UNIVERSITY OF
CAMBRIDGE
Department of Engineering

Estimation of Canine Dynamics from Monocular Video

Author: Oliver Boyne

Supervisors: Michael Sutcliffe & Matthew Allen

Date: May 27, 2020

I hereby declare that, except where specifically indicated, the work submitted herein is my own original work.

Signed: O.Boyne Date: 25/05/2020

Estimation of Canine Dynamics from Monocular Video

IIB Project C-MPFS-1

Oliver Boyne, Downing College

Supervisors: Michael Sutcliffe & Matthew Allen

May 27, 2020

Technical Abstract

This project investigates a viable method to extract dog kinematics and dynamics from motion capture and from ordinary images and video. This can allow for a high-level analysis of dog gait, potentially identifying health conditions without the need for invasive and costly lab testing.

The report details the several stages involved in building such a process. The first of these is data collection. Several large datasets were collected and processed in order to provide a basis for training and evaluating the other elements of the method.

Next, an image processing system is developed, which aims to extract as much kinematic information as possible from images of dogs and stills of videos. The chosen solution was a neural network which fitted a novel parameterised 3D dog mesh, the Skinned Multi-Breed Linear Dog Model, to input images. This network and model was developed in association with Benjamin Biggs and James Charles of Division F, as part of a submission to the European Conference for Computer Vision 2020. A version of this network, named the Kinematic Deep Network, was optimised specifically for this project to better predict the pose of dogs in motion.

Lastly, in order to predict dynamics from kinematic data, an inverse dynamic model was constructed. This model used a simplification of dog anatomy, combined with rigid and viscoelastic models from literature, to produce a set of linear equations, which could be solved to provide estimates for dynamic data. This method showed encouraging accuracy when used to predict known ground reaction forces for a dog, when the kinematics were captured via motion capture - ground reaction forces were predicted within an accuracy range of 3.7%.

The work undergone in this project demonstrated two powerful components, in image processing and inverse dynamics, which individually showed promise in achieving the task at hand.

When combining the two systems, it was found that the accumulation of errors, and inherent lack of resolution in the input images, resulted in significant errors in the predicted dynamics. Nevertheless, the viability of such a method was demonstrated, and a larger training base and further development of both the Kinematic Deep Network and the inverse dynamic model could provide a powerful tool in estimating dynamics of dogs 'in the wild'.

Contents

1 Introduction

1.1 Background

This project aims to investigate and develop methods for extracting dog dynamics from video taken 'in the wild' - outside of a controlled laboratory. This will involve a critical evaluation of current methods, and the development of image processing and dynamic models capable of extracting such information.

Understanding dog kinematics (motion) and dynamics (forces) are a crucial part of diagnosing dogs with limb conditions, including arthritis and lameness. Some examples of dynamic and kinematic artefacts from these conditions include:

- Lack of joint range of motion
- Asymmetric force profiles
  - Arthritis - studies have found lower peak vertical force, and a peak later in the gait cycle, to be strong indicators of canine limbs having osteoarthritis [1].
  - Lameness - studies have identified strong correlations between the degree of lameness, and the asymmetry of vertical impulses, slope of impulses, and peak ground reaction force. [2].

There are two possible routes for investigating internal dynamics in canines: *in vivo* (on the living organism), and *ex vivo* (on tissue samples and cadaver limbs). For the purposes of identifying conditions in living canines, only *in vivo* methods are considered.

A typical *in vivo* method to identify these behaviours would involve:

- Forceplates - these exist as both static plates and treadmills [3], and are capable of measuring the pressure distribution of animal paws over time.
- Motion capture - several high frequency cameras, fixed around a recording area, identify reflective markers, and collectively construct a 3D dataset of the markers' positions over time.
- Detailed computational models - this data can then be combined into computational models generated into specialist software to estimate the dynamics and produce kinematic data.

These methods are often applied to specific regions of the dog, such as a single pelvic limb [4], in order to identify dynamics in a specific limb or joint.

These systems have several drawbacks. A primary barrier to use is cost. These systems are quite expensive and require significant training to use effectively.

Additionally, the use of such methods is limited by the willingness of dogs to participate. While not largely invasive methods, many dogs are uncooperative to walking on a treadmill or having motion capture markers stuck to them. This can result in the dog either refusing to participate, or behaving abnormally, skewing any collected data.

A third drawback is the limitations of the controlled lab conditions. These conditions are often not representative of ordinary dog behaviour, potentially leading to incorrect conclusions.

Other methods of imaging exist that are less uncomfortable for the dog than motion capture, such as high speed radiography [5], but this and similar methods are still constrained to the laboratory, and the usage of radiation for imaging over long periods of time results in its own set of problems.

This project aims to develop and investigate the potential of a novel method of estimating canine dynamics from monocular video - video from a single, static camera. If this method is viable, it will be able to address the key concerns of usability and invasiveness. This method will require an image processing system capable of extracting kinematic data from video, and an inverse dynamics system capable of predicting the dynamics from this data.

1.2 Literature review

Previous projects within CUED. This project is part of a series undertaken by IIB engineers, in collaboration with the Veterinary Department, that investigate and develop methods of evaluating the internal and external dynamics of dogs. Recent projects include:

Modelling Dog Legs for Robotic Joint Testing (2017). This project investigated the modelling of canine limbs using commercial software, in order to test the stability of joints and implants. The project also discussed the process of experimentation on cadaver canine limbs using a *KUKA 6-axis robot*. This method provided accurate moment and force results.

Robotic Testing of Canine Total Knee Replacement (2018). This project developed a robust testing method of using robots to test knee replacements in canine cadavers.

Estimating Dog Limb Forces in Motion (2019). This project investigated methods of using forceplates, treadmills and accelerometers to extract kinematic motion from dogs. The project found that accelerometers could be used to predict the total force exerted by a dog during motion. Attempts to use accelerometers outside of laboratory conditions resulted in issues with clipping of data and difficulty in keeping the accelerometers attached to dogs.

Image recognition. Image processing to identify the positions and behaviour of animals and humans from video have been in development for a long time. These systems almost exclusively rely on machine learning - training systems with large amounts of data to fit simplified models of the human or animal to the images presented.

Research in tracking human pose from images is a well established area of study, with nearly 25 years of development [6]. These methods aim to identify the general shape and pose of humans through rudimentary models, that fit body parts broadly within an image, gaining an understanding of general pose.

In recent years, systems have been developed to predict a detailed mesh of the entire body. In 2015, the Skinned Multi-Person Linear Model was developed for humans [7], which parameterised human meshes in order to easily produce a wide variety of human meshes of different size, shape and pose, with as few input parameters as possible. These models have since been adapted to quadrupeds [8], and provide the basis for the prediction of 3D dog meshes.

The methods developed for humans have gradually been developed to work for animals. This can often prove to be a much more difficult task, due to several factors, including the difficulty of collection of data, and the larger variety in shape, size and texture. Recent developments include estimating the 3D meshes of zebras from monocular images [9]. While this and similar developments show strong promise, predicting 3D models of animals from 'in the wild' footage is an incredibly complex task.

Inverse dynamics. Inverse dynamics is a method of recovering dynamics from kinematic data, used in modern musculoskeletal modelling software [10]. It is widely accepted, and utilises simple Newtonian mechanics to solve the predicted dynamics of the system.

Several papers have laid out different procedures for estimating human dynamics and gait from kinematic data [11, 12]. The method selected for this project is the weighted least squares method [12]. The method proposed in the paper is able to use kinematic data to estimate joint moments and forces accurately, without any required external dynamic information. Providing some information, such as measured ground reaction forces, improves the accuracy of this method further.

In producing a model, it is necessary to measure physical parameters governing dynamic motion. Several papers have investigated the mass distribution and inertial properties of canine limbs [13, 14], both for the purposes of computational modelling and general biomechanical research. Furthermore, several papers [15, 16] have investigated the mechanical properties of canine paws, in order to apply viscoelastic spring models to them.

While many dynamic predictive models use very physically realistic systems to estimate the forces in human bodies, some models aim to provide reasonable estimates for real data, while not necessarily being derived from the physical system. A paper proposed a model of the human body as two springs attached to a central mass [17], which models the rigid-like behaviour of the legs when in contact with the ground. This model manages to capture the behaviour of ground reaction forces reasonably accurately, with relatively little input data.

1.3 Project overview

Aims. This project has three core aims:

1. To produce a robust method for extracting canine kinematic data from monocular video.
2. To produce a robust method for estimating canine dynamic data from input kinematic data.
3. To investigate qualitative and quantitative errors for the stages in the method.

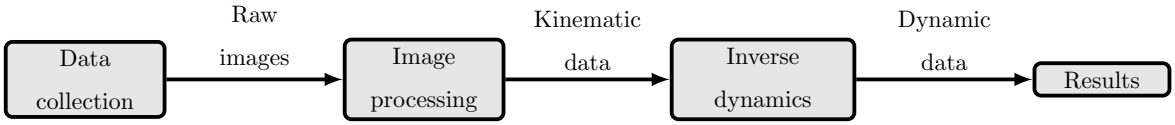


Figure 1: Overview of proposed method

Scheme of work. To satisfy these aims, the chosen method has two parts: a trained neural network capable of predicting canine kinematics from video frames; and an inverse dynamics canine model for estimating resultant dynamics. This system has a greatly reduced cost of implementation, and is far less invasive and dependent on location, and so aims to address many of the key limitations of existing methods.

The proposed scheme of work is therefore split into multiple connected sections, outlined in Figure 1:

1. **Data collection** - data must be collected for different purposes: datasets to train and develop the method; input data to be fed through the method for dynamics estimation; and ground truth data used to validate the results of the method.
2. **Image processing** - a method for converting input video footage into physically realistic and accurate kinematics.
3. **Inverse dynamics** - a method for estimating forces and torques of a canine model from kinematic data.
4. **Results** - a critical evaluation of the output data, with comparison to motion capture and ground reaction force data.

1.4 Collaboration

Cambridge Veterinary Department. Much of the data presented in this report was collected in the Vet Department, as part of work with the Surgical Discovery Centre, and Matthew Allen.

Division F. For developing the *Image Processing*, I worked with Benjamin Biggs of Division F. He has published a paper investigating the application of 3D quadruped models to images [18], which this project aims to build on.

ECCV Submission. I collaborated with Benjamin Biggs and James Charles of Division F to produce a submission to the 2020 European Conference on Computer Vision (ECCV), titled *Who left the dogs out? 3D Animal Reconstruction with Expectation Maximization in the Loop* [19]. This submission is not publicly available currently, but has been attached to this report as a supplementary document for further reading (it is not considered part of the report). The large dataset collected for the paper, included as part of this project, was funded by Roberto Cipolla of Division F.

The work outlined and figures provided in this report are entirely my own work. The exception to this is Section 3.3, which details the joint contributions to the final deep network produced as part of the ECCV submission.

1.5 Nomenclature

Dynamics. Table 1 defines the symbols to be used when describing dynamics.

Name	Symbol	Units
Position	x	m
Relative position	r	m
Velocity	v, $\dot{x}$	$\mathrm{m\,s^{-1}}$
Acceleration	a, $\ddot{x}$	$\mathrm{m\,s^{-2}}$
Time	t	s
Angle	θ	rad, $^\circ$
Angular velocity	ω	$\mathrm{rad\,s^{-1}}$
Angular acceleration	α	$\mathrm{rad\,s^{-2}}$

Name	Symbol	Units
Density	ρ	$\mathrm{kg\,m^{-3}}$
Mass	m	kg
Moment of inertia	I	$\mathrm{kg\,m^2}$
Force	F	N
Torque	τ	N m
Spring stiffness	k	$\mathrm{N\,m^{-1}}$
Spring damping	λ	$\mathrm{N\,s\,m^{-1}}$

Table 1: Terms used in dynamics and gait analysis

2 Data collection

This section outlines the data collected in order to train and evaluate the systems developed in this project. The project required four datasets:

1. A large dataset of dog images, accompanied with labelling of keypoints, and segmentations.
2. A multiview dataset, containing labelled clips of dogs in motion from multiple camera viewpoints.
3. A smaller dataset of labelled dog images, that better capture dogs in motion.
4. A dataset containing motion capture, video footage, and ground reaction forces for a single dog.

2.1 Dataset 1 - Large set of labelled and segmented dogs

The basis of training for any network tasked with identifying dog shape and positions is a large dataset of dog images, with several accompanying key features:

a. **Image metadata** - image name, width W, and height H.

b. **Bounding box** - rectangle within image containing the dog.

c. **Keypoint labels** - the pixel (x, y) coordinate for select keypoints, as well as a flag to indicate whether the keypoint is present in the image.

d. **Segmentations** - an array of size (H, W), where each value is 1 if that pixel is part of the dog itself, and 0 otherwise.

Small datasets of this form exist [20], but there is no such dataset sufficiently large (larger than 1000 images) to provide the diversity required for training the model used in this project. Therefore, a new dataset needed to be produced. This subsection details the process and results of constructing this dataset.

Much of the dataset work was completed using Amazon's *Mechanical Turk* - a marketplace for outsourcing tasks [21]. The platform allows for the uploading of a large dataset for hundreds of individuals, named *workers*, to perform repetitive tasks, such as simple operations on images or text.

To build this dataset, the Stanford Dogs dataset [22] was consulted. This dataset contains 20,580 images of dogs at various angles and positions, and has a diversity of 120 breeds. The dataset already contained the relevant image metadata and bounding boxes, so the addition of keypoint labels and segmentations was needed.

2.1.1 Keypoint labelling

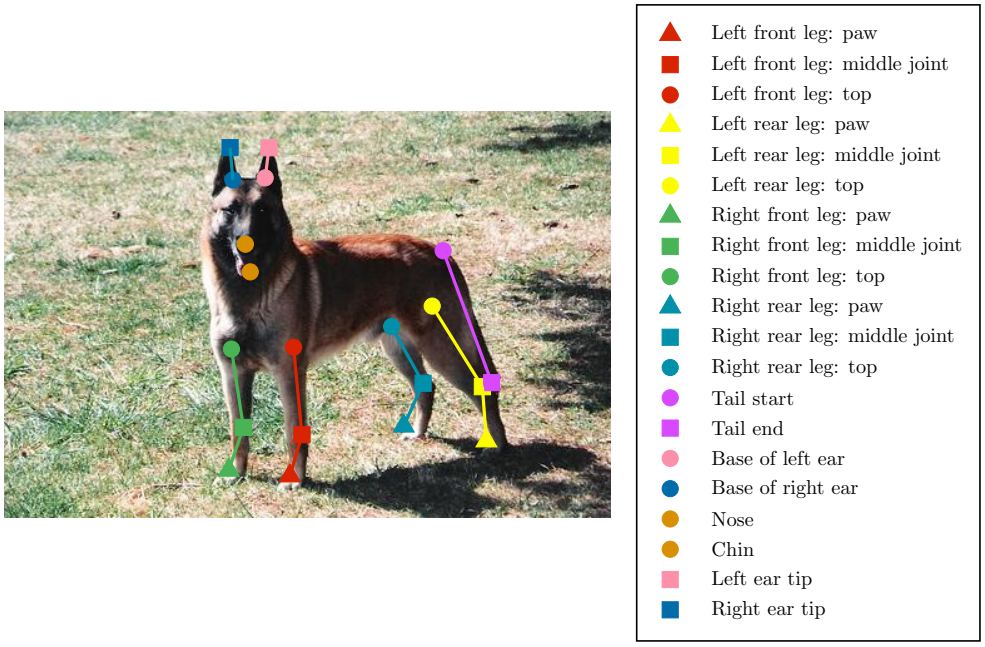


Figure 2: 20 keypoints selected for labelling on dogs

The first task given to the workers was to follow a labelling process. 15,433 images were selected from the Stanford Dogs dataset [22], and workers were instructed to identify as many of the 20 keypoints (3 per leg, 2 on the tail, 6 on the head) detailed in Figure 2 as were visible.

Each image in the dataset was sent to three separate workers to mark keypoints on, and suggested points were accepted according to the following scheme:

1. If the keypoint was highlighted by three workers, and all three points are within a certain tolerance of the mean, then accept the mean as the keypoint position. If not, reject the furthest point from the mean, and go to step 2.
2. If the keypoint was highlighted by two workers and both points are within a certain tolerance of the mean, accept the mean as the keypoint position. If not, reject the keypoint.
3. If the keypoint was highlighted by only one worker, reject the keypoint.

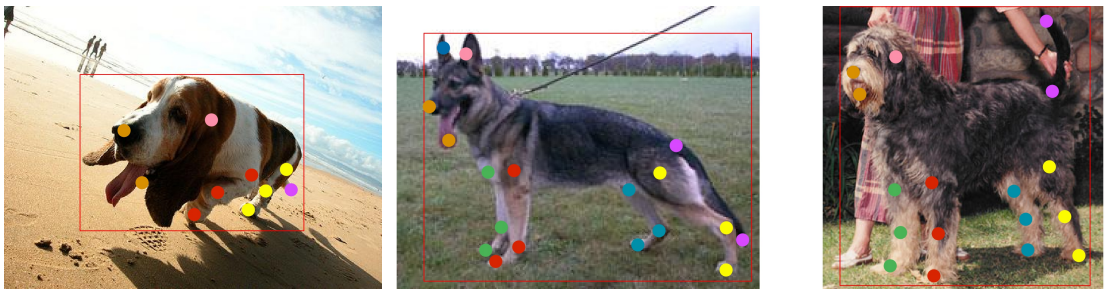


Figure 3: Some examples of dogs with keypoints and bounding box shown

Examples of the results from this labelling process can be seen in Figure 3.

2.1.2 Segmentations

Required output. For each image, a matrix is needed, detailing which pixels are part of the dog, and which are not. The matrix is fully binary (each pixel is either part of the dog or not).

Input data. For each image, 3 workers each submit a drawn segmentation of the dog. This is converted into a binary array $\mathbf{A}_w \in \mathbb{R}^{H \times W}$ for each worker w.

Filtering the data. In order to ensure high accuracy for the dataset and remove erroneous entries, a metric was defined to compare entries between workers. This was done by comparing each pixel in entries of the same image, and producing a correlation coefficient c, which gives the similarity between two segmentations w and x,

$$c_{wx} = \frac{|\mathbf{A}_w \odot \mathbf{A}_x|}{\max\left(|\mathbf{A}_w|, |\mathbf{A}_x|\right)} \tag{1}$$

Where $\odot$ denotes the element-wise product of the matrices, and $|\mathbf{A}|$ gives the sum of all elements in matrix $\mathbf{A}$. The accuracy a, of entry w was then taken as the largest of these values, $a_w = \max_{j \neq w} c_{wj}$.

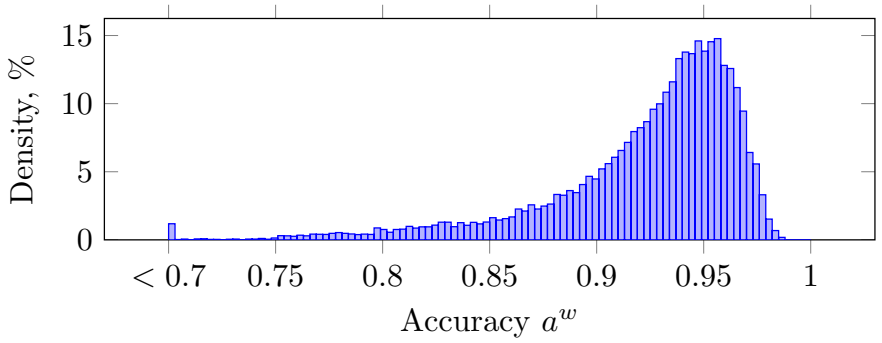


Figure 4: Distribution of accuracy metric across all worker segmentation entries

Truncated at 0.7 for simplicity

Figure 4 shows accuracy of the dataset. The set has mean accuracy $\overline{a} = 91\%$. Reviewing this distribution, it was decided to set the threshold for discarding an entry at 80%. Furthermore, any entry was rejected for which the bounding boxes were insufficiently large - to reject null entries. For this, any entry was rejected if,

$$|\mathbf{A}^w| < 0.01 \times WH \tag{2}$$

Result. The final segmentation binary mask for each image is then taken as all values appearing in the majority of entries,

$$A_{pq} = \begin{cases} 1, & \text{if } \sum_w \mathbf{A}^w_{pq} > 1.5 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

2.1.3 Producing the final dataset

20,580	Stanford dataset
- 5,147	Initial filtering based on bounding box not exceeding the image boundaries
- 5,650	Images with fewer than 8 identified keypoints
- 60	Images with insufficient segmentation accuracy
- 76	Images for which more than one keypoint is outside of the segmentation
9,647	Final dataset

Table 2: Breakdown of filtering process between the full Stanford dataset and the final, labelled dataset

In order to produce a high quality dataset, the number of entries in the dataset were reduced based on specific criteria. Table 2 details these reductions. Examples of the final labelled dataset can be seen in Figure 5.

Figure 5: Outlined segmentations and labelled keypoints for 24 representative images

Figure 6 shows the variation in worker submissions. The spread displayed about each keypoint shows the average pixel-wise deviation from the mean submission for each image, overlaid onto a single graphic. This figure demonstrates the greater uncertainty in the paws than the other leg keypoints, as well as there being a greater y directional uncertainty for the upper leg keypoints than x wise uncertainty.

Human error. The results from Figure 6 illustrate a particular shortfall of the labelled dataset - human error. As many of the workers may not be knowledgeable on canine anatomy, and are from a range of cultures and backgrounds, the instructions provided to them needed to be as clear as possible, with no inferred background knowledge of canine anatomy. Descriptions provided for the leg keypoints described the uppermost keypoint as 'where the leg

meets the body'. This in general provided a reasonable consistency, but some error could be seen for dogs with large amounts of fur, or in unusual poses, where this definition becomes less clear.

Figure 6: Heatmap of deviation of worker submitted results from mean for each submission, for all entries in the dataset

2.2 Dataset 2 - Labelled, multiview dogs in motion

The purpose of this dataset was to collect a series of videos that provided a three-dimensional view of dogs. This would allow for an evaluation of the performance of a 3D fitting process.

Figure 7: Sample data from Dataset 2. Same still of dog jumping captured from four cameras

Filming. All of the footage was filmed with a series of four synchronised cameras, in a single session at the Horse Arena in the University of Cambridge Vet School. The dogs were

all volunteered by departmental staff, and performed simple movements including walking, running, jumping, and climbing.

2.3 Dataset 3 - Supplementary labelled sets

As discussed in Section 3.1, the primary dataset, while capturing a range of breeds and static positions of dogs, has limitations when it comes to dogs in motion. This is a result of the significant bias in Dataset 1 towards dogs in 'photogenic' poses - facing the camera, with all four paws on the ground. This will result in any trained deep network preventing any fitted mesh from diverging from these biases. As a result, a supplementary dataset was produced which better captures dogs in motion.

Figure 8: Examples of labelled images from Dataset 3

This dataset was constructed from various YouTube clips, which were split up into frames and labelled in a similar process to Dataset 1. 1,553 labelled images were produced, some examples of which can be seen in Figure 8.

2.4 Dataset 4 - Kinematic and dynamic dataset of a single dog

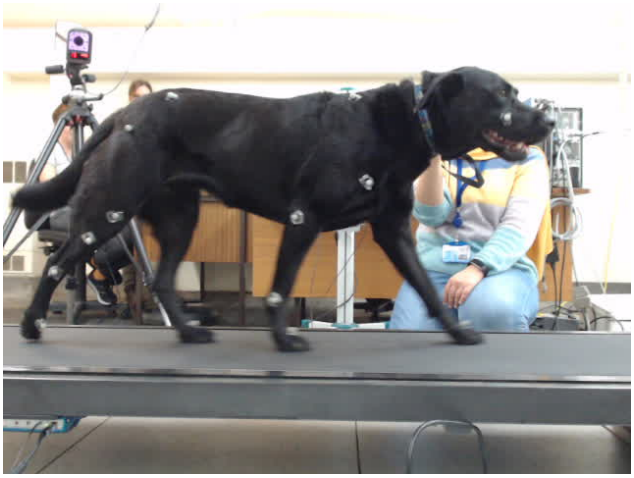


Figure 9: Labrador Ally, with motion capture markers, walking on the forceplate treadmill

To integrate the image processing and dynamics, it was necessary to collect a dataset that provided both forms of data. Using the Gait Lab in the Vet Department, a series of data was collected on Ally, a female black Labrador Retriever weighing 25.7 kg. The resulting data was in three forms:

- **Video footage** - synchronised video footage was collected from either side of the dog. This was captured using 2 Zebris cameras at 30 Hz.
- **Ground Reaction Forces** - the ground reaction forces on the treadmill were measured using a Zebris *FDM-TSTD CanidGait* - a treadmill forceplate capable of speeds from 1-12 $\mathrm{km\,h^{-1}}$, and a data capture frequency of 100 Hz. This treadmill captures pressure values for a grid of 48 $\times$ 192 individual squares of side length 8.46 mm.
- **Motion Capture** - motion capture data was collected using 25 strategic markers placed on the dog. This data was captured using 6 Qualisys *ProReflex 120* cameras, which record at 60 Hz.

Ground reaction force processing. The footplate pressure data extracted from the Zebris software is in the form of a tensor, $\mathbf{D} \in \mathbb{R}^{(fT)\times N_y \times N_x}$, where the data is sampled at frequency f for time T. N_x and N_y give the number of x and y-wise divisions provided above, 48 and 192 respectively.

This required processing, as the desired output is a value for the force at each timestep for each paw. This is expressed, for paw p, as $\mathbf{G}^p \in \mathbb{R}^{fT}$.

The first step in extracting this data is to identify individual footfalls - all pressure data corresponding to a single paw, from when it hits the plate, until it next leaves the plate. These were identified using the Python library `scipy.ndimage.label`, which is used to identify *clusters* of data in a (usually) sparse array. Applying this functionality to the tensor $\mathbf{D}$ results in N separate footfalls extracted, with the pressure data for the nth footfall denoted $\mathbf{P}^n \in \mathbb{R}^{(fT)\times N_y \times N_x}$. From this, two pieces of information are desired:

- Force over the footfall, $\mathbf{F}^n$. This is extracted from the pressure data via numerical integration over the forceplate surface, with each square in the grid having width w. At time t,
$$\mathbf{F}^n_t = \sum_{j=0}^{N_y} \sum_{i=0}^{N_x} \mathbf{P}^n_{t,i,j} \times w^2 \tag{4}$$
- The paw to which the footfall corresponds. This is recognised by comparing the footfall's mean x and y position to the other paws in contact within the same time frame. The result is a mapping $M(n) \to p$, which gives the paw p for each footfall n.

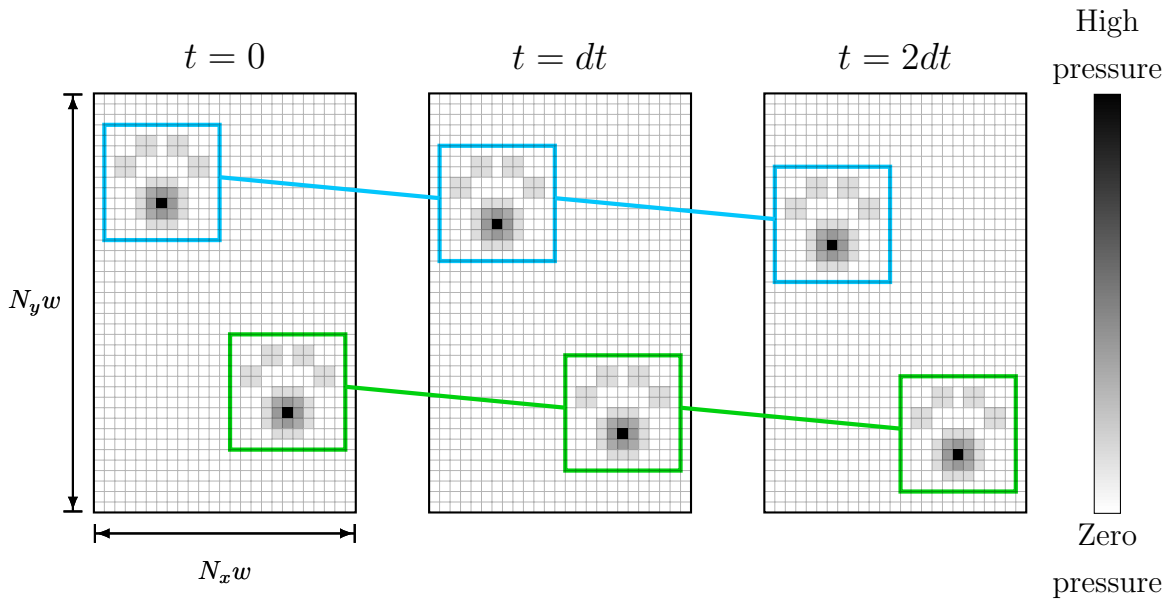


Figure 10: Representative visualisation of the pressure distribution on a forceplate for three distinct timesteps, and the paw detection system identifying the same paw over time

Finally, the ground reaction force for paw p, $\mathbf{G}^p$, can be found. For N total footfalls, all the footfalls that correspond to paw p are added together,

$$\mathbf{G}^p = \sum_{\substack{n=1 \\ M(n)=p}}^{N} \mathbf{F}^n \tag{5}$$

Motion capture processing. The motion capture data was processed using *Qualisys Track Manager* software. This software integrates with the motion capture cameras, and attempts to identify an individual marker's motion over time, by comparing adjacent frames.

Despite this, the software is not able to perfectly differentiate markers. To do this, a model can be produced in the software, linking known adjacent markers through *bones*. From this, the software learns the relative position of markers, and is able to effectively label a large series of data.

This process was applied to all of the runs collected, resulting in several clips of high precision motion capture data.

3 Image processing

Image processing is responsible for taking ordinary still images and video of dogs, and extracting from them useful and physically realistic kinematic data. In order to achieve this, several requirements must be met:

1. Demonstration of a network capable of labelling keypoints on images of dogs.
2. A parameterised mesh that can flexibly represent the variety of shapes and poses found in dogs.
3. A neural network trained to fit this parameterised mesh to a given image of a dog.
4. A modification of this network better trained for images specifically of dogs in motion.

Requirements 2 and 3 were developed as part of the ECCV submission.

3.1 2D Neural Network

As part of a preliminary process in designing a network capable of fitting to a parameterised model, it was first necessary to produce a network capable of identifying keypoints from images of dogs. This network would provide a basis for training the three-dimensional network, and would allow for the identification of any errors with the initial dataset.

Architecture and training. The network was adapted from a stacked hourglass method [23]. This is an architecture commonly used in human and animal pose estimation. It is a network composed of consecutive hourglass modules, which each contain an array of linear and convolutional layers.

The trained network was selected to have four hourglass stacks, and was trained on Dataset 1, with 1,000 images reserved for validation and the rest used for training. The training took approximately 16 hours using a Titan Xp GPU.

Figure 11: 2D neural network's attempts at labelling never before seen images of dogs

Qualitative Results. The trained network was capable of labelling a series of previously unseen images in seconds - examples of which can be seen in Figure 11.

Probability of Correct Keypoint. A performance metric was produced in order to validate the performance of the 2D keypoint identifier. This metric is based on the condition of Probability of Correct Keypoint (PCK) [24] used in evaluating human pose estimation systems. The metric identifies the percentage of candidate (predicted) keypoints that are deemed to be sufficiently accurate predictions of the ground truth (real keypoint locations). A candidate is said to be sufficiently accurate if it is within a square bounding box of the ground-truth of width W,

$$W = 2\alpha \cdot \max(w, h) \tag{6}$$

Where w and h denote the image width and height respectively, and α is a confidence parameter. Typical values of α chosen are 0.1 and 0.2, depending on the desired accuracy of the network [24].

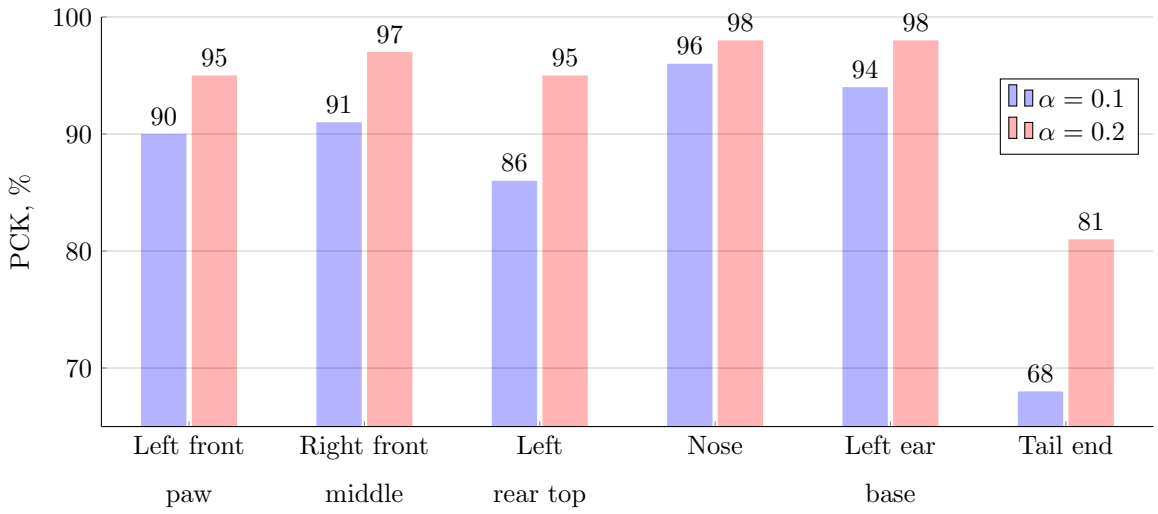


Figure 12: Probability of Correct Keypoint (PCK) metrics for a sample of keypoints, and for two values of confidence parameter $\alpha = 0.1, 0.2$

Figure 12 shows the PCK results for select keypoint locations. 'Middle' and 'top' refer to where the leg bends and meets the body respectively. These results show a reasonably high accuracy for keypoints on the head and legs - the keypoints important for dynamics predictions. The poor tail accuracy will have little bearing on dynamics performance. These results show that there is still room for accuracy improvement of the neural network.

Figure 13: An example of the 2D neural network unable to predict keypoints on a video still, as the dog is facing away from the camera

Failure cases. The neural network performs well on still images as expected, but can struggle with certain video frames. This is due to the fact that certain frames of video will involve a dog in positions unlikely to ever exist in a photograph, such that the neural network is not well trained to identify the keypoints. Figure 13 is a clear example of this.

Identifying this shortfall resulted in the collection of Dataset 3, which better captures dogs in motion. This dataset was applied to the 3D network, with the aim of introducing some novel dog positions to the training process. While not exhaustive, it should improve the performance of the deep network on trickier frames of videos.

3.2 Skinned Multi-Breed Linear Dog Model (SMBLD)

Next, a network capable of the extracting 3D dog kinematics from single images must be produced. While the 2D network has shown strong capability in identifying the pixel-wise positions keypoints for images of dogs, it suffers from two shortfalls: the network is unable to infer 3D position; and the network is not using any knowledge of typical dog anatomy to inform its fitting process - resulting in physical inaccuracy.

A 3D model, that is derived from anatomically correct canine data, is a viable solution to these shortfalls. As such, this section introduces the Skinned Multi-Breed Linear Dog model (SMBLD).

3.2.1 The SMAL Model

The SMBLD is based on the Skinned Multi-Animal Linear (SMAL) model [8]. The SMAL model depicts a mesh $\mathbf{S}$, composed of vertices, faces and joints. The model contains 3889 vertices, 35 joints, and 7774 faces.

The initial, or *T-pose*, SMAL model is defined by:

- A set of template vertices $V_0 \in \mathbb{R}^{3889\times 3}$.
- A joint regressor, a matrix $\mathbf{X} \in \mathbb{R}^{35\times 3889}$ which expresses each joint's position as a linear combination of vertex positions. The initial joint positions, J_0 are therefore given by

$$J_0 = \mathbf{X}V_0 \tag{7}$$

- A face matrix $F \in \mathbb{R}^{7774\times 3}$ assigning each triangular face to 3 vertices.

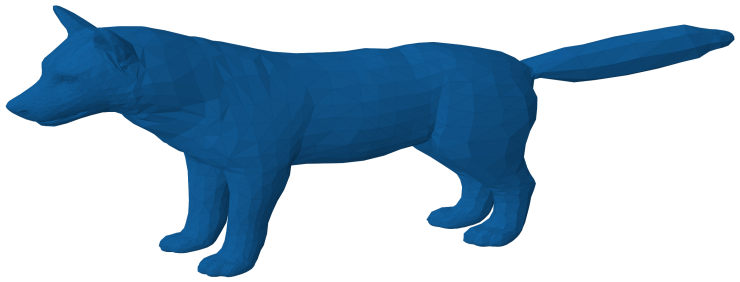


Figure 14: Mesh defined by template vertices V_0 of SMAL model

For any joint arrangement, these are then deformed to produce a set of vertices $V \in \mathbb{R}^{3889\times 3}$ and joint positions $J \in \mathbb{R}^{35\times 3}$. The T-pose model is deformed through shape parameters $\beta \in \mathbb{R}^B$, where B is a selected number of shape parameters, and joint rotation, or pose parameters $\theta \in \mathbb{R}^{35\times 3}$. These deformations are applied using a linear blend skinning function F_v, and a joint function F_J, such that

$$V = F_v(V_0, \theta, \beta), \quad J = F_J(J_0, \theta, \beta) \tag{8}$$

The mesh can then be further deformed through global rotation $R \in \mathbb{R}^{3\times 3}$, and global translation $t \in \mathbb{R}^3$, such that

$$V \coloneqq RV + t, \quad J \coloneqq RJ + t \tag{9}$$

This produces a mesh in global coordinates. For the specific purpose of fitting this to monocular images, it is also necessary to identify the mesh coordinates in a camera reference frame. This is achieved through translational parameters c_x, c_y, and c_z, and camera focal length f. These parameters, along with the SMAL vertices V, can then be fed to a Neural Renderer [25], which uses a perspective projection to calculate the pixel positions of the SMAL model in a camera reference frame.

3.2.2 The Skinned Multi-Breed Linear Dog model (SMBLD)

Motivation. The SMAL model allows for a good representation of different quadruped types. However, when varying within individual dog breeds, certain trends are not as well captured by the current model. This is to be expected, as only four distinct artist impressions of dogs were used in training the model.

Solution. The new SMBLD introduces a new set of parameters, the scale parameters, κ. This set is combined with the existing shape parameters set, β. This set aims to provide a means to scale individual body parts in meaningful ways. Allowing each joint in the model to scale independently can result in physically unrealistic shapes, so six scale parameters were selected, each linked to a set of joints and a dimension, and a modification of the parameter denotes a scaling of all joints in that dimension.

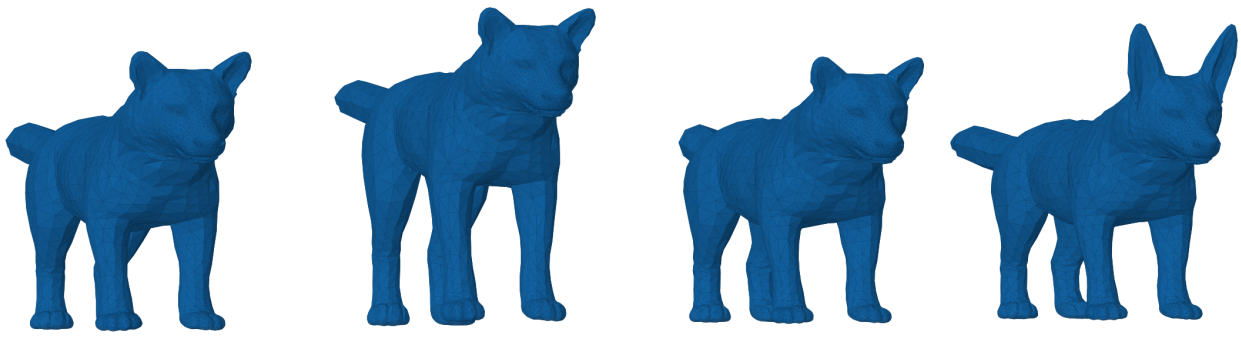


Figure 15: Effect of varying new scale parameters on the SMAL template mesh.

From left to right: mean mesh; 25% leg elongation; 50% tail shortening; 50% ear elongation

Figure 15 visualises some of the transformations made possible through the change of a single scale parameter.

Augmentation. The scale parameters are not fed directly into the SMAL network, unlike shape parameters. First, they are modified via the transformation,

$$\kappa' = e^{\kappa} \tag{10}$$

The benefits of this include:

- **Stability** - This transformation forces the sensible range of values for κ to lie in the range -1 to 1. This is of a similar order of magnitude to the regular shape parameters β. Matching these ranges allows for stability in optimising these parameters during training.
- **Positivity criterion** - This method allows the scale parameters, as with the other β, to assume negative values, representing a reduction from the mean shape in scale. If this transformation were not in place, $\kappa < 0$ would represent negative length scales in the dog, which is not physically feasible.

3.2.3 Producing a unimodal prior

Motivation. Now that a more flexible model is produced, a new unimodal shape and pose prior is required. These priors provide a mean and covariance of the pose and shape parameters, that can guide a network to produce meshes that have a shape and pose that is anatomically plausible for dogs. The SMAL model forms these priors from 3D scans of toy dogs, which lack in variety, not adequately capturing the variation of real dogs.

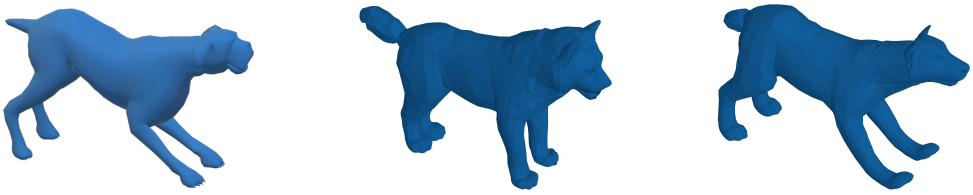
Input data. To provide a basis for training, a set of 13 3D dog models, complete with simple animation data, was purchased from the Unity asset store [26].

Training. To produce a prior, an energy minimisation scheme was used to fit the SMBLD model parameters (position ϕ, pose θ, shape β) to the shape and poses found in the Unity set. This was completed in two stages:

1. For each of the 13 dogs, the SMBLD Mesh (global rotation R and translation t, pose θ, shape β) was optimised individually to each dog. This provides a set of β parameters that vary with breed.

2. For one dog in particular, θ, R and t were optimised over 250 frames of animation, using the β parameters identified in Stage 1. This produced a set of joint rotations that can be used to train a prior.

Fitting a parameterised mesh to 3D scans is a much simpler task than to 2D images, as the desired 3D vertex positions are known, leaving less ambiguity in results. Additionally, the meshes used here as reference are relatively simple and smooth, allowing for strong fits to be found quite easily, with losses that encourage vertex alignment subject to some smoothing regularisation.



Target **Iteration 0** **Iteration 1000**

Figure 16: Visualisation of the fit process, showing the change in shape and pose over 1000 iterations of an initial SMBLD mesh to best match the target

The fitting process, depicted in Figure 16, utilises the `PyTorch3D` [27] framework, allowing for calculations of the following losses between the SMBLD mesh $\mathbf{S}$ and target meshes $\mathbf{T}$:

- **Chamfer loss** - a measure of the average minimum distance between each vertex of the SMBLD mesh and the target mesh, when p vertices are sampled from each mesh. The Chamfer loss, l_c, is calculated as

$$l_c = \frac{1}{p} \sum_{i=1}^{p} \min_j |\mathbf{S}_i - \mathbf{T}_j| \tag{11}$$

- **Edge loss** - a loss equal to the average edge length of the mesh, to encourage uniform distribution of vertices.

- **Normal loss** - this loss measures the average consistency of normals between adjacent faces in the mesh, with the aim of producing a smooth mesh. For two faces with normals $\mathbf{n_0}$ and $\mathbf{n_1}$, the normal consistency is $1 - \frac{\mathbf{n_0} \cdot \mathbf{n_1}}{|\mathbf{n_0}||\mathbf{n_1}|}$.

- **Uniform Laplacian loss** - this loss provides further mesh smoothing, while preserving key mesh features [28].

Producing the prior. Now that a variety of shape and pose parameters have been identified, a simple shape unimodal prior can be constructed. For B shape parameters, the unimodal shape prior, U, can then be expressed in terms of a set of means $\mu_U \in \mathbb{R}^B$ and covariances $\Sigma_U^2 \in \mathbb{R}^{B \times B}$. Similarly, a unimodal pose prior, P, can be expressed in terms of means $\mu_P \in \mathbb{R}^{3J}$ and covariances $\Sigma_U^2 \in \mathbb{R}^{3J \times 3J}$, to give the unimodal Gaussians,

$$U \sim \mathcal{N}(\mu_U, \Sigma_U^2), \quad P \sim \mathcal{N}(\mu_P, \Sigma_P^2) \tag{12}$$

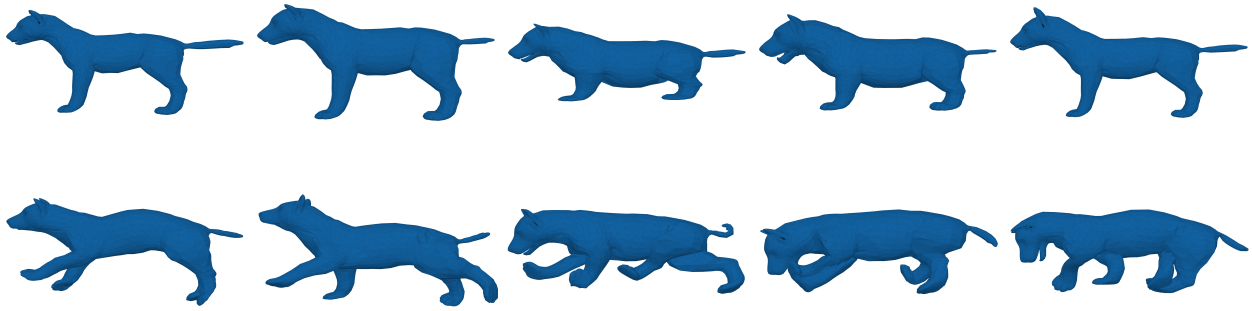


Figure 17: Variation in priors, demonstrated by 5 shape prior samples (top row), and 5 pose prior samples (bottom row).

Some random samples drawn from U and P can be seen in Figure 17.

3.3 The deep network

This section outlines the collective work of myself, Benjamin Biggs and James Charles in producing the deep network presented in our ECCV submission. All figures are my own work.

Network Architecture. The deep network architecture is inspired by recent works on estimating zebra pose, shape and texture, named the SMALST method [9]. The input image is resized, colour normalised based on typical image colour distribution, and cropped to a size of (224, 224, 3), and then applied through a Resnet-50 [29] backbone network. After passing this output through a convolutional layer and two linear layers, the output is a *feature map* vector of size 1024. This vector can then be mapped through various linear transformations to produce the SMBLD parameters.

While the SMALST method uses the output of the network to predict the individual deformations of each vertex from the standard SMAL mesh, the deep network directly predicts the set of shape and scale parameters of the SMBLD model.

Constructing a multimodal prior. The unimodal prior ensures that all produced models are physically realistic, and consistent with the 'average dog' - a dog that represents the average of all input breeds. However, this does not fully capture the reality of dog anatomy - dogs of similar breeds are anatomically similar, but different breeds have wildly different shapes. To account for this, the multimodal shape prior is introduced. This prior will result in M unimodal, Gaussian distributions, where every resultant mesh can be thought of as a linear mixture of these distributions. Each Gaussian is referred to as a cluster.

During training, each image is assigned a set of cluster weights $w \in \mathbb{R}^M$, which indicate the weightings of each individual cluster that contribute to the mesh produced for that image.

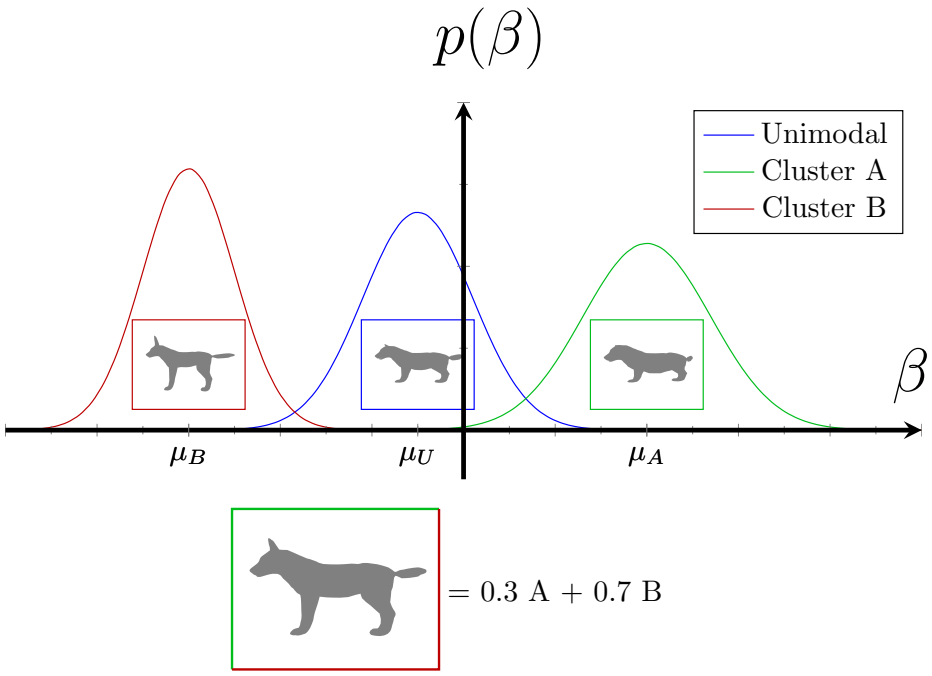


Figure 18: Visualisation of multi-modal prior in a generic β space, with 2 clusters visualised. Also visualised is a sample that shares certain characteristics from the two clusters

Imposing optimisation losses. Several losses are minimised as part of the training process:

- Joint reprojection - this loss predominantly optimises the pose of the dog. It is a measure of the Euclidean distance between the predicted 2D keypoint locations of the mesh, and the ground truth provided by the dataset.
- Silhouette loss - this loss predominantly optimises the shape of the dog. It is a measure of the Euclidean distance difference between the projected silhouette of the mesh, and the ground truth provided by the dataset.
- Prior losses - one loss for pose and one for shape, these losses measure the log likelihood of the pose or shape predicted, given the distributions of the priors provided. This prevents outputs from straying too far from the priors defined. For a distribution $U \sim \mathcal{N}(\mu, \Sigma)$,

the loss for parameter $\mathbf{x}$ is computed using the *Mahalanobis distance* D_M,

$$D_M(\mathbf{x}) = (\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu) \tag{13}$$

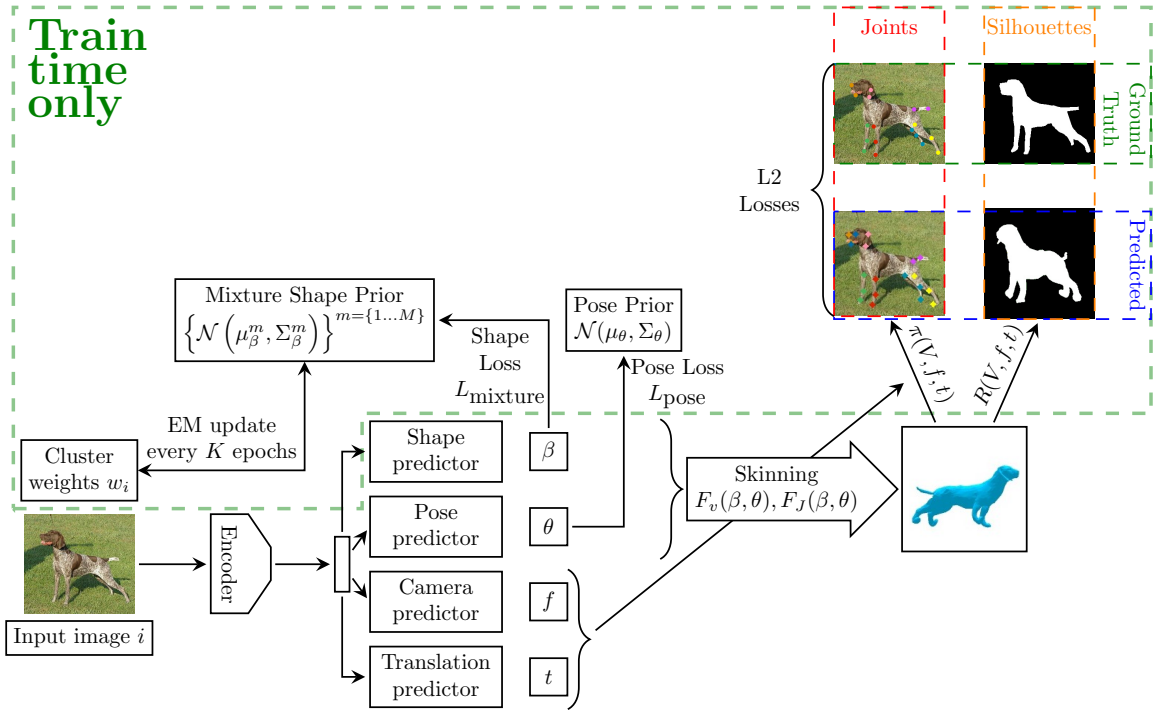


Figure 19: Overview of the training process for the deep network

Training the network. The training procedure, as outlined in Figure 19, features two stages of training:

1. Network training - The network responsible for predicting SMBLD parameters is optimised at each epoch to minimise the various losses.
2. Expectation Maximisation - Every K epochs, the multimodal prior, along with the cluster weights assigned to each image, are updated to better capture the range of shape parameters.

The initial multimodal clusters are obtained through sampling from the unimodal prior U.

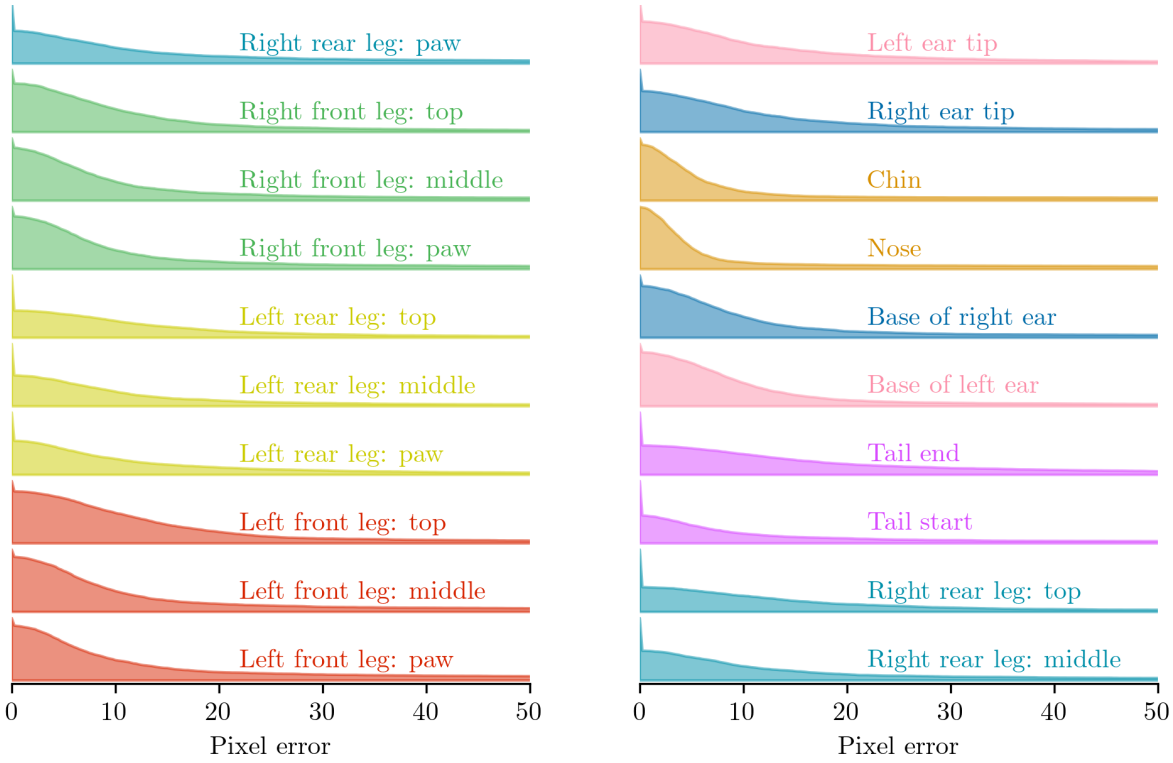


Figure 20: Reversed error cumulative distributions for each keypoint, for the deep network on the validation set of Dataset 1

Deep network results. The ECCV submission [19] provides detailed results on the performance of this network. Figure 20 shows the pixel-wise error distribution for each keypoint. From this figure, it can be seen that the network has some of the largest errors for the paws and other leg joints.

3.4 Kinematic Deep Network (KDN)

Motivation. While the deep network worked well to produce mesh fits for a variety of dog stills, it did not perform well on predicting pose for a versatile clip of a dog in motion.

Modifications. To address this weakness of the method, a separate network was produced for this project, that operated on similar principles but had some modifications to better work with dynamic dogs:

- Dataset - Dataset 1 was combined with a smaller dataset that better captured dogs in motion, Dataset 3.
- Joint loss - a weighting was applied to the joint losses to prioritise the accuracy of joints in the legs, as these are more critical for dynamic pose prediction. This was identified as an issue due to the poor paw detection performance demonstrated in Figure 20.

- Silhouette loss - the weighting of silhouette loss was reduced, as canine shape was not a priority of the network.
- Leg regularisation loss - early experimentation found that, when prioritising pose estimation, the legs would often be too thick. To correct this, a loss was imposed on the shape parameters governing the leg width.
- Temporal losses - where possible, a loss is imposed across images from the same clip (in Dataset 3), encouraging camera and shape parameters to be consistent for images from the same set.

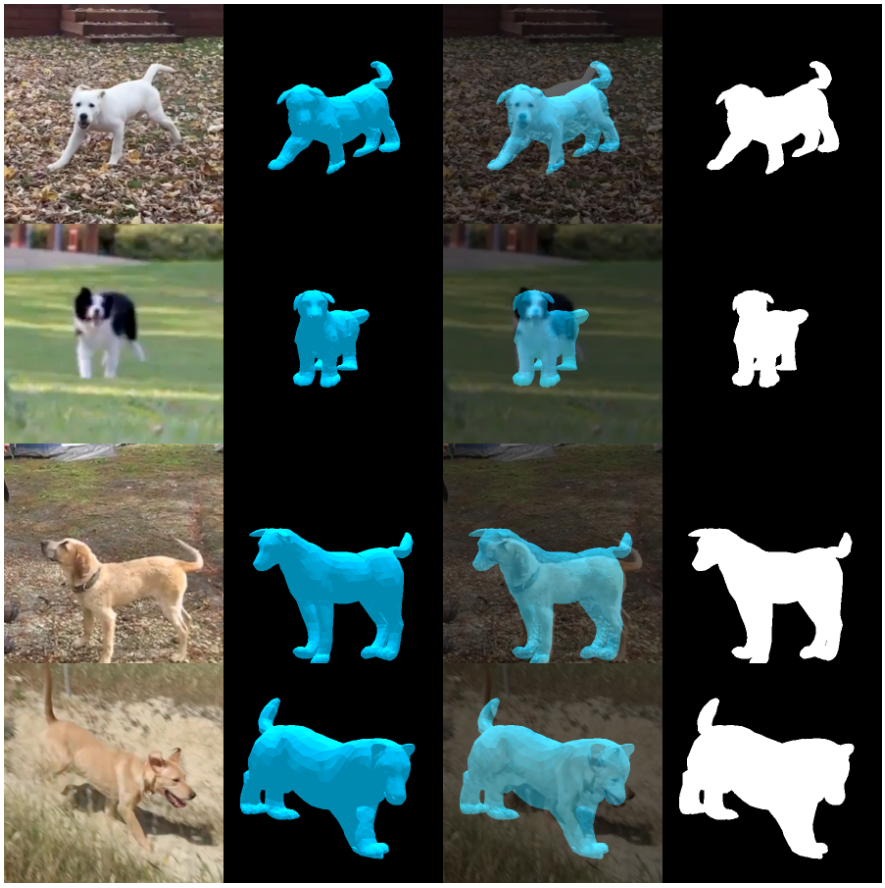


Figure 21: Some example results from the KDN.

Left to right: raw image, fitted mesh, overlaid, mesh silhouette

Training and results. This network was trained using the same method as the deep network, for 36 hours on a Titan Xp GPU. Figure 21 gives some typical results.

3.5 Post-processing

This section outlines the processing of kinematic data undertaken before attempting to estimate dynamics. This is necessary for three reasons:

- **Lack of scale** - data estimated from video footage has no associated scale, so the data must be scaled to what is physically reasonable.
- **Roughness and noise** - data carries with it a lot of noise, to which time derivatives of position are especially sensitive. It is therefore necessary to smooth data beforehand.
- **Systematic errors** - different systems can experience systematic errors due to problems with data collection or data processing. Understanding these errors can allow for them to be accounted for and corrected.

Due to the different limitations, frame rates, and resolutions of motion capture and KDN outputs, different post-processing approaches are required.

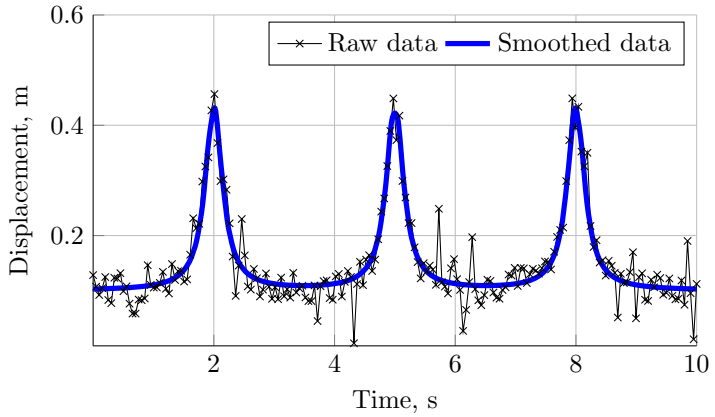


Figure 22: Visualisation of noisy raw data, and smoothed result

Smoothing. Smoothing is necessary in order to reduce the effect of noise on the results, and to provide stability for the calculation of derivatives. For smoothing, a Savitzky-Golay filter was selected. This filter uses convolution to fit collections of adjacent data points, called *windows*, with polynomials of a fixed order, according to a least squares algorithm [30]. For motion capture data, smoothing was selected with a window length of 0.5 seconds, and a polynomial order of 5. For video data, a polynomial order of 3 was used.

Eliminating motion capture paw rotations. Near the end of each footfall, the vertical motion capture position data showed a vertical drop in the paws. This is due to the marker being placed on top of the paw, as that is a location from which the marker can be seen from as many cameras as possible. This results in the drop seen as the paws rotate at the end of each footfall. This artefact must be removed through linear interpolation from the finished computation, as it does not accurately represent the vertical position of the paw within the inverse dynamics model.

SMAL post processing. Before considering the kinematic data itself, first a level of processing is applied to the extracted SMBLD parameters. A Savitzky-Golay filter is applied to smooth the joint rotations θ, with a window length of 1 second, and a polynomial order of 3. Additionally, the shape and camera parameters are averaged over the whole sequence, as they are not expected to change.

4 Inverse dynamics

4.1 The model

Overview. The model detailed in this section aims to provide a good approximation of dynamics from kinematics, whilst containing a small number of parameters in order to increase usability and adaptability.

In the model, the bones are treated as rigid coaxial cylinders, connected by pin joints. These coaxial cylinders have an inner radius comprised of a bone-like material, and an outer radius comprised of a muscle-like material.

Notation. The vector to be solved, f, contains the forces and torques on all of the pin joints in the system. F_{ij} denotes the force on the ith joint in the jth dimension. T_i denotes the torque on the ith joint. The torque is defined as acting in a direction perpendicular to both connecting bones.

A series of linear equations can then be set up that relate these forces and torques. For M joints, and N equations, This is depicted by system

$$\mathbf{WAf} = \mathbf{Wb}, \tag{14}$$

where:

- $\mathbf{f} \in \mathbb{R}^{4M} = [F_{11}, F_{12}, F_{13}, F_{21}, ..., T_1, T_2, ...]$.
- $\mathbf{A} \in \mathbb{R}^{N \times 4M}$ is a sparse matrix with the coefficients of the forces and torques for each equation. As most equations only concern two joints, most rows of $\mathbf{A}$ will have 8 or fewer non-zero values.
- $\mathbf{b} \in \mathbb{R}^N = [b_1, b_2, ..., b_N]$ gives the expected value of each of the N equations.
- $\mathbf{W} \in \mathbb{R}^{N \times N}$ is a diagonalised matrix, in which the values correspond to a weighting for each equation. This allows for certain equations to be given more precedence in the least squares problem. The selection process for these weightings is discussed in Section 4.3.3.

4.1.1 Joints

In this model, all joints are assumed to behave as *hinge* joints, in that they are only capable of carrying torque in a single direction. This assumption is anatomically accurate for the elbow/knee joint, which behaves as a snapping hinge joint [31].

However, the shoulder joints are spheroid joints, which allow movement in all directions. Despite this, the simplification will be made, primarily for the significant performance improvement achieved by reducing the number of unknowns per joint from six to four. This

does mean that, apart from the elbow/knee joints, the accuracy of the solver for internal torques may be limited.

4.1.2 Bones

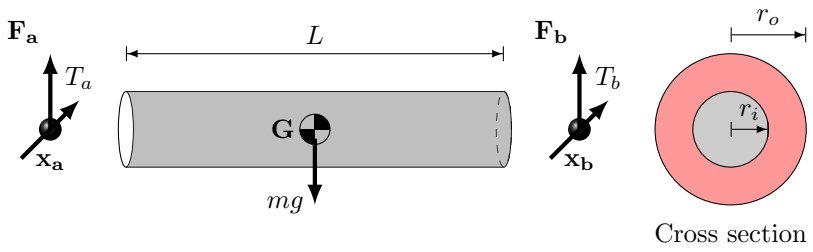


Figure 23: Forces and torques acting on a bone model

Definition. For each bone, a series of equations are known, both inertial and rotational. Let $\mathbf{x_i}$ indicate the position vector of joint i, and the two bone joints be a and b. Figure 23 visualises the bone model.

Inertial equilibrium. The inertial forces are governed by Newton's second law,

$$\sum_{i\in\{a,b\}} \mathbf{F_i} = m\ddot{\mathbf{x}}_\mathbf{G} \tag{15}$$

Rotational equilibrium. The rotational balance of forces and torques is given by the equation

$$\mathbf{T}_{\text{net}} = \mathbf{I}\boldsymbol{\alpha} = \sum_{i\in\{a,b\}} \left[(\mathbf{x_i} - \mathbf{x_G}) \times \mathbf{F_i} + \mathbf{T_i}\right] \tag{16}$$

Where the moment of inertia matrix $\mathbf{I}$ for a coaxial cylinder, in a bone reference frame,

$$\begin{aligned} \mathbf{I} &= \mathbf{I_i} + \mathbf{I_o} \\ &= \frac{\pi r_i^2 L \rho_i}{4}\begin{bmatrix} \frac{L^2}{3}+r_i^2 & 0 & 0 \\ 0 & \frac{L^2}{3}+r_i^2 & 0 \\ 0 & 0 & 2r_i^2 \end{bmatrix} + \frac{\pi(r_o^2-r_i^2)L\rho_o}{4}\begin{bmatrix} \frac{L^2}{3}+r_o^2+r_i^2 & 0 & 0 \\ 0 & \frac{L^2}{3}+r_o^2+r_i^2 & 0 \\ 0 & 0 & 2(r_o^2+r_i^2) \end{bmatrix} \end{aligned} \tag{17}$$

4.1.3 The body

Definition. To better model the behaviour of the dog's spine, the body is modelled differently. It is composed of non-coaxial cylinders, with off-axis displacement of d of the 'muscle' from the bone. Note that the terms 'joint' and 'bone' here are being used as modelling approximations - the spine is not, in reality, a single bone.

In the model, the body is connected to leg bones by 4 joints, denoted $\{a, b, c, d\}$.

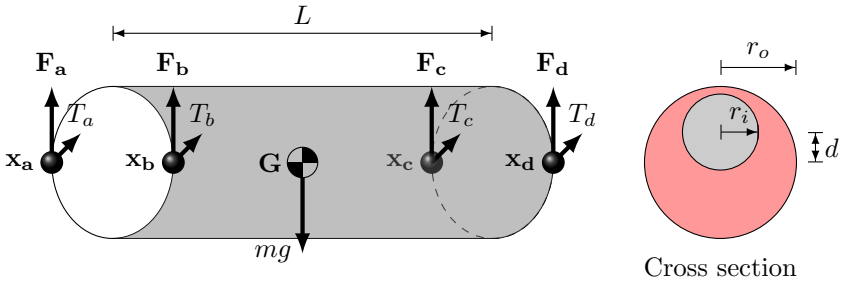


Figure 24: Forces and torques acting on the body model

Equilibrium. The equilibrium equations are identical to those defined in Equation 15 and 16. The only difference is the offset d results in a shift of the centre of gravity, and modifies the moment of inertia matrix by a factor $md^2\widehat{\mathbf{d}}$, where $\widehat{\mathbf{d}}$ is the unit vector pointing in the direction of the displacement.

4.1.4 The paws

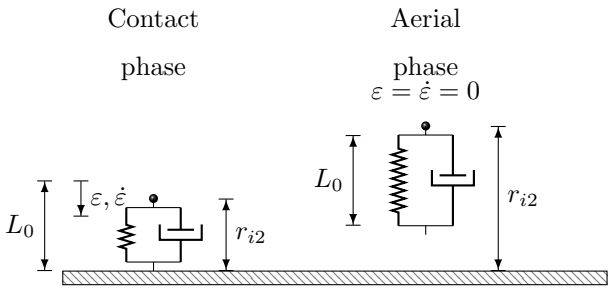


Figure 25: Paw pad linear spring model

The paws are modelled as a simple spring and damper in parallel, with values described in Section 4.2.2. This model is relatively simple, but allows for both restoring and viscoelastic effects seen in real dog paws.

Each paw i is defined as having a spring stiffness k_i, a damping factor λ_i, and a compression ε, which can only be positive, i.e.

$$\varepsilon_i = \begin{cases} L_0 - r_{i2}, & r_{i2} \leq L_0 \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

The vertical force produced by each paw is then calculated via

$$F_{i2} = k_i\varepsilon + \lambda_i\dot{\varepsilon} \tag{19}$$

4.1.5 The leg-spring model

The paw-spring model is derived from physical properties [15], and, for precision motion capture input data, can be used to estimate ground reaction forces. However, it is very sensitive

to the height of the feet, and requires millimetre sensitivity in order to be used.

As a supplementary model, and to be used for the less accurate monocular system, the leg-spring model was integrated into the system. This model adapts an oscillatory human gait model [17], which treats the human legs as two rigid springs, each supporting a central human mass when in contact with the ground. In this model, the legs act as linear springs, around which the centre of mass of the body rotates.

This method is not as immediately applicable to quadrupeds, as the 2 pairs of legs lie along 2 planes, rather than the single plane of bipeds. However, a variant of this model is implemented, in which legs in contact with the ground are modelled as springs rotating about the relevant shoulder or hip. The vertical oscillation of the rotating point is therefore considered, rather than the overall centre of mass.

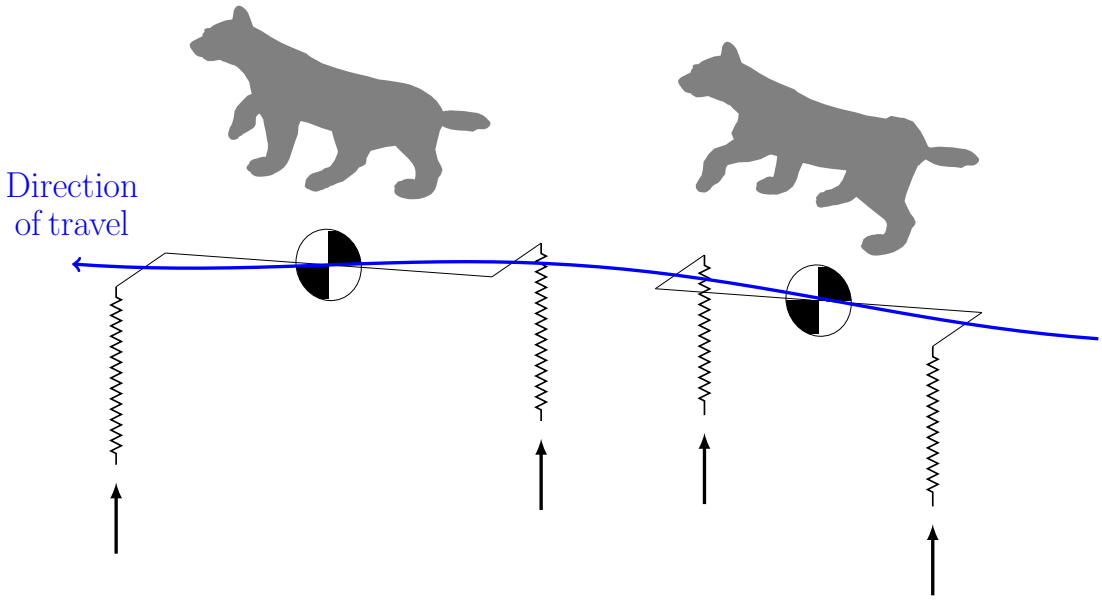


Figure 26: Visualisation of the leg-spring model at two points in the gait cycle. Only 'active' springs are shown

The assumption is made that, while in contact with the ground, resultant motion is entirely in the xz plane (there is no motion in the left to right direction). As a result, the vertical ground reaction force F_z, is calculated via

$$F = k_l(L - L_0)\cos\phi \tag{20}$$

Where L is the current length of the spring, L_0 is the length before impact, ϕ is the angle between the spring and the vertical, and k_l is the leg spring stiffness.

4.2 Physical parameter selection

4.2.1 Dynamic parameters

To identify sensible dynamic parameters to construct this model, existing literature [13, 14] producing more complex computational models was consulted. The literature found a mean density for canine bone and muscle,

$$\rho_{\text{bone}} = 1950\,\text{kg}\,\text{m}^{-3}, \quad \rho_{\text{muscle}} = 1060\,\text{kg}\,\text{m}^{-3} \tag{21}$$

	L/r_i	L/r_o	L/d
Bone	100	20	
Body	20	7	5

Table 3: Ratios between bone length L and other bone size parameters selected for the inverse dynamics model

Furthermore, a detailed investigation of the inertial matrices [14] for individual canine limbs was used as a basis to generate the size ratios governing the model.

The individual radii of each bone and body were determined from its length L. The ratios selected to do this were chosen as they provided a strong fit to the experimental data in literature. The selected ratios are outlined in Table 3.

4.2.2 Paw stiffness

In this model, the paw pads are modelled as linear springs, with unloaded length L_0, and stiffness k. When a paw is off the ground, the pads are unloaded.

To generate this model, literature was identified that measured the mechanical properties of animal paws [15]. The paper identified that animal paws behave as non-linear springs, with a somewhat linear region for loading between 0 and 2.5 times the animal's bodyweight per limb. It has been found that, in standard ranges of motion, animals generally exert ground reaction forces (GRFs) of 2-3 times their body weight per limb [32]. As such, the spring is not perfectly linear in the desired range, but will be assumed so for this model.

Assuming this loading, this project's model will extract linear spring parameters.

Dog	Mass, M	Pad thickness, L_0		Pad stiffness, k		Source
	kg	mm		$\mathrm{N\,mm^{-1}}$		
		front	rear	front	rear	
Dalmation	21	13	14	49	61	[15]
Foxhound	26	16	19	46	78	
	10	12.0	9.5	56.2	45.7	
	25.0	12.6	12.5	139.8	103.1	
	13.6	14.0	13.7	44.3	42.2	[33]
	9.1	10.6	7.0	39.8	28.8	
	22.7	17.0	16.2	76.2	60.5	

Table 4: Experimental data on elastic properties of paw pads for several canines

Literature results. Table 4 shows data extracted from multiple sources on pad behaviour. This can be used to develop an understanding of the scaling of the pad behaviour with size of the dog. It is important to note the different behaviour between fore and hind paws. Investigations into the scaling of stiffness with size conducted previously [33] propose a scaling of fore limb stiffness with M, and a scaling of rear limb stiffness of $M^{0.85}$. These conclusions have been used to form the following relationships used for the model,

$$k_{\text{front}} = 3.42 \times 10^3 M, \quad k_{\text{rear}} = 5.16 \times 10^3 M^{0.85} \tag{22}$$

Adapting to the model. However, the experimentation with the model has shown that these values significantly overestimate the actual ground reaction forces. This is due to the motion capture markers actually being placed at the top of the dog's paw due to physical constraints. This means that the deflection measured is a function of not just the thin paw pads, but the entire elasticity of the paw.

As such, the paw stiffness used in the model was treated as two springs in series - the paw pads with stiffness defined in Equation 22, and a second, unknown stiffness, which will be modified through experimentation to best match the output data.

4.2.3 Paw displacement

Due to the sensitivity of such measurements, defining a fixed 'unstretched' length of the paw-pads is not feasible. Instead, it is necessary to somehow identify the displacement-time characteristics for each individual footfall.

Common solutions. State of the art systems use a series of methods for identifying the footfalls of animals from vertical position data. Methods used on hoofed animals [34] use a

combination of position, velocity and acceleration based detection methods. However, these cannot be applied to this system as easily for several reasons, including:

- Paw behaviour - as hoofs are incredibly stiff, the resulting footfall is very clearly defined - there is little vertical motion after the hoof contacts the ground. This is not the case for canines, for which the entire paw experiences significant compression.
- Precision - while the motion capture data collected has a high level of precision, the kinematic data from ordinary video will suffer from a lack of precision, resulting in difficulties with detection.

Chosen solution. A neural network was trained to identify when these footfalls were taking place. To do this, the network was provided extracts from Dataset 4, containing paw vertical position data, and corresponding ground reaction force data. From this, it was optimised to best predict when each paw was in contact with the ground.

The network was set up with two hidden layers, which allows it to evaluate first and second order time differences - effectively providing it with the means to evaluate position, velocity and acceleration in predicting the footfalls.

The network was only trained on the limited dataset collected as part of Dataset 4, and a small amount of supplementary data, and so would not robustly detect footfalls in a range of scenarios. To overcome this, as discussed in 6.1, a large dataset could be collected to provide a rigorous training background.

4.2.4 Paw damping

Damping values were selected in order to provide the best results in the datasets provided. A final value of $15\,\mathrm{N\,s\,m^{-1}}$ for the front paws, and $20\,\mathrm{N\,s\,m^{-1}}$ for the rear was chosen.

4.2.5 Leg spring stiffness

The effective leg stiffness, k_l, used in the leg spring model, cannot be as easily obtained in literature. The model used on human gait uses the empirical formula,

$$k_l = 40\frac{mg}{h}\overline{v} \tag{23}$$

Where m is the human mass, h the height of the human, and $\overline{v}$ the average forward velocity. However, the paper allows for up to 40% tuning of this value when optimising it to fit the data.

Adapting this estimate to dogs proved surprisingly accurate. The mass m was selected as one quarter the dog's mass (which is approximately the total mass supported by that paw), and h was selected as the dog's hip or shoulder height.

4.3 Solving the equations

4.3.1 Inverse kinematics

First, it is necessary to identify the kinematics of all of the joints and bones in the model. To start with, all that is known is the position vector of each joint j, $\mathbf{x_j^f}$, where f denotes the frame number, and the timestep, dt, between frames.

The first stage is to extract velocity and acceleration for each joint, $\mathbf{v_j}$ and $\mathbf{a_j}$ respectively, through simple numerical differentiation,

$$\mathbf{v_j^f} = \frac{\mathbf{x_j^{f+1}} - \mathbf{x_j^f}}{dt}, \quad \mathbf{a_j^f} = \frac{\mathbf{v_j^{f+1}} - \mathbf{v_j^f}}{dt} \tag{24}$$

Next, the kinematics of the bones are required. For this, bone b is defined as having start and end joints p and q respectively. It is useful to define the position vector between the start and end of the bone, $\mathbf{r_b^f}$,

$$\mathbf{r_b^f} = \mathbf{x_q^f} - \mathbf{x_p^f} \tag{25}$$

The position, velocity and acceleration of the bone centre of mass are also required, which are simply the average of those of the joints either side of the bone.

Lastly, the rotational mechanics of the rigid bones are required. The rotation between two frames can be represented by a vector $\boldsymbol{\theta_b^f}$, which corresponds to an axis-angle representation,

$$\boldsymbol{\theta_b^f} = \theta \mathbf{e}, \qquad \theta = \frac{\mathbf{r_b^f} \cdot \mathbf{r_b^{f+1}}}{\left|\mathbf{r_b^f}\right| \left|\mathbf{r_b^{f+1}}\right|}, \quad \mathbf{e} = \frac{\mathbf{r_b^f} \times \mathbf{r_b^{f+1}}}{\left|\mathbf{r_b^f} \times \mathbf{r_b^{f+1}}\right|} \tag{26}$$

From this, the angular velocity ω and angular acceleration α can be found through numerical differentiation,

$$\boldsymbol{\omega_b^f} = \frac{\boldsymbol{\theta_b^{f+1}} - \boldsymbol{\theta_b^f}}{dt}, \quad \boldsymbol{\alpha_b^f} = \frac{\boldsymbol{\omega_b^{f+1}} - \boldsymbol{\omega_b^f}}{dt} \tag{27}$$

It is also necessary to identify the moment of inertia matrix, $\mathbf{I}$, in a fixed reference frame (as all of the kinematics calculated are in a fixed reference frame). To do this:

- Identify local principal axis $\mathbf{k} = \mathbf{r_b}/|\mathbf{r_b}|$
- Select any two perpendicular vectors to form a principal set $\mathbf{i}, \mathbf{j}, \mathbf{k}$. Any perpendicular pair is valid by axial symmetry.
- Use these to produce a transformation matrix $\mathbf{T} = [\mathbf{i}, \mathbf{j}, \mathbf{k}]$, which transforms the fixed frame moment of inertia matrix, $\mathbf{I_f}$, via

$$\mathbf{I} = \mathbf{T}^T \mathbf{I_f} \mathbf{T} \tag{28}$$

4.3.2 Boundary conditions

Several boundary conditions can also be imposed to constrain the problem and improve results. These include:

- **Free joints** - 'joints' in this model that are actually at the end of the skeleton, such as on the dogs paws, are only acted on by external forces, and can be said to have no torque.
- **Paw off ground condition** - for a paw deemed to be fully off of the ground, there is no reaction force in any direction on the paw.
- **Paw on ground condition** - for a paw in contact with the ground, the vertical reaction force can only be positive.

4.3.3 Weightings

Complete method. Inverse dynamics methods often use a method of applying small perturbations, of a magnitude equal to the typical error in measurements, to a series of input motion capture data [12]. A covariance matrix is then developed, detailing the sensitivity of each equation to a change in measurements. Finally, a dense weightings matrix is calculated from the covariance matrix.

Diagonal method. However, a lack of diversity of motion capture data meant that the method chosen for this project was simply to select a set of weightings applied individually to each equation. In other words, W would be a diagonal matrix. Table 5 shows the weightings selected, based on experimentation with the dataset.

	Weighting	
Equation	Motion Capture	KDN
Inertial	2	2
Rotational	1	1
Paw-spring model	1	0.5
Leg-spring model	0.5	1

Table 5: Weightings selected for inverse dynamics model

4.3.4 Calculating the solution

Now that the constituent equations have been defined, the solution for a given frame $\mathbf{f}^\star$ can be found, that minimises the least square error in the weighted equations,

$$\mathbf{f}^\star = \min_{\mathbf{f}} \|\mathbf{WAf} = \mathbf{Wb}\|_2 \tag{29}$$

This equation is optimised using the Python functionality `scipy.optimize.lsq_linear`, which is able to solve least square problems subject to boundary conditions. The boundary conditions applied are as defined in Section 4.3.2. This method is applied on each frame of motion individually.

4.4 Testing

This section describes some unit tests, confirming that the model behaves physically as expected for some simple use cases.

Static weight loading. This test validates the model by modelling a dog resting on its paws. All paws are set to have the same stiffness k, giving a static height off of the ground of

$$h = L_0 - \frac{mg}{4k} \tag{30}$$

Where L_0 is the unstretched spring length.

The inverse dynamics model predicts a consistent $F_z/mg = 0.25$ for each paw, which is as to be expected for a symmetric static loading.

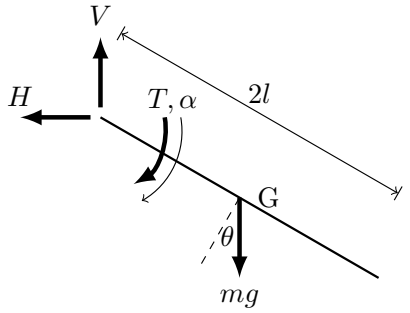


Figure 27: Idealised bone undergoing constant acceleration about one end

Validation of torque - pure angular acceleration. To validate the results of pure torque, the unit dog was set to be resting on three of its paws, whilst the fourth paw was free to rotate. The free paw was given a uniform angular acceleration, and the expected torques and forces were,

$$\left.\begin{aligned} T &= mgl\cos\theta - (I_{yy} + ml^2)\ddot{\theta} \\ V &= mg + ml\alpha\cos\theta + \dot{\theta}^2 l\sin\theta \\ H &= m\dot{\theta}^2 l\cos\theta - \alpha l\sin\theta \end{aligned}\right\} \text{ for } \theta(t) = \theta_0 - \frac{1}{2}\alpha t^2 \tag{31}$$

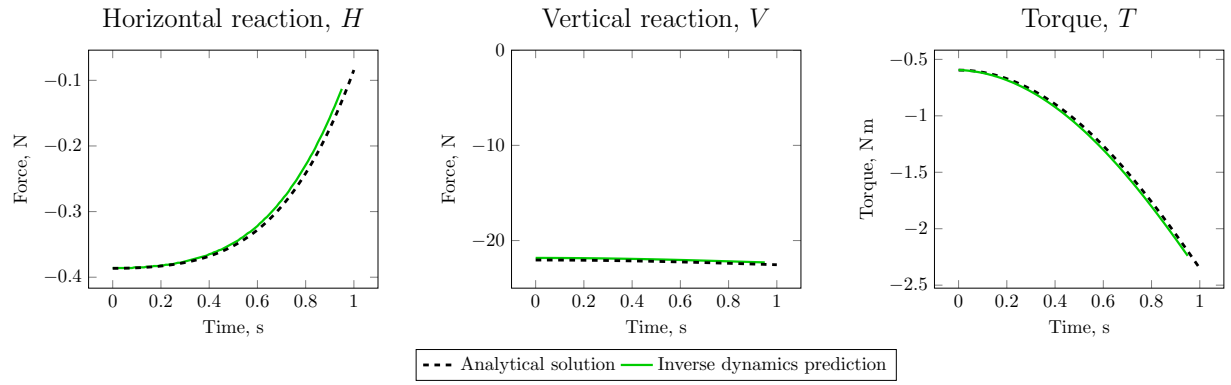


Figure 28: Analytical and experimental dynamic results from angular acceleration test

This was plotted in Figure 28. The horizontal reaction H, and torque T agree very well with the experimental results. V has a small but more significant error of approximately 1%, which arises from this force being included in the body dynamic equations for this model, resulting in greater sensitivity to initial conditions.

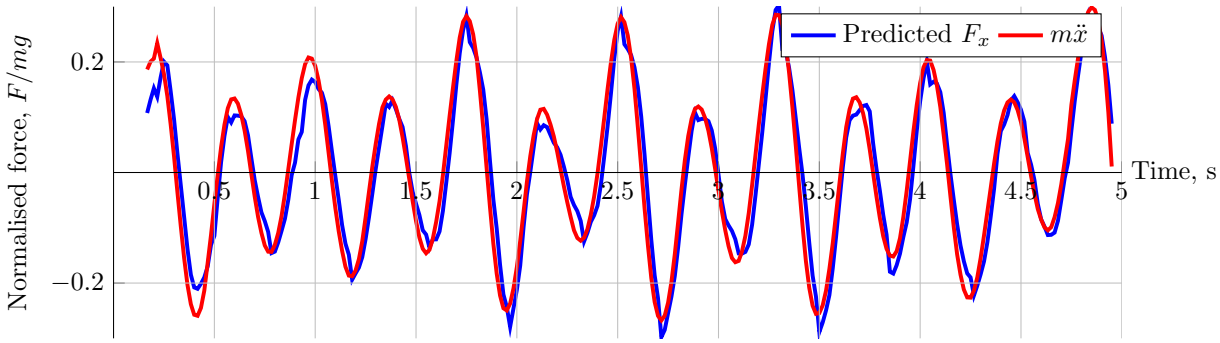


Figure 29: Comparison of the predicted net x-direction reaction force to the net D'Alembert force

Evaluation of global consistency. The series of constituent equations used by this model do not consider the model as a whole - each equation only concerns one bone or joint. As such, reviewing how well the global dynamics and kinematics agree provides a route for evaluating the model.

Shown in Figure 29 is the net predicted x-wise force acting on the dog, and the mass multiplied by the acceleration of the model centre of mass. Newton's second law would predict

$$F_x = m\ddot{x} \tag{32}$$

Figure 29 therefore shows a strong global consistency of the model. Despite this, the curves do not match perfectly. This is likely caused by noise in the motion capture data, that was not effectively removed by the smoothing methods.

5 Results

5.1 KDN results

In order to review the accuracy of the Kinematic Deep Network, its results can be compared to either known motion capture data, or evaluated using multiview data.

5.1.1 Motion capture comparison

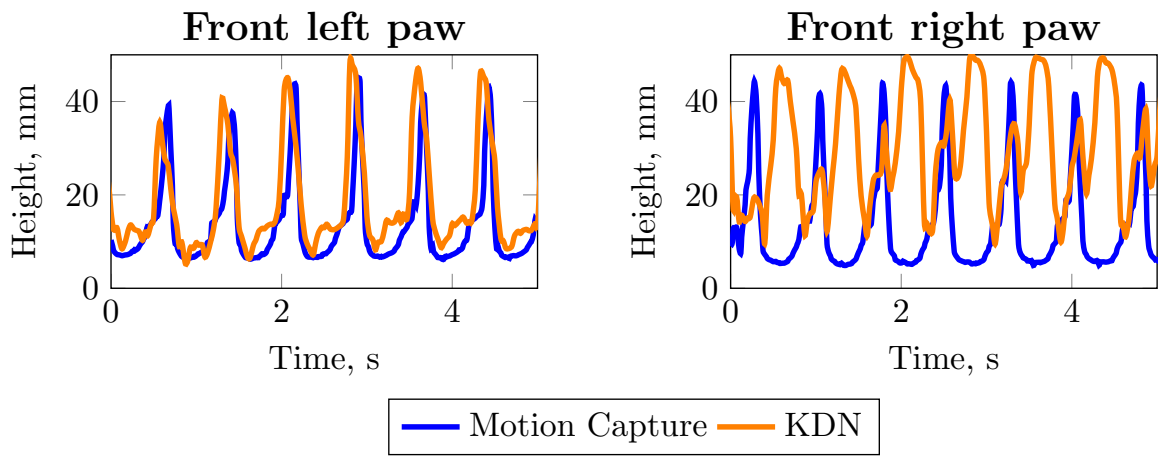


Figure 30: Predictions of paw vertical height by two methods, compared for the front left and right paw on a $3\,\mathrm{km\,h^{-1}}$ run. The video is taken from the right hand side of the dog

To evaluate the physical accuracy of the KDN, the network was applied to the video extracted from Dataset 4. The height values of the front two paws both from the KDN and the motion capture data are shown in Figure 30.

The results show reasonable agreement in vertical position for the left paws, and significant error for the right. This is due to the video provided being from the left hand side of the dog - resulting in a confident network prediction for the left side, and errors for the side that was occluded by the left side legs.

5.1.2 Multiview consistency

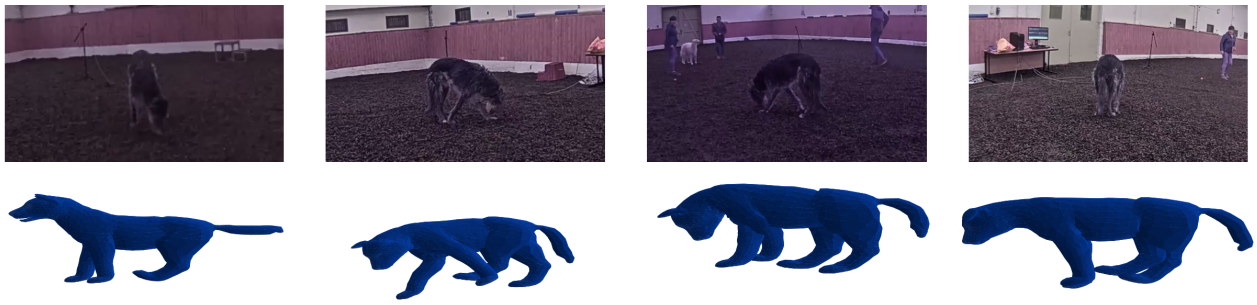


Figure 31: Visualisation of the KDN fit to a multiview snapshot

Motivation. The multiview dataset collected as Dataset 2 provides a means to measure the consistency of the KDN: the similarity of parameters for meshes fitted to different views of the same dog. Figure 31 provides an example of a mesh fit for the same moment in time from the 4 cameras present.

Metrics. In order to quantitatively evaluate this, for several clips, the standard deviation in the main SMBLD parameters (pose θ, shape β and scale κ) are evaluated. Global position and rotation are not considered, as consistency in these is not to be expected for cameras in different locations and orientations.

The data presented is generated by identifying frames of each clip in which the dog is visible in more than one frame, and fitting the mesh to each view. Then, the standard deviation is calculated across all parameters. The final metric is the mean standard deviation in the given parameter across all frames in which the dog can be seen from multiple cameras.

Dog in clip	$\bar{\sigma}_\theta$, °	$\bar{\sigma}_\beta$	$\bar{\sigma}_\kappa$
Ally	6.12	0.050	0.036
Douglas	6.43	0.054	0.044
Gracie	7.01	0.062	0.048
Patrick	6.88	0.049	0.045

Table 6: Multiview consistency for several clips from Dataset 2

Results. Table 6 details the results of this global consistency analysis. The shape and scale are relatively consistent, with a usual deviation of approximately 0.05. Typical shape values are between -1 and 1, so this suggests a maximum deviation of 5%.

The average pose consistency is approximately 6.5°, which is reasonable considering that it

is likely that, for any still, at least two cameras will view the dog in a position from which recovering pose is challenging, such as from directly behind or in front.

5.2 Dynamics results

5.2.1 Gait analysis methodology

First, it is necessary to introduce terminology and methodology in analysing the ground reaction forces (GRFs) of canines.

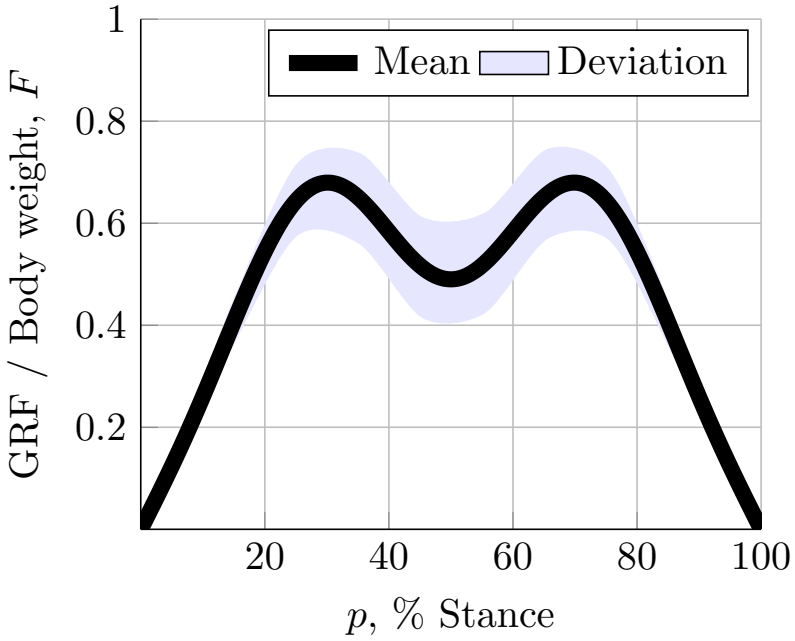


Figure 32: Characteristic stance graph

Force stance graph. In order to cluster several footfalls into a single gait profile each footfall is combined on a single graph of fraction of cycle time, p (% Stance) against normalised ground reaction force, F. Figure 32 shows an example of a characteristic load graph.

Modal fitting. Due to the collection of many individual data, it is necessary to combine these into a smooth function from which results can be extrapolated. The chosen function to fit to is a sine wave summation,

$$F(p) = \sum_{n=1}^{N} a_n \sin(n\pi p) \tag{33}$$

Where n denotes the mode shape, and the vector of coefficients $a \in \mathbb{R}^N$ is optimised to the data using a least squares method. This function allows for a strong fitting to typical dog ground reaction force profiles.

Similar methods have been applied to analysing ground reaction forces of humans [35], and have found that the first two to four harmonics are sufficient to fully analyse gait patterns. As such, a number of modes $N = 4$ was selected.

Comparison metrics. The modal decomposition, and time series comparison, allow for several different comparison metrics between predicted and actual ground reaction forces. For this report, two metrics will be considered:

- **First order error** - Error between two stance curves, in the first mode shape. This gives a sense of the difference in amplitude between the predicted and actual gait.
- **RMS error** - The root mean squared error between the two time series.

Symmetry index. One way in which the results of dynamic data can be used to diagnose problematic limbs is through a symmetry index - which quantifies the tendency of peak force towards one side (left/right) of a dog [36]. This can be defined independently for the front and rear of the dog, and is defined as

$$SI = \left| \frac{F_{z,\text{left}} - F_{z,\text{right}}}{F_{z,\text{left}} + F_{z,\text{right}}} \right| \tag{34}$$

F_z is the peak footfall force (for this report, taken as the magnitude of the first modal constant a_0). An SI value of 0 indicates perfectly symmetric movement.

5.2.2 Dynamic predictions from motion capture

In order to evaluate the performance of the solver, the motion capture and ground reaction force data collected as part of Dataset 4 was used.

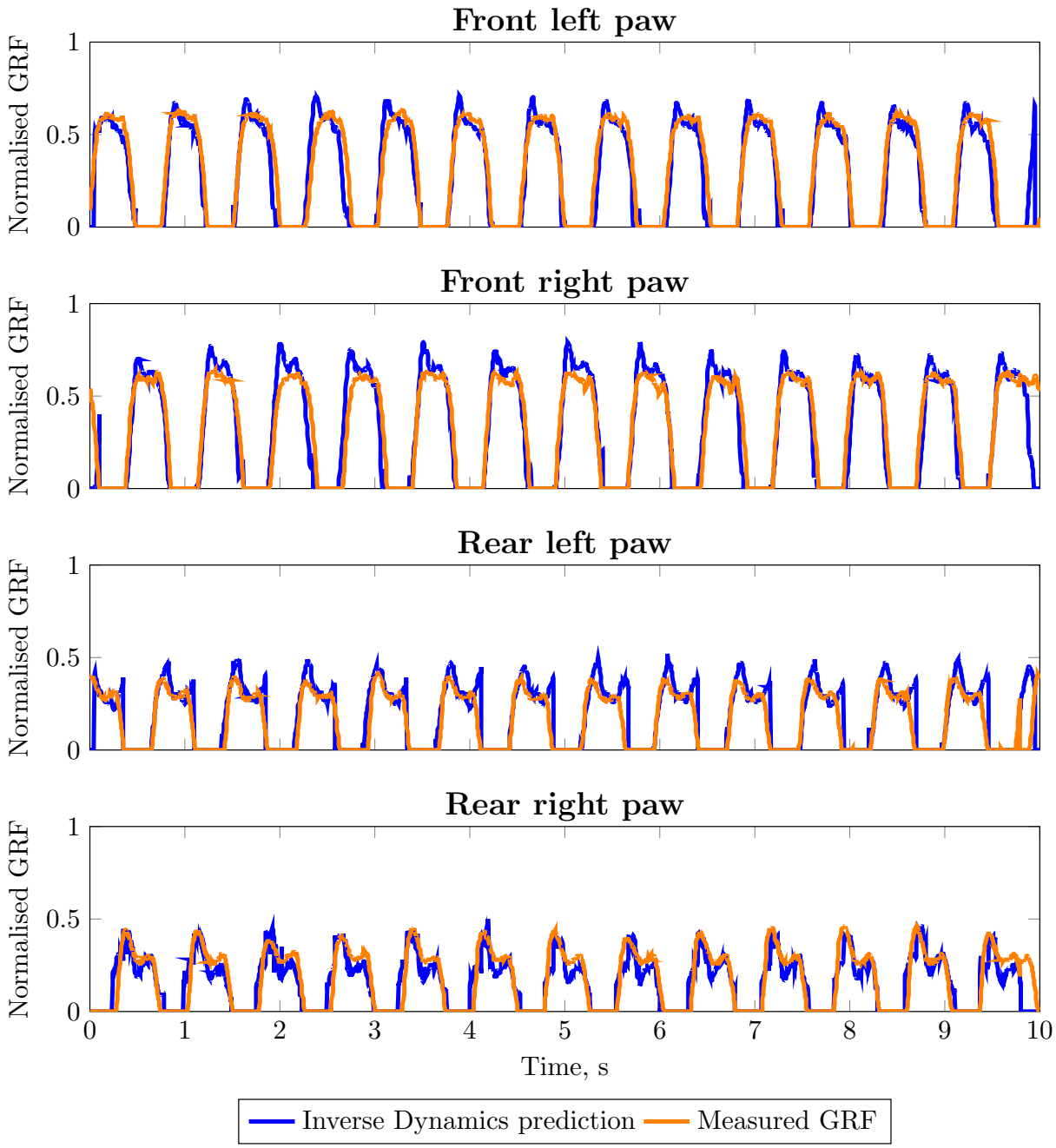


Figure 33: Predicted ground reaction forces for a 10 second clip of a 3 km h^{-1} run

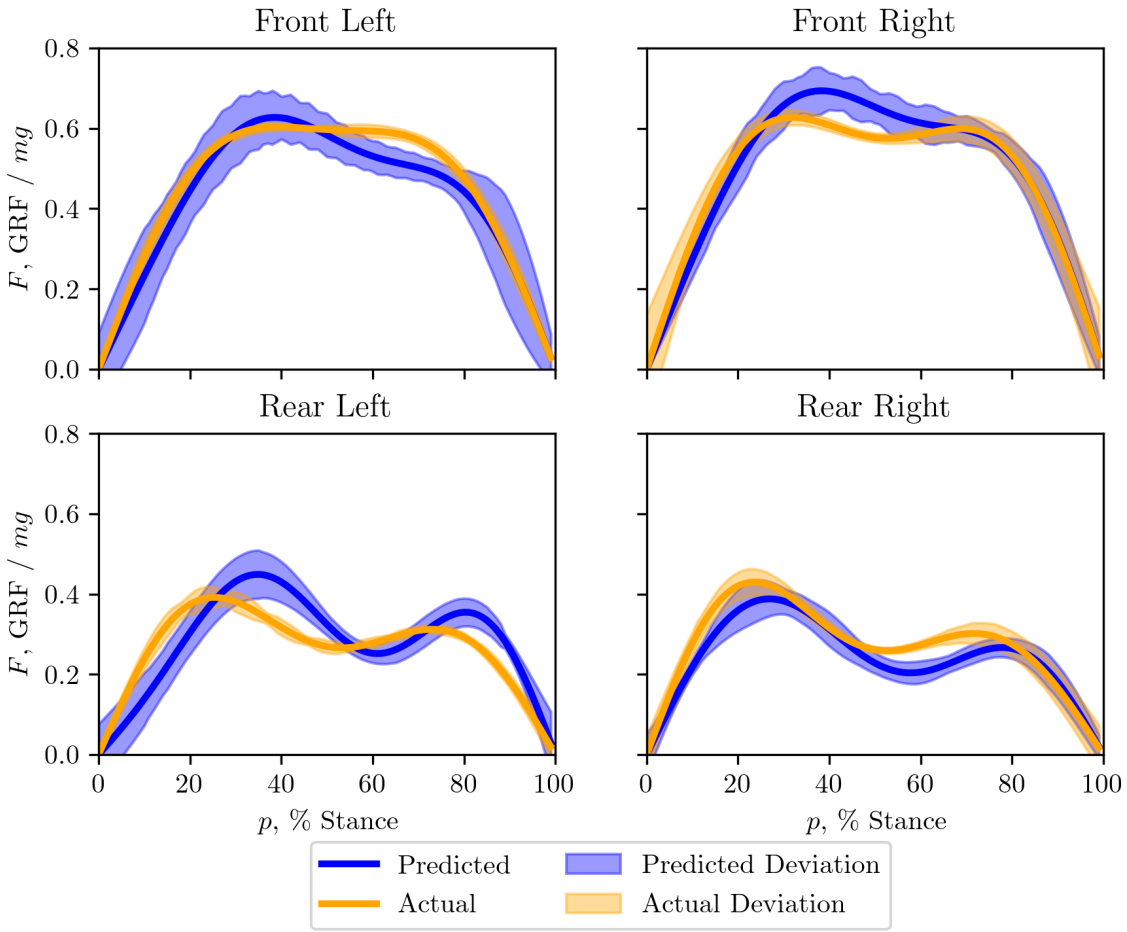


Figure 34: Predicted ground reaction forces averaged as a fraction of stance for a 10 second clip of a $3\,\mathrm{km\,h^{-1}}$ run

Qualitative results. Figures 33 and 34 compare the predictions of the solver to the known ground reaction forces, for a 10 second extract of a single run at $3\,\mathrm{km\,h^{-1}}$ captured as part of the dataset.

The rear right paw unfortunately had an erroneous placement of the marker - reviewing the footage revealed that the marker was placed slightly above the paw on the leg. This meant that the nature of the elastic behaviour was not fully captured, and so the paw spring equation was not considered for this paw. This is why the resultant data is significantly 'noisier', as the spring model helps provide smoothness to the result.

Speed	Paw	First Order Error		RMS Error
$km\,h^{-1}$		Nominal	%	
3	Front left	0.02	2.3	0.09
	Front right	-0.02	-3.1	0.10
	Rear left	0.03	7.5	0.08
	Rear right	0.01	1.8	0.08
6	Front left	0.18	-3.5	0.11
	Front right	0.18	17.9	0.23
	Rear left	-0.02	-2.6	0.14
	Rear right	-0.15	-23.8	0.13

Table 7: Errors in inverse dynamic predictions, compared with experimental results, from two runs of Dataset 4

Quantitative results. Table 7 shows the evaluation metrics of two separate dynamics predictions. The inverse dynamics predictor shows a strong agreement with experimental data for the $3\,km\,h^{-1}$ run, with an average first order error of 3.7%. The predictor has a reduced accuracy for the $6\,km\,h^{-1}$ run, with an average first order error of 12.0%. This is in part due to the frequency of the solver not adequately capturing the quickly changing paw heights.

	Symmetry Index, SI	
	Experimental result	Inverse dynamics prediction
Front	0.02	0.05
Rear	0.01	0.03

Table 8: Calculated symmetry indices for the $3\,km\,h^{-1}$ run

In order to demonstrate the application of data produced by the solver, the symmetry indices have been calculated for the data presented in Figure 34. Table 8 demonstrates these results. While the symmetry indices predicted by the inverse dynamics solver are larger than the experimental values, they are sufficiently small to justify the same conclusion as the actual data - that this dog has minimal gait asymmetry.

5.2.3 Dynamic predictions from monocular video

Dog 1		Dog 2	
Paw	First order error %	Paw	First order error %
Front left	-27.6	Front left	-37.3
Front right	-28.4	Front right	-44.8
Rear left	-0.9	Rear left	-18.8
Rear right	102.1	Rear right	63.7

Table 9: First order errors of inverse dynamic predictions, sourced from two separate canines filmed at the Vet Department

To demonstrate the full method designed in this project, videos taken of dogs on the Zebris forceplate in the Vet Department were used, and the accuracy of the predictions detailed in Table 9.

Due to the compounding of errors propagating through the system between the image processing and inverse dynamics stages, the errors present are quite large. This is especially true for the rear right paws. This is because these videos are from the left side of the dog, and so the rear right leg suffers from significant occlusion, as its view is partially blocked by the left hand side of the dog.

6 Discussion

Motion capture inverse dynamics performance. The inverse dynamics model showed strong predictive potential of ground reaction forces from input motion capture data. The model was tested on several individual clips of runs at both $3\,\mathrm{km\,h^{-1}}$ and $6\,\mathrm{km\,h^{-1}}$, and demonstrated accurate predictions of ground reaction forces.

For the paw-spring model used, a relatively high frequency data stream is required for strong performance, so the model becomes less accurate for higher gait speeds due to faster footfalls.

KDN kinematic predictions. The kinematic predictions from the KDN on ordinary video footage show a strong agreement with the data for the legs closer to the camera. The resolution of data for these legs is promising, reasonably matching the resolution required for the inertial equations and leg-spring model.

The network struggles to predict kinematics for the occluded legs. A possible scheme to rectify this is to have two synchronised videos, one from each side of the dog, and use the KDN predictions for each half of the dog independently. Although this solution strays from the objective of developing a system capable of being easily used 'in the wild', it is still a solution that addresses many of the shortfalls of current dynamic prediction methods.

KDN dynamic predictions. Although the KDN is able to make some reasonable kinematic predictions, transforming this to the inverse dynamics suffers significant error, as errors compounding through the system result in large discrepancies in force predictions.

Despite this, the kinematic predictions do suggest that this system is capable of reasonable predictions, and a much more thorough development and training basis could make this method viable for the task at hand.

Computation time. Current methods for gait analysis on custom software are capable of relatively quick analysis, and so this method should be fast to be able to compete. A computer using a Titan Xp GPU is able to run images through the network at a speed of approximately 30 images per second. Inverse dynamics computation occurs at 40 frames per second on i7-8700 6 core CPU (3.2 GHz). This means that the full method can run on a high-end PC at approximately 17 frames per second.

Modern cameras typically capture video at 24-30 fps, so this method currently runs at 0.5-0.75x real time speed. This is a promising speed, providing results in a very short time frame.

Usability. The system produced has been run on relatively high end computers, capable of quick computation. However, there is no reason this system could not be adapted to more standard devices, even including mobile phones. This kind of development strengthens

the advantage of this method over industry standards, as it opens up the possibility for a fast, preliminary analysis of canine gait, completely removed from laboratory technology and equipment.

Furthermore, the input to the KDN is an ordinary stream of images, so the integration of this method with any modern camera is trivial.

Limitations of the Inverse Dynamics Model. It is important to acknowledge the individual limitations of the 3D inverse dynamics model presented here. These include:

- Parameter selection - To obtain a more accurate and adaptable model, the parameters should be optimised over a much larger dataset of dogs, to capture the variation between breeds, gait speeds, and external conditions.
- Twisting and torsion - A significant assumption has been made that the torque at each joint acts only in a direction perpendicular to the two bones connected to the joint. This assumption does not account for twisting (a torque vector parallel to a bone), or out of plane rotations. Twisting cannot be sufficiently determined by this model, as the rotation of the axisymmetric bones about their axis of symmetry is not calculated, and is incredibly difficult to determine from markers or image data.

6.1 Future work

While the project has demonstrated the viability of the method set out in its aims, there is scope for further developing the model, both for accuracy and versatility.

Niche use cases. The viability of the video system has been demonstrated for gait analysis. For more complex cases, such as a dog jumping, or the plane of motion relative to the camera changing (walking in a circle), the Kinematic Deep Network produced has had insufficient training data to accurately fit to these cases. Future work could involve collecting a much larger dataset to capture these cases to train a more versatile KDN.

Ground stiffness. The stiffness of ground is a significant factor to include in modelling canine gait. This is especially true 'in the wild', where surfaces such as grass may have significant compliance. Modelling this requires high precision, so is a possible avenue to explore for motion capture modelling, but may not be possible for monocular video modelling.

The primary reason for not exploring this further is a lack of validation data - all collected ground reaction force data was from a forceplate of unknown stiffness. An experiment could have been devised to measure the stiffness experimentally, but unfortunately this was not possible in the time frame of the project.

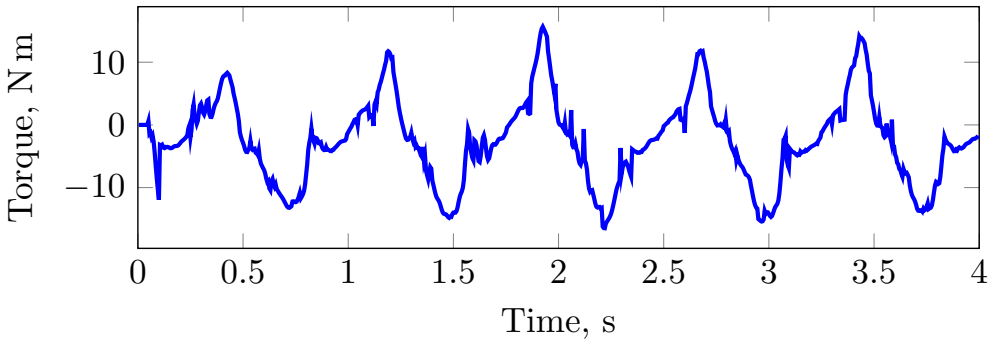


Figure 35: The predicted torque in the left front knee for a run at $3\,\mathrm{km\,h^{-1}}$ from Dataset 4

Validation of internal dynamics. While internal forces and torques are predicted by the solver, they are unable to be validated by the data collected by this project. Validating this data would either require the use of invasive or *ex vivo* methods, or comparison to benchmark musculoskeletal modelling systems.

Figure 35 shows an example of such a result - the torque in the left front knee of a $3\,\mathrm{km\,h^{-1}}$ run from Dataset 4. While the values seem physically reasonable and are consistent with the speed of the gait in question, there was no available method to extract the actual internal torques for verification.

KDN speed. The KDN could be adapted to a less accurate, but much more lightweight network. This would be a probable next step if the method was either desired to run in real time; or to be usable on a device with less processing power, such as a mobile phone.

Adaptability of footfall detector. The footfall detector described in Section 4.2.3 was trained on the limited data collected. A much larger dataset collected would result in a more versatile detector that could work on a much larger range of canine motion capture data.

7 Conclusions

Image processing. The novel image processing system developed as part of this project has shown significant promise in the prediction of canine kinematics from monocular video. Examples have been shown of the KDNs performance, both of qualitative matching to images, and qualitative agreement with motion capture data.

However, the KDN requires much more development for high reliability and accuracy. The network can struggle to predict kinematics for the set of limbs farther from the camera, and has only been trained to a limited set of canine poses and motions.

Additionally, the work carried out has highlighted several key issues faced when attempting to extract kinematic information from monocular videos, including leg occlusion, low frame rate and resolution, and complex poses or orientations not seen in controlled laboratory conditions.

Inverse dynamics. The inverse dynamics model produced for this project has allowed for powerful predictions of ground reaction forces from relatively few model parameters and complexity.

To further develop this model, a more thorough experimental data set to diversify model parameters used would significantly increase the model's adaptability to new canines and environments.

Method viability. This project demonstrates that there is viability for a non-invasive, easily available method of extracting dog dynamics from ordinary video. The process presented has feasibility as a tool to be used as an initial assessment of canine gait, avoiding many of the barriers to use present in current methods.

The process outlined in the report does not demonstrate the full accuracy that could be possible given a much larger development time frame, and a greater range of data to train and develop the method.

References

[1] L Braun, A Tichy, C Peham, and B Bockstahler. Comparison of vertical force redistribution in the pads of dogs with elbow osteoarthritis and healthy dogs. *The Veterinary Journal*, 250:79–85, 2019.

[2] Laurent Fanchon and Dominique Grandjean. Accuracy of asymmetry indices of ground reaction forces for diagnosis of hind limb lameness in dogs. *American journal of veterinary research*, 68(10):1089–1094, 2007.

[3] Zebris Candidgait for Dogs, `https://www.zebris.de/en/medical/products-solutions/canidgait-for-dogs`. Accessed: 02-05-2020.

[4] Jason F Headrick, Songning Zhang, Ralph P Millard, Barton W Rohrbach, Joseph P Weigel, and Darryl L Millis. Use of an inverse dynamics method to describe the motion of the canine pelvic limb in three dimensions. *American journal of veterinary research*, 75(6):544–553, 2014.

[5] Scott Tashman and William Anderst. In-vivo measure-

ment of dynamic joint motion using high speed biplane radiography and ct: application to canine acl deficiency. *J. Biomech. Eng.*, 125(2):238–245, 2003.

[6] D Gavrila and LS Davis. Tracking of humans in action: A 3-d model-based approach. In *ARPA Image Understanding Workshop*, pages 737–746. (Palm Springs), 1996.

[7] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015.

[8] Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 3d menagerie: Modeling the 3d shape and pose of animals. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6365–6373, 2017.

[9] Silvia Zuffi, Angjoo Kanazawa, Tanya Berger-Wolf, and Michael J Black. Three-d safari: Learning to estimate zebra pose, shape, and texture from images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5359–5368, 2019.

[10] Opensim Inverse Dynamics, `https://simtk-confluence.stanford.edu/display/OpenSim/How+Inverse+Dynamics+Works/`. Accessed: 01-09-2019.

[11] Lei Ren, Richard K Jones, and David Howard. Whole body inverse dynamics over a complete gait cycle based only on measured kinematics. *Journal of biomechanics*, 41(12):2750–2759, 2008.

[12] Antonie J Van Den Bogert and Anne Su. A weighted least squares method for inverse dynamic analysis. *Computer methods in biomechanics and biomedical engineering*, 11(1):3–9, 2008.

[13] Nathan P Brown, Gina E Bertocci, Gwendolyn J Levine, Jonathan M Levine, Dena R Howland, et al. Development of a canine rigid body musculoskeletal computer model to evaluate gait. *Frontiers in Bioengineering and Biotechnology*, 8:150, 2020.

[14] Tamar Amit, BR Gomberg, J Milgram, and R Shahar. Segmental inertial properties in dogs determined by magnetic resonance imaging. *The Veterinary Journal*, 182(1):94–99, 2009.

[15] R McN Alexander, MB Bennett, and RF Ker. Mechanical properties and function of the paw pads of some mammals. *Journal of Zoology*, 209(3):405–419, 1986.

[16] Reinhard Blickhan. The spring-mass model for running and hopping. *Journal of biomechanics*, 22(11-12):1217–1227, 1989.

[17] Hansol X Ryu and Sukyung Park. Estimation of unmeasured ground reaction force data based on the oscillatory characteristics of the center of mass during human walking. *Journal of biomechanics*, 71:135–143, 2018.

[18] Benjamin Biggs, Thomas Roddick, Andrew Fitzgibbon, and Roberto Cipolla. Creatures great and smal: Recovering the shape and motion of animals from video. In *Asian Conference on Computer Vision*, pages 3–19. Springer, 2018.

[19] Benjamin Biggs, James Charles, and Oliver Boyne. Who left the dogs out? 3d animal reconstruction with expectation maximisation in the loop. In *European Conference on Computer Vision*, 2020. (Submission under review).

[20] Jinkun Cao, Hongyang Tang, Hao-Shu Fang, Xiaoyong Shen, Cewu Lu, and Yu-Wing Tai. Cross-domain adaptation for animal pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9498–9507, 2019.

[21] Amazon Mechanical Turk, `https://www.mturk.com`. Accessed: 25-10-2019.

[22] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*, volume 2, 2011.

[23] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016.

[24] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2878–2890, 2012.

[25] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3907–3916, 2018.

[26] Unity 3D Dog pack, `https://assetstore.unity.com/packages/3d/characters/dog-big-pack-105660`. Accessed: 20-01-2020.

[27] Pytorch 3D, `https://pytorch3d.org`. Accessed: 15-02-2020.

[28] Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. Laplacian mesh optimization. In *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pages 381–389, 2006.

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[30] Abraham Savitzky and Marcel JE Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964.

[31] Klaus-Dieter Budras. *Anatomy of the Dog: With Aaron Horowitz and Rolf Berg*. Schlütersche, 2010.

[32] Andrew A Biewener. Biomechanics of mammalian terrestrial locomotion. *Science*, 250(4984):1097–1103, 1990.

[33] Kai-Jung Chi and V Louise Roth. Scaling and mechanics of carnivoran footpads reveal the principles of footpad design. *Journal of the Royal Society Interface*, 7(49):1145–1155, 2010.

[34] Sandra D Starke and Hilary M Clayton. A universal approach to determine footfall timings from kinematics of a single foot marker in hoofed animals. *PeerJ*, 3:e783, 2015.

[35] E Schneider and EY Chao. Fourier analysis of ground reaction forces in normals and patients with knee joint disease. *Journal of biomechanics*, 16(8):591–601, 1983.

[36] Steven C Budsberg, Dermot J Jevens, John Brown, Tim L Foutz, Charles E DeCamp, and Lynn Reece. Evaluation of limb symmetry indices, using ground reaction forces in healthy dogs. *American Journal of Veterinary Research*, 54(10):1569–1574, 1993.

8 Appendix and Supplementary

8.1 Declarations

Ethics statement. All data collection on dogs was performed in the Cambridge University Veterinary Department. All filming was collected on staff owned dogs voluntarily, for the non-intrusive methods of measurement used. All data collected from Amazon's Mechanical Turk was in accordance with their Terms of Service. No personally identifiable worker details are included in the dataset.

Coronavirus declaration. The outbreak of coronavirus had little effect on the project. While it prevented the collection of any further data from the Vet Department, there was sufficient data collected to produce a finished project. Furthermore, the vast majority of the work was carried out computationally, and this could continue despite the lockdown.

Risk assessment retrospective. The risks involved in this project were minimal. The initial risk assessment carried out in association with the Vet Department outlined the two primary risks of the project: animal handling and computer usage. All animal handling was conducted under supervision from an experienced member of the Department, as advised in the initial risk assessment. Additionally, breaks were routinely taken from work to minimise risks of injury or eye strain.

8.2 Supplementary - ECCV Submission

Attached is the paper submitted to the European Conference for Computer Vision (ECCV) 2020 [19]. This paper is considered supplementary to the report, and is not currently available publicly.

Who left the dogs out? 3D Animal Reconstruction with Expectation Maximization in the Loop

Anonymous ECCV submission

Paper ID 1303

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

Keywords: animal tracking, 3D morphable models, shape from silhouette

1 Introduction

Animals contribute greatly to our society, in numerous ways economic and otherwise (there are 41 million dogs in the US alone). In consequence, there has been considerable attention in the computer vision research community to interpret imagery of animals. Although these techniques share similarities to techniques for understanding images of humans, a key difference is that obtaining labelled training data for animals is more difficult than for humans, because of the wide range of shapes and species of animals, and the difficulty of educating manual labellers in animal physiology.

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

1. We reconstruct pose and shape on a test set of 2000 low-quality internet images of a complex 3D object class (dogs).

Fig. 1. End-to-end 3D Dog Reconstruction from monocular images. 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 multimodal prior, which is tuned during training with an expectation maximization scheme.

2. We directly regress to object pose and shape from a single image without a model fitting stage.
3. We use easy-to-obtain 2D annotations in training, and none at test time.
4. We incorporate fitting of a new multi-modal prior into the training phase, rather than fitting it to 3D data as in previous work.
5. We introduce new degrees of freedom to the SMAL model, allowing explicit scaling of subparts.

The closest work in terms of scale is the Category-specific Mesh Reconstruction of Kanazawa et al. [12], where 2850 images of birds were reconstructed. However we argue that doing so for the complex pose and shape variations of dogs is a significant advance in the state of the art.

Table 1 summarizes previous work on animal reconstruction. It is interesting to note that while several papers demonstrate reconstruction across species, which *prima facie* is a richer class than just dogs, the quality of the test-time input is considerably higher for those systems. Thus we claim that the achievement of reconstructing a full range of dog breeds, with variable fur length, and varying shape and pose of ears, and with considerable occlusion, is a significant advance in the field's capabilities.

2 Related work

Paper	Animal Class	Training requirements	Template Model	Video required	Test Time Annotation	Model Fitting	Test Set Size
This paper	Dogs	2D Joints, Silhouettes, 3D Template, 3D Priors	SMAL	No	None	No	2000
3D-Safari [23]	Zebras, horses	3D models (albeit synthetic), 2D Joints, Silhouettes, 3D Priors	SMAL	3-7 frames / animal	None	Yes	200
Lions, Tigers and Bears (SMALR) [24]	MLQ	Not trained	SMAL	3-7 frames / animal	2D Joints, Silhouettes	Yes	14
3D Menagerie (SMAL) [25]	MLQ	Not trained	SMAL	No	2D Joints, Silhouettes	Yes	48
Creatures Great and SMAL [3]	MLQ	Not trained	SMAL	Yes	Silhouettes (for best results shown)	Yes	9
Category Specific Mesh Reconstructions [12]	Birds	2D Joints, Silhouettes	Sphere initialized to bird convex hull	No	None	No	2850
What Shape are Dolphins [4]	Dolphins, Pigeons	Not trained	Dolphin Template	25 frames / category	2D Joints, Silhouettes	Yes	25
Animated 3D Creatures [21]	MLQ	Not trained	Generalized Cylinders	Yes	2D Joints, Silhouettes	Yes	15

Table 1. Literature summary: Our paper extends large-scale "in-the-wild" reconstruction to the difficult class of diverse breeds of dogs. MLQ: Medium-to-large quadrupeds.

Monocular 3D reconstruction of human bodies. Extensive work has gone into tracking the 3D shape and pose of articulated subjects with complex skeletal configurations from single viewpoints. In particular, much attention has been diverted into the reconstruction of human bodies. Due to the commercial applications (e.g. virtual clothing try-on [7], VR meetings or for human-computer interaction), humans have been considered a special case, resulting large data collection exercises in this area.

This plentiful data supply has led to multiple data sources being available which provide strong supervisory signals for training deep neural networks. These include accurate 3D deformable template models (e.g. SMPL [18]), multiple 3D motion capture datasets (e.g. Human36M [9], 3DPW [19]), and large 2D datasets (e.g. COCO [17], LSP [10],MPII [2]) which provide keypoint and silhouette annotations. By comparison, in the context of animal reconstruction, which is the focus of this work, little to no 3D ground truth is available, and even datasets which provide 2D ground truth (e.g. COCO, Pascal [6]) contain an order of magnitude fewer annotations.

The abundance of human data has enabled the development of successful monocular 3D reconstruction pipelines [16, 11] that rely on the accurate 3D data to build complex priors over the distribution of human shapes and poses, while 2D keypoints and silhouettes significantly assist such models generalize to in-the-wild scenarios. For example, silhouette data has been shown to assist in accurate reconstruction of clothing [22, 1]; a task which mirrors the high shape variance encountered in animal reconstruction problems. However unlike our work, these methods have access to a much greater degree of 3D knowledge which is not typically available for the purpose of recovering the 3D shapes of animals.

While the dominant paradigm in human reconstruction is now end-to-end deep learning methods, SPIN [14] show impressive improvement by incorpo-

rating an energy minimization process within their training loop. Inspired by this innovation, we learn an ever-improving shape prior by applying expectation maximization during the training process.

3D reconstruction of quadruped animals.

While animals are often featured in computer vision literature, there are still relatively few works that focus on accurate 3D animal reconstruction. A primary reason for this is absence of large scale 3D datasets stemming from the practical challenges associated with 3D motion capture, as well as a lack of 2D data which captures a wide variety of animals.

Early work relied on fitting simple primitives such as cylinders to user-provided limb annotations. A significant advancement to the field came with the introduction of SMAL [25], a deformable 3D model (analagous to the SMPL model for human reconstruction), which was generated from 41 3D scans of toy figurines. This model was shown to provide accurate fits to a wide variety of quadruped animal species. Subsequent work showed that even broader animal categories could be reconstructed by incorporating multi-view constraints from video sequences [24]. Biggs et al.[3] use the SMAL model to build a synthetic dataset of animal silhouette images for training a joint predictor, before running a simpler optimization. A major drawback of these approaches is that they all rely on a slow test-time energy-based optimization procedure, making them unsuitable for real-time applications and susceptible to catastrophic failure given poor quality input data. Furthermore, these methods depend on user-provided keypoint or silhouette annotations.

Following recent trends in human reconstruction, Zuffi et al.[23] uses a deep network to recover detailed zebra shapes in the wild. This approach is still far from practical in general applications however, requiring an energy minimization step after network inference and relying on annotated video data for each individual training subject. By contrast our method is fast, requiring no additional energy-based refinement, and is trained purely from single in-the-wild images. In order to demonstrate that our method is able to capture a much more diverse set of shapes, we choose to focus our study on dogs, which exhibit a considerably greater variety of shapes and poses than the animals considered by [23].

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

3 Parametric animal model

3.1 SMAL

At the heart of our method is a parametric representation of a 3D animal mesh, which is based on the Skinned Multi-Animal Linear (SMAL) model proposed by [25]. SMAL is a deformable 3D animal mesh parameterized by shape and

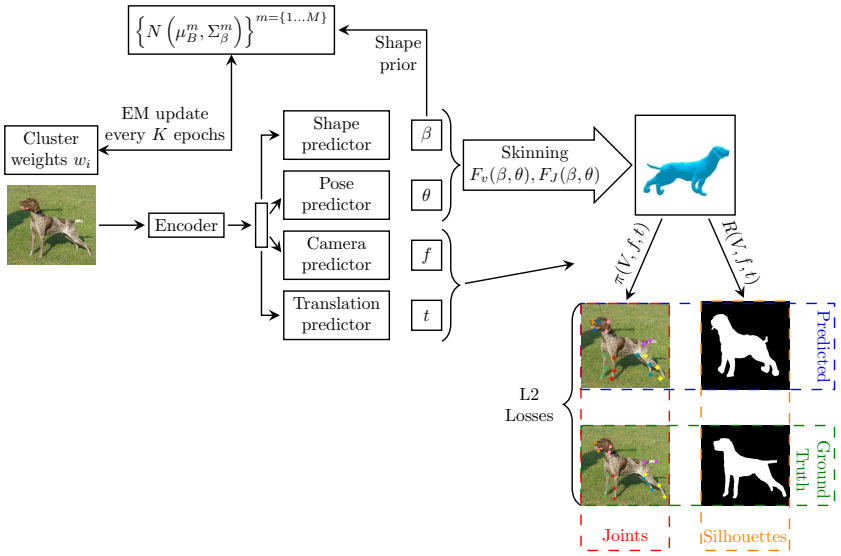


Fig. 2. System overview. Our method consists of (1) a deep CNN encoder which condenses the input image into a feature vector (2) a set of prediction heads which generate parameters for shape β, pose θ, camera focal length f and translation t (3) skinning functions F_v and F_J which construct the mesh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

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 which effect the articulated limb movement. The model consists of a linear blend skinning function $F_v : (\theta, \beta) \mapsto V$, which generates a set of vertex positions $V \in \mathbb{R}^{3889\times3}$, and a joint function $F_J : (\theta, \beta) \mapsto J$, which generates a set of joint positions $J \in \mathbb{R}^{35\times3}$.

3.2 Introducing scale parameters

While SMAL has been shown adequate for representing a variety of quadruped types, we find that the types of variation encountered in dog reconstruction is considerably more subtle than can adequately be captured by the current model. This unsurprising, given that the model is designed to be more general and only included four distinct artist impressions of dogs in its construction.

We therefore introduce a simple but effective way to improve the model's representational power over this particularly diverse and challenging animal category. We augment the set of shape parameters β with an additional set κ which have the effect of independently scaling different parts of the mesh. For each joint in the model, we define parameters $\kappa_x, \kappa_y, \kappa_z$ which apply a local scaling of the mesh along the local coordinate x, y, z axes, before pose is applied. Allowing each joint to scale entirely independently can however lead to unrealistic deformations, so we find it useful to share scale parameters between multiple joints, e.g. leg lengths. The new Skinned Multi-Breed Linear Model for Dogs

(SMBLD) is therefore adapted from SMAL by augmented the existing set of shape parameters with an additional 6 scale parameters. Figure 3 shows how introducing scale parameters increases the flexibility of the SMAL model.

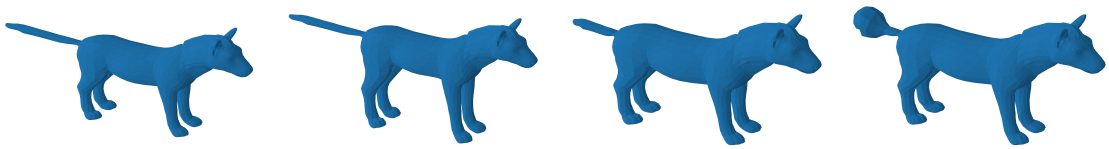


Fig. 3. Effect of varying new scale parameters. Use of shape parameters to modify SMAL mesh [mean, 25% longer legs, 50% shorter tail, poodle tail]

3.3 Learning a Unimodal 3D Prior via Fitting

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

We improve the quality of the prior (and learn a prior over our new scale parameters) by fitting to a set of 13 artist-designed 3D dog meshes, which are more varied than the original set. We apply an energy minimization scheme which aligns the SMAL vertices to each scan, under smoothing regularizers. Further details left to the supplementary.

4 End-to-end dog reconstruction from monocular images

We now take on the task of reconstructing a 3D dog mesh from a monocular image. We achieve this by training an end-to-end convolutional network that predicts a set of parameters for our SMBLD model together with perspective camera parameters. In particular, we train our network to predict pose θ and shape β parameters of SMBLD, translation t and camera focal length f for a perspective camera setup. A complete overview of the proposed system is shown in Figure 2.

4.1 Model architecture

Our network architecture is inspired by the model of Zuffi et al. [23] Given an input image cropped to (224, 224, 3), we apply a Resnet-50 [8] backbone network to encode 1024-dimensional feature map, which extended with a subsequent convolutional layer and two additional linear layers. The output of this stage

is a feature vector of size 1024. We then pass these features through various linear prediction heads to produce each of the required parameters. The pose, translation and camera prediction modules follow the design of Zuffi et al., but we describe differences in our shape module.

Pose, translation and camera prediction These modules are independent multi-layer perceptrons which map the above features to the various parameter types. As with Zuffi et al. we use two linear layers to map to a set of 35×3 3D pose parameters (three parameters for each joint in the SMBLD kinematic tree) given in Rodrigues form. We use independent heads to predict camera frame translation $t_{x,y}$ and depth t_z independently. We do not find the need to offset t_x from our data, so we simply assign these values directly from the network. We also take advantage of the perspective camera described in Zuffi et al., and obtain the camera focal length as $f = f_0 + f_1 x$, where x is the network output and $f_0 = f_1$ is fixed to a constant value.

Shape and scale prediction Unlike Zuffi et al. we design our network to directly predict the set of shape (including scale parameters), rather than vertex offsets. We observe improvement by handling the standard 20 blend-shape parameters and our new scaling parameters in separate prediction heads. Each uses a single linear layer to map the features to the desired output shape. We retrieve the scale parameters by $\kappa = \exp x$ where x is the network predictions, as we find predicting the log scale helps stabilise early training and conveniently ensures the scale parameters remain positive.

4.2 Training losses

A usual approach for training such an end-to-end system would be to supervise the prediction of (θ, β, t, f) with ground truth annotations collected on the dataset. However, in this work, we do not have any provision of such annotation. Instead, we must develop a method that relies on only *weak 2D supervision* to guide the network training.

In this section, we describe the set of losses used to supervise the network at train time.

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

$$L_{\text{joints}}(\theta, \beta, t, f; \hat{X}) = \|\hat{X} - \pi(F_J(\theta, \beta), t, f)\|_2 \quad (1)$$

Silhouette loss. The silhouette loss L_{sil} is primarily used to promote shape alignment between the SMBLD dog mesh and the input dog. In order to compute the silhouette loss, we define a rendering function $R : (\nu, t, f) \mapsto S$ which produces a segmentation mask by projecting the SMBLD model vertices. In order to allow derivatives to be propagated through R, we implement R using the differentiable Neural Mesh Renderer [13]. The loss is computed as the ℓ_2 difference between a projected silhouette and the ground truth $\hat{S}$:

$$L_{\text{sil}}(\theta, \beta, t, f; \hat{S}) = \|\hat{S} - R\big(F_V(\theta, \beta), t, f\big)\|_2 \tag{2}$$

Priors. In the absence of 3D ground truth training data, we rely on priors obtained from artist graphics models to ensure the produced 3D models remain realistic. We model both pose and shape using a multivariate Gaussian prior, consisting of a set of means $\mu_\theta, \mu\beta$ and covariance matrices $\Sigma_\theta, \Sigma\beta$.The loss is therefore given as as the log likelihood of a given shape or pose vector under these distributions, which corresponds to the Mahalanobis distance between the predicted parameters and their corresponding means:

$$L_{\text{pose}}(\theta; \mu_\theta, \Sigma_\theta) = (\theta - \mu_\theta)^T \Sigma_\theta^{-1} (\theta - \mu_\theta) \tag{3}$$

$$L_{\text{shape}}(\beta; \mu_\beta, \Sigma_\beta) = (\beta - \mu_\beta)^T \Sigma_\beta^{-1} (\beta - \mu_\beta) \tag{4}$$

Unlike previous work, we find there is no need to use a loss that penalizes pose parameters if they exceed manually specified joint angle limits. We suspect using a network to fit joints across an entire dataset rather than on a per-image basis provides natural regularisation that prevents infeasible joint configurations.

4.3 Learning a multi-modal shape prior.

Using a unimodal prior tends to result in predictions which look relatively similar in shape. To promote diversity among predicted 3D dog shapes, our method extends the formulation above to incorporate a mixture of Gaussians prior. We represent the mixture as a set of M Gaussians, whose means are initialized by drawing samples from our existing prior:

$$\mu_\beta^m \sim N(\mu_\beta, \Sigma_\beta) \tag{5}$$

$$\Sigma_\beta^m := \Sigma_\beta \tag{6}$$

We assign each training image i with a set of mixture weights $\{w_i^1, \dots w_i^M\}$, where initially $w_i^m := \frac{1}{M}$.

We can then apply the following mixture shape loss:

$$L_{mixture} = \sum_{m=1}^{M} w_i^m L_{shape}(\beta_i, \mu_\beta^m, \Sigma_\beta^m) \tag{7}$$

In order to allow our mixture prior to learn "in-the-loop" from the available training data, we apply expectation maximization every k epochs during training. This step recomputes the means and variances for each mixture component based on the observed shapes in the training set, and updates the per-image mixture weights:

$$\mu_\beta^m := \mathrm{E}_i[\beta_i W_i^m] \tag{8}$$

$$\Sigma_\beta^m := \mathrm{Cov}_i[\beta_i W_i^m, \beta_i W_i^m] \tag{9}$$

$$w_i^m := \frac{L_{shape}(\beta_i, \mu_\beta^m, \Sigma_\beta^m)}{\sum_{m'}^{M} L_{shape}(\beta_i, \mu_\beta^{m'}, \Sigma_\beta^{m'})} \tag{10}$$

5 Experiments

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

5.1 Stanford Dog Dataset with Joints and Silhouettes

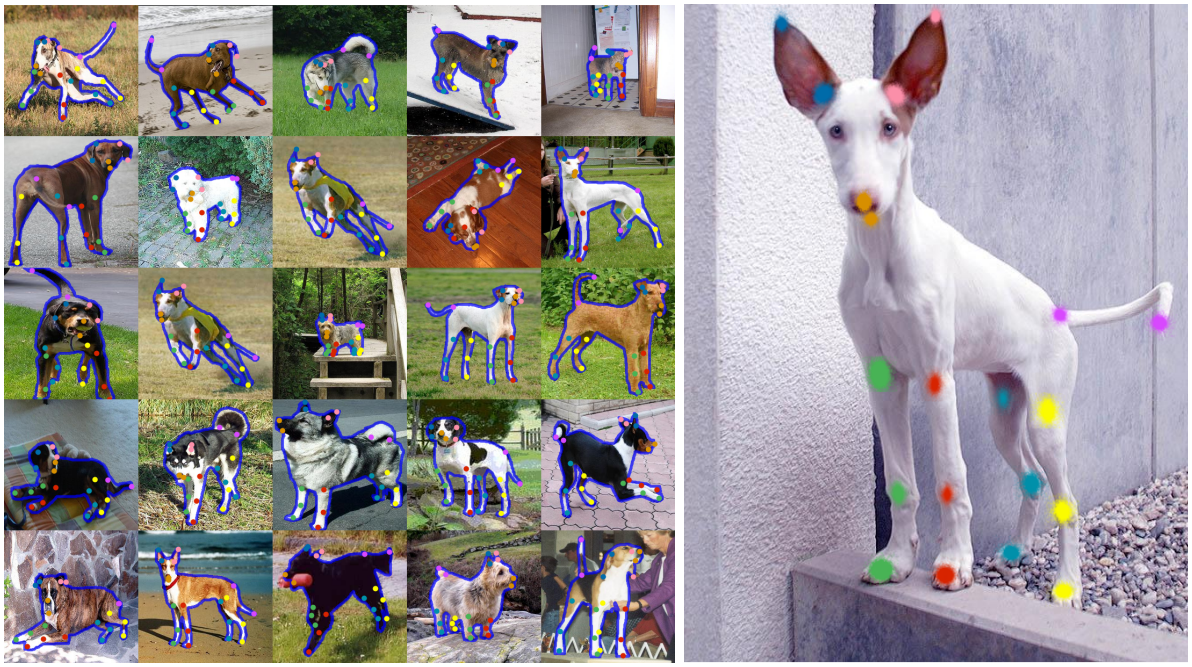


Fig. 4. Stanford Dog Dataset with Joints and Silhouettes - Left: outlined segmentations and labelled keypoints for 25 representative images. Right: heatmap of deviation of worker submitted results from mean for each submission.

In order to evaluate our method, we introduce the first large scale keypoint dataset for animals. In order to promote applicability to real-world scenarios,

we opted to take source images from the Stanford Dog Dataset, which consists of 20,580 dog images taken "in the wild" and covers 120 dog breeds. Figure 4 shows samples from the final dataset, showing the vast shape and pose variation between subjects, as well as complex environmental factors which typically hinder 3D reconstruction, such as occlusion, interaction with other objects/humans and partial views.

Since reconstruction methods for dogs should aim to capture the wide shape variation between subjects, we additionally collect silhouette masks per image.

We use the Amazon Mechanical Turk crowd-sourcing platform to collect a set of 20 keypoints per image; 3 per leg (knee, ankle, toe), 2 per ear (base, tip), 2 per tail (base, tip), 2 per face (nose and jaw). We also ask workers to produce a silhouette mask. We can approximate the difficulty of the dataset by analysing the variance between 3 annotators at both the joint labelling and silhouette task. Figure 4 shows the per-joint variance in joint labelling. Further details of the data curation procedure are left to the supplementary.

5.2 Evaluation protocol

Our evaluation is based on our new dataset. As is common with 3D reconstruction of articulated subject pipelines (e.g. SPIN [15], GraphCMR [16]), we remove images from the dataset for which a large proportion of the subject (over 50% of annotated keypoints) are invisible. We consider these examples unsuitable for our task of full dog reconstruction. In order to ensure our test set contains the largest possible variety of dog shapes, we divide each breed into an 80%/20% train and test split.

We consider two primary evaluation metrics. IoU is used to evaluate the accuracy of the 3D shape, by computing the intersection-over-union of the projected model silhouette vs. the ground truth annotation. 2D Percentage of Correct Keypoints (PCK) is used to evaluate the accuracy of the 3D pose, by computing the fraction of joints which are within a given distance from their corresponding ground truth, using a distance normalized by the area of the 2D silhouette.

5.3 Training procedure

We train our model in two stages. The first omits the silhouette loss which we find can lead the network to unsatisfactory local minima if applied to early. Since the silhouette loss which primarily guides accurate shape prediction is turned off, we find it adequate to use the simple unimodal prior (and also without EM) for this stage. After this, we introduce all losses, the mixture prior and begin applying the expectation maximization update. We train the first stage for 250 epochs, the second stage for 150 and apply the EM step every 15 epochs. All losses are weighted, as described in the supplementary. The entire training procedure takes around 4 days on a single P100 GPU.

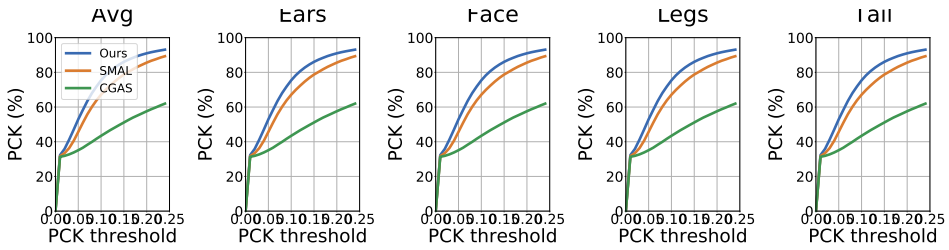


Fig. 5. PCK accuracy. On left PCK accuracy vs PCK threshold for each method is shown averaged over all test images for all joints, ears, face, legs and tail body parts. On right we illustrate the PCK accuracy for our method as a heatmap textured on the default shape model. Unlike SMAL our model is not provided with GT keypoints as input, yet can still produce much better 3D model joint alignment.

5.4 Comparison to baselines

We first compare our method to various baseline methods. SMAL [25] is an approach which fits the 3D SMAL model using per-image energy minimization. The method was originally demonstrated using hand-clicked keypoints and segmentation masks, which does not make a fair comparison to our method since we assume no user interaction at test time. We provide additional comparisons where the SMAL method is applied to automatically detected joints and silhouettes. To obtain these automatic results we make use of the Stacked Hourglass Network [20] for joint prediction and DeepLab v3+ [5] for segmentation.

We train the Stacked Hourglass Network with 8 stacks and 1 block and DeepLab v3+ trained on the joint and silhouette annotations provided by our Stanford Dog training set. We also train the Creatures Great and SMAL (CGAS) method [3], using a dataset of synthetic dog silhouette renderings where the pose and shape parameters are drawn from the SMAL pose prior as described in their paper. All methods are trained from scratch and evaluated on our Stanford Dog validation set. The results are shown in Table 2.

Method	IoU	PCK
DeepLab v3+	83.4	N/A
HourglassNet	N/A	71.4
CGAS Joint Predictor	N/A	21.8

Table 2. Competitive Networks. We evaluate the performance of Stacked Hourglass Network with 8 stacks and 1 block and DeepLab v3+ trained on our Stanford Dog training set and evaluated on the test set. We train CGAS on the synthetic data and evaluate on ground truth segmentations in the Stanford Dog test set.

Table 3 shows the comparison between our method and these competitive methods. In order to fully examine the competition, we additionally provide

evaluation of SMAL and CGAS in their original setting when ground-truth keypoints and/or segmentations are provided at test time.

The results show that our end-to-end method incorporating in-the-loop expectation maximization outperforms all comparative methods (e.g. when evaluating others on predicted methods) across all categories, resulting in a new state-of-the-art in this task. Further, we show our method improves over these methods in terms of average silhouette IoU score and mean PCK 2D joint accuracy even when competitive methods are provided ground truth annotations at test time. It is also worth noting that since both SMAL and CGAS methods incorporate energy minimization phases, our method is significantly faster. We also show that augmenting the original SMAL model with two of the contributions from our work: additional scale shape parameters and an improved prior; leads to significant improvements compared to the model proposed by [25].

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

Table 3. Baseline comparisons. Both PCK and silhouette IOU scores are shown for SOTA methods under varying conditions. A combination of both ground truth (GT) and predicted (Pred) keypoints/segmentations using hourglass network and deeplab respectively. For the CGAS method we also test using their keypoint predictor (CGAS). The addition of scaling and a new prior is shown to be superior over the standard SMAL model.

5.5 Ablation Study

In order to show the contribution of individual components of our method, we conduct an ablation study which assesses the performance of our method when certain parts are removed.

We evaluate three variants, (1) **Ours w/o EM** that doesn't perform the EM update, (2) **Ours w/o MoG** which replaces our mixture shape prior with a unimodal prior, (3) **Ours w/o Scale** which removes the scale parameters.

The results in 4 show that each individual component has a significant impact on the overall performance of our method. In particular, it can be seen that the inclusion of the EM and mixture of Gaussians term leads to a substantial improvement in IoU, suggesting that the model is able to more accurately fit

to the exact shape of the dog. Incorporating the additional scale parameters has a significant effect on both the IoU performance and the keypoint detection accuracy.

Method	IoU	Legs	Tail	Ears	Face	Avg
Ours	**73.6**	**75.0**	**77.6**	**6**9.9	90.0	**75.7**
Ours w/o EM	67.7	72.9	75.2	**72.5**	88.3	74.6
Ours w/o MoG	68.0	74.3	73.3	70.0	**90.2**	74.9
Ours w/o Scale	67.3	72.9	75.3	62.3	89.1	72.6

Table 4. Ablation study. Demonstration of how our methods perform when the following are removed: (a) Expectation Maximization updates, (b) Mixture Shape Prior, (c) Removing SMBLD scale parameters.

5.6 Qualitative evaluation

We provide a qualitative evaluation of our methods in Figure 7, in which we show example outputs of our system when tested on a range of dogs with varying pose and shape.

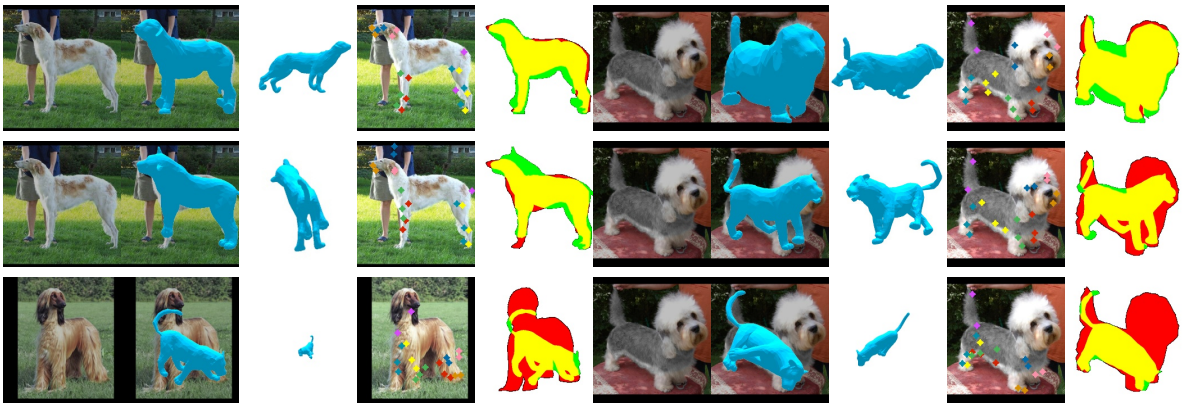


Fig. 6. State of the art comparison Model fitting, keypoint and silhouette reprojection error are illustrated on two example images for each of the methods: Ours, SMAL and CGAS on each row respectively. From left to right: input image, registration of 3D mesh, mesh from alternative viewpoint, mesh keypoint projections, intersections (green) with silhouette projections from mesh (green) and ground truth (red). Notice the failures of SMAL due to restricted prior and CGAS for inaccurate keypoint localisation. Our method is capable of recovering the dog shape more accurately without the need for vertex deformations.

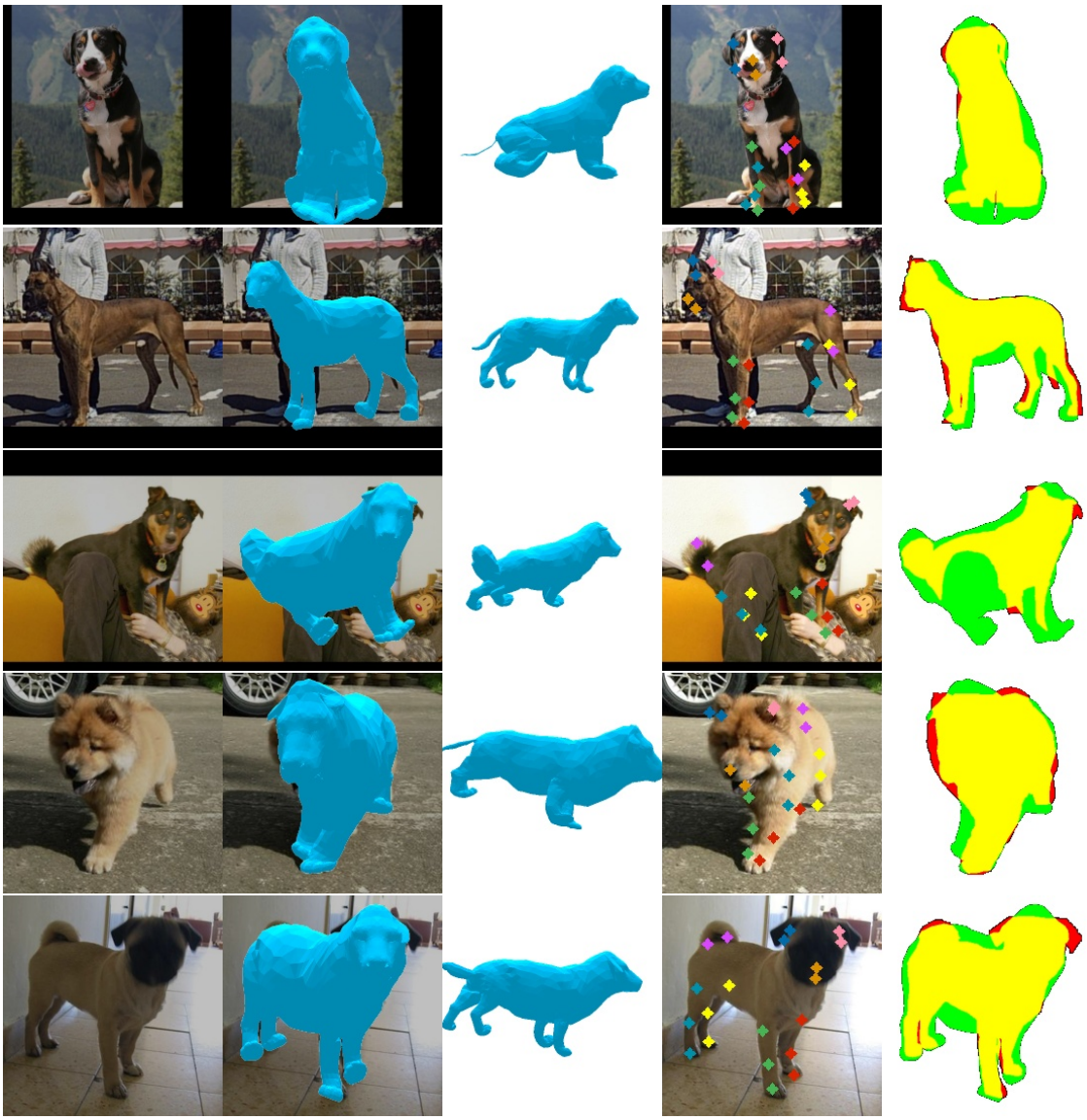


Fig. 7. Qualitative results on Stanford Dog Dataset For each sample we show from left to right: input image, predicted 3D mesh, reversed 3D mesh, joint reprojection error and silhouette reprojection error.

6 Conclusions

In this paper, we have presented an end-to-end method for automatic 3D dog reconstruction from monocular image input. We achieve this using only 2D losses, which we apply on annotations collected on the Stanford Dog Dataset. Further, we show we can learn a more detailed shape prior by tuning a gaussian mixture during training to achieve better reconstructions. We have demonstrated our method's superior performance over the current state-of-the-art, even when competitive methods are given access to ground truth data at test time.

Future work should involve extending our EM "in-the-loop" process to video input, in which further shape constraints can be applied. We also see potential in transferring the knowledge over pose and shape accumulated on our dog dataset to other species, preferably with few annotations from the new target domain.

References

1. Alldieck, T., Magnor, M., Bhatnagar, B.L., Theobalt, C., Pons-Moll, G.: Learning to reconstruct people in clothing from a single rgb camera. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1175–1186 (2019)
2. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.:
3. Biggs, B., Roddick, T., Fitzgibbon, A., Cipolla, R.: Creatures great and SMAL: Recovering the shape and motion of animals from video. In: ACCV (2018)
4. Cashman, T.J., Fitzgibbon, A.W.: What shape are dolphins? Building 3D morphable models from 2D images. IEEE transactions on pattern analysis and machine intelligence **35**(1), 232–244 (2013)
5. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. CoRR **abs/1606.00915** (2016), http://dblp.uni-trier.de/db/journals/corr/corr1606.htmlChenPK0Y16
6. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International journal of computer vision **88**(2), 303–338 (2010)
7. Han, X., Wu, Z., Wu, Z., Yu, R., Davis, L.S.: Viton: An image-based virtual try-on network. In: CVPR (2018)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
9. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. IEEE transactions on pattern analysis and machine intelligence **36**(7), 1325–1339 (2013)
10. Johnson, S., Everingham, M.: Clustered pose and nonlinear appearance models for human pose estimation. In: Proceedings of the British Machine Vision Conference (2010), doi:10.5244/C.24.12
11. Kanazawa, A., Black, M.J., Jacobs, D.W., Malik, J.: End-to-end recovery of human shape and pose. In: Computer Vision and Pattern Regognition (CVPR) (2018)
12. Kanazawa, A., Tulsiani, S., Efros, A.A., Malik, J.: Learning category-specific mesh reconstruction from image collections. In: European Conference on Computer Vision. pp. 371–386 (2018)
13. Kato, H., Ushiku, Y., Harada, T.: Neural 3d mesh renderer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3907–3916 (2018)
14. Kolotouros, N., Pavlakos, G., Black, M.J., Daniilidis, K.: Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2252–2261 (2019)
15. Kolotouros, N., Pavlakos, G., Black, M.J., Daniilidis, K.: Learning to reconstruct 3D human pose and shape via model-fitting in the loop. In: Proc. ICCV (2019)
16. Kolotouros, N., Pavlakos, G., Daniilidis, K.: Convolutional mesh regression for single-image human shape reconstruction. In: Proc. CVPR (2019)
17. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollar, P., Zitnick, L.: Microsoft COCO: Common objects in context. In: ECCV. European Conference on Computer Vision (September 2014), https://www.microsoft.com/en-us/research/publication/microsoft-coco-common-objects-in-context/

18. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: A skinned multi-person linear model. ACM Trans. Graphics (Proc. SIGGRAPH Asia) **34**(6), 248:1–248:16 (Oct 2015)
19. von Marcard, T., Henschel, R., Black, M., Rosenhahn, B., Pons-Moll, G.: Recovering accurate 3d human pose in the wild using imus and a moving camera. In: European Conference on Computer Vision (ECCV) (sep 2018)
20. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: European Conference on Computer Vision. pp. 483–499. Springer (2016)
21. Reinert B, Ritschel T, S.H.P.: Animated 3d creatures from single-view video by skeletal sketching. In: Proc. Graphics Interface (2016)
22. Saito, S., , Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., Li, H.: Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. arXiv preprint arXiv:1905.05172 (2019)
23. Silvia Zuffi, Angjoo Kanazawa, T.B.W.M.J.B.: Three-d safari: Learning to estimate zebra pose, shape, and texture from images "in the wild". In: The IEEE International Conferene on Computer Vision (ICCV) (2019)
24. Zuffi, S., Kanazawa, A., Black, M.J.: Lions and tigers and bears: Capturing non-rigid, 3D, articulated shape from images. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society (2018)
25. Zuffi, S., Kanazawa, A., Jacobs, D., Black, M.J.: 3D menagerie: Modeling the 3D shape and pose of animals. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (Jul 2017)