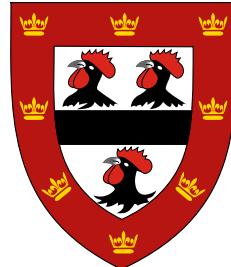




UNIVERSITY OF  
CAMBRIDGE

# 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*



"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  
June 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

<b>List of figures</b>	<b>xv</b>
<b>List of tables</b>	<b>xix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Approach . . . . .	2
1.2.1 Problem definition . . . . .	3
1.2.2 Relation to human reconstruction . . . . .	5
1.3 Contributions . . . . .	6
1.4 Co-Authored Papers . . . . .	6
1.5 Thesis Structure . . . . .	7
<b>2 Related Work</b>	<b>9</b>
2.1 Introduction . . . . .	9
2.2 Predicting point correspondences . . . . .	9
2.2.1 Relating separate views of the same object/scene . . . . .	10
2.2.2 Predicting semantically-meaningful keypoints . . . . .	11
2.3 Modelling articulated subjects . . . . .	13
2.3.1 Constructing 3D morphable models . . . . .	13
2.3.2 Modelling shapes (e.g. faces) . . . . .	15
2.3.3 Modelling articulation (e.g. hands) . . . . .	16
2.3.4 Modelling the human body surface . . . . .	18
2.3.5 Modelling animals . . . . .	19
2.4 Methods for monocular reconstruction of articulated subjects . . . . .	21
2.4.1 3D Pose Estimation . . . . .	22
2.4.2 Model-based human shape and pose . . . . .	25
2.4.3 Model-based animals . . . . .	28

<b>3 Learning from Synthetic Data; Bridging the Domain Gap</b>	<b>35</b>
3.1 Introduction . . . . .	35
3.1.1 Limited animal training data . . . . .	35
3.1.2 Synthetic image generation . . . . .	36
3.1.3 Handling silhouette ambiguity . . . . .	38
3.1.4 Contributions . . . . .	38
3.2 Preliminaries . . . . .	40
3.2.1 Deformable 3D quadruped model . . . . .	40
3.2.2 Camera model, joint reprojection and silhouette rendering . . . . .	40
3.2.3 System overview . . . . .	41
3.3 Predicting 2D joint candidate locations . . . . .	42
3.3.1 Generating synthetic quadruped images . . . . .	42
3.3.2 Prediction of 2D joint locations using multimodal heatmaps . . . . .	43
3.4 Optimal joint assignment (OJA) . . . . .	45
3.4.1 Basic formulation . . . . .	46
3.4.2 QP solution. . . . .	48
3.4.3 Incorporating coverage priors . . . . .	49
3.4.4 Formulation as a genetic algorithm . . . . .	50
3.5 3D model fitting . . . . .	51
3.6 Experiments . . . . .	53
3.6.1 BADJA Dataset . . . . .	53
3.6.2 Joint prediction . . . . .	53
3.6.3 Optimal joint assignment . . . . .	54
3.6.4 Model fitting . . . . .	55
3.6.5 Automatic silhouette prediction . . . . .	57
3.7 Conclusions . . . . .	58
<b>4 End-to-end Dog Shape Recovery with a Learned Shape Prior</b>	<b>59</b>
4.1 Introduction . . . . .	59
4.1.1 Adding local parameters to the PCA shape space . . . . .	60
4.1.2 Automatic and real-time 3D dog reconstruction . . . . .	60
4.1.3 Learning a 3D animal prior . . . . .	61
4.1.4 Refinement steps “in the loop” . . . . .	62
4.1.5 Contributions . . . . .	63
4.2 Building SMBLD: a new parametric dog model . . . . .	63
4.2.1 Introducing scale parameters . . . . .	66
4.2.2 Sharing scale parameters . . . . .	67
4.2.3 Initializing a dog shape/scale prior . . . . .	67
4.3 End-to-end dog reconstruction from monocular images . . . . .	70

4.3.1	Model architecture . . . . .	71
4.3.2	Training losses . . . . .	71
4.4	Expectation Maximization in the Loop . . . . .	73
4.4.1	Expectation Maximization update steps . . . . .	73
4.5	Building StanfordExtra: a new large-scale dog keypoint dataset . . . . .	74
4.5.1	Keypoints. . . . .	75
4.5.2	Segmentation. . . . .	75
4.5.3	Approximating the dataset complexity. . . . .	76
4.6	Experiments . . . . .	76
4.6.1	Evaluation protocol . . . . .	76
4.6.2	Training procedure . . . . .	77
4.6.3	Generalization to unseen dataset . . . . .	79
4.6.4	Ablation study . . . . .	80
4.6.5	PCK-MAX results . . . . .	81
4.6.6	Qualitative evaluation . . . . .	81
4.7	Conclusions . . . . .	83
<b>5</b>	<b>Handling Ambiguous Input with Multi-Output Learning</b>	<b>87</b>
5.1	Introduction . . . . .	87
5.1.1	Limitations of state-of-the-art 3D reconstruction techniques . . . . .	88
5.1.2	Modelling ambiguities in 3D reconstruction . . . . .	88
5.1.3	Overcoming limitations with best-of- $M$ frameworks . . . . .	94
5.1.4	Methods to learn a 3D articulated pose prior . . . . .	95
5.1.5	Sourcing datasets for training and evaluation . . . . .	95
5.1.6	System overview . . . . .	96
5.2	Preliminaries . . . . .	96
5.2.1	3D Morphable Model (SMPL/SMAL). . . . .	97
5.2.2	Predicting the 3D model parameters from a single image. . . . .	97
5.2.3	Normalizing flows. . . . .	97
5.3	Method . . . . .	98
5.3.1	Learning with multiple hypotheses . . . . .	98
5.3.2	Best-of- $M$ loss. . . . .	99
5.3.3	Limitations of best-of- $M$ . . . . .	99
5.3.4	$n$ -quantized-best-of- $M$ . . . . .	99
5.3.5	Learning the pose prior with normalizing flows. . . . .	100
5.3.6	2D re-projection loss. . . . .	100
5.3.7	Mini SMAL/SMPL optimization . . . . .	101
5.3.8	Overall loss. . . . .	101
5.4	Experiments . . . . .	102

5.4.1	Datasets . . . . .	102
5.4.2	Evaluation Protocol . . . . .	103
5.4.3	Multipose metrics. . . . .	104
5.4.4	Multi-hypothesis baselines . . . . .	104
5.4.5	Single hypothesis baselines . . . . .	106
5.4.6	Training details . . . . .	106
5.4.7	Results . . . . .	106
5.5	Results on MS-COCO . . . . .	106
5.6	Conclusions . . . . .	107
<b>6</b>	<b>Conclusions</b>	<b>111</b>
6.1	Discussion and Limitations . . . . .	111
6.1.1	Discussion . . . . .	111
6.1.2	Applications in Animal Tracking . . . . .	111
6.1.3	Future Work . . . . .	111
<b>References</b>		<b>113</b>

# List of figures

1.1	An example input video sequence. . . . .	3
1.2	Sample output printed from Deformable Mesh Animation [110]. . . . .	3
1.3	An example prior, in this case a template mesh. . . . .	4
1.4	Varying human shape parameters while pose remains fixed. Reprinted from [111]. . . . .	4
2.1	A polygon mesh [129]. . . . .	14
2.2	Dinosaur mesh undergoing ARAP deformation, obtained by translating the highlighted yellow vertex. Reprinted from [109]. . . . .	14
2.3	SMPL model showing pose-invariant shape changes, reprinted from [69]. . . . .	20
2.4	SMAL with varying shape parameters. . . . .	21
2.5	SMAL with varying pose parameters. . . . .	21
2.6	SMPLify: Fitting the SMPL model to the Leeds Sports Dataset. . . . .	27
2.7	8-parameter dolphin model with annotated contour (left) and contour generators (middle and right). . . . .	31
2.8	User input required for the deformable mesh animation algorithm, reprinted from [110].	31
2.9	Example of an dog template being fit to input video sequence, reprinted from [110] .	32
2.10	Fitting SMAL to a hand segmented animal, reprinted from [143]. . . . .	32
2.11	Diagram showing raycast rendering. [130]. . . . .	33
3.1	<b>System overview:</b> input video (a) is automatically processed using DeepLabv3+ [22] 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). . . . .	39
3.2	Example predictions from a network trained on unimodal (top) and multi-modal (bottom) ground-truth for front-left leg joints. . . . .	45
3.3	Example outputs from the joint prediction network, with maximum likelihood predictions linked into skeleton. . . . .	45

---

3.4	Skeleton Prior: Synthetic quadruped training data examples (rendered with texture to show 3D) generated by sampling pose, shape and position parameters and applying to SMAL model (a), (b). The set of 2D skeletal positions are used to create a vector of means $\mu \in \mathbb{R}^{2J}$ and covariance matrix $\Sigma \in \mathbb{R}^{2J \times 2J}$ . The Gaussian distribution constructed can be sampled to create new skeletons, such as those shown in (c), (d).	48
3.5	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).	50
3.6	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).	50
3.7	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.	54
3.8	Example skeletons from raw predictions (a), processed with OJA-QP (b), and OJA-GA (c).	55
3.9	Results are comparable in quality to SMAL [143], but note that we do not require hand-clicked keypoints.	56
3.10	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.	56
3.11	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.	57
3.12	Failure modes of the proposed system. <i>Left:</i> Missing interior contours prevent the optimizer from identifying which way the dog is facing. <i>Middle:</i> The model has never seen an elephant, so assumes the trunk is the tail. <i>Right:</i> Heavy occlusion. The model interprets the tree as background and hence the silhouette term tries to minimize coverage over this region.	57
4.1	<b>End-to-end Dog Shape Recovery with a Learned Shape Prior.</b> 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.	63

---

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 $\beta$ , pose $\theta$ , camera focal length $f$ and translation $t$ (3) skinning functions $F_v$ and $F_t$ 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. . . . .	64
4.3 <b>Effect of varying SMBLD scale parameters.</b> <i>From left to right:</i> Mean SMBLD model, 25% leg elongation, 50% tail elongation, 50% ear elongation. . . . .	68
4.4 <b>StanfordExtra example images.</b> <i>Left:</i> outlined segmentations and labelled keypoints for 24 representative images. <i>Right:</i> heatmap of deviation of worker submitted results from mean for each submission. . . . .	74
4.5 Accuracy distribution of all submitted dog segmentations across the entire StanfordExtra dataset. . . . .	76
4.6 <b>Qualitative comparison to SOTA.</b> Row 1: <b>Ours</b> , Row 2: 3D-M [143], Row 3: CGAS [9]. (a) input image, (b) predicted 3D mesh, (c) canonical view 3D mesh, (d) reprojected model joints and (e) silhouette reprojection error. . . . .	79
4.7 <b>Qualitative comparison to SOTA (continued).</b> Row 1: <b>Ours</b> , Row 2: 3D-M [143], Row 3: CGAS [9]. (a) input image, (b) predicted 3D mesh, (c) canonical view 3D mesh, (d) reprojected model joints and (e) silhouette reprojection error. . . . .	80
4.8 <b>Qualitative results on StanfordExtra.</b> 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. . . . .	84
4.9 <b>Qualitative results on StanfordExtra (continued).</b> 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. . . . .	85
4.10 <b>Qualitative results on StanfordExtra (continued).</b> 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. . . . .	86
4.11 <b>Qualitative results on Animal Pose [16].</b> 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. . . . .	86
5.1 <b>Human mesh recovery in an ambiguous setting.</b> 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. . . . .	88

---

5.2	<b>Top:</b> Pretrained SPIN model tested on an ambiguous example, <b>Bottom:</b> SPIN model after fine-tuning to ambiguous examples. Note the network tends to regress to the mean over plausible poses, shown by predicting the missing legs vertically downward — arguably the average position over the training dataset. . . . .	89
5.3	<b>Overview of our method.</b> 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 $f$ . . . . .	96
5.4	<b>Example samples from the normalizing flow</b> $f : X \mapsto z$ ; $p(z) \sim \mathcal{N}(0, 1)$ , trained on a dataset of ground truth 3D SMPL control skeletons $\{X_1, \dots, X_N\}$ . . . . .	101
5.5	Example image and corresponding annotation from the ambiguous H36M dataset <b>AH36M</b> . Best viewed in colour. . . . .	104
5.6	<b>Qualitative results from <math>n = 5</math> quantization on monocular mesh recovery on AH36m and A3DPW.</b> 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. . . . .	108
5.7	<b>Qualitative results from <math>n = 5</math> quantization on monocular mesh recovery on AH36m and A3DPW (continued).</b> 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. . . . .	109
5.8	<b>Qualitative results on monocular mesh recovery on MS-COCO</b> From left to right, each group of figures depicts the input image with three hypotheses of our network. Note that no 3D ground truth exists for this dataset. . . . .	110

# List of tables

2.1	Dataset summary: analysis of available datasets. Bold rows are datasets introduced in this thesis. S2: 2D Silhouettes, BB2: 2D bounding boxes, J2: 2D Joints, MM3: 3D Morphable Model. P3: 3D Priors. M3: 3D Model. D: dog, Ca: cat, Ho: horse, Z: zebra, S: sheep, E: elephant, Co: cow, B: bear, G: giraffe, T: tiger, L: lion: Ch: cheetah, Hi: hippo . . . . .	29
2.2	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. . . . .	30
3.1	Literature summary: Analysis of required input annotations. MLQ: Medium-to-large quadrupeds. J2: 2D Joints. S2: 2D Silhouettes. T3: 3D Template. P3: 3D Priors. M3: 3D Model. . . . .	36
3.2	Example inputs and output to the joint assignment problem. Non maximum suppression applied to predicted heatmap tensors yields a set of proposals $p$ for each each skeleton joint $j$ at 2D location $x_{jp}$ . $a_{jp}$ is an illustrative assignment vector which is predicted by the OJA algorithm. . . . .	46
3.3	Assignment variables $A = \bar{a}_j = \{a_{jp}\} \in \{0, 1\}^{N_j+1}$ for the current frame stored as a jagged array. . . . .	46
3.4	2D joint locations $X^* = X(A)$ selected by the assignment variable $A$ . . . . .	46
3.5	Accuracy of OJA on BADJA test sequences. . . . .	55
3.6	Quantitative evaluation on synthetic test sequences. The performance of the raw network outputs and OJA methods are evaluated using the probability of correct keypoint (PCK) metric. Mesh fitting accuracy is evaluated by computing the mean distance between the predicted and ground truth vertices. . . . .	55

4.4	<b>PCK-MAX baselines.</b> PCK-MAX scores are shown for SOTA methods under varying conditions. Directly comparable baseline methods (requiring only an input image) are highlighted. <i>Pred</i> keypoints generated with Hourglass-Net [79] and segmentations with DeepLab v3+ [? ]. 3D-M/CGAS are also analysed when they have access to ground-truth keypoints and/or segmentation masks. We also analyse adding this paper’s innovations (scale parameters and EM prior) to the 3D-M method [143]. . . . .	82
4.5	<b>PCK-MAX Animal Pose dataset [16].</b> Evaluation on recent Animal Pose dataset with no fine-tuning to our method nor joint/silhouette predictors used for 3D-M. . . . .	82
4.6	<b>PCK-MAX ablation study.</b> Evaluation with the following contributions removed: (a) EM updates, (b) Mixture Shape Prior, (c) SMBLD scale parameters. . . . .	82
5.1	<b>Monocular multi-hypothesis human mesh recovery</b> comparing our approach to two multi-hypothesis baselines (SMPL-CVAE, SMPL-MDN) and state-of-the-art single mode evaluation models [60, 61, 53] on Human3.6m (H36M), its ambiguous version AH36M, on 3DPW and its ambiguous version A3DPW. . . . .	102
5.2	<b>Ablation study on 3DPW</b> removing either the normalizing flow or the mode re-projection losses and reporting the change in performance. . . . .	103

# 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 [33]. 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 presence 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 occurred 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 [138], or by installing an overhead camera which performs simple visual blob detection via colour thresholding [122], [93]. 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 ??.

### 1.2.1 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 [110].

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 controllerless gameplay. However, recovering 3D models from human subjects is a growing field. Applications are found in fashion to facilitate online ‘try-ons’ for virtual clothing [65], in animation and visual effects to generate virtual characters from live actor performances [62], and in healthcare for tracking patients’ body weight over time [125]. 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.

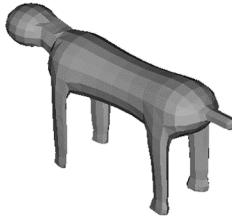


Fig. 1.3 An example prior, in this case a template mesh.

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 [111].

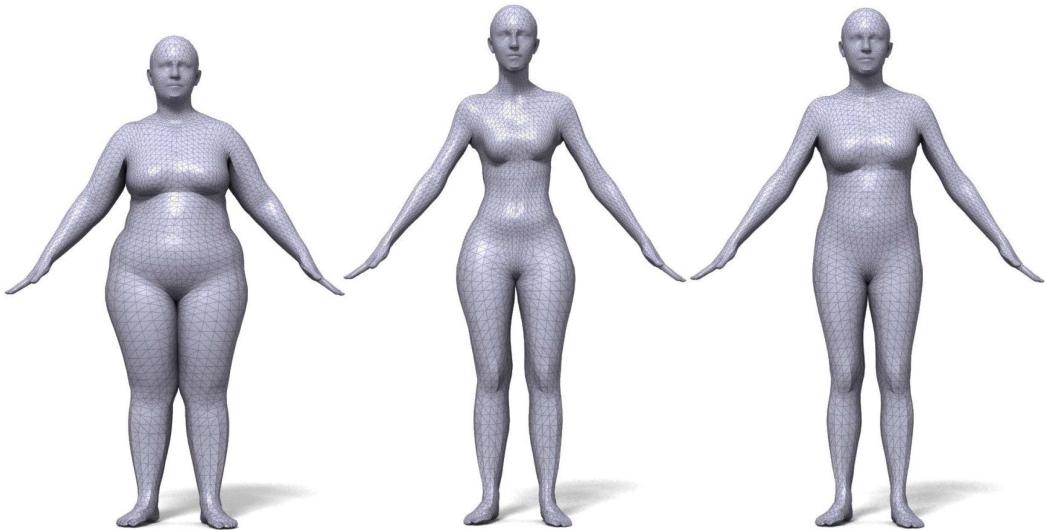


Fig. 1.4 Varying human shape parameters while pose remains fixed. Reprinted from [111].

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 [78] 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 [4, 64, 49]. Of these two classes of labelling, animal *segmentation* data is available in datasets such as MSCOCO [64], PASCAL VOC [29] and DAVIS [85]. 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 [131], it remains impractical to consider mocap for a family of tigers at play.

## 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 3 contains work from:

**Benjamin Biggs**, Thomas Roddick, Andrew Fitzgibbon and Roberto Cipolla. *Creatures Great and SMAL: Recovering the Shape and Motion of Animals from Video*. ACCV 2018, Oral Presentation.

Chapter 4 contains work from:

**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.

Chapter 5 contains work from:

**Benjamin Biggs**, Sebastien Ehrhardt, 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 thesis chapters discuss methods for deriving 3D animal reconstructions from monocular input images and video. Chapter 1 begins by covering the necessary background and motivation before an in-depth literature review in Chapter 2. Chapter 3 discusses an approach for animal reconstruction learning only from synthetic training data. Chapter 4 describes an end-to-end and real-time technique applied to the challenging dog category. Chapter 5 introduces a method for handling input images with significant ambiguity. The final Chapter 6 summarizes the work and offers a discussion for worthy 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 [40] or optical flow [45] 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 pipelines would first identify *interest points* (typically corners [76, 41, 107] or blobs [72]), 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 [72], which encodes points according to local histograms of gradient orientations and was later speed-up by SURF [7] and DAISY [120]. There have been modern attempts to learn sophisticated feature representations using convolutional neural network architectures [137, 39], 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 corresponding 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 [91]) 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 semantically-meaningful keypoints

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 recognition accuracy. Human detection and pose estimation methods progressed from simple bounding box representations [25], to object part prediction [8, 8], poselets [35] and subsequently 2D keypoint localization [8, 8]. 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 [77] and fit shape primitives (e.g. cylinders [43]) to detected edges. Tree-based graphical models known as pictorial structures [32] were adopted and later made efficient [31]. Improvements were made with models capable of expressing complex relationships between joints, such as flexible part mixtures [136, 50].

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 [67] and birds [128]. 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 representations that can be learnt for joints that, when applied discriminatively, enable accurate recognition. An early example [116] 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 [114] and full body [123] landmarks. More recent works improve

robustness by regressing keypoint confidence maps [121] rather than 2D keypoints directly, enabling spatial priors to be applied to remove outliers [17, 87, 88, 21, 121, 124, 89]. More recent methods are able to directly produce accurate confidence maps through a multi-stage pipeline [127]. Of particular note are hourglass [79] (relied upon in this thesis Chapter 3) and multi-level [112, 133] structures, which combine global reasoning of full-body attributes and of fine-grained details. A related class of methods [37, 117] 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 parsing multiple subjects in challenging poses and in the presence of various occluders. However, part of what enables these achievements is the prevalence 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 [90, 26].

### **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 [66] which employed optical flow methods to match image using SIFT features, and Bristow et al. [14] 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. [140] demonstrate a method for solving correspondences across an image collection by enforcing cycle consistency. Kanazawa et al. [54] 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. [119] 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. [48] show they can estimate 2D human pose without training data by leveraging 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 [74], LEAP [86] and DeepPoseKit [36] exemplify such techniques, in which existing architectures [89, 79, 46, 98] 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 [16] 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 [37] described above to proximal animal classes [97], such as chimpanzees, by aligning the geometry between the animal category to humans for which data is plentiful.

## 2.3 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 appearance 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 octopus? 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 [11] presented the first 3DMM, which expressed a low-dimensional face 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 [15, 84, 34], including for cars [100], other human body parts including the hands [57] and ears [24], the human body surface [5, 69] and a restricted set of animal categories [113, 143].

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

### 2.3.1 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 [106]. Although other convex shapes are allowed, this thesis only has need to discuss



Fig. 2.1 A polygon mesh [129].

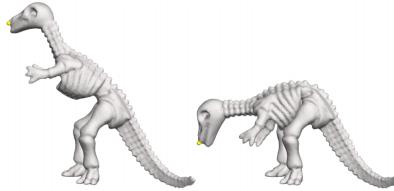


Fig. 2.2 Dinosaur mesh undergoing ARAP deformation, obtained by translating the highlighted yellow vertex. Reprinted from [109].

triangular mesh polygons, which henceforth will be referred to as *triangles*. An example mesh is shown in Figure 2.1.

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 necessary to employ techniques for regularization. One such regularizer is known as the *As Rigid as Possible* scheme:

**Definition 2.3.1** (As Rigid as Possible). As Rigid as Possible (ARAP) surface deformation [109] 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  $i$ :

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

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.

Once aligned, a  $d$ -dimensional *shape space* can then be defined (with  $d \ll n$ ), 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.

### 2.3.2 Modelling shapes (e.g. faces)

The concepts raised above were first introduced in the seminal work of Blanz and Vetter [11]. 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:

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

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 design decision explored further in this thesis Chapter 4). In addition, the function  $f(w)$  which defines the likelihood shape space vector  $\alpha$  represents a plausible face, is therefore given by the Mahalanbois distance of  $\alpha$  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:

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

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. [8], 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 [69].

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 [8] 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 [8, 8]. Possibly the closest to later sections which require a focus on *pose deformation* is the work of Wu et al [8], who combine 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 appearance is generally referred to as *texture*). Early work modelled shape and texture independently [8, 8], 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 [8, 8], who propose methods based on deep convolutional models to jointly model shape and texture.

### 2.3.3 Modelling articulation (e.g. hands)

3D morphable models have also influenced work in 3D hand tracking and modelling. Human hands serve 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 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 2.3.2** (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  $\gamma$  around the  $x$  axis:

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

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

$$p' = R_x(\theta)p \quad (2.5)$$

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  $\gamma$  rotation around the

$x$ -axis (pitch), followed by a  $\beta$  rotation around the  $y$ -axis (yaw) and finally by an  $\alpha$  rotation around the  $z$ -axis (roll).

$$R = R_z(\alpha)R_y(\beta)R_x(\gamma) \quad (2.6)$$

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 2.3.3** (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:

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

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  $\theta$  and input vertex position  $v$  as follows:

$$v' = LBS(R(\theta), v) \quad (2.8)$$

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 proportions.

Allen et al [8] 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 [8] 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 scanning plaster casts [8]. A significant advance was presented by Taylor et al [8], 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. [8] follow a similar process by making use of multi-view input.

### 2.3.4 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 [8] (although [8] 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 [5] 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 [69] for a comprehensive overview). An important advance was made by Hasler et al. [8], 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. [69]. 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 [92] 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  $\theta \in \mathbb{R}^{24 \times 3}$ . *Shape* is then given as coefficients for a learned shape basis  $\beta \in \mathbb{R}^{10}$ . The SMPL function can then be viewed as:

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

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 [132, 52, 82]. CAPE [73] 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 [80] 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. [135] 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.

### 2.3.5 Modelling animals

There is still relatively little work specifically focusing on the 3D scanning [8] 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

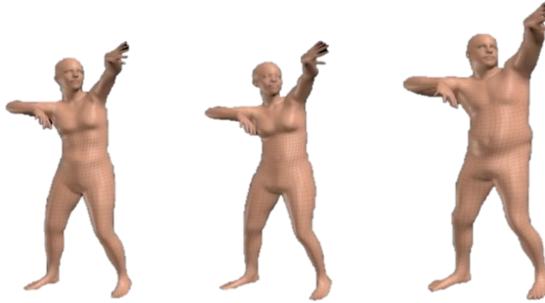


Fig. 2.3 SMPL model showing pose-invariant shape changes, reprinted from [69].

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 [8] describe a method for animating an artist-designed rigged 3D model, by tracking a 2D sequence. Chen et al. [8] learn a shape space by registering 11 3D shark models downloaded from the Internet. Cashman et al. [8] learn a morphable model of dolphin shapes by adapting a representative 3D model to 2D images. Ntouskos et al. [8] fit geometric primitives to manually-segmented animal parts generated from an input collection. Reinert et al. [8] demonstrate 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) [143], 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, and 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.

From the paper, SMAL is defined as a function  $\text{SMAL}(\theta, \beta)$  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  $\gamma$  is expressed by a further three parameters. The SMAL function returns a triangulated surface comprising  $6890 \times 3$  vertex locations. Chapter 3

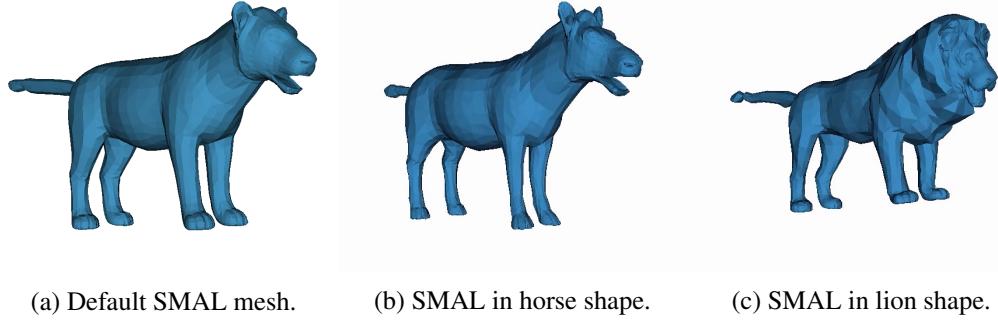


Fig. 2.4 SMAL with varying shape parameters.

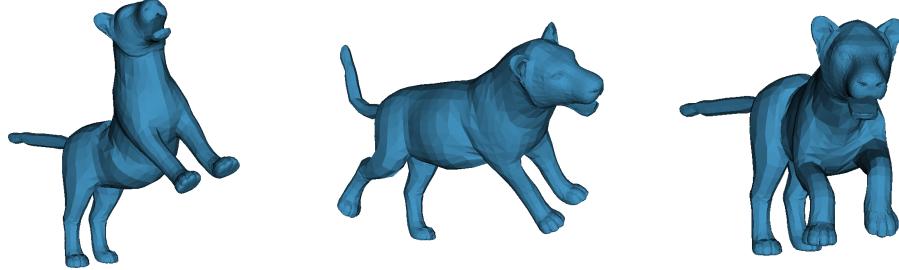


Fig. 2.5 SMAL with varying pose parameters.

and Chapter 4 of this thesis make use of the SMAL model in order to reconstruct various quadruped categories.

## 2.4 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 [123], 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 constraints.

The theme of this section is therefore to discuss how 3D morphable models (3DMMs) can be incorporated into 3D reconstruction pipelines. In general, algorithms take as input an image or video

and predict a set of 3D model parameters  $\alpha$  (often factored into shape  $\beta$  and pose  $\theta$ ). 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  $M = (V, T)$  with fixed triangulation  $T$ . For completeness and comparison, this section will make some mention of the small class of 3D reconstruction methods for articulated subjects 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 [8]. 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 [8] and later with Active Shape Models [8] 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 particularly on *humans* as a special case. The first example of such an approach is the seminal work of Blanz and Vetter [11] 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 accurate 3D deformable template models [69] generated from real human scans, 3D motion capture datasets [47, 126] and large 2D datasets [64, 49, 4] which provide keypoint and silhouette annotations. The combination of these publically available datasets and their incorporation 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 [8, 8].

#### 2.4.1 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 [102]), motion

capture (e.g. for digital character generation [8]), gait analysis (e.g. identifying lameness in cattle [8]) 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 [8], length ratios [8] or that limb lengths are isometric across individuals and vary only in global scaling [8]. More advanced techniques built statistical models of shape variation using anthropometric tables or learnt them from motion capture data [6].

A broad category of approaches for this are methods for *non-rigid structure from motion* [8]. 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  $S_1, \dots, S_k$  where  $S_i \in \mathbb{R}^{3 \times P}$ . Precisely:

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

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

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

or equivalently:

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

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:

$$\underbrace{\begin{bmatrix} u_i^{(1)} & \dots & u_P^{(1)} \\ v_i^{(1)} & \dots & v_P^{(1)} \\ u_i^{(2)} & \dots & u_P^{(2)} \\ v_i^{(2)} & \dots & v_P^{(2)} \\ \vdots & & \vdots \\ u_i^{(N)} & \dots & u_P^{(N)} \\ v_i^{(N)} & \dots & v_P^{(N)} \end{bmatrix}}_W = \underbrace{\begin{bmatrix} l_1^{(1)} R^{(1)} & \dots & l_K^{(1)} R^{(1)} \\ l_1^{(2)} R^{(2)} & \dots & l_K^{(2)} R^{(2)} \\ \vdots & & \vdots \\ l_1^{(N)} R^{(N)} & \dots & l_K^{(N)} R^{(N)} \end{bmatrix}}_Q \cdot \underbrace{\begin{bmatrix} S_1 \\ S_2 \\ \dots \\ S_K \end{bmatrix}}_B \quad (2.13)$$

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 [8, 8, 8], but start with a shape basis learnt from available motion capture datasets (e.g. CMU [8]).

More recent approaches were designed to be fully automatic. Shotton et al. [102] designed a commercially-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. [117] predict dense correspondences 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. [8, 8] form a probabilistic model that models both 3D pose and 2D keypoints jointly, overcoming noise among 2D body parts. Other approaches [8, 8] 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 [8] 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 [123], an approach which employs a CNN to reason jointly about 2D landmark detection and 3D pose estimation from single RGB images. Pishchulin et al. [89] 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 [8]) is required which is generally expensive to obtain, particularly for animal categories. One branch of approaches [118] predicts body configuration in terms of angles. Other approaches include Pavlakos et al. [8], who use a 2D joint predictor [8] followed by a deep architecture to regress 3D heatmaps. Moreno-Noguer [8] 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. [8] who demonstrate the effectiveness of a simple architecture at regressing accurate 3D keypoints from 2D predictions. This technique was later interpreted probabilistically by Li et al. [8], 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 [8] who train their network with data generated synthetically with a 3D human body model.

### 2.4.2 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 [8, 8, 8]. The introduction of the 3D deformable human body model known as SCAPE [8] enabled various fitting approaches. Sigal et al [8] compute shape features from manually extracted silhouettes and use a mixture of experts formulation for predicting SCAPE model parameters. Later, Guan et al. [8] 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. [8], Zhou et al. [8] and Chen et al. [8] present a similar approach, although show optimization only to input keypoints and manually or semi-manually (e.g. GraphCut [8]) extracted silhouettes.

A significant advance was made by the introduction of SMPLify [8], 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 [69] model to a set of 2D image locations predicted by DeepCut [8], 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  $J_{\text{SMPL}}$ . Precisely  $J_{\text{SMPL}} = R_\theta(J(\beta))$  where  $J(\beta)$  computes 3D skeleton positions from SMPL shape parameters  $\beta$ , and  $R_\theta$  is the global rigid transformation effected by SMPL pose parameters  $\theta$ . 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  $(\beta, \theta)$ , 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  $E_J(\beta, \theta; K, J_{\text{est}})$  and measures the weighted 2D distance between estimated keypoints  $J_{\text{est}}$  and the corresponding SMPL joints  $J_{\text{SMPL}}$ .

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

The weighted 2D distance is implemented using the Geman-McClure [8] penalty function  $\rho$  which helps deal with noisy DeepCut estimates. SMPLify implements  $\Pi_K$  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 2.4.1** (Primer on camera geometry). Perspective projection is a function which maps a 3D structure to blah blah.

$$2X = Y \quad (2.15)$$

Orthographic projection is the following:

$$3X = Z \quad (2.16)$$

The full energy formulation is then given as:

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

where the following energy terms are employed, balanced according to the  $\lambda$  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 Huang et al. [Huang Multiview] who fit SMPL to multi-view images and Pavlakos et al. [SMPL-X fitting] who extend SMPL with hand and facial expression parameters and follow a similar fitting procedure. 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. An early approach by Tan et al. [115] used deep neural networks to learn 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. Improvements were realised by incorporating an abundance of available human data into the training pipeline. Apart from 3D morphable models (e.g. SMPL [69]), methods use paired 3D motion capture data (e.g. Human3.6M [47, 20], 3DPW [126]) to help relate 2D appearance and the underlying 3D structure, unpaired 3D motion capture data (e.g. CMU [CMU]) for learning detailed priors over the distribution of human shapes and poses, and large 2D keypoint datasets (e.g. MSCOCO [COCO], LSP [LSP], MPI [MPI]) which help promote generalization to ‘in-the-wild’ scenarios. A notable work in this category is Human Mesh Recovery (HMR) of Kanazawa et al. [53], although multiple concurrent works exist [Hsiao-Yu Fish Tung], [pavlakos], [neural body fitting] and [DenseRaC]. Chapter 5 of this thesis will explore a method for modelling uncertainty in 3D reconstruction, by exploring extensions to the aforementioned architectures.

BodyNet [bodynet] follow a similar pipeline with the inclusion of a texture prediction module supervised by 2D body part segmentations [dataset]. Silhouette data has also been shown to assist in accurate reconstruction of clothes, hair and other appearance detail [2] [SiCloPe] [ARCH]. Methods typically represent 3D clothes as ‘freeform’ vertex deformations; in other words, deformations beyond the standard SMPL blend shapes which represent unclothed bodies. This presents significantly more

degrees of freedom in the optimization, which must be controlled; either by large training datasets with limited variation or mesh-based deformation priors such as ARAP.

While the dominant paradigm in human reconstruction is now end-to-end deep learning methods, SPIN [60] 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. This idea inspired the work presented in Chapter 4 of this thesis, in which a 3D dog shape prior is learnt during the training loop via expectation maximization.

There have also been a few recent works that reconstruct 3D humans without an explicit template prior. The general idea is to interpret a neural network as an implicit representation of the 3D surface. In particular, the network  $f$  is trained to map sampled 3D locations  $(x, y, z) \in \mathbb{R}^3$  to an occupancy value  $\{0, 1\}$  (and optionally a texture value  $(r, g, b) \in \mathbb{R}^3$ ). This results in a memory efficient representation of the 3D surface as the space used to embed the surface does not need to be explicitly stored. Saito et al. [95, 96] present an example of such an architecture. Their network maps 3D locations with ResNet [Resnet] features sampled at coordinates projected with a known weak-perspective camera to occupancy and texture values. Li et al. [MonoPort] later sped up reconstruction and rendering to allow real-time inference. Neural Radiance Fields (NeRF) capture high resolution details of static scenes by mapping 3D coordinates  $(x, y, z)$  and viewing directions  $(\theta, \phi)$  to volume density and view-dependent emitted radiances. The work was adapted to articulated structures [ANerf] by learning new skin for the SMPL model, using SPIN [60] as a basic 3D skeleton predictor. A current downside of these techniques is the need to train a separate model per scene/subject and the slow inference time. However, these works do demonstrate high quality results and are a worthy direction for future research.

### 2.4.3 Model-based animals

Having discussed reconstruction methods for other articulated subjects, this section will move on work related to the main topic of this thesis: 3D reconstruction of animal subjects. As summarized above, there are multiple challenges associated with animals which are not present with humans. Firstly, the general class of animal subjects is significantly more diverse in appearance and structure than the human category. This holds (although to a lesser extent) with various important animal subcategorizations, such as the class of medium-to-large quadrupeds, dog breeds, or birds. This diversity leads to challenges when designing 3D morphable models, since extreme deformations must be allowed while penalizing even subtle adaptations deemed unnatural. Also of concern are the significant variations in animal motion patterns (causing complex self-occlusion), body textures including fur (which vary even within breeds) and difficult environmental conditions. An interesting advantageous aspect to reconstructing animals is that they are less frequently clothed; a common nuisance factor for human reconstruction. These factors combined with practical challenges associated with capturing and annotating images has also led to a lack of open-source datasets which depict animal subjects. Of particular concern to this thesis is the lack of 3D training data captured from

Dataset	Modality	Annotation	Subject	Num. Examples
MSCOCO [64]	Image (RGB)	S2, BB2	DCaHoZSECoBG	23,467
PascalVOC [29]	Image (RGB)	S2, BB2	xxx	xxx
DAVIS [58]	Video (RGB)	S2	xxx	xxx
StanfordDogs	Image (RGB)	BB2	D (120 breeds)	20,580
<b>StanfordExtra</b>	Image (RGB)	J2, S2	D (120 breeds)	12,000
<b>BADJA</b>	Video (RGB)	J2, S2	xxx	xxx
AnimalPose	Image (RGB)	J2	xxx	xxx
SMAL	N/A	MM3	DCaHoZCoTLHi	47

Table 2.1 Dataset summary: analysis of available datasets. Bold rows are datasets introduced in this thesis. S2: 2D Silhouettes, BB2: 2D bounding boxes, J2: 2D Joints, MM3: 3D Morphable Model. P3: 3D Priors. M3: 3D Model. D: dog, Ca: cat, Ho: horse, Z: zebra, S: sheep, E: elephant, Co: cow, B: bear, G: giraffe, T: tiger, L: lion, Ch: cheetah, Hi: hippo

real animal subjects (typically using motion capture or static scans). Equivalent resources made available for humans have been used as a basis for building 3D morphable models (e.g. SMPL [69], FAUST [FAUST]), for learning priors over human shape and pose [learning priors], and to provide per-image 3D supervision when training deep neural networks [53]. Thankfully, online animal imagery which could serve as the basis for future datasets is plentiful, as are datasets with 2D silhouette data [64, 29, 58]. A formal comparison is given in Table 2.1. While animals are often featured in computer vision literature, there are still relatively few works that focus on accurate 3D animal reconstruction. However, the creative approaches used in these methods makes a formal review worthwhile. A summary of recent approaches is tabulated in Table 2.2.

### Learning animal shape from 2D image collections

Early work in animals focuses on learning 3D animal shape spaces by registering an input 3D model to image collections. Examples include Chen et al.[chensharks] who learn a shark model and Cashman et al. [19] who recover a parameterized, morphable 3D model from unrelated 2D images depicting examples of the target class. Their 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 sourced from the Internet. 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 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 suffers from an overly smooth shape representation which causes trouble for strongly articulated classes such as polar bears. Related is the work of Ntouskos et al.[ntouskos] who use an optimization scheme to combine geometric primitives fit to segmented animal parts.

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 [141]	Zebras, horses	M3 (albeit synthetic), J2, S2, P3	SMAL	3-7 frames / animal	None	Yes	200
Lions, Tigers and Bears (SMALR) [142]	MLQ	Not trained	SMAL	3-7 frames / animal	J2, S2	Yes	14
3D Menagerie (SMAL) [143]	MLQ	Not trained	SMAL	No	J2, S2	Yes	48
Creatures Great and SMAL [9]	MLQ	Not trained	SMAL	Yes	S2 (for best results shown)	Yes	9
Category Specific Mesh Reconstructions [55]	Birds	J2, S2	Bird convex hull	No	None	No	2850
What Shape are Dolphins [19]	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 2.2 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.

### Reconstructing animals with unknown skeleton

The approach was later extended to articulated classes in work by Stebbing et al. [110], who optimize a 3D template model to animal video sequences. In this work, rather than defining an internal skeleton, the template model is rigged with virtual markers, which assign each mesh vertex  $v_i$  to one of  $M$  groups that share a set of basis rotations  $B_m$ . User interaction is again required to segment the animal from the background and to provide mesh-to-image keypoint correspondences.

Through reasonably accurate pose fitting and by allowing some pose-invariant shape deformation, this work produces smooth meshes which match the input video. Moreover, experimentation demonstrates 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 are shown on 11 sequences.

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. Similar is the technique of Favreau et al. [favreau], who fit geometric primitives to a 2D video sequence with manually sketched body part labels.

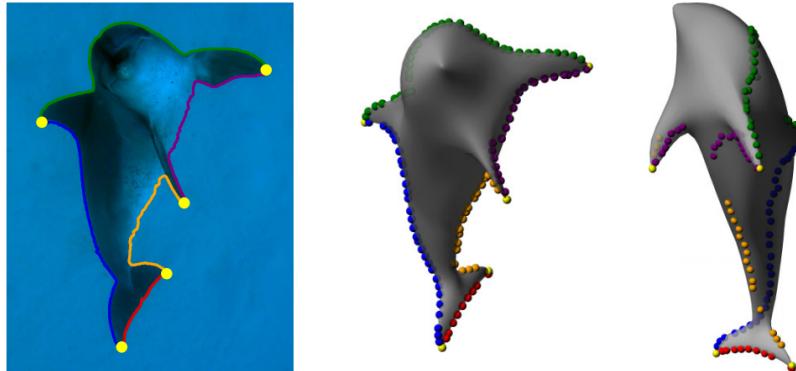
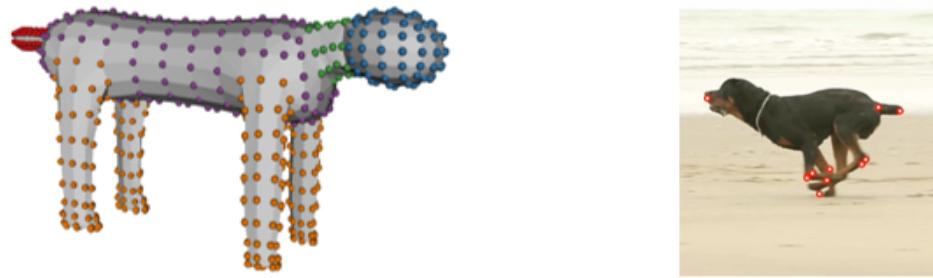


Fig. 2.7 8-parameter dolphin model with annotated contour (left) and contour generators (middle and right).



(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 [110].

### Fitting a morphable animal mesh to images

Unfortunately, none of the techniques above are suitable for reconstructing an animal category with significant shape diversity. Zuffi et al. [143] made a significant contribution by releasing SMAL, a deformable 3D quadruped model (analogous to SMPL [69] for human reconstruction) build from 41 scans of artist-designed toy figurines. The authors also released shape and pose priors generated from artist data. The authors also discuss a small modification to the SMPLify [smplify] approach in order to fit the SMAL model to RGB animal input images. 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. The whole optimization process takes around 1 minute per frame.

The terms of the optimization largely mirror the work of SMALify, although the interpenetration term is omitted and joints are provided manually rather than by a CNN. Importantly, the optimization incorporates an additional *silhouette* term which aligns the 3D model shape to a binary silhouette image extracted by a user. Chapter 3 of this thesis shows silhouette terms are of particular importance when reconstructing accurate animal shape. The following definition describes the process for differentiably rendering a 3D model to form a silhouette image which can then be compared to the provided 2D silhouette:



Fig. 2.9 Example of an dog template being fit to input video sequence, reprinted from [110]

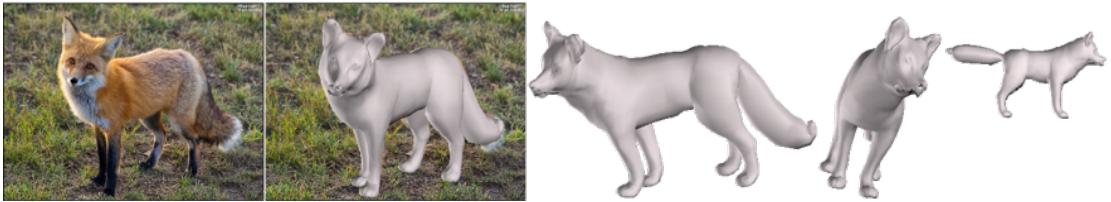


Fig. 2.10 Fitting SMAL to a hand segmented animal, reprinted from [143].

**Definition 2.4.2** (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 computation 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).

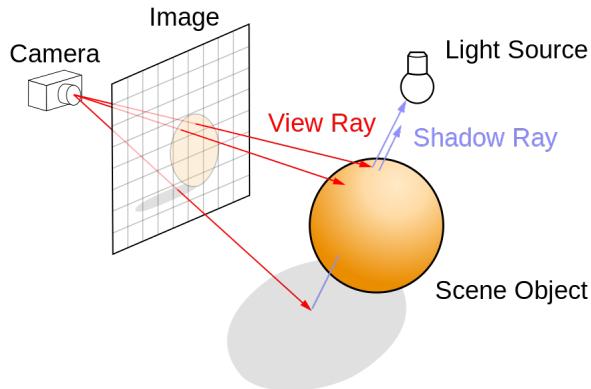


Fig. 2.11 Diagram showing raycast rendering. [130].

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 [70] are available for these purposes.

The SMAL authors [143] demonstrate their system by fitting their deformable 3D model to quadruped species. Subsequent work includes SMALR [142], who fit to multiple images of an animal and incorporate similar freeform deformation term used to model clothing in 3D human pipelines. They show their technique, which takes advantage of multi-view constraints afforded by fitting to video sequences, allows high-quality reconstructions of more exotic animal categories with details not covered in the original SMAL 3D model training set.

A common property of techniques discussed so far is the reliance on manual annotations (commonly 2D silhouettes and keypoint correspondences) at test time. 3D-Safari [141] overcome this by training a deep network on synthetic data (generated by applying SMALR [142] on a large input dataset) to recover detailed zebra shapes in the wild. Chapter 3 of this thesis demonstrates a technique which overcomes the need for a large input dataset of joint annotations by training on synthetic data generated from the graphics model alone.

A further drawback of the 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 4 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.

Tangential to these approaches is the work of Kulkarni et al. [Canonical Surface Mapping] who reconstruct pose (although not shape) parameters for a rigged mesh using a deep network, learning from geometry cycle consistency terms rather than annotated keypoints. However, the shape is fixed to the input mesh and pose is also often confused, particularly determining the difference between the legs. The results appear of significantly poorer quality than the results in Chapter 3 and Chapter 4 of this thesis. Also of consideration is the end-to-end network of Kanazawa et al. [55] and subsequent paper which overcomes the need for keypoints by capturing uncertainty in camera viewpoints. In

addition, their network does not require an input 3D morphable model, although they do initialize a sphere model to a mean bird. In addition, the bird category exhibits more limited articulation than our dog category.

# Chapter 3

## Learning from Synthetic Data; Bridging the Domain Gap

### 3.1 Introduction

This chapter introduces a system for recovering the 3D shape and motion of a wide variety of quadrupedal animals from video. Inspired by techniques from the recent human body and hand tracking literature (most notably SMPLify [13]), the system comprises a machine learning front-end for predicting 2D joint positions followed by an energy minimization stage which fits a detailed 3D model to the image. However, this chapter will discuss a number of key challenges which arise when reconstructing animal subjects compared with humans and methods to overcome them.

#### 3.1.1 Limited animal training data

For human tracking, hand labelled sequences of 2D segmentations and joint positions have been collected from a wide variety of sources [4, 64, 49]. Of these two classes of labelling, animal *segmentation* data is available in datasets such as MSCOCO [64], PASCAL VOC [29] and DAVIS [85]. 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. In the case of SMPLify, the network is trained on large-scale human keypoint datasets, including MPII [4] (approx. 40,000 people) and the Leeds Sport Dataset [49]. Unfortunately, there is no keypoint dataset for animals that cover even a small fraction of the quadruped types that we aim to reconstruct.

Of greater 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

Paper	Release Year	Animal Class	Training requirements	Template Model	Video required	Test Time Annotation	Model Fitting	Test Size
What Shape are Dolphins [19]	2013	Dolphins, Pigeons	Not trained	Dolphin Template	25 frames / category	J2, S2	Yes	25
Animated 3D Creatures [? ]	2016	MLQ	Not trained	Generalized Cylinders	Yes	J2, S2	Yes	15
3D Menagerie (SMAL) [143]	2017	MLQ	Not trained	SMAL	No	J2, S2	Yes	48
Category Specific Mesh Reconstructions [55]	2018	Birds	J2, S2	Bird convex hull	No	None	No	2850
Lions, Tigers and Bears (SMALR) [142]	2018	MLQ	Not trained	SMAL	3-7 frames / animal	J2, S2	Yes	14
This paper	2018	MLQ	Not trained	SMAL	Yes	S2 (for best results shown)	Yes	9
Who Left the Dogs Out?	2018	Dogs	J2, S2, T3, P3	SMAL	No	None	No	1703
3D-Safari [141]	2019	Zebras, horses	M3 (albeit synthetic), J2, S2, P3	SMAL	3-7 frames / animal	None	Yes	200
U-CMR [141]	2020	Birds	M3 (albeit synthetic), J2, S2, P3	Bird convex hull	3-7 frames / animal	None	Yes	200

Table 3.1 Literature summary: Analysis of required input annotations. MLQ: Medium-to-large quadrupeds. J2: 2D Joints. S2: 2D Silhouettes. T3: 3D Template. P3: 3D Priors. M3: 3D Model.

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 [131], it remains impractical to consider mocap for a family of tigers at play.

These dataset challenges have, at least in part, resulted in the lack of *automatic* approaches for 3D animal reconstruction. Precisely, prior to the design of the system discussed in this chapter, 3D animal reconstruction techniques relied on at least one of manually provided silhouettes, keypoint locations or limb tracks. The techniques are summarized in Table 3.1 and discussed in depth in Chapter 2.

### 3.1.2 Synthetic image generation

In the absence of real-world motion capture data, the recent publication of the Skinned Multi-Animal Linear (SMAL) model [143] offers a method for *synthetic* data generation. Analogous in design to the popular SMPL [69] parametric human model, SMAL can generate a wide range of quadruped species. As discussed in Chapter 2, SMAL overcomes the lack of 3D animal training data by instead learning from 41 scans of toy figurines. This results in SMAL being of considerably lower fidelity than the SMPL, which was constructed from 1786 3D human scans. However, the model’s PCA shape space offers a mechanism for sampling potentially infinite quadruped animals of satisfactory realism which can form a valuable training dataset.

The use of synthetic images to train 3D reconstruction systems has been attempted in recent computer vision literature. The general approach is to capture training images by capturing multiple images of a representative 3D model (or scene) by randomly sampling camera and environmental parameters. A machine learning model is then trained on this dataset  $D_{synth}$  of synthetic images and later tested on a dataset  $D_{real}$  of real-world examples. This procedure offers an important practical advantage; the training images can be automatically generated with corresponding annotations. Depending on the task, this can overcome a lengthy and complex process of sourcing manual annotations from users. However, designing a data pipeline which can generate synthetic images with sufficient realism to be representative of real test images  $D_{real}$  is challenging. A common pitfall of such approaches is the tendency to train machine learning models that perform well on unseen synthetic images but poorly on real world examples. This phenomenon is often caused by systematic differences between the  $D_{synth}$  and  $D_{real}$ , resulting in the model becoming overreliant on features present only in  $D_{synth}$  or unfamiliar with features present only in  $D_{real}$ . The differences between datasets is often described as the *domain gap* which must be bridged in order to achieve a successful predictor.

Fortunately, there are a number of options for tackling this problem. A popular strategy is to ensure the generation pipeline is of extremely high quality. SURREAL[SURREAL] is a method which learns from synthetic human images rendered using the popular SMPL [69] parametric model. In this work, the authors go to significant effort to source realistic UV texture maps for SMPL and apply a high quality rendering engine complete with a lighting and reflectance model to generate their training images. A similar approach is followed by SMALST [SMALST], who follow a multi-view optimization pipeline to construct a synthetic zebra training dataset complete with realistic textures. Related also are autonomous driving systems that train (or pre-train) on synthetic ‘virtual worlds’ [Virtual Worlds]. An alternative option is to process the test image dataset to find a representation which is easier to synthesise. For 3D human skeleton prediction, Shotton et al. [101] synthesise a set of depth images used to train a keypoint regressor. An unfortunate byproduct of this training style however is that it necessitates a depth sensor at test time.

A significant advance has been made recently with the introduction of generative adversarial networks (GANs). The idea of a GAN is to train two ‘competing’ networks: the *generator* is tasked with producing synthetic images and the *discriminator* determines if a given input image originated in a real-world dataset or was instead synthesized by the generator. These two networks are given opposite goals: the generator is penalized if it fails to ‘fool’ the discriminator, and the discriminator is penalized if it fails to recognize a synthesized image. By training these networks jointly, the generator eventually learns to produce synthetic images of extremely high quality, since the discriminator provides feedback on even subtle aspects of the synthetic images which are unrealistic. Despite the remarkable quality of ‘GANerated’ images, the challenge of conditioning GANs to generate *structured* images is still an evolving field. For example, a sensible challenge might be to train a GAN to generate UV texture maps for the SMAL model.

of this is that there should be an analoug

The system presented in this section adopts an alternative (and much simpler) approach for synthetic image generation. The idea builds on the realisation that a 2D silhouette image encodes a significant amount of the object’s shape features, and by virtue of being textureless and having no background detail is much simpler to synthesise. As previously discussed, there are also a great number of ‘off-the-shelf’ image segmentation engines which can produce accurate animal silhouettes of a high quality.

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.

### 3.1.3 Handling silhouette ambiguity

An unfortunate drawback of using silhouette images (for example, when compared to RGB or depth counterparts) is that missing interior contours leads to a source of reconstruction ambiguity. An example of this is shown in Figure 6 in which two distinct 3D reconstructions are shown to yield similar 2D silhouette reprojections. Clearly, a naive joint predictor which regresses a single set of 2D coordinates from an input silhouette would perform suboptimally, since the silhouette alone contains insufficient information to resolve such cases.

However, the fact that the system is designed to accept a video sequence as input (rather than a single frame) provides additional information which can be used to resolve per-frame ambiguities. This section describes a method for incorporating temporal knowledge in order to obtain a reliable set of predicted 2D joint coordinates that later form the basis for the 3D model fitting stage. The key insight is to modify a standard joint heatmap predictor [hourglassnet] in order to encourage multi-modal outputs, both when trained on synthetic data and later tested on real world frames. A novel discrete optimization stage is then used to recover the skeleton trajectory of maximal likelihood from the sequence of multi-modal predictions. This section will explore two methods for achieving this: the first frames the problem as a quadratic program and the latter achieves a significant efficiency improvement by using a genetic algorithm.

### 3.1.4 Contributions

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. The system exhibits the following contributions:

- A machine-learned mapping from silhouette data of a large class of quadrupeds to generic 2D joint positions.



**Fig. 3.1 System overview:** input video (a) is automatically processed using DeepLabv3+ [22] 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).

- A novel optimal joint assignment (OJA) algorithm extending the bipartite matching of Cao *et al.* [18] 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 sequences. Previous work in 3D animal reconstruction has relied on bespoke hand-clicked keypoints [143, 142] 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 provides 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. Experiments are also conducted on synthetic animal videos to produce 3D joint accuracy statistics and full mesh comparisons. A qualitative comparison is given to recent work [143] on the related single-frame 3D shape and pose recovery problem. The paper concludes with an assessment of strengths and limitations of the work.

## 3.2 Preliminaries

### 3.2.1 Deformable 3D quadruped model

This section provides a formal definition for the deformable 3D model that is used to generate synthetic training data and in the model fitting stage to obtain the final mesh. Our system assumes a deformable 3D model such as SMAL [143] which parametrizes a 3D mesh as a function of *pose* parameters  $\theta \in \mathbb{R}^P$  (e.g. joint angles) and *shape* parameters  $\beta \in \mathbb{R}^B$ . As discussed in Chapter 2, 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 SMAL may be viewed as supplying a set of triangles, and a function

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

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 a mesh will 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  $\phi$  which may be for example translation as a 3-vector and rotation in axis angle form. Application of such a transformation to a  $3 \times V$  matrix will be denoted by  $*$ , so that

$$\phi * v(\theta, \beta) \quad (3.2)$$

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

It is also necessary to define a model’s *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. Note that even in the case of rigged models, this provides a mechanism to add additional joints beyond the ones required to drive model deformation. In any case, joints are defined by post-multiplying by a  $V \times J$  matrix  $K$ . The  $j^{\text{th}}$  column of  $K$  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).

### 3.2.2 Camera model, joint reprojection and silhouette rendering

For both synthetic image generation and the later model fitting stage, it is necessary to be able to *render* the 3D model. 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

$$\kappa(\phi, \theta, \beta) := \pi(\phi * v(\theta, \beta) K) \quad (3.3)$$

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

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,

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

for an image resolution of  $W \times H$ . In order to allow derivatives to be propagated through  $R$  (essential for the silhouette term in the model fitting stage), the *differentiable renderer* of Loper et al. [70] is used. Please see the relevant section in Chapter 2 for further details.

### 3.2.3 System overview

The test-time problem to be solved is to take a sequence of input images

$$\mathcal{I} = [I_t]_{t=1}^T$$

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

$$\mathcal{S} = [S_t]_{t=1}^T.$$

The computational task is to output for each input image the shape, pose, and position parameters describing the animal’s motion. Inspired by recent work in human 3D reconstruction, this objective can be broken down into a multiple stage pipeline.

The core components are as follows:

1. The discriminative front-end extracts silhouettes from video, and then uses the silhouettes to predict multi-modal heatmaps, from which 2D joint positions are obtained 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.
3. Generative deformable 3D model is fitted to the silhouettes and joint candidates as an energy minimization process.

These stages are described in detail over the following three sections.

### 3.3 Predicting 2D joint candidate locations

The goal of the first stage is to take, for each video frame, an image representing the animal and to output a  $W \times H \times J$  tensor of heatmaps. To achieve this, we train the stacked hourglass network [79] of Newell et al. to a large dataset of synthetically generated quadruped images.

#### 3.3.1 Generating synthetic quadruped images

In order to render a synthetic quadruped image, a set of pose  $\theta$ , shape  $\beta$  and position  $\phi$  parameters are required. With these parameters, a (textureless) training image can be generated by applying Equation (3.4) and corresponding ground truth 2D joints locations can be generated via Equation (3.3). What remains is to ensure a model trained on the synthetic images generalizes to the real-world test images. To achieve this, it is important that the training dataset captures the modes of variation by appropriately sampling the model parameters.

##### Sampling shape and pose parameters

Given that the real-world test images exhibit multiple quadruped species, a primary mode of variation is the animal's *shape* characteristics. Naively, generating a dataset large enough to capture possible test animal shapes would be a challenging task, since it would require access to multiple real or artist-generated 3D scans. Fortunately, the SMAL model defines a linear shape space which allows repeated sampling of different (and realistic) quadruped shapes. Of course, having been built from only toy figurines, the variation is still somewhat limited, but as the later experimental section will show, the model is fit for purpose. Synthetic shape parameters  $\beta$  are obtained by sampling from a Gaussian prior:

$$X = Y \tag{3.5}$$

Another mode of variation in test images is the various animal limb positions. These can again be synthesised by sampling from a gaussian prior constructed from a dataset of animal poses. Since no such dataset exists for real animals, a prior is instead built from a small set of artist-generated poses, originally provided by the SMAL model authors. The sampling strategy is therefore given by:

$$X = y \tag{3.6}$$

##### Sampling position (camera) parameters

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 textures and backgrounds?

At this point, the synthetic test images have plausible shape, pose and positional parameters but are lacking in realistic texture. Furthermore, it is not clear how to render plausible background scenes, nor how to convincingly place the 3D animal in such a scene. This challenge allows for a primary contribution of this work: using the silhouette domain as a representation much easier to synthesise (since surface texture maps and background textures are lost) and can be obtained from real-world test images. Another important characteristic of the mapping between real-world images and silhouette counterparts is that the shape and pose information is mostly preserved, apart from a few ambiguities (e.g. limb ordering due to lacking interior contours).

To summarise, training data comprises  $(S, \kappa)$  pairs, that is pairs of binary silhouette images, and the corresponding 2D joint locations as a  $2 \times J$  matrix. See an example in To generate each image, a random shape vector  $\beta$ , pose parameters  $\theta$  and camera position  $\phi$  are drawn, and used to render a silhouette  $R(\phi * v(\theta, \beta))$  and 2D joint locations  $\kappa(\phi, \theta, \beta)$ .

#### 3.3.2 Prediction of 2D joint locations using multimodal heatmaps

A pose estimation network can now be trained on a dataset of binary silhouette images  $S$  with corresponding 2D joint locations  $\kappa(\phi, \theta, \beta)$ . The core network architecture used for this task is the stacked hourglass network [79] of Newell et al., with an adaptation to produce multi-modal outputs. The stacked hourglass network is now briefly described:

##### Stacked hourglass network

A stacked hourglass network is a convolutional neural network specifically designed for the task of pose estimation. Hourglass allows inference to take place across multiple scales; an important advantage allowing the network to reason about global information (such as the full body) and local information (such as face features). This is achieved using a repeated bottom-up, top-down modules which each produce a  $W \times H \times J$  heatmap tensor. Supervision is applied to each of these heatmaps, which are then passed to the subsequent block, allowing high level features to be reevaluated for higher order spatial relationships (generally only present at low resolutions). The network’s architecture can be seen in detail in Figure XXX. Hourglass uses a mean squared error loss against a ground truth heatmap tensor, which is applied to the network’s final output and all intermediary heatmaps. Therefore, ground truth joint coordinates  $\kappa(\phi, \theta, \beta)$  must first be encoded into a  $W \times H \times J$  tensor of heatmaps. The original authors find that the network has better convergence properties if the ground truth heatmaps are blurred slightly, since a gradient is then provided to predicted “near misses”. This is achieved by blurring ground truth heatmaps with a Gaussian kernel of radius  $\sigma$ . The final 2D joint positions are then obtained using non-maximum suppression on the output heatmaps

$$A = B \quad (3.7)$$

### Adaptations for training on synthetic data

This setup is mostly suitable for training on synthetic quadruped data, subject to a couple of adaptations. Firstly, the silhouette training images differ considerably in appearance to the RGB images used by the original Hourglass authors. A common difficulty for training pose estimation networks on full RGB images is in trying to distinguish between objects and the background class. With binary silhouette images, this distinction is made trivial as background and foreground are assigned values 0 or 1 respectively. Unfortunately, this causes a problem when naively training HourglassNet on the generated synthetic images as the network can greatly minimize the loss by simply predicting ‘background’ for every image pixel (since background pixels tend to outnumber foreground pixels). The fact that ‘background’ is a much simpler class to predict than the other joint classes causes a training instability which is challenging to overcome by adjusting the learning rate. Instead, the loss function is replaced with a *weighted* version of the mean squared error loss. This modification encourages the network to devote its attention to all classes equally. The precise formulation for this is given as follows:

$$A = B \quad (3.8)$$

### Adapting stacked hourglass network for multi-output learning

As detailed in the later experimental section, the network trained using this process generalizes well from synthetic to real images due to the use of the silhouette, and produces accurate predictions for most joints. However, the predictor performs poorly on some joints due to ambiguities which result from the lack of interior contours in the silhouette input data. These missing details cause often result in confusion between joint “aliases”: left and right or front and back legs. When these predictions are wrong and are represented by high confidence heatmap regions, little probability mass is assigned to the area around the correct leg, meaning no available proposal is present after non-maximal suppression.

This is handled with a further technical contribution; handling the prediction uncertainty by adapting the stacked hourglass network to produce multiple outputs. This is achieved by explicitly training the network to assign some probability mass to the “aliased” joints. For each joint, a list of potential aliases are defined as weights  $\lambda_{j,j'}$  and linearly blend the unimodal heatmaps  $G$  to give the final training heatmap  $H$ :

$$H_j(p) = \sum_{j'} \lambda_{j,j'} G(p; \kappa_{j'}, \sigma) \quad (3.9)$$

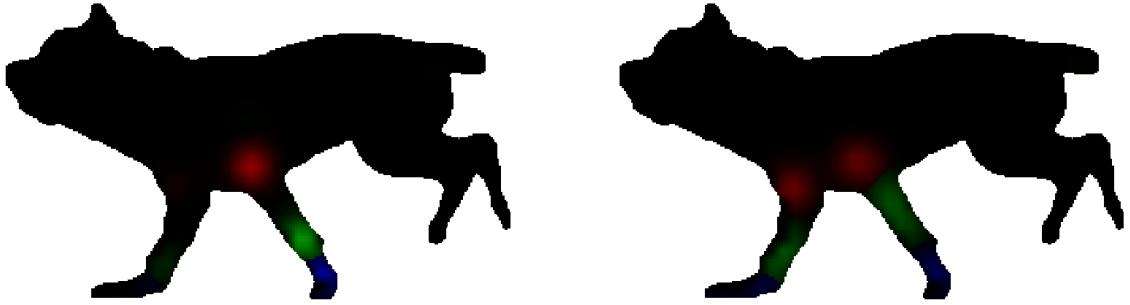


Fig. 3.2 Example predictions from a network trained on unimodal (top) and multi-modal (bottom) ground-truth for front-left leg joints.



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

For non-aliased joints  $j$  (all but the legs),  $\lambda_{j,j} = 1$  and  $\lambda_{j,j'} = 0$ , yielding the unimodal maps. For aliased joints, the joint is assigned a weight  $\lambda_{j,j} = 0.75$  and aliases are assigned a weight  $\lambda_{j,j'} = 0.25$ . This ratio is shown to ensure opposite legs have sufficient 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. Note that the construction of at most bi-modal ground truth heatmaps sets a practical constraint on the number of output modes. In other words, the loss is minimized if the network produces one output mode for non-aliased joints and two output modes for aliased joints.

### 3.4 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. The set of joint proposals is represented as  $X = \{x_{jp}\}$ , where  $x_{jp}$  indicates the 2D position of proposal  $p \in \{1, \dots, N_j\}$  associated with joint  $j \in J$ . Before applying the optimizer, a subset of proposals  $X^* \subseteq X$  should be selected in order to form a complete skeleton, i.e. precisely *one proposal is selected for every joint*. This section will 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  $\{x_{jp}\}$ , a binary indicator vector  $\bar{a}_j = \{a_{jp}\} \in \{0, 1\}^{N_j+1}$  is introduced, where  $a_{jp} = 1$  indicates that the  $p^{\text{th}}$  proposal

Joint $j$	Proposal $p$	$x_{jp}$	$a_{jp}$
Nose	0	(0, 2)	1
	1	(4, 2)	0
	NULL	—	0
Upper Leg	NULL	—	1
Paw	0	(4, 2)	0
	1	(8, 10)	1
	NULL	—	0
Tail	0	(4, 2)	0
	1	(8, 10)	0
	2	(4, 2)	1
	NULL	—	0

Table 3.2 Example inputs and output to the joint assignment problem. Non maximum suppression applied to predicted heatmap tensors yields a set of proposals  $p$  for each each skeleton joint  $j$  at 2D location  $x_{jp}$ .  $a_{jp}$  is an illustrative assignment vector which is predicted by the OJA algorithm.

Joint $j$	Proposal $p$
Nose	1 0 0
Upper Leg	1
Paw	0 1 0
Tail	0 0 1 0

Table 3.3 Assignment variables  $A = \bar{a}_j = \{a_{jp}\} \in \{0, 1\}^{N_j+1}$  for the current frame stored as a jagged array.

Joint $j$	$X^* = X(A)$
Nose	(0, 2)
Upper Leg	—
Paw	(8, 10)
Tail	(4, 2)

Table 3.4 2D joint locations  $X^* = X(A)$  selected by the assignment variable  $A$

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

$$L(A) = L_{\text{conf}}(A) + L_{\text{null}}(A) + L_{\text{prior}}(A) + L_{\text{temp}}(A) + L_{\text{cov-sil}}(A) + L_{\text{cov-bone}}(A) \quad (3.10)$$

which balances agreement of the joint configuration with the network-supplied *confidences*, a learned *prior*, *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 better results are obtained by including 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.

### 3.4.1 Basic formulation

**Network confidences:**  $L_{\text{conf}}(A)$

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

network confidences compared with that of the skeleton prior. Note that if only this term is included, the OJA would simply produce the result of the standard non-maximal suppression algorithm, selecting the heatmap location with highest network confidence. Note that this function can be rewritten as

$$L_{\text{conf}}(A) = \lambda_{\text{conf}} \log(y^T) \text{vec}(A) \quad (3.11)$$

### Null proposals: $L_{\text{null}}(A)$

Under some circumstances, for example when a body part is heavily occluded or ambiguous, *all* available proposals for a given joint may be of poor quality. In such a case, including any of these options in a skeleton configuration may have a detrimental impact on the later model fitting stage. Under these circumstances, it may be preferable to exclude the joint in question from the optimization entirely. Null proposals pay a fixed cost  $\lambda_{\text{null}}$ , effectively acting as a threshold whereby the null proposal will be selected if no other proposal is of sufficient likelihood. Precisely, a jagged array  $D$  is defined

$$D_{jp} = \begin{cases} \lambda_{\text{null}} & \text{if } p \text{ is null proposal} \\ 0 & \text{otherwise} \end{cases} \quad (3.12)$$

and resulting energy becomes

$$L_{\text{null}}(A) = \text{vec}(D) \text{vec}(A) \quad (3.13)$$

### Skeleton Prior: $L_{\text{prior}}(A)$

The next energy term is used to discourage anatomically implausible skeletal configurations from being selected. The prior probability of a skeletal assignment  $A$  is represented as a multivariate Gaussian distribution over the selected joint positions  $X^* = X(A)$

$$P_{\text{prior}}(A) = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} \exp\left(-\frac{1}{2}(x^* - \mu)^T \Sigma^{-1} (x^* - \mu)\right) \quad (3.14)$$

The mean  $\mu \in \mathbb{R}^{2J}$  and covariance  $\Sigma \in \mathbb{R}^{2J \times 2J}$  terms are obtained from synthetic training examples generated earlier. The prior is shown in Figure 3.4. The objective of the OJA is to find the assignment vector  $A$  which maximizes this prior, which is equivalent to minimizing the negative log prior

$$L_{\text{prior}}(A) = \frac{k}{2} \log(2\pi) + \frac{1}{2} |\Sigma| + \frac{1}{2} (x^* - \mu)^T \Sigma^{-1} (x^* - \mu) \quad (3.15)$$

This formulation is reducible to a minimization over the Mahalanobis distance, which is given by the summation

$$L_{\text{prior}}(A) = \sum_j^J \sum_p^{N_j} \sum_k^J \sum_q^{N_k} a_{jp} a_{kq} (x_{jp} - \mu_j) \Sigma_{jk}^{-1} (x_{kq} - \mu_k) \quad (3.16)$$

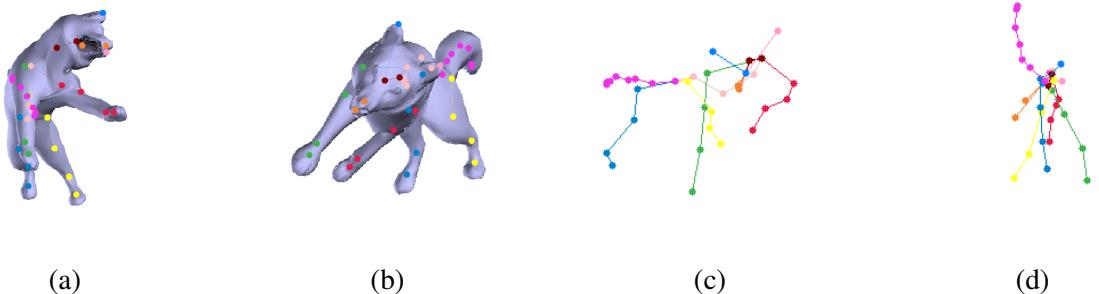


Fig. 3.4 Skeleton Prior: Synthetic quadruped training data examples (rendered with texture to show 3D) generated by sampling pose, shape and position parameters and applying to SMAL model (a), (b). The set of 2D skeletal positions are used to create a vector of means  $\mu \in \mathbb{R}^{2J}$  and covariance matrix  $\Sigma \in \mathbb{R}^{2J \times 2J}$ . The Gaussian distribution constructed can be sampled to create new skeletons, such as those shown in (c), (d).

Notice this is a quadratic function of  $A$ , so  $L_{\text{prior}}(A) = \text{vec}(A)^\top Q \text{vec}(A)$  for a fixed matrix  $Q$ . Precisely, the elements of the matrix  $Q$  are precisely the Mahalanobis distance between each individual pair of joint proposals

$$[Q]_{jp,kq} = (x_{jp} - \mu_j) \Sigma_{jk}^{-1} (x_{kq} - \mu_k) \quad (3.17)$$

Null proposals are simply excluded from the sum, equivalent to marginalizing over their position.

### Temporal Prior: $L_{\text{temp}}(A)$

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. A prior is imposed 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  $\tau$  controls the strength of the interaction between distant frames. This results in an additional quadratic term in our objective function, which has the form  $L_{\text{temp}} = a^\top T^{(t_0,t_1)} a$  for matrix  $T^{(t_0,t_1)}$  given by

$$\left[ T^{(t_0,t_1)} \right]_{jp,kq} = \begin{cases} e^{-\alpha|t_1-t_0-1|} \|x_{jp}^{(t_0)} - x_{kq}^{(t_1)}\|^2 & \text{if } j = k \\ 0 & \text{otherwise} \end{cases} \quad (3.18)$$

#### 3.4.2 QP solution.

A general quadratic program is made up of a quadratic objective function and linear equality and inequality constraints. Thus far, all terms in  $L(A)$  are quadratic or linear.

To optimize over a sequence of frames, we construct the block diagonal matrix  $\hat{Q}$  whose diagonal elements are the prior matrices  $Q^{(t)}$  and off-diagonal elements are the temporal matrices  $T^{(t_0, t_1)}$ . The confidence term Equation (3.11) and null penalty term Equation (3.13) are combined and the vector  $\hat{c}$  is obtained by stacking across each frame. The solution vector for the sequence  $\hat{a}$  is similarly constructed by stacking the vectorized assignment matrices  $A$  across timesteps. The jagged array  $B$  is used to formalize the constraint that only one proposal should be selected per joint. Precisely

$$[B]_{jp,kq} = \begin{cases} 1 & \text{if } j = k \\ 0 & \text{otherwise} \end{cases} \quad (3.19)$$

and (similarly to  $A$ ) is vectorized and stacked across timesteps to form the constraint  $\hat{B}\hat{a} = 1$ . Finally, the constraint  $\hat{a}(1 - \hat{a}) = 0$  is applied to ensure binary values of  $\hat{a}$ . The resulting quadratic program formulation is then given in Equation (3.20).

$$\begin{aligned} \underset{\hat{a}}{\text{minimize}} \quad & \hat{a}^T \hat{Q} \hat{a} + \hat{c}^T \hat{a} \\ \text{subject to} \quad & \hat{B} \hat{a} = 1, \\ & \hat{a}(1 - \hat{a}) = 0 \end{aligned} \quad (3.20)$$

The quadratic program is specified using the open source CVXPY library [27] and solved using the “*Suggest-and-Improve*” framework proposed by Park and Boyd [81]. It is initialized by choosing the proposal with the highest confidence for each joint. Appropriate values for the free parameters  $\lambda_{\text{conf,temp,null}}$  and  $\alpha$  were chosen empirically via grid search.

### 3.4.3 Incorporating coverage priors

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.

#### Silhouette coverage: $L_{\text{cov-sil}}$

The silhouette coverage term is designed to penalize 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)[12] of the silhouette,  $Z' = \text{MAT}(S')$  with a cubed loss strongly penalizing projection outside the silhouette:

$$L_{\text{cov-sil}}(A^t; X^t, Z^t) = \sum_i \min_j \|Z_i^t - \hat{X}_j^t\|^3 \quad (3.21)$$

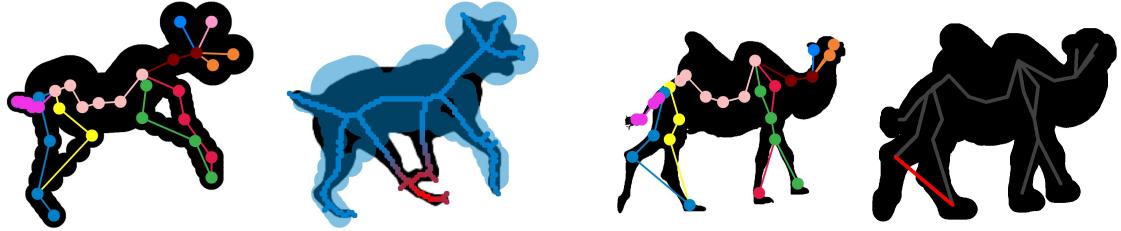


Fig. 3.5 Silhouette coverage loss. The error (shown in red) is the distance between the median axis transform (right) and the nearest point on an approximate rendering (left).

Fig. 3.6 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).

#### Bone coverage: $L_{\text{cov-bone}}$

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

$$L_{\text{cov-bone}}(A^t; X^t, S^t, K) = \sum_{\{j, k\} \in K} \left( 1 - \min_{\lambda \in [0:0.1:1]} S^t(\hat{X}_j^t + \lambda(\hat{X}_j^t - \hat{X}_k^t)) \right) \quad (3.22)$$

#### 3.4.4 Formulation as a genetic algorithm

We minimize this more complex objective using a genetic algorithm (GA)[44]. A genetic algorithm is a method for solving optimization problems using a natural selection process that mimics biological evolution. Unlike the quadratic program described previously, the GA optimization procedure relies on crossover and mutation, rather than by calculating derivatives. In some cases (and as shown in this work), this can lead to much faster convergence properties. The following components are required:

##### Genes.

Candidate solutions to the optimization problem are referred to as a set of ‘genes’. In this case, a gene is the assignment array  $A$  although is represented as a vector of  $J$  integers (indicating the proposal ID), rather than one-hot encodings.

##### Initial population.

Genes are constructed subject to the constraints given in Equation (3.20). The genetic algorithm is initialized with a population size of 128 genes. Of these, the first 32 are set equal to the max confidence solutions given by the network, which speeds up convergence. The remaining 96 genes are generated by selecting a random proposal for each joint.

### **Fitness function.**

A fitness function is used to evaluate the quality of a gene. In this setting, the fitness function is precisely the energy  $L(A)$  given above which minimizes to yield an optimal skeleton configuration.

### **Crossover.**

Crossover is a fundamental operation common to evolutionary methods which defines the process of combining the information of two ‘parent’ genes to generate an offspring gene. *Single-point* crossover operates by slicing parent genes each into two parts, and combining first and second parts from different parents to yield the next generation. Following standard practice, the crossover point is randomly selected for each combination.

### **Mutation.**

Analogous to biological mutation and used to maintain diversity among genes, each gene is assigned some probability of undergoing a *mutation*. If a gene is selected for mutation, between 1 and 4 joints have new proposals randomly assigned.

The genetic algorithm described has weights set empirically and is run for 1000 generations. Examples of errors corrected by the additional coverage energy terms are shown in Fig. 3.5 and Fig. 3.6.

## **3.5 3D model fitting**

The model optimization stage refines model parameters to match the silhouette sequence  $\mathcal{S}$ . The procedure defined here is inspired by the human reconstruction method of SMALify [13] (defined in detail in Chapter 2) and the non-automatic quadruped fitting method presented in 3D Menagerie (3DM) [143]. The technique presented in this section can be viewed as an extension of these approaches to input video sequences.

A naive 3D model fitting implementation would be to simple apply 3DM independently to each of the  $N$  video frame to yield a set of pose parameters  $[\theta_t]_{t=1}^N$ , shape parameters  $[\beta_t]_{t=1}^N$  and position parameters  $[\phi_t]_{t=1}^N$ . However, fitting to a video sequence rather than single frames offers additional opportunities to constrain the challenging monocular 3D reconstruction task. For example, video sequences typically offer multiple views of the animal subject, although the animal’s limb positions often change between frames. However, the animal’s *shape* characteristics (such as height, body proportions etc.) can be relied upon to remain largely consistent between frames. This fact is exploited through an extension that learns a single set of *global* shape parameters  $\beta$  and is assigned to across all frames  $\beta_t := \beta$ . Another benefit offered by video is the opportunity to constrain inter-frame subject motion. Assuming a reasonable framerate, it is expected that the animal’s pose and position

parameters should vary only slightly between successive frames. This intuition is characterized in Section 3.5.

The following section defines the 4 energy terms used in the optimization:

### Silhouette energy.

The silhouette energy  $E_{\text{sil}}$  compares the 3D animal model to the silhouette image according to the L2 distance between the OpenDR rendered binary image and the input silhouette  $S_t$ :

$$E_{\text{sil}}(\phi_t, \theta_t, \beta; S_t) = \|S_t - R(\phi_t * v(\theta_t, \beta))\| \quad (3.23)$$

### Unimodal Prior energy.

The prior term  $E_{\text{prior}}$  encourages the regressed shape and pose parameters to remain close to 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.

$$E_{\text{lim}}(\theta_t) = \max\{\theta_t - \theta_{\max}, 0\} + \max\{\theta_{\min} - \theta_t, 0\}. \quad (3.24)$$

### Joints energy.

The joints energy  $E_{\text{joints}}$  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.

$$E_{\text{joints}}(\phi_t, \theta_t, \beta; X^*) = \|X^* - \phi_t * v(\theta_t, \beta) K_t(:, j)\| \quad (3.25)$$

### 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:

$$E_{\text{temp}}(\phi_t, \theta_t) = (\phi_t - \phi_{t+1})^2 + (\theta_t - \theta_{t+1})^2 \quad (3.26)$$

The optimization is via a second order dogleg method [71].

## 3.6 Experiments

This section details both quantitative and qualitative evaluation of the system described in the previous sections. Due to the lack of animal keypoint datasets in the public domain, it is necessary to construct a new Benchmark Animal Dataset of Joint Annotations (BADJA). BADJA is a dataset comprising several video sequences with hand-clicked 2D joint labels and segmentation masks. Qualitative results are provided on BADJA and also on images used in the evaluation of 3D Menagerie (3DM). However, it should be noted this comparison is not entirely fair, since 3DM requires hand-clicked keypoints as input and system introduced in this chapter is optimized for video input.

### 3.6.1 BADJA Dataset

The BADJA dataset contains 9 videos, with manually collected keypoint and silhouette annotations. 7 video sequences were obtained from the DAVIS video segmentation dataset [85] and came with existing silhouettes, and the remainder were sourced from online stock footage. Segmentation masks for these were generated using Adobe’s UltraKey tool [1]. For all sequences, a set of 20 joints were labelled as illustrated in Fig. 3.7. The first 16 joints are located on legs, neck and tail, and defined by the rigged 3D SMAL skeleton. The final joints (nose tip, chin and ears) do not directly relate to 3D SMAL joints, but are instead defined by particular SMAL model vertices. A similar trick is used in SMALify [13] to indicate the position of the human nose which has no corresponding SMPL joint. The collection of joints were chosen on the basis of being informative to the skeleton and being simple for a non-expert human annotator to localize. To make manual annotation feasible and to ensure a diverse set of data, annotations are provided for every fifth video frame and were collected using the excellent LabelMe tool [LabelMe].

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.

### 3.6.2 Joint prediction

For the joint predictor  $\rho$  the stacked hourglass network [79] modified for multi-modal output is trained on synthetic animal silhouette data. Following state-of-the-art performance on related human 2D pose estimation datasets ([4, 64]), the network consists of 8 stacks, 256 features and 1 block. Synthetic silhouette images of size  $256 \times 256$  are provided as input, 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. As a benefit of working with synthetic data, training samples can be generated on the fly, which effectively results in a training set of infinite size. A small adaptation was required to prevent the network degenerating to an unfavourable solution on silhouette input: foreground masks were applied to both

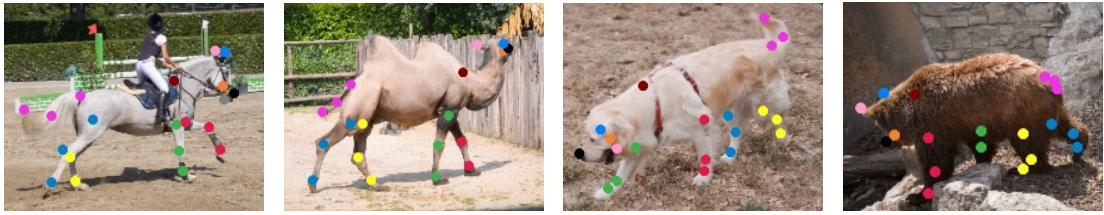


Fig. 3.7 Example joint annotations from the BADJA dataset. A total of 11 video sequences are in the dataset, annotated every 5 frames with 20 joint positions and visibility indicators.

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  $2.5 \times 10^{-4}$ . 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 [136]. 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  $\alpha$  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 3.12 contains examples of failure modes.

### 3.6.3 Optimal joint assignment

Following non-maximum suppression of the joint heatmaps obtained in Section 3.6.2, OJA is applied to select an optimal set of joints which initialize the final 3D model fitting 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 configurations (Fig. 3.8, row 1) or by resolving the ambiguity between the left and right legs (Fig. 3.8, row 2). Table 3.5 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

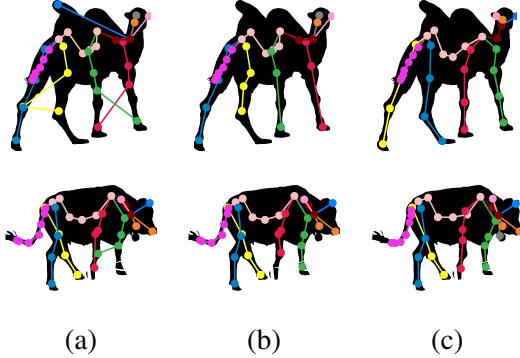


Fig. 3.8 Example skeletons from raw predictions (a), processed with OJA-QP (b), and OJA-GA (c).

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

Table 3.5 Accuracy of OJA on BADJA test sequences.

term. Across all sequences the selected OJA-GA method improves joint prediction accuracy by 7% compared to the raw network output.

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 3.6 Quantitative evaluation on synthetic test sequences. The performance of the raw network outputs and OJA methods are evaluated using the probability of correct keypoint (PCK) metric. Mesh fitting accuracy is evaluated by computing the mean distance between the predicted and ground truth vertices.

### 3.6.4 Model fitting

The predicted joint positions and silhouette are input to the 3D model fitting optimization phase, which proceeds in four stages. The first stage solves for the model’s global rotation and translation parameters, which positions the camera. Following SMPLify [13], this camera stage is solved for torso points only, which remain largely fixed through shape and pose variation. The remaining stages solve for all shape, pose and translation parameters the emphasis of the priors gradually decreased. The silhouette term is introduced in the penultimate stage, as including this too early can lead to the optimizer finding unsatisfactory local minima.

The final outputs of our optimization pipeline are shown in Fig. 3.11. 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. 3.11 demonstrates the generalizability of the

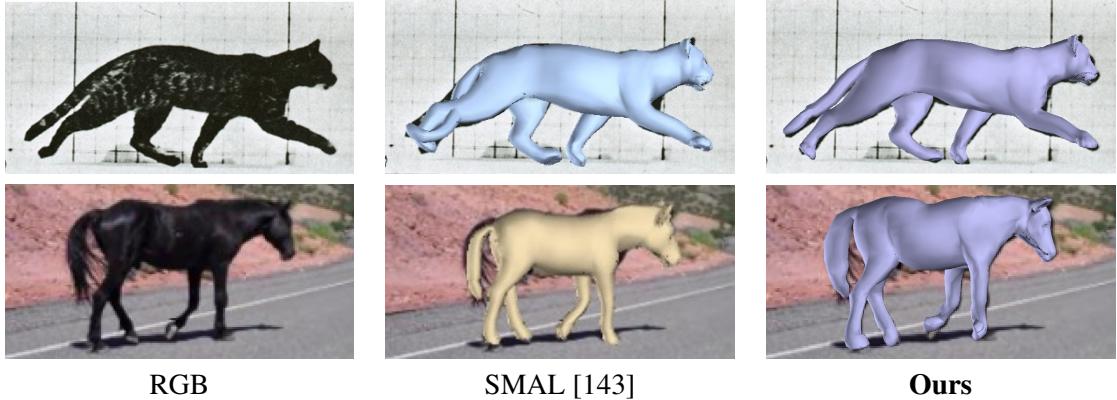


Fig. 3.9 Results are comparable in quality to SMAL [143], but note that we do not require hand-clicked keypoints.



Fig. 3.10 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.

proposed method: the algorithm is able to find a reasonable pose despite no camel figurines being included in the original SMAL model.

### Comparison to other work.

Our approach is compared visually to that given by Zuffi *et al.* [143]. Recall that their results require hand-clicked keypoints rather than fitting 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 the focus of this work is on the more challenging task of tracking animals in video. Fig. 3.9 shows the application of our model to a number of single frame examples from the SMAL result data [143].

### 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 3.10. Table 3.6 shows results on these sequences.

### 3.6.5 Automatic silhouette prediction

While not the main focus of our work, the system presented in this chapter is able to perform the full 3D reconstruction process from an input image with no user intervention. This is achieved by incorporating the DeepLabv3+ network [22] as a front-end segmentation engine which automatically generates 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. 3.11 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. 3.12 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.

### 3.7 Conclusions

The chapter introduces a technique for reconstructing 3D quadrupeds from video by using a quadruped model parameterized in shape and pose. By incorporating automatic segmentation tools, the pipeline can be deployed without requiring human intervention or even precise knowledge of the species of animal being considered. The method performs well on examples encountered in the real world, generalizes to unseen animal species and is robust to challenging shapes and poses.

As a direction for future work, it would be worthwhile to look at methods for synthetic image generation that preserve important edge information lost during silhouette extraction. In particular, the lost interior contours cause ambiguities which necessitate the OJA method described here. An alternative is to extend the recent work of SMALST [SMALST] (which requires hand-clicked training images) by instead synthetizing realistic textures using generative adversarial networks. Of course, it is important to ensure the texture generation process preserves the sampled pose parameters. A naive method is to render texture as a UV map on top of the SMAL model, but this could also be framed as an image-to-image translation problem, beginning with a low detail SMAL render and mapping to a photorealistic image with a carefully designed pose-preserving generator. Other components of the system which could be improved would be building a more sophisticated motion prior, able to represent likely animal trajectories. In addition, robustness to environmental factors as shown in Figure 3.12 could be improved by rendering synthetic causes of occlusion (perhaps even as rectangles).

## Chapter 4

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

### 4.1 Introduction

This chapter introduces WLDO, an automatic and end-to-end neural network which recovers 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. In addition, the dog category is poorly represented by existing 3D morphable models. Despite the ubiquity of dogs in society (there are more than 63 million pet dogs in the US alone [3]), these factors have perhaps contributed to the lack of effective 3D reconstruction methods. This chapter demonstrates that the natural variation present in a 2D dataset is sufficient to learn a detailed 3D dog prior, which helps regularize parameter estimation. WLDO achieves this by adding additional limb scaling parameters to the morphable model, and including expectation maximization (EM) update steps into the network training loop. These contributions are shown experimentally to improve the quality of reconstruction. Presently, state of the art animal 3D reconstruction techniques incorporate per-image (or per-sequence as in Chapter 3) test time energy minimization procedures that prevent real-time application. Inspired by recent methods for human reconstruction [53], WLDO comprises an end-to-end convolutional neural network which directly regresses morphable model parameters with no subsequent energy minimization phase, achieving inference at approximately 10 frames per second. This is important for downstream tasks, such as animal monitoring systems, which rely on live data in order to alert experts to immediate causes for concern. The experimentation section is based on StanfordExtra, a new ‘in the wild’ dataset of dog images, containing 120 breeds. The method presented achieves state of the art performance on this dataset, shows strong generalization characteristics to a new dataset, and outperforms model fitting approaches, even when they are given access to ground truth annotations at test time.

### 4.1.1 Adding local parameters to the PCA shape space

As discussed in Chapter 2, there is a long history of 3D reconstruction approaches which use morphable models to represent articulated subjects. Recently, the dominant paradigm is to factor the deformation space into a parameterization based on pose (which governs articulated limb positions) and a global PCA shape space governing body proportions. While efficient and differentiable, even as early as the seminal paper of Blanz and Vetter [1] it was noted that globally defined shape representations are poor at representing fine details (in their case, features such as the eyes and nose). Blanz and Vetter proposed tackling this by manually segmenting the face into separate regions and learning separate PCA models per regions. While this does achieve higher fidelity modelling, it comes at the cost of a less compact shape space representation and a need to manually partition the object. Alternatively, and most closely related to the work in this chapter, is the recent work of [STAR] which serves as a drop-in replacement for the original SMPL [SMPL] human model. The authors note that the use of SMPL’s global blend parameters result in the need for dense pose-corrective offsets in order to relate every mesh vertex with every kinematic tree joint. This causes an unwanted effect of capturing spurious long-range correlations between seemingly unrelated parts of the mesh. STAR offers an improvement with a local formulation, learning a set of mesh vertices which are influenced by each joint’s movement. Another limitation of globally defined PCA spaces (which is of particular relevance to Chapter 5) is that it is difficult to relate uncertainties in reconstructed shape to specific parts of the mesh. For example, since each body part is governed by multiple shape parameters, so are the uncertainties, making it challenging to reason about occluded parts. These challenges are compounded in a low training data setting. The SMAL animal model is built from scans of 41 toy figurines, which results in a PCA shape space that captures unwanted correlations coincidentally present in the training corpus. Examples of this are shown in Figure XXX. This chapter demonstrates an approach for adding a few extra local shape parameters to the SMAL model, allowing the body part to scale independently to the rest of the mesh. These parameters help the model generalize to dogs outside of the 3D training samples, are an inexpensive addition to the generator function and produce better shape reconstructions for WLDO and a previous approach based on energy minimization.

### 4.1.2 Automatic and real-time 3D dog reconstruction

Early 3D reconstruction approaches in humans [examples] use an energy minimization framework which iteratively optimizes a 3D morphable model to match an input image or video sequence. These methods typically design an energy function that balances *data terms* to encourage strong alignment between the 3D model and input image, and *prior terms* which ensure realistic predictions. However, more recent works frame the reconstruction task as a direct regression from the input image to 3D model parameters. These are typically implemented using convolutional neural networks which are trained on large and varied datasets and now achieve state-of-the-art performance on a number of benchmark datasets. Alongside fast test-time performance, deep learning methods learn accurate priors over the object category which lead to improved performance on images with occlusion and

they do not fall into failure cases common to optimization techniques, generally caused by poor model initialization. However, a downside of deep learning methods is their reliance on large datasets. Apart from the input 3D morphable model (e.g. SMPL [SMPL]), typically required are large *paired 3D datasets* and *unpaired 2D datasets* containing images and 2D annotations. Paired 3D datasets help models learn an association between input image features and 3D shape and pose parameters. However, such datasets require specialized equipment to collect (e.g. motion capture) resulting in limited variety in captured scenes. To overcome this, unpaired 2D datasets typically containing 2D keypoint and/or silhouette annotations are used to help models generalize to “in-the-wild” scenarios. Occasionally, these methods are also evaluated in an ‘unpaired’ setting, in which the paired datasets are omitted. Under these conditions, methods must overcome fundamental ambiguities to learn the 2D-to-3D mapping, or risk predicting 3D bodies with impossible joint angles or have implausible body weight distributions. Explicit 3D priors are often learned during training to ensure the predicted models lie on the manifold of plausible bodies. The “unpaired” training mode is of most relevance to this chapter’s task of reconstructing 3D dogs, since paired 3D animal data is severely limited. Of concern, then, is how to design a suitable prior for the dog category. This is explored in depth in the following sections.

Existing 3D animal reconstruction techniques are either entirely optimization based or incorporate optimization procedures as a part of the prediction pipeline. Notable exceptions include deep networks used to reconstruct unarticulated categories such as birds (e.g. CMU, UCMU) or the technique of Kulkarni et al. [] which operates on articulated categories but does not recover shape characteristics. SMALST is perhaps the closest attempt to an end-to-end technique for articulated shape and pose recovery, although video sequences are required for training and a test-time optimization strategy is used to refine initial regressed parameters, preventing real time operation. Further, it can be argued that the zebra species tackled in this paper is more limited than the 120 dog breeds examined in this chapter. Inspired by these approaches and the recent end-to-end work in human mesh reconstruction, WLDO comprises a deep neural network that directly regresses 3D dog model parameters *with no subsequent optimization phase* to enable real-time inference.

### 4.1.3 Learning a 3D animal prior

While detailed 3D morphable models and associated priors are already available for humans [,] the limited 3D training data for animals has made designing equivalent resources more difficult. The SMAL paper proposed an approach for building a morphable quadruped models from a few toy figurines rather than scans of real subjects (such as used for SMPL [SMPL]). Shape and pose priors were similarly collected by fitting the SMAL model to 2-3 artist animation sequences. Consequently, the SMAL model and priors are of relatively low fidelity and poorly represent some species, particularly dogs. Alongside the prevalence of challenging poses, occlusion and difficult environmental factors, overly restrictive animal priors have likely contributed to the lack of effective 3D reconstruction approaches for the dog category. Chapter 3 has already demonstrated an example of this, in which

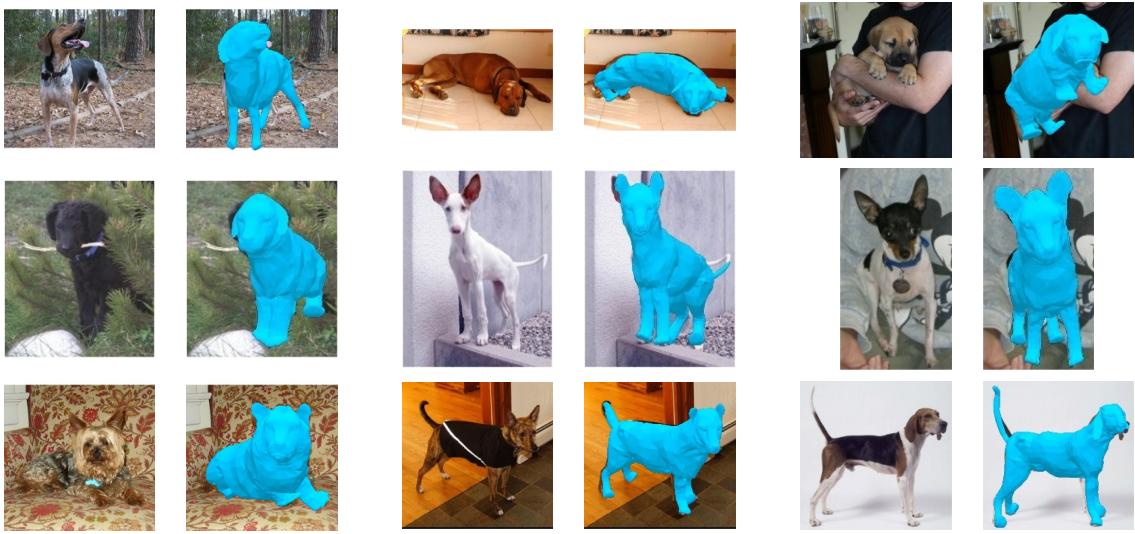
synthetically data generated according to the SMAL prior was used to train a 2D joint predictor. While high quality results are shown on categories better represented by SMAL, the experimental section of this chapter shows the method generalizes poorly to some out-of-domain dog breeds.

Some recent deep learning methods overcome this by forgoing data-driven 3D priors for the regression task altogether. SMALST is one such example; they supervise a deep network using 3D synthetic models generated using video sequences. Other techniques [] instead use smoothness terms, deformation constraints and symmetry constraints, although these are only effective when modelling unarticulated categories such as birds and dolphins. Of course, the quality of 3D morphable models, animal priors and potentially reconstruction results could be improved by collecting larger datasets of detailed 3D scans. However, collecting such a dataset would be expensive and time-consuming, either relying on scans of real animal subjects or contracting talented 3D graphics artists to build hundreds (or thousands) of accurate models. This chapter proposes an alternative approach, arguing that even a low-quality, unimodal 3D shape prior can act as a useful initialization for a novel *refinement process*, which learns a more expressive, multimodal prior by learning from an annotated 2D dataset.

#### 4.1.4 Refinement steps “in the loop”

A key insight relied upon in this chapter is the potential for collaboration between a deep neural network which processes input images to regress 3D model parameters and an optimization process that tunes a 3D shape prior. The technique used for this was inspired by the SPIN network of Kolotouros et al.[] who introduced a 3D human reconstruction approach based on a similar hybrid. In their approach, the HMR backbone [HMR] is used to yield a set of SMPL [SMPL] human body shape and pose parameters  $\Theta_{reg}$  from an input image. However, during SPIN’s training phase the initial fit  $\Theta_{reg}$  is processed by the SMPLify [SMPLify] model fitting procedure, which refines the initial fit based on the ground truth joints to yield a new estimate  $\Theta_{opt}$ . The difference between these two predictions is expressed as a loss  $\|\Theta_{reg} - \Theta_{opt}\|$  which is backpropagated to incrementally improve the predictions of the regression network. At test time, only the regression network is used meaning the inference speed is unaffected.

A related approach is used in this chapter in order to incrementally improve the representational power of a 3D shape prior. In particular, WLDO’s training phase incorporates a optimization strategy based on expectation maximization which regularly updates the prior based on weights learned from an annotated 2D image dataset. Similarly to SPIN, this creates a collaborative effect: the representational power of the 3D shape prior gradually improves by learning from better predictions produced by the 3D reconstruction network, and the 3D reconstruction network learns from the improving shape prior to produce accurate fits. Experimentally, it is shown that WLDO’s reconstructed dog models are of good enough quality to avoid a time consuming test time optimization process. Further details introducing expectation maximization and it’s use in WLDO’s training loop are deferred to the method section below.



**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.

#### 4.1.5 Contributions

This chapter proposes a number of contributions which extend the state of the art in 3D animal reconstruction in several ways. While each contribution is inspired by recent literature, the WLDO approach is the first to exhibit the combination, leading to a new state of the art state of the art in terms of scale and object diversity.

1. Reconstructing pose and shape on a test set of 1703 low-quality internet images of a complex 3D object class (dogs).
2. Direct regression to object pose and shape parameters from a single image without a model fitting stage.
3. Use of easily obtained 2D annotations in training, and none at test time.
4. Learning of a new multi-modal prior in the training phase (via EM update steps), rather than fitting it to 3D data as in previous work.
5. Introducing new degrees of freedom to the SMAL model, allowing explicit scaling of subparts.

## 4.2 Building SMBLD: a new parametric dog model

At the heart of the method is a parametric mesh representation of a 3D animal, which is based on the Skinned Multi-Animal Linear (SMAL) model proposed by Zuffi et al. [143]. As discussed

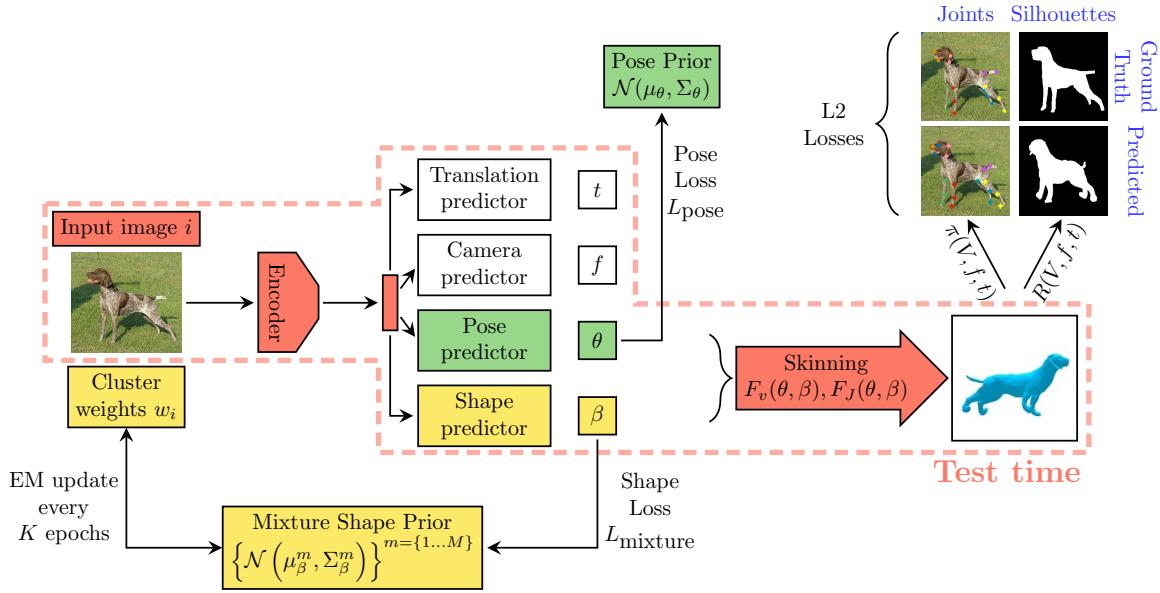


Fig. 4.2 Our method consists of (1) a deep CNN encoder which condenses the input image into a feature vector (2) a set of prediction heads which generate SMBLD parameters for shape  $\beta$ , pose  $\theta$ , camera focal length  $f$  and translation  $t$  (3) skinning functions  $F_v$  and  $F_J$  which 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.

in Chapter 3, SMAL is a deformable 3D quadruped 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 ( $35 \times 3$  Rodrigues parameters) which effect the articulated limb movement.

First, it is important to give a detailed description of the SMAL generator function, which is later augmented with scale parameters to improve model flexibility.

$$v(\theta, \beta) = W(T_P(\theta, \beta), J(\beta), \theta, \mathcal{W}) \quad (4.1)$$

$$T_P(\theta, \beta) = \bar{T} + B_S(\beta) + B_P(\theta) \quad (4.2)$$

Here, SMAL relies on a standard linear blend skinning (LBS) formulation to return a set of posed vertices from SMAL template vertices in rest pose  $\bar{T}$ , SMAL 3D joint locations  $J$ , a pose vector  $\theta$  and blend weights  $\mathcal{W}$

$$W(\bar{T}, J, \theta, \mathcal{W}) : \mathbb{R}^{3N \times 3K \times |\theta| \times |\mathcal{W}|} \mapsto \mathbb{R}^{3N} \quad (4.3)$$

The rest pose mesh  $T_P(\theta, \beta) \in \mathbb{R}^{3889 \times 3}$  is constructed from the SMAL template vertices  $\bar{T}$  and additive vertex offsets  $B_S(\beta)$  and  $B_P(\theta) \in \mathbb{R}^{3889 \times 3}$ . These are referred to as shape and pose blend

shapes respectively. Note that despite being a function of pose,  $B_P(\theta)$  still represents a shape modification. This is often referred to as a *corrective* offset, fixing artefacts common to linear blend skinning, often by fat. The pose vector  $\theta = [\omega_0^T, \dots, \omega_J^T]^T$  contains 3 parameters for each joint. Each element  $\omega_j \in \mathbb{R}^3$  denotes the axis-angle representation of the relative rotation of each part  $j$  with respect to its parent in the kinematic tree. The local  $3 \times 3$  rotation matrix corresponding to a joint  $j$  is denoted  $\exp(\omega_j)$  and is obtained from axis angle representation using the *Rodrigues formula*

$$\exp(\omega_j) = \mathcal{I} + \hat{\omega}_j \sin(\|\omega_j\|) + \hat{\omega}_j^2 \cos(\|\omega_j\|) \quad (4.4)$$

where  $\hat{\omega}_j$  is the skew symmetric matrix of the 3-vector  $\bar{\omega} = \frac{\omega}{\|\omega\|}$  representing the unit norm axis of rotation.  $\mathcal{I}$  is the  $3 \times 3$  identity matrix.

The linear blend skinning function (LBS) is then used to articulated the rest pose mesh  $T_P(\theta, \beta)$ . Let  $t_i = \bar{t}_i + b_{S,i}(\beta) + b_{P,i}(\theta) \in T_P(\theta, \beta)$  be a *shaped* vertex in rest pose where  $b_{S,i}(\beta)$  and  $b_{P,i}(\theta)$  denotes the same vertex in  $B_S(\beta)$  and  $B_P(\theta)$  respectively. The transformation to obtain a *posed* vertex  $t'_i$  is then

$$t'_i = \sum_{k=1}^K w_{k,i} G'_k(\theta, J) t_i \quad (4.5)$$

$$G'_k(\theta, J) = G_k(\theta, J) G_k(\theta^*, J)^{-1} \quad (4.6)$$

$$G_k(\theta, J) = \prod_{j \in A(k)} \left( \begin{array}{c|c} \exp(\omega_j) & j_j \\ \hline \mathbf{0} & \mathbf{1} \end{array} \right) \quad (4.7)$$

Here,  $w_{k,i}$  is an element of the blend weight matrix  $\mathcal{W}$ , representing how much the rotation of part  $k$  effects the vertex  $i$ ,  $\exp(\omega_j)$  is the local  $3 \times 3$  rotation matrix corresponding to joint  $j$ ,  $G(\theta, J)$  is the world transformation of joint  $k$ , and  $G'(\theta, J)$  is the same transformation after removing the transformation due to the rest pose  $\theta^*$ . Each 3-element vector in  $J$  corresponding to a single joint center  $j$ , is denoted  $j_j$ . Finally,  $A(k)$  denotes the ordered set of joint ancestors of joint  $k$ .

### Shape blend shapes

Shape blend shapes are applied to the SMAL rest model to smoothly modify the mesh's body proportions. This has the effect of appearing to alter the animal category. The body shapes are globally represented by a linear function  $B_S$

$$B_S(\beta; \mathcal{S}) = \sum_{n=1}^{|\beta|} \beta_n S_n \quad (4.8)$$

where  $\beta \in \mathbb{R}^{41}$  is a vector of linear shape coefficients and  $S_n$  represents the orthonormal principle components of shape displacements, learned from the registered training scans.

### Pose blend shapes

Let  $R : \mathbb{R}^{|\theta|} \mapsto \mathbb{R}^{9K}$  be a function which maps the pose vector  $\theta$  to a vector of concatenated part relative rotation matrices  $\exp(\theta)$ . Given the SMAL model contains 33 joints,  $R(\theta)$  is a vector of length  $33 \times 9 = 297$ . Since elements of  $R(\theta)$  comprise sine and cosine functions of the input pose vector,  $R(\theta)$  is therefore non-linear with respect to  $\theta$ .

The pose blend shapes are defined to be linear in  $R^*(\theta) = (R(\theta) - R(\theta^*))$  where  $\theta^*$  denotes the rest pose. Letting  $R_n(\theta)$  denote the  $n^{th}$  element of  $R(\theta)$ , the vertex deviations from the rest template are

$$B_P(\theta; \mathcal{P}) = \sum_{n=1}^{9K} (R_n(\theta) - R_n(\theta^*)) P_n \quad (4.9)$$

An important observation is that by subtracting the rest pose  $R(\theta^*)$ , the contribution of the pose blend shape is zero when the model is in the rest pose.

### Joint locations

Differing animal body shapes have an effect on the 3D joint locations.

$$J(\beta; J, \bar{T}, \mathcal{S}) = \mathcal{J}(\bar{T} + B_S(\beta; \mathcal{S})) \quad (4.10)$$

where  $\mathcal{J}$  is a matrix that transforms rest vertices into rest joints.

### Complete formulation

The complete formulation is therefore given as

$$v(\theta, \beta; \bar{T}, \mathcal{W}, \mathcal{S}, \mathcal{J}, \mathcal{P}) = W(T_P(\theta, \beta; \bar{T}, \mathcal{S}, \mathcal{P}), J(\beta; \mathcal{J}, \bar{T}, \mathcal{S}), \theta, \mathcal{W}) \quad (4.11)$$

#### 4.2.1 Introducing scale parameters

While SMAL has been shown to adequately represent a variety of quadruped types, the modes of dog shape variation are poorly captured by the current model. This is unsurprising, since SMAL used only four dogs in its construction. This limitation is overcome using a simple but effective method that improves the model's representational power over this particularly diverse animal category. The set of shape parameters  $\beta$  are augmented with an additional set  $\kappa$  which independently scale parts of the mesh. For each model joint, parameters  $\kappa_x, \kappa_y, \kappa_z$  are defined which apply a local scaling of the mesh along the local coordinate  $x, y, z$  axes, before pose is applied.

A simple method for adding scaling parameters is to apply a transformation to the rotation matrices for each joint. Equation (4.7) then becomes

$$G_k(\theta, J) = \prod_{j \in A(k)} \left( \frac{Q_j \exp(\omega_j)}{\mathbf{0}} \middle| j_j \right) \quad (4.12)$$

where the matrix

$$Q_j = \begin{pmatrix} \kappa_{j,x} & 0 & 0 \\ 0 & \kappa_{j,y} & 0 \\ 0 & 0 & \kappa_{j,z} \end{pmatrix} \quad (4.13)$$

Unfortunately, this formulation causes challenges should a user wish to scale a mesh subpart along the kinematic tree. While it is natural for joint *rotations* to be compositionally applied along the kinematic tree (mirroring the effect of skeletal limb motion) this does not follow for scaling parameters. For example, it is common for a user to apply scaling to a subpart which does not necessarily affect subsequent parts. With the formulation at present, it would be the responsibility of the user to “undo” scaling for the remaining tree, requiring them to be aware of the kinematic tree index ordering. This is seen as suboptimal behaviour, so the following formulation is preferred, defining *absolute* scale parameters that do not propagate their effect. Taking advantage of the hierarchical ordering of the kinematic tree for joint  $k$ ,  $A(k) = [A(k)_0, A(k)_1, \dots, k]$  are joints along the tree. Therefore, the absolute scaling matrix is given by

$$G_k(\theta, J) = \left( \begin{array}{c|c} \exp(\omega_{A(k,0)}) Q_{A(k,0)} & j_{A(k,0)} \\ \mathbf{0} & \mathbf{1} \end{array} \right) \prod_{j \in [A(k)_1, \dots, k]} \left( \begin{array}{c|c} Q_{j-1}^{-1} \exp(\omega_j) Q_j & j_j \\ \mathbf{0} & \mathbf{1} \end{array} \right) \quad (4.14)$$

Note that  $Q_{j-1}$  refers to the parent scaling matrix of joint  $j$ . Note that the scale parameters in this formulation determine only the scale for the particular subpart by undoing that of the parent.

### 4.2.2 Sharing scale parameters

Allowing each joint to scale entirely independently can however lead to unrealistic deformations. To overcome this, scale parameters are shared between multiple joints which are physiologically related and would naturally scale together. For example, the y-scaling for the four legs (which governs the length) is controlled by only a single parameter. 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. The next task is to learn an prior (with which to initialize the later EM process) that covers these new scale parameters.

### 4.2.3 Initializing a dog shape/scale prior

The SMBLD shape prior is initialized by fitting the model to a set of 13 artist-designed 3D dog meshes, downloaded from the internet and are similar in style to the SMAL toys. While the collection does offer a few more examples of dogs than used in the original SMAL set, the models still lack realism and are very constrained when compared to the 120 different breeds represented in the later test set. An energy minimization process is used to align the SMBLD vertices to each scan, in order to obtain a set of shape  $\beta$ , pose  $\theta$  and  $\kappa$  parameters.

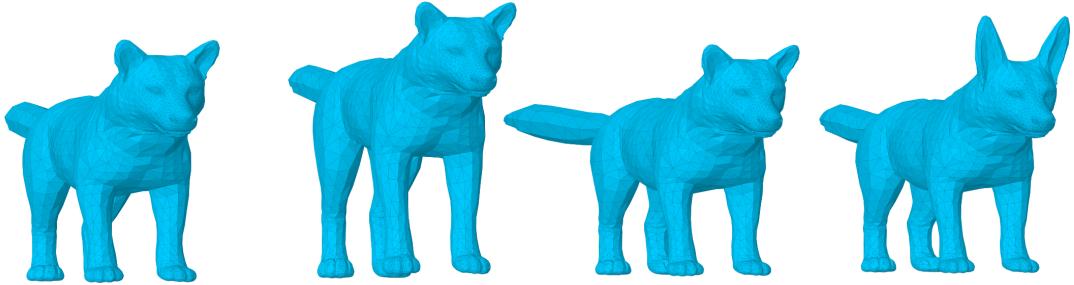


Fig. 4.3 **Effect of varying SMBLD scale parameters.** *From left to right:* Mean SMBLD model, 25% leg elongation, 50% tail elongation, 50% ear elongation.

### Fitting SMBLD to 3D scans

Recall that SMBLD’s shape  $\beta$  and  $\theta$  parameters are exactly as defined in SMAL, meaning there is no need to relearn (or adapt) the blend shape data  $\mathcal{P}, \mathcal{S}$  or joint selectors  $K$ . The scale  $\kappa$  parameters, however, are new to SMBLD. The goal of this section is to learn a new 3D prior over both shape and scale parameters to initialize the prior refined during the WLDO training loop.

Note that fitting SMBLD to 3D scans emits much simpler optimization in comparison to 2D images, since the complete 3D information of the target mesh is available. In addition, the target meshes are not particularly detailed and are already aligned in the canonical T-pose, so we avoid need for a complex alignment technique as discussed in Chapter 3.

An energy minimization process is used to align the SMBLD mesh  $v(\theta, \beta)$  to each of the 3D scans  $[(V, T)_i]_{i=1}^N$  subject to smoothing regularizers. The following energy formulation is minimized using stochastic gradient descent (SGD) with learning rate  $1.0e^{-4}$  for 1000 iterations

$$E_{\text{opt}} = E_{\text{chamfer}} + E_{\text{laplacian}} + E_{\text{edge}} + E_{\text{normal}} \quad (4.15)$$

where each of these terms has a scalar weight  $\lambda$ . Here,  $\lambda_{\text{chamfer}} = \lambda_{\text{edge}} = 1.0$ ,  $\lambda_{\text{normal}} = 0.01$  and  $\lambda_{\text{laplacian}} = 0.1$ . Further details on the specific energy terms are now provided.

#### Chamfer energy.

Chamfer energy measures the average distance between vertices of the SMBLD mesh  $V = v(\theta, \beta)$  and the target mesh vertices  $V'$ . Let  $V_P = [v_i]_{i=1}^p$  and  $V'_P = [v'_i]_{i=1}^p$  be random  $p$ -subsets of  $V$  and  $V'$  respectively. The chamfer energy is then

$$E_{\text{chamfer}}(V_P, V'_P) = \frac{1}{p} \sum_{i=1}^p \min_j |v_i - v'_j| \quad (4.16)$$

### Uniform laplacian energy.

The SMAL mesh can be viewed as  $M = (v(\theta, \beta), T)$  where the vertices are given by the generator function above and the triangulation is fixed. The uniform Laplacian matrix  $L \in \mathbb{R}^{N \times N}$  is defined as  $LV = w_i \sum_j (v_j - v_i)$  where  $w_{ij} = 1/|S_i|$  and  $S_i$  is the set of neighbouring vertices to  $i$ . Note that  $LV_i$  points to the centroid of the neighbouring vertices. The energy formulation is then

$$E_{\text{laplacian}} = \sum_i w_i \sum_j (v_j - v_i) \quad (4.17)$$

### Edge energy.

This energy is equal to the average edge length across the mesh, and is used to encourage uniform distribution of vertices. Let  $E = \{(i, j) \mid v_i, v_j \in V \text{ are adjacent vertices}\}$ , then

$$E_{\text{edge}} = \frac{1}{|E|} \sum_{(i,j) \in E} \|v_j - v_i\| \quad (4.18)$$

### Normal energy.

Normal energy is a measure of the average normal consistency between adjacent triangles. For two triangles  $t_i, t_j \in T$  in SMAL's fixed triangle set, let  $n_i, n_j$  be the respective normals. Recall that  $n = (v_1 - v_0) \cdot (v_2 - v_0)$ . Then normal consistency is defined by

$$E_{\text{normal}} = 1 - \frac{n_i \cdot n_j}{|n_i| |n_j|} \quad (4.19)$$

At the end of this process, we have a collection of fits  $\{(\theta, \beta)\}_{\{i=1, \dots, 13\}}$  from which we can learn our unimodal pose and shape priors. As discussed, we eventually use this unimodal shape prior to initialize our mixture shape prior, which is tuned with the expectation-maximization step in the training loop.

### Learning a unimodal prior

Recall that the SMAL body shapes are represented by the linear function in Equation (4.8). In this formulation, the matrix  $S_n$  represents the orthonormal principal components of shape displacements. A consequence of the PCA construction is that the features in  $B$ -dimensional parameter space (as they are linear combinations of the original features) are normally distributed with mean  $\mu_\beta$  and covariance matrices  $\Sigma_\beta$ . With this construction, a likelihood function measures the probability of a given shape vector  $\beta \in \mathbb{R}^B$

$$\mathbb{P}_\beta(\beta) = (2\pi)^{-\frac{B}{2}} \det(\Sigma_\beta)^{-\frac{1}{2}} e^{-\frac{1}{2}(\beta - \mu_\beta)^T \Sigma_\beta^{-1} (\beta - \mu_\beta)} \quad (4.20)$$

For problems which aim to optimize  $\beta$  as a free variable, the 3D shape prior is obtained by maximizing  $\mathbb{P}_\beta(\beta)$  or equivalently, by minimizing the negative log likelihood

$$-\ln [\mathbb{P}_\beta(\beta)] = -\frac{1}{2} [\ln \det(\Sigma_\beta) + B \ln(2\pi) + (\beta - \mu_\beta)^T \Sigma_\beta^{-1} (\beta - \mu_\beta)] \quad (4.21)$$

Finally, terms with no dependency on  $\beta$  can be dropped since they remain constant during the optimization. This reduces the formulation of the loss as the Mahalanobis distance of  $\mu_\beta$

$$L_{\text{shape}}(\beta; \mu_\beta, \Sigma_\beta) = (\beta - \mu_\beta)^T \Sigma_\beta^{-1} (\beta - \mu_\beta) \quad (4.22)$$

A similar construction applies to the scale parameters  $\kappa$  with means and covariances computed from the 13 training scans. The scale loss function is therefore given as

$$L_{\text{scale}}(\kappa; \mu_\kappa, \Sigma_\kappa) = (\kappa - \mu_\kappa)^T \Sigma_\kappa^{-1} (\kappa - \mu_\kappa) \quad (4.23)$$

Continuing from here, it is convenient to group shape and scale parameters into a new shape-scale variable  $\delta$  and the loss function can be grouped as follows

$$L_{\text{shape-scale}}(\delta = [\beta, \kappa]; \mu_\delta = [\mu_\beta, \mu_\kappa], \Sigma_\delta = [\Sigma_\beta, \Sigma_\kappa]) = L_{\text{shape}}(\beta; \mu_\beta, \Sigma_\beta) + L_{\text{scale}}(\kappa; \mu_\kappa, \Sigma_\kappa) \quad (4.24)$$

### Extending to a multimodal prior

So far, a shape-scale prior has been introduced based on a unimodal, multivariate Gaussian with mean  $\mu_\delta$  and covariance matrix  $\Sigma_\delta$ . However, enforcing this throughout training tends to result in model predictions that appear similar in 3D shape, even when tested on dog images of different breeds. Diversity is improved among predicted 3D dog shapes by extending the above formulation to a Mixture of  $M$  Gaussians prior.

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

The mixture shape loss is given as:

$$L_{\text{mixture}}(\delta_i; \mu_\delta, \Sigma_\delta, \Pi_\delta) = \sum_{m=1}^M \Pi_\delta^m [L_{\text{shape-scale}}(\delta_i; \mu_\delta^m, \Sigma_\delta^m)] \quad (4.25)$$

## 4.3 End-to-end dog reconstruction from monocular images

This section describes a technique for reconstructing a 3D dog mesh from a single, monocular image. This is achieved using an end-to-end convolutional neural network, which predicts pose  $\theta$  and shape-scale  $\delta$  for the SMBLD morphable dog model, translation and orientation  $\phi$  and focal length  $f$  for a perspective camera  $\pi_f$ . A complete overview of the proposed system is shown in Figure 4.2.

### 4.3.1 Model architecture

The network architecture for WLDO is inspired by the model of 3D-Safari [141]. Given an input image cropped to (224, 224), a Resnet-50 [42] backbone network is applied in order 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-scale 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, two linear layers are used to map to a set of  $35 \times 3$  3D pose parameters (three parameters for each joint in the SMBLD kinematic tree and three parameters for the model orientation) given in Rodrigues form. The translation component of the SMBLD model is predicted with two linear layers that predict translation in the camera frame  $t_{x,y}$  and depth  $t_z$  independently. The camera prediction layer predicts the focal length of the perspective camera. Note that while this parameter is theoretically unnecessary since model depth is already predicted, as in the original 3D-Safari implementation, including the parameter produces empirically better fits. The camera focal length is obtained as  $f = f_0 + f_1 x$  where  $x$  is predicted by the network and  $f_0 = f_1 = 2700$ .

#### Shape and scale prediction.

Unlike 3D-Safari, WLDO predicts a set of shape and scale parameters rather than vertex offsets. Separate predictions heads are used to predict 20 shape parameters  $\beta$  and the new scale parameters  $\kappa$ . The scale parameters are obtained as  $\kappa = \exp x$  where  $x$  are the network predictions, as predicting log scale helps stabilise early training and conveniently guarantees non-negative values for scale.

### 4.3.2 Training losses

A typical approach for training such an end-to-end system would be to supervise the prediction of  $(\theta, \delta, \phi, f)$  with 3D ground truth annotations [60, 53, 83]. 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.

Instead, WLDO relies on *weak 2D supervision* provided by 2D keypoints and silhouette segmentations to guide network training. These data sources are significantly cheaper to obtain than 3D models.

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

### Joint reprojection.

As in Chapter 3, the joint projection loss  $L_{\text{joints}}$  promotes accurate limb positioning by comparing the projected model joints to the ground truth annotations  $X^*$ . To achieve this, the SMBLD generator function processes the network’s pose  $\theta$  and shape-scale  $\delta$  parameters to generate a set of 3D joint positions which are translated and orientated with  $\phi$ . The joint positions are then projected to the image plane using the camera parameters. The joint loss  $L_{\text{joints}}$  is given by the  $\ell_2$  error between the ground truth and projected joints:

$$L_{\text{joints}}(\theta, \delta, \phi, f; X^*) = \|X^* - \pi_f(\phi * v(\theta, \delta)K)\|_2 \quad (4.26)$$

Note that many of the training images contain significant occlusion, so  $X^*$  contains many invisible joints. This is handled by masking  $L_{\text{joints}}$  so that invisible joints do not contributing to the loss.

### Silhouette loss.

The silhouette loss  $L_{\text{sil}}$  is used to promote shape alignment between the SMBLD dog mesh and the input dog. In order to compute the silhouette loss, a rendering function  $R(\phi * v(\theta, \delta)) \in \mathbb{B}^{W \times H}$  projects the SMBLD mesh to produce a binary segmentation mask. In order to allow derivatives to be propagated,  $R$  is implemented using the differentiable Neural Mesh Renderer [56]. The loss is computed as the  $\ell_2$  difference between a projected silhouette and the ground truth mask  $S$ :

$$L_{\text{sil}}(\theta, \delta, \phi, f; S) = \|S - R(\phi * v(\theta, \delta))\|_2 \quad (4.27)$$

### Pose prior.

In the absence of 3D ground truth training data, priors are obtained from artist graphics models in order to encourage realism in the network predictions. The pose prior is modelled as a multivariate Gaussian prior consisting of mean  $\mu_\theta$  and a covariance matrix  $\Sigma_\theta$ . 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:

$$L_{\text{pose}}(\theta; \mu_\theta, \Sigma_\theta) = (\theta - \mu_\theta)^T \Sigma_\theta^{-1} (\theta - \mu_\theta) \quad (4.28)$$

$$(4.29)$$

Unlike previous work, there is no need to use a loss for penalizing pose parameters if they exceed manually specified joint angle limits. It is suspected that the WLDo network learns this regularization naturally by training on the large dataset.

### Shape-scale prior.

As discussed in the previous section, the shape prior is modelled as a mixture of  $M$  Gaussians. Recall that  $\mu_\delta^m$ ,  $\Sigma_\delta^m$  and  $\Pi_\delta^m$  are the mean, covariance and mixture weight respectively for each Gaussian component  $m$ . From before, the mixture shape loss is given as:

$$L_{\text{mixture}}(\delta_i; \mu_\delta, \Sigma_\delta, \Pi_\delta) = \sum_{m=1}^M \Pi_\beta^m [L_{\text{shape-scale}}(\delta_i; \mu_\delta^m, \Sigma_\delta^m)] \quad (4.30)$$

## 4.4 Expectation Maximization in the Loop

As previously discussed, the parameters for the mixture shape-scale prior are initialized using artist data which results in a prior that poorly represents the diverse shapes present in the real dog dataset. A key contribution presented in this chapter is the introduction of a procedure based on Expectation Maximization which gradually improves the representational power of the shape-scale prior by learning from monocular images of in-the-wild dogs and their respective 2D training labels.

Firstly, each training image is assigned a set of latent variables  $\{w_i^1, \dots, w_i^M\}$  which encode the likelihood that the dog in image  $i$  was generated according to each shape-scale component  $m \in \{1, \dots, M\}$ .

Expectation Maximization (EM) is then used to regularly update these image weights, as well as the parameters for the mixture  $(\mu_\delta^m, \Sigma_\delta^m, \Pi_\delta^m)_{i=1}^M$ . During the training loop of the 3D reconstruction network, these parameters are tuned according to alternating expectation ('E-') and maximization ('M-') steps which are described below:

### 4.4.1 Expectation Maximization update steps

#### The 'E' Step.

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

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

$$w_i^m := \frac{\mathcal{N}(\delta_i | \mu_\delta^m, \Sigma_\delta^m) \Pi_\delta^m}{\sum_{m'=1}^M \mathcal{N}(\delta_i | \mu_\delta^{m'}, \Sigma_\delta^{m'}) \Pi_\delta^{m'}} \quad (4.31)$$

To improve numerical stability, this is formulated using the log-sum-exp trick:

$$\log w_i^m := \frac{\log \left[ \Pi_{\delta}^m (2\Pi)^{\frac{d}{2}} [\det \Sigma_{\delta}^m]^{-\frac{1}{2}} \exp -\frac{1}{2} (\delta_i - \mu_{\delta}^m)^T (\Sigma_{\delta}^m)^{-1} (\delta_i - \mu_{\delta}^m) \right]}{\log \left[ \sum_{m'=1}^M \Pi_{\delta}^{m'} (2\Pi)^{\frac{d}{2}} [\det \Sigma_{\delta}^{m'}]^{-\frac{1}{2}} \exp -\frac{1}{2} (\delta_i - \mu_{\delta}^{m'})^T (\Sigma_{\delta}^{m'})^{-1} (\delta_i - \mu_{\delta}^{m'}) \right]} \quad (4.32)$$

And this simplifies to

$$\begin{aligned} \log w_i^m &:= \log \Pi_{\delta}^m - \frac{1}{2} [d \log 2\pi + \log \det (\Sigma_{\delta}^m) + M_{i,m}] \\ &\quad - \log \sum_{m'=1}^M \exp \log \Pi_{\delta}^{m'} - \frac{1}{2} [d \log 2\pi + \log \det (\Sigma_{\delta}^{m'}) + M_{i,m'}] \end{aligned} \quad (4.33)$$

where

$$M_{i,m} := (\delta_i - \mu_{\delta}^m)^T (\Sigma_{\delta}^m)^{-1} (\delta_i - \mu_{\delta}^m) \quad (4.34)$$

### The ‘M’ Step.

The ‘M’ step computes new values for  $(\mu_{\delta}^m, \Sigma_{\delta}^m, \Pi_{\delta}^m)$ , assuming fixed  $w_i^m$  for all  $i \in \{1, \dots, N\}$  and  $m \in \{1, \dots, M\}$ . The update equations are given as follows:

$$\mu_{\delta}^m := \frac{\sum_i w_i^m \delta_i}{\sum_i w_i^m} \quad \Sigma_{\delta}^m := \frac{\sum_i w_i^m (\delta_i - \Sigma_{\delta}^m)(\delta_i - \Sigma_{\delta}^m)^T}{\sum_i w_i^m} \quad \Pi_{\delta}^m := \frac{1}{N} \sum_i w_i^m \quad (4.35)$$

## 4.5 Building StanfordExtra: a new large-scale dog keypoint dataset



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.

In order to train and evaluate the method, it is necessary to source a large-scale dataset of dog images with associated 2D annotations. This chapter therefore introduces *StanfordExtra*: a new large-scale dataset with annotated 2D keypoints and binary segmentation masks for dogs. The source

images were sourced from the existing Stanford Dog Dataset [59], 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.

Annotatations were collected using Amazon Mechanical Turk. The annotators (termed ‘turkers’ on the platform) provided a binary silhouette mask and 20 keypoints per image. The remainder of this section describes the process for obtaining keypoint and segmentation annotations for the Stanford Dog Dataset [59].

The entire set of 20,580 dog images were first augmented with a single bounding box (provided with the original data) to indicate the largest dog in the image which the annotator should label. Each image was sent to 3 independent turkers with instructions explaining how to label the 20 keypoints and segmentation mask.

### 4.5.1 Keypoints.

Amazon turkers were given a list of 20 keypoints to click: 3 per leg (knee, ankle, toe), 2 per ear (base, tip), 2 per tail (base, tip), 2 per face (nose and jaw). They were additionally asked to provide a visibility flag per point.

For each keypoint, we process the three clicks to yield a reliable coordinate. From the 3 clicks, we discard clicks that are further than a set tolerance from the mean. If at least 2 clicks remain, the mean coordinate is accepted as the keypoint position. Otherwise, the point has not been reliably identified so is set invisible. In line with other methods which tackle “in-the-wild” 3D reconstruction of articulated subjects [60, 61], images with fewer than a threshold number of visible keypoints (in this case 8 keypoints) are not included in StanfordExtra filtered from the dataset. This is since these images are unsuitable for the primary purpose of StanfordExtra, to train and evaluate full-body dog reconstruction task.

### 4.5.2 Segmentation.

For each image, each worker  $t \in \{t_1, t_2, t_3\}$  submits a binary segmentation mask  $\mathbf{S}^t$ . Submissions which fail simple criteria, such as if the segmentation area is below a threshold number of pixels, are sent for relabelling.

For each image, the most likely segementation is generated by comparing submissions across workers. For any two turkers  $t, t'$ , a correlation coefficient is computed between submissions, summing over all pixels  $i$ . Note that  $\otimes$  denotes the element-wise product of the matrices.

$$c(t, t') = \frac{\sum_i [\mathbf{S}^t \otimes \mathbf{S}^{t'}]_i}{\max_{\tilde{t}=\{t, t'\}} \sum_i \mathbf{S}_{\tilde{t}}^{\tilde{t}}} \quad (4.36)$$

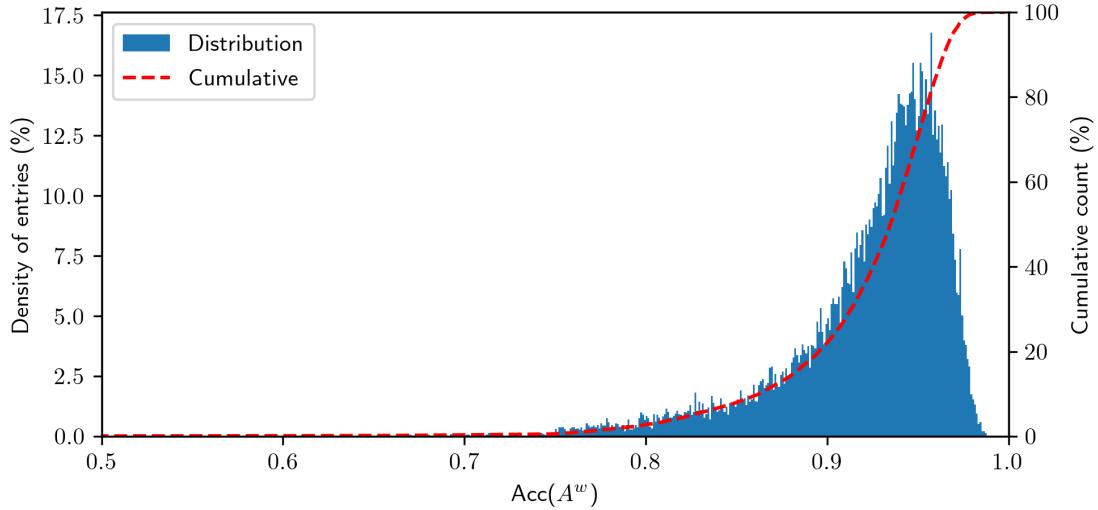


Fig. 4.5 Accuracy distribution of all submitted dog segmentations across the entire StanfordExtra dataset.

A worker’s segmentation  $S^t$  is removed if all correlation coefficients  $c(t, t')$  are below a set threshold. The final binary mask is computed from the remaining submissions:

$$S_i^* = \begin{cases} 1 & \text{if } \sum_t S_i^t > 1 \\ 0 & \text{otherwise} \end{cases} \quad (4.37)$$

With this, the accuracy of a worker’s segmentation can be estimated according to the largest of their correlation coefficients:

$$\text{Acc}(S^t) = \max_{\bar{t} \neq t} \{c(t, \bar{t})\} \quad (4.38)$$

### 4.5.3 Approximating the dataset complexity.

Figure 4.4 gives an indication of the complexity of the dataset by analysing the distribution of worker accuracies across the entire StanfordExtra dataset.

## 4.6 Experiments

In this section, the WLDO method is compared against competitive baselines. The protocol for evaluation is described first, followed by a quantitative and qualitative evaluation.

### 4.6.1 Evaluation protocol

Evaluation is based on the StanfordExtra dataset introduced earlier in the chapter, which contains 8,476 images for 120 breeds. These images are divided per-breed into an 80%/20% train and test split.

Two primary evaluation metrics are considered. 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. In addition, PCK results are produced for various joint groups (legs, tail, ears, face) in order to compare the reconstruction accuracy for different parts of the dog. Also used for evaluation is a new comparison metric *PCK-MAX*. This protocol is similar to the Percentage of Correct Keypoints (PCK) metric [136] but incorporates the ‘invisible’ ground-truth points. The standard PCK metric ignores these points, meaning even correct 3D reconstructions will receive no credit. PCK-MAX instead assumes reconstructed 3D points for missing ground-truth data are correct, providing an interesting upper bound.

#### 4.6.2 Training procedure

The WLDO network is trained in two stages. The first omits the silhouette loss which tends to lead the network to unsatisfactory local minima if applied too early. With the silhouette loss turned off, it is 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, the silhouette loss and mixture prior with  $M = 10$  clusters are introduced, and expectation maximization updates are applied every 50 epochs. The first stage is trained for 250 epochs and the second stage for 150. The entire training procedure takes approximately 96 hours on a single P100 GPU.

Recall that the training objective for our end-to-end system for predicting SMBLD parameters consistent with a monocular dog input image is given by:

$$L_{\text{opt}} = L_{\text{joints}} + L_{\text{sil}} + L_{\text{pose}} + L_{\text{shape}} + L_{\text{mixture}} \quad (4.39)$$

Each loss term is weighted with a scalar  $\lambda$ . The following details the specific weights used for each training stage:

##### **Stage 1.**

$\lambda_{\text{joints}} = 10.0, \lambda_{\text{pose}} = 1.0, \lambda_{\text{shape}} = 1.0, \lambda_{\text{sil}} = 0.0, \lambda_{\text{mixture}} = 0.0$ . This stage is trained for 250 epochs using the Adam optimizer, with learning rate set to  $10^{-4}$ .

##### **Stage 2.**

$\lambda_{\text{joints}} = 10.0, \lambda_{\text{pose}} = 0.5, \lambda_{\text{shape}} = 0.0, \lambda_{\text{sil}} = 100.0, \lambda_{\text{mixture}} = 0.1$ . This stage is trained for 150 epochs and run the described EM update step every  $K = 50$  epochs. The number of clusters  $M = 10$  was selected based on a grid search over  $M = 1, 5, 10, 25$  with IoU scores compared. The Adam optimizer is again applied with learning rate to  $10^{-5}$ .

Method	Kps	Seg	IoU	PCK @ 0.15			
				Avg	Legs	Tail	Ears
3D-M [143]	Pred	Pred	69.9	69.7	68.3	68.0	57.8
3D-M	GT	GT	71.0	75.6	74.2	89.5	60.7
3D-M	GT	Pred	70.7	75.5	74.1	88.1	60.2
3D-M	Pred	GT	70.5	70.3	69.0	69.4	58.5
CGAS [9]	CGAS	Pred	63.5	28.6	30.7	34.5	25.9
CGAS	CGAS	GT	64.2	28.2	30.1	33.4	26.3
3D-M + scaling	Pred	Pred	70.4	70.9	69.8	66.9	59.7
3D-M + scaling + EM prior	Pred	Pred	71.8	73.4	72.5	<b>70.3</b>	62.6
<b>Ours</b>	—	—	<b>74.2</b>	<b>78.8</b>	<b>76.4</b>	63.9	<b>78.1</b>
							92.1

Table 4.1 **Baseline comparisons.** PCK and silhouette IOU scores are shown for SOTA methods under varying conditions. Directly comparable baseline methods (requiring only an input image) are highlighted. *Pred* keypoints generated with Hourglass-Net [79] and segmentations with DeepLab v3+ [22]. 3D-M/CGAS are also analysed when they have access to ground-truth keypoints and/or segmentation masks. We also analyse adding this paper’s innovations (scale parameters and EM prior) to 3D-M [143].

To begin, WLDO is compared with various baseline methods. 3D Menagerie (3D-M) [143] is an approach which fits the 3D SMAL model using per-image energy minimization. Creatures Great and SMAL (CGAS) [9] 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 3D-M and CGAS rely on manually-provided segmentation masks, and 3D-M 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 [79] with 8 stacks and 1 block, and *predicted* segmentation masks using DeepLab v3+ [22]. 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.

Table 4.1, Figure 4.6 and Figure 4.7 show the comparison against competitive methods. For full examination, results for 3D-M and CGAS are additionally provided 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). The WLDO method therefore achieves a new state-of-the-art on this 3D reconstruction task. In addition, the method is shown to achieve improved average IoU/PCK scores than competitive methods, even when competitors are provided ground truth annotations at test time (grey rows). Also demonstrated is the wider applicability of two contributions of this chapter (scale parameters and improved prior) in the improved performance of the 3D-M method when these are incorporated. Finally, WLDO’s test-time speed is significantly faster than competitive methods as no subsequent energy minimization procedure is required.

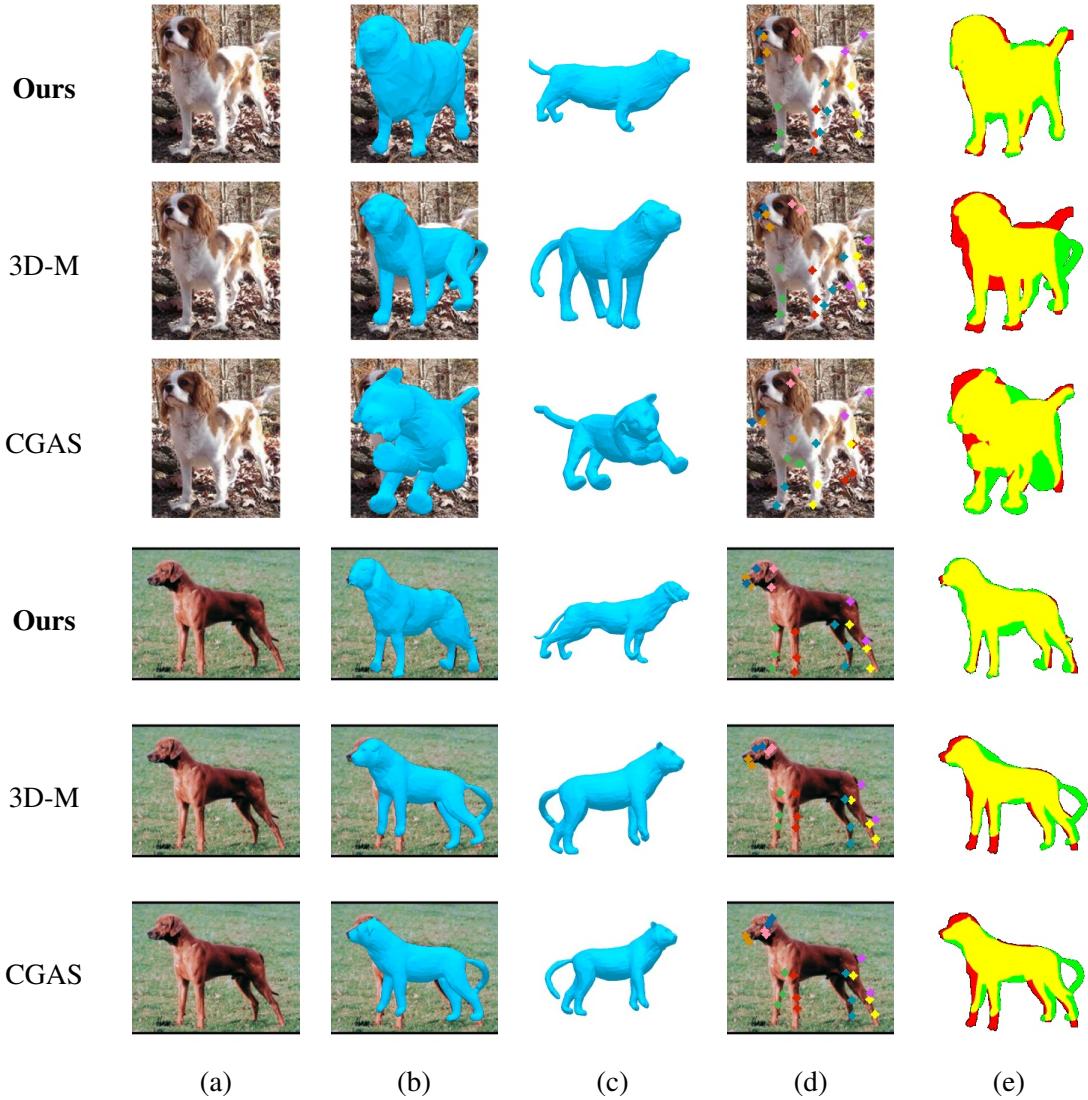


Fig. 4.6 **Qualitative comparison to SOTA.** Row 1: **Ours**, Row 2: 3D-M [143], Row 3: CGAS [9]. (a) input image, (b) predicted 3D mesh, (c) canonical view 3D mesh, (d) reprojected model joints and (e) silhouette reprojection error.

#### 4.6.3 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 3D-M [143] method (using predicted keypoints and segmentations as above for fairness) on the recent Animal Pose dataset [16]. The data preparation process is the same as for StanfordExtra and no fine-tuning was used for either method. Good results are achieved in this unseen domain and still improve over the 3D-M optimizer.

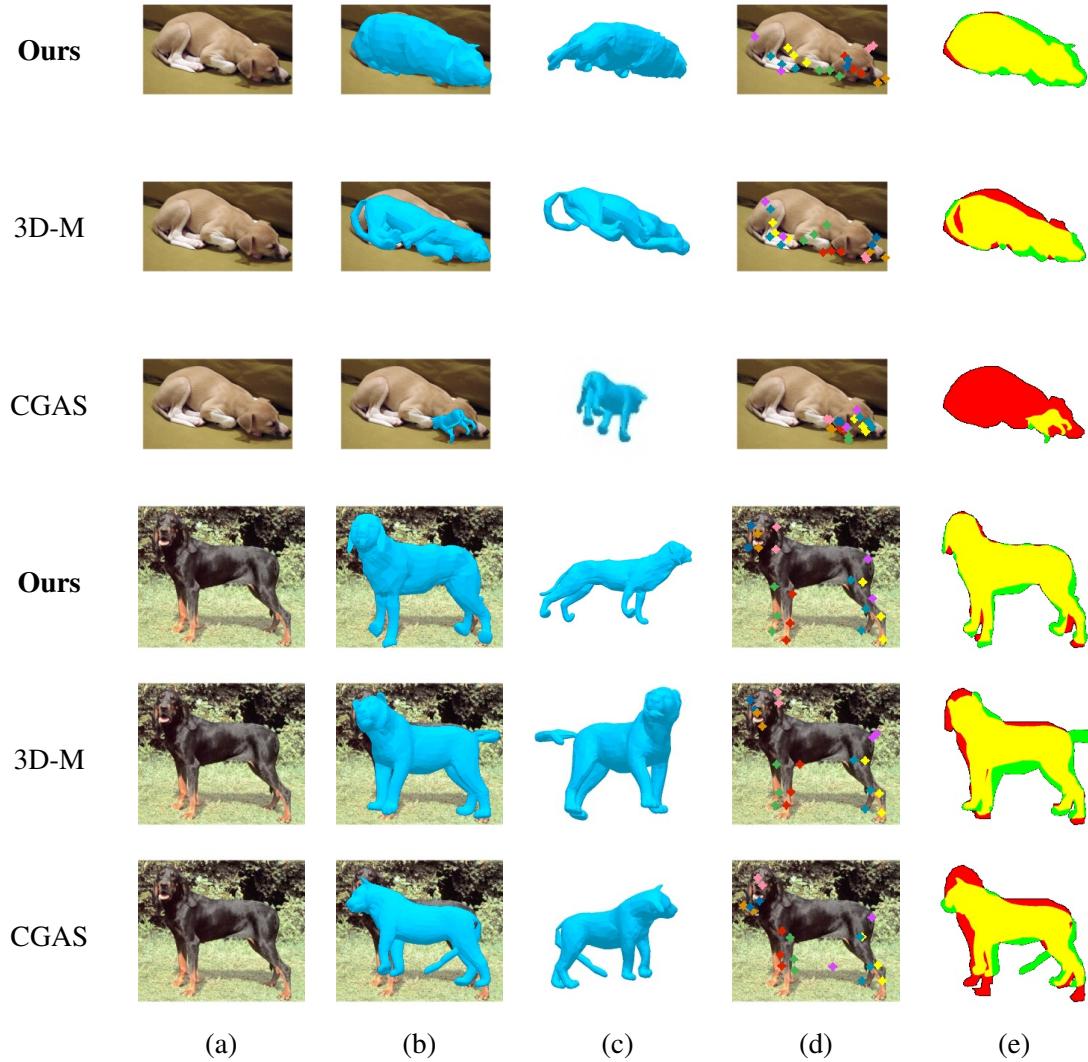


Fig. 4.7 **Qualitative comparison to SOTA (continued)**. Row 1: **Ours**, Row 2: 3D-M [143], Row 3: CGAS [9]. (a) input image, (b) predicted 3D mesh, (c) canonical view 3D mesh, (d) reprojected model joints and (e) silhouette reprojection error.

#### 4.6.4 Ablation study

Secondly, an ablation of the individual components of the WLDO method is provided in order to examine the effect of each contribution on the PCK/IoU performance. Three variants are evaluated: (1) **Ours w/o EM** that omits EM updates, (2) **Ours w/o MoG** which replaces the 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 prior refinements steps help the model accurately fit the exact dog shape. Interestingly, adding the Mixture of Gaussians prior but

Method	IoU	PCK @ 0.15			
		Avg	Legs	Tail	Ears
3D-M [143]	64.9	59.2	55.7	56.9	61.3
<b>Ours</b>	<b>67.5</b>	<b>67.6</b>	<b>60.4</b>	<b>62.7</b>	<b>86.0</b>
					<b>86.7</b>

Table 4.2 **Animal Pose dataset [16] results.** Evaluation on recent Animal Pose dataset with no fine-tuning to our method nor joint/silhouette predictors used for 3D-M.

Method	IoU	PCK @ 0.1			
		Avg	Legs	Tail	Ears
<b>Ours</b>	<b>74.2</b>	<b>63.7</b>	<b>59.5</b>	<b>48.1</b>	60.1
–EM	68.7	63.2	58.8	44.5	<b>62.6</b>
–MoG	69.0	63.1	<b>59.5</b>	40.0	60.0
–Scale	68.3	60.1	58.2	45.2	50.5
					88.3

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

omitting EM steps slightly hinders performance, most likely due to an sub-optimal initialization for the  $M$  clusters. However, adding EM updates to the Mixture of Gaussian model improves all metrics except the ear keypoint accuracy. It is observed that the error here is caused by the shape prior learning slightly imprecise shapes for dogs with extremely “floppy” ears. Although there is good silhouette coverage for these regions, the fact the SMBLD 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, the increased model flexibility afforded by the SMBLD scale parameters is shown to improve the IoU/PCK scores.

#### 4.6.5 PCK-MAX results

The final evaluation metric analyses PCK-MAX, a protocol which provides an upper bound on the PCK score by assuming invisible keypoints were successfully recovered.

#### 4.6.6 Qualitative evaluation

Figures 4.8, 4.9 and 4.10 shows a range of example system outputs when tested on range of StanfordExtra dogs with varying pose and shape and in challenging conditions. Figure 4.11 shows results on the Animal Pose dataset. Note that only StanfordExtra is used for training.

Method	Kps	Seg	PCK-MAX @ 0.1				
			Avg	Legs	Tail	Ears	Face
3D-M [143]	Pred	Pred	67.1	65.7	79.5	54.9	87.4
3D-M	GT	GT	72.6	69.9	92.0	58.6	96.9
3D-M	GT	Pred	72.6	70.2	91.5	58.1	96.9
3D-M	Pred	GT	67.4	66.0	79.9	55.0	88.2
CGAS [9]	CGAS	Pred	43.7	46.5	64.1	36.5	21.4
CGAS	CGAS	GT	43.6	46.3	64.2	36.3	21.6
3D-M + scaling	Pred	Pred	69.6	69.4	79.3	56.5	87.6
3D-M + scaling + EM prior	Pred	Pred	71.6	71.5	<b>80.7</b>	59.3	88.0
<b>Ours</b>	—	—	<b>75.7</b>	<b>75.0</b>	77.6	<b>69.9</b>	<b>90.0</b>

Table 4.4 **PCK-MAX baselines.** PCK-MAX scores are shown for SOTA methods under varying conditions. Directly comparable baseline methods (requiring only an input image) are highlighted. *Pred* keypoints generated with Hourglass-Net [79] and segmentations with DeepLab v3+ [? ]. 3D-M/CGAS are also analysed when they have access to ground-truth keypoints and/or segmentation masks. We also analyse adding this paper’s innovations (scale parameters and EM prior) to the 3D-M method [143].

Method	PCK-MAX @ 0.1				
	Avg	Legs	Tail	Ears	Face
3D-M [143]	69.1	60.9	83.5	75.0	93.0
<b>Ours</b>	<b>73.8</b>	<b>65.1</b>	<b>85.6</b>	<b>84.0</b>	<b>93.6</b>

Table 4.5 **PCK-MAX Animal Pose dataset [16].** Evaluation on recent Animal Pose dataset with no fine-tuning to our method nor joint/silhouette predictors used for 3D-M.

Method	PCK-MAX @ 0.1				
	Avg	Legs	Tail	Ears	Face
<b>Ours</b>	<b>75.7</b>	<b>75.0</b>	<b>77.6</b>	69.9	90.0
–EM	74.6	72.9	75.2	<b>72.5</b>	88.3
–MoG	74.9	74.3	73.3	70.0	<b>90.2</b>
–Scale	72.6	72.9	75.3	62.3	89.1

Table 4.6 **PCK-MAX ablation study.** Evaluation with the following contributions removed: (a) EM updates, (b) Mixture Shape Prior, (c) SMBLD scale parameters.

## 4.7 Conclusions

This paper presents an end-to-end method for automatic, monocular 3D dog reconstruction. This is achieved using only weak 2D supervision, provided by the StanfordExtra dataset which has been introduced in this chapter. Further, it has been shown that a more detailed shape prior can be learned by tuning a gaussian mixture during model training and this leads to improved reconstructions. In addition, the WLDO method improves over competitive baselines, even when they are given access to ground truth data at test time.

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 the EM formulation introduced here to handle video input to take advantage of multi-view shape constraints. Finally, an interesting direction may be to transfer knowledge accumulated through training on StanfordExtra dogs to enable accurate 3D shape reconstruction for other species.



Fig. 4.8 **Qualitative results on StanfordExtra.** 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.



**Fig. 4.9 Qualitative results on StanfordExtra (continued).** 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.

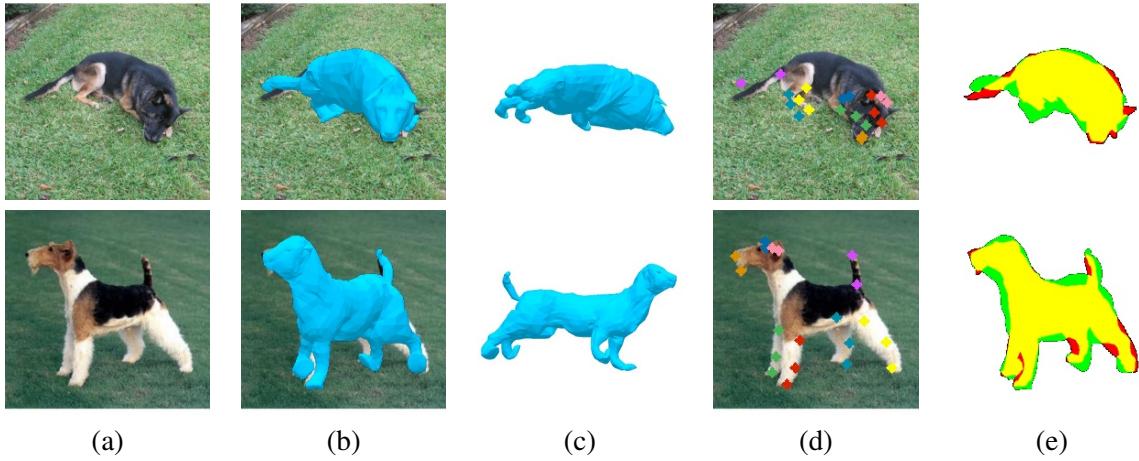


Fig. 4.10 **Qualitative results on StanfordExtra (continued).** For each sample we show: (a) input image, (b) predicted 3D mesh, (c) canonical view 3D mesh, (d) reprojected model joints and (e) silhouette reprojection error.

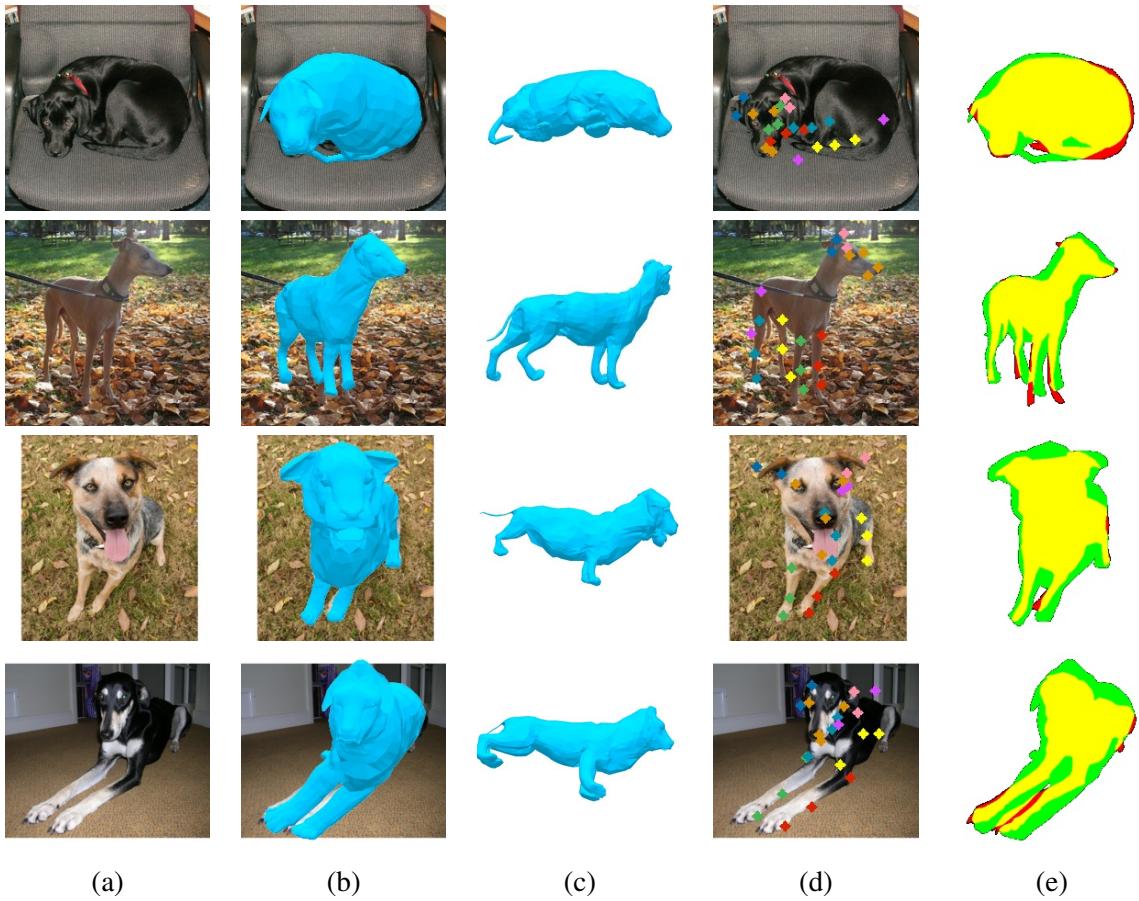


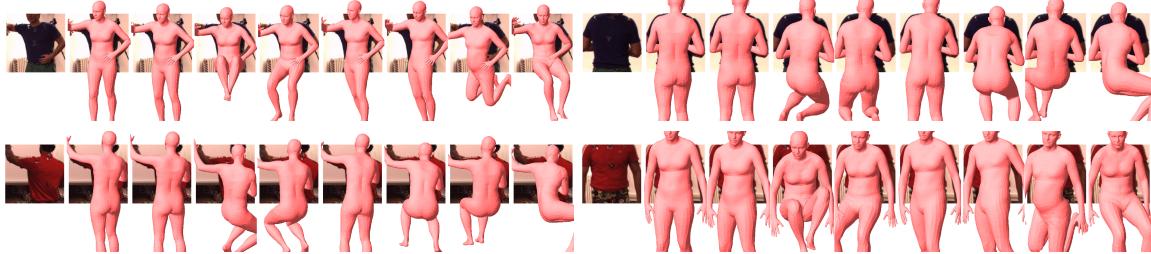
Fig. 4.11 **Qualitative results on Animal Pose [16].** For each sample we show: (a) input image, (b) predicted 3D mesh, (c) canonical view 3D mesh, (d) reprojected model joints and (e) silhouette reprojection error.

# Chapter 5

## Handling Ambiguous Input with Multi-Output Learning

### 5.1 Introduction

This chapter considers the problem of obtaining dense 3D reconstructions of articulated subjects from single and partially occluded views. Humans generally have no problem in predicting, at least approximately, the 3D structure of most scenes including the pose and shape of animals or other people, even from a single view. However, the visual evidence available in a single 2D image notoriously [30] contains insufficient information for the third dimension to be recovered uniquely. This chapter follows a growing body of work that views 3D reconstruction of articulated subjects in a probabilistic setting, and proposes that the goal of reconstruction methods should be to make the posterior distribution as sharp as possible by learning a strong prior over the space of possible solutions. In particular, the method described in this chapter recovers *several* plausible and diverse 3D reconstructions which are compatible with the input data. This is implemented using a novel multi-hypothesis neural network architecture, trained using a best-of- $M$  loss where each of the  $M$  hypotheses is constrained to lie on a manifold of plausible poses by means of a generative model. Various improvements are suggested to the standard best-of- $M$  setup, to produce reconstruction sets of an arbitrary size, ensuring these are plausible and diverse by means of a learnt normalizing flow prior and discouraging mode degeneration with a multi-hypothesis reprojection loss. A fundamental insight is that ambiguities in 3D body poses can be effectively modelled using a similar parameterization as in previous chapters, using a suitable 3D morphable model. As well as being the first system capable of modelling ambiguities in 3D animal reconstruction, the method is shown to outperform alternative approaches in ambiguous pose recovery on standard benchmarks for 3D humans and in heavily occluded versions of these benchmarks. The chapter begins by introducing key concepts relied upon in this chapter as well as related literature, followed by a description of the method, an evaluation section and finishes with conclusions and opportunities for future work.



**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.

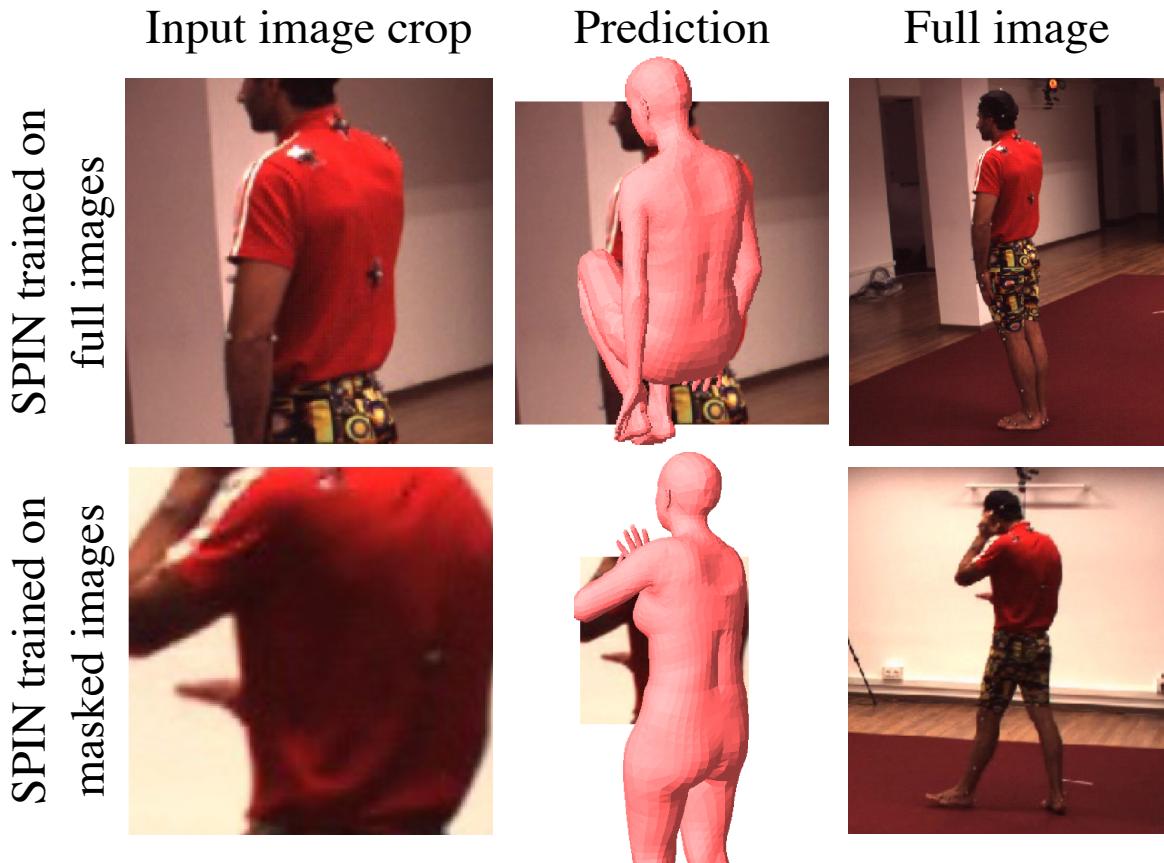
### 5.1.1 Limitations of state-of-the-art 3D reconstruction techniques

As discussed in Chapter 2, recent progress in single-view 3D model reconstruction for articulated subjects has been impressive. Methods such as 3D-Safari [8] and Chapter 4 of this thesis for animals and HMR [53], GraphCMR [61] and SPIN [60] for humans, formulate the task using a deep neural network that maps 2D images to the parameters of a 3D morphable model (usually SMAL [143] for animals and SMPL [69] for humans). Although these methods work well in general, they are not immune from failure cases (Figure 5.2). Their main weakness is when processing *heavily occluded images* of the subject. When a large part of the subject is missing, say the hind legs of an animal or lower body of a sitting human, they output reconstructions that are often implausible. Since these approaches 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. This can be understood with a simple thought experiment. Consider an agent which must randomly choose to travel left or right to avoid an oncoming lake. If the agent applies a policy to average these two possibilities, it will lead the agent to travel straight forwards, realising a suboptimal outcome of travelling straight into the lake. Unfortunately, the failure modality caused by ambiguous imagery is rather common in animal and human scenes due to environmental and self-occlusion, due to environmental clutter and crowds.

This chapter proposes a system that handles these existing limitations. The approach recovers 3D mesh reconstructions of complex articulated objects such as animals or 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, it is still possible to predict a *set* containing all possible reconstructions (see Figure 5.1). While ambiguous pose reconstruction has been previously investigated, this work proposes the first deep learning approach for ambiguous reconstructions of the *full articulated mesh*.

### 5.1.2 Modelling ambiguities in 3D reconstruction

Several previous papers have looked at the problem of modelling ambiguous 3D human pose reconstructions. Early work includes [105] who rely on known body segment lengths to resolve depth



**Fig. 5.2 Top:** Pretrained SPIN model tested on an ambiguous example, **Bottom:** SPIN model after fine-tuning to ambiguous examples. Note the network tends to regress to the mean over plausible poses, shown by predicting the missing legs vertically downward — arguably the average position over the training dataset.

ambiguities at each joint. Sidenbladh et al. [103] define a probabilistic method for tracking 3D articulated humans based on a prior probability distribution over pose that models how humans are expected to move. Sminchisescu et al. [104] describe a mixture density propagation algorithm for predicting the conditional state distribution necessary for 3D body pose tracking based on input silhouettes. More recently, [23] tackle the problem of video 3D reconstruction in the presence of occlusions, and show that temporal cues can be used to disambiguate the solution.

The following paragraphs describe implementations of common approaches used to model prediction uncertainty in deep networks. Note that the methods discussed, including Mixture Density Networks [10, 63] and Conditional Variational Auto-Encoders [99] differ from the work in this chapter as they output a distribution over reconstructed *3D locations* of a finite set of human body joints, rather than a full mesh. This chapter argues that the parameters of a 3D morphable model offers a better space for coding not just 3D shapes and poses, but also their ambiguities. In addition, predicting a full mesh is a more challenging scenario due to the increased complexity of the desired 3D shape. A

later section in this chapter describes how these techniques can be extended to act as baselines for the system proposed here.

### Mixture Density Networks (MDN)

Li et al. [63] use the Mixture Density Networks model of [10] to capture ambiguous 3D reconstructions of sparse human body keypoints (rather than a full mesh as in this work) directly in physical space.

The idea of a Mixture Density Network for articulated 3D reconstruction is to represent the probability of a 3D pose vector  $x \in \mathbb{R}^{3N}$  given an input set of features  $y \in \mathbb{R}^M$ .  $y$  is left general here, but could be for example a set of 2D keypoints, extracted image features etc.  $w$  are the neural network weights. The posterior density is given as a linear combination of  $M$  Gaussian functions

$$p(x|y, w) = \sum_i^M \Pi_i(y, w) \phi_i(x|y, w) \quad (5.1)$$

Here,  $\Pi_i(y, w)$  are mixing coefficients which must satisfy  $0 \leq \Pi_i(y) \leq 1$  and  $\sum_{i=1}^M = 1$ .  $\phi(x|y, w)$  is the conditional distribution of the 3D pose  $x$  for the  $i^{th}$  Gaussian component.

$$\phi(x|y, w) = \frac{1}{(2\pi)^{\frac{d}{2}} \sigma_i(y, w)^d} \exp -\frac{\|x - \mu_i(y, w)\|^2}{2\sigma_i(y, w)^2} \quad (5.2)$$

A dataset of  $K$  independent and identically distributed training examples including input  $X$  and 3D poses  $Y$  can be used to optimize the weights  $w$  of such a network. The posterior distribution of  $w$  is given by

$$p(w|X, Y) \propto p(X|Y, w)p(w, Y) \quad (5.3)$$

$$= p(w|Y) \prod_{j=1}^K p(x_j|y_j, w) \quad (5.4)$$

The optimal weight  $w^*$  is obtained by minimizing the negative log-posterior

$$w^* = \arg \min_w -\log p(w|X, Y) \quad (5.5)$$

$$= \arg \min_w \left[ -\sum_{j=1}^K \log p(x_j|y_j, w) \right] - \log p(w|Y) \quad (5.6)$$

This formulation can be used to form a loss function to optimize the network.

$$L = \left[ -\sum_{j=1}^K \log p(x_j|y_j, w) \right] - \log p(w|Y) \quad (5.7)$$

This can be deconstructed into two terms,  $L = L_{3D} + L_{PRIORITY}$ .  $L_{3D}$  tunes the network weights such that ground truth 3D poses are likely under the model, and  $L_{PRIORITY}$  which acts as a prior over  $\mu(w, Y)$ ,  $\sigma(w, Y)$  and  $\Pi(w, Y)$ . In the Li et al. [8] implementation, the prior is uniform over  $\mu(w, Y)$  and  $\sigma(w, Y)$ , and a Dirichlet [8] conjugate prior is used over the mixture coefficients  $\Pi(w, Y)$ .

For completeness, the 3D alignment term is given as

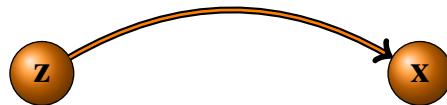
$$L_{3D} = - \sum_{j=1}^K \log \sum_{i=1}^M \Pi_i(y_j, w) \phi_i(x_j | y_j, w) \quad (5.8)$$

A version of this setup is used as a baseline in the later experiment section, although adapted to model the conditional probability of 3D morphable model joint angles through the generator function. Another notable work in this category (released after the content of this chapter) is that of Sengupta et al [] who extend the MDN formulation to model ambiguity in pose and shape, operating on a collection of human images.

### Conditional Variational Auto-Encoder (CVAE)

Another approach, proposed by Sharma et al. [99] is to learn a conditional variational auto-encoder (CVAE) to model ambiguous reconstructions as a posterior distribution.

To explain this, first consider a simple directed, latent variable model.  $x$  is the observed variable, and  $z$  is a latent variable, or *code*, which should provide information about  $x$ .



The joint distribution of such a model is given as

$$p_\theta(x, z) = p(x|z)p(z)$$

This type of latent variable model can be viewed as a generative process for obtaining the observed data  $x$ . The procedure for this is:

$$\begin{aligned} z &\sim p(z) \\ x &\sim p(x|z) \end{aligned}$$

This represents sampling a latent vector  $z$  from the prior distribution  $p(z)$  followed by sampling data  $x$  from the conditional likelihood distribution  $p(x|z)$ . The variational autoencoder comprises a probabilistic encoder  $p(z|x)$  that describes the distribution of the latent variables given data, and a probabilistic decoder  $p(x|z)$  which describes the distribution of the data given the latent variable.

Bayes theorem can be used to draw the link between these terms.

$$\begin{aligned} p(z|x) &= \frac{p(x|z)p(z)}{p(x)} \\ &= \frac{p(x|z)p(z)}{\int p(x|z')p(z')dz'} \end{aligned}$$

Unfortunately,  $p(z|x)$  is generally intractable for high-dimensional  $z$  since the integral is taken over all possible configurations of latent variables  $z$  (or  $z'$  as in the expansion above). Even estimation attempts based on Monte Carlo tend to fail due to high variance in gradient estimates.

Instead, a variational method is used to estimate the intractable posterior  $p(z|x)$  using a new tractable distribution  $q(z)$ . To do this, Kullback-Leibler (KL) divergence can be used to measure the difference between two probability distributions. For example, it can be used to measure how closely the approximation  $q(z)$  matches the true distribution  $p(z|x)$   $KL(q_\lambda(z)\|p(z|x))$ :

The goal is to find the variational parameters for  $q$  that minimize the divergence:

$$q^*(z) = \arg \min_{\lambda} KL(q(z)\|p(z|x)) \quad (5.9)$$

And looking at this term

$$\begin{aligned} KL(q(z)\|p(z|x)) &= KL(q(z)\|p(z|x)) \\ &= - \sum_{z \sim q(z)} q(z) \log \frac{p(z|x)}{q(z)} \\ &= - \sum_{z \sim q(z)} q(z) \log \frac{p(x,z)}{q(z)} \frac{1}{p(x)} \\ &= - \sum_{z \sim q(z)} q(z) \log \left[ \frac{p(x,z)}{q(z)} - \log p(x) \right] \\ &= - \sum_{z \sim q(z)} q(z) \log \frac{p(x,z)}{q(z)} + \sum_{z \sim q(z)} q(z) \log p(x) \\ &= - \sum_{z \sim q(z)} q(z) \log \frac{p(x,z)}{q(z)} + \log p(x) \underbrace{\sum_{z \sim q(z)} q(z)}_{=1} \end{aligned}$$

Notice that rearranging gives the following expression

$$\log p(x) = KL(q_\lambda(z)\|p(z|x)) + \underbrace{\sum_{z \sim q(z)} q(z) \log \frac{p(x,z)}{q(z)}}_{ELBO}$$

Recall the aim of this exercise is to minimize  $KL(q_\lambda(z)\|p(z|x))$ . Since  $x$  is given,  $\log p(x)$  is a constant and as KL is non-negative this can be achieved by maximizing the final term, known as the

Evidence Lower Bound (ELBO).

$$\begin{aligned}
ELBO &= \sum_{z \sim q(z)} q(z) \log \frac{p(x, z)}{q(z)} \\
&= \sum_{z \sim q(z)} q(z) \log \frac{p(x|z)p(z)}{q(z)} \\
&= \sum_{z \sim q(z)} q(z) \left[ \log p(x|z) + \log \frac{p(z)}{q(z)} \right] \\
&= \sum_{z \sim q(z)} q(z) \log p(x|z) + \sum_{z \sim q(z)} q(z) \log \frac{p(z)}{q(z)} \\
&= E_{z \sim q(z)} \log p(x|z) - KL(q(z) \| p(z))
\end{aligned}$$

Peering at the two terms, one can think the first term  $E_{z \sim q(z)} \log p(x|z)$  as a reconstruction ‘loss’ which promotes  $x'$ s generated from  $z'$ s being likely. The second term  $KL(q(z) \| p(z))$  is a regularizer that keeps the approximate distribution  $q$  close to  $p$ . Further, note that the tightness of the lower bound is determined by the specific choice of  $q$ . In particular, the error between the original expression  $\log p(x)$  and the ELBO is given by  $KL(q(z) \| p(z|x))$ . The error is zero if these exactly match.

The next question is how to implement such a model using a deep neural network. Here,

1. **Encoder**  $q_\lambda(z|x) = \mathcal{N}(z|\mu_\lambda(x), \Sigma_\lambda(x))$  where  $(\mu_\lambda(x), \Sigma_\lambda(x)) = \Phi_\lambda(x)$  are computed by an encoder neural network  $\Phi$ .
2. **Decoder**  $p_\theta(x|z) = \mathcal{N}(x|\mu_\theta(z), \Sigma_\theta(z))$  where  $(\mu_\theta(z), \Sigma_\theta(z)) = \Psi_\theta(z)$  are computed by a decoder neural network  $\Psi$ .
3. **Prior**  $p_\theta(z)$  is set to a simple fixed distribution (e.g. std Gaussian) to ensure  $KL(q_\lambda(z|x) \| p_\theta(z))$  is trivial to compute.
4. We express samples  $z = \mu_\theta(x) + \Sigma_\theta(x)^{-\frac{1}{2}}\varepsilon$  where  $\varepsilon$  is a sample from an std Gaussian, also known as \*reparameterization trick\*. We also use the shorthand notation  $z = \Phi(x)\square\varepsilon$  that captures the fact that  $\mu_\theta(x)$  and  $\Sigma_\theta(x)$  are estimated by the encoder  $\Phi$ .

Sharma et al. [8] adapt this formulation to model 3D skeletons and produce results on Human3.6M and HumanEva-I. They also propose two scoring methods to extract a single 3D reconstruction from the distribution which can be thought of as an alternative mechanism to the weighted quantizer scheme proposed here. There have also techniques [8] which extend these variational methods to generative adversarial netwrks. These are interesting and further reading here is available from xxx.

### Min-of- $M$

The idea of Min-of-N frameworks is to view the posterior distribution as a discrete set of  $M$  hypotheses. A neural network is trained to produce  $M$  hypotheses  $\{f^m(Y_i)\}_{m=1}^M$ , where  $Y_i$  is an input vector for

the  $i^{th}$  movie. 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 [38]. For example, with ground truth 3D points  $X_i$  the min-of- $M$  loss is given as

$$\mathcal{L}_{\text{best}}(Y; m^*) = \frac{1}{N} \sum_{i=1}^N \|X_i - f^{m_i^*}(Y_i)\|, \quad m_i^* = \operatorname{argmin}_{m=1,\dots,M} \|X_i - f^m(Y_i)\|, \quad (5.10)$$

Note that this formulation can be seen as a variant of the mixture density networks as defined previously. In particular, consider an MDN with  $M$  modes and reduce the covariance of each Gaussian to a single point. Further analysis of this model is left to the later method section.

This chapter opts for the *best-of- $M$*  approach since it has been shown 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. For example, Rupprecht et al. [94] demonstrate the model on classically ambiguous tasks including human pose estimation and future prediction. The primary contribution in this chapter is the introduction of a principled multi-hypothesis framework which models ambiguities in monocular pose recovery.

### 5.1.3 Overcoming limitations with best-of- $M$ frameworks

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.

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 of arbitrarily poor quality.

Not only does this mean that most of the hypotheses may be uninformative, but in an application it is impossible to determine *which* hypothesis should be used, leaving open the possibility of selecting 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, a first contribution in this chapter 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, another drawback of the best-of- $M$  pipeline is to be tied to a particular value of  $M$ , whereas in applications it is useful to be able to tune the number of hypotheses. In the standard best-of- $M$  implementation, this would necessitate completely retraining the network with the updated value of

$M$ . Furthermore, minimizing the reprojection loss makes hypotheses geometrically consistent with the observation, but not necessarily likely. The second contribution is therefore 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, referred to henceforth as  $n$ -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.

#### 5.1.4 Methods to learn a 3D articulated pose prior

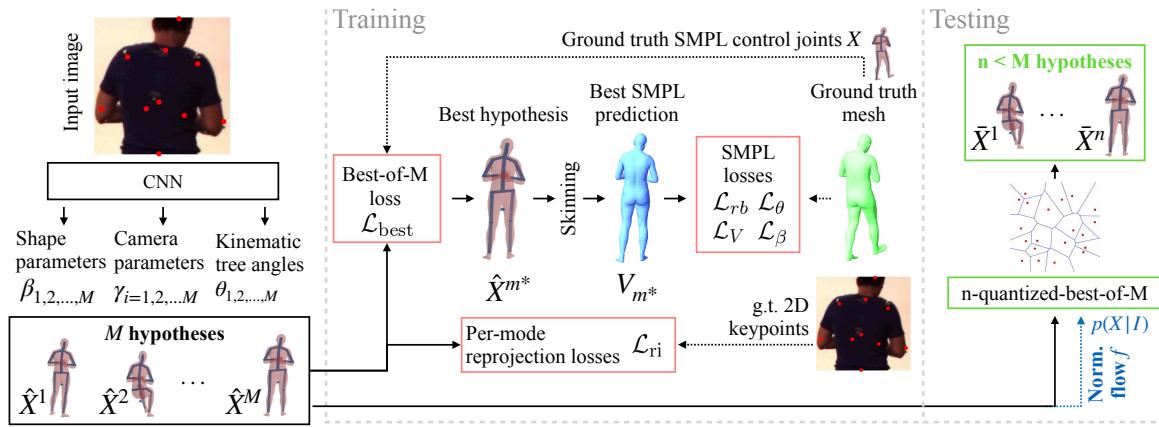
There are a number of techniques in the literature for building precise priors over human pose deformation. A common formulation, relied upon in previous chapters, is to impose a unimodal Gaussian on the 3D kinematic tree rotations. The mean and covariance parameters can either be obtained from the 3D scans used to construct the morphable model, or alternatively from a separate 3D dataset (e.g. CMU) which exhibits a wider range of motion. Alternative priors have been designed to specifically penalize “interpenetration”, a phenomenon that a body part self-intersects the mesh or a prior that specifically penalizes “out-of-range” rotations.

Recently, more expressive 3D pose priors have been designed based on modern deep learning advancements. Kanazawa et al. [8] propose a method that uses a per-part discriminator to ensure network-predicted 3D models lie on a manifold of plausible bodies. In addition, VAE methods have been proposed which constrain poses to a Gaussian parameterized by higher-level features.

Concurrently to the work in this chapter, priors have been built over 3D human pose using *normalizing flows*. In particular, [134] release a prior for their new GHUM/GHUML model, and [139] build a prior on SMPL joint angles to constrain their weakly-supervised network. The prior introduced in this chapter differs slightly as it is learnt on 3D morphable model joint locations, rather than the Rodrigues rotations.

#### 5.1.5 Sourcing datasets for training and evaluation

The method described in this chapter focuses on modelling ambiguities in 3D reconstructions using a min-of- $M$  loss. To train such a model, gradients are propagated through the best hypothesis – in this case, the closest to a 3D ground truth mesh. Consequently, a dataset of images with corresponding 3D annotations is required. As previously described, such datasets are limited in number and size for animals. Despite featuring only one animal subject and being captured in the controlled environment, the RGBD Dog dataset [8] does allow for direct evaluation on animals. However, in order to provide a fuller evaluation of the method, extensive evaluation is conducted against competitive 3D human reconstruction benchmarks.



**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  $f$ .

### 5.1.6 System overview

To summarise, the key contributions in this chapter are as follows. First, a multiple hypothesis network is introduced in order to deal with the challenge of 3D mesh reconstruction of articulated objects in *ambiguous* scenarios. Second, a mechanism referred to as *n-quantized-best-of-M* allows best-of- $M$  models to generate an arbitrary number of  $n < M$  predictions. Third, a mode-wise re-projection loss for multi-hypothesis prediction is used to ensure that predicted hypotheses are *all* consistent with the input.

Empirically, this work demonstrates the first formal study of ambiguous 3D reconstruction for an animal category. In addition, the method achieves competitive state-of-the-art monocular mesh recovery accuracy on Human36M, its more challenging version augmented with heavy occlusions, and the 3DPW datasets. The method is also ablated in a formal study to validate each of the modelling choices, demonstrating their positive effect.

## 5.2 Preliminaries

Before discussing the method, it is important to cover necessary background. The overall technique is suitable for modelling ambiguities in articulated subjects, including quadrupedal animals and humans. To demonstrate results on both object classes, separate 3D deformable models are employed. For animals, this chapter continues using the Skinned Multi-Animal Linear (SMAL) [8] model relied upon in previous chapters. For modelling humans, the popular Skinned Multi-Person Linear model (SMPL) [8] is used. Fortunately, both models are designed as drop in replacements for each other so this detail can be abstracted during description of the method. This section begins by clarifying

the abstraction, details a standard single hypothesis approach for 3D reconstruction before a detailed explanation of normalizing flows.

### 5.2.1 3D Morphable Model (SMPL/SMAL).

Both SMAL and SMPL are 3D morphable models representing quadrupedal animals and humans respectively. Each supplies a set of triangles and a function  $v(\theta, \beta) : \mathbb{R}^P \times \mathbb{R}^B \mapsto \mathbb{R}^{3 \times V}$  which outputs a set of vertex positions from pose  $\theta \in \mathbb{R}^P$  and shape  $\beta \in \mathbb{R}^{n_{shape}}$  parameters. Recall pose is determines joint angles and shape governs global body proportions (e.g. limb lengths). Parameters are also required to express the model's translation and orientation, which can be condensed into a variable  $\phi$ . A 3D model of the subject given pose, shape and position parameters is therefore given as  $\phi * v(\theta, \beta)$ . A comparison between properties of each model is given in Table XXX.

### 5.2.2 Predicting the 3D model parameters from a single image.

Given an image  $\mathbf{I}$  containing a person, the goal is to recover the 3D model parameters  $(\theta, \beta, \phi)$  that provide the best possible 3D reconstruction. Conceptually, this is an inverse problem since the image  $\mathbf{I} = \Gamma(\phi * v(\theta, \beta), \eta)$  can be thought to be generated from model parameters  $(\theta, \beta)$ , position  $\phi$  plus a number of unknown factors  $\eta$  capturing details of the appearance, background etc.

Existing algorithms [?] cast this as learning a deep network  $G(I) = (\theta, \beta, \phi, f)$  that predict the model parameters and a focal length  $f$  for a perspective camera  $\pi_f$  observing the subject. As is common practice, in this chapter a fixed set of camera parameters are used. As demonstrated in previous chapters, 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 ([37] is an exception), and since these keypoints do not necessarily correspond directly to any of the 3D mesh vertices, a mechanism is used to translate between them. This mechanism is a fixed linear regressor which determines the 3D locations of the joints. This can be viewed as  $\phi * v(\theta, \beta)K$ , where the mesh vertices are post-multiplied by a  $V \times J$  matrix  $K$ . As usual, the 3D joint positions can be compared to available 2D annotations by means of the projection function  $\pi_f$ . Model 2D annotations are therefore given as  $\pi_f(\phi * v(\theta, \beta)K)$ .

### 5.2.3 Normalizing flows.

The idea of normalizing flows 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 case, the relation between  $p(\theta)$  and  $p(z)$  is given by a change of variable

$$p(z = f(X)) = \left| \frac{df(X)}{dX} \right| p(X),$$

where, for notational simplicity, we have assumed that  $z, X \in \mathbb{R}^D$  are vectors.

The challenge is to learn  $f$  from data in a way that maintains its invertibility and smoothness. This is done by decomposing  $z = f_L \circ \dots \circ f_1(X)$  in  $n$  layers, where  $X_l = f_l(X_{l-1})$ ,  $x = X_n$  and  $X = X_0$ , and each layer is in turn smooth and invertible. Then one can write

$$\log p(z = f(X)) = \log p(X) + \sum_{l=1}^L \log \left| \frac{df_l(X_{l-1})}{dX_{l-1}} \right|.$$

Now the challenge reduces to making sure that individual layers are in fact smooth and invertible and that their inverses and Jacobian determinants are easy to compute. RealNVP [28] does so by writing each layer as  $f_l(X_{0:d,l}, X_{d:D,l-1}) = (X_{0:d,l-1}, X_{d:D,l-1} \odot e^{g_l(X_{0:d,l-1})} + h_l(X_{0:d,l-1}))$  where  $g_l, h_l : \mathbb{R}^d \rightarrow \mathbb{R}^{D-d}$  are two arbitrary neural networks.

### Worked example of normalizing flows

To assist with explanation, consider the following toy example. In this case, the data distribution is given as the multi-modal ‘moons’ dataset.

### Theoretical comparison to other methods of modelling ambiguities

TODO.

## 5.3 Method

The method starts with a neural network architecture that implements the function  $G(I) = (\theta, \beta, \phi, f)$  described above. For quadrupedal animals, the WLDO network introduced in Chapter 4 is used. For humans, as shown in SPIN [60], the HMR [53] architecture attains state-of-the-art results for this task. However, the regressors  $G(I)$  for both networks 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, this chapter focuses on predicting a *set* of plausible 3D poses rather than a single one. Therefore, the model is extended to explicitly produce a set of  $M$  different hypotheses  $G_m(I) = (\theta_m, \beta_m, \phi_m, f_m)$ ,  $m = 1, \dots, M$ . This is easily achieved by modifying the final output layer of both networks to produce a tensor  $M$  times larger, effectively stacking the hypotheses. In what follows, the learning scheme that drives the monocular predictor  $G$  to achieve an optimal coverage of the plausible poses consistent with the input image. The approach is summarized in fig. 5.3.

### 5.3.1 Learning with multiple hypotheses

For learning the model, access is assumed to a training set of  $N$  images  $\{I_i\}_{i=1,\dots,N}$ , each cropped around a subject (either an animal or human). Furthermore, for each training image  $I_i$ , it is assumed that the following are known: (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 selector  $K$  to convert ground truth 3D morphable model parameters to the 3D joints  $X_i$  and then the camera projection to obtain  $Y_i$ ). In the sequel, the following notation is used to express the 3D joint positions of the model predicted by the network's  $m$ -th SMAL/SMPL predictor  $G_m(I) = (\theta_m, \beta_m, \phi_m, f_m)$  applied to image  $I_i$ :

$$\hat{Y}^m(I_i) = \phi_m * v(\theta_m, \beta_m)K \quad (5.11)$$

### 5.3.2 Best-of- $M$ loss.

Given a single input image, the network predicts a set of poses, where at least one should be similar to the ground truth 3D annotation  $Y_i^*$ . This is captured by the best-of- $M$  loss [38]:

$$\mathcal{L}_{\text{best}}(K, G; m^*) = \frac{1}{N} \sum_{i=1}^N \|Y_i^* - \hat{Y}^{m_i^*}(I_i)\|, \quad m_i^* = \operatorname{argmin}_{m=1,\dots,M} \|Y_i^* - \hat{Y}^m(I_i)\|, \quad (5.12)$$

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.

### 5.3.3 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 it is often useful to 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.

These issues are addressed by introducing a method trains a best-of- $M$  model for a large  $M$  *once* which can be leveraged 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.

### 5.3.4 $n$ -quantized-best-of- $M$

Formally, given a set of  $M$  predictions  $\hat{\mathcal{Y}}^M(I) = \{\hat{Y}^1(I), \dots, \hat{Y}^M(I)\}$  the goal is to generate a smaller  $n$ -sized set  $\bar{\mathcal{Y}}^n(I) = \{\bar{Y}^1(I), \dots, \bar{Y}^n(I)\}$  which preserves the information contained in  $\hat{\mathcal{Y}}^M$ . In other words,  $\bar{\mathcal{Y}}^n$  optimally quantizes  $\hat{\mathcal{Y}}^M$ . To this end, the output of the best-of- $M$  model can be interpreted as a set of choices  $\hat{\mathcal{Y}}^M(I)$  for the possible pose. These poses are of course not all equally likely, but it is difficult to infer their probability from (5.12). Instead, the following approximation is used. A prior  $p(X)$  is designed that expresses the likelihood of possible poses (defined in the next section), and set:

$$p(Y|I) = p(Y|\hat{\mathcal{Y}}^M(I)) = \sum_{i=1}^M \delta(Y - \hat{Y}^i(I)) \frac{p(\hat{Y}^i(I))}{\sum_{k=1}^M p(\hat{Y}^k(I))} \quad (5.13)$$

This can be interpreted as using the best-of- $M$  output as a conditioning *set* (i.e. an unweighted selection of plausible poses) and then use the prior  $p(Y)$  to weight the samples in this set. With the weighted samples,  $K$ -means [68] further quantizes the best-of- $M$  output while minimizing the quantization energy  $E$ :

$$E(\bar{\mathcal{Y}}|\hat{\mathcal{Y}}) = \mathbb{E}_{p(Y|I)} \left[ \min_{\{\bar{Y}^1, \dots, \bar{Y}^n\}} \|Y - \bar{Y}^j\|^2 \right] = \sum_{i=1}^M \frac{p(\hat{Y}^i(I))}{\sum_{k=1}^M p(\hat{Y}^k(I))} \min_{\{\bar{Y}^1, \dots, \bar{Y}^n\}} \|\hat{Y}^i(I) - \bar{Y}^j\|^2 \quad (5.14)$$

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

### 5.3.5 Learning the pose prior with normalizing flows.

In order to obtain  $p(Y)$ , a normalizing flow model is proposed in the form of the RealNVP network  $f$  described in section 5.2 and the supplementary. RealNVP optimizes the log likelihood  $\mathcal{L}_{\text{nf}}(f)$  of training ground truth 3D skeletons  $\{Y_1, \dots, Y_N\}$  annotated in their corresponding images  $\{I_1, \dots, I_N\}$ :

$$\mathcal{L}_{\text{nf}}(f) = -\frac{1}{N} \sum_{i=1}^N \log p(Y_i) = -\frac{1}{N} \sum_{i=1}^N \left( \log \mathcal{N}(f(Y_i)) - \sum_{l=1}^L \log \left| \frac{df_l(Y_{li})}{dY_{li}} \right| \right). \quad (5.15)$$

### 5.3.6 2D re-projection loss.

Since the best-of- $M$  loss optimizes a single prediction at a time, often some members of the ensemble  $\hat{\mathcal{Y}}(I)$  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, a 2D re-projection loss acts across all hypotheses for a given image. More specifically, the set of predicted 3D reconstructions are constrained to lie on projection rays passing through the 2D input keypoints with the following *hypothesis re-projection loss*:

$$\mathcal{L}_{\text{ri}}(K, G) = \frac{1}{N} \sum_{i=1}^N \sum_{m=1}^M \|X_i^* - \pi_f(\hat{Y}^m(I))\|. \quad (5.16)$$

Note that many training images exhibit significant occlusion, so  $X^*$  often contains invisible or missing points. This is handled by masking  $\mathcal{L}_{\text{ri}}$  to prevent these points contributing to the loss.

**SMAL/SMPL loss.** The final loss terms, introduced by prior work [53, 83, 60], penalize deviations between the predicted and ground truth 3D model parameters. For this method, these are only applied



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  $\{X_1, \dots, X_N\}$ .

to the best hypothesis  $m_i^*$  found above:

$$\mathcal{L}_\theta(G; m^*) = \frac{1}{N} \sum_{i=1}^N \|\theta_i - G_{\theta, m_i^*}(I_i)\| \quad (5.17)$$

$$\mathcal{L}_V(G; m^*) = \frac{1}{N} \sum_{i=1}^N \|\phi_i * v(\theta_i, \beta_i) - G_{\phi, m_i^*}(I_i) * v(G_{(\theta, \beta), m_i^*}(I_i))\| \quad (5.18)$$

$$\mathcal{L}_\beta(G; m^*) = \frac{1}{N} \sum_{i=1}^N \|\beta_i - G_{\beta, m_i^*}(I_i)\|; \quad (5.19)$$

$$\mathcal{L}_{\text{rb}}(G; m^*) = \frac{1}{N} \sum_{i=1}^N \|X_i^* - \pi_{G_{f, m_i^*}(I_i)}(\hat{Y}^{m_i^*}(I_i))\| \quad (5.20)$$

Note here that  $\mathcal{L}_{\text{rb}}$  is used to refer to a 2D re-projection error between the best hypothesis and ground truth 2D points  $X_i^*$ . This differs from the earlier loss  $\mathcal{L}_{\text{ri}}$ , which is applied across all modes to enforce consistency to the visible *input* points. Note that it would have been possible to use eqs. (5.18) and (5.20) 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.

### 5.3.7 Mini SMAL/SMPL optimization

TODO - explain this.

### 5.3.8 Overall loss.

The model is thus trained to minimize:

$$\begin{aligned} \mathcal{L}(K, G) = & \lambda_{\text{ri}} \mathcal{L}_{\text{ri}}(K, G) + \lambda_{\text{best}} \mathcal{L}_{\text{best}}(K, G; m^*) + \lambda_\theta \mathcal{L}_\theta(K, G; m^*) \\ & + \lambda_\beta \mathcal{L}_\beta(K, G; m^*) + \lambda_V \mathcal{L}_V(K, G; m^*) + \lambda_{\text{rb}} \mathcal{L}_{\text{rb}}(K, G; m^*) \end{aligned} \quad (5.21)$$

where  $m^*$  is given in eq. (5.12) and  $\lambda_{\text{ri}}, \lambda_{\text{best}}, \lambda_\theta, \lambda_\beta, \lambda_V, \lambda_{\text{rb}}$  are weighing factors. A consistent set of loss weights are used across all experiments  $\lambda_{\text{best}} = 25.0, \lambda_\theta = 1.0, \lambda_\beta = 0.001, \lambda_V = 1.0$ , and set  $\lambda_{\text{ri}} = 1.0$ . Since the training of the normalizing flow  $f$  is independent of the rest of the model,  $f$  is trained separately by optimizing  $\mathcal{L}_{\text{nf}}$  with the weight of  $\lambda_{\text{nf}} = 1.0$ . Samples from the trained normalizing flows are shown in Figure 5.4.

Dataset	Quantization $n$		1		5		10		25	
	Metric	MPJPE	RE	MPJPE	RE	MPJPE	RE	MPJPE	RE	
H36M	HMR [53]	—	56.8	—	—	—	—	—	—	—
	GraphCMR [61]	71.9	50.1	—	—	—	—	—	—	—
	SPIN [60]	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	
	<b>Ours</b>	<b>61.5</b>	<b>41.6</b>	<b>59.8</b>	<b>42.0</b>	<b>59.2</b>	<b>42.2</b>	<b>58.2</b>	<b>42.2</b>	
3DPW	HMR [53]	—	81.3	—	—	—	—	—	—	—
	GraphCMR [61]	—	70.2	—	—	—	—	—	—	—
	SPIN [60]	96.9	<b>59.3</b>	—	—	—	—	—	—	—
	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	
	<b>Ours</b>	<b>93.8</b>	59.9	<b>82.2</b>	<b>57.1</b>	<b>79.4</b>	<b>56.6</b>	<b>75.8</b>	<b>55.6</b>	
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	
	<b>Ours</b>	<b>103.6</b>	<b>67.8</b>	<b>96.4</b>	<b>67.1</b>	<b>93.5</b>	<b>66.0</b>	<b>90.0</b>	<b>64.2</b>	
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	
	<b>Ours</b>	<b>149.6</b>	<b>78.5</b>	<b>125.6</b>	<b>74.4</b>	<b>116.7</b>	<b>73.7</b>	<b>107.8</b>	<b>72.1</b>	

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 [60, 61, 53] on Human3.6m (H36M), its ambiguous version AH36M, on 3DPW and its ambiguous version A3DPW.

## 5.4 Experiments

In this section the method is compared to several strong baselines. To start, the datasets and baselines are described followed by a detailed quantitative and qualitative evaluation.

### 5.4.1 Datasets

The method is first tested on the RGBD-Dog dataset, containing images of dogs captured with corresponding 3D annotations. Since the dataset is somewhat limited, and to show the generality and accuracy of the approach, the method is applied to competitive human 3D reconstruction benchmarks Human3.6m (**H36M**) [47, 20] and 3DPW [126]. Finally, qualitative results are demonstrated on images from MSCOCO which are challenging and varied. This section discusses each of these datasets before moving on to the evaluation protocol.

#### RGBT-Dog Dataset

RGBD-Dog is a new dataset of dogs captured with calibrated depth cameras. A 3D dog model is fit to the sequences in order to provide ground truth pose  $\theta$  and shape  $\beta$  parameters.

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	<b>57.0</b>	81.9	56.7	77.8	55.8
	✓	82.7	57.5	79.9	57.0	76.2	55.9
✓	✓	<b>82.2</b>	57.1	<b>79.4</b>	<b>56.6</b>	<b>75.8</b>	<b>55.6</b>

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

### Human3.6M

H36M is one of the largest datasets of humans annotated with 3D pose using MoCap sensors. As is common practice, the method is trained on subjects S1, S5, S6, S7 and S8, and tested on S9 and S11.

### 3D People in the Wild

3DPW is only used for evaluation and, following SPIN [61], evaluation is conducted on its test set.

#### Constructing ambiguous datasets (AH36M/A3DPW/ADOG).

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. 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.

The procedure for constructing AH36M/A3DPW starts with the full size image with a set of 2D joints and apply synthetic occlusions to the subject’s body parts by randomly hiding a subset of the 2D keypoints and re-computing a slightly padded image crop around the joints that remained visible. For each image, we randomly choose one of 4 possible strategies for hiding the keypoints: 1) Hiding arm and head keypoints; 2) legs; 3) head; 4) no keypoints hidden. The selection probabilities are  $p(1) = p(2) = p(3) = 0.3$ ,  $p(4) = 0.1$ .

#### 5.4.2 Evaluation Protocol

Our evaluation is consistent with GraphCMR [60] and SPIN [61] - metrics are reported 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”.

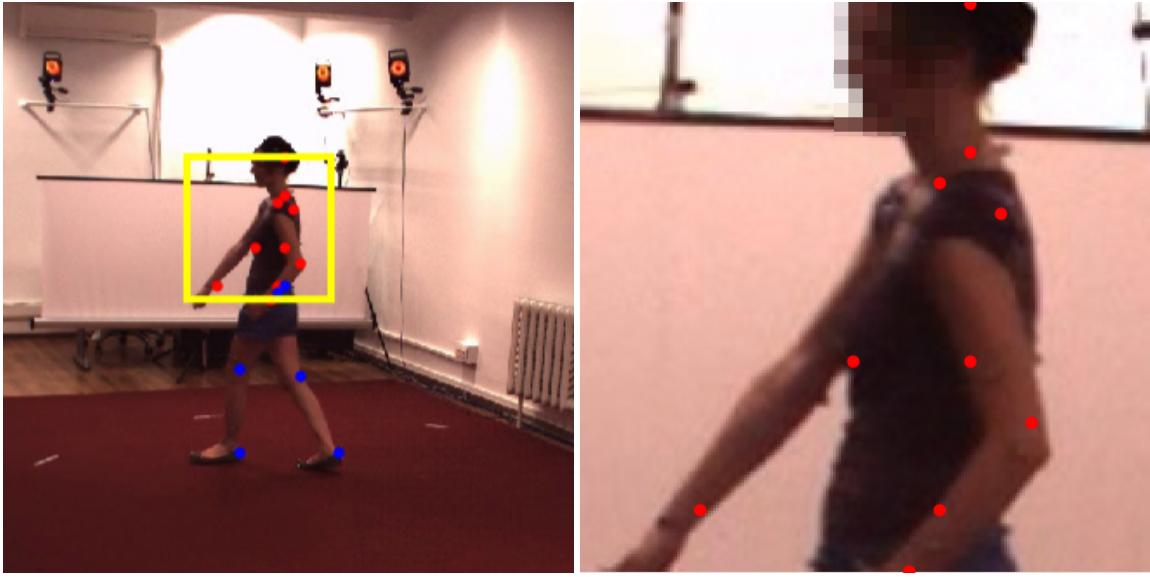


Fig. 5.5 Example image and corresponding annotation from the ambiguous H36M dataset **AH36M**. Best viewed in colour.

For each H36M test skeleton, MPJPE calculates the mean distance between 14 ground truth skeleton 3D joints and the predicted joints obtained by using a fixed linear regressor that maps the array of 6890 3D coordinates of the predicted dense mesh to the skeleton 3D joint coordinates. We report an average of all MPJPE errors measured for each test skeleton. The reconstruction error (RE) is a modification of MPJPE which consists of finding an additional rigid Procrustes alignment between the pair of assessed poses before evaluating the inter-joint distances.

#### 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 [63], 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 Multi-hypothesis 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** SMPL-MDN follows [63] 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 ( $\approx 200\text{mm MPJPE-}n = 5$  on H36M), we introduced several improvements that allow optimization of the total loss eq. (5.21).

SMPL-MDN predicts parameters of a Gaussian mixture defined over the log-rotation parameters  $\theta$  of the SMPL kinematic tree. Here,  $m$ -th Gaussian in the mixture is parametrized with a mean  $\mu_m$ , covariance matrix  $\sigma_m$  and mixture weight  $\alpha_m$ . As noted in ??, for SMPL-MDN, it was crucial to enable optimization of the total loss (5.21) in addition to optimizing the log-likelihood of the predicted Gaussian mixture. While the mixture log-likelihood optimizes directly the mixture parameters, the total loss requires a *single* prediction of  $\theta$ . In order to obtain a single estimate of  $\theta$  that can enter the total loss, similar to the Best-of- $M$  loss, we utilize the Gaussian mixture parameters and the ground truth angles  $\theta$  to generate a virtual prediction  $\hat{\theta}$  that lies close to the ground truth in the sense of the posterior probability of  $\theta$ . More specifically, the virtual  $\hat{\theta}$  is defined as a weighted combination of mixture means  $\mu_m$ , where the weights are the posterior probabilities of  $\theta$  being assigned to  $m$ -th mixture component:

$$\hat{\theta} = \sum_{m=1}^M \mu_m \frac{p(\theta|\alpha_m, \mu_m, \sigma_m)}{\sum_{n=1}^M p(\theta|\alpha_n, \mu_n, \sigma_n)} ; \quad p(\theta|\alpha_m, \mu_m, \sigma_m) = \alpha_m N(\theta|\mu_m, \sigma_m), \quad (5.22)$$

where  $\alpha_m, \sigma_m, \mu_m$  are the weight, variance and mean of the  $m$ -th mixture component respectively; and  $N(\theta|\mu_m, \sigma_m)$  is an evaluation of the multivariate normal distribution with mean  $\mu_m$  and variance  $\sigma_m$  at  $\theta$ . Note that the best quantitative results were obtained with fixing  $\forall m : \alpha_m = \frac{1}{M}, \sigma_m = 0.001$  and only allowing  $\mu_m$  to learn.

This way, the SMPL-MDN regressor  $G_{MDN}$  is altered to generate a single prediction  $G_{MDN}(I) = (\hat{\theta}, \beta, \gamma, t)$  that enters the total loss (5.21).

At test-time, following [63], the predicted hypotheses are randomly sampled per-mode predictions  $\{(\mu_m, \beta, \gamma, t)\}_{m=1}^M$  rather than random samples from the mixture. We observed that randomly sampling from the mixture density gave worse quantitative results.

**SMPL-CVAE** SMPL-CVAE, the second baseline, is a conditional variational autoencoder [108] 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- $n$ .

SMPL-CVAE consists of a pair of encoder and decoder networks. The encoder network takes as input the ground truth SMPL mesh  $V$  and outputs a Gaussian vector  $z$  whose goal is to encode the mode of ambiguity. The decoder  $G_{CVAE}(I, z) = (\theta, \beta, \gamma, t)$  then takes as input the  $z$ , together with the input image  $I$ , in order to generate the standard tuple of SMPL parameters. At train-time, the network minimizes the total loss (5.21) and a KL divergence between the predicted distribution of  $z$  and a standard multivariate normal distribution  $N(0, 1)$ .

### 5.4.5 Single hypothesis baselines

For completeness, we also compare to three more baselines that tackle the standard single-mesh prediction problem: HMR [53], GraphCMR [83], and SPIN [60], where the latter currently attain state-of-the-art performance on H36M/3DPW. All methods were trained on H36M [47], MPI-INF-3DHP [75], LSP [51], MPII [4] and COCO [64].

### 5.4.6 Training details

Our network is trained in two stages. First we train the original HMR model until convergence according to the training protocol from [61]. We then convert the model to our  $n$ -quantized-best-of- $M$  architecture and continue training until convergence with an Adam optimizer with an initial learning rate of  $10^{-5}$ . The overall training time (including the HMR pre-training step) is 5 days on a single gpu.

### 5.4.7 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 than 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 ( $\text{dim}=6890 \times 3$ ) Gaussian mixture is prohibitively costly.

Furthermore, it is very encouraging to observe that our method is also able to outperform the single-mode baselines [53, 61, 60] 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. Qualitative results are shown in fig. 5.6.

## 5.5 Results on MS-COCO

In addition to the experiments with quantitative evaluation on H36m, AH36m and 3DPW in section 5.4, in this section, we also provide qualitative results on challenging examples from the MS-COCO dataset [64].

**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. (5.14), 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.

**Performance analysis.**

A single inference pass takes on average 0.14s per image on NVIDIA V100 GPU.

## 5.6 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.



**Fig. 5.7 Qualitative results from  $n = 5$  quantization on monocular mesh recovery on AH36m and A3DPW (continued).** 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.



Fig. 5.8 **Qualitative results on monocular mesh recovery on MS-COCO** From left to right, each group of figures depicts the input image with three hypotheses of our network. Note that no 3D ground truth exists for this dataset.

# **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] 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.
- [3] American Pet Products Association (2020). *2019-2020 APPA National Pet Owners Survey*. <http://www.americanpetproducts.org>.
- [4] Andriluka, M., Pishchulin, L., Gehler, P., and Schiele, B. (2014). 2D human pose estimation: New benchmark and state of the art analysis.
- [5] 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*.
- [6] 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.
- [7] 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.
- [8] Biggs, B. (2020). Todo.
- [9] Biggs, B., Roddick, T., Fitzgibbon, A., and Cipolla, R. (2018). Creatures Great and SMAL: Recovering the shape and motion of animals from video.
- [10] Bishop, C. M. (1994). Mixture density networks. Technical report, Aston University.
- [11] 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.
- [12] Blum, H. (1967). A transformation for extracting new descriptors of shape. *Models for Perception of Speech and Visual Forms, 1967*, pages 362–380.
- [13] 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.
- [14] 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.
- [15] 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.

- [16] Cao, J., Tang, H., Fang, H., Shen, X., Tai, Y., and Lu, C. (2019a). Cross-domain adaptation for animal pose estimation. pages 9497–9506.
- [17] 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.
- [18] Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). Realtime multi-person 2D pose estimation using part affinity fields.
- [19] Cashman, T. J. and Fitzgibbon, A. W. (2013). What shape are dolphins? Building 3D morphable models from 2D images. 35(1):232–244.
- [20] Catalin Ionescu, Fuxin Li, C. S. (2011). Latent structured models for human pose estimation.
- [21] 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*.
- [22] Chen, L., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation.
- [23] Cheng, Y., Yang, B., Wang, B., Yan, W., and Tan, R. T. (2019). Occlusion-aware networks for 3d human pose estimation in video.
- [24] 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.
- [25] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. CVPR '05, page 886–893, USA. IEEE Computer Society.
- [26] 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.
- [27] Diamond, S. and Boyd, S. (2016). CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5.
- [28] Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2017). Density estimation using Real NVP.
- [29] 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.
- [30] Faugeras, O. and Luong, Q.-T. (2001). *The Geometry of Multiple Images*. MIT Press.
- [31] Felzenszwalb, P. F. and Huttenlocher, D. P. (2005). Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79.
- [32] Fischler, M. A. and Elschlager, R. A. (1973). The representation and matching of pictorial structures. *IEEE Trans. Comput.*, 22(1):67–92.
- [33] Food and Agriculture Organization of the United Nations (2016). FAOSTAT statistics database. [Online; data retrieved from FAOSTAT on 21-November-2017].
- [34] 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.

- [35] Gkioxari, G., Hariharan, B., Girshick, R., and Malik, J. (2014). Using k-poselets for detecting people and localizing their keypoints. In *CVPR*.
- [36] 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.
- [37] Güler, R. A., Neverova, N., and Kokkinos, I. (2018). Densepose: Dense human pose estimation in the wild. pages 7297–7306.
- [38] Guzman-Rivera, A., Batra, D., and Kohli, P. (2012). Multiple choice learning: Learning to produce multiple structured outputs. pages 1799–1807.
- [39] Han, X., Leung, T., Jia, Y., Sukthankar, R., and Berg, A. C. (2015). Matchnet: Unifying feature and metric learning for patch-based matching.
- [40] Hannah, M. J. (1974). Computer matching of areas in stereo images. AAI7427032.
- [41] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151.
- [42] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. pages 770–778.
- [43] Hogg, D. (1983). Model-based vision: a program to see a walking person. *Image and Vision Computing*, 1(1):5–20.
- [44] Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.
- [45] Horn, B. K. P. and Schunck, B. G. (1981). Determining optical flow. *Artifical Intelligence*, 17(1–3):185–203.
- [46] 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.
- [47] 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.
- [48] 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.
- [49] Johnson, S. and Everingham, M. (2010). Clustered pose and nonlinear appearance models for human pose estimation. pages 12.1–12.11.
- [50] Johnson, S. and Everingham, M. (2011). Learning effective human pose estimation from inaccurate annotation. In *CVPR 2011*, pages 1465–1472.
- [51] Johnson, S. and Everingham, M. (2011). Learning effective human pose estimation from inaccurate annotation.
- [52] Joo, H., Simon, T., and Sheikh, Y. (2018). Total capture: A 3D deformation model for tracking faces, hands, and bodies.

- [53] Kanazawa, A., Black, M. J., Jacobs, D. W., and Malik, J. (2018a). End-to-end recovery of human shape and pose.
- [54] 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.
- [55] Kanazawa, A., Tulsiani, S., Efros, A. A., and Malik, J. (2018b). Learning category-specific mesh reconstruction from image collections. pages 371–386.
- [56] Kato, H., Ushiku, Y., and Harada, T. (2018). Neural 3d mesh renderer.
- [57] 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.
- [58] 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*.
- [59] 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.
- [60] 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.
- [61] Kolotouros, N., Pavlakos, G., and Daniilidis, K. (2019b). Convolutional mesh regression for single-image human shape reconstruction.
- [62] 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.
- [63] Li, C. and Lee, G. H. (2019). Generating multiple hypotheses for 3d human pose estimation with mixture density network.
- [64] 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.
- [65] 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*.
- [66] Liu, C., Yuen, J., and Torralba, A. (2011). Sift flow: Dense correspondence across scenes and its applications. 33(5):978–994.
- [67] Liu, J., Kanazawa, A., Jacobs, D., and Belhumeur, P. (2012). Dog breed classification using part localization. pages 172–185. Springer.
- [68] Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.
- [69] 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*.
- [70] Loper, M. M. and Black, M. J. (2014). OpenDR: An approximate differentiable renderer. pages 154–169. Springer.

- [71] Lourakis, M. and Argyros, A. A. (2005). Is Levenberg-Marquardt the most efficient optimization algorithm for implementing bundle adjustment? pages 1526–1531.
- [72] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.
- [73] 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.
- [74] 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*.
- [75] Mehta, D., Rhodin, H., Casas, D., Fua, P., Sotnychenko, O., Xu, W., and Theobalt, C. (2017). Monocular 3d human pose estimation in the wild using improved cnn supervision. IEEE.
- [76] Moravec, H. (1980). Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical report, Carnegie Mellon University, Pittsburgh, PA.
- [77] Nevatia, R. and Binford, T. O. (1977). Description and recognition of curved objects. *Artificial Intelligence*, 8(1):77–98.
- [78] 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.
- [79] Newell, A., Yang, K., and Deng, J. (2016). Stacked hourglass networks for human pose estimation.
- [80] 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)*.
- [81] Park, J. and Boyd, S. (2017). General heuristics for nonconvex quadratically constrained quadratic programming.
- [82] 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.
- [83] Pavlakos, G., Zhu, L., Zhou, X., and Daniilidis, K. (2018). Learning to estimate 3D human pose and shape from a single color image.
- [84] 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.
- [85] 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.
- [86] 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*.
- [87] Pfister, T., Charles, J., and Zisserman, A. (2015). Flowing convnets for human pose estimation in videos. In *International Conference on Computer Vision*.

- [88] 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*.
- [89] 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.
- [90] 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.
- [91] 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.
- [92] 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.
- [93] 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.
- [94] 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.
- [95] 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.
- [96] Saito, S., Simon, T., Saragih, J., and Joo, H. (2020). Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *CVPR*.
- [97] Sanakoyeu, A., Khalidov, V., McCarthy, M. S., Vedaldi, A., and Neverova, N. (2020). Transferring dense pose to proximal animal classes. In *CVPR*.
- [98] 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.
- [99] Sharma, S., Varigonda, P. T., Bindal, P., Sharma, A., and Jain, A. (2019). Monocular 3d human pose estimation by generation and ordinal ranking.
- [100] Shelton, C. (2000). Morphable surface models. *International Journal of Computer Vision*, 38:75–91.
- [101] 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.
- [102] 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.
- [103] 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.
- [104] 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.

- [105] Sminchisescu, C. and Triggs, B. (2003). Kinematic jump processes for monocular 3d human tracking. CVPR'03, page 69–76, USA. IEEE Computer Society.
- [106] 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.
- [107] 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.
- [108] Sohn, K., Lee, H., and Yan, X. (2015). Learning structured output representation using deep conditional generative models.
- [109] Sorkine, O. and Alexa, M. (2007). As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4.
- [110] Stebbing, R. and Fitzgibbon, A. (2017). Personal communication.
- [111] 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.
- [112] Sun, K., Xiao, B., Liu, D., and Wang, J. (2019). Deep high-resolution representation learning for human pose estimation. In *CVPR 2019*.
- [113] 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.
- [114] 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.
- [115] Tan, V., Budvytis, I., and Cipolla, R. (2017). Indirect deep structured learning for 3D human body shape and pose prediction.
- [116] 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.
- [117] 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.
- [118] Tekin, B., Rozantsev, A., Lepetit, V., and Fua, P. (2016). Direct prediction of 3d body poses from motion compensated sequences. pages 991–1000.
- [119] 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.
- [120] Tola, E., Lepetit, V., and Fua, P. (2010). Daisy: An efficient dense descriptor applied to wide-baseline stereo. 32(5):815–830.
- [121] 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.

- [122] 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.
- [123] Toshev, A. and Szegedy, C. (2014). DeepPose: Human pose estimation via deep neural networks. pages 1653–1660.
- [124] Tulsiani, S. and Malik, J. (2015). Viewpoints and keypoints. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1510–1519.
- [125] 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.
- [126] 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.
- [127] Wei, S.-E., Ramakrishna, V., Kanade, T., and Sheikh, Y. (2016). Convolutional pose machines. In *CVPR*.
- [128] 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.
- [129] Wikimedia Commons (2007a). Dolphin triangle mesh.
- [130] Wikimedia Commons (2007b). Ray trace diagram.
- [131] Wilhelm, N., Vögele, A., Zsoldos, R., Licka, T., Krüger, B., and Bernard, J. (2015). Fury-explorer: visual-interactive exploration of horse motion capture data. In *Visualization and Data Analysis 2015*, volume 9397, page 93970F.
- [132] Xiang, D., Joo, H., and Sheikh, Y. (2019). Monocular total capture: Posing face, body, and hands in the wild.
- [133] Xiao, B., Wu, H., and Wei, Y. (2018). Simple baselines for human pose estimation and tracking.
- [134] 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.
- [135] 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.
- [136] Yang, Y. and Ramanan, D. (2013). Articulated human detection with flexible mixtures of parts. 35(12):2878–2890.
- [137] Yi, K. M., Trulls, E., Lepetit, V., and Fua, P. (2016). Lift: Learned invariant feature transform. pages 467–483, Cham. Springer International Publishing.
- [138] 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.
- [139] 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.

- [140] 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)*.
- [141] 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".
- [142] Zuffi, S., Kanazawa, A., and Black, M. J. (2018). Lions and tigers and bears: Capturing non-rigid, 3D, articulated shape from images.
- [143] 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.

