Animal 3D reconstruction from videos

by

Youdong Ma

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Engineering

Department of Electrical and Computer Engineering
University of Alberta

Abstract

3D reconstruction of quadruped animals is a challenging problem, where key issues lie in their large shape variety and deformation within the same animal species as well as the lack of sufficient training data. In this thesis, we present two approaches toward this task. Our first approach is a model-based method called SMALF, where a parametric 3D shape template, SMAL, is employed to recover the 3D pose and shape of an animal from its raw video. On top of SMAL, our predicted mesh is further allowed to perform per-vertex deformations: this way, our proposed model benefits from the use of both parametric and non-parametric representations. Second, we present AnimalRecon, yet another method that uses a neural implicit function to represent 3D animal shapes. Neural implicit representation methods enjoy a higher degree of flexibility and expressiveness due to the continuous nature of this shape representation. However, how to extract a space-time coherent representation for a video can be difficult to achieve. First, we implement neural blend skinning, a method to enable our implicit shape to deform. Our implicit shape model is then fitted into the given video. Qualitative and quantitative results of our approach SMALF are conducted on two BADJA and RGBD-Dog datasets, where our method is demonstrated to improve over the baselines. Our method AnimalRecon is further examined on the RGBD-Dog dataset, where higher-fidelity 3D reconstruction results are shown when comparing with the existing efforts.

Preface

This thesis is an original work by Youdong Ma. No part of this thesis has been previously published

Acknowledgements

I am grateful to my supervisors: Professor Dr. Li Cheng, Professor Dr. Xingyu Li and Professor Dr. Wang Yang for giving me lots of opportunities and valuable suggestions. It would have been impossible to finish my thesis without their professional guidance. In addition, I would like to thank all my fellow lab mates for their productive discussions and friendship. I also want to thank all of my friends who made our cold city warm.

I especially want to thank my parents and my girlfriend for their support and understanding. During the pandemic, they supported me financially or spiritually.

Table of Contents

1 Introduction **1**

1.1 Contributions 3

1.2 Thesis Outline 4

2 Literature Review **5**

2.1 Template-based Reconstruction 5

2.2 Category-specific Mesh Reconstruction 6

2.3 Template-free Reconstruction 6

3 Preliminaries **9**

3.1 Data 10

3.2 Data Pre-processing 11

3.3 Evaluation Metric 11

4 Model-based Animal Reconstruction **15**

4.1 Method 16

4.1.1 SMAL model 16

4.1.2 Video-specific Feature Embedding 17

4.1.3 Differentiable Rendering 18

4.1.4 Video Pre-processing 18

4.1.5 Optimization Routine 18

4.2 Experiments 22

4.2.1 Dataset 22

4.2.2 Implementation details 22

4.2.3 Comparison to Baselines 22

4.2.4 Ablation study 23

5 Animal Reconstruction Using Implicit Function **28**

5.1 Method 29

5.1.1 Implicit Neural Representation 29

5.1.2 Neural Blend Skinning 30
5.1.3 Model Initialization 30
5.1.4 Differentiable Neural Rendering 31
5.1.5 Optimization 32
5.2 Experiments 35
5.2.1 Dataset 35
5.2.2 Implementation Details 35
5.2.3 Baselines 35
5.2.4 Comparison to Baselines 36
5.2.5 Ablation study 36

6 Discussion and Conclusion **41**
6.1 Limitation 42
6.2 Future Work 42

Bibliography **43**

List of Tables

4.1 IoU and PCK@0.15 for the RGBD dataset. 23
4.2 IoU and PCK@0.15 for the BADJA dataset. 23
4.3 IoU and PCK@0.15 for unseen animals. 24
4.4 Ablation study for the importance of each loss component 24

5.1 Comparison to baseline methods on the RGBD-Dog dataset (IoU) . . 37
5.2 Comparison of using different numbers of frames as input on the RGBD-Dog dataset (IoU) 37

List of Figures

3.1 **RGBD-Dog** includes seven dogs, each wearing a motion capture suit to provide landmark ground truth. 12

3.2 **BADJA** includes seven videos. 2D keypoint annotations are provided approximately every 5 video frames with the exception of the dog's video which is annotated densely. 13

3.3 **Data preprocessing** Given an image, we retrieve 2D keypoints and mask using pre-trained network. Given two consecutive images, we can also get optical flow. Then We will crop the images based on the mask. Note: Optical flow is a technique used to describe motion, the image on the top-right is the result after visualization. The brighter area means there is a larger movement. 14

4.1 **Overall framework** Part(a) shows the SMAL model workflow. Part(b) shows the procedure of differentiable rendering. Part(c), given an input video, the pre-trained network will predict the silhouette, the keypoints location, and the optical flow in the image. The blue arrow shows the forward process, while the brown arrow shows the back-propagation flow. 16

4.2 **Laplacian vector** 18

4.3 **Video-specific** pixel-surface embedding 21

4.4 **2D keypoints estimation** on RGBD-dog dataset 25

4.5 **2D keypoints estimation** on BADJA dataset 26

4.6 **Qualitative results** comparison to existing methods on the RGBD-Dog and BADJA datasets. For each sample, we show: (a) the input image, (b) our result, (c) the SMALify result, (d) the WLDO result, (e) the Course-to-Fine result, (d) the BARC result and (g) an alternative view of our result 27

5.1 Implicit shape with properties 29

5.2 **2D cycle re-projection loss** ensures the pixel-to-surface matches can be mapped back to their pixel locations. 34
5.3 From a set of posed meshes(Left), we learn an implicit neural 3D shape and a skinning field(Right) in the canonical pose. 38
5.4 **Qualitative results** comparison to existing methods on the RGBD-Dog dataset. For each sample, we show: (a) the input image, (b) our result, (c) the SMALify result, and (d) the Barc result 39
5.5 **Video result** on the RGBD-Dog dataset. Each second raw shows another view of the 3D shape. 40

Abbreviations

CNN Convolutional Neural Network.

CSE Continuous Surface Embeddings.

DNN Deep Neural Network.

IoU Intersection-over-union.

LBS Linear Blend Skinning.

MLP Multilayer Perceptron.

PCA Principal Component Analysis.

PCK Percentage of Correct Keypoint.

SMAL Skinned Multi-Animal Linear Model.

SMPL Skinned Multi-Person Linear Model.

SOTA State Of The Art.

Chapter 1

Introduction

A large variety of fields including zoology, farming, entertainment, biology, video gaming, and neuroscience, can all benefit from the study of animals. Future generations may rely solely on virtual reality to understand nature. Computer vision will play a considerable role here by providing tools for animal tracking, 3D capture, or modeling.

While human 3D reconstruction from images or videos has become a hot topic for many years, animal 3D reconstruction has progressed slowly. Although many mature techniques have been developed for human 3D reconstruction in recent years [1–4], it is unrealistic to directly migrate most of these methods from human beings to animals due to the following reasons: 1. The lack of annotated data. Most animals are not lab-friendly and cooperative, which means we can not use marker-based motion capture techniques or scan them to provide us with 2D or 3D ground truth directly. It can be very time-consuming to make 2D annotations for animals and almost impossible to label 3D annotations by hand. 2. More diversity in shape. For human beings, a widely used parametric model SMPL [5] uses principal component analysis (PCA) to characterize the space of human body shape. After removing unnecessary degrees of freedom, SMPL can use only 10 shape variables to parametrize the human body. The SMAL [6] model, an animal version of SMPL, uses the same PCA technology to characterize animal body shapes. However, although SMAL increases the number of

shape parameters to 41, it still fails to characterize many unseen animals, for example, camels and certain breeds of dogs. These reasons limit the development of 3D animal reconstruction.

Most existing animal reconstruction methods can be categorized into three different kinds. Besides template-based reconstruction, the other two popular ways are category-specific mesh reconstruction and template-free reconstruction. category-specific methods [7–13] can learn a 3D category-specific animal mesh from a spherical mesh gradually. Although some of these methods [9, 13] can be self-supervised, the key issue is that a learned category-specific mesh is isomorphic to a naive spherical triangular mesh with no information or limited information about animal skeletons, pose deformation, etc. This fatal flaw prevents this method from performing well for quadruped animals. On the hand, Template-free methods are becoming more and more popular recently. Banmo [14] can learn and build an animatable 3D animal model from casual videos.

In this thesis, we are going to present one template-based method and one template-free method respectively in Chapter 4 and Chapter 5. Both of our methods have an extremely easy setting. By only inputting a raw video without annotations, our methods can generate a 3D animal model in canonical space and time-varying body articulations.

1.1 Contributions

We develop two methods for 3D animal reconstruction and make the following contributions:

1. We propose a model-based method "SMALF" to reconstruct 3D animal shapes from a given video in Chapter 4. Building on the top of SMAL, our method gives each vertex the freedom to deform to match the ground truth or estimated 2D information. Our model demonstrates state-of-the-art reconstruction performance in the BADJA [15] animal video dataset and RGBD-Dog [16] dataset.

2. We designed an end-to-end self-supervised method called "AnimalRecon" using neural implicit representation to reconstruct a high-fidelity animatable 3D animal shape from a video of a fixed camera. Although we do not use an existing 3D mesh model to represent the shape, we utilize SMAL to initialize our neural implicit function and make it deformable. We outperform prior works in the RGBD-Dog dataset.

1.2 Thesis Outline

This dissertation contains the following chapters. Chapter 2 provides the relevant background for human and animal 3D reconstruction. Chapter 3 describes the database we work on, the data pre-processing methods, and the evaluation metrics that we will use in Chapter 4 and 5. In Chapter 4, we present our template-based method called SMALF (Based on SMAL). We demonstrate that our method can solve unusual body shapes that other template-based methods are hard to solve. In Chapters 5, we are going to present our model "AnimalRecon". A method using implicit shape representation to recover the animal body. Finally, in Chapter 6, we provide the summary and limitations of this thesis and discuss possible plans.

Chapter 2

Literature Review

This chapter reviews the existing methods for articulate object 3D reconstruction. All those works can be roughly categorized into three types: Template-based Reconstruction, Category-specific Mesh Reconstruction, and Template-free Reconstruction. We include not only animal reconstruction, but human reconstruction since these two tasks share many similarities. For example, humans and animals can be viewed as articulated shapes with non-rigid deformations.

2.1 Template-based Reconstruction

We first review template-based 3D reconstruction methods. These methods leverage a parametric model to fit with monocular images or videos. Currently, the SMPL [5] model plays a significant role in human reconstruction. The SMPL model provides an explicit function, given the shape and poses parameters it will return you the 3D location of each vertex and the vertices will form a human 3D triangular mesh. [17–23] estimate 3D body shape and pose parameters from estimated 2D or 3D joints location. Kanazawa *et al.*. [24] propose an end-to-end approach, they train a CNN that directly learns the mapping from images to pose and shape parameters of the SMPL model and utilize a pose discriminator to regularize the estimated pose. Kocabas *et al.*. [1] captures human motions from videos using an RCNN and leverages AMASS [25] to discriminate between real human motions and estimated temporal

pose. While for animal 3D reconstruction, the most popular parametric model is SMAL [6], which is the animal version of SMPL. From then on, many ideas are built on top of SMAL. SMALR [26] shows a way to create SMAL models from images by estimating a deviation from the SMAL fit. SMALST [27] designs an end-to-end approach and uses global image features to recover 3D meshes. SMALify [28] uses the SMAL model to fit the predicted silhouettes and joints as an energy minimization process. WLDO [29] leverages a statistical method to analyze pose and shape parameters for the SMAL model. In the RGBD-dog approach [16], they utilized a stacked hourglass network to produce a set of 2D heatmaps from a given depth image. This can determine the 3D coordinates of the body joints. A Hierarchical Gaussian Process Latent Variable Model (H-GPLVM) [30] is applied to refine the pose. However, this method requires depth image, which is not always available. Li *et al.*, [31] designed a two-stage approach to combine parametric and non-parametric representations. BARC [32] focuses on dogs, it uses breed information at training time via triplet and classification losses to learn how to regress realistic 3D shapes at test time.

2.2 Category-specific Mesh Reconstruction

This type of method [7–13] first initial a sphere triangular mesh as the mean shape and then gradually learn a category-specific template from a collection of images of the same category object, most commonly, birds. The recent work VMR [12] improves the performance by adding instance-specific details during testing time. UMR [9] and SMR [13] are both self-supervised and don't rely on any annotations. However, this kind of method is hard to generate space-time coherent meshes for a non-rigid object.

2.3 Template-free Reconstruction

LASR [33] proposes a method for articulated shape reconstruction from a monocular video. It takes advantage of dense two-frame optical flow to overcome the inherent

ambiguity in the nonrigid structure and motion estimation. ViSER [34] builds on the top of LASR and establishes long-range correspondence to force long-range video pixel correspondences to be consistent with an underlying canonical 3D mesh. LASSIE [35] proposes a neural part representation based on a generic 3D skeleton, which is robust to appearance variations and generalizes well across different animal classes. KeyTr [36] fits a dynamic point cloud to the video data using Sinkhorn's algorithm to associate the 3D points to 2D pixels and use a differentiable point renderer to ensure the compatibility of the 3D deformations with the measured optical flow.

Implicit functions play a significant role in many other template-free reconstructions. PIFu [3] makes use of a deep network to extract image-level features and concatenates pixel-level features and their corresponding 3D point depth information as the input of an implicit function to determine the occupancy for any given 3D location to obtain high-fidelity 3D clothed humans. PIFuHD [4] is built on top of PIFu and utilizes higher resolution features and predicted normal information to amend the result. However, these two methods require ground-truth 3D shapes to train, which does not apply to most animal datasets. StereoPIFu [37] focuses on binocular images and uses volume alignment feature and predicted depth map to guide implicit functions to train. PaMIR [38] trains an MLP to represent the human's implicit geometry from images and accomplish impressive results. However, they also require the corresponding high-quality 3D data of color images to train the model, which does not apply to animals in the wild.

Recently, some implicit representation methods, which can extract geometry and synthesize novel views based on multi-view images, have attracted researchers' attention. NeuralBody [39] reconstructs per frame's NeRF [40] field conditioned at body-structured latent codes and utilizes the NeRF field to synthesize new images. H-NeRF [41] utilizes an implicit parametric model [42] to reconstruct the temporal motion of humans. Neural Actor [43] integrates texture map features to refine volume rendering. IDR [44] combines implicit signed distance field and differential neural

rendering to generate high-quality rigid reconstruction from multi-view images. Concurrent IMAvatar [45] expands IDR to learn implicit head avatars from monocular videos. BANMo [14] can reconstruct articulated shapes for a wide range of objects and learn a neural blend skinning, making the output animatable by manipulating bone transformations. [46–48]

Chapter 3

Preliminaries

The materials covered here will be used in Chapters 4 and 5. Section 3.1 introduces the datasets called BADJA [15] and RGBD-Dog [16] which are used for testing our methodologies. Then, in Section 3.2, we explain the image prepossessing methods, which include the neural networks we use to generate silhouette and optical flow. Finally, Section 3.3 discusses the evaluation metrics we use for estimating our models.

3.1 Data

RGBD-Dog: Dog is one of the lab-friendly animals. As shown in fig 3.1 This dataset [16] is a large-scale 3D dog pose benchmark collected in the lab, which consists of five motions for seven different breeds of dogs from different angles. Additionally, the camera's intrinsic parameters are given. The reason why we chose this dataset for evaluation is that this dataset contains videos that meet our input requirements.

BADJA: BADJA is the benchmark animal dataset of joint annotations released with the paper [15]. This dataset contains seven videos of different animals in the wild, as shown in fig 3.2. Twenty joints are annotated by hand; It contains challenging animals for our task, for example, impala and camel, which can be considered as unseen animals for the SMAL model. In our work, we won't consider the two horse riding videos because our algorithm can not handle the interaction between humans and animals.

3.2 Data Pre-processing

Our approach assumes that a reliable segmentation of the foreground object is given, which can be manually annotated or estimated using instance segmentation and tracking methods [49]. Besides object silhouette, our method in Chapter 4 also requires 2D keypoints and optical flow for supervising. Off-the-shelf methods, [50] and [51], is used for retrieving keypoints and optical flow respectively. After getting all the 2D information, we will crop the image based on the silhouette as shown in fig 3.3.

3.3 Evaluation Metric

For 3D reconstruction, 3D Chamfer distances are the most popular evaluation metric. However, it is hard to evaluate the result quantitatively especially for an animal dataset, because 3D ground truth is missing. Therefore, We have to approximate the accuracy of 3D reconstruction by using 2D methods.

For our template-based method in Chapter 4, the joint locations can be easily computed. To approximate the accuracy of 3D shape and articulation recovery, we adopt the percentage of correct keypoint(PCK) [7, 11, 52] metric. **PCK** computes the percentage of joints within a normalized distance to the true joint locations. The normalized distance $d_{th} = 0.15\sqrt{|S|}$ is defined as the square root of 2D silhouette area $|S|$ times a threshold. Here we set the threshold as 0.15. The accuracy is averaged over all frames. Secondly, we use the **IoU** metric defined as the Intersection-over-Union of the projected silhouette and the ground truth silhouette. We use it to indicate the quality of the reconstructed 3D shape.

For our template-free method in Chapter 5, we don't define the joint location in our method. Thus, **IoU** will be the only metric we apply to our model.

Figure 3.1: **RGBD-Dog** includes seven dogs, each wearing a motion capture suit to provide landmark ground truth.

Figure 3.2: **BADJA** includes seven videos. 2D keypoint annotations are provided approximately every 5 video frames with the exception of the dog's video which is annotated densely.

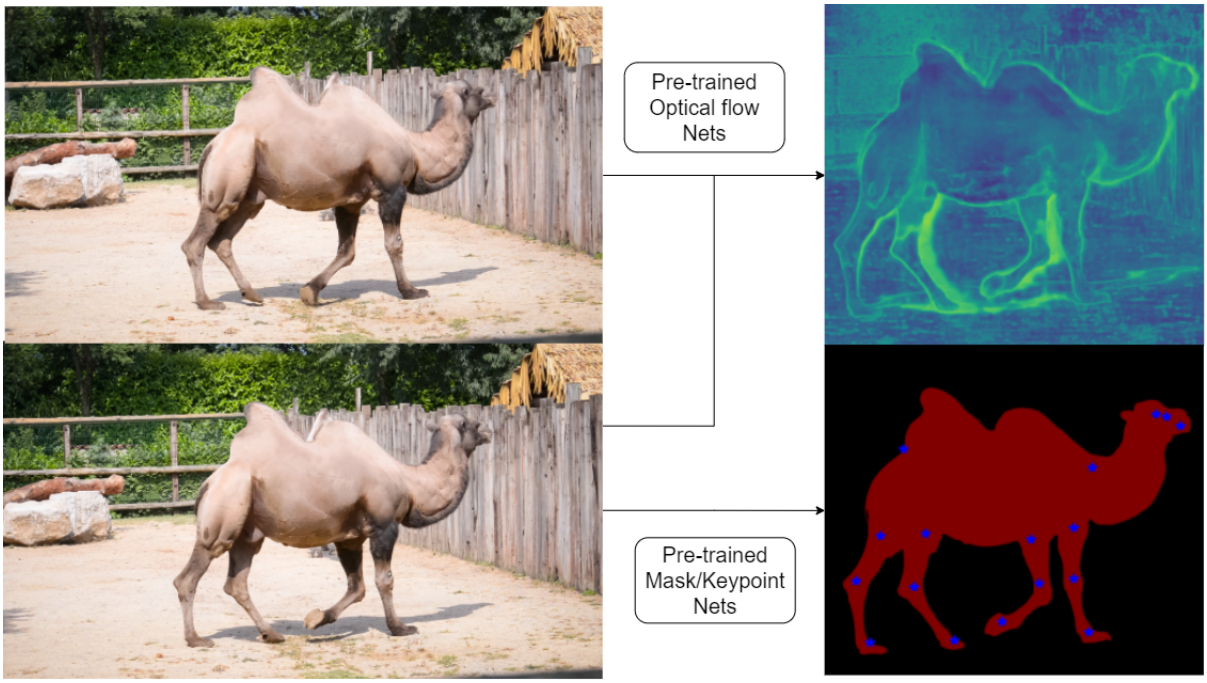

Figure 3.3: **Data preprocessing** Given an image, we retrieve 2D keypoints and mask using pre-trained network. Given two consecutive images, we can also get optical flow. Then We will crop the images based on the mask. Note: Optical flow is a technique used to describe motion, the image on the top-right is the result after visualization. The brighter area means there is a larger movement.

Chapter 4

Model-based Animal Reconstruction

In this section, we introduce an automatic system called SMALF to recover the 3D shape and pose of quadrupeds from a monocular video. To be specific, given a raw video with no annotations, we tackle the non-rigid 3D shape and motion estimation problem, which should include estimating the shape of the animal in canonical shape and the per-frame articulations. The system contains two stages. First, we utilize pre-trained DNNs to extract 2D keypoints and object silhouette optical flow from the whole video. The next will be our optimization stage, we fit a detailed 3D SMAL-based model with each frame of the video and find the best model-image alignment.
Why videos? Existing works [7, 11, 27, 29, 31, 32] usually focus on single image animal 3D reconstruction. However, a promise is that a large collection of targeting objects with silhouettes and 2D keypoints will be given for training their neural network. While videos are easier to acquire in the wild and we can extract more information from them than images, for instance, the optical flow and the motion.

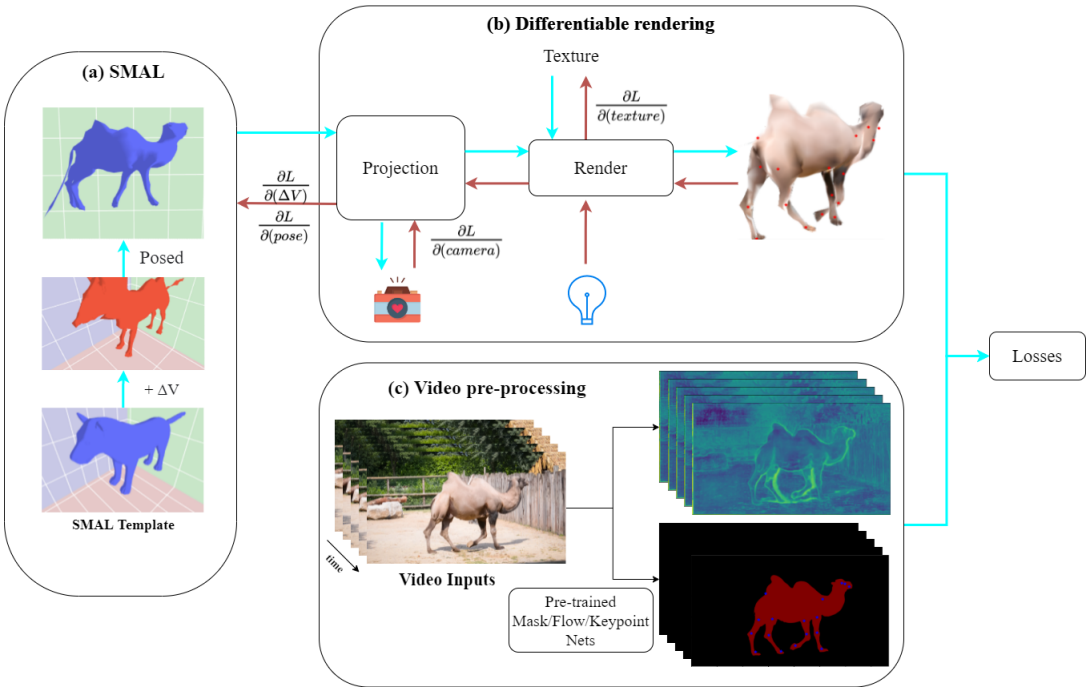

Figure 4.1: **Overall framework** Part(a) shows the SMAL model workflow. Part(b) shows the procedure of differentiable rendering. Part(c), given an input video, the pre-trained network will predict the silhouette, the keypoints location, and the optical flow in the image. The blue arrow shows the forward process, while the brown arrow shows the back-propagation flow.

4.1 Method

The framework of SMALF is summarized in Figure 4.1. In the following, we describe the main components we use, The SMAL [6] model, feature embedding, differentiable renderer [53], video pre-processing, and loss terms. Then, we will provide more details about an iterative optimization approach.

4.1.1 SMAL model

Our method is based on the Skinned Multi-Animal Linear (SMAL) model proposed by [6]. The SMAL model is a 3D animal triangular mesh $M \equiv (V, F)$ with vertices $V \in \mathbb{R}^{|V|\times 3}$ and faces F. The set of faces F defines the vertices connectivity in the mesh, and it remains fixed in the SMAL model. The vertices $V_{posed} = \mathcal{M}(\beta, \theta)$ are parameterized by shape $\beta \in \mathbb{R}^{41}$ and pose $\theta \in \mathbb{R}^{105}$. The shape parameters are

a vector of the coefficients of a learned principal component analysis (PCA) shape space. While the pose parameters denote the joint angle rotations (35×3 Rodrigues parameters), which control the motion of the articulated limb movement. The body joints J_{3D} are simply defined as a linear combination of the mesh vertices. A linear transformation matrix W is provided by the SMAL model. The 3D joints location can be calculated as

$$J_{3D} = \mathcal{W} \times V_{posed}. \tag{4.1}$$

On the top of SMAL, we give each vertex the freedom to deform. We model the new vertex positions as $V_f = \Delta V + V$, the summation of an instance-specific deformation ΔV and the vertices in canonical space $V = \mathcal{M}(\beta, \vec{0})$.

4.1.2 Video-specific Feature Embedding

As shown in Figure 4.3, we learn pixel and mesh vertices embedding that map corresponding pixels in different frames to the same point on a canonical 3D mesh. Intuitively, consider a particular region on the canonical surface mesh that is the "left back foot" of an animal. The mesh vertices embedding captures a descriptor for the left back foot, which can then be matched to pixel-level descriptors at each frame. Given an input image I_t, the pixel-wise descriptor embedding is computed by a U-Net [54] encoder with a learnable weights α:

$$F_I[x, y, t] = CNN_\alpha(I_t)[x, y] \in R^{16}, \tag{4.2}$$

where [x, y, t] are pixel locations at frame t. The surface embedding is computed by a position-encoded MLP with learnable weights β:

$$F_S[X, Y, Z] = MLP_\beta(X, Y, Z) \in R^{16}, \tag{4.3}$$

where (X, Y, Z) is a 3D point on the surface of the animal mesh in the canonical space, augmented with Fourier positional encoding [40]. The parameter weights of the two embeddings are optimized in the optimization routine.

4.1.3 Differentiable Rendering

With the development of differentiable rendering recently [53, 55, 56], these technology tools provide us with more efficiency and modularity to customize our own functionality to meet our requirements and experiment with alternate formulations. We denote the differentiable rendering function that renders the animal mesh to an image as $\mathcal{R}(M, T, C, P)$, where T is the texture of the mesh, C is the camera position, and P are the projection matrix to covert world space to camera space. T, C, and P are all unknown here. In our method, we will give them a reasonable initial value. Then, since the rendering function is differentiable, we can backpropagate through the renderer to learn the texture of the mesh T, the position of the camera C, and the projection matrix P as shown in Figure 4.1.

4.1.4 Video Pre-processing

Our approach assumes that a reliable segmentation of the foreground object is given, which can be manually annotated, or estimated using instance segmentation and tracking methods [57]. Besides object silhouette, our method requires keypoints and optical flow for supervising. Off-the-shelf methods, [50] and [51], is used for keypoints and optical flow estimation respectively.

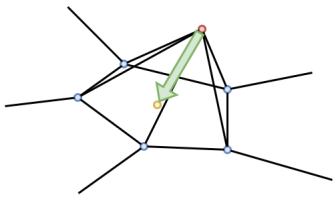

Figure 4.2: **Laplacian vector**

4.1.5 Optimization Routine

According to the description above, for an N-frame video, the set of all optimizable parameters is:

$$X = \{\Delta V, T, \alpha, \beta\} \cup \{C_i, P_i, | i \in 1, ..., N\}, \tag{4.4}$$

where ΔV is vertex-level deformation, T is the texture of the mesh, α, and β are the weights of embeddings, C_i and P_i are the camera parameters and animal pose parameters respectively for the i-th frame.

Our target is to design a workflow and a set of loss functions to optimize X according to the input video. We design an iterative fitting routine which is shown in Figure 4.1. At each iteration, we simply go through the video in order. Each time, we choose two consecutive frames from the video. Thus, we have two raw images, two silhouettes, two sets of joint locations, and one optical flow to supervise us in learning the video-specific animal mesh, camera parameters, etc. The object is to minimize a group of loss terms. The loss terms can be categorized into four groups, mesh loss, temporal dimension loss, 2D matching loss, and embedding matching loss.

Mesh loss is a combination of laplacian smooth loss and mesh normal consistency loss. Both are used to preserve the smoothing of triangular meshes. Mesh laplacian loss is proposed by Nealen *et al.* [58] For each vertex on the mesh, we calculate a uniform Laplacian loss which is the distance between the vertex and the centroid of its neighboring vertices. As shown in Figure 4.2,

$$\mathcal{L}_{laplacian} = \sum_{i=1}^{V} \|V_i - \frac{1}{|N_i|} \sum_{e_{ij}} V_j\|_2. \tag{4.5}$$

While $\mathcal{L}_{normal}$ computes the normal consistency for each pair of neighboring faces.

2D matching loss is a combination of keypoints loss, silhouette loss, image loss, and flow loss. $\mathcal{L}_{kp}$ is defined as the distance between the projected 3D keypoints and pre-estimated keypoints.

$$\mathcal{L}_{kp} = \|J_{estimated2D} - \Pi(J_{3D}, P)\|_2 \tag{4.6}$$

$\mathcal{L}_{sil}$ and $\mathcal{L}_{flow}$ are defined as the $L2$ loss between the rendered result and measurements. $\mathcal{L}_{img}$ is computed as the perceptual distance [59] between the masked input image and the rendered image.

Temporal loss As the input is a video, we assume that the camera position and the

animal posing will not change dramatically between the two frames. Since we want to punish more on the large change, we use MSE to regularize them. The loss terms are defined here,

$$\mathcal{L}_{camera} = \|C_{t+1} - C_t\|_2, \tag{4.7}$$

$$\mathcal{L}_{pose} = \|\theta_{t+1} - \theta_t\|_2. \tag{4.8}$$

Embedding matching loss consists of three loss terms: consistency, contrastive, and 2D cycle losses. Where $L_{consistency}$ is defined to minimize the difference between the rendered surface embedded value and the image pixel embedded value:

$$\mathcal{L}_{consistency} = \sum_{(x,y)} 1 - cos(Rend(F_S[x, y, z]), F_I[x, y]) \tag{4.9}$$

where $cos(\cdot)$ denotes the inner product between two normalized vectors and $Rend$ is a differentiable renderer. Contrastive loss is to ensure pixel embeddings only match surface embeddings rendered at the pixel location:

$$\mathcal{L}_{contrastive} = \sum_{(x,y)} \|location[x, y] - Prob[x, y]\|_2, \tag{4.10}$$

where $location(\cdot)$ returns one hot vector which indicates the corresponding surface point of a pixel and $Prob(\cdot)$ is the estimated pixel-to-surface mapping probability distribution computed based on the similarity of pixel-to-surface embeddings. By minimizing this loss, the embeddings of unmatched surface-pixel will be pulled away. The above consistency and contrastive losses aim to learn pixel-surface embeddings that are consistent over video frames and discriminative over different surface locations. However, in terms of optimizing articulation parameters, consistency and contrastive losses based on differentiable rendering tend to suffer from bad local optima. For instance, when the rendered body part lies outside the ground-truth object silhouette, a gradient update of articulation parameters would likely not incur a lower loss. Therefore, we add 2D cycle loss to prevent this from happening, we use feature

matching to compute the expected surface coordinate Prob[x, y] at every pixel and ensure the differentiably rendered canonical surface coordinate lands back on the original pixel coordinate (x, y),

$$\mathcal{L}_{2D-cycle} = \sum_{(x,y)} \|Rend(Prob[x, y]) - (x, y)\|_2. \tag{4.11}$$

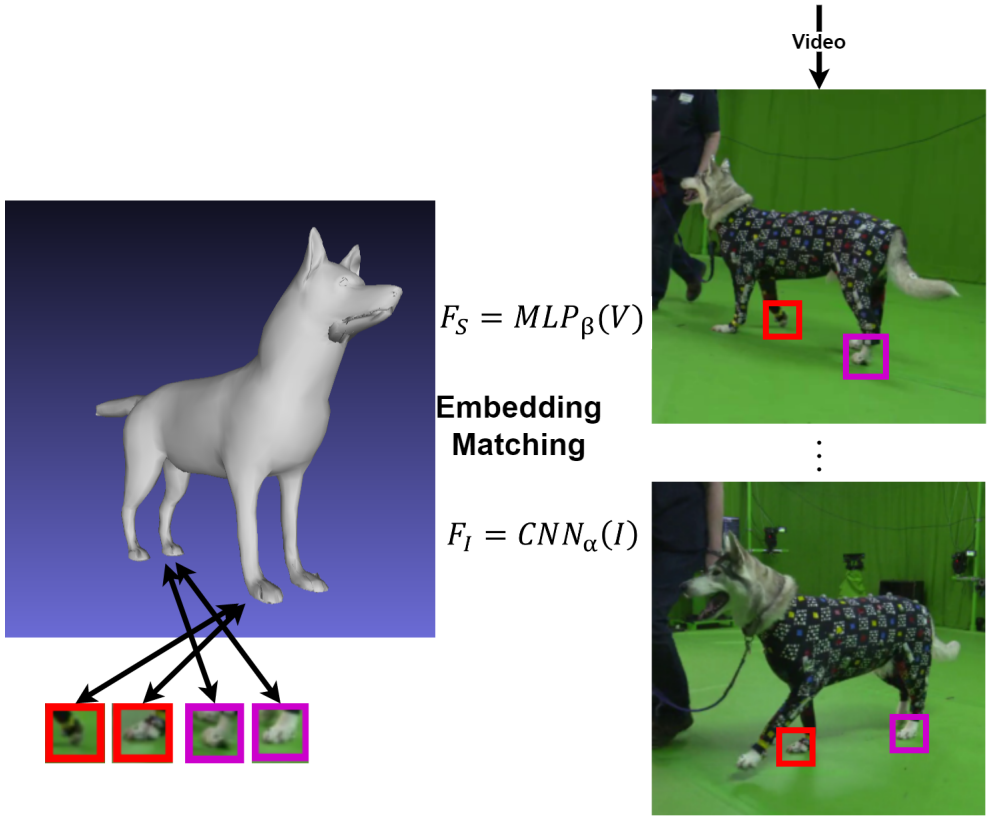

Figure 4.3: **Video-specific** pixel-surface embedding

4.2 Experiments

4.2.1 Dataset

BADJA dataset and RGBD-Dog dataset are chosen for our evaluation. For more details, see section 3.1.

4.2.2 Implementation details

We first optimize our model on every single image using 2D keypoints loss only. Our initial result heavily relies on the 2D keypoints estimation. As shown in fig 4.4 and 4.5, the 2D keypoints estimation result is generated by a pre-trained network and fine-tuned on our dataset. Then, we start to add other losses during the optimization process and keep updating the weights of each loss. According to [29], applying silhouette loss too early will likely lead our optimization process to local minima. Additionally, we found that estimated keypoints and silhouettes can be inaccurate in specific frames and the original image and optical flow are more reliable. Thus, we will keep adding weight to these two losses. For a camera, we will initiate it with a far position. And the keypoints loss will help us to find the proper camera position and orientation. This is also significant for our model not to get stuck in local minima.

4.2.3 Comparison to Baselines

We first compare our method to various baseline methods [28, 29, 31, 32] on the RGBD-Dog dataset and the BADJA dataset. The visual result is shown in Figure 4.6. SMALify [28] is an optimization-based method whose input is a raw video. WLDO, Coarse-to-fine, and BARC are all predication-based methods. Given an input image, it will return predicted pose and shape parameters. WLDO and BARC focus on dogs, WLDO learns a richer prior over shapes than the original SMAL model for dogs. While BARC can produce recognizable breed-specific shapes for dogs. Coarse-to-fine estimates the animal's pose and shape parameters first and then refine the result by using the comparison between the first stage result and the image. In terms

Method	I0U	PCK@0.15				
		Avg	Legs	Tail	Body	Face
Ours	**80.4**	**80.1**	**78.1**	**69.9**	**83.1**	**92.1**
SMALify	51.9	76.1	69.2	59.2	83.6	85.1
WLDO	30.8	13.9	13.2	11.5	13.8	12.5
Coarse-to-Fine	63.5	61.8	64.8	45.6	71.6	70.5
BARC	73.7	63.5	64.2	50.3	76.1	80.9

Table 4.1: IoU and PCK@0.15 for the RGBD dataset.

Method	IoU	PCK@0.15				
		Avg	Legs	Tail	Body	Face
Ours	**78.8**	**78.1**	**76.3**	**62.1**	**83.2**	**82.9**
SMALify	47.7	64.4	69.9	53.8	68.2	71.9
WLDO	32.8	15.4	12.7	16.7	20.4	15.8
Coarse-to-Fine	59.0	58.7	64.1	35.5	62.3	61.0
BARC	70.7	62.0	64.2	46.1	72.1	70.6

Table 4.2: IoU and PCK@0.15 for the BADJA dataset.

of 2D error metrics (IoU and PCK), our method SMALR outperforms prior baseline methods, as shown in Table 4.1 and 4.2. We test our model performance specifically on unseen animals. The result shown in Table 4.3 proves our method can handle unseen animals better than baseline methods.

4.2.4 Ablation study

We also investigate the effect of different loss terms by dropping one component from our entire pipeline each time and examine the effect on the PCK/IoU performance. As shown in Table 4.4, we evaluate our SMALF model by removing embedding matching loss and mesh loss respectively. The result shows that these two losses play a positive

Method	IoU	PCK@0.15				
		Avg	Legs	Tail	Body	Face
Ours	**70.1**	**75.1**	**73.4**	**64.9**	**77.0**	**88.2**
SMALify	41.6	66.8	59.8	49.5	73.1	75.2
WLDO	27.7	14.5	14.7	13.2	15.3	11.2
Coarse-to-Fine	53.6	50.4	54.6	34.5	61.5	59.5
BARC	51.0	49.3	52.7	38.3	63.8	68.0

Table 4.3: IoU and PCK@0.15 for unseen animals.

Method	IoU	PCK@0.15				
		Avg	Legs	Tail	Body	Face
Ours	**80.3**	**79.2**	**78.3**	**67.7**	**83.4**	**89.0**
-W/O Embedding loss	74.2	74.2	72.9	62.2	79.7	85.6
-W/O Mesh loss	71.1	73.5	74.0	59.3	81.4	74.0

Table 4.4: Ablation study for the importance of each loss component

role in our model.

Figure 4.4: **2D keypoints estimation** on RGBD-dog dataset

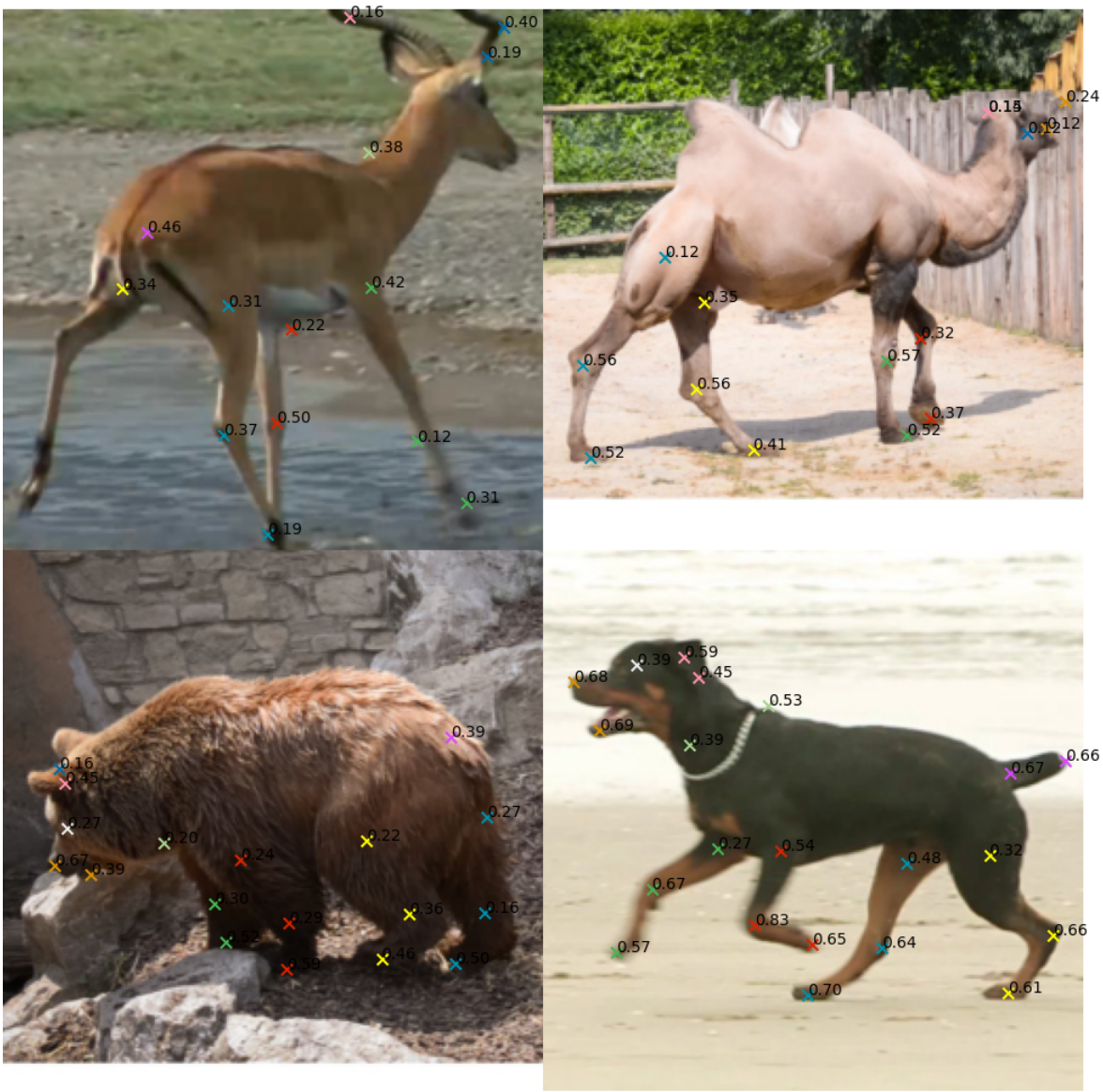

Figure 4.5: **2D keypoints estimation** on BADJA dataset

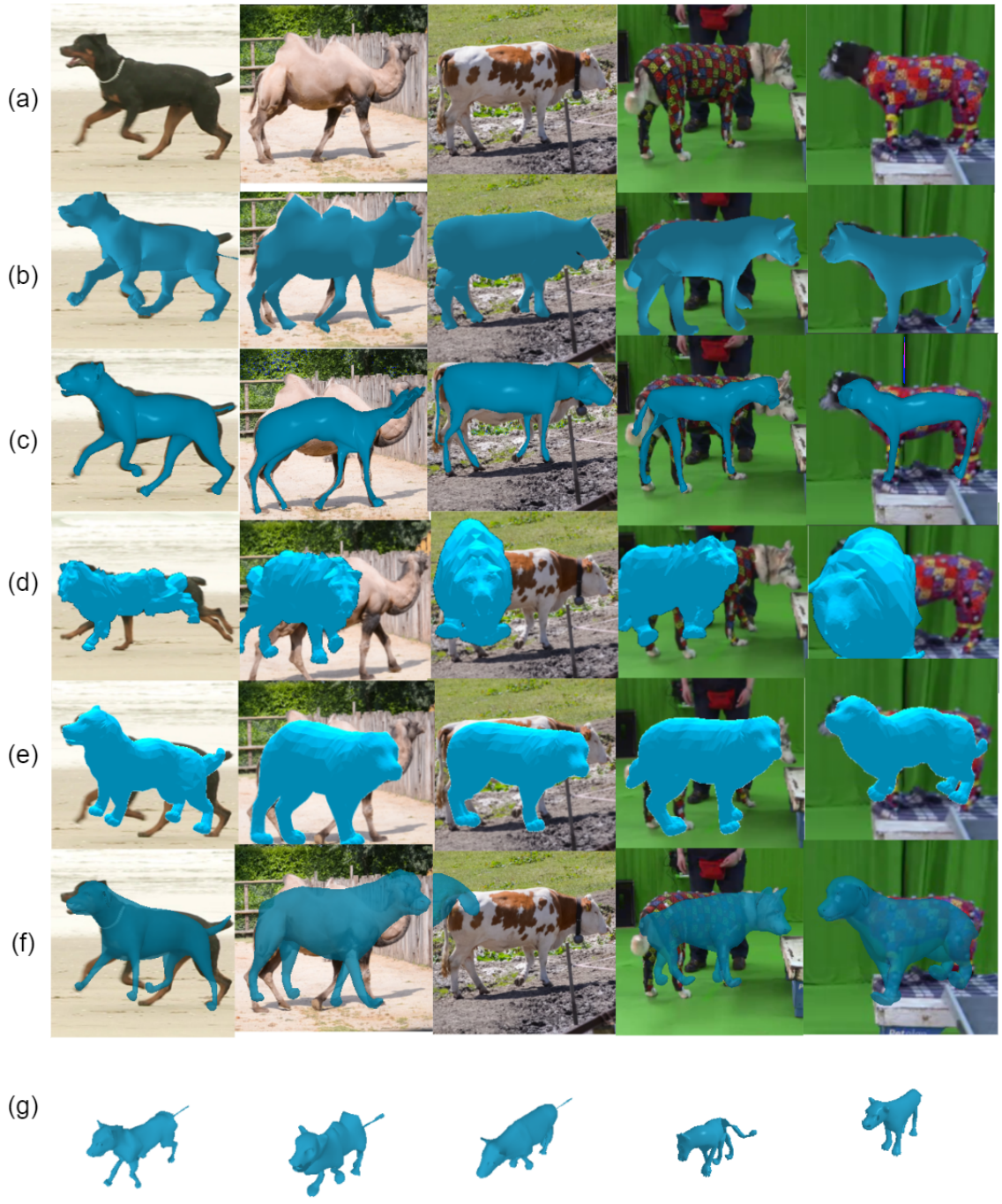

Figure 4.6: **Qualitative results** comparison to existing methods on the RGBD-Dog and BADJA datasets. For each sample, we show: (a) the input image, (b) our result, (c) the SMALify result, (d) the WLDO result, (e) the Course-to-Fine result, (d) the BARC result and (g) an alternative view of our result

Chapter 5

Animal Reconstruction Using Implicit Function

Our goal is to reconstruct a high-quality and space-time coherent 3D animal body from a monocular video. In chapter 4, we demonstrate that we can do the animal reconstruction using a polygon mesh (SMAL model). Motivated by cutting-edge technologies [3, 4, 40], we are aiming to use an implicit neural function to represent the animal's body shape. Unlike polygon meshes, implicit neural representations can capture geometry at high fidelity. However, most approaches [47, 60] focus on static objects. Thus, we not only need to construct an implicit neural function but find a way to make it deformable. In this chapter, we design a new framework called "AnimalRecon" to fulfill our aim. One important benefit of implicit neural representations is that they easily support the modeling of arbitrary surface topology. One favorable family of implicit representations is occupancy functions. In Section 5.1.1 and Section 5.1.2, we present how we represent an object by an occupancy function and how to make our implicit shape representation deformable. Then, in Section 5.1.3, we implement the method proposed by SNARF [61] to learn the initial shape and skinning weights. 5.1.4 shows how we use a differentiable ray-casting algorithm to render the image. In Section 5.1.5, we present the optimization workflow and define our loss functions.

5.1 Method

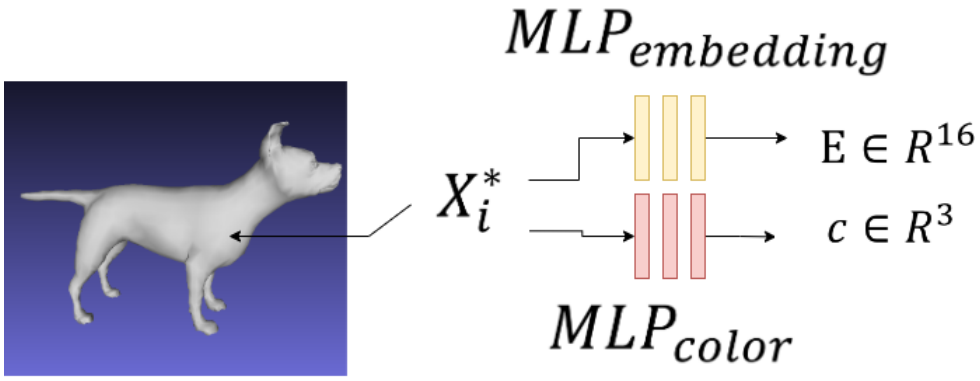

Figure 5.1: Implicit shape with properties

5.1.1 Implicit Neural Representation

Like SMAL [6], We want to represent an articulated object by its shape in canonical space and skinning weights for deformation. To represent the shape of an animal in a canonical space, we implement an implicit neural function called "occupancy network" proposed by [46] to predict the occupancy probability for any input 3D point x in a certain pose p:

$$f_\theta : R^3 \times R^{n_p} \rightarrow [0, 1], \tag{5.1}$$

Where θ are the parameters of our occupancy network and n_p is the dimensionality of the pose p. The canonical shape is implicitly defined as the 0.5-level set of the neural function S:

$$S_\theta = \{x | f_\theta(x, \vec{0}) = 0.5\}. \tag{5.2}$$

A 3D point $x \in R^3$ in a canonical space is also associated with two properties: color $c_x \in R^3$ and a learned canonical embedding $E_x \in R^{16}$, as shown in figure 5.1. For color, a Multilayer Perceptron (MLP) network with learnable weights γ is used

to predict the color of a point:

$$c_x = MLP_{color}(x). \tag{5.3}$$

Another MLP network with learnable weights β maps 3D points to a feature descriptor that can be matched by pixels from different frames of a video:

$$E_x = MLP_{embedding}(x), \tag{5.4}$$

which enables long-range correspondence across frames and videos. These two MLPs can be learned in a self-supervised manner. See details in Section 5.1.5.

5.1.2 Neural Blend Skinning

Like other animatable objects, we use linear blend skinning (LBS) to control the non-rigid deformation induced by skeleton changes. To achieve this goal, we represent an LBS weight field in canonical space using another neural network:

$$w_\alpha : R^3 \rightarrow R^{n_b}, \tag{5.5}$$

where α are the network parameters and n_b denotes the number of bones. Following the setting of LBS, we use a softmax activation function to enforce the weights $w = \{w_1, w_2, ..., w_{n_b}\}$ of each point x to satisfy $w_i > 0$ and $\sum_i w_i = 1$

Let x_p denote the position of a 3D point x in pose p. To compute the x_p, we have the bone transformations $B = \{B_1, ..., B_{n_b}\}$ corresponding to a particular body pose p, and x_p can be computed by the formula:

$$x_p = d_\alpha(x, B) = \sum_{i=1}^{n_b} w_\alpha(x) B_i x. \tag{5.6}$$

5.1.3 Model Initialization

Inspired by SelfRecon [48], we want to initialize the canonical shape and neural blend skinning weights by leveraging an explicit model. For a given video, we adopt a

model-based method to generate the corresponding shape a and pre-frame's pose $\{b_i | i \in \{1, ..., N\}\}$ parameters of SMAL model. Then we can get a set of posed 3D meshes with shape a. Then, we use the method described in SNARf [61] to initialize the canonical shape and neural blend skinning weights. Specifically, given animal meshes in canonical and deformed spaces, we first find all canonical correspondences of a deformed point via iterative root finding. We then predict the occupancy of the deformed point by aggregating occupancy information from the set of all roots while conditioning on the object pose. This enables our model to learn both the shape and the skinning weights to end from deformed observations.

5.1.4 Differentiable Neural Rendering

Differentiable rendering systems can usually be categorized into two flavors, differentiable rasterization and differentiable ray casting. If we want to use differentiable rasterization, we need to convert our implicit shape into a polygon mesh that would require a very high polygon count, which is computationally expensive. While differentiable ray casting can render an implicit shape directly.

Given a pixel, indexed by p, associated with some input images. Let

$$RAY_p = \{m + t \cdot v_p | t \geq 0\}, \tag{5.7}$$

denotes the ray through pixel p. Where m is the center point of the camera and v_p is the direction of the ray passing through p. Let

$$i_p = I(RAY_p, S_\theta), \tag{5.8}$$

denotes the first intersection of RAY_p and the surface S_θ. The 3D position of the intersection, the color of the 3D point, and the environment determine the color of the pixel p

$$C_p = M_p(\phi, \gamma, \theta), \tag{5.9}$$

where M_p is an MLP with learnable weights ϕ to approximate the rendering equation. Note, here we assume our camera is at a fixed position. Our M is used to simulate the environment, for example, lighting. We utilize C_p in the loss function to train our model's parameters simultaneously.

5.1.5 Optimization

To register pixel observations at different frames, we use pixel-to-surface mapping. For surface embedding, we have introduced it in Section 5.1.1. While for pixel embedding, given an input image I_t, is the pixel-wise descriptor embedding is computed by a U-Net [54] encoder with learnable weights τ:

$$F_I[x, y, t] = CNN_\tau(I_t)[x, y] \in R^{16}, \tag{5.10}$$

where [x, y, t] are pixel locations at frame t. The pixel embedding and the surface embedding will be jointly optimized in our optimization stage.

After all the pre-steps settle down, we start to go through the video iteratively to optimize our model. According to the description above, for an N-frame video, the set X of all optimizable parameters is

$$X = \{\alpha, \beta, \theta, \gamma, \phi, \tau\} \cup \{b_i | i \in 1, ..., N\}, \tag{5.11}$$

Where $\alpha, \beta, \theta, \gamma, \phi, \tau$ are learnable weights of our MLPs, and b_i is the pose for the i-th frame. The loss functions are defined below to optimize X. Let $I_p \in [0, 1]^3, O_p \in \{0, 1\}$ be the ground truth RGB and mask values, respectively, corresponding to a pixel p

RGB Loss We formulate the RGB image loss as:

$$\mathcal{L}_{RGB} = \frac{1}{|P|} Where_{p \in P^{in}} |I_p - M_p(\theta, \phi, \gamma)| \tag{5.12}$$

where $|.|$ represents the L_1 norm, and $P^{in} \subset P$ denotes the subset of pixel p where an intersection can be found. Intuitively, this loss requires that the rendered images match the input images.

Mask Loss

$$\mathcal{L}_{MASK} = \frac{1}{|P|} \sum_{p \in P} CE(O_p, N_p(\theta, \phi)), \tag{5.13}$$

where CE is the cross-entropy loss and N is the mask renderer.

Optical Flow Loss Besides RGB Loss and Mask Loss, we compute flow reconstruction loss, defined as the $L2$ loss between the rendered result and measurements.

Motion Loss

$$\mathcal{L}_{motion} = \|b_{t+1} - b_t\|_2. \tag{5.14}$$

This loss is to ensure that the animal posing will not change dramatically between the two consecutive frames

Embedding Matching Loss consists of three loss terms: consistency, contrastive, and 2D cycle re-projection losses. Where $L_{consistency}$ is defined to minimize the difference between the rendered surface embedded value and the image pixel embedded value:

$$\mathcal{L}_{consistency} = \sum_{(x,y)} 1 - cos(Rend(MLP_{embedding}[x, y, z]), F_I[x, y]) \tag{5.15}$$

where $cos(\cdot)$ denotes the inner product between two normalized vectors and $Rend$ is a differentiable renderer. Contrastive loss is to ensure pixel embeddings only match surface embeddings rendered at the pixel location:

$$\mathcal{L}_{contrastive} = \sum_{(x,y)} \|location[x, y] - Prob[x, y]\|_2, \tag{5.16}$$

where $location(\cdot)$ returns a one-hot vector, which indicates the corresponding surface point of a pixel, and $Prob(\cdot)$ is the estimated pixel-to-surface mapping probability distribution computed based on the similarity of pixel-to-surface embeddings. By minimizing this loss, the embeddings of unmatched surface-pixel will be pulled away. The above consistency and contrastive losses aim to learn pixel-surface embeddings that are consistent over video frames and discriminative over different surface locations. However, in terms of optimizing articulation parameters, the consistency and

contrastive losses based on differentiable rendering tend to suffer from bad local optima. For instance, when the rendering of a body part is outside the ground-truth object silhouette, a gradient update of articulation parameters would likely not incur a lower loss. Therefore, we add 2D cycle re-projection loss, as shown in 5.2, to prevent this from happening, we compute the expected surface coordinate Prob[x, y] at every pixel and ensure the differentiably rendered canonical surface coordinate lands back on the original pixel coordinate (x, y),

$$\mathcal{L}_{re-projection} = \sum_{(x,y)} \|Rend(Prob[x, y]) - (x, y)\|_2, \tag{5.17}$$

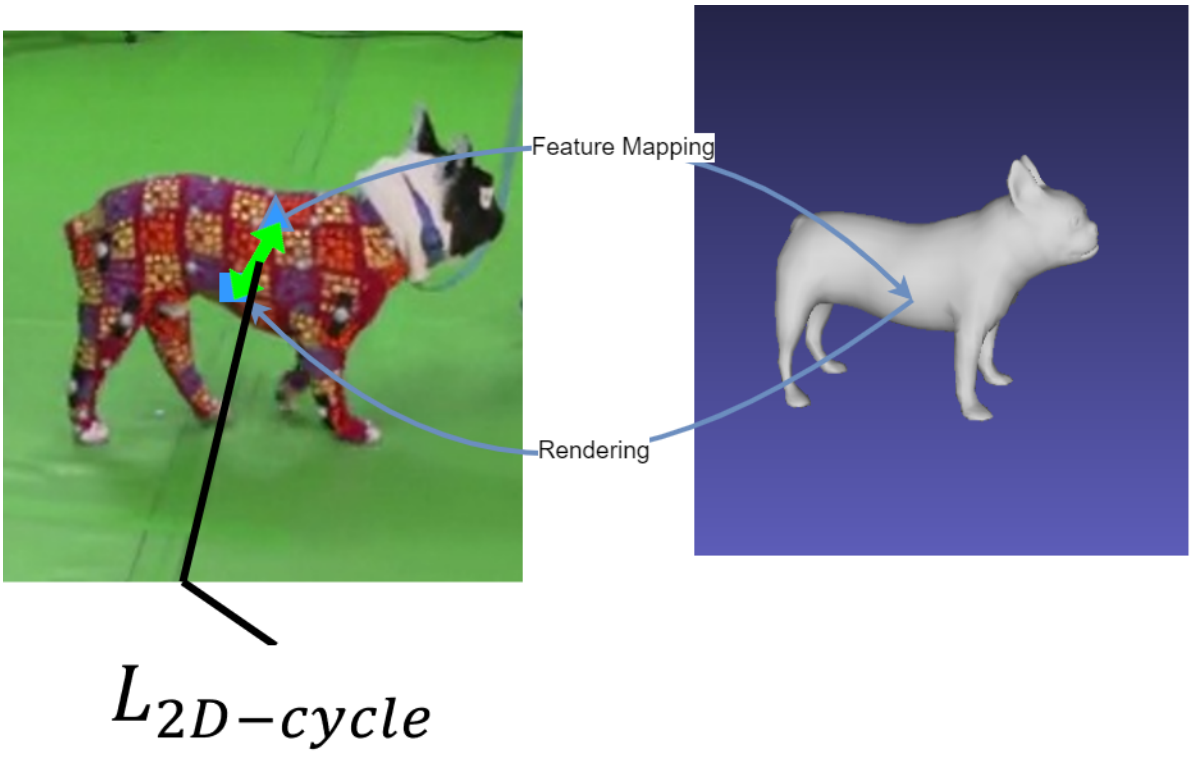

Figure 5.2: **2D cycle re-projection loss** ensures the pixel-to-surface matches can be mapped back to their pixel locations.

5.2 Experiments

5.2.1 Dataset

Because our setting requires a video taken by a fixed camera with known intrinsic parameters, the RGBD-Dog dataset (see Section 3.1) is the only dataset that fulfills our requirements to the best of our knowledge.

5.2.2 Implementation Details

Architecture For representing 3D shapes, we used level sets of MLP $f(x;\theta); f : R^3 \times R^m \longrightarrow R$, with 8 layers each containing 256 hidden units and a single skip connection from the input to the middle layer. For neural linear skinning weights, we use MLP with 4 layers, each containing 128 hidden units. We can initialize the weight of these two MLPs and per-frame pose parameters using the method described in Section 5.1.3. The visual result shows in Figure 5.3. The MLP for our renderer consists of 4 layers with hidden layers of width 512. We use the non-linear maps of NeRF [40] to improve the learning of high-frequencies. Our implementation of appearance models also follows NeRF.

5.2.3 Baselines

We define the following baseline approaches for comparison:

SMALify [28] is a SMAL-based and optimization-based method. It presents a system to recover the 3D shape and motion of a wide variety of quadrupeds from video. The system comprises a machine learning front-end that predicts the candidate 2D joint positions, a discrete optimization that finds plausible joint correspondences, and an energy minimization stage that fits a detailed 3D model to the image.

BARC [32] is a SMAL-based and prediction-based method. Given an input image, it will return predicted pose and shape parameters. BARC focuses on dogs

and modifies the SMAL shape space to be more appropriate for representing dog shapes. Then it utilizes a DNN to classify the breed of dogs, which helps to produce recognizable breed-specific shapes.

BANMo [14] is a template-free method that uses implicit functions to represent 3D shapes. It exploits the dense temporal correspondence in video frames to reconstruct articulated shapes and can learn the shape, the joint locations, the bone length, and the skinning weights of an animal from scratch. We implement this method on the RGBD-Dog dataset. However, it doesn't provide us with reasonable results. We further analyze the reason, and we find that this method is heavily dependent on a pre-trained model DensePose-CSE [62] to provide rough root body pose registration. However, in the RGBD-Dog dataset, dogs are clothed, which gives the DensePose-CSE model a huge difficulty in estimating universal canonical maps. Thus, we won't show the result for BANMo.

5.2.4 Comparison to Baselines

We compare our method with the baseline methods. The visual result comparison is shown in Figure 5.4. Regarding the IoU, our method AnimalRecon outperforms prior methods, as shown in Table 5.1. In Figure 5.5, we show the visual result for a video. Each second raw shows another view of the 3D shape.

5.2.5 Ablation study

During our experiments, we found the performance of our method might be highly related to the length of the video. We designed an ablation study to test our hypothesis. Given a video, we extract 400 frames, 200 frames, and 100 frames, respectively, as the input of our model. The result in terms of the IoU is shown in Table 5.2. Given more frames, our method can get a better result.

Method	I0U
Ours	**91.4**
SMALify	51.9
WLDO	30.8
Coarse-to-Fine	63.5
BARC	73.7

Table 5.1: Comparison to baseline methods on the RGBD-Dog dataset (IoU)

Method	I0U
400 frames	**91.0**
200 frames	89.2
100 frames	85.8

Table 5.2: Comparison of using different numbers of frames as input on the RGBD-Dog dataset (IoU)

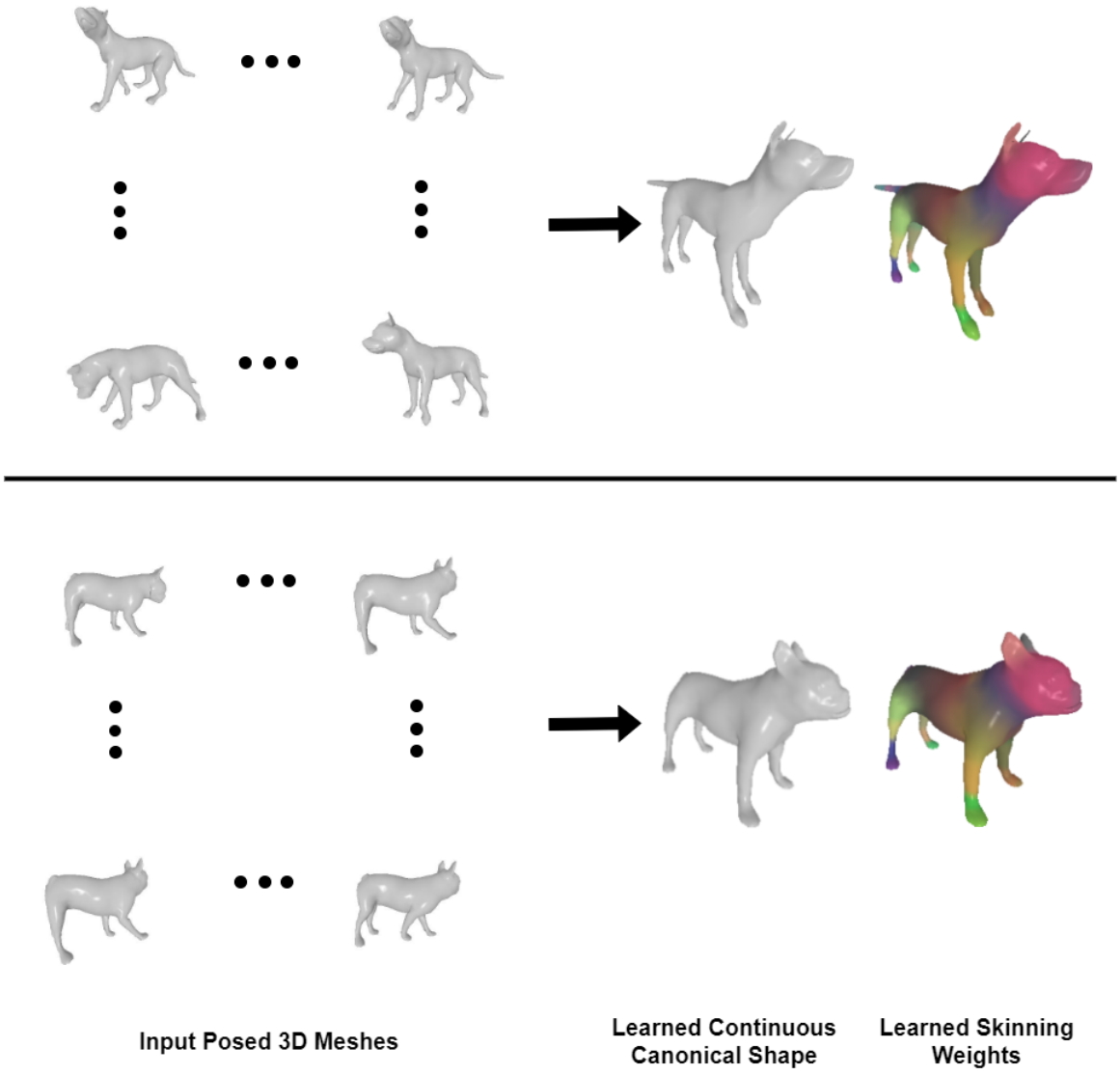

Figure 5.3: From a set of posed meshes(Left), we learn an implicit neural 3D shape and a skinning field(Right) in the canonical pose.

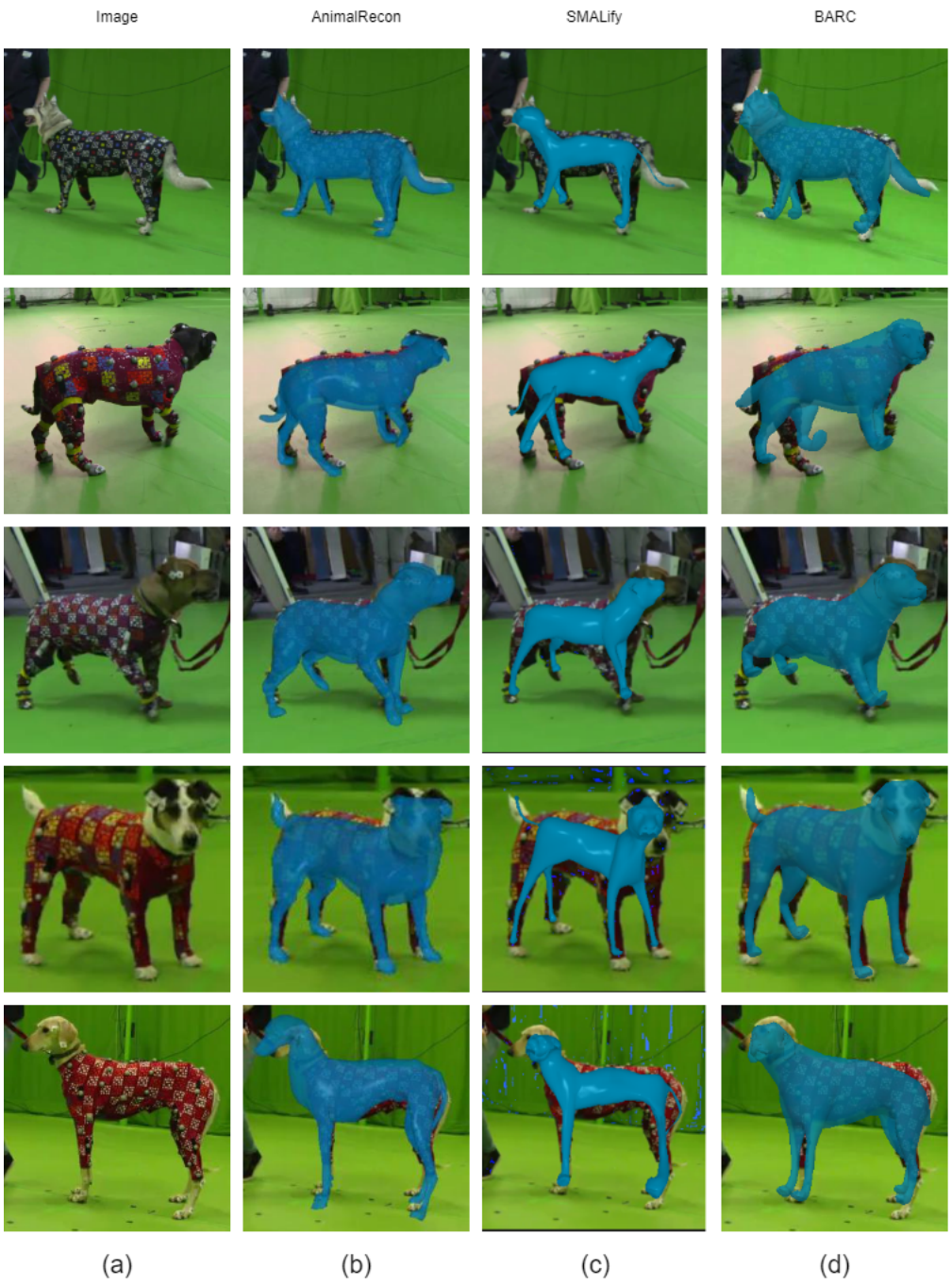

Figure 5.4: **Qualitative results** comparison to existing methods on the RGBD-Dog dataset. For each sample, we show: (a) the input image, (b) our result, (c) the SMALify result, and (d) the Barc result

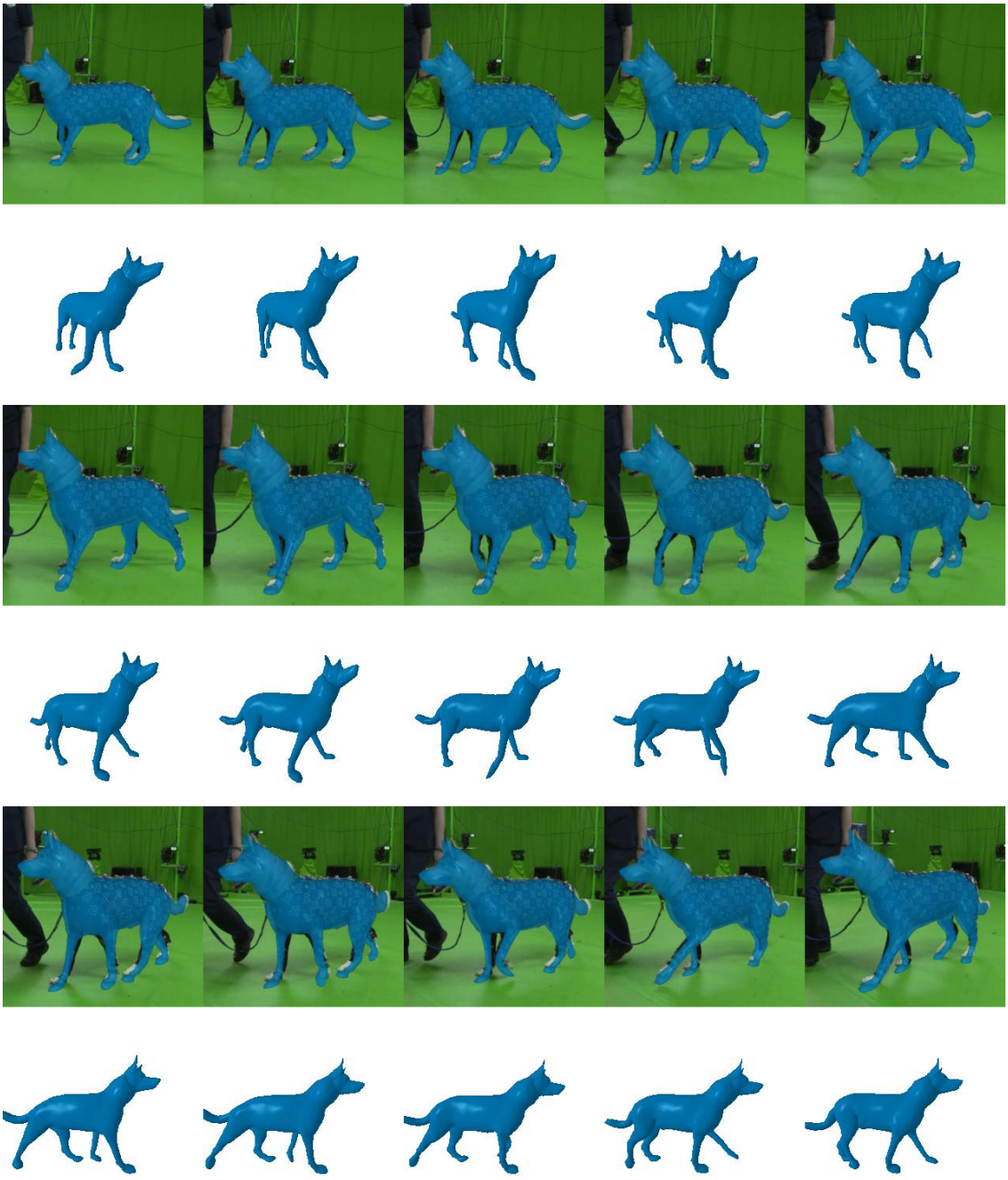

Figure 5.5: **Video result** on the RGBD-Dog dataset. Each second raw shows another view of the 3D shape.

Chapter 6

Discussion and Conclusion

There are two main contributions to this thesis.

(1) We design a framework "SMALF" for a 3D mesh to match the estimated and ground-truth 2D information, and we demonstrate state-of-the-art reconstruction performance among model-based methods in the BADJA [15] animal video dataset, and the RGBD-Dog [16] dataset.

(2) We proposed a framework AnimalRecon in Chapter 5. We use an implicit function to represent animals, and the implicit function shape can handle large body movements and recover space-time coherent deformation.

In these two works, we use similar loss functions to optimize our model. The main difference between our two works is for the first one, we use a parametric 3D mesh model to represent the shape of an animal. And we use the rasterization algorithm to render a 3D object into a 2D image. While in the second work, we use an implicit neural function to represent the shape of an animal. And we switch from rasterization to the ray-tracing algorithm to render a 2D image. The nature of implicit functions makes them achieve better results. Will SMAL and SMPL models be history? Time will tell us the answers.

6.1 Limitation

Our methods have several limitations. SMALF relies on pre-trained networks to provide rough root body pose registration. For both our methods, any obstacles during the video can affect the performance. AnimalRecon utilizes the SMAL model to initialize neural implicit function and the skinning weight, we can not train them from the stretch.

6.2 Future Work

Below we describe four future extensions for our study.

1. We can create more databases for animals. For example, if we have normal map ground truth for animals, then there are more algorithms we can transfer from human reconstruction to animal reconstruction.

2. There are some losses we defined in SMALF that cannot be used in AnimalRecon because the losses are not differentiable for our rendering method. Mathematically, there might be a way to make it differentiable. This is a possible way we can improve our method.

3. AnimalRecon only works with a fixed camera. We can work on finding a way to make our model work with unknown cameras.

4. Give a raw video, can we learn an implicit neural shape and its skinning weight from stretch?

Bibliography

[1] M. Kocabas, N. Athanasiou, and M. J. Black, "VIBE: video inference for human body pose and shape estimation," *CoRR*, vol. abs/1912.05656, 2019. arXiv: 1912.05656. [Online]. Available: http://arxiv.org/abs/1912.05656.

[2] Y. Jafarian and H. S. Park, "Learning high fidelity depths of dressed humans by watching social media dance videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 753–12 762.

[3] S. Saito *et al.*, "Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization," *arXiv preprint arXiv:1905.05172*, 2019.

[4] S. Saito, T. Simon, J. M. Saragih, and H. Joo, "Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization," *CoRR*, vol. abs/2004.00452, 2020. arXiv: 2004.00452. [Online]. Available: https://arxiv.org/abs/2004.00452.

[5] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "Smpl: A skinned multi-person linear model," *ACM transactions on graphics (TOG)*, vol. 34, no. 6, pp. 1–16, 2015.

[6] S. Zuffi, A. Kanazawa, D. W. Jacobs, and M. J. Black, "3d menagerie: Modeling the 3d shape and pose of animals," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017.

[7] A. Kanazawa, S. Tulsiani, A. A. Efros, and J. Malik, "Learning category-specific mesh reconstruction from image collections," *CoRR*, vol. abs/1803.07549, 2018. arXiv: 1803.07549. [Online]. Available: http://arxiv.org/abs/1803.07549.

[8] S. Goel, A. Kanazawa, and J. Malik, "Shape and viewpoint without keypoints," *CoRR*, vol. abs/2007.10982, 2020. arXiv: 2007.10982. [Online]. Available: https://arxiv.org/abs/2007.10982.

[9] X. Li *et al.*, "Self-supervised single-view 3d reconstruction via semantic consistency," *CoRR*, vol. abs/2003.06473, 2020. arXiv: 2003.06473. [Online]. Available: https://arxiv.org/abs/2003.06473.

[10] S. Tulsiani, N. Kulkarni, and A. Gupta, "Implicit mesh reconstruction from unannotated image collections," *CoRR*, vol. abs/2007.08504, 2020. arXiv: 2007.08504. [Online]. Available: https://arxiv.org/abs/2007.08504.

[11] N. Kulkarni, A. Gupta, D. F. Fouhey, and S. Tulsiani, "Articulation-aware canonical surface mapping," *CoRR*, vol. abs/2004.00614, 2020. arXiv: 2004.00614. [Online]. Available: https://arxiv.org/abs/2004.00614.

[12] X. Li *et al.*, "Online adaptation for consistent mesh reconstruction in the wild," *CoRR*, vol. abs/2012.03196, 2020. arXiv: 2012.03196. [Online]. Available: https://arxiv.org/abs/2012.03196.

[13] T. Hu, L. Wang, X. Xu, S. Liu, and J. Jia, "Self-supervised 3d mesh reconstruction from single images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 6002–6011.

[14] G. Yang, M. Vo, N. Neverova, D. Ramanan, A. Vedaldi, and H. Joo, "Banmo: Building animatable 3d neural models from many casual videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 2863–2873.

[15] B. Biggs, T. Roddick, A. W. Fitzgibbon, and R. Cipolla, "Creatures great and SMAL: recovering the shape and motion of animals from video," *CoRR*, vol. abs/1811.05804, 2018. arXiv: 1811.05804. [Online]. Available: http://arxiv.org/abs/1811.05804.

[16] S. Kearney, W. Li, M. Parsons, K. I. Kim, and D. Cosker, "Rgbd-dog: Predicting canine pose from RGBD sensors," *CoRR*, vol. abs/2004.07788, 2020. arXiv: 2004.07788. [Online]. Available: https://arxiv.org/abs/2004.07788.

[17] F. Bogo, A. Kanazawa, C. Lassner, P. V. Gehler, J. Romero, and M. J. Black, "Keep it SMPL: automatic estimation of 3d human pose and shape from a single image," *CoRR*, vol. abs/1607.08128, 2016. arXiv: 1607.08128. [Online]. Available: http://arxiv.org/abs/1607.08128.

[18] "Towards accurate markerless human shape and pose estimation over time," *CoRR*, vol. abs/1707.07548, 2017, Withdrawn. arXiv: 1707.07548. [Online]. Available: http://arxiv.org/abs/1707.07548.

[19] R. A. Güler and I. Kokkinos, "Holopose: Holistic 3d human reconstruction in-the-wild," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 10 876–10 886. DOI: 10.1109/CVPR.2019.01114.

[20] M. Kocabas, S. Karagoz, and E. Akbas, "Self-supervised learning of 3d human pose using multi-view geometry," *CoRR*, vol. abs/1903.02330, 2019. arXiv: 1903.02330. [Online]. Available: http://arxiv.org/abs/1903.02330.

[21] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis, "Learning to reconstruct 3d human pose and shape via model-fitting in the loop," *CoRR*, vol. abs/1909.12828, 2019. arXiv: 1909.12828. [Online]. Available: http://arxiv.org/abs/1909.12828.

[22] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis, "Learning to estimate 3d human pose and shape from a single color image," *CoRR*, vol. abs/1805.04092, 2018. arXiv: 1805.04092. [Online]. Available: http://arxiv.org/abs/1805.04092.

[23] M. Omran, C. Lassner, G. Pons-Moll, P. V. Gehler, and B. Schiele, "Neural body fitting: Unifying deep learning and model-based human pose and shape estimation," *CoRR*, vol. abs/1808.05942, 2018. arXiv: 1808.05942. [Online]. Available: http://arxiv.org/abs/1808.05942.

[24] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," *CoRR*, vol. abs/1712.06584, 2017. arXiv: 1712.06584. [Online]. Available: http://arxiv.org/abs/1712.06584.

[25] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "AMASS: archive of motion capture as surface shapes," *CoRR*, vol. abs/1904.03278, 2019. arXiv: 1904.03278. [Online]. Available: http://arxiv.org/abs/1904.03278.

[26] S. Zuffi, A. Kanazawa, and M. J. Black, "Lions and tigers and bears: Capturing non-rigid, 3d, articulated shape from images," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3955–3963. DOI: 10.1109/CVPR.2018.00416.

[27] S. Zuffi, A. Kanazawa, T. Y. Berger-Wolf, and M. J. Black, "Three-d safari: Learning to estimate zebra pose, shape, and texture from images "in the wild"," *CoRR*, vol. abs/1908.07201, 2019. arXiv: 1908.07201. [Online]. Available: http://arxiv.org/abs/1908.07201.

[28] B. Biggs, T. Roddick, A. W. Fitzgibbon, and R. Cipolla, "Creatures great and SMAL: recovering the shape and motion of animals from video," *CoRR*, vol. abs/1811.05804, 2018. arXiv: 1811.05804. [Online]. Available: http://arxiv.org/abs/1811.05804.

[29] B. Biggs, O. Boyne, J. Charles, A. W. Fitzgibbon, and R. Cipolla, "Who left the dogs out? 3d animal reconstruction with expectation maximization in the loop," *CoRR*, vol. abs/2007.11110, 2020. arXiv: 2007.11110. [Online]. Available: https://arxiv.org/abs/2007.11110.

[30] N. Lawrence, "Gaussian process latent variable models for visualisation of high dimensional data," *Advances in neural information processing systems*, vol. 16, 2003.

[31] C. Li and G. H. Lee, "Coarse-to-fine animal pose and shape estimation," *CoRR*, vol. abs/2111.08176, 2021. arXiv: 2111.08176. [Online]. Available: https://arxiv.org/abs/2111.08176.

[32] N. Rueegg, S. Zuffi, K. Schindler, and M. J. Black, *Barc: Learning to regress 3d dog shape from images by exploiting breed information*, 2022. DOI: 10.48550/ARXIV.2203.15536. [Online]. Available: https://arxiv.org/abs/2203.15536.

[33] G. Yang *et al.*, "Lasr: Learning articulated shape reconstruction from a monocular video," in *CVPR*, 2021.

[34] G. Yang *et al.*, "Viser: Video-specific surface embeddings for articulated 3d shape reconstruction," in *NeurIPS*, 2021.

[35] C.-H. Yao, W.-C. Hung, Y. Li, M. Rubinstein, M.-H. Yang, and V. Jampani, *Lassie: Learning articulated shapes from sparse image ensemble via 3d part discovery*, 2022. DOI: 10.48550/ARXIV.2207.03434. [Online]. Available: https://arxiv.org/abs/2207.03434.

[36] D. Novotny *et al.*, "Keytr: Keypoint transporter for 3d reconstruction of deformable objects in videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 5595–5604.

[37] L. Yariv, M. Atzmon, and Y. Lipman, "Universal differentiable renderer for implicit neural representations," *CoRR*, vol. abs/2003.09852, 2020. arXiv: 2003.09852. [Online]. Available: https://arxiv.org/abs/2003.09852.

[38] Z. Zheng, T. Yu, Y. Liu, and Q. Dai, "Pamir: Parametric model-conditioned implicit representation for image-based human reconstruction," *CoRR*, vol. abs/2007.03858, 2020. arXiv: 2007.03858. [Online]. Available: https://arxiv.org/abs/2007.03858.

[39] S. Peng *et al.*, "Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans," *CoRR*, vol. abs/2012.15838, 2020. arXiv: 2012.15838. [Online]. Available: https://arxiv.org/abs/2012.15838.

[40] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *CoRR*, vol. abs/2003.08934, 2020. arXiv: 2003.08934. [Online]. Available: https://arxiv.org/abs/2003.08934.

[41] H. Xu, T. Alldieck, and C. Sminchisescu, "H-nerf: Neural radiance fields for rendering and temporal reconstruction of humans in motion," *CoRR*, vol. abs/2110.13746, 2021. arXiv: 2110.13746. [Online]. Available: https://arxiv.org/abs/2110.13746.

[42] T. Alldieck, H. Xu, and C. Sminchisescu, "Imghum: Implicit generative models of 3d human shape and articulated pose," *CoRR*, vol. abs/2108.10842, 2021. arXiv: 2108.10842. [Online]. Available: https://arxiv.org/abs/2108.10842.

[43] L. Liu, M. Habermann, V. Rudnev, K. Sarkar, J. Gu, and C. Theobalt, "Neural actor: Neural free-view synthesis of human actors with pose control," *CoRR*, vol. abs/2106.02019, 2021. arXiv: 2106.02019. [Online]. Available: https://arxiv.org/abs/2106.02019.

[44] L. Yariv, M. Atzmon, and Y. Lipman, "Universal differentiable renderer for implicit neural representations," *CoRR*, vol. abs/2003.09852, 2020. arXiv: 2003.09852. [Online]. Available: https://arxiv.org/abs/2003.09852.

[45] Y. Zheng, V. F. Abrevaya, X. Chen, M. C. Bühler, M. J. Black, and O. Hilliges, "I M avatar: Implicit morphable head avatars from videos," *CoRR*, vol. abs/2112.07471, 2021. arXiv: 2112.07471. [Online]. Available: https://arxiv.org/abs/2112.07471.

[46] L. M. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," *CoRR*, vol. abs/1812.03828, 2018. arXiv: 1812.03828. [Online]. Available: http://arxiv.org/abs/1812.03828.

[47] M. Michalkiewicz, J. K. Pontes, D. Jack, M. Baktashmotlagh, and A. Eriksson, "Implicit surface representations as layers in neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019.

[48] B. Jiang, Y. Hong, H. Bao, and J. Zhang, "Selfrecon: Self reconstruction your digital avatar from monocular video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5605–5615.

[49] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, *Detectron2*, https://github.com/facebookresearch/detectron2, 2019.

[50] M. Contributors, *MMCV: OpenMMLab computer vision foundation*, https://github.com/open-mmlab/mmcv, 2018.

[51] Z. Teed and J. Deng, "RAFT: recurrent all-pairs field transforms for optical flow," *CoRR*, vol. abs/2003.12039, 2020. arXiv: 2003.12039. [Online]. Available: https://arxiv.org/abs/2003.12039.

[52] Y. Yang and D. Ramanan, "Articulated pose estimation with flexible mixtures-of-parts," in *CVPR 2011*, 2011, pp. 1385–1392. DOI: 10.1109/CVPR.2011.5995741.

[53] N. Ravi *et al.*, "Accelerating 3d deep learning with pytorch3d," *CoRR*, vol. abs/2007.08501, 2020. arXiv: 2007.08501. [Online]. Available: https://arxiv.org/abs/2007.08501.

[54] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. arXiv: 1505.04597. [Online]. Available: http://arxiv.org/abs/1505.04597.

[55] S. Liu, T. Li, W. Chen, and H. Li, "Soft rasterizer: A differentiable renderer for image-based 3d reasoning," *CoRR*, vol. abs/1904.01786, 2019. arXiv: 1904.01786. [Online]. Available: http://arxiv.org/abs/1904.01786.

[56] M. M. Loper and M. J. Black, "Opendr: An approximate differentiable renderer," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., Cham: Springer International Publishing, 2014, pp. 154–169, ISBN: 978-3-319-10584-0.

[57] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, *Detectron2*, https://github.com/facebookresearch/detectron2, 2019.

[58] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa, "Laplacian mesh optimization," in *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, 2006, pp. 381–389.

[59] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," *CoRR*, vol. abs/1801.03924, 2018. arXiv: 1801.03924. [Online]. Available: http://arxiv.org/abs/1801.03924.

[60] T. W. Costain and V. A. Prisacariu, "Towards generalising neural implicit representations," *CoRR*, vol. abs/2101.12690, 2021. arXiv: 2101.12690. [Online]. Available: https://arxiv.org/abs/2101.12690.

[61] X. Chen, Y. Zheng, M. J. Black, O. Hilliges, and A. Geiger, "SNARF: differentiable forward skinning for animating non-rigid neural implicit shapes," *CoRR*, vol. abs/2104.03953, 2021. arXiv: 2104.03953. [Online]. Available: https://arxiv.org/abs/2104.03953.

[62] N. Neverova, A. Sanakoyeu, P. Labatut, D. Novotný, and A. Vedaldi, "Discovering relationships between object categories via universal canonical maps," *CoRR*, vol. abs/2106.09758, 2021. arXiv: 2106.09758. [Online]. Available: https://arxiv.org/abs/2106.09758.