Sim2real Cattle Pose Prediction in 3D pointclouds

Mohammad Okour, Raphael Falque, Teresa Vidal-Calleja and Alen Alempijevic
Robotics Institute, University of Technology Sydney, NSW, Australia
{mohammad.okour}@student.uts.edu.au,
{raphael.guenot-falque, teresa.vidallcalleja, alen.alempijevic}@uts.edu.au

Abstract

Cattle's body shape and joint articulation carry significant information about their well-being. Building a large dataset of any animals' 3D scans is a challenging task. However, such a dataset is required for training deep learning algorithms for 3D body pose estimation. In this work, we investigate how such a dataset can be constructed for cattle from a single 3D model animated by a digital artist. Further, we reduce the *sim2real* gap between the virtual dataset and real scans of animals by augmenting the shape of the 3D model to cover the range of possible body shapes. The generated dataset is tested on semantic keypoints detection with an encoder-decoder architecture.

1 Introduction

Robotics and automation are being adopted in livestock agriculture production systems, assisting in labor-intensive tasks. Perception systems are critical to collect robust and precise information about the farm and animals on it [Duong *et al.*, 2020]. The animal body structure has an impact on behavior, well-being, and fertility [Saad *et al.*, 2021]. Changes in the locomotion of cattle are often a potential indicator of health issues [B. Sadiq *et al.*, 2017]. Specifically, in cattle, farmers often monitor their herd for structural soundness [Saad *et al.*, 2021], body condition score [Bell *et al.*, 2018], and lameness [Russello *et al.*, 2021] that can all affect the profitability of a herd. Crucial in their assessment is identifying the pose of joints and limb actuation while cattle are in motion, which is referred to as body pose estimation.

Human pose detection and tracking frameworks have already received significant attention, stemming from their wide-ranging use for human-computer interaction or activity recognition [Wang *et al.*, 2021]. Animal pose frameworks are emerging; the difficulty of data acquisition is a byproduct of unwilling cooperation and implicit difficulty coordinating animals in the process. Available animal data sets, such as [Yu *et al.*, 2021], [Jianguo *et al.*, 2019], are 2D images or sequences with no ground truth for joint position containing significant human-annotation [Joska *et al.*, 2021].

Attempts to estimate animal pose have relied on domain adaptation, transferring knowledge from human datasets [Cao *et al.*, 2019a]. Using synthetic training data for 3D pose estimation of animals [Fangbemi *et al.*, 2020] leverages human pose estimation, creating RGB images that can be fed into networks. The focus is on generating realistic views by minimizing the difference of distributions between the synthetic RGB training data and real data of animals from the wild. The approaches generally consider a single animal in the wild, not part of a herd (group). Apart from pose estimation, the quality of the joints and general shape of animals are not exploited for animal assessment. Further, the simulations used do not include variability within the animal species [Fangbemi *et al.*, 2020] and therefore do not fully account for *sim2real* challenges of shape variability under deformation while animals are in motion [Höfer *et al.*, 2021].

Our previous work directly exploits cattle shapes captured in high fidelity for beef cattle assessment [McPhee *et al.*, 2017]. Such data requires close proximity of depth sensors to cattle isolated from the group, thus obtaining unobstructed complete views of the body. Requirements of reliable depth and adequate coverage of animals' bodies, together with the constraint of the limited camera field-of-view, require a multi-depth camera system for acquisition. Generating a realistic RGB view from such a system, which is non-trivial, could be orthogonal to the use of this data for 3D pose estimation. Data from unstructured point clouds may present a challenge for some of the deep learning models. Pointnet++ [Qi *et al.*, 2017] solves this problem by dealing with raw point cloud data while respecting the permutation invariance of individual points in the input.

This work deals with the lack of realistic synthetic an-

imal data and proposes a methodology to directly learn joint coordinates from 3D pointcloud data. We present an approach for scaling a single model animation into a dataset with a large variety of body shapes and postures. The *sim2real* gap is investigated by comparing the training of a model for keypoint detection [Falque *et al.*, 2022] on our generated dataset and on a real dataset collected by scanning more than 200 animals. We further propose reformulating the encoder-decoder proposed in [Falque *et al.*, 2022] to solve the problem of keypoints located outside of the pointcloud collected from the animals (e.g., skeletal joints prediction).

2 Related Work

Human pose detection and tracking frameworks have received significant attention, stemming from their wide-ranging use for human-computer interaction in virtual reality and gaming consoles to human activity recognition for surveillance applications. Seminal work on utilizing depth data for this task by [Shotton *et al.*, 2011] leveraged a significant amount of data from controlled environments (motion capture room) where actors perform a variety of actions. This work was followed by a raft of RGB-based approaches that leverage deep learning frameworks such as [Mehta *et al.*, 2017; Mathis *et al.*, 2018] to learn body joints pose. OpenPose [Cao *et al.*, 2019b], a benchmark for human pose estimation, represented the first real-time multi-person system for body keypoints.

Animal pose estimation models are mainly inspired by human models and have progressively evolved through the last decade. Available animal data sets, such as [Yu *et al.*, 2021; Jianguo *et al.*, 2019; Russello *et al.*, 2022], are 2D images or sequences with no ground truth for joint positions. Custom animal datasets, such as AcinoSet for cheetahs [Joska *et al.*, 2021], contain significant human annotation.

As a result of a shortage of data sets in the animal field, data input can be expanded by either transforming learning models from another domain (another animal) or humans and fine-tuning [Sanakoyeu *et al.*, 2020] [Mathis *et al.*, 2021; Pereira *et al.*, 2020; Cao *et al.*, 2019a], or by using synthetic data augmentation process [Mu *et al.*, 2020; Del Pero *et al.*, 2017; Del Pero *et al.*, 2015; Li and Lee, 2021; Zuffi *et al.*, 2019]. While some of these approaches extract 3D keypoints, they may not be on actual joints [Badger *et al.*, 2020], or the fitted 3D models do not have an evaluation of mesh quality [Yao *et al.*, 2019; Li and Lee, 2021]. Difficulties also arise in assigning keypoint locations, needing imputable relationships between the model and the actual animal shape [Zuffi *et al.*, 2019]. Further, the result might be a visually appealing 3D shape, though it is not geometrically consistent with the animal structure [Yao *et al.*, 2022].

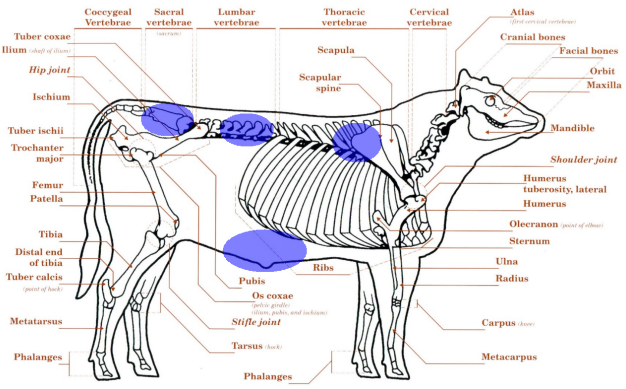

Figure 1: Skeletal view of a cattle. The generation of the simulated dataset leverage the scaling of anatomically correct skeletal components to create a variety of body shapes. Illustration amended from [Black *et al.*, 2001]. Areas in blue represent four of the local areas that are scaled for augmentation.

Existing work on synthetic animated training data for 3D pose estimation of animals [Fangbemi *et al.*, 2020] leverages human pose estimation, retraining existing networks such as OpenPose and Pose3D with RGB and joint pose data from a simulator. In doing so, the main challenge to overcome is creating RGB images where the difference in distribution between the synthetic training data and the real data from the wild is minimized. While overcoming significant difficulties in getting anatomically related keypoints (on animal joints), this approach requires transforming RGB images into realistic scenes. There is no attempt to validate the *sim2real* gap of keypoint locations on real data, or use the 3D data for the purpose of animal assessment.

3 Methodology

3.1 Dataset generation from a 3D model animation

A large dataset is generated by augmenting an artist-created animated 3D model of a bovine (steer) into a set of point clouds that represent scans with different body shapes. The 3D model animation is made of a triangle mesh defined as a set of vertices and faces $\{\boldsymbol{V}, \boldsymbol{F}\}$ and an animated skeleton defined as a graph which defined a set of nodes and edges $\{\boldsymbol{G}, \boldsymbol{E}\}$. The mesh, from which the vertices are defined as $\boldsymbol{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ such that $\boldsymbol{v}_i \in \mathbb{R}^3$, is *sculpted* digitally to provide a realistic 3D shape of cattle. The skeleton joints, $\boldsymbol{G} = \{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_g\}$ such that $\boldsymbol{g}_j \in \mathbb{R}^3$, are anatomically correct and located to provide realistic degrees of freedom when compared to a skeleton such as the one shown in Figure 1. The an-

Table 1: Parameters Scaling Factors Limits. Parameters unmentioned in the truncated column are set to 1.

Parameter	μ	σ	truncated
Sacral Vertebrae	1.2	0.2	$1 < s_x < 1.4$
Lumbar Vertebrae	1.1	0.2	$1 < s_x < 1.2$
Thoracic Vertebrae	1.15	0.2	$1 < s_x < 1.3$
abdomen	1.2	0.2	$1 < s_x < 1.4$
full scale (length)	0.965	0.2	$0.93 < s_y < 1$
full scale (height)	0.9	0.2	$0.85 < s_z < 0.95$

imation is then constructed by morphing the mesh with linear blend skinning using the guidance of the animated skeleton.

Mesh deformation

Two augmentation methods are used for generating the different body shapes. First, an augmentation of the animal's joint positions is performed by selecting a different pose from the animation and non-uniformly scaling the skeleton structure. This provides us with a variety in terms of under-laying shapes of animals (i.e., variation in the joints' position). Secondly, the shape of the 3D model is then augmented to generate animals that have different body shapes (e.g., animals with more/less fat and muscle).

More specifically, the joints' position augmentation is achieved by selecting one of the poses from the first forty frames of the model animation. For each frame, eight parameters were assigned to the scaling of anatomical parts as in (1), from which three points are assigned to the spine (Sacral Vertebrae, Lumbar Vertebrae, Thoracic Vertebrae), abdomen, and full body scale (length and height). These anatomical areas can be visualized in Figure 1.

Each anatomical area that requires scaling is included into the skeleton as an additional armature and is assigned a dedicated scaling matrix defined such that the j^{th} armature is scaled with:

$$\boldsymbol{S}_j = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \tag{1}$$

where $s_x \sim N(\mu, \sigma^2)$, $s_y \sim N(\mu, \sigma^2)$, $s_z \sim N(\mu, \sigma^2)$, the parameters applied for scaling the areas in the X, Y, and Z direction respectively. The parameters for each respective area are defined in Table 1.

The mesh is then deformed with respect to the skeleton animation using linear blend skinning (LBS). The mesh vertices' position is calculated by the weighted linear combination of transformations of armature joints as follows:

$$\boldsymbol{v}_i^* = \sum_j w_j \boldsymbol{M}_j \boldsymbol{S}_j \boldsymbol{v}_i \tag{2}$$

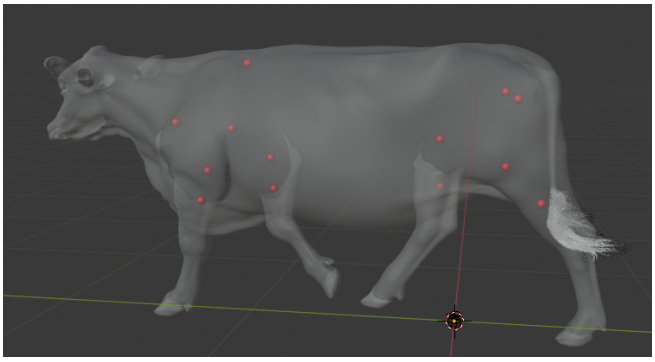

Figure 2: Each model is annotated by thirteen points, including three points at each of the front legs, two at the back legs, two at the hip bones, and one on the front side of the spine.

where $\boldsymbol{v}_i^*$ is the transformed vertex position of i^{th} vertex, w_j is the weight, $\boldsymbol{M}_j$ is the transformation matrix (rotation and translation) of joint j, and $\boldsymbol{S}_j$ the scaling matrix defined in (1).

Additionally, the shape of the 3D model is augmented by using the smooth Laplacian modifier. The Laplacian operator offers the advantage of smoothing or sharpening the shape of the 3D model [Sorkine *et al.*, 2004]. This allows for generating shapes which look more/less fat depending on the sign of the weight applied to the Laplacian operator. The Laplacian smoothing is defined using the standard cotangent Laplacian operator defined as:

$$\Delta \boldsymbol{v}_i = \frac{1}{2A_i} \sum_{\boldsymbol{v}_j \in \mathcal{N}_1(\boldsymbol{v}_i)} (\cot\alpha_{i,j} + \cot\beta_{i,j})(\boldsymbol{v}_j - \boldsymbol{v}_i) \tag{3}$$

where $\Delta \boldsymbol{v}_i$ is the Laplacian of vertex i on a mesh, A_i the vertex area, $\mathcal{N}_1$ contains the one-ring neighborhood of the i^{th} vertex, $\alpha_{i,j}$ and $\beta_{i,j}$ are the opposite angle to the edge $\{\boldsymbol{v}_i, \boldsymbol{v}_j\}$ with respect to the mesh faces. The smoothing of the shape is applied locally by assigning weights to the vertices in the hip area.

From the skeleton joints $\boldsymbol{G}$, a set of these joints are selected as annotations and stored in $\boldsymbol{J} = \{\boldsymbol{J}_1, \ldots, \boldsymbol{J}m\}$, such that $\boldsymbol{J}_j \in \boldsymbol{G}$. More specifically, the semantic meaning of these joints is defined as follows: knee joints, hock joints, shoulder joints, and hip joints. The location of these joints is displayed in Figure 2.

Raycasting and point cloud generation

To generate the required point cloud, seventeen realsense D432 depth-cameras are used, eight on each side, and one at the top, to replicate the real setup, as shown in Figure 4. The cameras are hardware synchronized as per [Sterzentsenko *et al.*, 2018].

After defining the intrinsic cameras' parameters, we found the center of each pixel (u,v) while taking the top

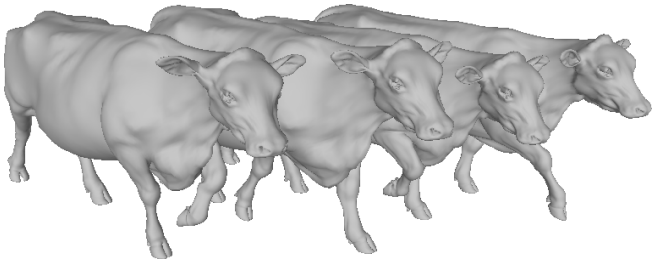

Figure 3: Sample of mesh generated with different body positions.

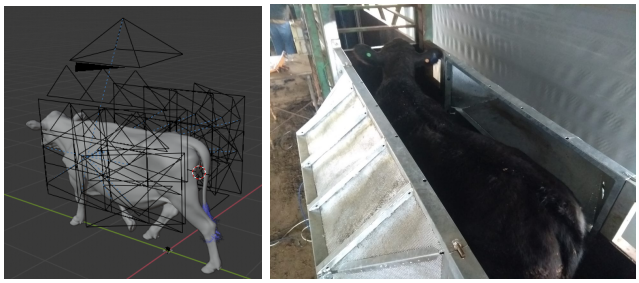

Figure 4: Seventeen depth-cameras are used in the simulation (left) to replicate the physical setup used for data collection of cattle (right).

right side of the image plane as a reference point as

$$\boldsymbol{p}_{u,v} = \begin{bmatrix} \frac{w}{2} \\ \frac{h}{2} \end{bmatrix} - \begin{bmatrix} \frac{w}{n}(v-0.5) \\ \frac{h}{m}(u-0.5) \end{bmatrix} \tag{4}$$

where $\boldsymbol{p}_{u,v}$ is the center of pixel (u, v), (w, h) image width and height respectively, (n, m) image resolution (number of pixels).

Afterward, an iteration over each pixel is performed to project a ray and determine if it intersects the 3D model,

$$\boldsymbol{u}_{r_{u,v}} = |\boldsymbol{p}^*_{u,v} - \boldsymbol{o}^*| \tag{5}$$

where $\boldsymbol{o}^*$: the camera focal point location, $\boldsymbol{p}^*_{u,v}$: $\boldsymbol{p}_{u,v}$ translated to the image plane, $\boldsymbol{u}_{r_{u,v}}$: directional vector of each raycast pixel from the focal point towards each pixel,

$$\boldsymbol{X}(t) = t\boldsymbol{u}_{r_{u,v}} \tag{6}$$

where t: distance along the ray, $\boldsymbol{X}$: points on $\boldsymbol{u}_{r_{u,v}}$. The raycasting can then be performed with an AABB tree [Bergen, 1997], which returns the point $\boldsymbol{i}$ where the ray hits the surface and returns nothing if the ray does not hit anything.

The pointcloud viewed by one depth-camera is then the concatenation of all points hit by the rays. More formally, the pointcloud of the k^{th} camera is obtained as $\boldsymbol{I}_{i,k} = \bigcup_{\forall u, \forall v} \boldsymbol{i}_{u,v}$. The point clouds from all depth-cameras are then concatenated into a single point cloud such as:

$$\boldsymbol{I}_i = \bigcup_{k=1}^{17} \boldsymbol{I}_{i,k} \tag{7}$$

The i^{th} instance of our dataset now had the point cloud $\boldsymbol{I}_i$ and its associated annotations $\boldsymbol{J}_i$.

3.2 Network Architecture

Given a set of point clouds $\{\boldsymbol{I}_1, \ldots, \boldsymbol{I}_n\}$ and corresponding anatomically correct joints position $\{\boldsymbol{J}_1, \ldots, \boldsymbol{J}_n\}$ generated following Section 3.1, we train an encoder-decoder that takes as an input a point cloud $\boldsymbol{I}_i$ and predict the position of the joints $\boldsymbol{J}_i$. The architecture builds upon [Falque *et al.*, 2022] where the distance on the manifold to the joints' keypoints is estimated through an encoder-decoder. In contrast with [Falque *et al.*, 2022], where particular care is given to the data augmentation of the limited dataset, here we solve the data augmentation through the generation of a larger dataset. The encoder-decoder used for the experiments is Pointnet++ [Qi *et al.*, 2017]. The generation of the simulated dataset and its integration with the network is represented in the diagram shown in Figure 5.

In brief, [Falque *et al.*, 2022] precomputes the distance on the manifold $\boldsymbol{D}$ using the heat kernel method [Crane *et al.*, 2017] by leveraging the *tufted Laplacian* [Sharp and Crane, 2020] which can be computed directly on a point cloud structure. The distance on the manifold is then estimated through learning as a feature using an encoder-decoder network. The inputs of the encoder-decoder are defined by the size of the pointcloud (i.e., a matrix $\boldsymbol{I}$ of size $n \times 3$) and the outputs by the number of joints that should be predicted (i.e., a matrix $\hat{\boldsymbol{D}}$ of size $n \times m$).

In contrast to the method proposed in [Falque *et al.*, 2022], this paper investigates the prediction of joints' position. Inherently, the joints are not located on the surface, and their position has to be estimated from the distances on the manifold. We propose to define the joint position as the weighted sum of the point cloud position with respect to the encoder-decoder prediction of the distance on the manifold $\hat{\boldsymbol{D}}$ such that:

$$\boldsymbol{J}_j = \frac{1}{\sum_i \hat{\boldsymbol{D}}_{i,j}} \sum_{i=0}^{|\boldsymbol{I}|} \boldsymbol{I}_{i,j} \hat{\boldsymbol{D}}_{i,j} \tag{8}$$

This formulation allows predicting joints' position outside of the point cloud in the case where the underlying shape is convex. This is particularly relevant for joints located in the legs. A sample of the joint detection using *argmax* and the weighted sum is displayed in Figure 6.

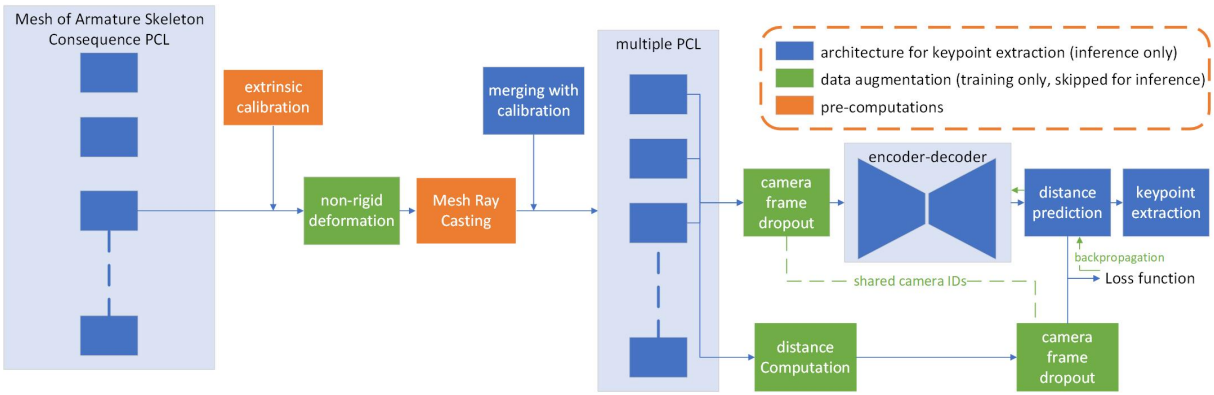

Figure 5: Method overview: from the simulated model, the armature undergoes rigid scaling and meshes a non-rigid deformation. Through raycasting over a number of cameras, individual pointclouds are generated and merged. At inference time, the point cloud is passed into an encoder-decoder architecture (Pointnet++ [Qi *et al.*, 2017]) to extract the keypoints. During training, the dataset uses keypoints from the armature and the distances on the manifold are pre-computed. The encoder-decoder inputs are $n \times 3$ points and the outputs are the $n \times 13$ distances to the 13 joints keypoints.

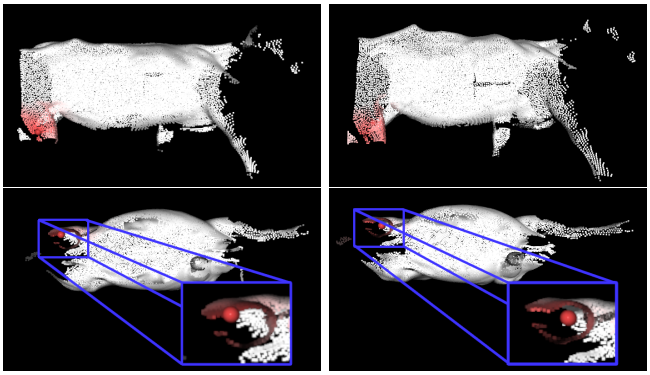

Figure 6: Given the predicted distance on the manifold shown in red for the rear leg: joint prediction using the *argmax* of the network prediction (on the left) versus the weighted sum of the point cloud described in (8) (on the right). The side view (on top) and the bottom view (below) are displayed.

In Figure 5, a diagram of the proposed methodology is provided, showing how a random sample can be generated and passed through the encoder-decoder.

4 Experiments

Within the simulated dataset, a set of 1200 samples has been generated by producing 30 different shapes for each of the 40 frames.

First, a quantitative evaluation of the *sim2real* gap generated by training the network on a simulated dataset versus a real dataset annotated manually. This is achieved by comparing the performance of the proposed method to the unmodified network proposed in [Falque *et al.*, 2022] (only the training data is changed). To perform a fair comparison, the canonical mesh of the 3D model is annotated manually, and the vertices (with the same indexes) of the deformed mesh, generated by following the method described in Section 3.1, are used as keypoints. In contrast with [Falque *et al.*, 2022], this provides the advantage of annotating the 3D model a single time instead of manually annotating all the instances from the dataset. The RMSE for both the keypoints prediction and the estimated distance on the manifold is reported in Table 2. In the first column, the training set is the simulated dataset, and in the second column the training set is real data with 200 samples annotated manually. These dataset are used to train Pointnet++ [Qi *et al.*, 2017] to predict the distance on the manifold to annotated keypoints [Falque *et al.*, 2022]. The evaluation is performed on real data with 100 samples. As expected, there is a *sim2real* gap which creates a drop in performance when training the encoder-decoder on simulated data. Further qualitative results are available in Figure 7. As shown in the figure, the learned distances on the manifold are similar.

Table 2: Study of the *sim2real* gap. The RMSE of the keypoint annotation is reported in centimetres. For reference, the bounding box around the animal would have a length of approximately two meters.

trained with	simulated	real data
RMSE (keypoints)	11.44	5.57
RMSE ($\hat{\boldsymbol{D}} - \boldsymbol{D}$)	$4.94.10^{-4}$	$2.1.10^{-4}$

A further qualitative evaluation is performed by di-

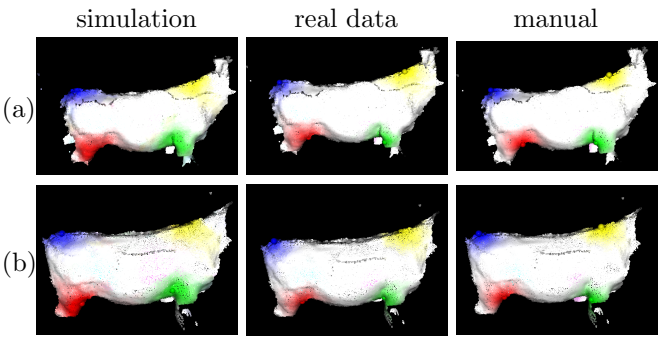

Figure 7: Comparison between the encoder-decoder described in [Falque *et al.*, 2022] using the generated dataset and real data. The first column shows the prediction of the joints' distance while training with the simulated dataset, the second column shows the prediction using the real dataset with data augmentation, and the last column shows the distance with respect to the manual annotation.

rectly regressing the distance to the 13 joints position $\boldsymbol{J}$ defined in Section 3.1. Samples of the joints prediction are shown in Figure 8. For this experiment, it is impossible to compare quantitatively with [Falque *et al.*, 2022] as the joints' position can not be annotated manually.

5 Discussion

The results from Table 2, show that the augmentation of the shape described in Section 3.1 can be used as a synthetic data to train a neural network for semantic keypoints annotation. While a *sim2real* gap still exists, the experiment shows promising generalization on real data. In the case where a limited real dataset is available, the synthetic data provides valuable pre-training of the network and would bootstrap the fine-tuning of the model on more realistic data.

Additionally, the main advantages of the proposed method are the automation of the annotation (removal of the erroneous and cumbersome human annotation process) and the possibility of annotating points external to the point cloud (e.g., joints).

6 Conclusion

In this paper, we propose a method for generating a large dataset from a single 3D model animation. We propose to augment this model through variations of poses, skeletal scaling, and local smoothing of the shape. The application of such dataset is studied by investigating how the dataset can be used for estimating joints positions. The *sim2real* gap between training a deep model with a simulated dataset versus real data is investigated for this application.

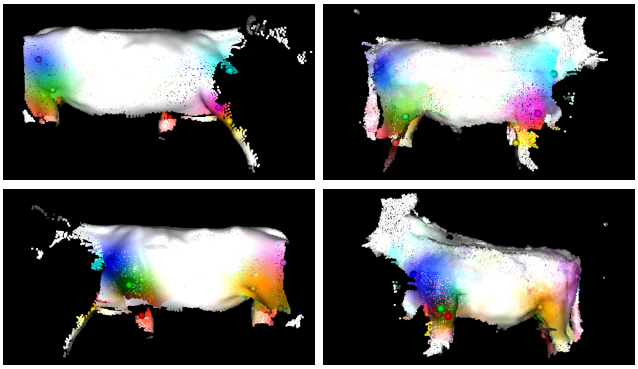

Figure 8: Qualitative evaluation of the learning of the joints prediction. Each colour represents the distance to the joint learned by the network (i.e., the stronger the colour is, the closer to the joint). In column (a), the prediction of the network on the simulated dataset. In column (b), the prediction of the network for the real dataset.

While the results for joint prediction look promising, further work is needed to see if such an approach would translate well to other applications (e.g., body condition score). In terms of additional future work, we are planning to investigate how the Euclidean distance can be used as an alternative to the distance on the manifold. This would provide a better proxy to estimate the joints' positions compared to Equation (8), though implementation is not trivial as the Euclidean distance still needs to be clamped on the local manifold.

Furthermore, the anatomically correct modelling of body muscle and fat layers would open doors for many additional applications. Such modelling is extremely challenging and rarely performed by a few talented 3D artists. Making these models parametric to cover the different types of body conditions for large data augmentation would add an extra layer of complexity.

Acknowledgments

This work was partly supported by an Australian Government Research Training Program (RTP) Scholarship, Food Agility CRC top-up scholarship, the University of Technology Sydney and Meat and Livestock Australia under grant number B.GBP.0051. The use of animals and the procedures performed in this study were approved by the University of New England (UNE) Animal Ethics Committee (Approval number: ARA21-070).

References

[B. Sadiq *et al.*, 2017] Mohammed B. Sadiq, Siti Z Ramanoon, Wan Mastura Shaik Mossadeq, Rozaihan Mansor, and Sharifah Salmah Syed-Hussain. Asso-

ciation between lameness and indicators of dairy cow welfare based on locomotion scoring, body and hock condition, leg hygiene and lying behavior. *Animals*, 7(11):79, 2017.

[Badger *et al.*, 2020] Marc Badger, Yufu Wang, Adarsh Modh, Ammon Perkes, Nikos Kolotouros, Bernd G Pfrommer, Marc F Schmidt, and Kostas Daniilidis. 3d bird reconstruction: a dataset, model, and shape recovery from a single view. In *European Conference on Computer Vision*, pages 1–17. Springer, 2020.

[Bell *et al.*, 2018] Matthew Bell, Mareike Maak, Marion Sorley, and Robert Proud. Comparison of methods for monitoring the body condition of dairy cows. *Frontiers in Sustainable Food Systems*, 2, 11 2018.

[Bergen, 1997] Gino van den Bergen. Efficient collision detection of complex deformable models using aabb trees. *Journal of graphics tools*, 2(4):1–13, 1997.

[Black *et al.*, 2001] Jodi P. Black, R. Warren Flood, John Grimes, and Jeanne M. Osborne. *Beef Resource Handbook*. The Ohio State University, 2001.

[Cao *et al.*, 2019a] 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/CVF International Conference on Computer Vision*, pages 9498–9507, 2019.

[Cao *et al.*, 2019b] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[Crane *et al.*, 2017] Keenan Crane, Clarisse Weischedel, and Max Wardetzky. The heat method for distance computation. *Commun. ACM*, 60(11):90–99, October 2017.

[Del Pero *et al.*, 2015] Luca Del Pero, Susanna Ricco, Rahul Sukthankar, and Vittorio Ferrari. Articulated motion discovery using pairs of trajectories. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2151–2160, 2015.

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

[Duong *et al.*, 2020] Linh NK Duong, Mohammed Al-Fadhli, Sandeep Jagtap, Farah Bader, Wayne Martindale, Mark Swainson, and Andrea Paoli. A review of robotics and autonomous systems in the food industry: From the supply chains perspective. *Trends in Food Science & Technology*, 106:355–364, 2020.

[Falque *et al.*, 2022] Raphael Falque, Teresa Vidal-Calleja, and Alen Alempijevic. Semantic keypoint extraction for scanned animals using multi-depth-camera systems. *arXiv preprint arXiv:2211.08634*, 2022.

[Fangbemi *et al.*, 2020] Abassin Sourou Fangbemi, Yi Fei Lu, Mao Yuan Xu, Xiao Wu Luo, Alexis Rolland, and Chedy Raissi. Zoobuilder: 2d and 3d pose estimation for quadrupeds using synthetic data, 2020.

[Höfer *et al.*, 2021] Sebastian Höfer, Kostas Bekris, Ankur Handa, Juan Camilo Gamboa, Melissa Mozifian, Florian Golemo, Chris Atkeson, Dieter Fox, Ken Goldberg, John Leonard, et al. Sim2real in robotics and automation: Applications and challenges. *IEEE transactions on automation science and engineering*, 18(2):398–400, 2021.

[Jianguo *et al.*, 2019] L. Jianguo, L. Weiyao, H. Tang, M. Greg, and D. Joachim. Iccv 2019 workshop & challenge on computer vision for wildlife conservation (CVWC), 2019.

[Joska *et al.*, 2021] Daniel Joska, Liam Clark, Naoya Muramatsu, Ricardo Jericevich, Fred Nicolls, Alexander Mathis, Mackenzie W Mathis, and Amir Patel. Acinoset: a 3d pose estimation dataset and baseline models for cheetahs in the wild. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13901–13908. IEEE, 2021.

[Li and Lee, 2021] Chen Li and Gim Hee Lee. From synthetic to real: Unsupervised domain adaptation for animal pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1482–1491, 2021.

[Mathis *et al.*, 2018] Alexander Mathis, Pranav Mamidanna, Kevin M Cury, Taiga Abe, Venkatesh N Murthy, Mackenzie Weygandt Mathis, and Matthias Bethge. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nature neuroscience*, 21(9):1281–1289, 2018.

[Mathis *et al.*, 2021] Alexander Mathis, Thomas Biasi, Steffen Schneider, Mert Yuksekgonul, Byron Rogers, Matthias Bethge, and Mackenzie W Mathis. Pretraining boosts out-of-domain robustness for pose estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1859–1868, 2021.

[McPhee *et al.*, 2017] Malcolm J McPhee, Bradley J Walmsley, B Skinner, B Littler, JP Siddell, LM Cafe, JF Wilkins, VH Oddy, and A Alempijevic. Live animal assessments of rump fat and muscle score in angus cows and steers using 3-dimensional imaging. *Journal of Animal Science*, 95(4):1847–1857, 2017.

[Mehta *et al.*, 2017] 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.

[Mu *et al.*, 2020] Jiteng Mu, Weichao Qiu, Gregory D Hager, and Alan L Yuille. Learning from synthetic animals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12386–12395, 2020.

[Pereira *et al.*, 2020] Talmo D Pereira, Nathaniel Tabris, Junyu Li, Shruthi Ravindranath, Eleni S Papadoyannis, Z Yan Wang, David M Turner, Grace McKenzie-Smith, Sarah D Kocher, Annegret L Falkner, et al. Sleap: Multi-animal pose tracking. *BioRxiv*, 2020.

[Qi *et al.*, 2017] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[Russello *et al.*, 2021] Helena Russello, Rik van der Tol, and Gert Kootstra. T-LEAP: occlusion-robust pose estimation of walking cows using temporal information. *CoRR*, abs/2104.08029, 2021.

[Russello *et al.*, 2022] Helena Russello, Rik van der Tol, and Gert Kootstra. T-leap: Occlusion-robust pose estimation of walking cows using temporal information. *Computers and Electronics in Agriculture*, 192:106559, 2022.

[Saad *et al.*, 2021] Hamad M Saad, R Mark Enns, Milton G Thomas, Lee L Leachman, and Scott E Speidel. Foot scores genetic parameters estimation in beef cattle. *Translational Animal Science*, 5(Supplement_S1):S180–S184, 2021.

[Sanakoyeu *et al.*, 2020] Artsiom Sanakoyeu, Vasil Khalidov, Maureen S McCarthy, Andrea Vedaldi, and Natalia Neverova. Transferring dense pose to proximal animal classes. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 5233–5242, 2020.

[Sharp and Crane, 2020] Nicholas Sharp and Keenan Crane. A Laplacian for Nonmanifold Triangle Meshes. *Computer Graphics Forum (SGP)*, 39(5), 2020.

[Shotton *et al.*, 2011] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *CVPR 2011*, pages 1297–1304, 2011.

[Sorkine *et al.*, 2004] Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and H-P Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 175–184, 2004.

[Sterzentsenko *et al.*, 2018] Vladimiros Sterzentsenko, Antonis Karakottas, Alexandros Papachristou, Nikolaos Zioulis, Alexandros Doumanoglou, Dimitrios Zarpalas, and Petros Daras. A low-cost, flexible and portable volumetric capturing system. In *2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pages 200–207. IEEE, 2018.

[Wang *et al.*, 2021] Jinbao Wang, Shujie Tan, Xiantong Zhen, Shuo Xu, Feng Zheng, Zhenyu He, and Ling Shao. Deep 3d human pose estimation: A review. *Computer Vision and Image Understanding*, 210:103225, 2021.

[Yao *et al.*, 2019] Yuan Yao, Yasamin Jafarian, and Hyun Soo Park. Monet: Multiview semi-supervised keypoint detection via epipolar divergence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 753–762, 2019.

[Yao *et al.*, 2022] Chun-Han Yao, Wei-Chih Hung, Yuanzhen Li, Michael Rubinstein, Ming-Hsuan Yang, and Varun Jampani. Lassie: Learning articulated shapes from sparse image ensemble via 3d part discovery. *arXiv preprint arXiv:2207.03434*, 2022.

[Yu *et al.*, 2021] Hang Yu, Yufei Xu, Jing Zhang, Wei Zhao, Ziyu Guan, and Dacheng Tao. Ap-10k: A benchmark for animal pose estimation in the wild. *arXiv preprint arXiv:2108.12617*, 2021.

[Zuffi *et al.*, 2019] 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". In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5359–5368, 2019.