RGBD-Dog: Predicting Canine Pose from RGBD Sensors

Sinéad Kearney [1] Wenbin Li [1] Martin Parsons [1] Kwang In Kim [2] Darren Cosker [1]

[1]University of Bath [2]UNIST

{s.kearney,w.li,m.m.parsons,d.p.cosker}@bath.ac.uk kimki@unist.ac.kr

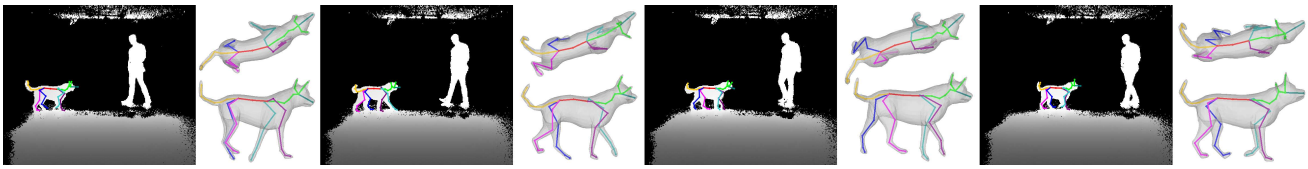

Figure 1: We present a system to predict the skeleton pose of a dog from RGBD images. If the size and shape of the dog is unknown, an estimation is provided. Displayed here are frames [4,7,13,18] of a Kinect sequence, showing 2D projection, 3D skeleton and skinned mesh as produced by the pipeline. All figures in this paper are most informative when viewed in colour.

Abstract

The automatic extraction of animal 3D pose from images without markers is of interest in a range of scientific fields. Most work to date predicts animal pose from RGB images, based on 2D labelling of joint positions. However, due to the difficult nature of obtaining training data, no ground truth dataset of 3D animal motion is available to quantitatively evaluate these approaches. In addition, a lack of 3D animal pose data also makes it difficult to train 3D pose-prediction methods in a similar manner to the popular field of body-pose prediction. In our work, we focus on the problem of 3D canine pose estimation from RGBD images, recording a diverse range of dog breeds with several Microsoft Kinect v2s, simultaneously obtaining the 3D ground truth skeleton via a motion capture system. We generate a dataset of synthetic RGBD images from this data. A stacked hourglass network is trained to predict 3D joint locations, which is then constrained using prior models of shape and pose. We evaluate our model on both synthetic and real RGBD images and compare our results to previously published work fitting canine models to images. Finally, despite our training set consisting only of dog data, visual inspection implies that our network can produce good predictions for images of other quadrupeds – e.g. horses or cats – when their pose is similar to that contained in our training set.

1. Introduction

While pose estimation has traditionally focused on human subjects, there has been an increased interest on animal subjects in recent years ([7], [3], [37], [38]). It is possible to put markers on certain trained animals such as dogs to employ marker-based motion capture techniques. Nevertheless, there are far more practical difficulties associated with this when compared with human subjects. Some animals may find markers distressing and it is impossible to place them on wild animals. Neural networks currently achieve the best results for human pose estimation, and generally require training on widely available large-scale data sets that provide 2D and/or 3D annotations ([33], [1], [15], [16]). However, there are currently no datasets of 3D animal data available at the same scale concerning the number of samples, variety and annotations, making comparable studies or approaches to pose prediction difficult to achieve.

In this paper, we propose a markerless approach for 3D skeletal pose-estimation of canines from RGBD images. To achieve this, we present a canine dataset which includes skinned 3D meshes, as well as synchronised RGBD video and 3D skeletal data acquired from a motion capture system which acts as ground truth. Dogs are chosen as our capture subject for several reasons: they are familiar with human contact and so generally accept wearing motion capture suits; they can be brought into the motion capture studio with ease; they respond to given directions producing comparable motions across the numerous subjects; their diverse body shape and size produces data with interesting variations in shape. We propose that our resulting dog skeleton structure is more anatomically correct when compared with that of the SMAL model and a larger number of bones in the skeleton allows more expression.

It is challenging to control the capture environment with (uncontrolled) animals - covering wide enough variability in a limited capture session proved to be challenging. Hence our method utilises the dog skeletons and meshes produced by the motion capture system to generate a large synthetic

dataset. This dataset is used to train a predictive network and generative model using 3D joint data and the corresponding projected 2D annotations. Using RGB images alone may not be sufficient for pose prediction, as many animals have evolved to blend into their environment and similarly coloured limbs can result in ambiguities. On the other hand, depth images do not rely on texture information and give us the additional advantage of providing surface information for predicting joints. We choose to use the Microsoft Kinect v2 as our RGBD depth sensor, due to its wide availability and the established area of research associated with the device. Images were rendered from our synthetically generated 3D dog meshes using the Kinect sensor model of Li et al. [20] to provide images with realistic Kinect noise as training data to the network.

Details of the dataset generation process are provided in Section 3.2. Despite training the network with purely synthetic images, we achieve high accuracy when tested on real depth images, as discussed in Section 4.1. In addition to this, Section 4.3, we found that training the network only with dogs still allowed it to produce plausible results on similarly rendered quadrupeds such as horses and lions.

The joint locations predicted by deep networks may contain errors. In particular, they do not guarantee that the estimated bone lengths remain constant throughout a sequence of images of the same animal and may also generate physically impossible poses. To address these limitations, we adopt a prior on the joint pose configurations – a Hierarchical Gaussian Process Latent Variable Model (H-GPLVM) [18]. This allows the representation of high-dimensional non-linear data in lower dimensions, while simultaneously exploiting the skeleton structure in our data. In summary, our main contributions are:

- Prediction of 3D shape as PCA model parameters, 3D joint locations and estimation of a kinematic skeleton of canines using RGBD input data.
- Combination of a stacked hour glass CNN architecture for initial joint estimation and a H-GPLVM to resolve pose ambiguities, refine fitting and convert joint positions to a kinematic skeleton.
- A novel dataset of RGB and RGBD canine data with skeletal ground truth estimated from a synchronised 3D motion capture system and a shape model containing information of both real and synthetic dogs. This dataset and model are available at [1].

2. Related work

2D Animal Pose Estimation. Animal and insect 2D pose and position data is useful in a range of behavioural studies. Most solutions to date use shallow trained neural network architectures whereby a few image examples of the animal or insect of interest are used to train a keyframe-based feature tracker, e.g. LEAP Estimates Animal Pose [28], DeepLabCut ([22], [26]) and DeepPoseKit ([12]). Cao et al. [7] address the issue of the wide variation in interspecies appearance by presenting a method for cross-domain adaption when predicting the pose of unseen species. By creating a training dataset by combining a large dataset of human pose (MPII Human Pose [2]), the bounding box annotations for animals in Microsoft COCO [21], and the authors' animal pose dataset, the method achieves good pose estimation for unseen animals.

3D Animal Pose Estimation. Zuffi et al. [39] introduce the Skinned Multi-Animal Linear model (SMAL), which separates animal appearance into PCA shape and pose-dependent shape parameters (e.g. bulging muscles), created from a dataset of scanned toy animals. A regression matrix calculates joint locations for a given mesh. SMAL with Refinement (SMALR) [38] extends the SMAL model to extract fur texture and achieves a more accurate shape of the animal. In both methods, silhouettes are manually created when necessary, and manually selected keypoints guide the fitting of the model. In SMAL with learned Shape and Texture (SMALST) [37] a neural network automatically regresses the shape parameters, along with the pose and texture of a particular breed of zebra from RGB images, removing the requirement of silhouettes and keypoints.

Biggs et al. [3] fit the SMAL model to sequences of silhouettes that have been automatically extracted from the video using Deeplab [8]. A CNN is trained to predict 2D joint locations, with the training set generated using the SMAL model. Quadractic programming and genetic algorithms choose the best 2D joint positions. SMAL is then fit to the joints and silhouettes.

In training our neural network, we also generate synthetic RGBD data from a large basis of motion capture data recorded from the real motion of dogs as opposed to the SMAL model and its variants where the pose is based from toy animals and a human-created walk cycle.

Pose Estimation with Synthetic Training Data. In predicting pose from RGB images, it is generally found that training networks with a combination of real and synthetic images provides a more accurate prediction than training with either real or synthetic alone ([35], [9], [29]). Previous work with depth images has also shown that synthetic training alone provides accurate results when tested on real images [17]. Random forests have been used frequently for pose estimation from depth images. These include labelling pixels with human body parts ([32]), mouse body parts ([25]) and dense correspondences to the surface mesh of a human model ([34]). Sharp et al. [31] robustly track a hand in real-time using the Kinect v2.

[1] https://github.com/CAMERA-Bath/RGBD-Dog.

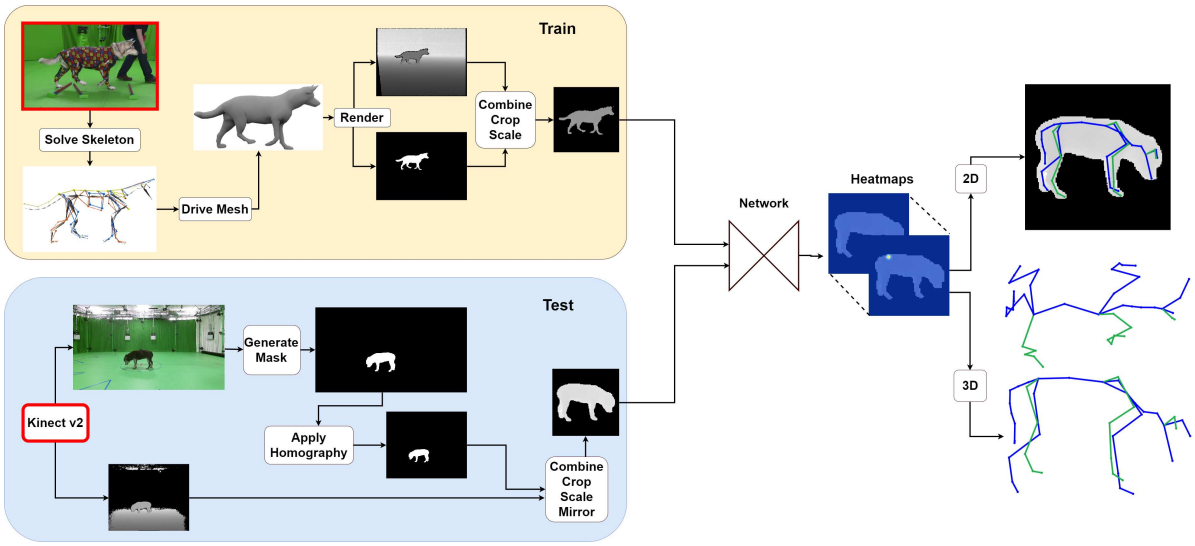

Figure 2: Overview of the network section of our pipeline. In the training stage, a synthetic dataset is generated from dog motion data. A pair of images is rendered for each frame: depth images are rendered using the Kinect model of InteriorNet [21] and silhouette masks rendered using OpenGL. In the testing stage, the RGB Kinect image is used to generate a mask of the dog, which is then applied to the depth Kinect image and fed into the network. The network produces a set of 2D heatmaps from which the 2D and 3D joint locations are extracted.

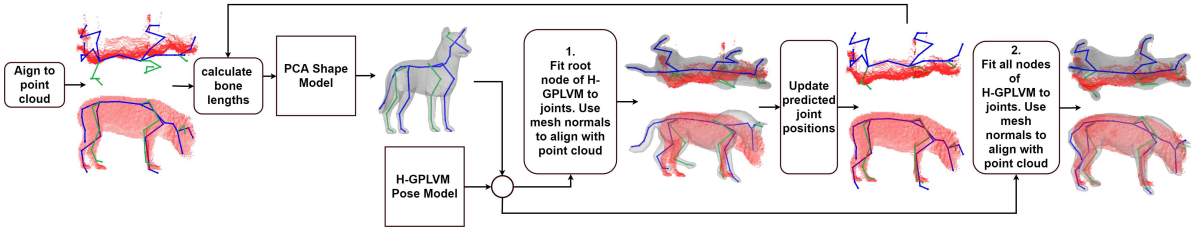

Figure 3: Overview of the refinement section of our pipeline, showing the steps taken when the dog's neutral body shape is unknown. The point cloud from the depth image initialises the scale of the skeleton and a PCA model predicts body shape from the bone lengths. The H-GPLVM is used to estimate a rough pose of the dog mesh, with the mesh normals then used to refine the mesh/point cloud alignment. The dog scale is refined, the PCA model produces the final shape prediction, and the H-GPVLM fully fits the skinned dog mesh to the point cloud. For known shapes, the PCA prediction steps are not required.

Recently, neural networks have also been used in pose estimation from depth images. Huang & Altamar [14] generate a dataset of synthetic depth images of human body pose and use this to predict the pose of the top half of the body. Mueller et al. [24] combine two CNNs to locate and predict hand pose. A kinematic model is fit to the 3D joints to ensure temporal smoothness in joint rotations and bone lengths are consistent across the footage.

In our work, we use motion capture data from a selection of dogs to generate a dataset of synthetic depth images. This dataset is used to train a stacked hourglass network, which predicts joint locations in 3D space. Given the joints predicted by the network, a PCA model can be used to predict the shape of an unknown dog, and a H-GPLVM is used to constrain the joint locations to those which are physically plausible. We believe ours is the first method to train a neural network to predict 3D animal shape and pose from RGBD images, and to compare our pipeline results to 3D ground truth which is difficult to obtain for animals and has therefore as yet been unexplored by researchers.

3. Method

Our pipeline consists of two stages; a prediction stage and refinement stage. In the prediction stage, a stacked hourglass network by Newell et al. [27] predicts a set of 2D heatmaps for a given depth image. From these, 3D joint positions are reconstructed. To train the network, skeleton motion data was recorded from five dogs performing the same five actions using a Vicon optical motion capture system (Section 3.1). These skeletons pose a mesh of the

Figure 4: Dogs included in our dataset, each wearing a motion capture suit. The two dogs on the left were used for test footage only.

respective dog which are then rendered as RGBD images by a Kinect noise-model to generate a large synthetic training dataset (Section 3.2). We provide more detail about the network training data and explain 3D joint reconstruction from heatmaps in Section 3.3. In the refinement stage, a H-GPLVM [19] trained on skeleton joint rotations is used to constrain the predicted 3D joint positions (Section 3.4). The resulting skeleton can animate a mesh, provided by the user or generated from a shape model, which can then be aligned to the depth image points to further refine the global transformation of the root of the skeleton. We compare our results with the method of Biggs et al. [3] and evaluate our method with ground truth joint positions in synthetic and real images in Section 4. Figures 2 and 3 outline the prediction and refinement stages of our approach respectively.

3.1. Animal Motion Data Collection

As no 3D dog motion data is available for research, we first needed to collect a dataset. A local rescue centre provided 16 dogs for recording. We focused on five dogs that covered a wide range of shape and size. The same five actions were chosen for each dog for the training/validation set, with an additional arbitrary test sequence also chosen for testing. In addition to these five dogs, two dogs were used to evaluate the pipeline and were not included in the training set. These dogs are shown in Figure 4.

A Vicon system with 20 infrared cameras was used to record the markers on the dogs' bespoke capture suits. Vicon recorded the markers at 119.88 fps, with the skeleton data exported at 59.94 fps. Up to 6 Kinect v2s were also simultaneously recording, with the data extracted using the libfreenect2 library [4]. Although the Kinects recorded at 30fps, the use of multiple devices at once reduced overall frame rate to 6fps in our ground truth set. However, this does not affect the performance of our prediction network. Further details on recording can be found in the supplementary material (Sec. 2.1).

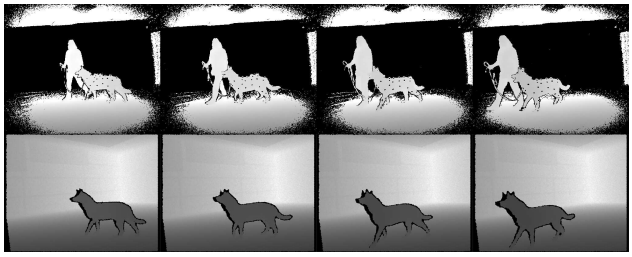

Figure 5: A comparison of a sequence of real Kinect v2 images (top) with those produced by InteriorNet [20] (bottom), where all images have been normalised.

3.2. Synthetic RGBD Data Generation

Our template dog skeleton is based on anatomical skeletons [11]. Unlike humans, the shoulders of dogs are not constrained by a clavicle and so have translational as well as rotational freedom [10]. The ears are modelled with rigid bones and also given translational freedom, allowing the ears to move with respect to the base of the skull. In total, there are 43 joints in the skeleton, with 95 degrees of freedom. The neutral mesh of each dog was created by an artist, using a photogrammetric reconstruction as a guide. Linear blend skinning is used to skin the mesh to the corresponding skeleton, with the weights also created by an artist.

To create realistic Kinect images from our skinned 3D skeletons, we follow a similar process from InteriorNet [20]. Given a 3D mesh of a dog within virtual environment, we model unique infrared dot patterns projected on to the object, and further achieve dense depth using stereo reconstruction. This process is presumed to retain most of characteristics of Kinect imaging system including depth shadowing and occlusion. A comparison of real versus synthetic Kinect images is shown in Figure 5.

Up to 30 synthetic cameras were used to generate the depth images and corresponding binary mask for each dog. Details of the image and joint data normalisation for the generation of ground truth heatmaps are given in the supplementary material. The size of the dataset is doubled by using the mirrored version of these images, giving a total number of 650,000 images in the training set and 180,000 images in the validation set. An overview of data generation can be seen in the "Train" section of Figure 2.

3.3. Skeleton Pose Prediction Network

In order to use the stacked-hourglass framework, we represent joints as 2D heatmaps. Input to the network are 256x256 greyscale images, where 3D joints J_{3D256} are defined in this coordinate space. Given an input image, the network produces a set of 129 heatmaps H, each being 64x64 pixels in size. Each joint j in the dog skeleton is associated with three heatmaps, the indices of which is

known: $h_{jXY}, h_{jYZ}, h_{jXZ}$, representing the xy-, yz- and xz-coordinates of j respectively. This set provided the most accurate results in our experiments. To produce the heatmaps required to train the network, J_{3D256} are transformed to a 64x64 image coordinates. Let J_{3D64} be these transformed coordinates, where $J_{3D64} = floor(J_{3D256}/4) + 1$. We generate 2D gaussians in the heatmaps centred at the xy-, yz- and xz-coordinates of J_{3D64}, with a standard deviation of one pixel. Inspired by Biggs et al. [3], symmetric joints along the sagittal plane of the animal (i.e. the legs and ears) produce multi-model heatmaps. Further technical details on heatmap generation may be found in the suplementary material.

Our neural network is a 2-stacked hourglass network by Newell et al. [27]. This particular network was chosen as the successive stages of down-sampling and up-scaling allow the combination of features at various scales. By observing the image at global and local scales, the global rotation of the subject can be more easily determined, and the relationship between joints can be utilised to produce more accurate predictions. We implement our network using PyTorch, based on code provided by Yang [36]. RMSprop is used as the optimiser, with a learning rate of 0.0025 and batch size 6. Our loss function is the MSE between the ground truth and network-generated heatmaps.

3.3.1 3D Pose Regression from 2D Joint Locations

Given the network-generated heatmaps, we determine the value of J_{3D64}, the location of each joint in the x-, y-, and z-axis in 64x64 image coordinates. Each joint j is associated with three heatmaps: $h_{jXY}, h_{jYZ}, h_{jXZ}$. For joints that produce unimodal heatmaps, the heatmap with the highest peak from the set of $h_{jXY}, h_{jYZ}, h_{jXZ}$ determines the value of two of the three coordinates, with the remaining coordinate taken from the map with the second highest peak.

For joints with multi-modal heatmaps, we repeat this step referring first to the highest peak in the three heatmaps, and then to the second highest peak. This process results in two potential joint locations for all joints that form a symmetric pair (j_{p1}, j_{p2}). If the XY position of the predicted coordinate of j_{p1} is within a threshold of the XY position of j_{p2}, we assume that the network has erroneously predicted the same position for both joints. In this case, the joint with the highest confidence retains this coordinate, and the remaining joint is assigned its next most likely joint.

Once J_{3D64} has been determined, the coordinates are transformed into J_{3D256}. Prior to this step, as in Newell et al. [27], a quarter pixel offset is applied to the predictions in J_{3D64}. We first determine, within a 4-pixel neighbourhood of each predicted joint, the location of the neighbour with the highest value. This location dictates the direction of the offset applied. The authors note that the addition of this offset increases the joint prediction precision. Finally, J_{3D64} is scaled to fit a 256x256 image, resulting in J_{3D256}. The image scale and translation acquired when transforming the image for network input is inverted and used to transform the xy-coordinates of J_{3D256} into J_{2Dfull}, the projections in the full-size image. To calculate the depth in camera space for each joint in J_{3D256}, the image and joint data normalisation process is inverted and applied. J_{2Dfull} is transformed into J_{3Dcam} using the intrinsic parameters of the camera and the depth of each predicted joint.

3.4. Pose Prior Model

While some previous pose models represent skeleton rotations using a PCA model, such as the work by Safonova et al. [30], we found that this type of model produces poses that are not physically possible for the dog. In contrast, a Gaussian Process Latent Variable Model (GPLVM) [18] can model non-linear data and allows us to represent our high dimensional skeleton on a low dimensional manifold. A Hierarchical GPLVM (H-GPLVM) [19] exploits the relationship between different parts of the skeleton. Ear motion is excluded from the model. As ears are made of soft tissue, they are mostly influenced by the velocity of the dog, rather than the pose of other body parts. This reduces the skeleton to from 95 to 83 degrees of freedom. Bone rotations are represented as unit quaternions, and the translation of the shoulders are defined with respect to their rest position. Mirrored poses are also included in the model. The supplementary material contains further technical specifications for our hierarchy (Sec. 2.3).

We remove frames that contain similar poses to reduce the number of frames included in the training set S. The similarity of two quaternions is calculated using the dot product, and we sum the results for all bones in the skeleton to give the final similarity value. Given a candidate pose, we calculate the similarity between it and all poses in S. If the minimum value for all calculations is above a threshold, the candidate pose is added to S. Setting the similarity threshold to 0.1 reduces the number of frames in a sequence by approximately 50-66%. The data matrix is constructed from S and normalised. Back constraints are used when optimising the model, meaning that similar poses are located in close proximity to each other in the manifold.

3.4.1 Fitting the H-GPLVM to Predicted Joints

A weight is associated with each joint predicted by the network to help guide the fitting of the H-GPLVM. Information about these weights is given in the supplementary material. To find the initial coordinate in the root node of H-GPLVM, we use k-means clustering to sample 50 potential coordinates. Keeping the root translation fixed, we find the rotation which minimises the Euclidean distance between the

network-predicted joints and the model-generated joints. The pose and rotation with the smallest error is chosen as the initial values for the next optimisation step.

The H-GPLVM coordinate and root rotation are then refined. In this stage, joint projection error is included, as it was found this helped with pose estimation if the network gave a plausible 2D prediction, but noisy 3D prediction. The vector generated by the root node of the model provides the initial coordinates of the nodes further along the tree. All leaf nodes of the model, root rotation and root translation are then optimised simultaneously.

During the fitting process, we seek to minimise the distance between joint locations predicted by the network and those predicted by the H-GPLVM: Equation 1 defines the corresponding loss function:

$$\mathcal{L}(X,R,T,t) = \sum_{b=1}^{B} \gamma_b \left\| j_b - F(X,R,T,t)_b \right\| + \lambda \sum_{b=1}^{B} \gamma_b \left\| \Phi(j_b) - \Phi(F(X,R,T,t)_b) \right\| . \quad (1)$$

Here, B is the number of joints in the skeleton, $J = [j_1, ..., j_b]$ is the set of predicted joint locations from the network, $\Gamma = [\gamma_1, ..., \gamma b]$ is the set of weights associated with each joint, Φ is the perspective projection function and λ is the influence of 2D information when fitting the model. Let X be the set of n-dimensional coordinates for the given node(s) of the H-GPLVM and F be the function that takes the set X, root rotation R, root translation T, shoulder translations t, and produces a set of 3D joints. Figure 3 shows the result of process.

4. Evaluation and Results

To evaluate our approach, we predict canine shape and pose from RGBD data on a set of five test sequences, one for each dog. Each sequence was chosen for the global orientation of the dogs to cover a wide range, both side-views and foreshortened views, with their actions consisting of a general walking/exploring motion. In each case we predict shape and pose and compare these predictions to ground truth skeletons as obtained from a motion-capture system (see Section 3.1). More detailed analysis of experiments as well as further technical details of experimental setup – as well as video results - may be found in the supplementary material.

As no previous methods automatically extract dog skeleton from depth images, we compare our results with Biggs et al. [3], which we will refer to as the BADJA result. We note that the authors' method requires silhouette data only and therefore it is expected that our method produces the more accurate results. Both algorithms are tested on noise-free images. We use two metrics to measure the accuracy of our system: Mean Per Joint Position Error (MPJPE)

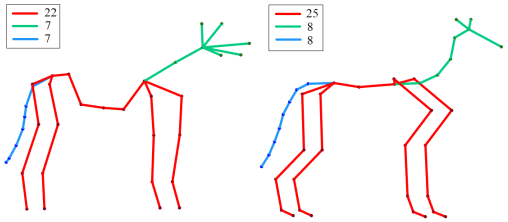

Figure 6: The number of joints in each skeleton group when evaluating the predicted skeleton against the ground truth skeleton. Left: the SMAL skeleton used by BADJA [3], and right: our skeleton.

and Probability of Correct Keypoint (PCK). MPJPE measures Euclidean distance and is calculated after the roots of the two skeletons are aligned. A variant PA MPJPE uses Procrustes Analysis to align the predicted skeleton with the ground truth skeleton. PCK describes the situation whereby the predicted joint is within a threshold from the true value. The threshold is $\alpha * A$, where A is the area of the image with non-zero pixel values and α = 0.05. The values range from [0,1], where 1 means that all joints are within the threshold. PCK can also be used for 3D prediction [23], where the threshold is set to half the width of the person's head. As we can only determine the length of the head bone, we set the threshold to one and we scale each skeleton such that the head bone has a length of two units. To compare the values of MPJPE and PCK 3D, we also use PA PCK 3D, where the joints are aligned as in PA MPJPE, and then calculate PCK 3D. Due to the frequent occlusion of limbs of the dogs, the errors are reported in the following groups: *All* – all joints in the skeleton; *Head* – the joints contained in the neck and head; *Body* – the joints contained in the spine and four legs; – *Tail*: the joints in the tail. Figure 6 shows the configuration of the two skeletons used and the joints that belong to each group. Our pipeline for each dog contains a separate neural network, H-GPLVM and shape model, such that no data from that particular dog is seen by its corresponding models prior to testing.

Table 1 contains the PA MPJPE and PA PCK 3D results for the comparison. Comparing these results with the MPJPE and PCK 3D results, for our method, the PA MPJPE decreases the error by an average 0.416 and PA PCK 3D increases by 0.233. For BADJA, the MPJPE PA decreases the error by an average 1.557 and PA PCK 3D increases by 0.523, showing the difficulty of determining the root rotation from silhouette alone, as is the case using BADJA.

4.1. Applying the Pipeline to Real Kinect Footage

Running the network on real-world data involves the additional step of generating a mask of the dog from the input image. We generate the mask from the RGB image for two reasons: (1) RGB segmentation networks pre-trained to detect animals are readily available, (2) the RGB image has

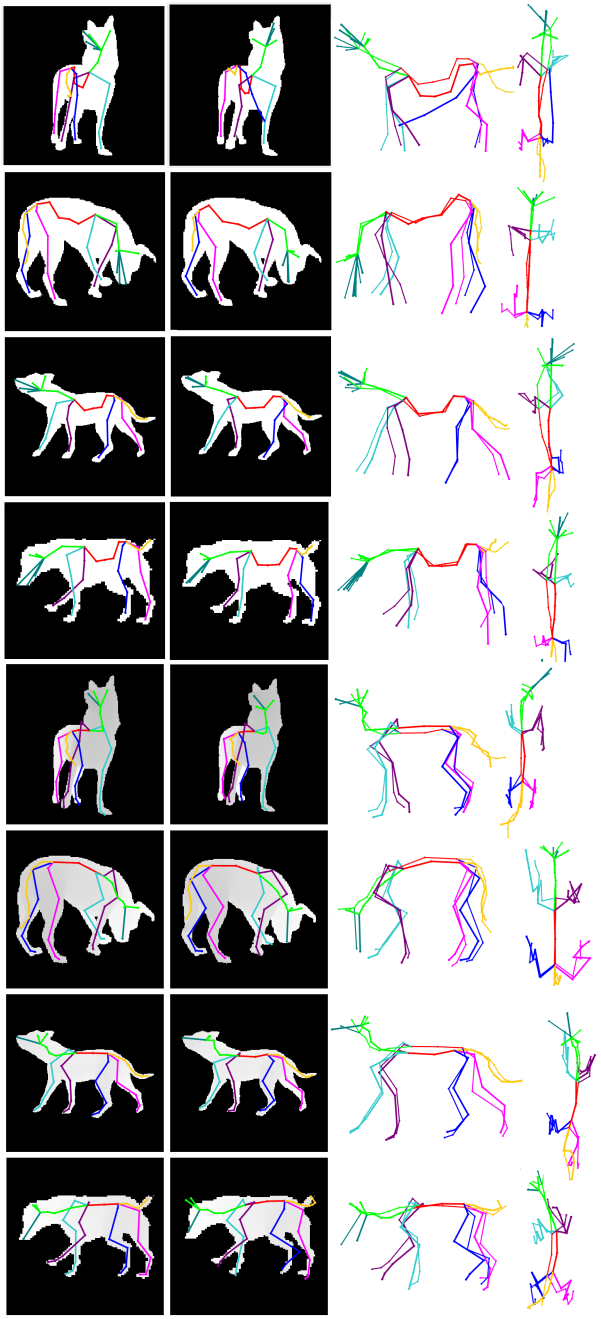

Figure 7: An example of results from BADJA [3] (rows 1-4) and our results (rows 5-8). Column 1 is the ground truth skeleton. Column 2 is the projection of 3D results. Column 3 is a side view of the 3D result as calculated in the PA MJPJE error (where the ground truth shown in a thinner line) and column 4 is a top-down view.

a higher resolution than the depth image and contains less noise, particularly when separating the dogs' feet from the ground plane. As such, the mask is generated from the RBG image before being transformed using a homography matrix into depth-image coordinates. A combination of two pretrained networks are used to generate the mask: Mask R-CNN [13] and Deeplab [8]. More details are included in the supplementary material. We display 3D results in Table 2, for cases where the neutral shape of the dog is unknown and known. Examples of skeletons are shown in Figure 8.

Dog	Method	Metric	All	Head	Body	Tail
Dog1	Ours	MPJPE	**0.471**	**0.382**	**0.527**	**0.385**
		PCK	**0.936**	**0.984**	**0.915**	**0.955**
	BADJA[3]	MPJPE	0.976	0.993	1.002	0.879
		PCK	0.665	0.607	0.685	0.661
Dog2	Ours	MPJPE	**0.402**	**0.303**	**0.410**	**0.473**
		PCK	**1.000**	**1.000**	**1.000**	**0.998**
	BADJA[3]	MPJPE	0.491	0.392	0.524	0.486
		PCK	0.956	**1.000**	**1.000**	0.928
Dog3	Ours	MPJPE	**0.392**	**0.439**	**0.390**	**0.353**
		PCK	**0.985**	**0.945**	**0.994**	0.999
	BADJA[3]	MPJPE	0.610	0.843	0.617	0.356
		PCK	0.866	0.707	0.874	**1.000**
Dog4	Ours	MPJPE	**0.417**	**0.395**	**0.421**	**0.428**
		PCK	**0.981**	**0.953**	**0.985**	**0.996**
	BADJA[3]	MPJPE	0.730	0.678	0.760	0.687
		PCK	0.787	0.861	0.754	0.817
Dog5	Ours	MPJPE	**0.746**	**0.542**	**0.748**	0.944
		PCK	**0.790**	**0.925**	**0.787**	0.664
	BADJA[3]	MPJPE	0.997	0.763	1.107	**0.885**
		PCK	0.692	0.794	0.658	**0.694**

Table 1: 3D error results as calculated using PA MPJPE and PA PCK 3D, comparing our pipeline and that used in BADJA [3] on each of the 5 dogs. Errors are reported relating to the full body or focussed body parts in Figure 6.

Dog	Method	Metric	All	Head	Body	Tail
Dog6	CNN	MPJPE	0.866	0.491	0.776	1.523
		PCK	0.745	0.956	0.780	0.425
	H-GPLVM	MPJPE	0.667	0.466	0.627	0.993
		PCK	0.873	0.969	0.938	0.575
	H-GPLVM (known shape)	MPJPE	**0.384**	**0.433**	**0.437**	**0.169**
		PCK	**0.967**	**0.975**	**0.954**	**1.000**
Dog7	CNN	MPJPE	0.563	**0.364**	0.507	0.939
		PCK	0.907	**0.993**	0.943	0.707
	H-GPLVM	MPJPE	**0.557**	0.494	**0.471**	**0.888**
		PCK	**0.922**	0.947	**0.982**	**0.711**

Table 2: 3D Error results of PA MPJPE and PA PCK 3D when using real Kinect images, where each skeleton is scaled such that the head has length of two units. We show the errors for the network prediction (CNN) and the final pipeline result (H-GPLVM). For Dog6, we also show the error where the shape of the dog mesh and skeleton is known.

4.2. Shape Estimation of Unknown Dogs

If the skeleton and neutral mesh for the current dog is unknown beforehand – as is the case in all our results apart from the 'known shape' result in Table 2 – a shape model is

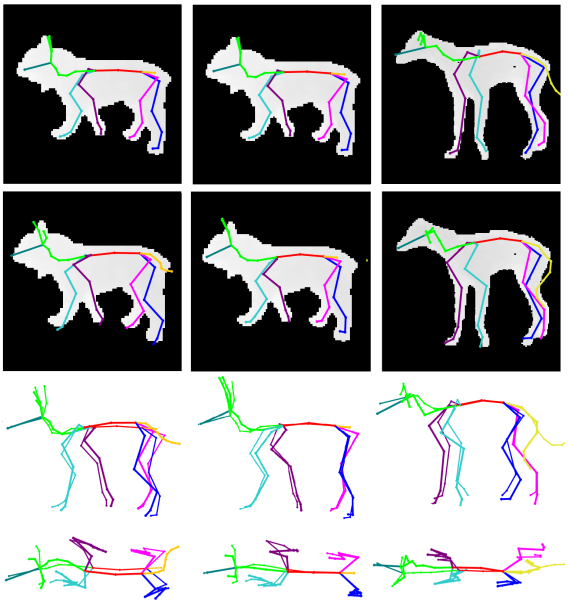

Figure 8: Example of results on real Kinect images. From the top: ground truth, projection of final 3D result, comparing the 3D result with the thinner ground truth result after calculating PA MPJPE. Left: Dog6, unknown shape. Centre: Dog6, known shape. Right: Dog7, unknown shape.

used to predict this information. The model is built from 18 dogs: five dogs are used to train the CNN and were created by an artist, an additional six dogs were also created by the artist, three dogs are scans of detailed toy animals, and four are purchased photogrammetry scans. All dogs are given a common pose and mesh with a common topology. The PCA model is built from the meshes, bone lengths and the joint rotations required to pose the dog from the common pose into its neutral standing position. The first four principal components of the model are used to find the dog with bone proportions that best match the recorded dog. This produces the estimated neutral mesh and skeleton of the dog.

4.3. Extending to Other Quadruped Species

We tested our network on additional 3D models of other species provided by Bronstein et al. ([5], [6]). Images of the models are rendered as described in Section 3.2. The training data for the network consists of the same five motions for the five training dogs. As no ground truth skeleton information is provided for the 3D models, we evaluate the performance based on visual inspection. The example results provided in the first three columns of Figure 9 show that the network performs well when the pose of a given animal is similar to that seen in the training set, even if the subject is not a dog. However, when the pose of the animal is very different from the range of poses in the training set, prediction degrades, as seen in the last three columns of Figure 9. This provides motivation for further work.

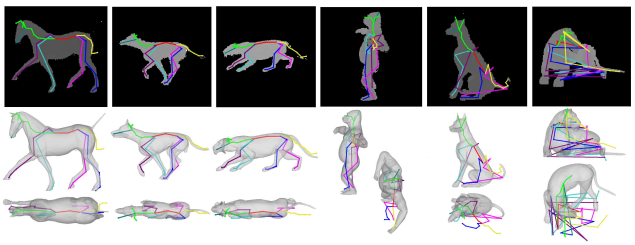

Figure 9: The network result when given images of a subset of 3D models provided by Bronstein et al. ([5], [6]), rendered as in Sec 3.2. Although the network is trained with only dog images, the first three columns show the network can generate a good pose for images where the animal is similar to that in the training set. The last three columns show where the network failed to predict a plausible pose.

5. Conclusion and Future Work

We have presented a system which can predict 3D shape and pose of a dog from depth images. We also present to the community a data set of dog motion from multiple modalities - motion capture, RGBD and RGB cameras – of varying shapes and breeds. Our prediction network was trained using synthetically generated depth images leveraging this data and is demonstrated to work well for 3D skeletal pose prediction given real Kinect input. We evaluated our results against 3D ground truth joint positions demonstrating the effectiveness of our approach. Figure 9 shows the potential in extending the pipeline to other species of animals. We expect that a more pose-diverse training set would produce results more accurate than the failure cases in Figure 9. Apart from the option to estimate bone length over multiple frames, our pipeline does not include temporal constraints, which would lead to more accurate and smoother predict sequences of motion. At present, mask generation requires an additional pre-processing step and is based on the RGB channel of the Kinect. Instead, the pose-prediction network could perform a step where the dog is extracted from the depth image itself. This may produce more robust masks, as extraction of the dog would no longer rely on texture information. As General Adversarial Networks (GANs) are now considered to produce state-of-the-art results, we intend to update our network to directly regress joint rotations and combine this with a GAN to constrain the pose prediction.

Acknowledgement. This work was supported by the Centre for the Analysis of Motion, Entertainment Research and Applications (EP/M023281/1), the EPSRC Centre for Doctoral Training in Digital Entertainment (EP/L016540/1) and the Settlement Research Fund (1.190058.01) of the Ulsan National Institute of Science & Technology.

References

[1] Carnegie mellon university motion capture database. http://mocap.cs.cmu.edu. Accessed: 2019-08-05. 1

[2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 2

[3] 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. 1, 2, 4, 5, 6, 7

[4] Joshua Blake, Christian Kerl, Florian Echtler, and Lingzhu Xiang. libfreenect2: open-source library for Kinect v2 depth camera, release 0.1.1, Jan. 2016. 4

[5] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. Efficient computation of isometry-invariant distances between surfaces. *SIAM Journal on Scientific Computing*, 28(5):1812–1836, 2006. 8

[6] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. Calculus of nonrigid surfaces for geometry and texture manipulation. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):902–913, 2007. 8

[7] Jinkun Cao, Hongyang Tang, Hao-Shu Fang, Xiaoyong Shen, Cewu Lu, and Yu-Wing Tai. Cross-domain adaptation for animal pose estimation, 2019. 1, 2

[8] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 2, 7

[9] Wenzheng Chen, Huan Wang, Yangyan Li, Hao Su, Zhenhua Wang, Changhe Tu, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. Synthesizing training images for boosting human 3d pose estimation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 479–488. IEEE, 2016. 2

[10] A. Farricelli. Do dogs have a collarbone? https://dogdiscoveries.com/do-dogs-have-a-collarbone/, 2019. Accessed: 2019-08-07. 4

[11] A. Gardiner and M. Raynor. *Dog Anatomy Workbook*. Trafalgar Square, 2014. 4

[12] Jacob M Graving, Daniel Chae, Hemal Naik, Liang Li, Benjamin Koger, Blair R Costelloe, and Iain D Couzin. Fast and robust animal pose estimation. *bioRxiv*, page 620245, 2019. 2

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 7

[14] Jingwei Huang and David Altamar. Pose estimation on depth images with convolutional neural network. 3

[15] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013. 1

[16] Sam Johnson and Mark Everingham. 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. 1

[17] Christoph Lassner, Gerard Pons-Moll, and Peter V Gehler. A generative model of people in clothing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 853–862, 2017. 2

[18] Neil D Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In *Advances in neural information processing systems*, pages 329–336, 2004. 2, 5

[19] Neil D Lawrence and Andrew J Moore. Hierarchical gaussian process latent variable models. In *Proceedings of the 24th international conference on Machine learning*, pages 481–488. ACM, 2007. 4, 5

[20] Wenbin Li, Sajad Saeedi, John McCormac, Ronald Clark, Dimos Tzoumanikas, Qing Ye, Yuzhong Huang, Rui Tang, and Stefan Leutenegger. Interiornet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. In *British Machine Vision Conference (BMVC)*, 2018. 2, 4

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2, 3

[22] Alexander Mathis, Pranav Mamidanna, Kevin M. Cury, Taiga Abe, Venkatesh N. Murthy, Mackenzie W. Mathis, and Matthias Bethge. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*, 2018. 2

[23] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *2017 International Conference on 3D Vision (3DV)*, pages 506–516. IEEE, 2017. 6

[24] Franziska Mueller, Dushyant Mehta, Oleksandr Sotnychenko, Srinath Sridhar, Dan Casas, and Christian Theobalt. Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1284–1293, 2017. 3

[25] Ashwin Nanjappa, Li Cheng, Wei Gao, Chi Xu, Adam Claridge-Chang, and Zoe Bichler. Mouse pose estimation from depth images. *arXiv preprint arXiv:1511.07611*, 2015. 2

[26] Tanmay Nath, Alexander Mathis, An Chi Chen, Amir Patel, Matthias Bethge, and Mackenzie W Mathis. Using deeplabcut for 3d markerless pose estimation across species and behaviors. *Nature protocols*, 2019. 2

[27] 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. 3, 5

[28] Talmo D Pereira, Diego E Aldarondo, Lindsay Willmore, Mikhail Kislin, Samuel S-H Wang, Mala Murthy, and Joshua W Shaevitz. Fast animal pose estimation using deep neural networks. *Nature methods*, 16(1):117, 2019. 2

[29] Grégory Rogez and Cordelia Schmid. Mocap-guided data augmentation for 3d pose estimation in the wild. In *Advances in neural information processing systems*, pages 3108–3116, 2016. 2

[30] Alla Safonova, Jessica K Hodgins, and Nancy S Pollard. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. In *ACM Transactions on Graphics (ToG)*, volume 23, pages 514–521. ACM, 2004. 5

[31] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, et al. Accurate, robust, and flexible real-time hand tracking. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3633–3642. ACM, 2015. 2

[32] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013. 2

[33] Leonid Sigal, Alexandru O Balan, and Michael J Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International journal of computer vision*, 87(1-2):4, 2010. 1

[34] Jonathan Taylor, Jamie Shotton, Toby Sharp, and Andrew Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 103–110. IEEE, 2012. 2

[35] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 109–117, 2017. 2

[36] W. Yang. A pytorch toolkit for 2d human pose estimation. https://github.com/bearpaw/pytorch-pose, 2019. Accessed: 2019-08-07. 5

[37] 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 the wild", 2019. 1, 2

[38] Silvia Zuffi, Angjoo Kanazawa, and Michael J Black. Lions and tigers and bears: Capturing non-rigid, 3d, articulated shape from images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3955–3963, 2018. 1, 2

[39] 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. 2