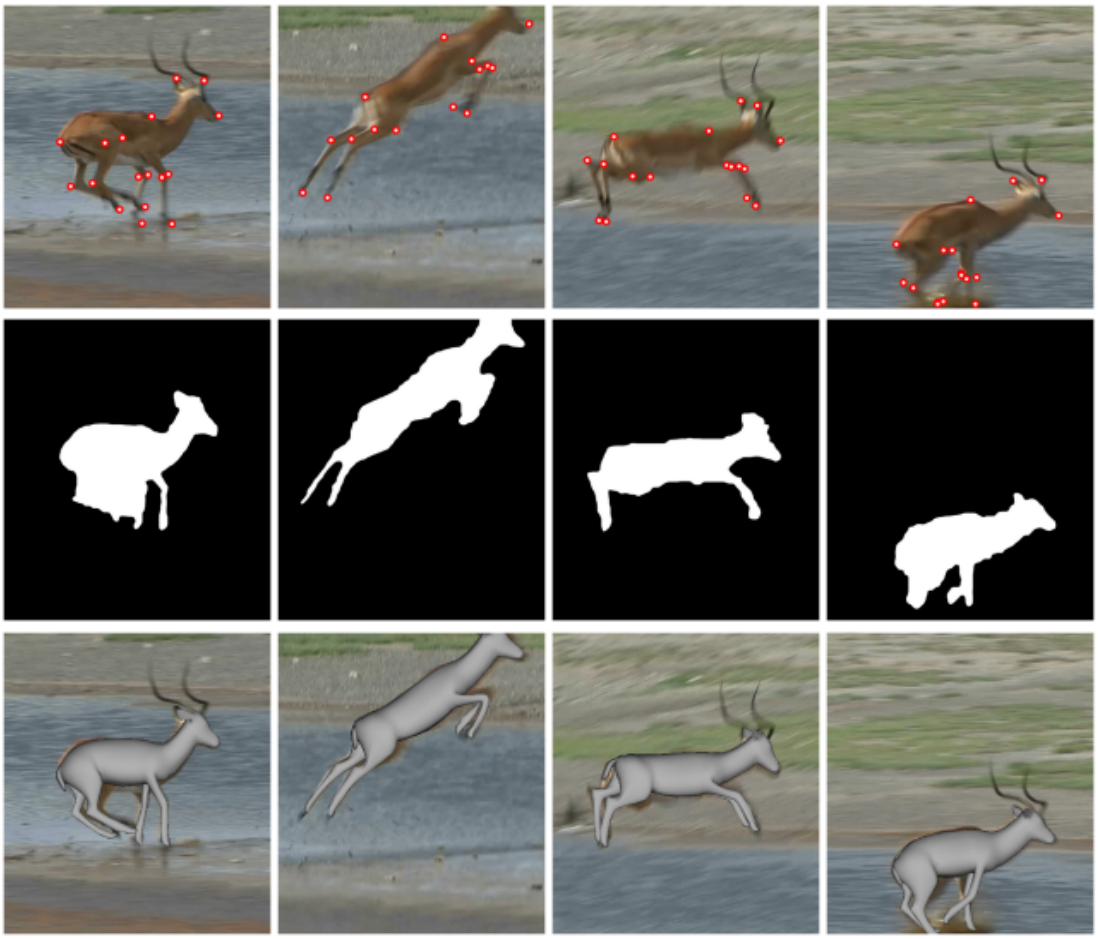


Fig. 2.9 Example of an impala template being fit to input video sequence, reprinted from [120]

Fitting the SMAL mesh to animal images

The SMAL paper briefly discusses a modification to the SMPLify approach in order to fit the SMAL model to RGB animal input images. The terms are largely the same, although the interpenetration term is omitted and joint positions are provided manually, rather than being predicted by a CNN. Finally, the optimizer requires a pre-segmented (i.e. silhouette) image which is also supplied by a user. An approach discussed in Chapter 4 builds on this work, so an in-depth description of this method is ommited here. However, an example result showing the result of the optimizer fitting the SMAL mesh to an RGB image of a fox can be seen in Figure 2.10. Note that the whole optimization process takes around 1 minute per frame.

Zuffi et al. [158] made a significant contribution to 3D animal reconstruction research by releasing SMAL, a deformable 3D quadruped model (analagous to SMPL [73] for human reconstruction) from 41 scans of artist-designed toy figurines. The authors also released shape and pose priors generated from artist data. In this work we develop *SMBLD*, an extension

of SMAL that better represents the diverse dog category by adding scale parameters and refining the shape prior using our large image dataset.

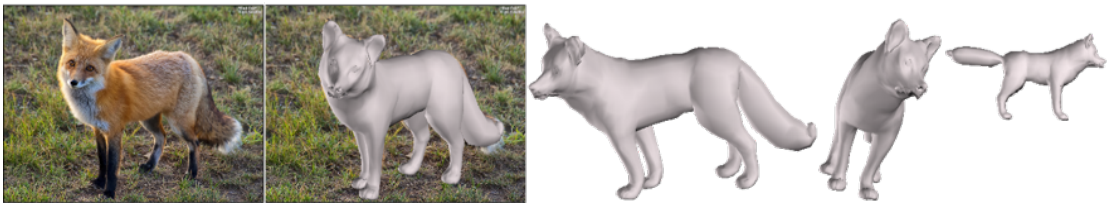


Fig. 2.10 Fitting SMAL to a hand segmented animal, reprinted from [158].

Definition 2.5.2 (Differentiable Rendering)**.** The process of generating a 2D image from a 3D polygon mesh is known as rendering and can be achieved through a process known as raytracing. Raytracing is a rendering technique able to generate photorealistic 2D images from the scene. It can be considered the opposite process by which the human eye perceives the world, as this method involves lines being cast outwards, beginning at a point known as the *camera origin*. Figure 2.11 shows a typical set up, in which rays are cast from the camera origin through each pixel on the image plane. The colour for the pixel is obtained by following the ray through the scene until a light source or non-reflective surface is reached, taking into account any reflections or non-opaque scene items. Due to the considerable comptuation required, the operation is often parallelized and assigned to the GPU. However, the technique is typically considered unsuitable for real-time rendering of complex scenes (due to complex ray paths) or when high resolution images (many rays required) are needed. However, for this work, scenes are typically made up of a single non-reflective, solid mesh surface and contain no complex elements (e.g. shadows, non-constant lighting.

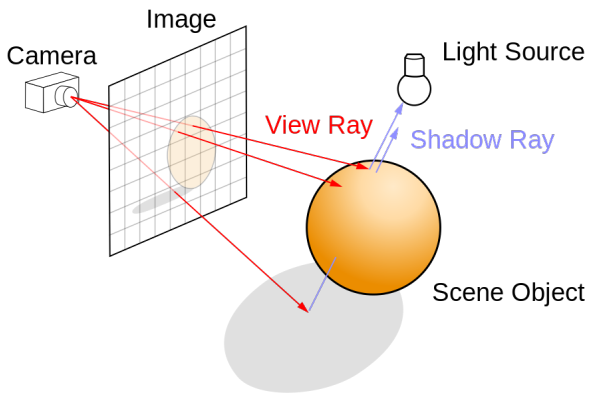


Fig. 2.11 Diagram showing raycast rendering. [144].

It is also worth noting that the standard method for raycasting is not differentiable, causing problems for differentiable optimizers (including neural networks). However, alternative rendering methods [74] are available for these purposes.

The SMAL authors [158] demonstrate fitting their deformable 3D model to quadruped species using user-provided keypoint and silhouette dataset. SMALR [157] then demonstrated fitting to broader animal categories by incorporating multi-view constraints from video sequences. 3D-Safari [156] further improve by training a deep network on synthetic data (built using SMALR [157]) to recover detailed zebra shapes in the wild. Chapter 4 of this this will present a method that overcomes the need for hand-clicked keypoints by training a joint predictor on synthetic data.

A drawback of these approaches is their reliance on a test-time energy-based optimization procedure, which is susceptible to failure with poor quality keypoint/silhouette predictions and increases the computational burden. Chapter 5 of this method presents an automatic reconstruction method that overcomes the need for additional energy-based refinement, and is trained purely from single in-the-wild images.

Model-free techniques

While there have been various "model-free" approaches which do not rely on an initial template model to generate the 3D animal reconstruction, these techniques often do not produce a mesh [2, 86] or rely heavily on input 2D keypoints or video at test-time [139, 99]. An exception is the end-to-end network of Kanazawa et al. [58], although we argue that the bird category exhibits more limited articulation than our dog category.

Chapter 3

Learning from Synthetic Data; Bridging the Domain Gap

3.1 Introduction

We present a system to recover the 3D shape and motion of a wide variety of quadrupeds from video. The system comprises a machine learning front-end which predicts candidate 2D joint positions, a discrete optimization which finds kinematically plausible joint correspondences, and an energy minimization stage which fits a detailed 3D model to the image. In order to overcome the limited availability of motion capture training data from animals, and the difficulty of generating realistic synthetic training images, the system is designed to work on silhouette data. The joint candidate predictor is trained on synthetically generated silhouette images, and at test time, deep learning methods or standard video segmentation tools are used to extract silhouettes from real data. The system is tested on animal videos from several species, and shows accurate reconstructions of 3D shape and pose.

We address this problem using techniques from the recent human body and hand tracking literature, combining machine learning and 3D model fitting. A discriminative front-end uses a deep hourglass network to identify candidate 2D joint positions. These joint positions are then linked into coherent skeletons by solving an optimal joint assignment problem, and the resulting skeletons create an initial estimate for a generative model-fitting back-end to yield detailed shape and pose for each frame of the video.

For human tracking, hand labelled sequences of 2D segmentations and joint positions have been collected from a wide variety of sources [5, 68, 52]. Of these two classes of labelling, animal *segmentation* data is available in datasets such as MSCOCO [68], PASCAL VOC [30] and DAVIS [93]. However this data is considerably sparser than human data,

and must be "shared" across species, meaning the number of examples for a given animal shape class is considerably fewer than is available for an equivalent variation in human shape. While segmentation data can be supplied by non-specialist human labellers, it is more difficult to obtain *joint position* data. Some joints are easy to label, such as "tip of snout", but others such as the analogue of "right elbow" require training of the operator to correctly identify across species.

Of more concern however, is 3D skeleton data. For humans, motion capture (mocap) can be used to obtain long sequences of skeleton parameters (joint positions and angles) from a wide variety of motions and activities. For animal tracking, this is considerably harder: animals behave differently on treadmills than in their quotidian environments, and although some animals such as horses and dogs have been coaxed into motion capture studios [145], it remains impractical to consider mocap for a family of tigers at play.

These concerns are of course alleviated if we have access to synthetic training data. Here, humans and animals share an advantage in the availability of parameterized 3D models of shape and pose. The recent publication of the Skinned Multi-Animal Linear (SMAL) model [158] can generate a wide range of quadruped species, although without surface texture maps. However, as with humans, it remains difficult to generate RGB images which are sufficiently realistic to train modern machine learning models. In the case of humans, this has been overcome by generating depth maps, but this then requires a depth camera at test time [110]. The alternative, used in this work, is to generate 2D silhouette images so that machine learning will predict joint heatmaps from silhouettes only.

Taking into account the above constraints, this work applies a novel strategy to animal tracking, which assumes a machine-learning approach to extraction of animal silhouettes from video, and then fits a parameterized 3D model to silhouette sequences. We make the following contributions:

- A machine-learned mapping from silhouette data of a large class of quadrupeds to generic 2D joint positions.
- A novel optimal joint assigment (OJA) algorithm extending the bipartite matching of Cao *et al.* [19] in two ways, one which can be cast as a quadratic program (QP), and an extension optimized using a genetic algorithm (GA).
- A procedure for optimization of a 3D deformable model to fit 2D silhouette data and 2D joint positions, while encouraging temporally coherent outputs.
- We introduce a new benchmark animal dataset of joint annotations (BADJA) which contains sparse keypoint labels and silhouette segmentations for eleven animal video

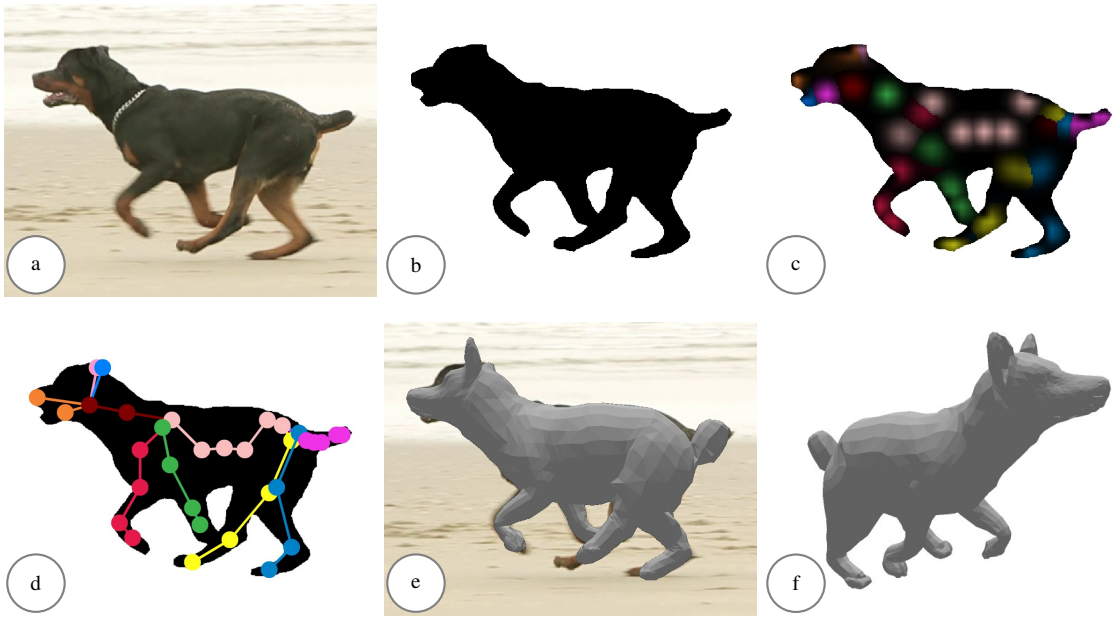


Fig. 3.1 **System overview**: input video (a) is automatically processed using DeepLabv3+ [23] to produce silhouettes (b), from which 2D joint predictions are regressed in the form of heatmaps (c). Optimal joint assignment (OJA) finds kinematically coherent 2D-to-3D correspondences (d), which initialize a 3D shape model, optimized to match the silhouette (e). Alternative view shown in (f).

sequences. Previous work in 3D animal reconstruction has relied on bespoke hand-clicked keypoints [158, 157] and little quantitative evaluation of performance could be given. The sequences exhibit a range of animals, are selected to capture a variety of animal movement and include some challenging visual scenarios such as occlusion and motion blur.

The system is outlined in Fig. 3.1. The remainder of the paper describes related literature before a detailed description of system components. Joint accuracy results at multiple stages of the pipeline are reported on the new BADJA dataset, which contains ground truths for real animal subjects. We also conduct experiments on synthetic animal videos to produce joint accuracy statistics and full 3D mesh comparisons. A qualitative comparison is given to recent work [158] on the related single-frame 3D shape and pose recovery problem. The paper concludes with an assessment of strengths and limitations of the work.

3.1.1 Related Work

In this section, we will discuss work specifically related to this paper. The broad

Non-automatic approaches for 3D reconstruction of articulated subjects

Much of this is handled in the previous section, but could be briefly recapped. Point out that we want an AUTOMATIC technique that can rapidly apply across different quadruped types.

Automatic approaches for 3D reconstruction of articulated subjects

The SMPLify approach is comprises of two stages, means of achieving 3D human reconstruction. an automatic To train such a system, we may look towards the SMPLify technique employed for 3D human mesh recovery

Learning from synthetic data

SURREAL dataset for humans.

Cleaning up predictions

Talk about pictorial structure models. Could be integrated later.

3.2 Design discussion for an automatic quadruped 3D reconstruction

The test-time problem to be solved is to take a sequence of input images and for each image, output the shape, pose and position parameters describing the animal's motion.

To train such a system, we start by considering the suitability of the SMPLify [14] method (discussed above) which describes a two-stage approach for automatic 3D human reconstruction. However, the approach has particular design requirements that prevents trivial extension to reconstructing quadrupeds.

3.2.1 Keypoint training data for joint predictor

Firstly, the preliminary stage of the approach is based on the DeepCut joint predictor, a convolutional neural network that takes an input an image and predicts a set of semantically meaningful keypoints. In the case of SMPLify, the network is trained on large-scale human keypoint datasets, including MPII [5] (approx. 40,000 people) and the Leeds Sport Dataset [52]. As previously discussed, there are no keypoint datasets for animals that cover even a small fraction of the quadruped types that we aim to reconstruct.

With no real-world data to use for training the keypoint predictor, we can turn instead to a novel approach that instead relies on synthetic (or fake) data for training.

3.2.2 Data driven shape and pose priors

Of course, we also don't have any shape or pose priors.

3.3 Preliminaries

3.3.1 Deformable 3D quadruped model

This section will formally define the deformable 3D model that is used to generate synthetic training data and will also be used in the model fitting stage to obtain the final mesh. We are given a deformable 3D model such as SMAL [158] which parametrizes a 3D mesh as a function of *pose* parameters $\theta \in \mathbb{R}^P$ (e.g. joint angles) and *shape* parameters $\beta \in \mathbb{R}^B$. In detail, a 3D mesh is an array of vertices $v \in \mathbb{R}^{3\times V}$ (the vertices are columns of a $3 \times V$ matrix) and a set of triangles represented as integer triples (i, j, k), which are indices into the vertex array. A deformable model such as SMPL or SMAL may be viewed as supplying a set of triangles, and a function

$$v(\theta, \beta) : \mathbb{R}^P \times \mathbb{R}^B \mapsto \mathbb{R}^{3\times V} \tag{3.1}$$

which generates the 3D model for a given pose and shape. The mesh topology (i.e. the triangle vertex indices) is provided by the deformable model, and is the same for all shapes and poses we consider, so in the sequel we shall consider a mesh to be defined only by the 3D positions of its vertices.

In any given image, the model's 3D *position* (i.e. translation and orientation) is also unknown, and will be represented by a parametrization ϕ which may be for example translation as a 3-vector and rotation as a unit quaternion. Application of such a transformation to a $3 \times V$ matrix will be denoted by $*$, so that

$$\phi * v(\theta, \beta) \tag{3.2}$$

represents a 3D model of given pose and shape transformed to its 3D position.

We will also have occasion to talk about model *joints*. These appear naturally in models with an explicit skeleton, but more generally they can be defined as some function mapping from the model parameters to an array of 3D points analogous to the vertex transformation

above. We consider the joints to be defined by post-multiplying by a $V \times J$ matrix $\mathtt{K}$. The j^{th} column of $\mathtt{K}$ defines the 3D position of joint j as a linear combination of the vertices (this is quite general, as v may include vertices not mentioned in the triangulation).

3.3.2 Camera model, joint reprojection and silhouette rendering

A general camera model is described by a function $\pi : \mathbb{R}^3 \mapsto \mathbb{R}^2$. This function incorporates details of the camera intrinsics such as focal length, which are assumed known. Thus

$$\kappa(\phi, \theta, \beta) := \pi(\phi * v(\theta, \beta)\mathtt{K}) \tag{3.3}$$

is the $2 \times J$ matrix whose columns are 2D joint locations corresponding to a 3D model specified by (position, pose, shape) parameters (ϕ, θ, β).

The model is also assumed to be supplied with a rendering function R which takes a vertex array in camera coordinates, and generates a 2D binary image of the model silhouette. That is,

$$R\big(\phi * v(\theta, \beta)\big) \in \mathbb{B}^{W \times H} \tag{3.4}$$

for an image resolution of $W \times H$. We use the differentiable renderer of Loper *et al.* [74] to allow derivatives to be propagated through R.

3.4 Joint prediction using synthetic data

The test-time problem to be solved is to take a sequence of input images $\mathscr{I} = [I_t]_{t=1}^{T}$ which are segmented to the silhouette of a single animal (i.e. a video with multiple animals is segmented multiple times), producing a sequence of binary silhouette images $\mathscr{S} = [S_t]_{t=1}^{T}$.

The computational task is to output for each image the shape, pose, and position parameters describing the animal's motion.

As outlined above, the method has three parts. (1.) The discriminative front-end extracts silhouettes from video, and then uses the silhouettes to predict 2D joint positions, with multiple candidates per joint. (2.) Optimal joint assignment (OJA) corrects confused or missing skeletal predictions by finding an optimal assignment of joints from a set of network-predicted proposals. Finally, (3.) a generative deformable 3D model is fitted to the silhouettes and joint candidates as an energy minimization process.

3.4.1 Prediction of 2D joint locations

The goal of the first stage is to take, for each video frame, an image reperesenting the animal and to output a $W \times H \times J$ tensor of heatmaps. The network architecture is standard: a stacked hourglass network [85] using synthetically generated training data, but the training procedure is augmented using "multi-modal" heatmaps.

3.4.2 Bridging the domain gap

We begin with a set of pose, shape and position parameters sampled according to the following procedure.

Sampling

. The random camera positions are generated as follows: the orientation of the camera relative to the animal is uniform in the range $[0, 2\pi]$, the distance from the animal is uniform in the range 1 to 20 meters and the camera height is in the range $[0, \frac{\pi}{2}]$. This smaller range is chosen to restrict unusual camera elevation. Finally, the camera "look" vector is towards a point uniformly in a 1m cube around the animal's center, and the "up" vector is Gaussian around the model Y axis.

What about texture?

Unfortunately, the application of texture is not entirely clear.

We can try and follow a similar process such as SURREAL and access a dataset of textures drawn from real images.

3.4.3 Prediction of 2D joint locations using multimodal heatmaps

For standard unimodal heatmaps, training data comprises (S, κ) pairs, that is pairs of binary silhouette images, and the corresponding 2D joint locations as a $2 \times J$ matrix. To generate each image, a random shape vector β, pose parameters θ and camera position ϕ are drawn, and used to render a silhouette $R\big(\phi * v(\theta, \beta)\big)$ and 2D joint locations $\kappa(\phi, \theta, \beta)$, which are encoded into a $W \times H \times J$ tensor of heatmaps, blurring with a Gaussian kernel of radius σ.

This training process generalizes well from synthetic to real images due to the use of the silhouette, but the lack of interior contours in silhouette input data often results in confusion between joint "aliases": left and right or front and back legs. When these predictions are wrong and of high confidence, little probability mass is assigned to the area around the correct leg, meaning no available proposal is present after non-maximal suppression.

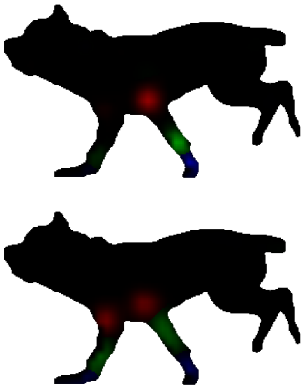


Fig. 3.2 Example predictions from a network trained on unimodal (top) and multi-modal (bottom) ground-truth for front-left leg joints.

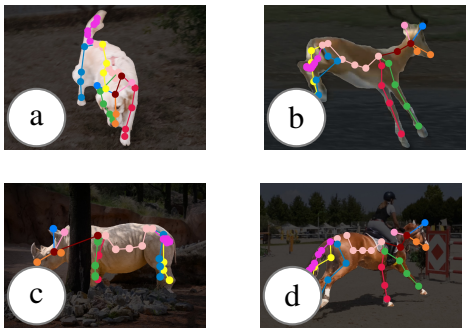


Fig. 3.3 Example outputs from the joint prediction network, with maximum likelihood predictions linked into skeleton.

We overcome this by explicitly training the network to assign some probability mass to the "aliased" joints. For each joint, we define a list of potential aliases as weights $\lambda_{j,j'}$ and linearly blend the unimodal heatmaps G to give the final training heatmap H:

$$H_j(p) = \sum_{j'} \lambda_{j,j'} G(p; \kappa_{j'}, \sigma) \tag{3.5}$$

For non-aliased joints j (all but the legs), we simply set $\lambda_{j,j} = 1$ and $\lambda_{j,j'} = 0$, yielding the unimodal maps, and for legs, we use 0.75 for the joint, and 0.25 for the alias. We found this ratio sufficient to ensure opposite legs have enough probability mass to pass through a modest non-maximal suppression threshold without overly biasing the skeleton with maximal predicted confidence. An example of a heatmap predicted by a network trained on multimodal training samples is illustrated in Fig. 3.2.

3.5 Optimal joint assignment (OJA)

Since heatmaps generated by the joint predictor are multi-modal, the non-maximum suppression procedure yields multiple possible locations for each joint. We represent the set of joint proposals $X = \{x_{jp}\}$, where x_{jp} indicates the 2D position of proposal $p \in \{1,...,N_j\}$ associated with joint $j \in J$. Before applying the optimizer, we must select a subset of proposals $X^* \subseteq X$ which form a complete skeleton, i.e. precisely one proposal is selected for every joint. In this section we consider how to choose the optimal subset by formulating the problem as an extended optimal assignment problem.

In order to select a complete skeleton proposal from the set of joint proposals $\{x_{jp}\}$, we introduce a binary indicator vector $\bar{a}_j = \{a_{jp}\} \in \{0,1\}^{N_j+1}$, where $a_{jp} = 1$ indicates that the p^{th} proposal for joint j is a correct assignment, and the $p = N_j + 1$ position corresponds to a *null proposal*, indicating that joint j has no match in this image. The null proposals are handled as described in each of the energy terms below. Let A be the jagged array $[\bar{a}_j]_{j=1}^{J}$ containing all assignment variables (for the current frame), and let $X^* = X(A)$ denote the subset of points selected by the binary array A. Optimal assignment minimizes the function

$$L(A) = L_{\text{prior}}(A) + L_{\text{conf}}(A) + L_{\text{temp}}(A) + L_{\text{cov-sil}}(A) + L_{\text{cov-bone}}(A) \tag{3.6}$$

which balances agreement of the joint configuration with a learned *prior*, the network-supplied *confidences*, *temporal* coherence, and *coverage* terms which encourage the model to correctly project over the silhouette. Without the coverage terms, this can be optimized as a quadratic program, but we obtain better results by using the coverage terms, and using a genetic algorithm. In addition, the parameters A must satisfy the J constraints $\sum_{p=1}^{N_j+1} a_{jp} = 1$, that exactly one joint proposal (or the null proposal) must be selected for each joint.

L_{prior}:

We begin by defining the prior probability of a particular skeletal configuration as a multivariate Gaussian distribution over selected joint positions.

The mean $\mu \in \mathbb{R}^{2J}$ and covariance $\Sigma \in \mathbb{R}^{2J \times 2J}$ terms are obtained from the training examples generated as above. The objective of OJA is to select a configuration X^* which maximizes the prior, which is equivalent to minimizing the Mahalanobis distance $(x^* - \mu)^T \Sigma^{-1} (x^* - \mu)$, which is given by the summation

$$L_{\text{prior}}(A) = \sum_{j}^{J} \sum_{p}^{N_j} \sum_{k}^{J} \sum_{q}^{N_k} a_{jp} a_{kq} (x_{jp} - \mu_j) \Sigma_{jk}^{-1} (x_{kq} - \mu_k) \tag{3.7}$$

This is a quadratic function of A, so $L_{\text{prior}}(A) = \text{vec}(A)^\top Q \text{vec}(A)$ for a fixed matrix Q, and can be formulated as a quadratic program (QP). Null proposals are simply excluded from the sum, equivalent to marginalizing over their position.

L_{conf}:

The next energy term comes from the output of the joint prediction network, which provides a confidence score y_{jp} associated with each joint proposal x_{jp}. Then $L_{\text{conf}}(A) = \sum_j \sum_p -\lambda \log(y_{jp}) a_{jp}$ is a linear function of A, and λ_{conf} is a tunable parameter to control

the relative contribution of the network confidences compared with that of the skeleton prior. Null proposals pay a fixed cost λ_{null}, effectively acting as a threshold whereby the null proposal will be selected if no other proposal is of sufficient likelihood.

L_{temp}:

A common failure case of the joint prediction network is in situations where a joint position is highly ambiguous, for example between the left and right legs. In such cases, the algorithm will commonly alternate between two equally likely predictions. This leads to large displacements in joint positions between consecutive frames which are difficult for the later model fitting stage to recover from. This can be addressed by introducing a temporal term into the OJA. We impose a prior on the distance moved by each joint between frame t_0 and t_1, which is given by a normal distribution with zero mean and variance $\sigma^2 = e^{\tau|t_1-t_0-1|}$. The parameter τ controls the strength of the interaction between distant frames. This results in an additional quadratic term in our objective function, which has the form $L_{temp} = a^\top T^{(t_0,t_1)} a$ for matrix $T^{(t_0,t_1)}$ given by

$$\left[T^{(t_0,t_1)}\right]_{jp,kq} = \begin{cases} e^{-\alpha|t_1-t_0-1|}||x_{jp}^{(t_0)} - x_{kq}^{(t_1)}||^2 & \text{if } j = k \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

3.5.1 QP solution.

Thus far, all terms in $L(A)$ are quadratic or linear. To optimize over an entire sequence of frames, we construct the block diagonal matrix $\hat{Q}$ whose diagonal elements are the prior matrices $Q^{(t)}$ and the block symmetric matrix $\hat{T}$ whose off-diagonal elements are the temporal matrices $T^{(t_0,t_1)}$. The solution vector for the sequence $\hat{A}$ is constructed by stacking the corresponding vectors for each frame. The quadratic program is specified using the open source CVXPY library [28] and solved using the "*Suggest-and-Improve*" framework proposed by Park and Boyd [89]. It is initialized by choosing the proposal with the highest confidence for each joint. Appropriate values for the free parameters $\lambda_{\text{conf,temp}}$ and α were chosen empirically via grid search.

$L_{\text{cov-\{sil,bone\}}}$:

The above quadratic formulation is sufficient to correct many errors in the raw output (which we later demonstrate in the experimental section), but suffers from an 'overcounting' problem, in which leg joint predictions both cover the same silhouette leg region, leaving another leg empty. We therefore extend the definition of $L(A)$ to include two additional terms.

Fig. 3.4 Silhouette coverage loss. The error (shown in red) is the the distance between the median axis transform (right) and the nearest point on an approximate rendering (left).

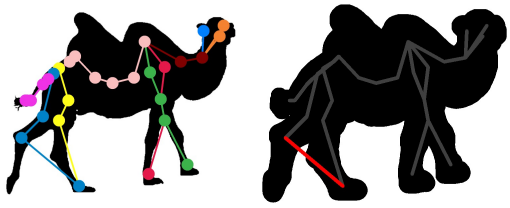


Fig. 3.5 Bone coverage loss. One of the back-right leg joints is incorrectly assigned (left), leading to a large penalty since the lower leg bone crosses outside the dilated silhouette (right).

$L_{\text{cov-sil}}$:

penalizes large silhouette areas with no nearby selected joint. This term requires a precomputed set of silhouette sample points $Z \subseteq \mathbb{R}^2$, which we aim to "cover" as best as possible with the set of selected joints. Intuitively, the silhouette is considered well-covered if all sample points are close to *some* selected joint proposal. The set Z is generated from the medial axis transform (MAT)[13] of the silhouette, $Z^t = \text{MAT}(S^t)$ with a cubed loss strongly penalizing projection outside the silhouette:

$$L_{\text{cov-sil}}(A^t; X^t, Z^t) = \sum_i \min_j \|Z_i^t - \hat{X}_j^t\|^3 \tag{3.9}$$

$L_{\text{cov-bone}}$:

is used to prevent bones crossing the background. The joint hierarchy is stored in a kinematic tree structure $K = \{\{j,k\}$ if joints j,k are connected by a bone$\}$.

$$L_{\text{cov-bone}}(A^t; X^t, S^t, K) = \sum_{\{j,k\} \in K} \left(1 - \min_{\lambda \in [0:0.1:1]} S^t(\hat{X}_j^t + \lambda(\hat{X}_j^t - \hat{X}_k^t))\right) \tag{3.10}$$

3.5.2 GA Solution.

We minimize this more complex objective using a genetic algorithm (GA)[46], which requires defining a fitness function, "genes", an initial population, crossover procedure, and mutation

procedure. The *fitness function* is precisely the energy $L(A)$ given above, and the *genes* are vectors of J integers, rather than one-hot encodings. We begin with a population size of 128 genes, in which the first 32 are set equal to the max confidence solutions given by the network in order to speed up convergence. The remaining 96 are generated by selecting a random proposal for each joint. *Crossover* is conducted as standard by slicing genes in two parts, and pairing first and second parts from different parents to yield the next generation. In each generation, each gene has some probability of undergoing a *mutation*, in which between 1 and 4 joints have new proposals randomly assigned. Weights were set empirically and we run for 1000 generations. Examples of errors corrected by these two energy terms are shown in Fig. 3.4 and Fig. 3.5.

3.6 Generative model optimization

The generative model optimization stage refines model parameters to better match the silhouette sequence $\mathscr{S}$, by minimizing an energy which sums 4 terms:

Silhouette energy.

The silhouette energy E_{sil} compares the rendered model to a given silhouette image, given simply by the L2 difference between the OpenDR rendered image and the given silhouette:

$$E_{\text{sil}}(\phi,\theta,\beta;S) = \|S - R\big(\phi * v(\theta,\beta)\big)\| \tag{3.11}$$

Unimodal Prior energy.

The prior term E_{prior} encourages the regressed shape and pose parameters to remain close to a those in the combined artist traininthose in our set of artist 3D dog meshes.

The Mahalanobis distance is used to encourage the model to remain close to: (1) a distribution over shape coefficients given by the mean and covariance of SMAL training samples of the relevant animal family, (2) a distribution of pose parameters built over a walking sequence. The final term ensures the pose parameters remain within set limits.

$$E_{\text{lim}}(\theta) = \max\{\theta - \theta_{\max}, 0\} + \max\{\theta_{\min} - \theta, 0\}. \tag{3.12}$$

Joints energy.

The joints energy E_{joints} compares the rendered model joints to the OJA predictions, and therefore must account for missing and incorrect joints. It is used primarily to stabilize the nonlinear optimization in the initial iterations, and its importance is scaled down as the silhouette term begins to enter its convergence basin.

$$E_{\text{joints}}(\phi, \theta, \beta; X^*) = \|X^* - \phi * v(\theta, \beta)\mathtt{K}(:, j)\| \tag{3.13}$$

Temporal energy.

The optimizer for each frame is initialized to the result of that previous. In addition, a simple temporal smoothness term is introduced to penalize large inter-frame variation:

$$E_{\text{temp}}(\phi, \theta, \beta; X^*) = (\phi_t - \phi_{t+1})^2 + (\beta_t - \beta_{t+1})^2 \tag{3.14}$$

The optimization is via a second order dogleg method [75].

3.7 Experiments

In order to quantify our experiments, we introduce a new benchmark animal dataset of joint annotations (BADJA) comprising several video sequences with 2D joint labels and segmentation masks.

3.7.1 BADJA Dataset

These sequences were derived from the DAVIS video segmentation dataset [93], as well as additional online stock footage for which segmentations were obtained using Adobe's UltraKey tool [1]. A set of twenty joints on the 3D SMAL mesh were labeled, illustrated in Fig. 3.6. These joints were chosen on the basis of being informative to the skeleton and being simple for a human annotator to localize. To make manual annotation feasible and to ensure a diverse set of data, annotations are provided for every fifth frame.

The video sequences were selected to comprise a range of different quadrupeds undergoing various movement typical of their species. Although the dataset is perhaps insufficient in size to train deep neural networks, the variety in animal shape and pose renders it suitable for evaluating quadruped joint prediction methods.

3.7.2 Joint prediction

For the joint predictor ρ we train a stacked hourglass network [85]. Following state-of-the-art performance on related human 2D pose estimation datasets ([5, 68]), we construct a network consisting of 8 stacks, 256 features and 1 block. As input we provide synthetically-generated silhouette images of size 256×256, which are obtained by randomly sampling shape and pose parameters from the SMAL model. The corresponding training targets are ground truth heatmaps produced by smoothing the 2D projected joint locations with a Gaussian kernel. Since we are working with synthetic data, we are able to generate training samples on the fly, resulting in an effectively infinite training set. A small adaptation was required to prevent the network degenerating to an unfavourable solution on silhouette input: foreground masks were applied to both ground truth silhouette and predicted heatmaps to prevent the network degenerating to an all-zero heatmap, which produces a reasonably good loss and prevents the network training successfully. The network was trained using the RMSProp optimizer for 40k iterations with a batch size of 18 and learning rate of 2.5×10^{-4}. The learning rate was decayed by 5% every 10k iterations. Training until convergence took 24 hours on a Nvidia Titan X GPU.

Joint accuracy is evaluated with the Probability of Correct Keypoint (PCK) metric defined by Yang and Ramanan [150]. The PCK is the percentage of predicted keypoints which are within a threshold distance d from the ground truth keypoint location. The threshold distance is given by $d = \alpha\sqrt{|S|}$ where $|S|$ is the area of the silhouette and α is a constant factor which we set to $\alpha = 0.2$ for these experiments.

Fig. 3.3 shows a selection of maximum likelihood joint predictions on real world images. Note that despite being trained only on synthetic data, the network generalizes extremely well to animals in the wild. The performance extends even to species which were not present in the SMAL model, such as the impala and rhino. The network is also robust to challenging poses (3.3b), occlusions (3.3c) and distraction objects such as the human rider in (3.3d). It is however susceptible to situations where the silhouette image is ambiguous, for example if the animal is facing directly towards or away from the camera. Figure 3.11 contains examples of failure modes.

3.7.3 Optimal joint assignment

Following non-maximum suppression of the joint heatmaps obtained in Section 3.7.2, we apply OJA to select an optimal set of joints with which to initialize the final optimization stage. It can be seen that the OJA step is able to address many of the failure cases introduced by the joint prediction network, for example by eliminating physically implausible joint

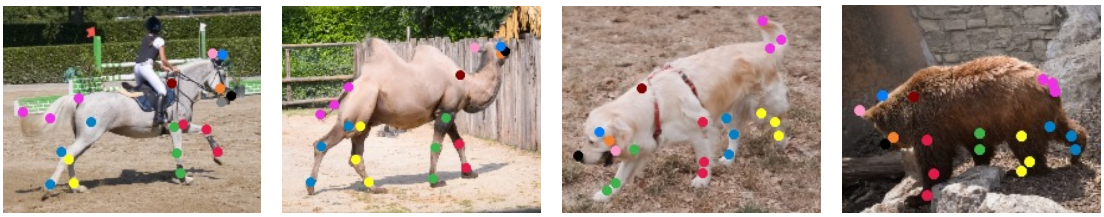


Fig. 3.6 Example joint annotations from the BADJA dataset. A total of 11 video sequences are in the dataset, annotated every 5 frames with 20 joint positions and visibility indicators.

(a) (b) (c)

Fig. 3.7 Example skeletons from raw predictions (a), processed with OJA-QP (b), and OJA-GA (c).

	Raw	QP	GA
bear	83.1	83.7	**88.9**
camel	73.3	74.1	**87.1**
cat	58.5	**60.1**	58.4
cows	89.2	88.4	**94.7**
dog	**66.9**	66.6	**66.9**
horsejump-high	26.5	**27.7**	24.4
horsejump-low	26.9	27.0	**31.9**
tiger	76.5	88.8	**92.3**
rs_dog	64.2	63.4	**81.2**
Average	62.8	64.4	**69.5**

Table 3.1 Accuracy of OJA on BADJA test sequences.

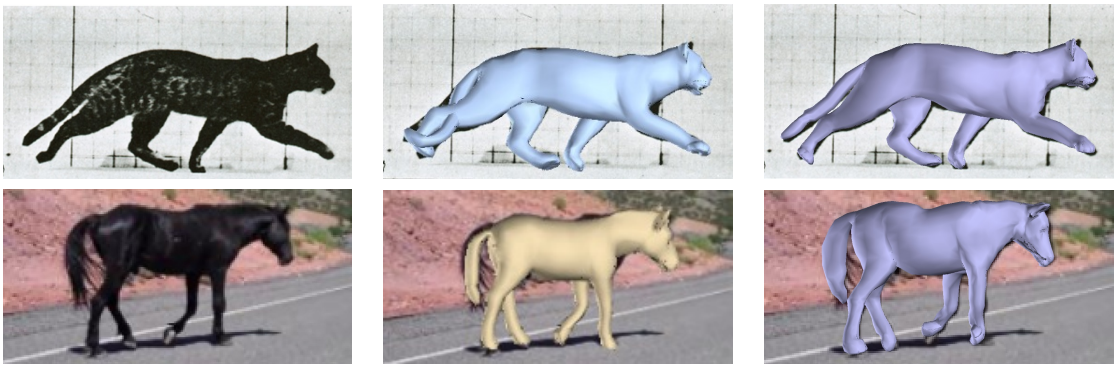
configurations (Fig. 3.7, row 1) or by resolving the ambiguity between the left and right legs (Fig. 3.7, row 2). Table 3.1 summarizes the performance of both the raw network predictions and results of the two OJA methods. Over most of the sequences in the BADJA dataset it can be seen that the use of coverage terms (employed by the OJA-GA model) improves skeleton accuracy. In particular, the bear, camel and rs_dog sequences show substantial improvements. The method does however struggle on the horsejump_high sequence, in which part of the silhouette is occluded by the human rider which adversely affects the silhouette coverage term. Across all sequences the selected OJA-GA method improves joint prediction accuracy by 7% compared to the raw network output.

3.7.4 Model fitting

The predicted joint positions and silhouette are input to the optimization phase, which proceeds in four stages. The first stage solves for the model's global rotation and translation

Seq.	Family	PCK (%)		Mesh	Seq.	Family	PCK (%)		Mesh
		Raw	OJA-GA				Raw	OJA-GA	
01	Felidae	91.8	91.9	38.2	06	Equidae	84.4	84.8	19.2
02	Felidae	94.7	95.0	42.4	07	Bovidae	94.6	95.0	40.6
03	Canidae	87.7	88.0	27.3	08	Bovidae	85.2	85.8	41.5
04	Canidae	87.1	87.4	22.9	09	Hippopotamidae	90.5	90.6	11.8
05	Equidae	88.9	89.8	51.6	10	Hippopotamidae	93.7	93.9	23.8

Table 3.2 Quantitative evaluation on synthetic test sequences. We evaluate the performance of the raw network outputs and quadratic program post-processing using the probability of correct keypoint (PCK) metric (see sec. 3.7.2). We evaluate mesh fitting accuracy by computing the mean distance between the predicted and ground truth vertices.



RGB SMAL [158] **Ours**

Fig. 3.8 Our results are comparable in quality to SMAL [158], but note that we do not require hand-clicked keypoints.

parameters, which positions the camera. We follow SMPLify [14] by solving this camera stage for torso points only, which remain largely fixed through shape and pose variation. We then solve for all shape, pose and translation parameters and gradually decrease the emphasis of the priors. The silhouette term is introduced in the penultimate stage, as otherwise we find this can lead to the optimizer finding unsatisfactory local minima.

The final outputs of our optimization pipeline are shown in Fig. 3.10. In each of the cases illustrated the optimizer is able to successfully find a set of pose and shape parameters which, when rendered, closely resembles the input image. The final row of Fig. 3.10 demonstrates the generalizability of the proposed method: the algorithm is able to find a reasonable pose despite no camel figurines being included in the original SMAL model.

Fig. 3.9 Evaluating synthetic data. Green models: ground truth, Orange models: predicted. Frames 5, 10 and 15 of sequence 4 shown. Error on this sequence 22.9.

Comparison to other work.

We compare our approach visually to that given by Zuffi *et al.* [158]. Recall that their results require hand-clicked keypoints whereas ours fits to points predicted automatically by the hourglass network, which was trained on synthetic animal images. Further, their work is optimized for single frame fitting and is tested on animals in simple poses, whereas we instead focus on the more challenging task of tracking animals in video. Fig. 3.8 shows the application of our model to a number of single frame examples from the SMAL result data [158].

Quantitative experiments.

There is no existing ground truth dataset for comparing reconstructed 3D animal meshes, but an estimate of quantitative error is obtained by testing on synthetic sequences for a range of quadruped species. These are generated by randomly deforming the model and varying the camera position to animate animal motion, see Figure 3.9. Table 3.2 shows results on these sequences.

3.7.5 Automatic silhouette prediction

While not the main focus of our work, we are able to perform the full 3D reconstruction process from an input image with no user intervention. We achieve this by using the DeepLabv3+ network [23] as a front-end segmentation engine to automatically generate animal silhouettes. This network was trained on the PASCAL VOC 2012 dataset, which includes a variety of animal quadruped classes. An example result generated using the fully automatic pipeline is shown in Fig. 3.1.

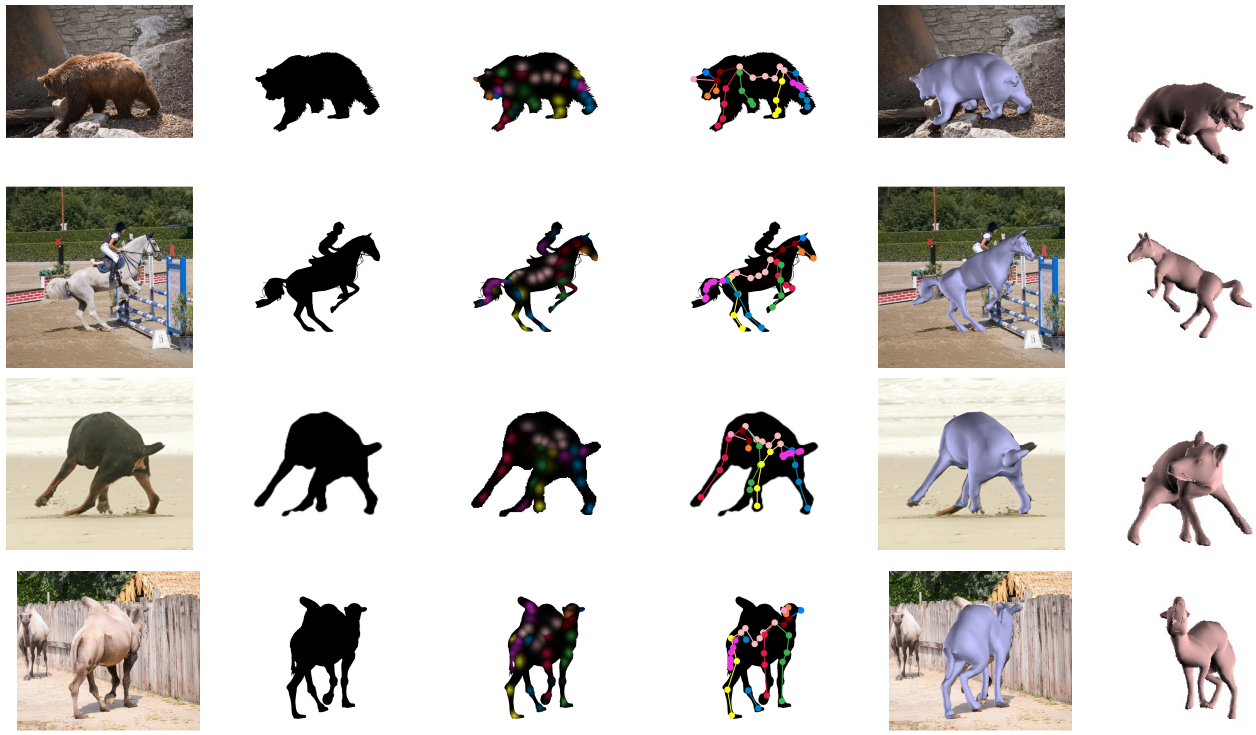


Fig. 3.10 Example results on various animals. From left to right: RGB input, extracted silhouette, network-predicted heatmaps, OJA-processed joints, overlay 3D fit and alternative view.

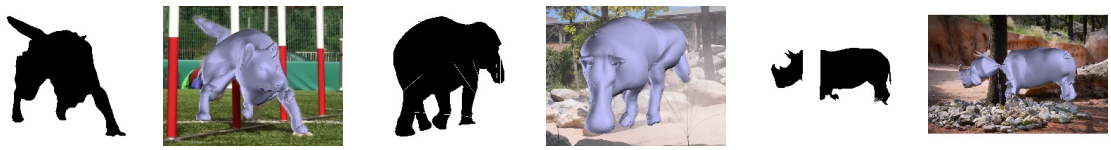


Fig. 3.11 Failure modes of the proposed system. *Left*: Missing interior contours prevent the optimizer from identifying which way the dog is facing. *Middle*: The model has never seen an elephant, so assumes the trunk is the tail. *Right*: Heavy occlusion. The model interprets the tree as background and hence the silhouette term tries to minimize coverage over this region.

3.8 Conclusions

In this work we have introduced a technique for 3D animal reconstruction from video using a quadruped model parameterized in shape and pose. By incorporating automatic segmentation tools, we demonstrated that this can be achieved with no human intervention or prior knowledge of the species of animal being considered. Our method performs well on

examples encountered in the real world, generalizes to unseen animal species and is robust to challenging input conditions.

Chapter 4

End-to-end Dog Shape Recovery with a Learned Shape Prior

4.1 Introduction

We introduce an automatic, end-to-end method for recovering the 3D pose and shape of dogs from monocular internet images. The large variation in shape between dog breeds, significant occlusion and low quality of internet images makes this a challenging problem. We learn a richer prior over shapes than previous work, which helps regularize parameter estimation. We demonstrate results on the Stanford Dog Dataset, an "in-the-wild" dataset of 20,580 dog images for which we have collected 2D joint and silhouette annotations to split for training and evaluation. In order to capture the large shape variety of dogs, we show that the natural variation in the 2D dataset is enough to learn a detailed 3D prior through expectation maximisation (EM). As a by-product of training, we generate a new parameterized model (including limb scaling) SMBLD which we release alongside our new annotation dataset *StanfordExtra* to the research community.

A particular species of interest is the dog, however it is noticeable that existing work has not yet demonstrated effective 3D reconstruction of dogs over large test sets. We postulate that this is partially because dog breeds are remarkably dissimilar in shape and texture, presenting a challenge to the current state of the art. The methods we propose extend the state of the art in several ways. While each of these qualities exist in some existing works, we believe ours is the first to exhibit this combination, leading to a new state of the art in terms of scale and object diversity.

1. We reconstruct pose and shape on a test set of 1703 low-quality internet images of a complex 3D object class (dogs).

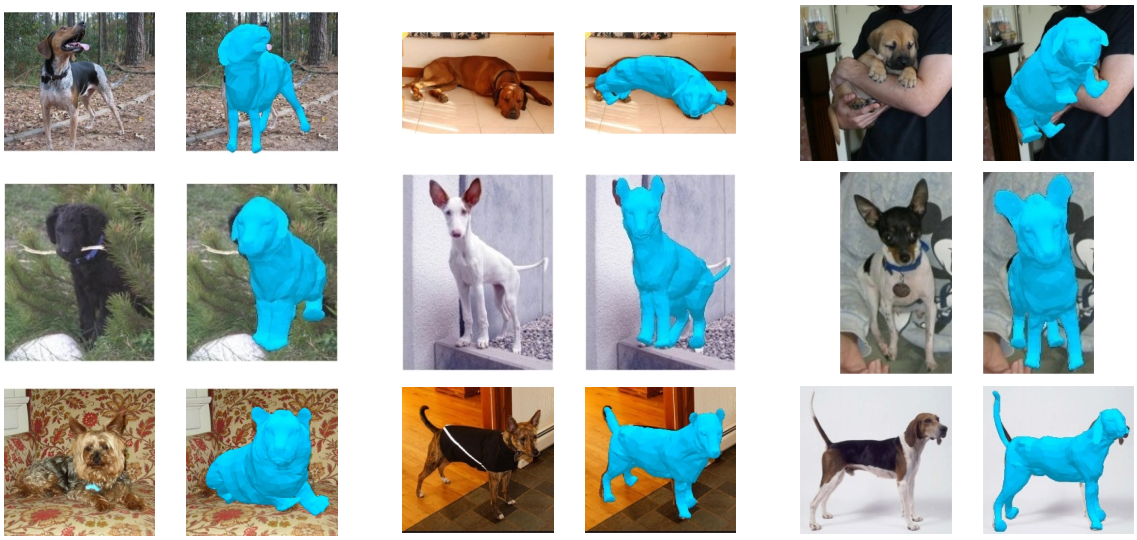


Fig. 4.1 **End-to-end Dog Shape Recovery with a Learned Shape Prior.** We propose a novel method that, given a monocular image of a dog can predict a set of parameters for our SMBLD 3D dog model which is consistent with the input. We regularize learning using a multi-modal shape prior, which is tuned during training with an expectation maximization scheme.

2. We directly regress to object pose and shape from a single image without a model fitting stage.

3. We use easily obtained 2D annotations in training, and none at test time.

4. We incorporate fitting of a new multi-modal prior into the training phase (via EM update steps), rather than fitting it to 3D data as in previous work.

5. We introduce new degrees of freedom to the SMAL model, allowing explicit scaling of subparts.

4.1.1 Related work

We will discuss some related techniques for this paper.

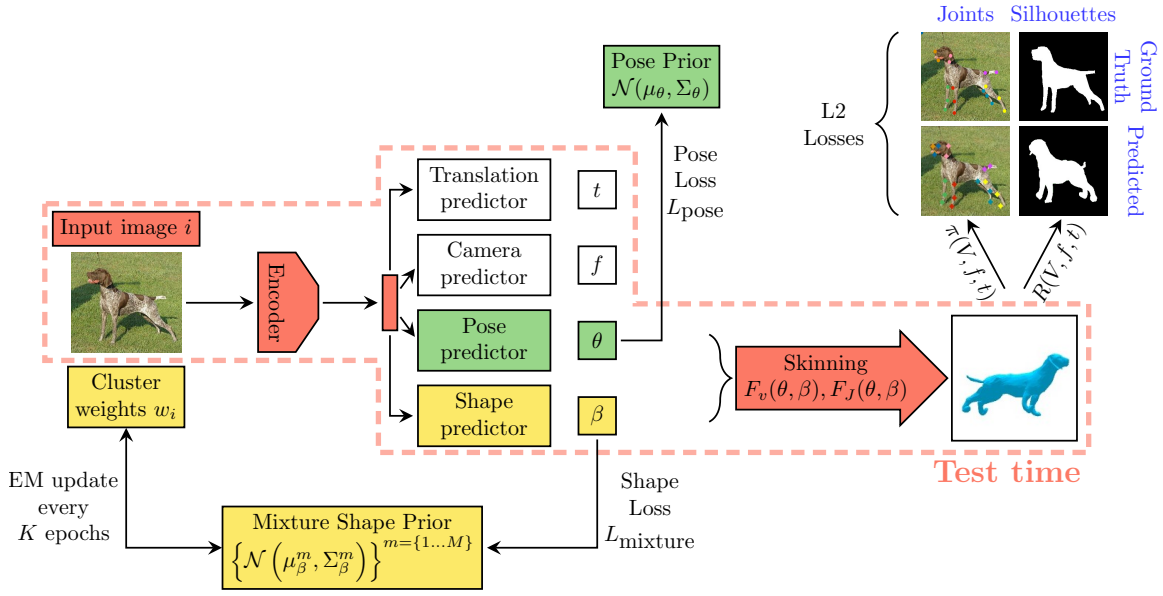


Fig. 4.2 Our method consists of (1) a deep CNN encoder which condenses the input image into a feature vector (2) a set of prediction heads which generate SMBLD parameters for shape β, pose θ, camera focal length f and translation t (3) skinning functions F_v and F_J which construct the mesh from a set of parameters, and (4) loss functions which minimise the error between projected and ground truth joints and silhouettes. Finally, we incorporate a mixture shape prior (5) which regularises the predicted 3D shape and is iteratively updated during training using expectation maximisation. At test time, our system (1) condenses the input image, (2) generates the SMBLD parameters and (3) constructs the mesh.

Automatic 3D reconstruction approaches

Animal datasets

4.2 Building SMBLD: a new parametric dog model

At the heart of our method is a parametric representation of a 3D animal mesh, which is based on the Skinned Multi-Animal Linear (SMAL) model proposed by [158]. SMAL is a deformable 3D animal mesh parameterized by shape and pose. The *shape* $\beta \in \mathbb{R}^B$ parameters are PCA coefficients of an undeformed template mesh with limbs in default position. The *pose* $\theta \in \mathbb{R}^P$ parameters meanwhile govern the joint angle rotations (35×3 Rodrigues parameters) which effect the articulated limb movement. The model consists of a linear blend skinning function $F_v : (\theta, \beta) \mapsto V$, which generates a set of vertex positions $V \in \mathbb{R}^{3889\times 3}$, and a joint function $F_J : (\theta, \beta) \mapsto J$, which generates a set of joint positions $J \in \mathbb{R}^{35\times 3}$.

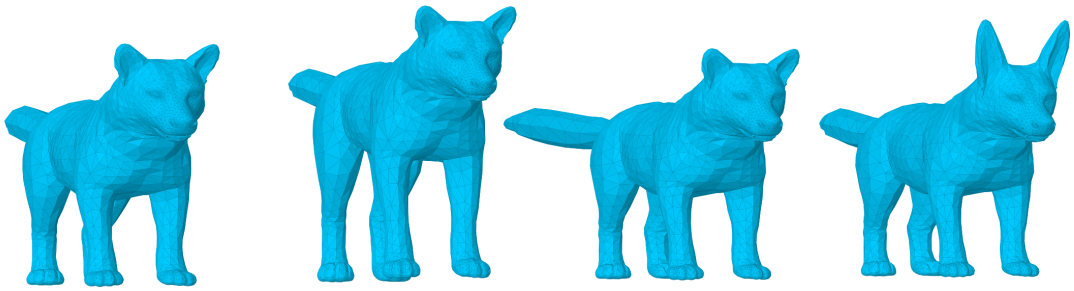


Fig. 4.3 **Effect of varying SMBLD scale parameters**. *From left to right*: Mean SMBLD model, 25% leg elongation, 50% tail elongation, 50% ear elongation.

4.2.1 Introducing scale parameters

While SMAL has been shown to be adequate for representing a variety of quadruped types, we find that the modes of dog variation are poorly captured by the current model. This is unsurprising, since SMAL used only four dogs in its construction.

We therefore introduce a simple but effective way to improve the model's representational power over this particularly diverse animal category. We augment the set of shape parameters β with an additional set κ which independently scale parts of the mesh. For each model joint, we define parameters $\kappa_x, \kappa_y, \kappa_z$ which apply a local scaling of the mesh along the local coordinate x, y, z axes, before pose is applied. Allowing each joint to scale entirely independently can however lead to unrealistic deformations, so we share scale parameters between multiple joints, e.g. leg lengths. The new Skinned Multi-Breed Linear Model for Dogs (SMBLD) is therefore adapted from SMAL by adding 6 scale parameters to the existing set of shape parameters. Figure 4.3 shows how introducing scale parameters increases the flexibility of the SMAL model. We also extend the provided SMAL shape prior (which later initializes our EM procedure) to cover the new scale parameters by fitting SMBLD to a set of 13 artist-designed 3D dog meshes. Further details left to the supplementary.

4.2.2 Building a 3D shape prior via model fitting

Another method for improving the generalizability of the SMAL model is to improve the 3D shape prior. Such priors are typically used to ensure shape deformation remain within a realistic and anatomically plausible range. Due to the limited diversity of scans used to build the SMAL model, while the shape prior does enforce realism among deformations, it does not allow for a wide enough range to cover the set of dogs in our dataset.

We improve the quality of the prior (and learn a prior over our new scale parameters) by fitting to a set of 13 artist-designed 3D dog meshes, which are more varied than the original

set. We apply an energy minimization scheme which aligns the SMAL vertices to each scan, under smoothing regularizers. Further details left to the supplementary.

4.3 End-to-end dog reconstruction from monocular images

We now consider the task of reconstructing a 3D dog mesh from a monocular image. We achieve this by training an end-to-end convolutional network that predicts a set of SMBLD model and perspective camera parameters. In particular, we train our network to predict pose θ and shape β SMBLD parameters together with translation t and focal length f for a perspective camera. A complete overview of the proposed system is shown in Figure 4.2.

4.3.1 Model architecture

Our network architecture is inspired by the model of 3D-Safari [156]. Given an input image cropped to (224, 224), we apply a Resnet-50 [44] backbone network to encode a 1024-dimensional feature map. These features are passed through various linear prediction heads to produce the required parameters. The pose, translation and camera prediction modules follow the design of 3D-Safari, but we describe the differences in our shape module.

Pose, translation and camera prediction.

These modules are independent multi-layer perceptrons which map the above features to the various parameter types. As with 3D-Safari we use two linear layers to map to a set of 35×3 3D pose parameters (three parameters for each joint in the SMBLD kinematic tree) given in Rodrigues form. We use independent heads to predict camera frame translation $t_{x,y}$ and depth t_z independently. We also predict the focal length of the perspective camera similarly to 3D-Safari.

Shape and scale prediction.

Unlike 3D-Safari, we design our network to predict the set of shape parameters (including scale) rather than vertex offsets. We observe improvement by handling the standard 20 blend-shape parameters and our new scale parameters in separate linear prediction heads. We retrieve the scale parameters by $\kappa = \exp x$ where x are the network predictions, as we find predicting log scale helps stabilise early training.

4.3.2 Training losses

A common approach for training such an end-to-end system would be to supervise the prediction of (θ, β, t, f) with 3D ground truth annotations [63, 56, 91]. However, building a suitable 3D annotation dataset would require an experienced graphics artist to design an accurate ground truth mesh for each of 20,520 StanfordExtra dog images, a prohibitive expense.

We instead develop a method that instead relies on *weak 2D supervision* to guide network training. In particular, we rely on only 2D keypoints and silhouette segmentations, are significantly cheaper to obtain.

The rest of this section describes the set of losses used to supervise the network at train time.

Joint reprojection.

The most important loss to promote accurate limb positioning is the joint reprojection loss L_{joints} which compares the projected model joints $\pi(F_J(\theta, \beta), t, f)$ to the ground truth annotations $\hat{X}$. Given the parameters predicted by the network, we apply the SMBLD model to transform the pose and shape parameters into a set of 3D joint positions $J \in \mathbb{R}^{35\times 3}$, and project them to the image plane using translation and camera parameters. The joint loss L_{joints} is given by the ℓ_2 error between the ground truth and projected joints:

$$L_{\text{joints}}(\theta, \beta, t, f; \hat{X}) = \|\hat{X} - \pi(F_J(\theta, \beta), t, f)\|_2 \tag{4.1}$$

Note that many of our training images exhibit significant occlusion, so $\hat{X}$ contains many invisible joints. We handle this by masking L_{joints} to prevent invisible joints contributing to the loss.

Silhouette loss.

The silhouette loss L_{sil} is used to promote shape alignment between the SMBLD dog mesh and the input dog. In order to compute the silhouette loss, we define a rendering function $R : (v, t, f) \mapsto S$ which projects the SMBLD mesh to produce a binary segmentation mask. In order to allow derivatives to be propagated through R, we implement R using the differentiable Neural Mesh Renderer [59]. The loss is computed as the ℓ_2 difference between a projected silhouette and the ground truth mask $\hat{S}$:

$$L_{\text{sil}}(\theta, \beta, t, f; \hat{S}) = \|\hat{S} - R\big(F_V(\theta, \beta), t, f\big)\|_2 \tag{4.2}$$

Priors.

In the absence of 3D ground truth training data, we rely on priors obtained from artist graphics models to encourage realism in the network predictions. We model both pose and shape using a multivariate Gaussian prior, consisting of means μ_θ, μ_β and covariance matrices $\Sigma_\theta, \Sigma_\beta$. The loss is given as the log likelihood of a given shape or pose vector under these distributions, which corresponds to the Mahalanobis distance between the predicted parameters and their corresponding means:

$$L_{\text{pose}}(\theta; \mu_\theta, \Sigma_\theta) = (\theta - \mu_\theta)^T \Sigma_\theta^{-1} (\theta - \mu_\theta) \tag{4.3}$$

$$L_{\text{shape}}(\beta; \mu_\beta, \Sigma_\beta) = (\beta - \mu_\beta)^T \Sigma_\beta^{-1} (\beta - \mu_\beta) \tag{4.4}$$

Unlike previous work, we find there is no need to use a loss to penalize pose parameters if they exceed manually specified joint angle limits. We suspect our network learns this regularization naturally because of our large dataset.

4.3.3 Learning a multi-modal shape prior.

The previous section introduced a unimodal, multivariate Gaussian shape prior, based on mean μ_β and covariance matrix Σ_β. However, we find enforcing this prior throughout training tends to result in predictions which appear similar in 3D shape, even when tested on dog images of different breeds. We propose to improve diversity among predicted 3D dog shapes by extending the above formulation to a Mixture of M Gaussians prior. The mixture shape loss is then given as:

$$L_{\text{mixture}}(\beta_i; \mu_\beta, \Sigma_\beta, \Pi_\beta) = \sum_{m=1}^{M} \Pi_\beta^m L_{\text{shape}}(\beta_i; \mu_\beta^m, \Sigma_\beta^m) \tag{4.5}$$

Where μ_β^m, Σ_β^m and Π_β^m are the mean, covariance and mixture weight respectively for Gaussian component m. For each component the mean is sampled from our existing unimodal prior and the covariance is set equal to the unimodal prior i.e. $\Sigma_\beta^m := \Sigma_\beta$. All mixture weights are initially set to $\frac{1}{M}$.

Each training image i is assigned a set of latent variables $\{w_i^1, \ldots w_i^M\}$ encoding the probability of the dog shape in image i being generated by component m.

4.3.4 Expectation Maximization in the loop

As previously discussed, our initial shape prior is obtained from artist data which we find is unrepresentative of the diverse shapes present in our real dog dataset. We address this by proposing to recover the latent variables w_i^m and parameters (μ_β^m, Σ_β^m and Π_β^m) of our 3D shape prior by learning from monocular images of in-the-wild dogs and their 2D training labels in our training dataset.

We achieve this using Expectation Maximization (EM), which regularly updates the means and variances for each mixture component and per-image mixture weights based on the observed shapes in the training set. While training our 3D reconstruction network, we progressively update our shape mixture model with an alternating 'E' step and 'M' step described below:

The 'E' Step.

The 'E' step computes the expected value of the latent variables w_i^m assuming fixed $(\mu_\beta^m, \Sigma_\beta^m, \Pi_\beta^m)$ for all $i \in \{1,\ldots,N\}, m \in \{1,\ldots,M\}$.

The update equation for an image i with latest shape prediction β_i and cluster m with parameters $(\mu_\beta^m, \Sigma_\beta^m, \Pi_\beta^m)$ is given as:

$$w_i^m := \frac{\mathcal{N}(\beta_i | \mu_\beta^m, \Sigma_\beta^m)\Pi_\beta^m}{\sum_{m'}^{M} \mathcal{N}(\beta_i | \mu_\beta^{m'}, \Sigma_\beta^{m'})\Pi_\beta^{m'}} \tag{4.6}$$

The 'M' Step.

The 'M' step computes new values for $(\mu_\beta^m, \Sigma_\beta^m, \Pi_\beta^m)$, assuming fixed w_i^m for all $i \in \{1,\ldots,N\}, m \in \{1,\ldots,M\}$.

The update equations are given as follows:

$$\mu_\beta^m := \frac{\sum_i w_i^m \beta_i}{\sum_i w_i^m} \quad \Sigma_\beta^m := \frac{\sum_i w_i^m (\beta_i - \Sigma_\beta^m)(\beta_i - \Sigma_\beta^m)^T}{\sum_i w_i^m} \quad \Pi_\beta^m := \frac{1}{N}\sum_i w_i^m \tag{4.7}$$

4.4 Building StanfordExtra: a new large-scale dog keypoint dataset

In order to evaluate our method, we introduce *StanfordExtra*: a new large-scale dataset with annotated 2D keypoints and binary segmentation masks for dogs. We opted to take source

Fig. 4.4 **StanfordExtra example images**. *Left*: outlined segmentations and labelled keypoints for 24 representative images. *Right*: heatmap of deviation of worker submitted results from mean for each submission.

images from the existing Stanford Dog Dataset [62], which consists of 20,580 dog images taken “in the wild" and covers 120 dog breeds. The dataset contains vast shape and pose variation between dogs, as well as nuisance factors such as self/environmental occlusion, interaction with humans/other animals and partial views. Figure 4.4 (left) shows samples from the new dataset.

We used Amazon Mechanical Turk to collect a binary silhouette mask and 20 keypoints per image: 3 per leg (knee, ankle, toe), 2 per ear (base, tip), 2 per tail (base, tip), 2 per face (nose and jaw). We can approximate the difficulty of the dataset by analysing the variance between 3 annotators at both the joint labelling and silhouette task. Figure 4.4 (right) illustrates typical per-joint variance in joint labelling. Further details of the data curation procedure are left to the supplementary materials.

4.5 Experiments

In this section we compare our method to competitive baselines. We begin by describing our new large-scale dataset of annotated dog images, followed by a quantitative and qualitative evaluation.

4.5.1 Evaluation protocol

Our evaluation is based on our new StanfordExtra dataset. In line with other methods which tackle “in-the-wild” 3D reconstruction of articulated subjects [63, 64], we filter images from the original dataset of 20,580 for which the majority of dog keypoints are invisible. We consider these images unsuitable for our full-body dog reconstruction task. We also remove images for which the consistency in keypoint/silhouette segmentations between the

3 annotators is below a set threshold. This leaves us with 8,476 images which we divide per-breed into an 80%/20% train and test split.

We consider two primary evaluation metrics. IoU is the intersection-over-union of the projected model silhouette compared to the ground truth annotation and indicates the quality of the reconstructed 3D shape. Percentage of Correct Keypoints (PCK) computes the percentage of joints which are within a normalized distance (based on square root of 2D silhouette area) to the ground truth locations, and evaluates the quality of reconstructed 3D pose. We also produce PCK results on various joint groups (legs, tail, ears, face) to compare the reconstruction accuracy for different parts of the dog model.

4.5.2 Training procedure

We train our model in two stages. The first omits the silhouette loss which we find can lead the network to unsatisfactory local minima if applied too early. With the silhouette loss turned off, we find it satisfactory to use the simple unimodal prior (and without EM) for this preliminary stage since there is no loss to specifically encourage a strong shape alignment. After this, we introduce the silhouette loss, the mixture prior and begin applying the expectation maximization updates over $M = 10$ clusters. We train the first stage for 250 epochs, the second stage for 150 and apply the EM step every 50 epochs. All losses are weighted, as described in the supplementary. The entire training procedure takes 96 hours on a single P100 GPU.

4.5.3 Comparison to baselines

We first compare our method to various baseline methods. SMAL [158] is an approach which fits the 3D SMAL model using per-image energy minimization. Creatures Great and SMAL (CGAS) [10] is a three-stage method, which employs a joint predictor on silhouette renderings from synthetic 3D dogs, applies a genetic algorithm to clean predictions, and finally applies the SMAL optimizer to produce the 3D mesh.

At test-time both SMAL and CGAS rely on manually-provided segementation masks, and SMAL also relies on hand-clicked keypoints. In order to produce a fair comparison, we produce a set of *predicted* keypoints for StanfordExtra by training the Stacked Hourglass Network [85] with 8 stacks and 1 block, and *predicted* segmentation masks using DeepLab v3+ [23]. The Stacked Hourglass Network achieves 71.4% PCK score, DeepLab v3+ achieves 83.4% IoU score and the CGAS joint predictor achieves 41.8% PCK score.

Method	Kps	Seg	IoU	PCK				
				Avg	Legs	Tail	Ears	Face
SMAL [158]	Pred	Pred	67.9	67.1	65.7	79.5	54.9	87.4
SMAL	GT	GT	69.2	72.6	69.9	**92.0**	58.6	**96.9**
SMAL	GT	Pred	68.6	72.6	70.2	91.5	58.1	**96.9**
SMAL	Pred	GT	68.5	67.4	66.0	79.9	55.0	88.2
CGAS [10]	CGAS	Pred	62.4	43.7	46.5	64.1	36.5	21.4
CGAS	CGAS	GT	63.1	43.6	46.3	64.2	36.3	21.6
SMAL + scaling	Pred	Pred	69.3	69.6	69.4	79.3	56.5	87.6
SMAL + scaling + new prior	Pred	Pred	70.7	71.6	71.5	80.7	59.3	88.0
Ours	—	—	**73.6**	**75.7**	**75.0**	77.6	**69.9**	90.0

Table 4.1 **Baseline comparisons.** Both PCK and silhouette IOU scores are shown for SOTA methods under varying conditions. A combination of both ground truth (GT) and predicted (Pred) keypoints/segmentations using hourglass network and deeplab respectively. For the CGAS method we also test using their keypoint predictor (CGAS). The addition of scaling and new prior are shown to improve the original SMAL method.

Table 4.1 and Figure 4.5 show the comparison against competitive methods. For full examination, we additionally provide results for SMAL and CGAS in the scenario that ground-truth keypoints and/or segmentations are available at test time.

The results show our end-to-end method outperforms the competitors when they are provided with predicted keypoints/segmentations (white rows). Our method therefore achieves a new state-of-the-art on this 3D reconstruction task. In addition, we show our method achieves improved average IoU/PCK scores than competitive methods, even when they are provided ground truth annotations at test time (grey rows). We also demonstrate wider applicability of two contributions from our work (scale parameters and improved prior) by showing improved performance of the SMAL method when these are incorporated. Finally, our model's test-time speed is significantly faster than the competitors as it does not require an optimizer.

4.5.4 Generalization to unseen dataset

Table 4.2 shows an experiment to compare how well our model generalizes to a new data domain. We test our model against the SMAL [158] method (using predicted keypoints and segmentations as above for fairness) on the recent Animal Pose dataset [17]. The data preparation process is the same as for StanfordExtra and no fine-tuning was used for either method. We achieve good results in this unseen domain and still improve over the SMAL optimizer.

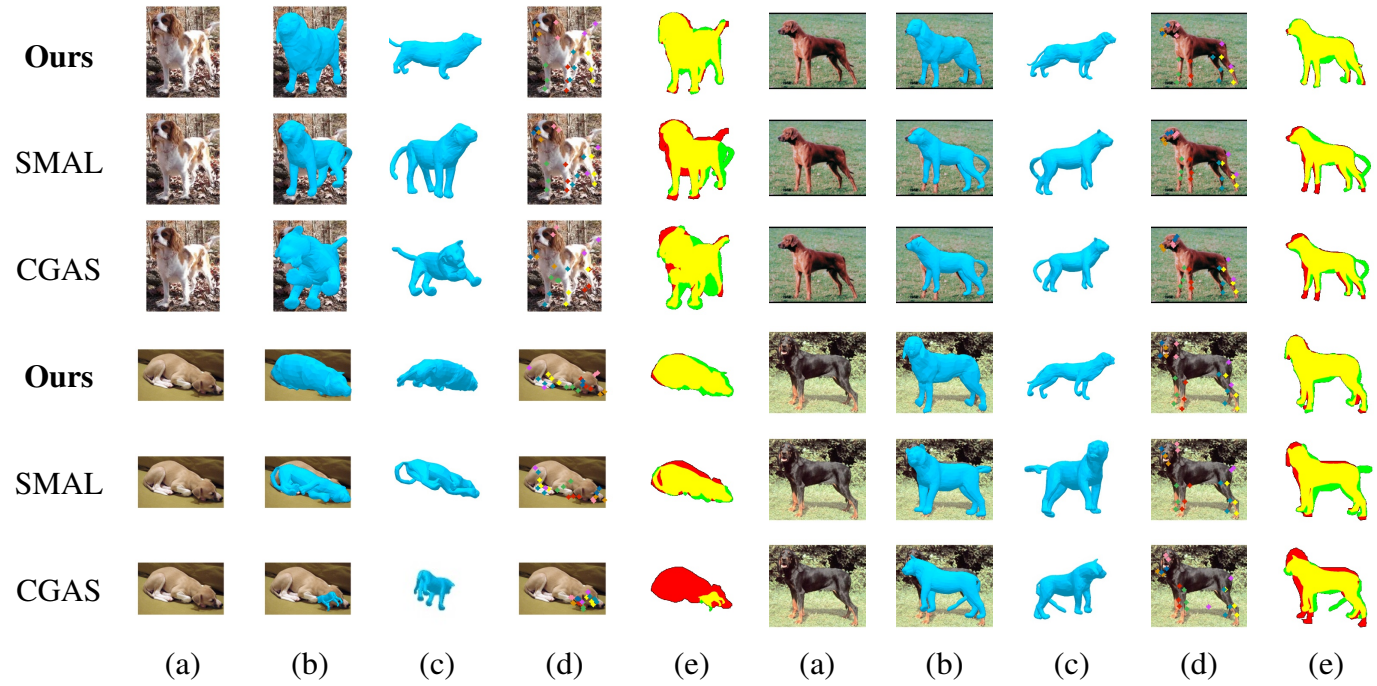


Fig. 4.5 **Qualitiative comparison to SOTA.** Row 1: **Ours**, Row 2: SMAL [158], Row 3: CGAS [10]. (a) input image, (b) predicted 3D mesh, (c) canonical view 3D mesh, (d) reprojected model joints and (e) silhouette reprojection error.

4.5.5 Ablation study

We also produce a study in which we ablate individual components of our method and examine the effect on the PCK/IoU performance. We evaluate three variants: (1) **Ours w/o EM** that omits EM updates, (2) **Ours w/o MoG** which replaces our mixture shape prior with a unimodal prior, (3) **Ours w/o Scale** which removes the scale parameters.

The results in Table 4.3 indicate that each individual component has a positive impact on the overall method performance. In particular, it can be seen that the inclusion of the EM and Mixture of Gaussians prior leads to an improvement in IoU, suggesting that the shape

Method	IoU	PCK				
		Avg	Legs	Tail	Ears	Face
SMAL [158]	63.6	69.1	60.9	83.5	75.0	93.0
Ours	**66.9**	**73.8**	**65.1**	**85.6**	**84.0**	**93.6**

Table 4.2 **Animal Pose dataset [17].** Evaluation on recent Animal Pose dataset with no fine-tuning to our method nor joint/silhouette predictors used for SMAL.

Method	IoU	PCK				
		Avg	Legs	Tail	Ears	Face
Ours	**73.6**	**75.7**	**75.0**	**77.6**	69.9	90.0
−EM	67.7	74.6	72.9	75.2	**72.5**	88.3
−MoG	68.0	74.9	74.3	73.3	70.0	**90.2**
−Scale	67.3	72.6	72.9	75.3	62.3	89.1

Table 4.3 **Ablation study.** Evaluation with the following contributions removed: (a) EM updates, (b) Mixture Shape Prior, (c) SMBLD scale parameters.

prior refinements steps help the model accurately fit the exact dog shape. Interestingly, we notice that adding the Mixture of Gaussians prior but omitting EM steps slightly hinders performance, perhaps due to an sub-optimal initialization for the M clusters. However, we find adding EM updates to the Mixture of Gaussian model improves all metrics except the ear keypoint accuracy. We observe the error here is caused by the our shape prior learning slightly imprecise shapes for dogs with extremely "floppy" ears. Although there is good silhouette coverage for these regions, the fact our model has only a single articulation point per ear causes a lack of flexibility that results in occasionally misplaced ear tips for these instances. This could be improved in future work by adding additional model joints to the ear. Finally, we find the increased model flexibility afforded by the SMBLD scale parameters have a positive effect on IoU/PCK scores.

4.5.6 Qualitative evaluation

Figure 4.5 shows a range of example system outputs when tested on range of StanfordExtra and Animal Pose [17] dogs with varying pose and shape and in challenging conditions. Note that only StanfordExtra is used for training.

4.6 Conclusions

This paper presents an end-to-end method for automatic, monocular 3D dog reconstruction. We achieve this using only weak 2D supervision, provided by our novel StanfordExtra dataset. Further, we show we can learn a more detailed shape prior by tuning a gaussian mixture during model training and this leads to improved reconstructions. We also show our method improves over competitive baselines, even when they are given access to ground truth data at test time.

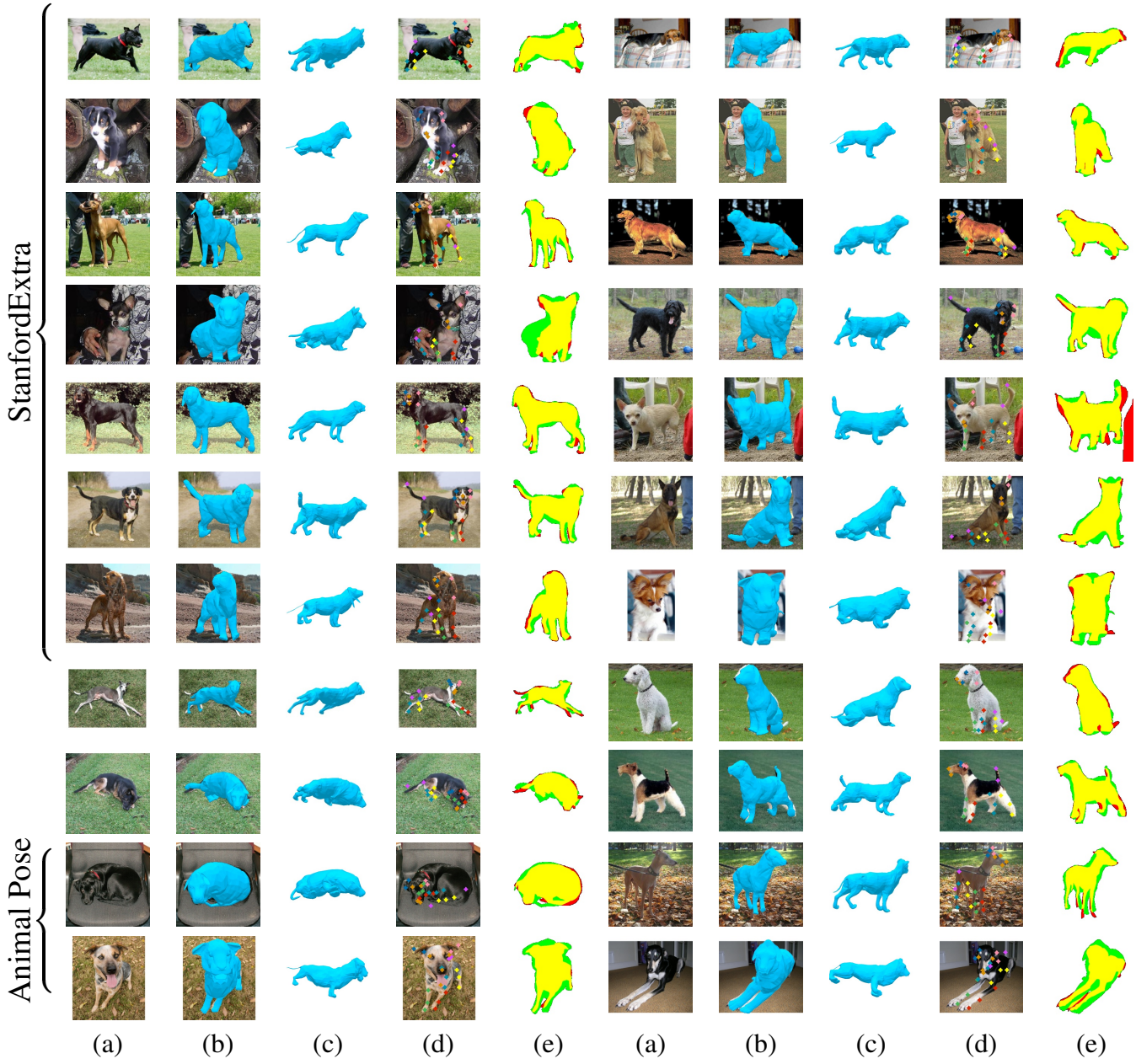


Fig. 4.6 **Qualitative results on StanfordExtra and Animal Pose [17].** For each sample we show: (a) input image, (b) predicted 3D mesh, (c) canonical view 3D mesh, (d) reprojected model joints and (e) silhouette reprojection error.

Future work should involve tackling some failure cases of our system, for example handling multiple overlapping dogs or dealing with heavy motion blur. Other areas for research include extending our EM formulation to handle video input to take advantage of multi-view shape constraints, and transferring knowledge accumulated through training on StanfordExtra dogs to other species.

Chapter 5

Handling Ambiguous Input with Multi-Output Learning

5.1 Introduction

We consider the problem of obtaining dense 3D reconstructions of humans from single and partially occluded views. In such cases, the visual evidence is usually insufficient to identify a 3D reconstruction uniquely, so we aim at recovering several plausible reconstructions compatible with the input data. We suggest that ambiguities can be modelled more effectively by parametrizing the possible body shapes and poses via a suitable 3D model, such as SMPL for humans. We propose to learn a multi-hypothesis neural network regressor using a best-of-M loss, where each of the M hypotheses is constrained to lie on a manifold of plausible human poses by means of a generative model. We show that our method outperforms alternative approaches in ambiguous pose recovery on standard benchmarks for 3D humans, and in heavily occluded versions of these benchmarks.

We are interested in reconstructing 3D human pose from the observation of single 2D images. As humans, we have no problem in predicting, at least approximately, the 3D structure of most scenes, including the pose and shape of other people, even from a single view. However, 2D images notoriously [31] do not contain sufficient geometric information to allow recovery of the third dimension. Hence, single-view reconstruction is only possible in a probabilistic sense and the goal is to make the posterior distribution as sharp as possible, by learning a strong prior on the space of possible solutions.

Recent progress in single-view 3D pose reconstruction has been impressive. Methods such as HMR [56], GraphCMR [64] and SPIN [63] formulate this task as learning a deep neural network that maps 2D images to the parameters of a 3D model of the human body,

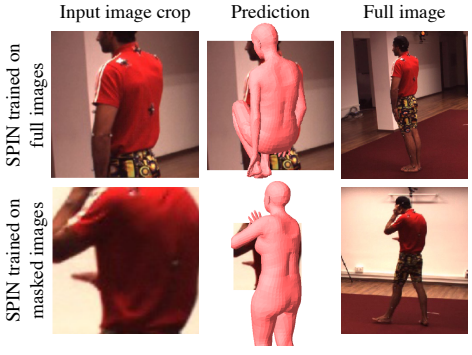
Fig. 5.1 **Human mesh recovery in an ambiguous setting.** We propose a novel method that, given an occluded input image of a person, outputs the set of meshes which constitute plausible human bodies that are consistent with the partial view. The ambiguous poses are predicted using a novel *n*-quantized-best-of-*M* method.

usually SMPL [73]. These methods work well in general, but not always (fig. 5.2). Their main weakness is processing *heavily occluded images* of the object. When a large part of the object is missing, say the lower body of a sitting human, they output reconstructions that are often implausible. Since they can produce only one hypothesis as output, they very likely learn to approximate the mean of the posterior distribution, which may not correspond to any plausible pose. Unfortunately, this failure modality is rather common in applications due to scene clutter and crowds.

In this paper, we propose a solution to this issue. Specifically, we consider the challenge of recovering 3D mesh reconstructions of complex articulated objects such as humans from highly ambiguous image data, often containing significant occlusions of the object. Clearly, it is generally impossible to reconstruct the object uniquely if too much evidence is missing; however, we can still predict a *set* containing all possible reconstructions (see fig. 5.1), making this set as small as possible. While ambiguous pose reconstruction has been previously investigated, as far as we know, this is the first paper that looks specifically at a deep learning approach for ambiguous reconstructions of the *full human mesh*.

Our primary contribution is to introduce a principled multi-hypothesis framework to model the ambiguities in monocular pose recovery. In the literature, such multiple-hypotheses networks are often trained with a so-called *best-of-M* loss — namely, during training, the loss is incurred only by the best of the M hypothesis, back-propagating gradients from that alone [40]. In this work we opt for the *best-of-M* approach since it has been show to outperform alternatives (such as variational auto-encoders or mixture density networks) in tasks that are similar to our 3D human pose recovery, and which have constrained output spaces [104].

A major drawback of the *best-of-M* approach is that it only guarantees that *one* of the hypotheses lies close to the correct solution; however, it says nothing



about the plausibility, or lack thereof, of the *other* $M-1$ hypotheses, which can be arbitrarily 'bad'.[1] Not only does this mean that most of the hypotheses may be uninformative, but in an application we are also unable to tell *which* hypothesis should be used, and we might very well pick a 'bad' one. This has also a detrimental effect during learning because it makes gradients sparse as prediction errors are back-propagated only through one of the M hypotheses for each training image.

In order to address these issues, our first contribution is a *hypothesis reprojection loss* that forces each member of the multi-hypothesis set to correctly reproject to 2D image keypoint annotations. The main benefit is to constrain the *whole* predicted set of meshes to be consistent with the observed image, not just the best hypothesis, also addressing gradient sparsity.

Next, we observe that another drawback of the best-of-M pipelines is to be tied to a particular value of M, whereas in applications we are often interested in tuning the number of hypothesis considered. Furthermore, minimizing the reprojection loss makes hypotheses geometrically consistent with the observation, but not necessarily likely. Our second contribution is thus to improve the flexibility of best-of-M models by allowing them to output any smaller number $n < M$ of hypotheses while at the same time making these hypotheses *more representative of likely* poses. The new method, which we call n-quantized-best-of-M, does so by quantizing the best-of-M model to output weighed by a *explicit pose prior*, learned by means of normalizing flows.

To summarise, our key contributions are as follows. First, we deal with the challenge of 3D mesh reconstruction for articulated objects such as humans in *ambiguous* scenarios. Second, we introduce a *n-quantized-best-of-M* mechanism to allow best-of-M models to generate an arbitrary number of $n < M$ predictions. Third, we introduce a mode-wise reprojection loss for multi-hypothesis prediction, to ensure that predicted hypotheses are *all* consistent with the input.

[1] Theoretically, best-of-M can minimize its loss by quantizing optimally (in the sense of minimum expected distortion) the posterior distribution, which would be desirable for coverage. However, this is *not* the only solution that optimizes the best-of-M training loss, as in the end it is sufficient that *one* hypothesis per training sample is close to the ground truth. In fact, this is exactly what happens; for instance, during training hypotheses in best-of-M are known to easily become degenerate and 'die off', a clear symptom of this problem.

Empirically, we achieve state-of-the-art monocular mesh recovery accuracy on Human36M, its more challenging version augmented with heavy occlusions, and the 3DPW datasets. Our ablation study validates each of our modelling choices, demonstrating their positive effect.

5.2 Related work

There is ample literature on recovering the pose of 3D models from images. We break this into five categories: methods that reconstruct 3D points directly, methods that reconstruct the parameters of a 3D model of the object via optimization, methods that do the latter via learning-based regression, hybrid methods and methods which deal with uncertainty in 3D human reconstruction.

5.2.1 Reconstructing 3D body points without a model.

Several papers have focused on the problem of estimating 3D body points from 2D observations [6, 81, 103, 123, 64]. Of these, Martinez et al. [78] introduced a particularly simple pipeline based on a shallow neural network. In this work, we aim at recovering the full 3D surface of a human body, rather than only lifting sparse keypoints.

5.2.2 Fitting 3D models via direct optimization.

Several methods *fit* the parameters of a 3D model such as SMPL [73] or SCAPE [6] to 2D observations using an optimization algorithm to iteratively improve the fitting quality. While early approaches such as [38, 113] required some manual intervention, the SMPLify method of Bogo et al. [14] was perhaps the first to fit SMPL to 2D keypoints fully automatically. SMPL was then extended to use silhouette, multiple views, and multiple people in [66, 49, 154]. Recent optimization methods such as [55, 90, 146] have significantly increased the scale of the models and data that can be handled.

5.2.3 Fitting 3D models via learning-based regression.

More recently, methods have focused on regressing the parameters of the 3D models directly, *in a feed-forward manner*, generally by learning a deep neural network [126, 136, 87, 91, 56]. Due to the scarcity of 3D ground truth data for humans in the wild, most of these methods train a deep regressor using a mix of datasets with 3D and 2D annotations in form of 3D

MoCap markers, 2D keypoints and silhouettes. Among those, HMR of Kanazawa et al. [56] and GraphCMR of Kolotouros et al. [64] stand out as particularly effective.

5.2.4 Hybrid methods.

Other authors have also combined optimization and learning-based regression methods. In most cases, the integration is done by using a deep regressor to initialize the optimization algorithm [113, 66, 103, 91, 137]. However, recently Kolotouros et al. [63] has shown strong results by integrating the optimization loop in learning the deep neural network that performs the regression, thereby exploiting the weak cues available in 2D keypoints.

5.2.5 Modelling ambiguities in 3D human reconstruction.

Several previous papers have looked at the problem of modelling ambiguous 3D human pose reconstructions. Early work includes Sminchisescu and Triggs [115], Sidenbladh et al. [112] and Sminchisescu et al. [114].

More recently, Akhter and Black [3] learn a prior over human skeleton joint angles (but not directly a prior on the SMPL parameters) from a MoCap dataset. Li and Lee [67] use the Mixture Density Networks model of [11] to capture ambiguous 3D reconstructions of sparse human body keypoints directly in physical space. Sharma et al. [108] learn a conditional variational auto-encoder to model ambiguous reconstructions as a posterior distribution; they also propose two scoring methods to extract a single 3D reconstruction from the distribution.

Cheng et al. [24] tackle the problem of video 3D reconstruction in the presence of occlusions, and show that temporal cues can be used to disambiguate the solution. While our method is similar in the goal of correctly handling the prediction uncertainty, we differ by applying our method to predicting *full mesh* of the human body. This is arguably a more challenging scenario due to the increased complexity of the desired 3D shape.

Finally, some recent concurrent works also consider building priors over 3D human pose using normalizing flows. Xu et al. [148] release a prior for their new GHUM/GHUML model, and Zanfir et al. [153] build a prior on SMPL joint angles to constrain their weakly-supervised network. Our method differs as we learn our prior on 3D SMPL joints.

5.3 Preliminaries

Before discussing our method, we describe the necessary background, starting from SMPL.

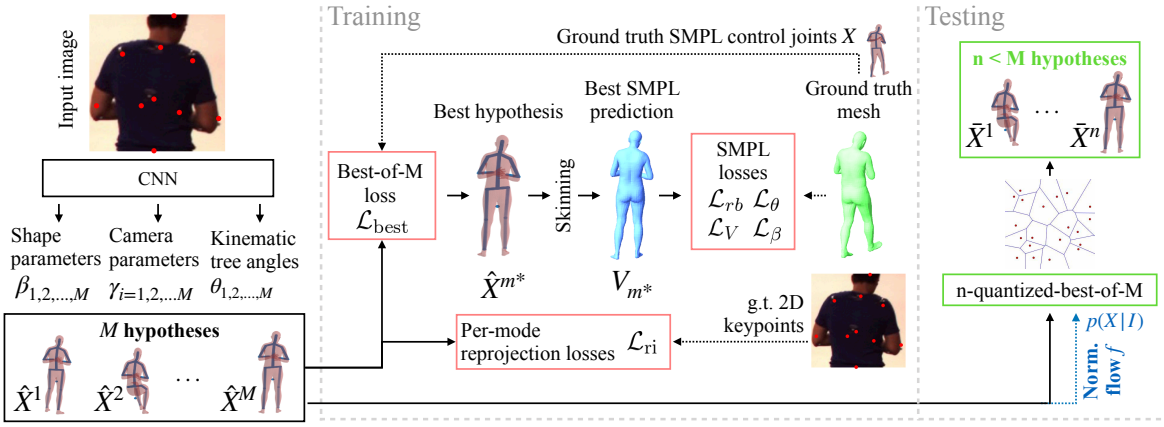


Fig. 5.3 **Overview of our method.** Given a single image of a human, during training, our method produces multiple skeleton hypotheses $\{\hat{X}^i\}_{i=1}^M$ that enter a Best-of-M loss which selects the representative $\hat{X}^{m^*}$ which most accurately matches the ground truth control joints X. At test time, we sample an arbitrary number of $n < M$ hypotheses by quantizing the set $\{\hat{X}^i\}$ that is assumed to be sampled from the probability distribution $p(X|I)$ modeled with normalizing flow f.

5.3.1 SMPL.

SMPL is a model of the human body parameterized by axis-angle rotations $\theta \in \mathbb{R}^{69}$ of 23 body joints, the shape coefficients $\beta \in \mathbb{R}^{10}$ modelling shape variations, and a global rotation $\gamma \in \mathbb{R}^3$. SMPL defines a *skinning function* $S : (\theta, \beta, \gamma) \mapsto V$ that maps the body parameters to the vertices $V \in \mathbb{R}^{6890\times 3}$ of a 3D mesh.

5.3.2 Predicting the SMPL parameters from a single image.

Given an image **I** containing a person, the goal is to recover the SMPL parameters (θ, β, γ) that provide the best 3D reconstruction of it. Existing algorithms [**?**] cast this as learning a deep network $G(I) = (\theta, \beta, \gamma, t)$ that predicts the SMPL parameters as well as the translation $t \in \mathbb{R}^3$ of the perspective camera observing the person. We assume a fixed set of camera parameters. During training, the camera is used to constrain the reconstructed 3D mesh and the annotated 2D keypoints to be consistent. Since most datasets only contain annotations for a small set of keypoints ([39] is an exception), and since these keypoints do not correspond directly to any of the SMPL mesh vertices, we need a mechanism to translate between them. This mechanism is a fixed linear regressor $J : V \mapsto X$ that maps the SMPL mesh vertices $V = S(G(I))$ to the 3D locations $X = J(V) = J(S(G(I)))$ of the K joints. Then, the

projections $\pi_t(X)$ of the 3D joint positions into image $\mathbf{I}$ can be compared to the available 2D annotations.

5.3.3 Normalizing flows.

The idea of normalizing flows (NF) is to represent a complex distribution $p(X)$ on a random variable X as a much simpler distribution $p(z)$ on a transformed version $z = f(X)$ of X. The transformation f is learned so that $p(z)$ has a fixed shape, usually a Normal $p(z) \sim \mathcal{N}(0,1)$. Furthermore, f itself must be *invertible* and *smooth*. In this paper, we utilize a particular version of NF dubbed RealNVP [29]. A more detailed explanation of NF and RealNVP has been deferred to the supplementary.

5.3.4 Method

We start from a neural network architecture that implements the function $G(I) = (\theta, \beta, \gamma, t)$ described above. As shown in SPIN [63], the HMR [**?**] architecture attains state-of-the-art results for this task, so we use it here. However, the resulting regressor $G(I)$, given an input image I, can only produce a single unique solution. In general, and in particular for cases with a high degree of reconstruction ambiguity, we are interested in predicting *set* of plausible 3D poses rather than a single one. We thus extend our model to explicitly produce a set of M different hypotheses $G_m(I) = (\theta_m, \beta_m, \gamma_m, t_m)$, $m = 1, \ldots, M$. This is easily achieved by modifying the HMR's final output layer to produce a tensor M times larger, effectively stacking the hypotheses. In what follows, we describe the learning scheme that drives the monocular predictor G to achieve an optimal coverage of the plausible poses consistent with the input image. Our method is summarized in fig. 5.3.

5.3.5 Learning with multiple hypotheses

For learning the model, we assume to have a training set of N images $\{I_i\}_{i=1,\ldots,N}$, each cropped around a person. Furthermore, for each training image I_i we assume to know (1) the 2D location Y_i of the body joints (2) their 3D location X_i, and (3) the ground truth SMPL fit $(\theta_i, \beta_i, \gamma_i)$. Depending on the set up, some of these quantities can be inferred from the others (e.g. we can use the function J to convert the SMPL parameters to the 3D joints X_i and then the camera projection to obtain Y_i).

5.3.6 Best-of-M loss.

Given a single input image, our network predicts a set of poses, where at least one should be similar to the ground truth annotation X_i. This is captured by the best-of-M loss [40]:

$$\mathscr{L}_{\text{best}}(J,G;m^*) = \frac{1}{N}\sum_{i=1}^{N} \left\|X_i - \hat{X}^{m_i^*}(I_i)\right\|, \;\; m_i^* = \underset{m=1,\ldots,M}{\operatorname{argmin}} \left\|X_i - \hat{X}^{m}(I_i)\right\|, \tag{5.1}$$

where $\hat{X}^m(I_i) = J(G_m(V(I_i)))$ are the 3D joints estimated by the m-th SMPL predictor $G_m(I_i)$ applied to image I_i. In this way, only the best hypothesis is steered to match the ground truth, leaving the other hypotheses free to sample the space of ambiguous solutions. During the computation of this loss, we also extract the best index m_i^* for each training example.

5.3.7 Limitations of best-of-M.

As noted in section 5.1, best-of-M only guarantees that one of the M hypotheses is a good solution, but says nothing about the other ones. Furthermore, in applications we are often interested in modulating the number of hypotheses generated, but the best-of-M regressor $G(I)$ only produces a fixed number of output hypothesis M, and changing M would require retraining from scratch, which is intractable.

We first address these issues by introducing a method that allows us to train a best-of-M model for a large M once and leverage it later to generate an arbitrary number of $n < M$ hypotheses without the need of retraining, while ensuring that these are good representatives of likely body poses.

5.3.8 n-quantized-best-of-M

Formally, given a set of M predictions $\hat{\mathscr{X}}^M(I) = \{\hat{X}^1(I),...,\hat{X}^M(I)\}$ we seek to generate a smaller n-sized set $\bar{\mathscr{X}}^n(I) = \{\bar{X}^1(I),...,\bar{X}^n(I)\}$ which preserves the information contained in $\hat{\mathscr{X}}^M$. In other words, $\bar{\mathscr{X}}^n$ *optimally quantizes* $\hat{\mathscr{X}}^M$. To this end, we interpret the output of the best-of-M model as a set of choices $\hat{\mathscr{X}}^M(I)$ for the possible pose. These poses are of course not all equally likely, but it is difficult to infer their probability from (5.1). We thus work with the following approximation. We consider the prior $p(X)$ on possible poses (defined in the next section), and set:

$$p(X|I) = p(X|\hat{\mathscr{X}}^M(I)) = \sum_{i=1}^{M} \delta(X - \hat{X}^i(I)) \frac{p(\hat{X}^i(I))}{\sum_{k=1}^{M} p(\hat{X}^k(I))}. \tag{5.2}$$

This amounts to using the best-of-M output as a conditioning *set* (i.e. an unweighted selection of plausible poses) and then use the prior $p(x)$ to weight the samples in this set. With the weighted samples, we can then run K-means [72] to further quantize the best-of-M output while minimizing the quantization energy E:

$$E(\bar{\mathscr{X}}|\hat{\mathscr{X}}) = \mathbb{E}_{p(X|I)}\left[\min_{\{\bar{X}^1,\ldots,\bar{X}^n\}} \|X-\bar{X}^j\|^2\right] = \sum_{i=1}^{M} \frac{p(\hat{X}^i(I))}{\sum_{k=1}^{M} p(\hat{X}^k(I))} \min_{\{\bar{X}^1,\ldots,\bar{X}^n\}} \|\hat{X}^i(I)-\bar{X}^j\|^2. \tag{5.3}$$

This can be done efficiently on GPU — for our problem, K-Means consumes less than 20% of the execution time of the entire forward pass of our method.

5.3.9 Learning the pose prior with normalizing flows.

In order to obtain $p(X)$, we propose to learn a normalizing flow model in form of the RealNVP network f described in section 5.3 and the supplementary. RealNVP optimizes the log likelihood $\mathscr{L}_{\text{nf}}(f)$ of training ground truth 3D skeletons $\{X_1,\ldots X_N\}$ annotated in their corresponding images $\{I_1,\ldots,I_N\}$:

$$\mathscr{L}_{\text{nf}}(f) = -\frac{1}{N}\sum_{i=1}^{N} \log p(X_i) = -\frac{1}{N}\sum_{i=1}^{N}\left(\log \mathscr{N}(f(X_i)) - \sum_{l=1}^{L} \log\left|\frac{df_l(X_{li})}{dX_{li}}\right|\right). \tag{5.4}$$

5.3.10 2D re-projection loss.

Since the best-of-M loss optimizes a single prediction at a time, often some members of the ensemble $\hat{\mathscr{X}}(I)$ drift away from the manifold of plausible human body shapes, ultimately becoming 'dead' predictions that are never selected as the best hypothesis m^*. In order to prevent this, we further utilize a re-projection loss that acts across all hypotheses for a given image. More specifically, we constrain the set of 3D reconstructions to lie on projection rays passing through the 2D input keypoints with the following *hypothesis re-projection loss*:

$$\mathscr{L}_{\text{ri}}(J,G) = \frac{1}{N}\sum_{i=1}^{N}\sum_{m=1}^{M} \left\|Y_i - \pi_{t_i}(\hat{X}^m(I))\right\|. \tag{5.5}$$

Note that many of our training images exhibit significant occlusion, so Y may contain invisible or missing points. We handle this by masking $\mathscr{L}_{\text{ri}}$ to prevent these points contributing to the loss.

SMPL loss. The final loss terms, introduced by prior work [**?** 91, 63], penalize deviations between the predicted and ground truth SMPL parameters. For our method, these are only

Fig. 5.4 **Example samples from the normalizing flow** $f : X \mapsto z;\ p(z) \sim \mathcal{N}(0,1)$, trained on a dataset of ground truth 3D SMPL control skeletons $\{X_1, ..., X_N\}$.

applied to the best hypothesis m_i^* found above:

$$\mathscr{L}_\theta(G;m^*) = \frac{1}{N}\sum_{i=1}^{N} \|\theta_i - G_{\theta,m_i^*}(I_i)\|;\ \mathscr{L}_V(G;m^*) = \frac{1}{N}\sum_{i=1}^{N} \|S(\theta_i,\beta_i,\gamma_i) - S(G_{(\theta,\beta,\gamma),m_i^*}(I_i))\| \quad (5.6)$$

$$\mathscr{L}_\beta(G;m^*) = \frac{1}{N}\sum_{i=1}^{N} \|\beta_i - G_{\beta,m_i^*}(I_i)\|;\ \mathscr{L}_{\text{rb}}(G;m^*) = \frac{1}{N}\sum_{i=1}^{N} \|Y_i - \pi_{t_i}(\hat{X}^{m_i^*}(I_i))\| \quad (5.7)$$

Note here we use $\mathscr{L}_{\text{rb}}$ to refer to a 2D re-projection error between the best hypothesis and ground truth 2D points Y_i. This differs from the earlier loss $\mathscr{L}_{\text{ri}}$, which is applied across all modes to enforce consistency to the visible *input* points. Note that we could have used eqs. (5.6) and (5.7) to select the best hypothesis m_i^*, but it would entail an unmanageable memory footprint due to the requirement of SMPL-meshing for every hypothesis before the best-of-M selection.

5.3.11 Overall loss.

The model is thus trained to minimize:

$$\begin{aligned}\mathscr{L}(J,G) = {} & \lambda_{\text{ri}}\mathscr{L}_{\text{ri}}(J,G) + \lambda_{\text{best}}\mathscr{L}_{\text{best}}(J,G;m^*) + \lambda_\theta\mathscr{L}_\theta(J,G;m^*) \\ & + \lambda_\beta\mathscr{L}_\beta(J,G;m^*) + \lambda_{\text{V}}\mathscr{L}_V(J,G;m^*) + \lambda_{\text{rb}}\mathscr{L}_{\text{rb}}(J,G;m^*)\end{aligned} \quad (5.8)$$

where m^* is given in eq. (5.1) and $\lambda_{\text{ri}}, \lambda_{\text{best}}, \lambda_\theta, \lambda_\beta, \lambda_{\text{V}}, \lambda_{\text{rb}}$ are weighing factors. We use a consistent set of SMPL loss weights across all experiments $\lambda_{\text{best}} = 25.0, \lambda_\theta = 1.0, \lambda_\beta = 0.001, \lambda_{\text{V}} = 1.0$, and set $\lambda_{\text{ri}} = 1.0$. Since the training of the normalizing flow f is independent of the rest of the model, we train f separately by optimizing $\mathscr{L}_{\text{nf}}$ with the weight of $\lambda_{\text{nf}} = 1.0$. Samples from our trained normalizing flow are shown in fig. 5.4

5.4 Experiments

In this section we compare our method to several strong baselines. We start by describing the datasets and the baselines, followed by a quantitative and a qualitative evaluation.

Dataset	Quantization n	1		5		10		25	
	Metric	MPJPE	RE	MPJPE	RE	MPJPE	RE	MPJPE	RE
H36M	HMR [56]	—	56.8	—	—	—	—	—	—
	GraphCMR [64]	71.9	50.1	—	—	—	—	—	—
	SPIN [63]	62.2	41.8	—	—	—	—	—	—
	SMPL-MDN	64.4	44.8	61.8	43.3	61.3	43.0	61.1	42.7
	SMPL-CVAE	70.1	46.7	68.9	46.4	68.6	46.3	68.1	46.2
	Ours	**61.5**	**41.6**	**59.8**	**42.0**	**59.2**	**42.2**	**58.2**	**42.2**
3DPW	HMR [56]	—	81.3	—	—	—	—	—	—
	GraphCMR [64]	—	70.2	—	—	—	—	—	—
	SPIN [63]	96.9	**59.3**	—	—	—	—	—	—
	SMPL-MDN	105.8	64.7	96.9	61.2	95.9	60.7	94.9	60.1
	SMPL-CVAE	96.3	61.4	93.7	60.7	92.9	60.5	92.0	60.3
	Ours	**93.8**	59.9	**82.2**	**57.1**	**79.4**	**56.6**	**75.8**	**55.6**
AH36M	SMPL-MDN	113.9	74.7	98.0	70.8	95.1	69.9	91.5	69.5
	SMPL-CVAE	114.5	76.5	111.5	75.7	110.6	75.4	109.7	75.1
	Ours	**103.6**	**67.8**	**96.4**	**67.1**	**93.5**	**66.0**	**90.0**	**64.2**
A3DPW	SMPL-MDN	159.7	82.8	154.6	83.0	149.6	80.7	122.1	76.6
	SMPL-CVAE	156.6	80.2	154.5	79.9	153.9	79.8	153.1	79.8
	Ours	**149.6**	**78.5**	**125.6**	**74.4**	**116.7**	**73.7**	**107.8**	**72.1**

Table 5.1 **Monocular multi-hypothesis human mesh recovery** comparing our approach to two multi-hypothesis baselines (SMPL-CVAE, SMPL-MDN) and state-of-the-art single mode evaluation models [63, 64, 56] on Human3.6m (H36M), its ambiguous version AH36M, on 3DPW and its ambiguous version A3DPW.

5.4.1 Datasets

We begin by testing on a recent animal dataset of dogs. Since this dataset is limited, we additionally run an evaluation on a Human3.6m dataset which is considerably more varied and challenging.

RGBD-Dog Dataset

Our evaluation begins with the recent RGBD-Dog dataset.

Our evaluation focuses on the Human3.6m (**H36M**) [50, 21] and **3DPW** datasets [140].

Human3.6M

H36M is one of the largest datasets of humans annotated with 3D pose using MoCap sensors.

Quantization n		5		10		25	
Mode reproj.	Flow weight	MPJPE	RE	MPJPE	RE	MPJPE	RE
		86.4	57.9	84.0	57.5	79.0	56.3
	✓	84.1	**57.0**	81.9	56.7	77.8	55.8
✓		82.7	57.5	79.9	57.0	76.2	55.9
✓	✓	**82.2**	57.1	**79.4**	**56.6**	**75.8**	**55.6**

Table 5.2 **Ablation study on 3DPW** removing either the normalizing flow or the mode re-projection losses and reporting the change in performance.

As common practice, we train on subjects S1, S5, S6, S7 and S8, and test on S9 and S11.

3D People in the Wild

3DPW is only used for evaluation and, following [64], we evaluate on its test set.

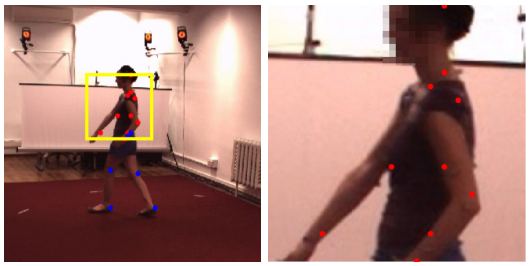


Fig. 5.5 Example image and corresponding annotation from the ambiguous H36M dataset **AH36M**. Best viewed in colour.

5.4.2 Evaluation Protocol

Our evaluation is consistent with [63, 64] - we report two metrics that compare the lifted dense 3D SMPL shape to the ground truth mesh: Mean Per Joint Position Error (**MPJPE**), Reconstruction Error (**RE**). For H36M, all errors are computed using an evaluation scheme known as "Protocol #2". Please refer to supplementary for a detailed explanation of MPJPE and RE.

5.4.3 Multipose metrics.

MPJPE and RE are traditional metrics that assume a single correct ground truth prediction for a given 2D observation. As mentioned above, such an assumption is rarely correct due to the inherent ambiguity of the monocular 3D shape estimation task. We thus also report MPJPE-n/RE-n an extension of MPJPE RE used in [67], that enables an evaluation of n different shape hypotheses. In more detail, to evaluate an algorithm, we allow it to output n possible predictions and, out of this set, we select the one that minimizes the MPJPE/RE metric. We report results for $n \in \{1, 5, 10, 25\}$.

5.4.4 Ambiguous H36M/3DPW (AH36M/A3DPW).

Since H36M is captured in a controlled environment, it rarely depicts challenging real-world scenarios such as body occlusions that are the main source of ambiguity in the single-view 3D shape estimation problem.

Hence, we construct an adapted version of H36M with synthetically-generated occlusions (fig. 5.5) by randomly hiding a subset of the 2D keypoints and re-computing an image crop around the remaining visible joints. Please refer to the supplementary for details of the occlusion generation process.

While 3DPW does contain real scenes, for completeness, we also evaluate on a noisy, and thus more challenging version (A3DPW) generated according to the aforementioned strategy.

5.4.5 Baselines

Our method is compared to two multi-pose prediction baselines. For fairness, both baselines extend the same (state-of-the-art) trunk architecture as we use, and all methods have access to the same training data.

SMPL-MDN follows [67] and outputs parameters of a mixture density model over the set of SMPL log-rotation pose parameters. Since a naïve implementation of the MDN model leads to poor performance ($\approx$ 200mm MPJPE-$n = 5$ on H36M), we introduced several improvements that allow optimization of the total loss eq. (5.8). **SMPL-CVAE**, the second baseline, is a conditional variational autoencoder [118] combined with our trunk network. SMPL-CVAE consists of an encoding network that maps a ground truth SMPL mesh V to a gaussian vector z which is fed together with an encoding of the image to generate a mesh V' such that $V' \approx V$. At test time, we sample n plausible human meshes by drawing $z \sim \mathcal{N}(0,1)$ to evaluate with MPJPE-n/RE-n. More details of both SMPL-CVAE and SMPL-MDN have been deferred to the supplementary material.

For completeness, we also compare to three more baselines that tackle the standard single-mesh prediction problem: HMR [56], GraphCMR [91], and SPIN [63], where the latter currently attain state-of-the-art performance on H36M/3DPW. All methods were trained on H36M [50], MPI-INF-3DHP [80], LSP [54], MPII [5] and COCO [68].

5.4.6 Results

Table 5.1 contains a comprehensive summary of the results on all 3 benchmarks. Our method outperforms the SMPL-CVAE and SMPL-MDN in all metrics on all datasets. For SMPL-

CVAE, we found that the encoding network often "cheats" during training by transporting all information about the ground truth, instead of only encoding the modes of ambiguity. The reason for a lower performance of SMPL-MDN is probably the representation of the probability in the space of log-rotations, rather in the space of vertices. Modelling the MDN in the space of model vertices would be more convenient due to being more relevant to the final evaluation metric that aggregates per-vertex errors, however, fitting such high-dimensional (dim=6890×3) Gaussian mixture is prohibitively costly.

Furthermore, it is very encouraging to observe that our method is also able to outperform the single-mode baselines [56, 64, 63] on the single mode MPJPE on both H36M and 3DPW. This comes as a surprise since our method has not been optimized for this mode of operation. The difference is more significant for 3DPW which probably happens because 3DPW is not used for training and, hence, the normalizing flow prior acts as an effective filter of predicted outlier poses. Qualitiative results are shown in fig. 5.6.

Ablation study.

We further conduct an ablative study on 3DPW that removes components of our method and measures the incurred change in performance. More specifically, we: 1) ablate the hypothesis reprojection loss; 2) set $p(X|I) = \text{Uniform}$ in eq. (5.3), effectively removing the normalizing flow component and executing unweighted K-Means in n-quantized-best-of-M. Table 5.2 demonstrates that removing both contributions decreases performance, validating our design choices.

5.5 Conclusions

In this work, we have explored a seldom visited problem of representing the set of plausible 3D meshes corresponding to a single ambiguous input image of a human. To this end, we have proposed a novel method that trains a single multi-hypothesis best-of-M model and, using a novel n-quantized-best-of-M strategy, allows to sample an arbitrary number $n < M$ of hypotheses.

Importantly, this proposed quantization technique leverages a normalizing flow model, that effectively filters out the predicted hypotheses that are unnatural. Empirical evaluation reveals performance superior to several strong probabilistic baselines on Human36M, its challenging ambiguous version, and on 3DPW. Our method encounters occasional failure cases, such as when tested on individuals with unusual shape (e.g. obese people), since we have very few of these examples in the training set. Tackling such cases would make for interesting and worthwhile future work.

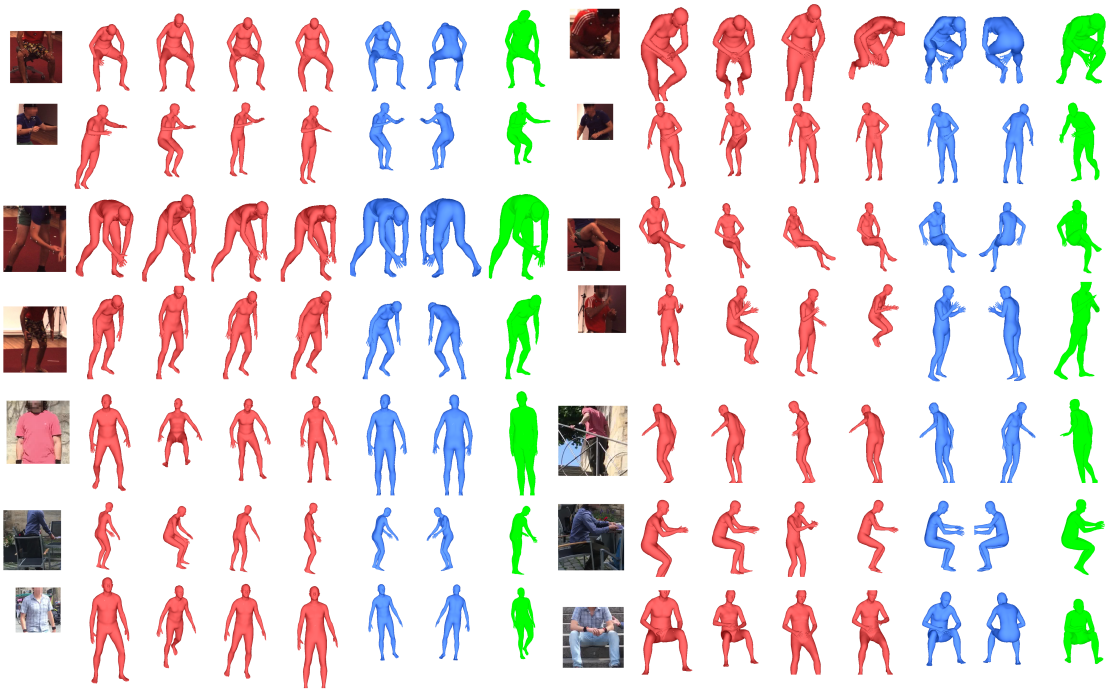


Fig. 5.6 **Qualitative results from $n = 5$ quantization on monocular mesh recovery on AH36m and A3DPW.** From left to right, each group of figures depicts the input ambiguous image, five network hypotheses with the closest to the ground truth in blue, and the ground truth pose in green.

Chapter 6

Conclusions

6.1 Discussion and Limitations

In this section I will conclude and discuss limitations

6.1.1 Discussion

Talk about meshes, radiance fields etc.

6.1.2 Applications in Animal Tracking

Discussion as to what GSK have been doing.

6.1.3 Future Work

What needs to happen etc.

References

[1] Adobe Systems Inc. (2018). Creating a green screen key using ultra key. https://helpx.adobe.com/premiere-pro/atv/cs5-cs55-video-tutorials/creating-a-green-screen-key-using-ultra-key.html. Accessed: 2018-03-14.

[2] Agudo, A., Pijoan, M., and Moreno-Noguer, F. (2018). Image collection pop-up: 3d reconstruction and clustering of rigid and non-rigid categories.

[3] Akhter, I. and Black, M. J. (2015). Pose-conditioned joint angle limits for 3D human pose reconstruction.

[4] Alldieck, T., Magnor, M., Bhatnagar, B. L., Theobalt, C., and Pons-Moll, G. (2019). Learning to reconstruct people in clothing from a single rgb camera. pages 1175–1186.

[5] Andriluka, M., Pishchulin, L., Gehler, P., and Schiele, B. (2014). 2D human pose estimation: New benchmark and state of the art analysis.

[6] Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., and Davis, J. (2005). SCAPE: shape completion and animation of people. In *ACM Trans. on Graphics*.

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

[8] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359.

[9] Biggs, B. (2020). Todo.

[10] Biggs, B., Roddick, T., Fitzgibbon, A., and Cipolla, R. (2018). Creatures Great and SMAL: Recovering the shape and motion of animals from video.

[11] Bishop, C. M. (1994). Mixture density networks. Technical report, Aston University.

[12] Blanz, V. and Vetter, T. (1999). A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, page 187–194, USA. ACM Press/Addison-Wesley Publishing Co.

[13] Blum, H. (1967). A transformation for extracting new descriptors of shape. *Models for Perception of Speech and Visual Forms, 1967*, pages 362–380.

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

[15] Bristow, H., Valmadre, J., and Lucey, S. (2015). Dense semantic correspondence where every pixel is a classifier. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4024–4031.

[16] Cao, C., Weng, Y., Zhou, S., Tong, Y., and Zhou, K. (2014). Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425.

[17] Cao, J., Tang, H., Fang, H., Shen, X., Tai, Y., and Lu, C. (2019a). Cross-domain adaptation for animal pose estimation. pages 9497–9506.

[18] Cao, Z., Hidalgo, G., Simon, T., Wei, S. E., and Sheikh, Y. (2019b). Openpose: Realtime multi-person 2d pose estimation using part affinity fields. 43(1):172–186.

[19] Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). Realtime multi-person 2D pose estimation using part affinity fields.

[20] Cashman, T. J. and Fitzgibbon, A. W. (2013). What shape are dolphins? Building 3D morphable models from 2D images. 35(1):232–244.

[21] Catalin Ionescu, Fuxin Li, C. S. (2011). Latent structured models for human pose estimation.

[22] Charles, J., Pfister, T., Magee, D., and Hogg, D. Zisserman, A. (2016). Personalizing human video pose estimation. In *Conference on Computer Vision and Pattern Recognition*.

[23] Chen, L., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation.

[24] Cheng, Y., Yang, B., Wang, B., Yan, W., and Tan, R. T. (2019). Occlusion-aware networks for 3d human pose estimation in video.

[25] Dai, H., Pears, N., and Smith, W. (2018). A data-augmented 3d morphable model of the ear. In *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, pages 404–408.

[26] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. CVPR '05, page 886–893, USA. IEEE Computer Society.

[27] Dang, Q., Yin, J., Wang, B., and Zheng, W. (2019). Deep learning based 2d human pose estimation: A survey. *Tsinghua Science and Technology*, 24(6):663–676.

[28] Diamond, S. and Boyd, S. (2016). CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5.

[29] Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2017). Density estimation using Real NVP.

[30] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338.

[31] Faugeras, O. and Luong, Q.-T. (2001). *The Geometry of Multiple Images*. MIT Press.

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

[33] Fischler, M. A. and Elschlager, R. A. (1973). The representation and matching of pictorial structures. *IEEE Trans. Comput.*, 22(1):67–92.

[34] Food and Agriculture Organization of the United Nations (2016). FAOSTAT statistics database. [Online; data retrieved from FAOSTAT on 21-November-2017].

[35] Gerig, T., Morel-Forster, A., Blumer, C., Egger, B., Luthi, M., Schoenborn, S., and Vetter, T. (2018). Morphable face models - an open framework. In *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, pages 75–82.

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

[37] Graving, J. M., Chae, D., Naik, H., Li, L., Koger, B., Costelloe, B. R., and Couzin, I. D. (2019). Deepposekit, a software toolkit for fast and robust animal pose estimation using deep learning. *eLife*, 8:e47994.

[38] Guan, P., Weiss, A., Balan, A. O., and Black, M. J. (2009). Estimating human shape and pose from a single image.

[39] Güler, R. A., Neverova, N., and Kokkinos, I. (2018). Densepose: Dense human pose estimation in the wild. pages 7297–7306.

[40] Guzman-Rivera, A., Batra, D., and Kohli, P. (2012). Multiple choice learning: Learning to produce multiple structured outputs. pages 1799–1807.

[41] Han, X., Leung, T., Jia, Y., Sukthankar, R., and Berg, A. C. (2015). Matchnet: Unifying feature and metric learning for patch-based matching.

[42] Hannah, M. J. (1974). Computer matching of areas in stereo images. AAI7427032.

[43] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151.

[44] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. pages 770–778.

[45] Hogg, D. (1983). Model-based vision: a program to see a walking person. *Image and Vision Computing*, 1(1):5–20.

[46] Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.

[47] Horn, B. K. P. and Schunck, B. G. (1981). Determining optical flow. *Artifical Intelligence*, 17(1–3):185–203.

[48] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269.

[49] Huang, Y., Bogo, F., Lassner, C., Kanazawa, A., Gehler, P. V., Romero, J., Akhter, I., and Black, M. J. (2017). Towards accurate marker-less human shape and pose estimation over time.

[50] Ionescu, C., Papava, D., Olaru, V., and Sminchisescu, C. (2014). Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. 36(7):1325–1339.

[51] Jakab, T., Gupta, A., Bilen, H., and Vedaldi, A. (2018). Unsupervised learning of object landmarks through conditional image generation. pages 1–12. Thirty-second Conference on Neural Information Processing Systems, NIPS 2018 ; Conference date: 03-12-2018 Through 08-12-2018.

[52] Johnson, S. and Everingham, M. (2010). Clustered pose and nonlinear appearance models for human pose estimation. pages 12.1–12.11.

[53] Johnson, S. and Everingham, M. (2011). Learning effective human pose estimation from inaccurate annotation. In *CVPR 2011*, pages 1465–1472.

[54] Johnson, S. and Everingham, M. (2011). Learning effective human pose estimation from inaccurate annotation.

[55] Joo, H., Simon, T., and Sheikh, Y. (2018). Total capture: A 3D deformation model for tracking faces, hands, and bodies.

[56] Kanazawa, A., Black, M. J., Jacobs, D. W., and Malik, J. (2018a). End-to-end recovery of human shape and pose.

[57] Kanazawa, A., Jacobs, D. W., and Chandraker, M. (2016). Warpnet: Weakly supervised matching for single-view reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3253–3261.

[58] Kanazawa, A., Tulsiani, S., Efros, A. A., and Malik, J. (2018b). Learning category-specific mesh reconstruction from image collections. pages 371–386.

[59] Kato, H., Ushiku, Y., and Harada, T. (2018). Neural 3d mesh renderer.

[60] Khamis, S., Taylor, J., Shotton, J., Keskin, C., Izadi, S., and Fitzgibbon, A. (2015). Learning an efficient model of hand shape variation from depth images. IEEE.

[61] Khoreva, A., Benenson, R., Ilg, E., Brox, T., and Schiele, B. (2017). Lucid data dreaming for object tracking. *The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*.

[62] Khosla, A., Jayadevaprakash, N., Yao, B., and Fei-Fei, L. (2011). Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO.

[63] Kolotouros, N., Pavlakos, G., Black, M. J., and Daniilidis, K. (2019a). Learning to reconstruct 3D human pose and shape via model-fitting in the loop.

[64] Kolotouros, N., Pavlakos, G., and Daniilidis, K. (2019b). Convolutional mesh regression for single-image human shape reconstruction.

[65] Laine, S., Karras, T., Aila, T., Herva, A., Saito, S., Yu, R., Li, H., and Lehtinen, J. (2017). Production-level facial performance capture using deep convolutional neural networks. page 10. ACM.

[66] Lassner, C., Romero, J., Kiefel, M., Bogo, F., Black, M. J., and Gehler, P. V. (2017). Unite the people: Closing the loop between 3D and 2D human representations.

[67] Li, C. and Lee, G. H. (2019). Generating multiple hypotheses for 3d human pose estimation with mixture density network.

[68] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context.

[69] Lin, Y.-L. and Wang, M.-J. J. (2014). Digital human modeling and clothing virtual try-on. In *Proceedings of the 2014 International Conference on Industrial Engineering and Operations Management Bali, Indonesia*.

[70] Liu, C., Yuen, J., and Torralba, A. (2011). Sift flow: Dense correspondence across scenes and its applications. 33(5):978–994.

[71] Liu, J., Kanazawa, A., Jacobs, D., and Belhumeur, P. (2012). Dog breed classification using part localization. pages 172–185. Springer.

[72] Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.

[73] Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., and and, M. J. B. (2015). SMPL: A skinned multi- person linear model. *ACM Trans. on Graphics*.

[74] Loper, M. M. and Black, M. J. (2014). OpenDR: An approximate differentiable renderer. pages 154–169. Springer.

[75] Lourakis, M. and Argyros, A. A. (2005). Is Levenberg-Marquardt the most efficient optimization algorithm for implementing bundle adjustment? pages 1526–1531.

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

[77] Ma, Q., Yang, J., Ranjan, A., Pujades, S., Pons-Moll, G., Tang, S., and Black, M. J. (2020). Learning to dress 3d people in generative clothing. In *Computer Vision and Pattern Recognition (CVPR)*, pages 6468–6477. IEEE.

[78] Martinez, J., Romero, J., Kiefel, M., Bogo, F., Black, M. J., and Gehler, P. V. (2017). A simple yet effective baseline for 3D human pose estimation.

[79] Mathis, A., Mamidanna, P., Cury, K. M., Abe, T., Murthy, V. N., Mathis, M. W., and Bethge, M. (2018). Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*.

[80] Mehta, D., Rhodin, H., Casas, D., Fua, P., Sotnychenko, O., Xu, W., and Theobalt, C. (2017a). Monocular 3d human pose estimation in the wild using improved cnn supervision. IEEE.

[81] Mehta, D., Sridhar, S., Sotnychenko, O., Rhodin, H., Shafiei, M., Seidel, H.-P., Xu, W., Casas, D., and Theobalt, C. (2017b). VNect: Real-time 3d human pose estimation with a single RGB camera.

[82] Moravec, H. (1980). Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical report, Carnegie Mellon University, Pittsburgh, PA.

[83] Nevatia, R. and Binford, T. O. (1977). Description and recognition of curved objects. *Artificial Intelligence*, 8(1):77–98.

[84] Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohi, P., Shotton, J., Hodges, S., and Fitzgibbon, A. (2011). KinectFusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE.

[85] Newell, A., Yang, K., and Deng, J. (2016). Stacked hourglass networks for human pose estimation.

[86] Novotny, D., Ravi, N., Graham, B., Neverova, N., and Vedaldi, A. (2019). C3DPO: Canonical 3d pose networks for non-rigid structure from motion.

[87] Omran, M., Lassner, C., Pons-Moll, G., Gehle, P. V., and Schiele, B. (2018). Neural body fitting: Unifying deep learning and model based human pose and shape estimation.

[88] Osman, A. A. A., Bolkart, T., and Black, M. J. (2020). STAR: A spare trained articulated human body regressor. In *European Conference on Computer Vision (ECCV)*.

[89] Park, J. and Boyd, S. (2017). General heuristics for nonconvex quadratically constrained quadratic programming.

[90] Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A. A. A., Tzionas, D., and Black, M. J. (2019). Expressive body capture: 3D hands, face, and body from a single image.

[91] Pavlakos, G., Zhu, L., Zhou, X., and Daniilidis, K. (2018). Learning to estimate 3D human pose and shape from a single color image.

[92] Paysan, P., Knothe, R., Amberg, B., Romdhani, S., and Vetter, T. (2009). A 3d face model for pose and illumination invariant face recognition. In *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 296–301.

[93] Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., and Sorkine-Hornung, A. (2016). A benchmark dataset and evaluation methodology for video object segmentation.

[94] Pereira, T., Aldarondo, D., Willmore, L., Kislin, M., Wang, S., Murthy, M., and Shaevitz, J. W. (2018). Fast animal pose estimation using deep neural networks. *bioRxiv*.

[95] Pfister, T., Charles, J., and Zisserman, A. (2015). Flowing convnets for human pose estimation in videos. In *International Conference on Computer Vision*.

[96] Pfister, T., Simonyan, K., Charles, J., and Zisserman, A. (2014). Deep convolutional neural networks for efficient pose estimation in gesture videos. In *Asian Conference on Computer Vision*.

[97] Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P. V., and Schiele, B. (2016). DeepCut: Joint subset partition and labeling for multi person pose estimation. pages 4929–4937.

[98] Poppe, R. (2007). Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding*, 108(1):4–18. Special Issue on Vision for Human-Computer Interaction.

[99] Probst, T., Pani Paudel, D., Chhatkuli, A., and Van Gool, L. (2018). Incremental non-rigid structure-from-motion with unknown focal length.

[100] Rister, B., Horowitz, M. A., and Rubin, D. L. (2017). Volumetric image registration from invariant keypoints. *IEEE Transactions on Image Processing*, 26(10):4900–4910.

[101] Robinette, K. M., Blackwell, S., Daanen, H., Boehmer, M., and Fleming, S. (2002). Civilian American and European Surface Anthropometry Resource (CAESAR), Final Report. Volume 1. Summary. Technical report, SYTRONICS INC DAYTON OH.

[102] Rodriguez, A., Zhang, H., Klaminder, J., Brodin, T., Andersson, P., and Andersson, M. (2018). Toxtrac: A fast and robust software for tracking organisms. *Methods in Ecology and Evolution*, 9:460–464.

[103] Rogez, G., Weinzaepfel, P., and Schmid, C. (2018). LCR-Net++: Multi-person 2D and 3D pose detection in natural images.

[104] Rupprecht, C., Laina, I., DiPietro, R., Baust, M., Tombari, F., Navab, N., and Hager, G. D. (2017). Learning in an uncertain world: Representing ambiguity through multiple hypotheses.

[105] Saito, S., , Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., and Li, H. (2019). Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization.

[106] Sanakoyeu, A., Khalidov, V., McCarthy, M. S., Vedaldi, A., and Neverova, N. (2020). Transferring dense pose to proximal animal classes. In *CVPR*.

[107] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520.

[108] Sharma, S., Varigonda, P. T., Bindal, P., Sharma, A., and Jain, A. (2019). Monocular 3d human pose estimation by generation and ordinal ranking.

[109] Shelton, C. (2000). Morphable surface models. *International Journal of Computer Vision*, 38:75–91.

[110] Shotton, J., Fitzgibbon, A., Blake, A., Kipman, A., Finocchio, M., Moore, B., and Sharp, T. (2011). Real-time human pose recognition in parts from a single depth image. IEEE.

[111] Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., and Moore, R. (2013). Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124.

[112] Sidenbladh, H., Black, M. J., and Fleet, D. J. (2000). Stochastic tracking of 3d human figures using 2d image motion. ECCV '00, page 702–718, Berlin, Heidelberg. Springer-Verlag.

[113] Sigal, L., Balan, A., and Black, M. J. (2008). Combined discriminative and generative articulated pose and non-rigid shape estimation.

[114] Sminchisescu, C., Kanaujia, A., Li, Z., and Metaxas, D. (2005). Discriminative density propagation for 3d human motion estimation. volume 1, pages 390– 397 vol. 1.

[115] Sminchisescu, C. and Triggs, B. (2003). Kinematic jump processes for monocular 3d human tracking. CVPR'03, page 69–76, USA. IEEE Computer Society.

[116] Smith, C. (2006). *On Vertex-vertex Systems and Their Use in Geometric and Biological Modelling*. PhD thesis, University of Calgary, Calgary, Alta., Canada, Canada. AAINR19574.

[117] Smith, S. M. and Brady, J. M. (1995). Susan - a new approach to low level image processing. *International Journal of Computer Vision*, 23:45–78.

[118] Sohn, K., Lee, H., and Yan, X. (2015). Learning structured output representation using deep conditional generative models.

[119] Sorkine, O. and Alexa, M. (2007). As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4.

[120] Stebbing, R. and Fitzgibbon, A. (2017). Personal communication.

[121] Streuber, S., Quiros-Ramirez, M. A., Hill, M. Q., Hahn, C. A., Zuffi, S., O'Toole, A., and Black, M. J. (2016). Body Talk: Crowdshaping realistic 3D avatars with words. 35(4):54:1–54:14.

[122] Sun, K., Xiao, B., Liu, D., and Wang, J. (2019). Deep high-resolution representation learning for human pose estimation. In *CVPR 2019*.

[123] Sun, X., Xiao, B., Wei, F., Liang, S., and Wei, Y. (2018). Integral human pose regression.

[124] Sun, Y. and Murata, N. (2020). Cafm: A 3d morphable model for animals. In *2020 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 20–24.

[125] Sun, Y., Wang, X., and Tang, X. (2013). Deep convolutional network cascade for facial point detection. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, page 3476–3483, USA. IEEE Computer Society.

[126] Tan, V., Budvytis, I., and Cipolla, R. (2017). Indirect deep structured learning for 3D human body shape and pose prediction.

[127] Taylor, G. W., Fergus, R., Williams, G., Spiro, I., and Bregler, C. (2010). Pose-sensitive embedding by nonlinear nca regression. In Lafferty, J., Williams, C., Shawe-Taylor, J., Zemel, R., and Culotta, A., editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc.

[128] Taylor, J., Shotton, J., Sharp, T., and Fitzgibbon, A. (2012). The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. pages 103–110. IEEE.

[129] Tekin, B., Rozantsev, A., Lepetit, V., and Fua, P. (2016). Direct prediction of 3d body poses from motion compensated sequences. pages 991–1000.

[130] Thewlis, J., Bilen, H., and Vedaldi, A. (2017). Unsupervised learning of object frames by dense equivariant image labelling. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[131] Tola, E., Lepetit, V., and Fua, P. (2010). Daisy: An efficient dense descriptor applied to wide-baseline stereo. 32(5):815–830.

[132] Tompson, J., Jain, A., LeCun, Y., and Bregler, C. (2014). Joint training of a convolutional network and a graphical model for human pose estimation. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, page 1799–1807, Cambridge, MA, USA. MIT Press.

[133] Tort, A. B., Neto, W. P., Amaral, O. B., Kazlauckas, V., Souza, D. O., and Lara, D. R. (2006). A simple webcam-based approach for the measurement of rodent locomotion and other behavioural parameters. *Journal of neuroscience methods*, 157(1):91–97.

[134] Toshev, A. and Szegedy, C. (2014). DeepPose: Human pose estimation via deep neural networks. pages 1653–1660.

[135] Tulsiani, S. and Malik, J. (2015). Viewpoints and keypoints. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1510–1519.

[136] Tung, H.-Y. F., Tung, H.-W., Yumer, E., and Fragkiadaki, K. (2017). Self-supervised learning of motion capture.

[137] Varol, G., Ceylan, D., Russel, B., Yang, J., Yumer, E., Laptev, I., and Schmid, C. (2018). BodyNet: Volumetric inference of 3D human body shapes.

[138] Velardo, C. and Dugelay, J.-L. (2010). Weight estimation from visual body appearance. In *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*, pages 1–6. IEEE.

[139] Vicente, S. and Agapito, L. (2013). Balloon shapes: Reconstructing and deforming objects with volume from images. pages 223–230.

[140] von Marcard, T., Henschel, R., Black, M., Rosenhahn, B., and Pons-Moll, G. (2018). Recovering accurate 3d human pose in the wild using imus and a moving camera.

[141] Wei, S.-E., Ramakrishna, V., Kanade, T., and Sheikh, Y. (2016). Convolutional pose machines. In *CVPR*.

[142] Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., and Perona, P. (2010). Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology.

[143] Wikimedia Commons (2007a). Dolphin triangle mesh.

[144] Wikimedia Commons (2007b). Ray trace diagram.

[145] Wilhelm, N., Vögele, A., Zsoldos, R., Licka, T., Krüger, B., and Bernard, J. (2015). Furyexplorer: visual-interactive exploration of horse motion capture data. In *Visualization and Data Analysis 2015*, volume 9397, page 93970F.

[146] Xiang, D., Joo, H., and Sheikh, Y. (2019). Monocular total capture: Posing face, body, and hands in the wild.

[147] Xiao, B., Wu, H., and Wei, Y. (2018). Simple baselines for human pose estimation and tracking.

[148] Xu, H., Bazavan, E. G., Zanfir, A., Freeman, W., Sukthankar, R., and Sminchisescu, C. (2020a). Ghum & ghuml: Generative 3d human shape and articulated pose models.

[149] Xu, H., Bazavan, E. G., Zanfir, A., Freeman, W. T., Sukthankar, R., and Sminchisescu, C. (2020b). GHUM & GHUML: generative 3d human shape and articulated pose models. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 6183–6192. IEEE.

[150] Yang, Y. and Ramanan, D. (2013). Articulated human detection with flexible mixtures of parts. 35(12):2878–2890.

[151] Yi, K. M., Trulls, E., Lepetit, V., and Fua, P. (2016). Lift: Learned invariant feature transform. pages 467–483, Cham. Springer International Publishing.

[152] Zammit, G. V., Menz, H. B., and Munteanu, S. E. (2010). Reliability of the TekScan MatScan® system for the measurement of plantar forces and pressures during barefoot level walking in healthy adults. *Journal of foot and ankle research*, 3(1):11.

[153] Zanfir, A., Bazavan, E. G., Xu, H., Freeman, W., Sukthankar, R., and Sminchisescu, C. (2020). Weakly supervised 3d human pose and shape reconstruction with normalizing flows.

[154] Zanfir, A., Marinoiu, E., and Sminchisescu, C. (2018). Monocular 3D pose and shape estimation of multiple people in natural scenes — the importance of multiple scene constraints.

[155] Zhou, T., Jae Lee, Y., Yu, S. X., and Efros, A. A. (2015). Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[156] Zuffi, S., Kanazawa, A., Berger-Wolf, T., and Black, M. J. (2019). Three-d safari: Learning to estimate zebra pose, shape, and texture from images "in the wild".

[157] Zuffi, S., Kanazawa, A., and Black, M. J. (2018). Lions and tigers and bears: Capturing non-rigid, 3D, articulated shape from images.

[158] Zuffi, S., Kanazawa, A., Jacobs, D., and Black, M. J. (2017). 3D menagerie: Modeling the 3D shape and pose of animals. pages 5524–5532. IEEE.