UNIVERSITY OF CALIFORNIA, MERCED

Monocular 3D Reconstruction of Articulated Shapes with Weak Supervision

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering and Computer Science

by

Chun-Han Yao

Committee in charge:

Professor Ming-Hsuan Yang, Chair
Professor Shawn Newsam
Professor Sungjin Im
Dr Varun Jampani

2023

The dissertation of Chun-Han Yao is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

(Professor Shawn Newsam)

(Professor Sungjin Im)

(Dr Varun Jampani)

(Professor Ming-Hsuan Yang, Chair)

University of California, Merced

2023

DEDICATION

To my amazing parents and sister.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

VITA

2016	B. S. in Electrical Engineering, National Taiwan University, Taipei, Taiwan
2019	M. S. in Computer Science, University of California, San Diego
2023	Ph. D. in Electrical Engineering and Computer Science, University of California, Merced

PUBLICATIONS

Chun-Han Yao, Amit Raj, Wei-Chih Hung, Yuanzhen Li, Michael Rubinstein, Ming-Hsuan Yang, and Varun Jampani. "ARTIC3D: Learning robust articulated 3D shapes from noisy web image collections." NeurIPS, 2023.

Chun-Han Yao, Wei-Chih Hung, Yuanzhen Li, Michael Rubinstein, Ming-Hsuan Yang, and Varun Jampani. "Hi-LASSIE: High-fidelity articulated shape and skeleton discovery from sparse image ensemble." CVPR, 2023.

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." NeurIPS, 2022.

Chun-Han Yao, Boqing Gong, Hang Qi, Yin Cui, Yukun Zhu, and Ming-Hsuan Yang. "Federated multi-target domain adaptation." WACV, 2022.

Chun-Han Yao, Jimei Yang, Duygu Ceylan, Yi Zhou, Yang Zhou, and Ming-Hsuan Yang. "Learning visibility for robust dense human body estimation." ECCV, 2022.

Chun-Han Yao, Wei-Chih Hung, Varun Jampani, and Ming-Hsuan Yang. "Discovering 3d parts from image collections." ICCV, 2021.

Chun-Han Yao, Chen Fang, Xiaohui Shen, Yangyue Wan, and Ming-Hsuan Yang. "Video object detection via object-level temporal aggregation." ECCV, 2020.

Han-Kai Hsu, **Chun-Han Yao**, Yi-Hsuan Tsai, Wei-Chih Hung, Hung-Yu Tseng, Maneesh Singh, and Ming-Hsuan Yang. "Progressive domain adaptation for object detection." WACV, 2020.

Chun-Han Yao, Chia-Yang Chang, and Shao-Yi Chien. ”Occlusion-aware video temporal consistency.” ACM Multimedia, 2017.

Chun-Han Yao, Chia-Yang Chang, and Shao-Yi Chien. ”Example-based video color transfer.” ICME, 2016.

ABSTRACT OF THE DISSERTATION

Monocular 3D Reconstruction of Articulated Shapes with Weak Supervision

by

Chun-Han Yao
Doctor of Philosophy in Electrical Engineering and Computer Science
University of California Merced, 2023
Professor Ming-Hsuan Yang, Chair

Reconstructing 3D objects from 2D images/videos is a fundamental yet challenging problem in computer vision, which can be applied to a wide range of applications such as AR/VR, gaming, content creation, and autonomous driving. The goal of monocular 3D reconstruction is to estimate the 3D pose and shape of an object in a single-view image or video. Recently, supervised methods have achieved significant progress using various 3D representations like voxel grids, deformable mesh, point clouds, and implicit functions. However, these methods depend heavily on the ground-truth 3D shapes or multi-view images for training, either from synthetic data or human-labeled datasets. Considering the big domain gap between synthetic and natural images as well as the difficulty to annotate large-scale 3D datasets, we aim to develop methods that can utilize weak supervisory signals like the 2D silhouettes, canonical surface mapping, and generic skeleton. Specifically, to produce robust and high-fidelity 3D shapes, we exploit the geometric priors of 3D object parts and dense visibility, semantic consistency between images, as well as generative priors from a Stable Diffusion model. In this thesis, we tackle the problem of monocular 3D reconstruction for three diverse categories: 1) general rigid objects, 2) human bodies, and 3) articulated shapes.

First, we design a reconstruction network to predict the 3D shape of general rigid ob-

jects from single-view images. In order to alleviate the 3D ambiguity of 2D appearance, we propose a part-based representation with multiple meshes and regularize the part shape by geometric primitives. We demonstrate that the network can automatically discover useful 3D parts while learning to reconstruct a whole object. In return, the discovered parts can fit the object shape faithfully and help improve the overall reconstruction accuracy. Moreover, the 3D parts enable interesting applications like shape interpolation and generation since they are consistent across instances of the same category.

Second, we learn dense human body estimation that is robust to partial observations. While prior methods with model-based representations can perform reasonably well on whole-body images, they often fail when parts of the body are occluded or outside the frame. Instead, we adopt a heatmap-based representation and explicitly model the visibility of human joints and vertices. The visibility in x and y axes help distinguishing out-of-frame cases, and the visibility in depth axis corresponds to occlusions (either self-occlusions or occlusions by other objects). We show that visibility can serve as 1) an additional signal to resolve depth ordering ambiguities of self-occluded vertices and 2) a regularization term when fitting a human body model to the predictions.

Finally, we propose a novel and practical problem setting to estimate 3D pose and shape of articulated animal bodies given only a few (10-30) in-the-wild images of a particular animal class. Contrary to existing works that rely on pre-defined template shapes, we do not assume any form of 2D or 3D ground-truth annotations, nor do we leverage any multi-view or temporal information. Our key insight is that 3D parts have much simpler shape compared to the overall animal and that they are robust w.r.t. animal pose articulations. Following these insights, we propose three novel optimization frameworks (LASSIE, Hi-LASSIE, and ARTIC3D) which discover 3D skeleton/parts in a self-supervised manner by combining geometric, semantic, and generative priors.

Chapter 1

Introduction

1.1 Monocular 3D Reconstruction

Recognizing and inferring objects surrounding us is essential for many computer vision systems. While deep learning models have been shown to perform well at recognizing [60, 116, 40] and localizing [31, 106, 77] objects in a 2D image, reasoning 3D attributes of objects from a single image remains a challenging task. 3D reasoning from single-view images is fundamentally ill-posed due to several factors that cause ambiguous object appearance in 2D images, *e.g.*, camera pose, self-occlusions, lighting, and material properties. In this thesis, we study the task of monocular 3D reconstruction, aiming to estimate the 3D pose and shape of an object from a single-view image.

Recently, supervised methods have achieved significant progress in monocular 3D reconstruction using various 3D representations like voxel grids [17, 124, 121, 67], point clouds [25, 45, 63, 92, 21], triangular mesh [129, 32] , and implicit functions [30, 83, 135, 37, 19, 29, 85]. However, most of them require ground-truth 3D shapes, multi-view/temporal information, or large-scale (labeled) images for training. Considering the difficulty to obtain these information as well as the big domain gap between synthetic and natural data, our goal is to develop weakly-supervised or self-supervised methods using

minimal ground-truth annotations.

Although 3D objects in general have complicated articulations and shapes, they can usually be decomposed into parts that are easier to model since they have rather simple geometry and rigid motion. Furthermore, most object instances of a particular category share similar part configurations, *e.g.*, the head, torso, and 4 legs of quadrupedal animals. Based on these insights, we demonstrate in this thesis how to learn high-fidelity 3D articulated shapes by designing part-based representation and utilizing part-level weak supervision.

1.2 Thesis Overview

In Chapter 2, we tackle single-view 3D reconstruction of general rigid objects without ground-truth 3D shapes as supervision. Reasoning 3D shapes from 2D images is an essential yet challenging task, especially when only single-view images are at our disposal. While an object can have a complicated shape, individual parts are usually close to geometric primitives and thus are easier to model. Furthermore, parts provide a mid-level representation that is robust to appearance variations across objects in a particular category. Instead of relying on manually annotated parts for supervision, we propose a self-supervised approach, latent part discovery (LPD). Our key insight is to learn a novel part shape prior that allows each part to fit an object shape faithfully while constrained to have simple geometry. Extensive experiments on the synthetic ShapeNet [10], PartNet [86], and real-world Pascal 3D+ [132] datasets show that our method discovers meaningful object parts and achieves favorable reconstruction accuracy compared to the existing methods with the same level of supervision.

In Chapter 3, we learn to estimate 3D human pose and shape from 2D images by combining a statistical shape model (SMPL [78]) and heatmap-based representation [90]. While prior methods with model-based representations can perform reasonably well on whole-body images, they often fail when parts of the body are occluded or outside the frame. Moreover, these results usually do not faithfully capture the human silhouettes due

to their limited representation power of deformable models (e.g., representing only the naked body). An alternative approach is to estimate dense vertices of a predefined template body in the image space, which is effective in localizing vertices within an image but cannot handle out-of-frame body parts. In this chapter, we learn dense human body estimation that is robust to partial observations. We explicitly model the visibility of human joints and vertices in the x, y, and z axes separately. The visibility in x and y axes help distinguishing out-of-frame cases, and the visibility in depth axis corresponds to occlusions (either self-occlusions or occlusions by other objects). We obtain pseudo ground-truths of visibility labels from dense UV correspondences and train a neural network to predict visibility along with 3D coordinates. We show that visibility can serve as 1) an additional signal to resolve depth ordering ambiguities of self-occluded vertices and 2) a regularization term when fitting a human body model to the predictions. Extensive experiments on multiple 3D human datasets demonstrate that visibility modeling significantly improves the accuracy of human body estimation, especially for partial-body cases.

In Chapter 4, we propose a practical problem setting to estimate 3D pose and shape of animals given only a few (10-30) in-the-wild images of a particular animal species (say, horse). Contrary to existing works that rely on pre-defined template shapes, we do not assume any form of 2D or 3D ground-truth annotations, nor do we leverage any multi-view or temporal information. Moreover, each input image ensemble can contain animal instances with varying poses, backgrounds, illuminations, and textures. Our key insight is that 3D parts have much simpler shape compared to the overall animal and that they are robust w.r.t. animal pose articulations. Following these insights, we propose LASSIE, a novel optimization framework which discovers 3D parts in a self-supervised manner with minimal user intervention. A key driving force behind LASSIE is the enforcing of 2D-3D part consistency using self-supervisory deep features. Experiments on Pascal-Part and self-collected in-the-wild animal datasets demonstrate considerably better 3D reconstructions as well as both 2D and 3D part discovery compared to prior arts.

In Chapter 5, we propose Hi-LASSIE, which tackles a similar problem setting as

LASSIE but makes two significant advances. First, instead of relying on a manually annotated 3D skeleton, we automatically estimate a class-specific skeleton from the selected reference image. Second, we improve the shape reconstructions with novel instance-specific optimization strategies that allow reconstructions to faithful fit on each instance while preserving the class-specific priors learned across all images. Experiments on in-the-wild image ensembles show that Hi-LASSIE obtains higher fidelity state-of-the-art 3D reconstructions despite requiring minimum user input.

In Chapter 6, we introduce ARTIC3D, which is built upon Hi-LASSIE and further guided by 2D diffusion priors from Stable Diffusion. First, we enhance the input images with occlusions/truncation via 2D diffusion to obtain cleaner mask estimates and semantic features. Second, we perform diffusion-guided 3D optimization to estimate shape and texture that are of high-fidelity and faithful to input images. We also propose a novel technique to calculate more stable image-level gradients via diffusion models compared to existing alternatives. Finally, we produce realistic animations by fine-tuning the rendered shape and texture under rigid part transformations. Extensive evaluations on multiple existing datasets as well as newly introduced noisy web image collections with occlusions and truncation demonstrate that ARTIC3D outputs are more robust to noisy images, higher quality in terms of shape and texture details, and more realistic when animated.

In Chapter 7, we conclude the main contributions of this thesis. We also raise several future research directions, including 1) generalizing ARTIC3D to more diverse animals and articulated objects, and 2) improving the quality of 3D articulated shapes and animations by better combining 3D geometric and 2D generative priors.

Chapter 2

Discovering 3D Parts from Image Collections

2.1 Overview

Reasoning 3D shapes from 2D images is an essential yet challenging task, especially when only single-view images are at our disposal. While an object can have a complicated shape, individual parts are usually close to geometric primitives and thus are easier to model. Furthermore, parts provide a mid-level representation that is robust to appearance variations across objects in a particular category. In this chapter, we tackle the problem of 3D part discovery from only 2D image collections. Instead of relying on manually annotated parts for supervision, we propose a self-supervised approach, latent part discovery (LPD). Our key insight is to learn a novel part shape prior that allows each part to fit an object shape faithfully while constrained to have simple geometry. Extensive experiments on the synthetic ShapeNet, PartNet, and real-world Pascal 3D+ datasets show that our method discovers consistent object parts and achieves favorable reconstruction accuracy compared to the existing methods with the same level of supervision.

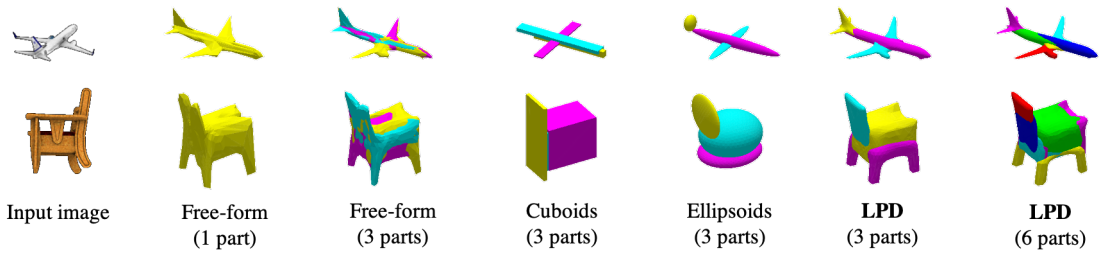

Figure 2.1: **Discovering 3D parts form single-view image collections.** Our method (LPD) enables self-supervised 3D part discovery while learning to reconstruct object shapes from single-view images. Compared to other methods using different part constraints, LPD discovers more faithful and consistent parts, which improve the reconstruction quality and allow part reasoning/manipulation.

2.2 Introduction

Recognizing and reasoning about objects surrounding us is essential for many computer vision systems. While deep learning models have been shown to perform well at recognizing [60, 116, 40] and localizing [31, 106, 77] objects in a 2D image, reasoning 3D attributes of objects from a single image remains a challenging task. Single-view 3D reasoning is fundamentally ill-posed due to several factors that cause ambiguous object appearance in 2D images, *e.g.*, camera pose, self-occlusions, lighting, and material properties. Although objects in general have complicated shapes, they can usually be decomposed into parts that have simpler geometry and are relatively easy to model. Furthermore, most object instances of a particular category share similar part configurations, *e.g.*, the wings, body, and tail of airplanes. In this chapter, we propose to tackle the problem by discovering faithful and consistent 3D parts from 2D image collections. Compared to prior single-view 3D reconstruction approaches that directly predict an object shape, we aim to learn rich and dense part configurations which form an entire object when combined.

Although several recent methods [123, 80, 13, 28, 70, 79, 94, 52] leverage part-based representations for 3D object reasoning, they rely on either 3D object shapes or explicit

part annotations as supervision. Moreover, the learned parts only serve as additional information and are not exploited to improve 3D reconstruction. Considering that collecting ground-truth 3D shapes and the corresponding part labels is labor intensive, we follow a practical scenario where only single-view images, 2D object silhouettes, and camera viewpoints are available for model training. In contrast to existing techniques, our method automatically discovers 3D parts from image collections in a self-supervised manner.

A common practice to represent 3D parts is to use geometric primitives such as ellipsoids or cuboids [123]. They provide a strong regularization of part shapes but are usually too coarse to faithfully represent object parts. As an alternative, several approaches adopt meshes [51, 129, 32, 50, 76, 42, 33, 69] or point-cloud [25, 45, 63, 92, 21] representations. Although these representations are more expressive and can faithfully describe part shapes, they lack part shape regularization that is particularly needed in a weakly-supervised or unsupervised setting. The key insight of this chapter is to represent 3D parts with deep latent embeddings. Specifically, we propose to learn a prior distribution of part shapes with a variational auto-encoder (VAE) [54] that encodes part shapes as latent embeddings. We call this network *Part-VAE* and pre-train it with a set of geometric primitives like cones, cylinders, cuboids, and ellipsoids. We then learn a reconstruction network that takes an input image and predicts part embeddings to obtain a 3D mesh by passing through the decoder of Part-VAE. To further improve the quality of part discovery and reconstruction, we propose a novel part adversarial loss which involves re-assembling parts from different objects in the same category. We name the proposed method *latent part discovery (LPD)*. Figure 2.1 shows several reconstruction results with and without part prior, which demonstrate that LPD can discover consistent parts and produce faithful reconstruction to the input image.

We evaluate LPD on the synthetic ShapeNet [10], PartNet [86] and the real-world Pascal 3D+ [132] datasets. Both quantitative and qualitative results demonstrate that our approach achieves favorable performance against the state-of-the-art methods using the same level of supervision. In addition to part discovery, our part representation enables

object manipulation like selective part swapping, interpolation, and random shape generation from the latent space. In this work, we make the following contributions:

- We propose a part-based single-view 3D reasoning network which can automatically discover object parts. To the best of our knowledge, this is the first work that discovers 3D parts in a self-supervised manner without using any 3D shape or multi-view supervision.
- We develop Part-VAE to learn a latent prior over part shapes. We show that training with geometric primitives can learn useful part embeddings, allowing each part to faithfully represent object shape while constrained to have simple geometry.
- We conduct extensive experiments on both synthetic and natural images. Qualitatively, our method produces more faithful and consistent object parts compared to other part-based methods. Quantitatively, the discovered parts improve whole-object reconstruction and achieve favorable accuracy against the state-of-the-art techniques. In addition, our Part-VAE allows us to manipulate object parts for various applications.

2.3 Related Work

3D Reconstruction. While 3D representation has been widely studied for decades, the best and unified way to represent general objects remains unclear. Voxel grids [17, 124, 121, 67], point clouds [25, 45, 63, 92, 21], and meshes [51, 129, 32, 50, 76, 11, 42, 33, 69] are commonly used to represent object shapes. Several recent methods [30, 83, 135, 37, 19, 29] explore the possibilities to represent 3D shapes in a functional space. While fine-grained voxels, point clouds, and local functions can represent complicated shapes, the flexibility of representation demands strong 3D or multi-view supervision for training. Meshes, on the other hand, are constrained to form a water-tight surface and are convenient to render 2D images. By deforming from a simple template mesh like sphere or cuboid, it is easier to apply shape regularization and thus can be applied to reconstruction scenarios with weaker supervision. One can learn single-view mesh reconstruction from multi-view

or single-view images using naive 2D projection or differentiable rendering [51, 76, 11]. For instance, Henderson *et al.* [42] generate background images and object rendering to learn textured mesh reconstructions from natural images. Kato *et al.* [50] propose view prior learning (VPL) to improve the reconstructed shapes from unseen views. Although they are effective for compact and deformable objects, a single mesh cannot represent complicated shapes with holes or disconnected parts. In this chapter, we exploit multi-mesh part representation which allows disconnected parts in an object while each part can be well-regularized.

Part Discovery. Parts provide a mid-level representation that is robust to appearance variations across objects in the same category. Hung *et al.* [44] learn 2D co-part segmentation on image collections with self-supervision. Lathuilière *et al.* [65] exploit motion cues in videos for part discovery. In the 3D domain, Tulsiani *et al.* [123] use volumetric cuboids as part abstractions to learn 3D reconstruction. Li *et al.* [70] assume known part shapes and learn to assemble them given an input image. Using a point-cloud representation, Mandikal *et al.* [80] predict part-segmented 3D reconstructions from a single image and Luo *et al.* [79] learn to form object parts by clustering 3D points. Paschalidou *et al.* [94] propose hierarchical part decomposition (HPD) by constraining 3D points with super-quadratic functions [96]. These methods require 3D ground-truth shape of a whole object or its parts as supervision. Furthermore, the part shapes in [123, 94] are limited by the expressiveness of their representations. Li *et al.* [69] leverage 2D semantic parts to improve single-view 3D reconstruction, which, however, does not produce individual part shapes. To the best of our knowledge, we propose the first 3D part reconstruction method without any part annotations or ground-truth 3D shapes for training, and our latent part representation enables each part to fit a given object shape faithfully.

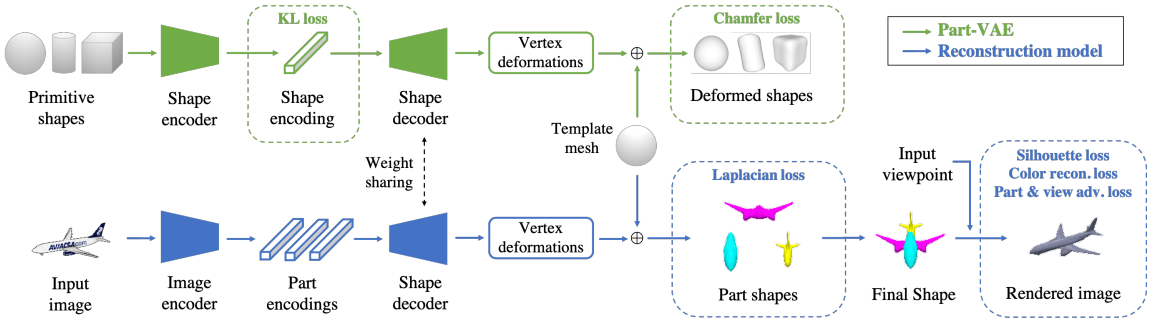

Figure 2.2: **Approach overview.** (top) Our Part-VAE is trained with geometric primitives. (bottom) Our reconstruction model shares the shape decoder with Part-VAE and predicts object parts. We then composite the reconstructed parts to form a 3D object.

2.4 Approach

Reasoning 3D objects from single-view images is inherently ill-posed since the reconstructed object can over-fit to a given view and be highly deformed in the unseen parts. To address this, we propose LPD to represent an object with multiple latent parts. Our intuition is that a complicated object shape can be expressed by assembling simple and regularized parts. Consistent with some recent approaches [50, 42], we propose a method under a weakly-supervised setting where only a single-view image, 2D object silhouette, and its camera viewpoint are available for each object. That is, we do not assume any 3D shape or multi-view images to supervise part discovery or reconstruction. In order to automatically discover underlying parts while reconstructing an object, we propose to learn part embeddings with a variational auto-encoder [54] (VAE) named *Part-VAE*. We train a reconstruction model that predicts part embeddings which are then decoded into part meshes to compose the whole object. Figure 2.2 illustrates the proposed method with two main modules: Part-VAE and reconstruction network.

2.4.1 Learning Part Prior with Part-VAE

We propose Part-VAE to learn a latent shape prior for object parts. The proposed method constrains parts with primitive shapes while allowing the flexibility to fit real-world object parts. In addition, it enables smooth part-interpolation and novel shape generation by random sampling in the latent space. Figure 2.2 (top) illustrates the training process of the Part-VAE with geometric primitives. We first collect a set of primitive shapes such as ellipsoids, cylinders, cones, and cuboids, which are centered at origin but with random scaling and rotation. The Part-VAE network consists of a shape encoder and a shape decoder. The encoder transforms each given primitive shape to a low-dimensional shape encoding, and the decoder reconstructs the input shape by predicting the vertex deformations of a spherical template mesh. To supervise the Part-VAE pre-training, we calculate the Chamfer distance between the input and output vertices as the loss function since the point sets are unordered and not densely corresponding. Given the vertices of an input shape Q and its reconstruction P, the Chamfer loss $\mathcal{L}_c$ can be expressed as:

$$\mathcal{L}_c(P,Q) = \frac{1}{\|P\|}\sum_{p\in P}\min_{q\in Q}\left\|p-q\right\|_2^2 + \frac{1}{\|Q\|}\sum_{q\in Q}\min_{p\in P}\left\|p-q\right\|_2^2. \tag{2.1}$$

To encourage the continuity of latent shape distribution, we adopt a standard KL divergence loss $\mathcal{L}_{kl}$ calculated between the shape embeddings and a standard normal distribution $\mathcal{N} \sim (0,1)$. The overall training loss of Part-VAE is: $\mathcal{L}_{vae} = \mathcal{L}_c + \lambda_{kl}\,\mathcal{L}_{kl}$, where λ_{kl} is a weight parameter.

2.4.2 Part Discovery by Learning to Reconstruct

Figure 2.2 (bottom) illustrates our reconstruction model. Instead of directly predicting an object mesh, we learn an image encoder that takes an input image and predicts the 3D part centroid, latent shape encodings, and surface texture for each part. The part encodings are then passed through the shape decoder of Part-VAE to generate part meshes. To compose an entire object, we simply shift the mesh vertices using the predicted part centroids

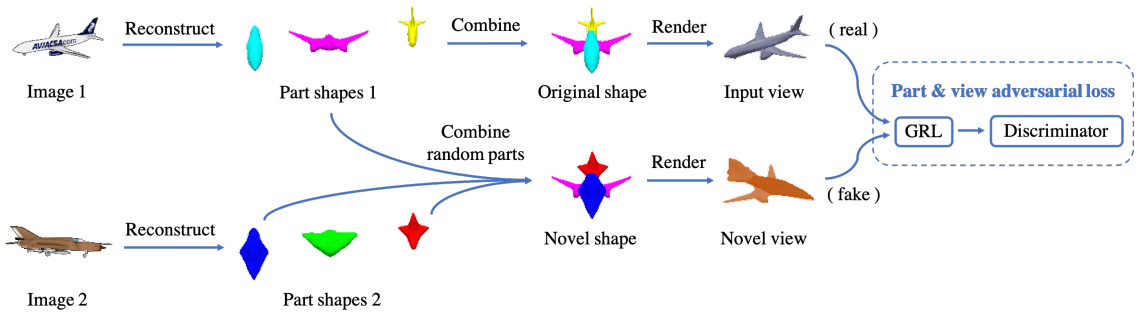

Figure 2.3: **Part and view adversarial learning.** Given two images with different objects, we randomly combine their reconstructed parts into a novel shape. The novel shape is then rendered from a novel viewpoint, which we treat as a 'fake' sample. We train a discriminator to distinguish the fake and real rendered images. By using a gradient reversal layer (GRL), the reconstruction model learns to produce parts that can compose realistic novel shapes.

and concatenate the vertices and surfaces of each part. In the remainder of the chapter, we denote the reconstruction model as $R(\cdot)$, which takes an image as input and outputs a part-composited mesh. The rendering function from viewpoint v is denoted as $G(\cdot, v)$, which produces a rendered image of the input mesh. Note that each training image I includes a silhouette channel I_s and RGB color channels I_c. Likewise, the rendering function G can be separated into G_s and G_c for silhouette projection and color rendering, respectively.

Shape Reconstruction Losses. To supervise shape reconstruction, we enforce the 2D projection of a reconstructed shape to be close to the ground-truth silhouette. In particular, we render the predicted meshes with input viewpoints using a differentiable renderer [76], and calculate the intersection-over-union (IoU) ratio between the rendered and ground-truth silhouettes. Then the silhouette loss $\mathcal{L}_{sil}$ is computed as:

$$\mathcal{L}_{sil}(I, v) = \frac{\|I_s \odot G_s(R(I), v)\|_1}{\|I_s + G_s(R(I), v)\|_1 - \|I_s \odot G_s(R(I), v)\|_1}, \tag{2.2}$$

where $\odot$ denotes element-wise multiplication. We further apply Laplacian regularization $\mathcal{L}_{lap}$ on the reconstructed mesh vertices:

$$\mathcal{L}_{lap}(P) = \frac{1}{\|P\|} \sum_{p \in P} \Big\| p - \frac{1}{\|\mathcal{N}(p)\|} \sum_{q \in \mathcal{N}(p)} q \Big\|_2^2, \tag{2.3}$$

where $\mathcal{N}(p)$ denotes the neighboring vertices of vertex p. It aims to smooth the mesh surfaces by pulling each vertex towards the center of its neighboring pixels. Note that this regularization is applied on each part mesh individually so discontinuity between part surfaces is allowed.

Color Reconstruction Loss. We further exploit the color information in input images by generating textured reconstructions. Given the part encodings, our model predicts texture flow to map the input image to a UV texture image. We then color the mesh surfaces by sampling from the texture image using a pre-defined UV mapping function. The texture flow is predicted per object part so each part has more coherent texture. We denote the overall color rendering process as G_c and the color reconstruction loss $\mathcal{L}_{cr}$ is defined on the semantic features of input and rendered images:

$$\mathcal{L}_{cr}(I, v) = \|F(I_c) - F(G_c(R(I), v))\|_2^2, \tag{2.4}$$

where F is the feature extractor of a fixed classification network. We use AlexNet [60] pre-trained on the ImageNet dataset [60] and extract the output of multiple convolutional layers as F. It encourages the rendered image to be perceptually similar to the input at different levels.

2.4.3 Part and View Adversarial Learning

Unlike the multi-view or 3D-supervised settings, single-view training requires stronger regularization to produce realistic 3D shapes and to discover meaningful parts. Based on the intuition that object parts are interchangeable and they should look realistic from various viewpoints, we extend the view adversarial learning in VPL [50] with part adversarial learning. As illustrated in Figure 2.3, we assemble a novel shape by randomly combining parts from different objects of the same class in a training batch, then render the novel

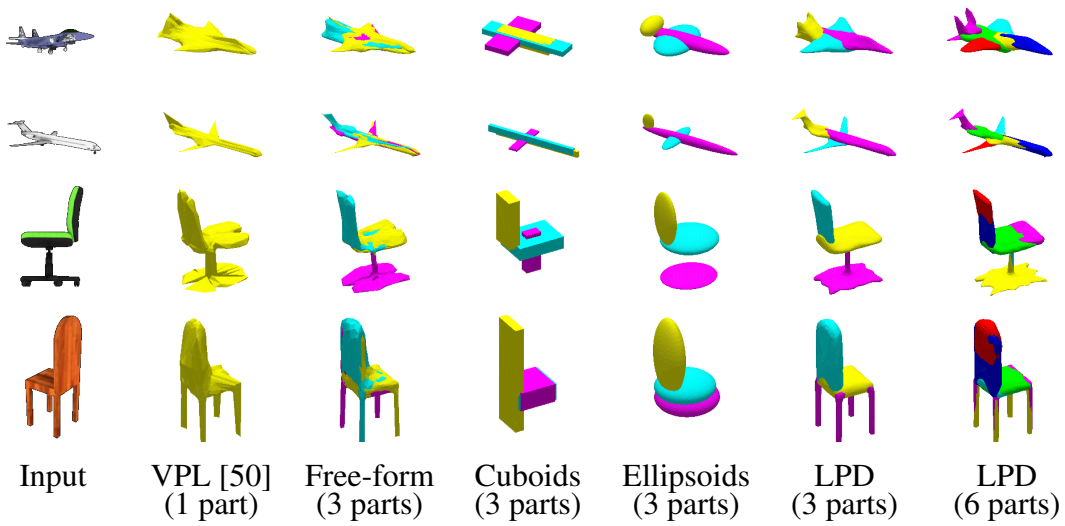

Figure 2.4: **Qualitative results on the ShapeNet dataset [10].** LPD models (ours) adopt the proposed Part-VAE and adversarial learning. The 3-part free-form model reconstructs a whole object with three fully-deformable meshes without any part prior. Compared to the baselines, our approach can produce more faithful and consistent parts from diverse objects.

shape from a novel viewpoint. To make the novel shape realistic, we treat the rendered images of novel shapes as fake examples and those of original shapes as real ones. A discriminator is then trained to classify each rendered image as real or fake. We train the discriminator with a binary cross-entropy between the positive and negative samples:

$$\mathcal{L}_{adv}(I, I', v) = -\log(D(G(R(I), v))) - \log(1 - D(G(R'(I, I'), v'))), \tag{2.5}$$

where I' is a random image different from the input I, $R'(\cdot, \cdot)$ is the reconstruction model with random part selection from two input images, v' is a random novel view, and D is the discriminator. To apply adversarial training, we add a gradient reversal layer (GRL) [27] before the discriminator. As a result, the reconstruction model is trained to fool the discriminator by generating novel objects with realistic shapes. Considering that different object classes may have different view and shape prior, we condition the discriminator with input class labels during training. Note that the class labels are not required during

Table 2.1: **Ablative evaluations on the ShapeNet dataset [10].** The base model reconstructs an object shape with 3 meshes, each is fully-deformable as in SoftRas [76] (PP: part prior, VA: view adversarial learning, PA: part adversarial learning, CR: color reconstruction).

PP	VA	PA	CR	Airplane	Bench	Dresser	Car	Chair	Display	Lamp	Speaker	Rifle	Sofa	Table	Phone	Vessel	All
	✓	✓	✓	56.7	34.4	56.0	68.3	43.1	34.8	47.2	59.9	50.6	48.5	41.1	42.7	53.4	49.0
✓		✓	✓	57.2	36.2	60.7	72.2	44.1	39.8	48.2	63.6	51.9	49.6	43.3	51.5	55.1	51.8
✓	✓		✓	57.1	35.8	**61.4**	73.7	45.1	39.4	48.5	63.7	52.7	49.3	43.9	52.5	54.9	52.2
✓	✓	✓		57.1	36.0	61.0	74.1	45.2	39.7	48.5	**63.8**	53.0	49.7	43.9	52.2	55.1	52.3
✓	✓	✓	✓	**57.3**	**37.3**	60.9	**75.2**	**45.5**	**40.8**	**49.6**	63.3	**54.5**	**50.1**	**44.3**	**52.7**	**56.2**	**52.9**

inference. That is, for given data, we train a single part-discovery/reconstruction model that operates across different object categories. This part-based adversarial learning approach is proposed as a semantic constraint to make the global part arrangements more feasible and realistic.

2.4.4 Model Training and Inference

We first pre-train the Part-VAE with primitive shapes to minimize the loss $\mathcal{L}_{vae}$. Next, the Part-VAE and reconstruction network are jointly trained using image collections together with primitive shapes. The overall objective function of reconstruction network is given by:

$$\mathcal{L}_{sil} + \lambda_{lap}\,\mathcal{L}_{lap} + \lambda_{cr}\,\mathcal{L}_{cr} - \lambda_{adv}\,\mathcal{L}_{adv}, \tag{2.6}$$

where (λ_{lap}, λ_{cr}, λ_{adv}) are weight parameters. The discriminator is trained to minimize $\mathcal{L}_{adv}$, whose gradients are reversed and back-propagated to the reconstruction network to perform adversarial learning. Note that we still fine-tune the Part-VAE with primitive shapes in this stage so that the shape decoder is regularized while adapting to various part shapes in the training images. The Part-VAE, reconstruction network, and discriminator are parametrized as deep neural networks, and the weights are optimized by mini-batch gradient descent. In the model inference phase, we discard the Part-VAE encoder and discriminator. An input image is simply passed through the image encoder and Part-VAE

decoder to reconstruct the 3D parts. We represent each object part by a deformable mesh with $N_v = 642$ vertices and $N_f = 1280$ faces. The size of texture image is 64×64. We implement the proposed method in PyTorch [97] framework and use Adam optimizer [53] for training. The hyper-parameters are tuned on a validation set.

2.5 Experiments

Metrics and Baselines. Evaluating self-supervised part discovery can be ambiguous and subjective as the discovered parts need not correspond to human-annotated ones. Due to the lack of standard metrics or benchmark for 3D part discovery, we qualitatively compare the discovered parts with other methods. As a reference, we also quantitatively evaluate the reconstruction accuracy at both object level and part level. We convert each predicted mesh into a volume of 32^3 voxels and calculate the intersection-over-union (IoU) ratio between the voxelized object and the ground-truth voxels. We report results of our model with $k = 3$ parts and latent part embedding dimension of $d = 64$ if not specified otherwise. Since our work is the first to discover 3D parts using single-view supervision, we mainly compare LPD against three part-based baselines: cuboids, ellipsoids, and free-form meshes. We implement the cuboid and ellipsoid models by reconstructing each object part with a scalable cuboid/ellipsoid. The free-form model adopts fully-deformable meshes without any part shape prior. For object-level reconstruction, we evaluate our method against SoftRas [76] and VPL [50] as they adopt a similar training setting with single-view images and known viewpoints. While there are several other methods for single-view 3D reconstruction, we omit the comparison with them since whole-object reconstruction is not the major focus of this thesis. We experiment on the synthetic ShapeNet [10], PartNet [86], and real-world Pascal 3D+ [132] datasets.

Figure 2.5: **Generalization across classes.** We show sample inputs (top) and LPD results (bottom) of different ShapetNet classes.

2.5.1 Results on ShapeNet

We first conduct experiments with the synthetic dataset provided by Kar *et al.* [49], which contains 43,784 objects in 13 classes from ShapeNet [10]. Each sample includes a 3D CAD model, 20 camera viewpoints for rendering, and the corresponding rendered images at a resolution of 224×224 pixels. We use the same training/validation/testing splits as the original dataset. The training images are augmented by random shuffling the RGB channels and horizontal flipping. We only use one view per object for training and evaluate on all the 20 views in the test set (with independent single-view reconstruction on each view). The ground-truth 3D shapes are only used for testing. We show some qualitative results of our method and other baselines in Figure 2.4. More part reconstruction results on car, lamp, rifle, table, and watercraft samples are shown in Figure 2.5 to demonstrate that our method generalizes well across diverse object classes.

Ablations on the Proposed Models. We perform ablation studies on the proposed method by removing each component at a time. As shown in Table 2.1, removing part prior learning causes a significant drop (3.9%) on the overall reconstruction accuracy. It shows that the part prior provided by Part-VAE effectively improves the generalization of the dis-

Table 2.2: **Voxel IoU results on the ShapeNet dataset [10]**. We compare our method with the state-of-the-art single-view supervised and 3D supervised approaches.

Method	Supervision	Airplane	Car	Chair	All
SIF [30]	3D shapes	53.0	65.7	38.9	49.9
OccNet [83]	3D shapes	57.1	73.7	50.1	57.1
CvxNet [19]	3D shapes	59.8	67.5	49.1	56.7
HPD [94]	3D shapes	52.9	70.2	52.6	58.0
SoftRas [76]	Single-view	52.2	65.7	40.4	46.9
VPL [50]	Single-view	53.1	70.1	45.4	51.3
LPD (ours)	Single-view	**57.3**	**75.2**	**45.5**	**52.9**

covered parts. Without the use of adversarial learning and color reconstruction, we also observe lower voxel IoU by a clear margin (0.6-1.1%).

Comparisons with the State-of-the-arts. Table 2.2 shows performance comparisons against the state-of-the-art methods. With the single-view training setting, our method performs favorably against the existing approaches. When compared to the 3D-supervised methods, our model achieves competitive results on many object classes even though we use weaker single-view supervision. Among the evaluated methods, SIF [30] and CvxNet [19] can subdivide an object into fine-grained regions, and HPD [94] performs hierarchical part reasoning. However, their shape representations require stronger supervision from 3D ground truths to produce faithful part shapes. The qualitative results in Figure 2.4 demonstrate that our model discovers more faithful and consistent parts compared to the baseline methods.

Ablations on Part Representations. We further compare 3D reconstruction methods that use different part shape constraints. A majority of existing methods represent objects with a single mesh [51, 129, 32, 50, 76, 42] and allow each shape to be fully deformable by predicting the vertex deformations directly. On the other end of the spectrum, most part-reasoning approaches represent parts with primitive shapes like cuboids [123] or super-

Table 2.3: **Quantitative evaluations on the ShapeNet dataset [10] using different metrics.** LPD allows part reasoning and achieves higher accuracy in terms of all metrics.

Method	Part	2D IoU ↑	SSIM ↑	CD ↓	Voxel IoU ↑
SoftRas [76]		80.5	86.7	4.76	46.9
VPL [50]		81.0	88.5	2.65	51.3
Free-form	✓	81.1	87.9	3.83	48.1
Cuboids	✓	72.5	67.3	6.12	39.7
LPD (ours)	✓	**83.6**	**91.0**	**2.37**	**52.9**

quadratic surfaces [94]. Our method lies within these two extremes and enables variable degree of freedom by adjusting the latent dimension of the part shape embedding. Table 2.3 shows the quantitative comparisons between part representations like free-form meshes, cuboid reconstructions, and our Part-VAE embedding via different evaluation metrics. In addition to 3D voxel IoU, we calculate the 2D re-projection IoU, 2D structural similarity (SSIM), and Chamfer distance (CD) between the point sets sampled from 3D volumes. The results show that LPD achieves a better trade-off between the degree of deformation and shape regularization among the part-based methods. To observe how our method adapts to different object classes, we perform single-class training with different number of parts k on airplane, car, and chair images. As shown in Table 2.4, the optimal number of parts k varies across object classes. This suggests that each class have a distinct underlying part configuration to optimally represent the object shapes. Note that LPD achieves higher accuracy than other methods on all three classes with more than one part. Our main reconstruction model is class-agnostic and we use the same number of parts for all classes, and yet the performance could be further improved if it is optimized for each class separately.

Table 2.4: **Voxel IoU results with different number of parts k.** Note that the models are trained and tested on a single class.

Method	k	Airplane	Car	Chair
SoftRas [76]	1	54.1	69.5	43.1
VPL [50]	1	54.6	74.1	45.3
LPD (ours)	1	54.5	74.3	45.0
LPD (ours)	2	55.4	**76.1**	45.9
LPD (ours)	3	55.6	75.5	**46.6**
LPD (ours)	6	**55.9**	75.2	46.4

2.5.2 Results on PartNet

To evaluate the quality of the discovered parts, we compare our results with the labeled parts in PartNet dataset. The dataset contains hierarchical part annotations of several ShapeNet models. We collect 111 chair samples that are in both the ShapeNet and PartNet testing sets, then combine the annotated part models into chair back, seat, and base at the coarsest level. Note that we do not use any 3D part supervision for training, so the PartNet annotations are not ground-truths but a reference. Since the discovered parts are not semantically labeled, we manually associate our parts to the closest corresponding PartNet annotations. We report the quantitative voxel IoU in Table 2.5 and the qualitative results in Figure 2.6. Compared to the baseline without part prior and other representations, our method discovers more faithful parts and achieves considerably higher IoU with respect to PartNet annotations.

2.5.3 Part Interpolation and Generation.

In addition to shape reconstruction, we demonstrate two applications of Part-VAE: part-interpolation and random shape generation. Since the object parts discovered by our models are consistent across instances, one can swap or interpolate parts to create new 3D

Table 2.5: **Voxel IoU results on the PartNet [86] chair samples.**

Method	Back	Seat	Base	Avg
Free-form	16.8	19.5	10.3	15.5
Cuboids	22.3	23.4	10.7	18.8
LPD (ours)	**30.4**	**46.0**	**16.2**	**30.9**

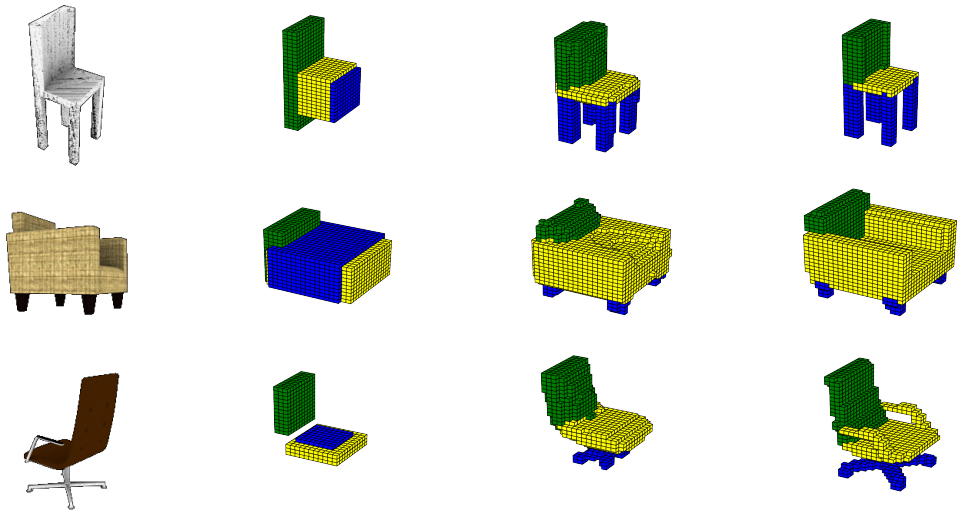

Figure 2.6: **Qualitative results on the PartNet dataset [86].** We show the voxelized 3-part results of our method and a cuboid baseline. Each part is specified with a color: chair back→green, seat→yellow, base→blue. Our method discovers faithful and consistent parts from diverse objects that are relatively closer to the pseudo-GT part annotations in PartNet.

objects. In Figure 2.7, we linearly interpolate the latent encodings (u_1, u_2) of two objects from different categories as: $u = \lambda u_1 + (1 - \lambda)u_2$. Compared to the baseline without part prior, our method deforms each object part smoothly and results in more realistic shapes. The Part-VAE can also be used as a generative model to create novel shapes. Specifically, we fit a Gaussian Mixture Model (GMM) with k components on the latent shape vectors

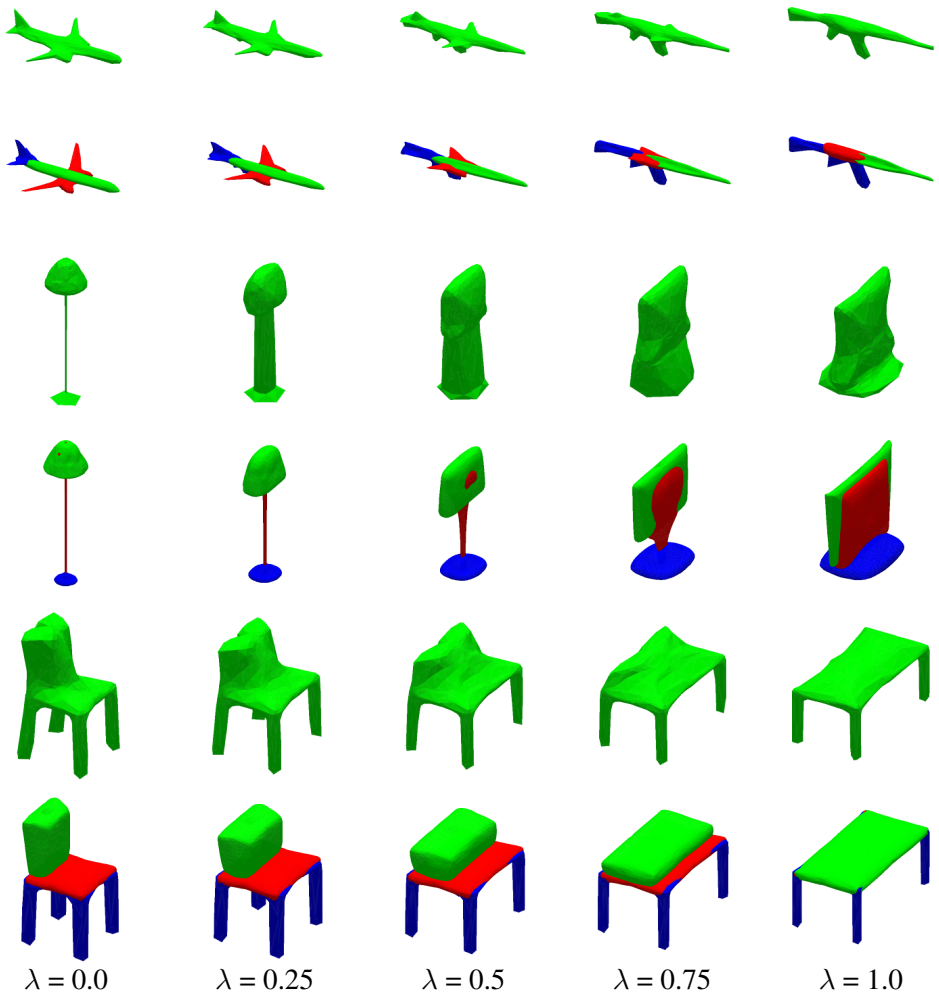

Figure 2.7: **Cross-category interpolation.** We perform interpolation on ShapeNet airplane-rifle, lamp-display, and chair-table. We show the VPL [50] results (mesh interpolation) in rows 1, 3, 5 and LPD results (latent interpolation) in rows 2, 4, 6.

of class-specific images. By sampling random vectors from individual GMM distribution, we can generate k random parts and combine them into a new 3D shape. In Figure 2.8, we show some randomly generated shapes of chairs and airplanes using the Part-VAE trained on the ShapeNet dataset.

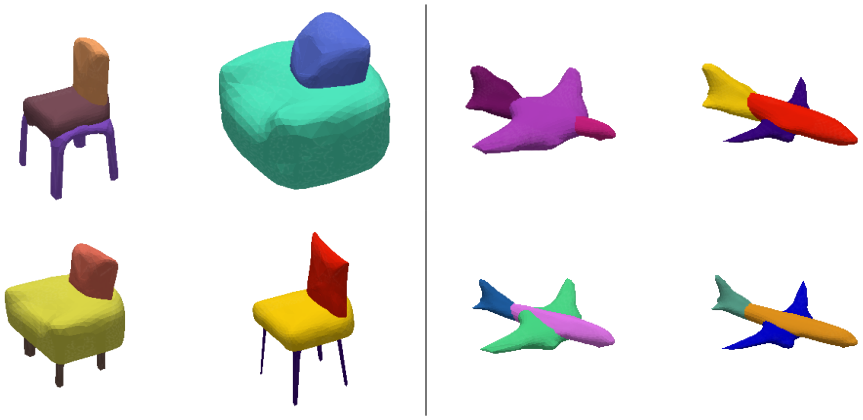

Figure 2.8: **Random shape generation of chairs and airplanes.** We fit a GMM model on the latent shape vectors and generate random parts by sampling from individual GMM components.

2.5.4 Results on Pascal 3D+

We also evaluate the proposed method on the real-world images from the Pascal 3D+ dataset [132] processed by Tulsiani *et al.* [124]. It consists of images in Pascal VOC [24], annotations of 3D models, silhouettes, and viewpoints in Pascal 3D+ [132], and additional images in ImageNet [111] with silhouettes and viewpoints automatically annotated by [66]. This dataset is more challenging due to the complicated object shapes, image background, occlusions, and noisy silhouette annotations. We train and evaluate our models with image resolution of 224×224. The quantitative and qualitative results are shown in Table 2.6 and Figure 2.9, respectively. Despite the challenges, our method discovers consistent object parts and achieves higher reconstruction accuracy than the state-of-the-art approaches.

2.6 Conclusion

In this chapter, we propose LPD to discover 3D parts from single-view image collections. By learning a part prior with Part-VAE, we demonstrate that each part can be

Table 2.6: **Voxel IoU results on the Pascal 3D+ dataset [132].**

Method	Part	Aeroplane	Car	Chair	Avg
SoftRas [76]		46.4	67.6	29.1	47.7
VPL [50]		47.5	67.9	30.4	48.6
Free-form	✓	47.0	68.5	28.7	48.0
Cuboids	✓	37.1	60.7	18.9	38.9
LPD (ours)	✓	**48.2**	**69.1**	**31.0**	**49.4**

deformed to fit a realistic object shape while constrained to have simple geometry. With the goal to compose an object with simple parts, our reconstruction model automatically learns a latent part configuration. In turn, the discovered parts can alleviate shape ambiguity and improve the quality of full object reconstruction. Extensive experimental results show that LPD can discover faithful parts from diverse object classes, and the parts are consistent across different instances within a same category. Furthermore, we achieve the state-of-the-art reconstruction accuracy in a single-view training setting. Our work opens up the possibilities to learn, infer, and manipulate object parts without the need for any ground-truth part labels or 3D shape supervision.

Figure 2.9: **Part and color reconstruction results on the Pascal 3D+ dataset [132].** Despite that the dataset contains complicated 3D objects in a realistic scene, our method is able to discover consistent parts and effectively reconstruct the objects shapes.

Chapter 3

Learning Visibility for Robust Dense Human Body Estimation

3.1 Overview

Estimating 3D human pose and shape from 2D images is a crucial yet challenging task. While prior methods with model-based representations can perform reasonably well on whole-body images, they often fail when parts of the body are occluded or outside the frame. Moreover, these results usually do not faithfully capture the human silhouettes due to their limited representation power of deformable models (e.g., representing only the naked body). An alternative approach is to estimate dense vertices of a predefined template body in the image space, which is effective in localizing vertices within an image but cannot handle out-of-frame body parts. In this chapter, we learn dense human body estimation that is robust to partial observations. We explicitly model the visibility of human joints and vertices in the x, y, and z axes separately. The visibility in x and y axes help distinguishing out-of-frame cases, and the visibility in depth axis corresponds to occlusions (either self-occlusions or occlusions by other objects). We obtain pseudo ground-truths of visibility labels from dense UV correspondences and train a neural network to predict

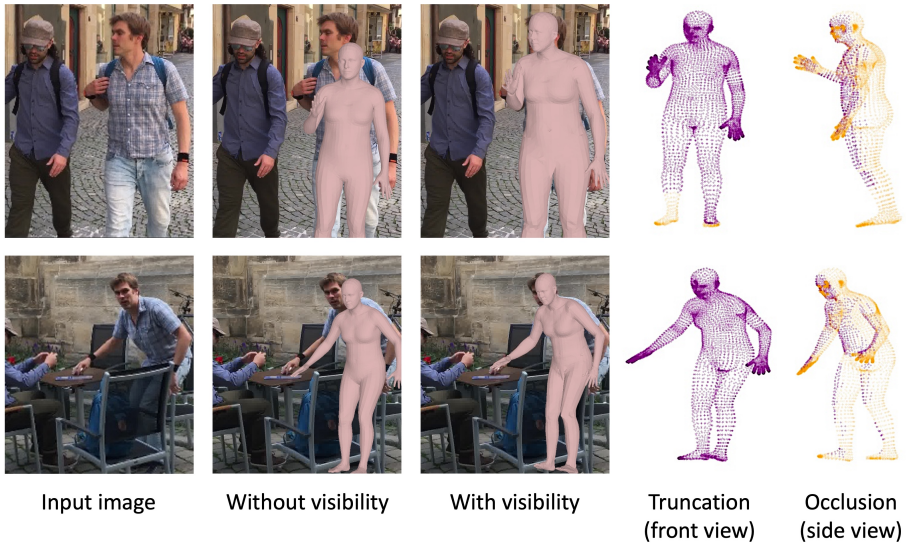

Figure 3.1: **Dense human body estimation with/without visibility modeling.** We propose to learn dense visibility to improve human body estimation in terms of faithfulness to the input image and robustness to truncation (top) or occlusions (bottom). We show the estimated meshes without/with visibility modeling in columns 2-3 and the vertex visibility labels in columns 4-5 (purple:visible, orange:invisible).

visibility along with 3D coordinates. We show that visibility can serve as 1) an additional signal to resolve depth ordering ambiguities of self-occluded vertices and 2) a regularization term when fitting a human body model to the predictions. Extensive experiments on multiple 3D human datasets demonstrate that visibility modeling significantly improves the accuracy of human body estimation, especially for partial-body cases.

3.2 Introduction

Estimating 3D human pose and shape from monocular images is a crucial task for various applications such as performance retargeting, virtual avatars, and human action recognition. It is a fundamentally challenging problem due to the depth ambiguity and the complex nature of human appearances that vary with articulation, clothing, lighting,

viewpoint, and occlusions. To represent the complicated 3D human bodies via compact parameters, model-based methods like SMPL [78] have been widely used in the community. However, SMPL parameters represent human bodies in a holistic manner, causing their limited flexibility to fit real-world images faithfully via direct regression. More importantly, the regression-based methods tend to fail when a human body is not fully visible in the image, *e.g.*, occluded or out of frame [56]. In this work, we aim to learn human body estimation that is faithful to the input images and robust to partial-body cases.

Instead of directly regressing SMPL parameters, we train a neural network to predict the coordinate heatmaps in three dimensions for each human joint and mesh vertex. The dense heatmap-based representation can preserve the spatial relationship in the image domain and model the uncertainty of predictions. It is shown to be effective in localizing visible joints/vertices and flexible to fit an input image faithfully [118, 88, 89, 90]. Nonetheless, the x and y-axis heatmaps are defined in the image coordinates, which cannot represent the out-of-frame (*i.e.*, truncated by image boundaries) body parts. In addition, occlusions by objects or the human body itself could cause ambiguity for depth-axis predictions. Without knowing which joints/vertices are visible, the network tends to produce erroneous outputs on partial-body images. To address this, we propose *Visibility-aware Dense Body (VisDB)*, a heatmap-based dense representation augmented by visibility. Specifically, we train a network to predict binary truncation and occlusion labels along with the heatmaps for each human joint and vertex. With the visibility modeling, the proposed network can learn to make more accurate predictions based on the observable cues. In addition, the vertex-level occlusion predictions can serve as a depth ordering signal to constrain depth predictions. Finally, by using visibility as the confidence of 3D mesh prediction, we demonstrate that VisDB is a powerful intermediate representation which allows us to regress and/or optimize SMPL parameters more effectively. In Figure 3.1, we show examples of truncation and occlusions as well as the dense human body estimations with and without visibility modeling.

Considering that most existing 3D human datasets lack dense visibility annotations,

we obtain pseudo ground-truths from dense UV estimations [35]. Given the estimated UV map of an image, we calculate the pixel-to-vertex correspondence by minimizing the distance of their UV coordinates. Each vertex mapped to a human pixel is considered visible, and vice versa. Note that this covers the cases of truncation, self-occlusions, and occlusions by other objects. We further show that the dense vertex-to-pixel correspondence provides a good supervisory signal to localize vertices in the image space. Since dense UV estimations are based on part-wise segmentation masks which are robust to partial-body images, the dense correspondence loss can mitigate the inaccurate pseudo ground-truth meshes and better align the outputs with human silhouettes. To demonstrate the effectiveness of our method, we conduct extensive experiments on multiple human datasets used by prior arts. Both qualitative and quantitative results on the Human3.6M [46], 3DPW [128], 3DPW-OCC [128, 151], and 3DOH [151] datasets show that learning visibility significantly improves the accuracy of dense human body estimation, especially on images with truncated or occluded human bodies.

The main contributions of our work are:

- We propose VisDB, a heatmap-based human body representation augmented with dense visibility. We train a neural network to predict the 3D coordinates of human joints and vertices as well as their truncation and occlusion labels. We obtain pseudo ground-truths of visibility labels from image-based dense UV estimates, which are also used as additional supervision signal to better align our predictions with the input image.
- We show how the dense visibility predictions can be used for robust human body estimation. First, we exploit occlusion labels to supervise vertex depth predictions. Second, we regress and optimize SMPL parameters to fit VisDB (partial-body) outputs by using visibility as confidence weighting.

3.3 Related Work

Model-based human body estimation. Most existing methods on human body estimation adopt a model-based representation. For instance, SMPL [78] is a widely-used statistical human body model that maps a set of pose $\theta \in \mathbb{R}^{72}$ and shape $\beta \in \mathbb{R}^{10}$ parameters to a 3D human mesh $V \in \mathbb{R}^{6890\times3}$. In SMPL, θ represents the axis-angle 3D rotations of 24 joints, and β is the top-10 PCA coefficients of a statistical human shape space. Early methods iteratively optimize the SMPL parameters to fit the estimated 2D keypoints [5] or silhouettes [64]. Several recent works [48, 100, 93, 99, 57, 59] train a deep neural network to directly regress SMPL parameters from an input image. However, the SMPL representation is not always informative enough for a network to learn as it embeds the articulated body shapes in a low dimensional space. The regression-based methods often fail on truncation and occlusion cases since the networks tend to make holistic predictions based on certain body parts only [56]. Instead, we show that localizing 3D vertices is a more suitable task to learn for such scenarios. The network needs to learn the relationship between the parameters and the shape as well in order to estimate accurate SMPL parameters.

Dense human body representations. To fit the complicated shapes more faithfully, dense human body representations have been proposed, including volumetric space [126], occupancy field [113, 114], dense UV correspondence [1, 147], and 3D mesh [58, 15, 90, 72, 73]. Among these methods, I2L-MeshNet [90] proposes an efficient heatmap representation to estimate human joints and vertices in the image space and root-relative depth axis. It can fit the input images accurately since heatmaps preserve the spatial relationship in image features extracted by a convolutional neural network (CNN). Nonetheless, even when certain body parts are not visible in the image, this model is designed to localize all the joints and vertices within the image frame. We show that it can negatively affect the model performance and emphasize the importance of additional visibility information.

Occlusion-aware methods. Several methods have been proposed to deal with the challenging scenarios where human bodies are partially truncated or occluded. Muller *et*

al. [91] and Hassan *et al.* [38] introduce explicit modeling of human body self-contact and human-scene interactions, respectively. These methods require ground-truth annotations which are hard to obtain. Other methods leverage human-centric heatmaps, part segmentation masks, or dense UV estimations [35], to increase the model robustness on truncated images [108], crowded scenes (occluded by other people) [119] or general occlusions [134, 34, 151, 56]. Although effective in particular scenarios, most of them directly regress SMPL parameters which still suffer from the limited representation strength. To the best our our knowledge, the proposed VisDB representation is the first to explicitly model dense human body visibility (including truncation and all occlusion scenarios), which is trained with pseudo ground-truth visibility labels from dense UV estimates.

3.4 Approach

We illustrate our overall framework in Figure 3.2. In Section 3.4.1, we describe a heatmap-based representation which we build our method upon. Then, we introduce the proposed Visibility-aware Dense Body (VisDB) in Section 3.4.2. Each human joint and mesh vertex is represented by 1) three 1D heatmaps (x, y, z dimensions) which define its 3D coordinate and 2) three binary labels indicating its visibility in three dimensions. We train a network model to predict the dense heatmaps and visibility, which represents a partial body faithful to the input image. The visibility estimations can be interpreted as depth ordering signals or prediction confidence. In Section 3.4.3, we design a visibility-guided depth ordering loss to self-supervise depth estimation. In Section 3.4.4, we show that VisDB outputs can be used to fit SMPL models accurately and efficiently. We train a regression network to estimate SMPL parameters based on the joint and vertex coordinates as well as their visibility labels. During inference, we initialize the SMPL parameters by the regressor and further optimize them to align with the VisDB predictions. Finally, in Section 3.4.5, we exploit dense UV correspondence to obtain robust pseudo labels of visibility and weakly supervise vertex localization in the image space.

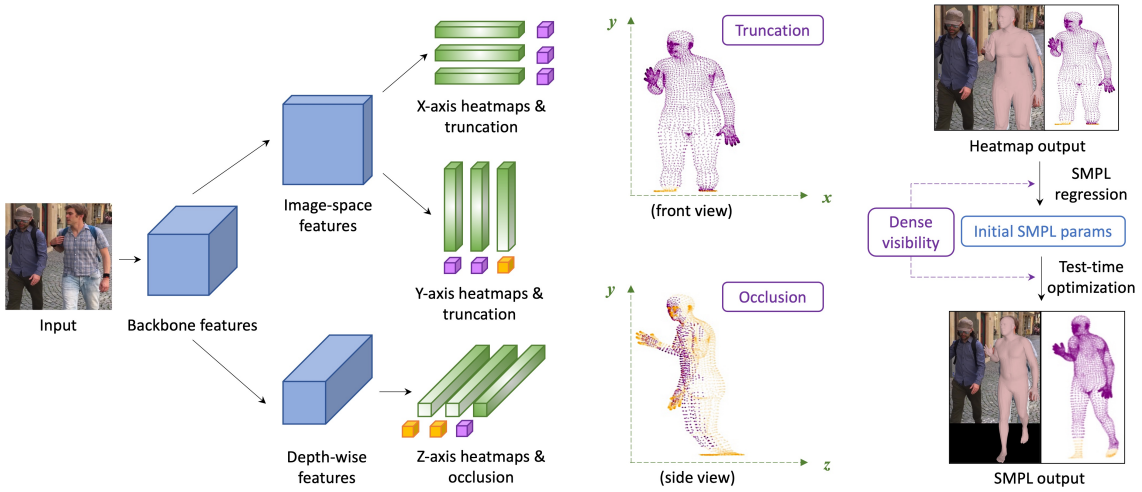

Figure 3.2: **VisDB framework overview** (best viewed in color). Given an input image, the network extracts features in the image and depth coordinates, from where we predict the x, y, z heatmaps for each human joint and vertex. In addition, we predict a binary visibility label (purple:visible, orange:invisible) of each axis, *i.e.*, x-truncation, y-truncation, z-occlusion. To obtain a more regularized and complete human body, we train a regression network to estimate SMPL parameters based on the dense 3D coordinates and visibility. At test time, we can further optimize the regressed SMPL parameters to fit the partial-body predictions from heatmaps.

3.4.1 Preliminaries: Heatmap-based Representation

Given an input image, a prior heatmap-based method [90] estimates three 1D heatmaps $H = \{H^x, H^y, H^z\}$ for each human joint and mesh vertex. The x and y-axis heatmaps H^x, H^y are defined in the image space, and the z-axis heatmaps H^z are defined in the depth space relative to root joint. We denote the joint heatmaps as $H_J \in \mathbb{R}^{N_J \times D \times 3}$ and vertex heatmaps as $H_V \in \mathbb{R}^{N_V \times D \times 3}$, where N_J is the number of joints, N_V is the number of vertices, and D is the heatmap resolution. The heatmaps are predicted based on image

features $F \in \mathbb{R}^{c \times h \times w}$ extracted by a backbone network as follows:

$$\begin{aligned} H^x &= f^{\,\text{1D},x}(\text{avg}^y(f^{\,\text{up}}(F))), \\ H^y &= f^{\,\text{1D},y}(\text{avg}^x(f^{\,\text{up}}(F))), \\ H^z &= f^{\,\text{1D},z}(\psi(\text{avg}^{x,y}(F)))), \end{aligned} \tag{3.1}$$

where $f^{\text{1D},i}$ is 1-by-1 1D convolution for the i-th axis heatmaps, avg^i is i-axis marginalization by averaging, $f^{\,\text{up}}$ denotes up-sampling by deconvolution, and ψ is a 1D convolution layer followed by reshaping operation. Finally, the continuous 3D coordinates of joints $J \in \mathbb{R}^{N_J \times 3}$ and vertices $V \in \mathbb{R}^{N_V \times 3}$ can be obtained by applying soft-argmax on the discrete heatmaps H_J and H_V, respectively.

3.4.2 Visibility-aware Dense Body

Heatmap-based representations are shown effective in estimating human bodies in the image space. However, they often fail when the human bodies are occluded or truncated since the predictions are based on spatial image features and limited by the image boundaries. Without knowing which joints/vertices are invisible, fitting a SMPL model on the entire body tends to generate erroneous outputs. To deal with more practical scenarios where only partial bodies are visible, we make the following adaptations to a heatmap-based representation: 1) To augment the x and y-axis heatmaps, we predict binary truncation labels S^x, S^y, indicating whether a joint or vertex is within the image frame, 2) For the z-axis heatmaps, we predict a binary occlusion label S^z which specifies the depth-wise visibility. The visibility labels are predicted in a similar fashion as the heatmaps in Eq. (3.1):

$$\begin{aligned} S^x &= \sigma(\text{avg}^x(g^{\text{1D},x}(\text{avg}^y(f^{\,\text{up}}(F))))), \\ S^y &= \sigma(\text{avg}^y(g^{\text{1D},y}(\text{avg}^x(f^{\,\text{up}}(F))))), \\ S^z &= \sigma(\text{avg}^z(g^{\text{1D},z}(\psi(\text{avg}^{x,y}(F))))), \end{aligned} \tag{3.2}$$

where g^{1D} is a 1-by-1 1D convolutional layer similar to f^{1D} and σ is a sigmoid operator. We then concatenate the $\{S^x, S^y, S^z\}$ predictions to obtain joint visibility $S_J \in \mathbb{R}^{N_J \times 3}$ and vertex visibility $S_V \in \mathbb{R}^{N_V \times 3}$. By applying the soft-argmax operators to the predicted 1D heatmaps, the final output of our network becomes $\{J, V, S_J, S_V\}$, referred to as Visibility-aware Dense Body (VisDB). With the visibility information, the network model can learn to focus on the visible body parts and push the invisible parts towards the image boundaries. In our experiments (Table 3.3), we demonstrate that visibility modeling significantly reduces the errors of visible vertices. Moreover, the visibility labels can be seen as the confidence of coordinate predictions, which are essential to mesh regularization and completion via SMPL model fitting as described in Section 3.4.4.

We denote the ground-truth VisDB as $\{J^*, V^*, {S_J}^*, {S_V}^*\}$ and train the network by using the following losses. The joint coordinate loss $\mathcal{L}_{joint}$ is defined as:

$$\mathcal{L}_{joint} = \|J - J^*\|_1. \tag{3.3}$$

The vertex coordinate loss $\mathcal{L}_{vert}$ is defined as:

$$\mathcal{L}_{vert} = \|V - V^*\|_1. \tag{3.4}$$

We also regress the joints from vertices using a pre-defined regressor $W \in \mathbb{R}^{N_V \times N_J}$ and calculate a regressed-joint loss $\mathcal{L}_{r-joint}$:

$$\mathcal{L}_{r-joint} = \|WV - J^*\|_1. \tag{3.5}$$

Similar to [90], we apply losses on the mesh surface normal and edge length as shape regularization. The normal loss $\mathcal{L}_{norm}$ and edge loss $\mathcal{L}_{edge}$ are:

$$\mathcal{L}_{norm} = \sum_f \sum_{\{v_i, v_j\} \subset f} \left| \left\langle \frac{v_i - v_j}{\|v_i - v_j\|_2}, {n_f}^* \right\rangle \right|, \tag{3.6}$$

$$\mathcal{L}_{edge} = \sum_f \sum_{\{v_i, v_j\} \subset f} \left| \|v_i - v_j\|_2 - \|{v_i}^* - {v_j}^*\|_2 \right|, \tag{3.7}$$

where f is a mesh surface, n_f is the unit normal vector of f, and v_i, v_j are the coordinates of vertex i and j, respectively. Finally, we define the joint and vertex visibility loss $\mathcal{L}_{vis}$ with binary cross entropy (BCE):

$$\mathcal{L}_{vis} = \text{BCE}(S_J, {S_J}^*) + \text{BCE}(S_V, {S_V}^*). \tag{3.8}$$

The VisDB prediction is illustrated in Figure 3.2 (left).

3.4.3 Resolving Depth Ambiguity via Visibility

Vertex-level visibility can not only be seen as model confidence for SMPL fitting but also provide depth ordering information. Intuitively, visible vertices should have lower depth value compared to the invisible vertices projected to the same pixel. We observe that VisDB network generally predicts accurate 2D coordinates and visibility, but sometimes fails at depth predictions when the human body occludes itself and the pose is less common in the training datasets. To resolve the depth ambiguity in self-occlusion cases, we propose a depth ordering loss $\mathcal{L}_{depth}$ based on vertex visibility as follows:

$$\mathcal{L}_{depth} = \sum_x \sum_y \text{ReLU}\Big(\max_{v \in Q(x,y)} v^z - \min_{\overline{v} \in \overline{Q}(x,y)} \overline{v}^z \Big), \tag{3.9}$$

where $Q(x, y)$ is the set of vertices projected to a discretized image coordinate (x, y) which belong to the front (occluding) part, and $\overline{Q}$ contains the vertices of the back (occluded) part(s). The definition can be written as:

$$\begin{aligned} Q(x,y) &= \Big\{ v \Big| v \mapsto (x,y) \wedge P(v) = p^*(x,y) \Big\} \\ \overline{Q}(x,y) &= \Big\{ v \Big| v \mapsto (x,y) \wedge P(v) \neq p^*(x,y) \Big\}, \end{aligned} \tag{3.10}$$

where $\mapsto$ denotes the discrete projection and $P(v)$ is the part label of vertex v defined in DensePose [35]. We define the front part $p^*(x, y)$ by finding the vertex with highest z-axis visibility score s^z as:

$$p^*(x,y) = P\Big(\arg\max_{v \mapsto (x,y)} {s_v}^z \Big). \tag{3.11}$$

$\mathcal{L}_{depth}$ is designed to push the self-occluded part(s) $\overline{Q}$ to the back and non-occluded part Q to the front, where the occlusion information is given by the z-axis visibility. Note that we compare the maximum depth (back side) of Q and the minimum depth (front side) of $\overline{Q}$, and thus $\mathcal{L}_{depth}$ will be nonzero if the depth ordering disagrees with occlusion prediction and zero if the parts do not overlap anymore. Since this loss depends on accurate visibility estimations, we only apply it during the fine-tuning stage.

3.4.4 SMPL Fitting from Visible Dense Body

From the VisDB predictions, we can obtain the 3D coordinates and visibility of human joints and vertices. While the partial-body outputs are faithful to the input image from the front view, they sometimes look abnormal from a side view or contain rough surfaces. To regularize the body shape and complete the truncated parts, we perform model fitting on the visible dense body predictions. Given the coordinates and visibility of joints and vertices, we train a regression network to estimate SMPL pose $\theta \in \mathbb{R}^{72}$ and shape $\beta \in \mathbb{R}^{10}$ parameters. The regressed parameters are then forwarded to the SMPL model to generate the mesh coordinates denoted as $\text{SMPL}(\theta, \beta) \in \mathbb{R}^{N_V \times 3}$. Unlike prior art [90] which regresses a SMPL model from all the joints regardless of their visibility, our VisDB representation allows us to fit the visible partial body only. The training objectives of the SMPL regressor include SMPL parameter error, vertex error, joint error, and the negative log-likelihood of a pose prior distribution. The SMPL parameter loss $\mathcal{L}_{smpl}$ is defined as:

$$\mathcal{L}_{smpl} = \|\theta - \theta^*\|_1 + \|\beta - \beta^*\|_1, \tag{3.12}$$

where θ^* and β^* are the ground-truth pose and shape parameters. The SMPL vertex loss $\mathcal{L}_{smpl-vert}$ and joint loss $\mathcal{L}_{smpl-joint}$ are defined similarly as in Eq. (3.4) and (3.5) but weighted by visibility S_V, S_J as:

$$\mathcal{L}_{smpl-vert} = S_V \odot \|\text{SMPL}(\theta, \beta) - {V_c}^*\|_1, \tag{3.13}$$

$$\mathcal{L}_{smpl-joint} = S_J \odot \|W\text{SMPL}(\theta, \beta) - {J_c}^*\|_1, \tag{3.14}$$

where $\odot$ denotes element-wise multiplication, and (V_c^*, J_c^*) are the ground-truth root-relative coordinates of vertices and joints in the camera space. Ideally, the VisDB network makes more confident predictions on the clearly visible joints and vertices. Hence, we see the visibility labels as prediction confidence and use them to weight the coordinate losses. In addition, we apply a pose prior loss $\mathcal{L}_{prior}$ using a fitted Gaussian Mixture Model (GMM) provided by [98]:

$$\mathcal{L}_{prior} = -\log\Big(\sum_i G_i(\theta)\Big), \tag{3.15}$$

where G_i is the i-th component of GMM.

We observe that the regressed SMPL meshes roughly capture the human pose and shape but do not always align with the VisDB predictions in details. Therefore, we use the regressed parameters as initialization and propose efficient test-time optimization to further optimize the SMPL parameters against VisDB predictions. For this optimization, we apply similar losses as in Eq. (3.13)-(3.15), except that the ground-truths $\{V_c^*, J_c^*\}$ are replaced by the VisDB predictions converted into root-relative coordinates in the camera space. Since we initialize the SMPL parameters by the regression network and the use strong supervisory signal, *i.e.*, 3D joint and vertex coordinates, the test-time optimization only takes around 100 iterations to converge using an Adam optimizer [53]. We illustrate the process of SMPL regression and optimization in Figure 3.2 (right).

3.4.5 Exploiting Dense UV Correspondence

Most existing 3D human datasets do not provide joint visibility labels, and none annotates vertex visibility. To train our VisDB network, we obtain pseudo ground-truths from the fitted SMPL meshes and dense UV estimations. For x and y-axis truncation, we can simply identify the truncated joints/vertices by projecting the fitted mesh onto the image plane. Occlusion, however, cannot be easily inferred from the input image or fitted mesh alone. One can estimate self-occlusion by rendering a fitted mesh, but this does not capture

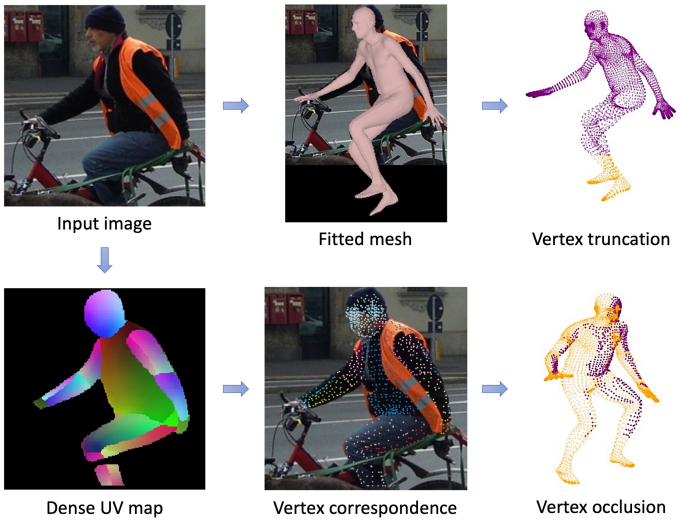

Figure 3.3: **Dense UV correspondence and visibility labels.** Given an input image, we obtain a fitted SMPL mesh and dense UV estimation from off-the-shelf algorithms. To acquire the dense visibility labels for training, we identify the truncated vertices from the fitted mesh. From the dense UV map, we calculate the pixel-to-vertex correspondence to obtain pseudo ground-truths of vertex occlusions as well as image-space coordinates for weak supervision.

occlusions by other objects. More importantly, the fitting algorithm used to get the pseudo ground-truth meshes is not robust to partial-body cases. To address this, we propose to exploit dense UV correspondence between the input image and a SMPL mesh. Dense UV estimation provides the part-based segmentation mask of a human body as well as continuous UV coordinates of each human pixel, which are robust to truncation and occlusions. We calculate the UV coordinate of each pixel by applying an off-the-shelf dense UV estimation method [35]. For each human pixel p, we then find the corresponding mesh vertex v whose UV coordinate is closest to the pixel. The pixel-to-vertex M_P and vertex-to-pixel M_V mappings can be expressed as:

$$\begin{aligned} M_P &= \left\{ p \rightarrow v \middle| v = \text{argmin}_{v'} \left\| \text{UV}(v') - \text{UV}(p) \right\|_2 \forall p \right\} \\ M_V &= \left\{ v \rightarrow \{p'\} \middle| M_P(p') = v \forall v \right\}. \end{aligned} \tag{3.16}$$

A vertex mapped to at least one pixel is labeled as visible or occluded otherwise.

Similar to [34, 134, 147], we also utilize the dense vertex-pixel correspondence as weak supervision for better alignment with the human silhouettes. For each vertex v, we calculate the center of its corresponding pixels $M_V(v)$ and define a UV correspondence loss $\mathcal{L}_{uv}$ as:

$$\mathcal{L}_{uv} = \sum_{v} {s_v}^z \left\| v^{x,y} - \sum_{p \in M_V(v)} \frac{p}{|M_V(v)|} \right\|_1, \tag{3.17}$$

where $v^{x,y}$ is the 2D projection of vertex v and ${s_v}^z$ is the binary occlusion label with ${s_v}^z = 1$ indicating that the vertex v is visible. The UV correspondence loss can not only mitigate the inaccurate pseudo ground-truth meshes, but improve the faithfulness to human silhouettes since it is based on segmentation mask predictions. We empirically discover that this direct vertex-level supervision is more efficient and effective for VisDB training compared to rendering-based losses [134, 23]. The proposed vertex-pixel correspondence and visibility labeling are illustrated in Figure 3.3.

3.4.6 Model Training and Inference

We first train the VisDB network on 3D data with mesh annotations, then fine-tune it on all training data by adding the depth ordering and UV correspondence losses. The regressor network is trained to estimate the SMPL parameters based on the estimated coordinates and visibility of joints and vertices. During inference, we apply optional optimization on the regressed SMPL parameters to best align with the VisDB predicted mesh. For the VisDB network backbone, we use a ResNet50 [40] model pre-trained on the ImageNet dataset [20]. The weights are updated by the Adam optimizer [53] with a mini-batch size of 64. We represent a human body by $N_J = 30$ joints and $N_V = 6890$ vertices, and the heatmap resolution $D = 64$. In addition, we use the ground-truth bounding boxes to crop the human region from an input image and resize it to 256×256. The bounding boxes of testing data are estimated by a pre-trained Mask R-CNN [39] model if not available in the dataset. We apply common data augmentations such as random scaling (±25%), rotation (±45°), horizontal flip, and color jittering (±20%) during training. Considering that trun-

cation and occlusion examples are rare in most 3D human datasets, we include random occlusion masks and bounding box shifting (±25%) as additional augmentations to increase the partial-body/whole-body ratio. Our models are implemented with PyTorch [97] and trained with NVIDIA Tesla V100 GPUs.

3.5 Experiments

3.5.1 Datasets and Metrics

Following most prior arts, we adopt mixed 2D-3D training on the MSCOCO [74], Human3.6M [46], MuCo-3DHP [81], and 3DPW [128] datasets. The pseudo ground-truth meshes of Human3.6M and MSCOCO are obtained by applying SMPLify-X [98] to fit the joint annotations. We evaluate our models on the Human3.6M, 3DPW, 3DPW-OCC [128, 151], and 3DOH [151] testing sets. Note that 3DOH is composed of images with object occlusions and 3DPW-OCC contains a subset of 3DPW sequences where the human bodies are partially occluded. For quantitative evaluation, we calculate the common joint and vertex error metrics in the camera space and report them in millimeters (mm), including MPJPE (mean per-joint position error) [46], PA-MPJPE (Procrustes-aligned mean per-joint position error) [153], and MPVE (mean per-vertex error) [100].

3.5.2 Quantitative Comparisons

Human3.6M and 3DPW. In Table 3.1, we compare the performance of our method and prior arts on the Human3.6M [46] and 3DPW [128] datasets. For VisDB and I2L-MeshNet [90], we report the results of both heatmap-based mesh outputs (mesh) and SMPL parameters (param). Our SMPL parameters are obtained from regression and test-time optimization. Note that each method uses different network backbone, human body representation, training datasets, and inference strategy. For instance, METRO [72] and Mesh Graphormer [73] adopt a transformer-based [127] network while the others use CNN

Table 3.1: **Quantitative evaluations on Human3.6M [46] and 3DPW [128].** To align the settings, we train our baseline, I2L-MeshNet [90], on the same datasets, and denote it by I2L-MeshNet†. Both our mesh and SMPL parameter outputs perform favorably against the prior state-of-the-arts.

		Human3.6M		3DPW		
Method	Output	MPJPE↓	PA-MPJPE↓	MPJPE↓	PA-MPJPE↓	MPVE↓
GraphCMR [58]	Mesh	-	50.1	-	70.2	-
Pose2Mesh [15]	Mesh	64.9	47.0	89.2	58.9	109.3
I2L-MeshNet [90]	Mesh	55.7	41.1	93.2	57.7	109.2
I2L-MeshNet† [90]	Mesh	-	-	84.5	51.1	98.2
METRO [72]	Mesh	54.0	36.7	77.1	47.9	88.2
Mesh Graphormer [73]	Mesh	51.2	**34.5**	74.7	45.6	87.7
VisDB (mesh)	Mesh	**51.0**	**34.5**	**73.5**	**44.9**	**85.5**
NBF [93]	Param	-	59.9	-	-	-
HMR [48]	Param	88.0	56.8	-	81.3	-
DenseRaC [134]	Param	76.8	48.0	-	-	-
I2L-MeshNet [90]	Param	-	-	100.0	60.0	121.5
OOH [151]	Param	-	41.7	-	-	-
SPIN [57]	Param	-	41.1	-	59.2	116.4
I2L-MeshNet† [90]	Param	-	-	88.0	55.5	102.3
DSR [23]	Param	60.9	40.3	85.7	51.7	99.5
VIBE [55]	Param	65.6	41.4	82.0	51.9	99.1
TCMR [14]	Param	62.3	41.1	-	-	-
DecoMR [147]	Param	60.6	39.3	-	-	-
PARE [56]	Param	-	-	79.1	46.4	94.2
VisDB (param)	Param	**50.0**	**33.8**	**72.1**	**44.1**	**83.5**

backbones. VIBE [55] and TCMR [14] are video-based approaches whereas the others only take images as input. Despite these differences, VisDB performs favorably against prior methods in term of most evaluation metrics. Particularly, our method achieves larger performance gains on the 3DPW dataset since it contains more truncation and occlusion cases. The VisDB performance is most directly comparable with I2L-MeshNet [90] as we adopt similar training settings. For fair comparisons, we re-train its model on the same datasets and denote it as I2L-MeshNet†. The results demonstrate that our visibility learning improves both the mesh and SMPL outputs significantly. In prior literature, SMPL parameters generally lead to higher errors compared to dense mesh outputs, which we conjecture is caused by the difficulty to directly regress low-dimensional parameters. On

Table 3.2: **Quantitative evaluations on 3DOH [151] and 3DPW-OCC [128, 151].** We compare VisDB with prior occlusion-aware methods to demonstrate its robustness on partial-body cases. For VisDB and I2L-MeshNet† [90], We report both the mesh and SMPL parameter (mesh/param) results.

	3DOH		3DPW-OCC		
Method	MPJPE↓	PA-MPJPE↓	MPJPE↓	PA-MPJPE↓	MPVE↓
OOH [151]	-	58.5	-	72.2	-
I2L-MeshNet† [90]	67.0/69.3	46.3/47.9	96.5/98.0	61.0/62.6	120.2/127.0
PARE [56]	63.3	44.3	91.4	57.4	115.3
VisDB	**62.1/60.9**	**43.2/42.7**	**90.3/87.3**	**57.1/56.0**	**114.0/110.5**

the contrary, VisDB is a powerful intermediate representation that provides dense 3D information of visible partial body, allowing us to regress and optimize SMPL parameters more accurately. In our experiments, we observe that VisDB (mesh) captures the human silhouettes better but VisDB (param) produces lower errors since the ground-truth meshes are also regularized by SMPL representation.

3DPW-OCC and 3DOH. To emphasize the robustness on partial-body images, we further evaluate on two occlusion datasets: 3DPW-OCC [128, 151] and 3DOH [151]. As shown in Table 3.2, VisDB produces lower errors on both datasets compared to prior occlusion-aware methods. While I2L-MeshNet† performs considerably worse on these images, the errors by our model remain relatively low.

3.5.3 Ablation Studies

VisDB network training. To evaluate the contribution of individual components in our method, we perform ablation studies on the 3DPW dataset [128]. Table 3.3 shows the performance of VisDB mesh outputs with/without truncation modeling ${\mathcal{L}_{vis}}^{x,y}$, occlusion modeling ${\mathcal{L}_{vis}}^{z}$, depth ordering loss $\mathcal{L}_{depth}$, and dense UV correspondence loss $\mathcal{L}_{uv}$. Without ${\mathcal{L}_{vis}}^{x,y}$, ${\mathcal{L}_{vis}}^{z}$, $\mathcal{L}_{depth}$, and $\mathcal{L}_{uv}$, the vertex error increases by 6.3mm, 3.1mm, 3.9mm,

Table 3.3: **Ablation studies of VisDB.** We compare the joint/vertex errors of VisDB mesh outputs on 3DPW [128] with/without individual components. The results show that truncation modeling ($\mathcal{L}_{vis}{}^{x,y}$), occlusion modeling ($\mathcal{L}_{vis}{}^{z}$), depth ordering loss $\mathcal{L}_{depth}$, and UV correspondence loss $\mathcal{L}_{uv}$ each reduces the errors by a clear margin.

$\mathcal{L}_{vis}{}^{x,y}$	$\mathcal{L}_{vis}{}^{z}$	$\mathcal{L}_{depth}$	$\mathcal{L}_{uv}$	MPJPE	PA-MPJPE	MPVE
				84.5	51.1	98.2
	✓	✓	✓	79.4	47.8	91.1
✓		✓	✓	75.8	45.5	88.0
✓	✓		✓	77.3	46.3	88.9
✓	✓	✓		74.9	45.6	87.1
✓	✓	✓	✓	**73.5**	**44.9**	**85.5**

and 1.9mm, respectively. These results show that both visibility modeling and depth ordering loss play a crucial role in VisDB training.

SMPL parameter fitting. In Table 3.4, we quantitatively compare the SMPL models obtained from different methods. Given an estimated VisDB mesh, we can regress the SMPL parameters and/or optimize them during inference, and each process can be done with/without dense visibility weighting (Eq. (3.13) and (3.14)). By using visibility, the mean vertex error of regressed SMPL models drops by 8.7mm. With the proposed test-time optimization, we can further reduce the error by 3.8mm.

3.5.4 Qualitative Results

Figure 3.4 shows sample results by VisDB and I2L-MeshNet [90] on the 3DPW dataset [128]. I2L-MeshNet [90] regresses SMPL parameters from the entire heatmap-based mesh output, which leads to erroneous output meshes on truncated or occluded examples. VisDB predicts accurate vertex visibility labels, improving both the image-space dense body estimation and SMPL parameter optimization. The results show that VisDB (mesh) outputs can fit the human silhouettes faithfully, and VisDB (params) further regularizes and

Table 3.4: **Ablation studies of SMPL models**. We report the performance of SMPL outputs on the 3DPW dataset [128], which shows the effectiveness of our optimization and the importance of visibility in both regression and optimization work flows.

Regression	Optimization	MPJPE	PA-MPJPE	MPVE
-	-	73.5	44.9	85.5
w/o vis	-	79.0	48.8	96.2
w/o vis	w/o vis	77.6	47.0	93.9
w/ vis	-	74.9	45.3	87.3
w/ vis	w/ vis	**72.1**	**44.1**	**83.5**

smooths the mesh surfaces.

3.6 Conclusion

In this chapter, we address the problem of dense human body estimation from monocular images. Particularly, we identify the limitations of existing model-based and heatmap-based representations on truncated or occluded bodies. As such, we propose a visibility-aware dense body representation, VisDB. We obtain visibility pseudo ground-truths from dense UV correspondences and train a network to predict 3D coordinates as well as truncation and occlusion labels for each human joint and vertex. Extensive experimental results show that visibility modeling can facilitate human body estimation and allow accurate SMPL fitting from partial-body predictions.

Figure 3.4: **Qualitative results on the 3DPW dataset [128].** For each example, we show the results of I2L-MeshNet [90] SMPL model, our VisDB mesh, our optimized SMPL model, as well as visibility predictions in the front and side views (purple:visible, orange:invisible). When the human body is occluded (top two rows) or truncated (bottom two rows), both our VisDB output and optimized SMPL mesh capture the human silhouettes faithfully (*e.g.*, the left hand in row 1 and the head region in rows 2,3,4).

Chapter 4

LASSIE: Learning Articulated Shapes from Sparse Image Ensemble via 3D Part Discovery

4.1 Overview

Creating high-quality articulated 3D models of animals is challenging either via manual creation or using 3D scanning tools. Therefore, techniques to reconstruct articulated 3D objects from 2D images are crucial and highly useful. In this chapter, we propose a practical problem setting to estimate 3D pose and shape of animals given only a few (10-30) in-the-wild images of a particular animal species (say, horse). Contrary to existing works that rely on pre-defined template shapes, we do not assume any form of 2D or 3D ground-truth annotations, nor do we leverage any multi-view or temporal information. Moreover, each input image ensemble can contain animal instances with varying poses, backgrounds, illuminations, and textures. Our key insight is that 3D parts have much simpler shape compared to the overall animal and that they are robust w.r.t. animal pose articulations. Following these insights, we propose LASSIE, a novel optimization

framework which discovers 3D parts in a self-supervised manner with minimal user intervention. A key driving force behind LASSIE is the enforcing of 2D-3D part consistency using self-supervisory deep features. Experiments on Pascal-Part and self-collected in-the-wild animal datasets demonstrate considerably better 3D reconstructions as well as both 2D and 3D part discovery compared to prior arts.

4.2 Introduction

The last decade has witnessed significant advances in estimating human body pose and shape from images [78, 5, 48, 58, 57, 98, 55]. Much progress is fueled by the availability of rich datasets [46, 81, 128] with 3D human scans in a variety of shapes and poses. In contrast, 3D scanning wild animals is quite challenging, and existing animal datasets such as SMALR [155] rely on man-made animal shapes. Manually creating realistic animal models for a wide diversity of animals (e.g. mammals and birds) is also challenging and time-consuming. In this chapter, we propose to automatically estimate 3D shape and articulation (pose) of animals from only a sparse set of image ensemble (collection) without using any 2D or 3D ground-truth annotations. Our problem setting is highly practical as each image ensemble consists of only a few (10-30) in-the-wild images of a specific animal species. Fig. 4.1 shows sample zebra images which forms our input. In addition, we assume a human-specified 3D skeleton (left in Fig. 4.1) that can be quite rough with incorrect bone lengths and this mainly provides the connectivity of animal parts. For instance, we use the same 3D skeleton for all the quadruped animals despite considerable shape variations across species, e.g., giraffe vs. zebra vs. elephant. Manually specifying a rough skeleton is a trivial task which only takes a few minutes.

Estimating articulated 3D shapes from in-the-wild image ensemble is a highly ambiguous and challenging problem. There are several varying factors across images: backgrounds; lighting; camera viewpoints; animal shape, pose and texture, etc. In addition, our highly practical problem setting does not allow access to any 3D animal models, per-image

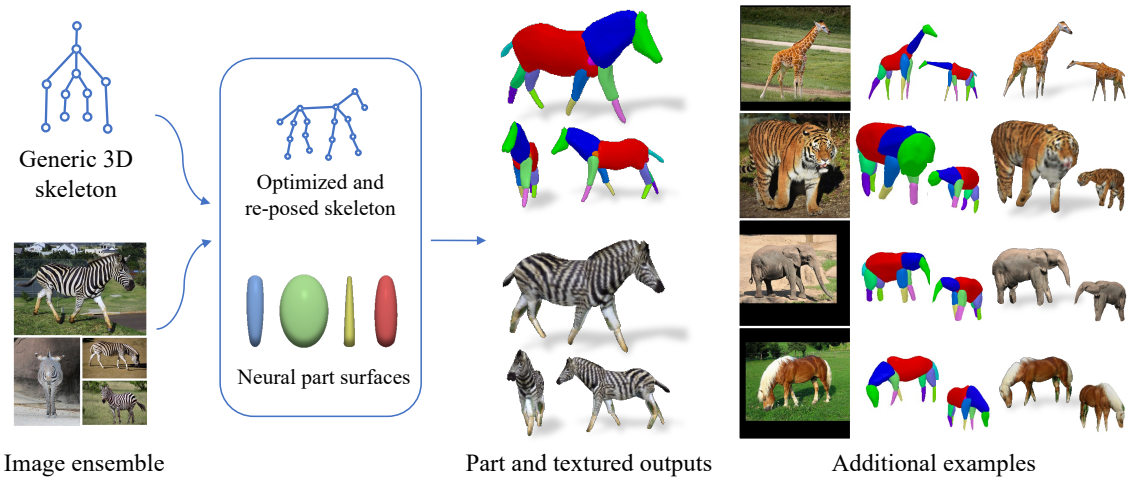

Figure 4.1: **Articulated shape optimization from sparse images in-the-wild.** Given 10-30 images of an articulated class and a generic 3D skeleton, we optimize the shared skeleton and neural parts as well as the instance-specific camera viewpoint and bone transformations. Our method is able to produce high-quality outputs without any pre-defined shape model or instance-specific annotations. The part-based representation also allows applications like texture and pose transfer, animation, etc.

annotations such as keypoints and silhouettes, or multi-view images.

In this chapter, we present LASSIE, an optimization technique to **L**earn **A**rticulated **S**hapes from **S**parse **I**mage **E**nsemble. Our key observation is that shapes are composed of 3D parts with the following characteristics: 1) 3D parts are geometrically and semantically consistent across different instances. 2) Shapes of 3D parts remain relatively constant w.r.t. changes in animal poses. 3) Despite the complex shape of overall animal body, 3D parts are usually made of simple convex shapes. Following these observations, LASSIE optimizes 3D parts and their articulation instead of directly optimizing the overall animal shape. Specifically, LASSIE optimizes a 3D skeleton and part shapes that are shared across instances; while also optimizing instance-specific camera viewpoint, pose articulation, and surface texture. To enforce semantic part consistency across different images, we leverage deep features from a vision transformer [22] (DINO-ViT [9]) trained in a self-supervised fashion. Recent methods [2, 125] demonstrate good 2D feature corre-

spondences in DINO-ViT features and we make use of these correspondences for 3D part discovery. To regularize part shapes, we constrain the optimized parts to a latent shape manifold that is learned using 3D geometric primitives (spheres, cylinders, cones etc.). Being self-supervised at the image level, LASSIE can easily generalize to a wide range of animal species with minimal user intervention.

We conduct extensive experiments on image ensembles from the Pascal-Part Dataset [12] with quantitative evaluations using 2D ground-truth part segmentation and keypoints. We also collect in-the-wild web image ensembles with more animal species and annotate 2D keypoints for evaluation. Both qualitative and quantitative results demonstrate considerably better 3D reconstructions compared to the prior arts that use even more supervision. In addition, estimation of explicit 3D parts enables easy editing and manipulation of animal parts. Fig. 4.1 shows sample 3D reconstructions with discovered parts from LASSIE. The main contributions of this work are:

- To the best of our knowledge, this is the first work that recovers 3D articulated shapes from in-the-wild image ensembles without using any pre-defined animal shape models or instance-specific annotations like keypoints or silhouettes.
- We propose a novel optimization framework called LASSIE, where we incorporate several useful properties of mid-level 3D part representation. LASSIE is category-agnostic and can easily generalize to a wide range of animal species.
- LASSIE can produce high-quality 3D articulated shapes with state-of-the-art results on existing benchmarks as well as self-collected in-the-wild web image ensembles.

4.3 Related Work

Animal shape and pose estimation. Estimating 3D pose and shape of animal bodies from in-the-wild images is highly ill-posed and challenging due to large variations across different classes, instances, viewpoints, articulations, etc. Most existing mesh reconstruction work [62, 69, 33, 122, 145] focus more on compact shapes like birds and cars, and thus

cannot handle articulations of animal bodies. Recent articulation-aware methods, on the other hand, deal with a simplified scenario by assuming a pre-defined class-level statistical model, instance-specific images, or accessible human annotations. For instance, Zuffi *et al.* [156] build a statistical shape model, SMAL, for common quadrupedal animals, similar to the SMPL [78] model for human bodies. Follow-up work either optimize the SMAL parameters based on human annotated images [155] or train a neural network on large-scale image datasets to directly regress the parameters [154, 110]. However, the output shapes are limited by the SMAL shape space or training images which contain one or few animal classes. Kulkarni *et al.* [61] propose A-CSM technique that align a known template mesh with skinning weights to input 2D images. Other recent approaches [137, 136, 138] exploit the dense temporal correspondence in video frames to reconstruct articulated shapes. In contrast to existing methods, we deal with a novel and practical problem setting: learn without any pre-defined shape model, instance-level human annotations, or temporal information to leverage.

Part discovery from image collections. Deep feature factorization (DFF) [18] shows that one could automatically obtain consistent part segments by clustering intermediate deep features of a classification network trained on ImageNet. Inspired by DFF, SCOPS [44] learns a 2D part segmentation network from image collections, which can be used as semantic supervision for 3D reconstruction [69]. Choudury *et al.* [16] proposed to use contrastive learning for 2D part discovery. Lathuilière *et al.* [65] exploit motion cues in videos for part discovery. Recently, Amir *et al.*[2] demonstrate that clustering self-supervisedly learned vision transformers (ViT) such as DINO [9] features can provide good 2D part segmentations. In this chapter, we make use of DINO features for 3D part discovery instead of 2D segmentation. In the 3D domain, Tulsiani *et al.* [123] use volumetric cuboids as part abstractions to learn 3D reconstruction. Mandikal *et al.* [80] predict part-segmented 3D reconstructions from a single image. In addition, Luo *et al.* [79] and Paschalidou *et al.* [94, 95] learn to form object parts by clustering 3D points. Considering that most 3D approaches require ground-truth 3D shape of a whole object or its parts as supervision,

Yao *et al.* [140] propose to discover and reconstruct 3D parts automatically by learning a primitive shape prior. Most of these works assume some form of supervision in terms of 3D shape or camera viewpoint. In this chapter, we aim to discover 3D parts of articulated animals from image ensembles without any 2D or 3D annotations, which, to the best of our knowledge, is unexplored in the literature.

3D reconstruction from sparse images. Optimizing a 3D scene or object from multi-view images is one of the widely-studied problems in computer vision. The majority of recent breakthroughs are based on the powerful Neural Radiance Field (NeRF) [85] representation. Given a set of multi-view images, NeRF can learn a neural volume from which one can render high-quality novel views. Closely related to our work, NeRS [148] optimizes a neural surface representation from a sparse set of image collections and also only requires rough camera initialization. In this chapter, we use a neural surface representation to represent animal parts. Existing methods assume multi-view images captured in the same illumination setting and also the object appearance (texture) to be same across images. Another set of work [6, 149, 8] propose neural reflectance decomposition on image collections captured in varying illuminations. But they work on rigid objects with ground-truth segmentation and known camera poses. A concurrent work, SAMURAI [7] jointly reasons about camera pose along with shape and materials of rigid objects from image collections. In contrast, the input for our method consists of in-the-wild animal images with varying textures, viewpoints and pose articulations and captured in different environments.

4.4 Approach

Given a generic 3D skeleton and few images of an articulated animal species, LASSIE optimizes the camera viewpoint and articulation for each instance as well as the resting canonical skeleton and part shapes that are shared across all instances. We introduce our neural part representation in Section 4.4.1 and the optimization framework in Sec-

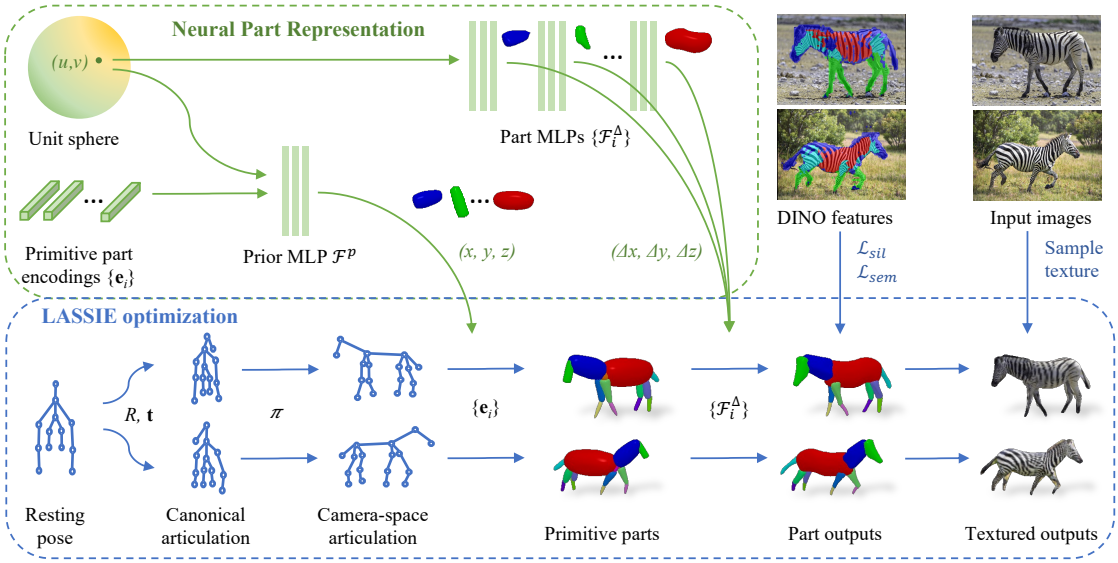

Figure 4.2: **Neural part surface representation.** Based on an optimized 3D skeleton, we reconstruct the articulated shape by optimizing the primitive latent codes and part deformation decoders. The final output is the composition of neural part surfaces with textures sampled from the input images.

tion 4.4.2. An overview of LASSIE is illustrated in Fig. 4.2.

4.4.1 Articulated Shapes with Neural Part Surfaces

3D skeleton. Constructing a template mesh or statistical shape model for specific animal classes requires either the ground-truth 3D shapes or extensive human annotation, which are both hard to obtain in real-world scenarios. A 3D skeleton, on the other hand, is easy to annotate since it only contains a set of sparse joints connected by bones. Furthermore, it generalizes well across classes, and one can effortlessly modify certain joints or bones to adapt to distinct animal types (e.g., mammals vs. birds). Assuming a simple and generic 3D skeleton is given, we propose a part-based representation for articulated shapes. A skeleton can be defined by a set of 3D joints $P \in \mathbb{R}^{p\times 3}$ and bones $B = \{(P_i, P_j)\}^b$ in a tree structure, where p and $b = p - 1$ are numbers of joints and bones/parts, respectively.

For instance, we use the same base skeleton for all four-legged animals in our experiments. The base skeleton specifies the connection of 16 joints and the initial joint coordinates in the root-relative 3D space. The skeleton is shared by all instances in the same class and can be optimized by scaling the bone lengths and rotating the bones with respect to the parent joints. Unlike the linear or neural blend skinning models [136, 138] where bones are unconstrained Gaussian components, our skeleton-based representation allows us to regularize bone transformations, discover meaningful and connected parts, and produce realistic shape animations.

Neural part surfaces. Given a skeleton which specifies the 3D joints and bones, we represent the articulated shape as a composition of several neural parts. Specifically, each part is defined as a neural surface wrapped around a skeleton bone. Motivated by object representation in NeRS [148], we represent part surface as a multi-layer perceptron (MLP) network that predicts the 3D deformation of a given continuous UV coordinate on the unit sphere. A key advantage of modeling parts instead of whole animal is that part shapes are usually of simple convex shapes and can be represented well with a deformed sphere. Compared to explicit mesh representation, the neural representation enables efficient mesh regularization on the output shapes while producing high resolution surfaces as MLPs can work with meshes of arbitrary vertex resolution. Each part surface is first reconstructed in the canonical space, then scaled by the corresponding bone length, rotated by the bone orientation, and centered at the bone centroid. This ensures that the parts are connected under various bone transformations and also easy to repose. Formally, let $X \in \mathbb{R}^{3\times m}$ denote the m uniformly distributed 3D vertices on a unit sphere. Further let $R_i \in \mathbb{R}^{3\times 3}$, $\mathbf{t}_i \in \mathbb{R}^3$ and $s_i \in \mathbb{R}$ denote the global rotation, translation and scaling of the i^{th} part ($i \in \{0, \cdots, b\}$), where b denotes the number of parts. We first deform the sphere to obtain part shape using the part MLP, $X \rightarrow \mathcal{F}_i(X)$; followed by global transformation to obtain part vertices $V_i \in \mathbb{R}^{3\times m}$ in the global coordinate frame:

$$V_i = s_i R_i \mathcal{F}_i(X) + \mathbf{t}_i. \tag{4.1}$$

Latent part prior. Although fitting neural part surfaces on sparse images can produce reasonable outputs that are faithful to the input view, we observe that part shapes tend to be non-uniform or unrealistic from other novel views. To further regularize the part shapes, we decompose the part MLP $\mathcal{F}_i$ into a prior MLP, $\mathcal{F}^p$ and a part deformation MLP, $\mathcal{F}_i^{\Delta}$. The prior MLP is trained to produce the base shapes that are close to geometric primitives such as spheres, cylinders, etc. The deformation MLP is then used to provide additional deformations on top of the base shapes. Fig. 4.2 illustrates these two MLPs. As shown in the figure, the prior MLP takes as input the surface coordinate as well as a d-dimensional latent shape code $\mathbf{e}_i \in \mathbb{R}^d$. Both the latent space and the prior MLP are trained using geometric primitive shapes. Concretely, we train a Variational Auto-Encoder (VAE) which takes a primitive mesh as input, predicts the latent shape encodings and reconstructs the primitive through a conditional neural surface decoder which forms our prior MLP. The primitive meshes in this chapter are randomly sampled geometric primitives (spheres, ellipsoids, cylinders, and cones). With the use of prior and deformation MLPs, Eq. 4.1 now becomes:

$$V_i = s_i R_i(\mathcal{F}^p(X, \mathbf{e}_i) + \mathcal{F}_i^{\Delta}(X)) + \mathbf{t}_i, \tag{4.2}$$

where $\mathbf{e}_i$ denotes the latent shape encodings of part i.

4.4.2 Discovering 3D Neural Parts from Image Ensemble

Optimization setting. The input to our optimization is a sparse set (typically 20-30) of n in-the-wild images $\{I_j\}_{j=1}^n$ along with a generic/rough 3D skeleton $P \in \mathbb{R}^{p\times 3}$ that has p joints and b bones/parts. The only image-level annotations for optimization are from the self-supervisory DINO [9] features. All the instances in the input image ensemble are of same species, but could have varying appearance due to texture, pose, camera angle, lighting, etc. We also assume that the entire animal body is visible in each image without truncated or occluded by another object although some degree of self-occlusion is allowed. Let j denote the index over images $j \in \{1, ., n\}$ and i denote the index over parts $i \in$

$\{1, ., b\}$. For each image, LASSIE optimizes the camera viewpoint $\pi^j = (R_0, t_0)$ and part rotations $R^j \in \mathbb{R}^{b\times3\times3}$. In addition, LASSIE optimizes these variables for each of the i^{th} part that are shared across all the animal instances: bone length scaling $s_i \in \mathbb{R}$, latent shape encodings $\mathbf{e}_i \in \mathbb{R}^d$ and part deformation MLPs $\mathcal{F}_i^{\Delta}$. That is, we assume that part shapes are shared across all instances whereas their articulation (pose transformation) is instance dependent. Although this may not be strictly valid in practice, this provides an important constraint for our highly ill-posed optimization problem.

Analysis by synthesis. Since we do not have access to any form of 3D supervision, we use analysis-by-synthesis to drive the optimization. That is, we use differentiable rendering to render the optimized 3D articulated shape and define loss functions w.r.t. input images. Using Eq. 4.2, we first obtain the complete articulated shape for each image as a combination of different part meshes. We then use a differentiable mesh renderer [76] to project the 3D points onto 2D image space via the learnable camera viewpoints. A key challenge is that we do not have access to any from of 2D annotations such as keypoints or part segmentation.

Self-supervisory deep features to the rescue. Since different instances in the ensemble can have different texture, we do not model and render image pixel values. Instead, we rely on learned deep features for our self-supervisory analysis-by-synthesis framework. Recent works [2, 36] demonstrate that deep features from DINO [9] vision transformer are semantically descriptive, robust to appearance variations, and higher-resolution compared to CNN features [2, 125]. In this chapter, we use the *key* features from the last layer of the DINO network as semantic visual features for each image. We assume that these DINO features are relatively similar for same parts across different instances. In addition to using DINO features for consistent semantic features across images, we also compute 2D parts and foreground animal mask following similar steps as in [2]. Concretely, we take the class tokens from the last layer, compute the mean attention of class tokens as saliency maps, and sample the keys of image patches with high saliency scores. We then collect the salient keys from all images and apply an off-the-shelf K-means clustering

algorithm to form c semantic part clusters. Finally, we obtain foreground silhouettes by simply thresholding the pixels with minimum distance to cluster centroids. Fig. 4.3 shows the DINO part clusters on sample images from different image ensembles. We observe that using 4 clusters works well in practice. With the use of self-supervised features (keys) and rough silhouettes, we alleviate the dependency of ground-truth segmentation mask and keypoint annotations in our optimization framework.

Silhouette loss. We compute the silhouette loss $\mathcal{L}_{mask}$ by rendering all part surfaces with a differentiable renderer [76] and comparing with the pseudo ground-truth foreground masks obtained via clustering DINO features: $\mathcal{L}_{mask} = \sum_j \|M^j - \hat{M}^j\|^2$, where M^j and $\hat{M}^j$ are the rendered and pseudo ground-truth silhouettes of instance j, respectively. Although these silhouette-based losses encourage the overall shape to match the 2D silhouettes, their supervisory signal is often too coarse to resolve the ambiguity caused by different camera viewpoints and pose articulations. Therefore, we design a novel semantic consistency loss to more densely supervise the reconstruction.

2D-3D semantic consistency. We propose a novel strategy to make use of the semantically consistent 2D DINO features for 3D part discovery. At a high-level, we impose the 2D feature consistency across the images via learned 3D parts. For this, we propose an Expectation-Maximization (EM) like optimization strategy where we alternatively estimate the 3D part features (E-step) from image ensemble features and then use these estimated 3D part features to update the parts (M-step). More formally, let $K^j \in \mathbb{R}^{h \times w \times f}$ denote the f-dimensional DINO key features from j^{th} image and $Q \in \mathbb{R}^{m \times f}$ denote the corresponding features on the 3D shape with total m vertices. In the E-step, we update Q by projecting the 3D coordinates onto all the 2D image spaces via differentiable rendering under the current estimated shape and camera viewpoints. We aggregate the 2D image features from the image ensemble corresponding to each 3D vertex to estimate Q. In the M-step, we optimize the 3D parts and camera viewpoints using a 2D-3D semantic consistency loss $\mathcal{L}_{sem}$. To define $\mathcal{L}_{sem}$, we discretize the foreground 2D coordinates in each image $\{p|\hat{M}^j(p) = 1\}$. Likewise, we sample a set of points $\{v \in V^j\}$ on the 3D

part surfaces and obtain their 2D projected coordinates $\pi^j(v)$. $\mathcal{L}_{sem}$ is then defined as the Chamfer distance in a high-dimensional space:

$$\mathcal{L}_{sem} = \sum_j \Big(\sum_{p|\hat{M}^j(p)=1} \min_{v \in V^j} \mathcal{D}(p, v) + \sum_{v \in V^j} \min_{p|\hat{M}^j(p)=1} \mathcal{D}(p, v) \Big). \tag{4.3}$$

For each instance j, the distance $\mathcal{D}$ between an image point p and 3D surface point v is defined as:

$$\mathcal{D}(p, v) = \underbrace{\|\pi^j(v) - p\|^2}_{\text{Geometric distance}} + \alpha \underbrace{\|Q(v) - K^j(p)\|^2}_{\text{Semantic distance}}, \tag{4.4}$$

where α is a scalar weighting for semantic distance. In effect, $\mathcal{L}_{sem}$ optimizes the 3D part shapes such that the aggregated 3D point features in the E-step would project closer to the similar pixel features in the image ensemble. We alternate between E and M steps every other iteration in the optimization process. In Eq. 4.4, the image coordinates p and semantic features K, Q are fixed in each optimization iteration, hence $\mathcal{L}_{sem}$ can effectively push the surface points towards their corresponding 2D coordinates by minimizing the overall distance. Since this EM-style optimization is prone to local-minima as we jointly optimize camera viewpoints, shapes and pose articulations, we find that a good initialization of 3D features is important. For this, we manually map each part index i to one of the DINO clusters (4 in our experiments) and then initialize the 3D features Q by assigning the average 2D cluster features to the corresponding 3D part vertices. This manual 3D part to 2D cluster assignment takes minimal effort since it is category-level (all instances share the same 3D parts). Alternatively, one could roughly initialize the camera viewpoints like in recent works such as NeRS [148] which allows the computation of initial Q features from 2D features. Nonetheless, 2D-3D assignment is easier and faster compared to camera viewpoint initialization as 2D-3D assignment only needs to be done once per image ensemble whereas camera initialization is required for each instance in the ensemble. To make the optimization process fully automatic, we also propose to use simple heuristics for the 2D-3D part assignment using intuitions like animal head are usually above the torso and legs at the bottom. We refer to the LASSIE optimization with this heuristic-based 3D

feature initialization as 'LASSIE-heuristic'.

Comparison to related methods. Our use of 2D-3D semantic consistency is closely related to prior works, UMR [69] and BANMo [138]. At a high-level, LASSIE and UMR both start from 2D part segmentations and then lift them onto 3D. A key difference is that LASSIE directly discovers parts in 3D using 2D feature consistency, whereas UMR uses 2D part consistency via canonical UV maps. More specifically, UMR uses 2D parts (not 2D features) and 2D UV maps as common canonical part representations. Instead, LASSIE directly optimizes 3D part features with feature based loss (not 2D parts) and LASSIE does not use UV maps like in UMR. We only use 2D parts for feature initialization and not during optimization. That is, we discover parts directly in 3D with skeleton constraints, without much reliance on intermediate 2D part discovery. Hence, LASSIE is less sensitive to the issues in 2D part segmentations compared to UMR. BANMo, on the other hand, learns a canonical feature embedding by enforcing the consistency between feature matching and geometric warping, which is made tractable by the dense temporal correspondence (optical flow) between video frames. The semantic loss in BanMo enforces the 2D-3D cycle consistency between 2D coordinates and canonical 3D points. Considering that our image ensemble is sparse and un-correlated, we coarsely initialize and regularize the 3D surface features at the part-level. The proposed semantic loss allows us to first localize the 3D parts then refine the detailed part shapes in our challenging setting.

Regularization. Our skeleton-based representation allows us to conveniently impose pose prior or regularization. Specifically, we calculate a pose prior loss $\mathcal{L}_{pose}$ to minimize the bone rotation deviations from the resting pose as: $\mathcal{L}_{pose} = \sum_j \|R^j - \bar{R}\|^2$, where R^j is the bone rotations of instance j and $\bar{R}$ denotes the bone rotations of shared resting pose. We also impose a regularization $\mathcal{L}_{ang}$ on joint angles on the animal legs to limit their sideway rotations:

$$\mathcal{L}_{ang} = \sum_j \sum_{i \in \text{leg bones}} \|{R_{i,y}}^j\|^2 + \|{R_{i,z}}^j\|^2, \tag{4.5}$$

where $R_{i,y}$ and $R_{i,z}$ are the rotations of bone i with respect to y and z axes respectively. Since the skeleton faces the $+z$ direction in the canonical space ($+x$ right, $+y$ up, $+z$ out), minimizing the y and z-axis rotations essentially constrains the sideway movements of the bones. Note that both $\mathcal{L}_{pose}$ and $\mathcal{L}_{ang}$ are generic to all quadrupeds in our experiments, which we find crucial to avoid unrealistic poses. Finally, we apply common mesh regularization terms to encourage smooth surfaces, including the Laplacian loss $\mathcal{L}_{lap}$ and surface normal loss $\mathcal{L}_{norm}$. $\mathcal{L}_{lap}$ regularizes 3D surfaces by pushing each vertex towards the center of its neighbors, and $\mathcal{L}_{norm}$ encourages neighboring mesh faces to have similar normal vectors. Note that we apply regularization losses to each part surface individually since the part shapes should be primitive-like and usually do not have sharp or pointy surfaces.

Optimization and textured outputs. The overall optimization objective is:

$$\mathcal{L}_{mask} + \lambda_1 \mathcal{L}_{sem} + \lambda_2 \mathcal{L}_{pose} + \lambda_3 \mathcal{L}_{ang} + \lambda_4 \mathcal{L}_{lap} + \lambda_5 \mathcal{L}_{norm}, \tag{4.6}$$

where $\{\lambda_i\}$ are weighting hyper-parameters. We first pre-train the part prior MLP $\mathcal{F}^p$ with 3D geometric primitives and freeze it during the optimization on a given image ensemble. For each image ensemble of an animal species, we perform multi-stage optimization on the camera, pose, and shape parameters until convergence. That is, we update the camera viewpoints and fix the rest first, then optimize the bone transformations, and finally the latent part codes as well as part deformation MLPs. In each iteration, we first update the semantic features of 3D surfaces, then use the updated features to update 3D surfaces, forming an EM-style optimization. We implement the framework in PyTorch [97] and update all the learnable parameters using an Adam optimizer [53]. To obtain the final textured outputs, we densely sample vertices from the optimized neural surfaces and directly sample the colors of visible vertices from individual images via 2D projection, where the visibility information is obtained from rasterization. For the invisible (self-occluded) vertices, we assign the color by finding their left-right symmetries or nearest visible neighbors.

Figure 4.3: **Qualitative results on the in-the-wild image collections.** We show example results of the baselines as well as the part and texture reconstruction by LASSIE on the self-collected animal image ensembles. The results demonstrate the semantic consistency of discovered parts across diverse classes and the high-quality shape reconstruction that allows dense texture sampling.

4.5 Experiments

Datasets. Considering the novel problem of sparse-image optimization, we select a subset of the Pascal-Part dataset [12] and collect web images of additional animal classes to evaluate LASSIE and related prior arts. The Pascal-Part dataset contains diverse images of several animal classes which are annotated with 2D part segmentation masks. We automatically select images of horse, cow, or sheep with only one object covering 1-50% area of the entire image, and find the keypoints by calculating the centers/corners of ground-truth part masks. In addition, for analysis on more diverse animal categories, we collect sparse image ensembles (with CC-licensed images) of some other animals from the internet and manually annotate the 2D keypoints for evaluation. The classes include quadrupeds like zebra, tiger, giraffe, elephant, as well as bipeds like kangaroo and penguin. We filter out

the images where the animal body is heavily occluded or truncated, resulting in roughly 30 images per category. The LASSIE optimization and evaluations are performed on each image ensemble separately. The self-collected data does not contain personally identifiable information or offensive content, and will be released to public.

Baselines. Due to the lack of prior works on our problem setting (sparse image optimization for articulated animal shapes), we mainly compare LASSIE with learning-based mesh reconstruction methods. Among these methods, we find 3D Safari [154] and A-CSM [61] to be most comparable to LASSIE since they also model articulation for animal classes of our interest. Other recent mesh reconstruction methods, on the other hand, either cannot handle articulations [62, 69, 33, 122, 145] or assume different inputs [68, 136, 138]. Note that both 3D Safari and A-CSM are trained on large-scale image sets while LASSIE optimizes on a sparse image ensemble. To evaluate their methods on our dataset, we use the released models trained on the closest animal classes. For instance, the zebra model of 3D Safari and the horse model of A-CSM are used to evaluate on similar quadrupeds.

Visual comparisons. Fig. 4.3 shows the qualitative results of LASSIE against 3D Safari and A-CSM on the self-collected animal images. The 3D Safari [154] model is trained on zebra images and does not generalize well to other animal classes. A-CSM [61] assumes high-quality skinned model of an object category which enables detailed shapes of animal bodies. However, its outputs do not align well with the given 2D images. Our results demonstrate that LASSIE can effectively learn from sparse image ensemble to discover 3D articulated parts that are high-quality, faithful to input images, and semantically consistent across instances. Note that our 3D reconstructions also improve the 2D part segmentation from DINO-ViT feature clustering [2].

Keypoint transfer metrics. As there is no ground-truth 3D annotation in our datasets, we quantitatively evaluate using 2D keypoint transfer between each pair of images which is a standard practice [154, 61]. We map a given set of 2D keypoints on a source image onto the 3D part surfaces, and then project them to a target image using the optimized camera viewpoints, shapes and pose articulations. Since this keypoint transfer goes through the

Table 4.1: **Keypoint transfer evaluations.** We evaluate on all the source-target image pairs and report the percentage of correct keypoints under two different thresholds (PCK@0.1/ PCK@0.05).

	Pascal-Part dataset			Our dataset					
Method	Horse	Cow	Sheep	Zebra	Tiger	Giraffe	Elephant	Kangaroo	Penguin
3D Safari [154]	71.8/ 57.1	63.4/ 50.3	62.6/ 50.5	**80.8**/ 62.1	63.4/ 50.3	57.6/ 32.5	55.4/ 29.9	35.5/ 20.7	49.3/ 28.9
A-CSM [61]	69.3/ 55.3	68.8/ 60.5	67.4/ 54.7	78.5/ 60.3	69.1/ 55.7	71.2/ 52.2	67.3/ 39.5	42.1/ 26.9	53.7/ 33.0
LASSIE w/o $\mathcal{L}_{sem}$	60.4/ 45.6	58.9/ 43.1	55.3/ 42.3	62.7/ 47.5	53.6/ 40.0	54.7/ 28.9	52.0/ 25.6	33.8/ 19.8	50.6/ 25.5
LASSIE w/o $\mathcal{F}^p$	71.1/ 56.3	69.0/ 59.5	68.9/ 52.2	77.0/ 60.2	71.5/ 59.2	79.3/ 57.8	66.6/ 37.1	44.3/ 30.1	63.3/ 38.2
LASSIE	**73.0/ 58.0**	**71.3/ 62.4**	**70.8/ 55.5**	79.9/ **63.3**	**73.3/ 62.4**	**80.8/ 60.5**	**68.7/ 40.3**	**47.0/ 31.5**	**65.5/ 40.6**
LASSIE-heuristic	72.1/ 57.0	69.5/ 61.1	69.7/ 55.3	78.9/ 61.9	71.9/ 60.0	80.4/ 60.1	66.5/ 38.7	45.6/ 30.9	60.8/ 37.6

2D-to-3D and 3D-to-2D mappings, a successful keypoint transfer requires accurate 3D reconstruction on both the source and target images. We calculate the percentage of correct keypoints (PCK) under two different thresholds: $0.1\times\max(h, w)$ and $0.05\times\max(h, w)$, where h and w are image height and width, respectively. Table 4.1 shows that LASSIE achieves higher PCK on the Pascal-part [12] images and most other classes in our in-the-wild image collections. We also show ablative results of LASSIE without using the semantic feature consistency loss $\mathcal{L}_{sem}$ or part prior MLP $\mathcal{F}^p$. Note that $\mathcal{L}_{sem}$ is essential to fit the skeleton and part surfaces faithfully to the input images, and $\mathcal{F}^p$ provides effective regularization on part surfaces to generate realistic shapes from various viewpoints. LASSIE-heuristic, by using automatic 3D feature initialization, achieves slightly lower PCK but still performs favorably against prior arts overall. Example visual results of keypoint transfer are shown in Fig. 4.5.

2D overall/part IOU metrics. Our part-based representation can generate realistic 3D shapes and can even improve 2D part discovery compared to prior arts. In Table 4.2 and Fig. 4.4, we show the qualitative and quantitative comparisons against prior 3D works as well as 2D co-part discovery methods of SCOPS [44] and DINO clustering [2]. For SCOPS [44] and DINO clustering [2], we set the number of parts $N_p = 4$ (which we find to be optimal for these techniques) and manually assign each discovered part to the best

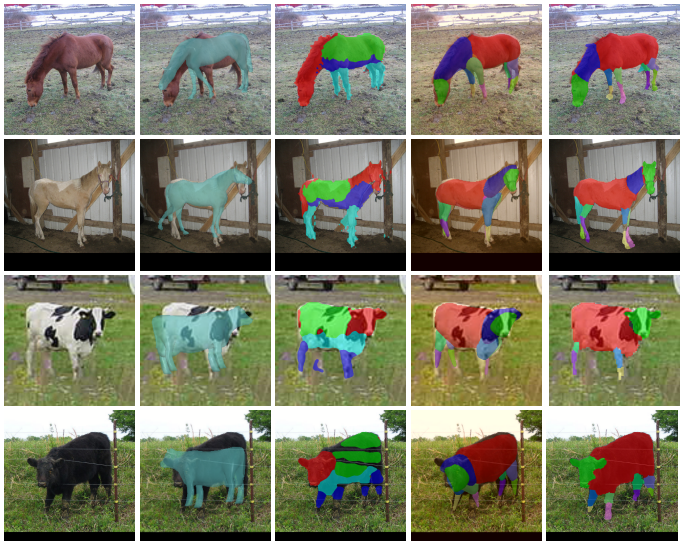

Figure 4.4: **2D segmentations.** We show the overall masks (A-CSM) and part masks (DINO clusters, LASSIE results, and Pascal-part GT) overlaid on the sample input images.

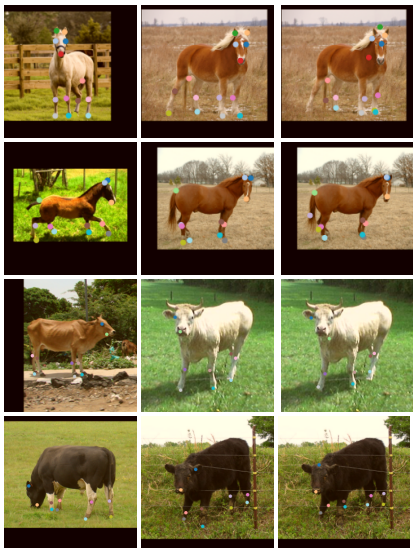

Figure 4.5: **Keypoint transfer results** using LASSIE from source to target images.

matched part in the Pascal-part annotations. Note that higher part IOU against Pascal-part annotations may not strictly indicate better part segmentation since the self-discovered parts need not correspond to human annotations. Compared to prior arts on articulated shape reconstruction, LASSIE results match the Pascal-part segmentation masks better and achieves higher overall IOU. On co-part segmentation, our 3D reconstruction captures the semantic parts in 2D while being more geometrically refined than DINO clustering. Moreover, our results can separate parts with similar semantic features, *e.g.*, 4 animal legs, and thus resulting in much higher part IOU.

Part transfer. While keypoint transfer and part IOU are commonly used in the literature, we observe that they are either sparse, biased towards certain body parts (*e.g.*, animal faces), or not directly comparable due to part mismatch. To address this issue, we propose a part transfer metric for better evaluation of 3D reconstruction consistency across image

Table 4.2: **Quantitative evaluations on the Pascal-part images.** We report the overall foreground IOU, part mask IOU, and percentage of correct pixels (PCP) under dense part segmentation transfer between all pairs of source-target images.

	Overall IOU			Part IOU			Part transfer (PCP)		
Method	Horse	Cow	Sheep	Horse	Cow	Sheep	Horse	Cow	Sheep
SCOPS [44]	62.9	67.7	63.2	23.0	19.1	26.8	-	-	-
DINO clustering [2]	81.3	85.1	83.9	26.3	21.8	30.8	-	-	-
3D Safari [154]	72.2	71.3	70.8	-	-	-	71.7	69.0	69.3
A-CSM [61]	72.5	73.4	71.9	-	-	-	73.8	71.1	72.5
LASSIE w/o $\mathcal{L}_{sem}$	65.3	67.5	60.0	29.7	23.2	31.5	62.0	60.3	59.7
LASSIE w/o $\mathcal{F}^p$	81.0	86.5	85.0	37.1	33.7	41.9	76.1	75.6	72.4
LASSIE	**81.9**	**87.1**	**85.5**	**38.2**	**35.1**	**43.7**	**78.5**	**77.0**	**74.3**

ensemble. This part transfer metric is similar to keypoint transfer but uses part segmentations. That is, we densely transfer the part segmentation in a source image to a target image through forward and backward mapping between the 2D pixels and the canonical 3D surfaces. A pixel is considered transferred correctly if and only if it is mapped to the same 2D part in the target image as where it belongs in the source image. Similar to PCK in keypoint transfer evaluation, we calculate the percentage of correct pixels (PCP) for each source-target pair. Since the part segmentation only covers the visible surfaces and not the occluded regions, we ignore the pixels mapped to a self-occluded surface when calculating the PCP metric. The quantitative results are shown in Table 4.2, which demonstrate the favorable performance of LASSIE against prior arts.

Applications. Our 3D neural part representation not only produces more faithful and realistic shapes but also enables various applications like part manipulation, texture transfer, motion re-targeting, etc. Specifically, we can generate a new shape by swapping or interpolating certain parts, transfer the sampled surface texture from one object class to another

Pose transfer / motion re-targeting
(top: target pose, bottom: re-posed output)

Texture transfer
(top: target texture, bottom: re-textured output)

Figure 4.6: **Applications of neural part surfaces.** We use the LASSIE results of zebra images (top) as target and apply pose transfer (bottom left) and texture transfer (bottom right) to the giraffe shapes. These applications demonstrate the realistic part surfaces discovered by LASSIE.

using the same base skeleton, or specify the bone rotations to produce desirable re-posed shape or animation. We show some example results of pose transfer and texture transfer between different animal classes in Fig. 4.6.

Limitations. LASSIE relies heavily on the 3D skeleton and strong pose and shape regularization. As the first attempt to address this novel and highly ill-posed problem, we find the proposed pose and shape constraints essential to produce good quality results and avoid unrealistic outputs. Moreover, LASSIE can not handle image collections with heavy truncation, occlusions, or extreme pose variations since we leverage self-supervisory DINO-ViT features on sparse image ensemble as our main supervision. For instance, partial-body or noisy DINO features can cause ambiguities in camera pose and part localization. Finally, we observe that LASSIE currently struggles with a) highly articulated parts like elephant trucks and b) fluffy animals that appear with more instance variations and ambiguous articulations.

4.6 Conclusion

In this chapter, we study a novel and practical problem of articulated shape optimization of animals from sparse image collections in-the-wild. Instead of relying on predefined shape models or human annotations like keypoints or segmentation masks, we propose a neural part representation based on a generic 3D skeleton, which is robust to appearance variations and generalizes well across different animal classes. In LASSIE, our key insight is to reason about animal parts instead of whole shape as parts allow imposing several constraints. We leverage self-supervisory dense ViT features to provide both silhouette and semantic part consistency for supervision. Both quantitative and qualitative results on Pascal-Part and our self-collected in-the-wild image ensembles show that LASSIE can effectively learn from few in-the-wild images and produce high-quality results. Our part-based representation also enables various applications such as pose interpolation, texture transfer, and animation. We hope that this work will facilitate future research efforts on learning, inferring, and manipulating articulated shapes from image ensembles in the wild.

Chapter 5

Hi-LASSIE: High-Fidelity Articulated Shape and Skeleton Discovery from Sparse Image Ensemble

5.1 Overview

Automatically estimating 3D skeleton, shape, camera viewpoints, and part articulation from sparse in-the-wild image ensembles is a severely under-constrained and challenging problem. Most prior methods rely on large-scale image datasets, dense temporal correspondence, or human annotations like camera pose, 2D keypoints, and shape templates. We propose Hi-LASSIE, which performs 3D articulated reconstruction from only 20-30 online images in the wild without any user-defined shape or skeleton templates. We follow the recent work of LASSIE that tackles a similar problem setting and make two significant advances. First, instead of relying on a manually annotated 3D skeleton, we automatically estimate a class-specific skeleton from the selected reference image. Second, we improve the shape reconstructions with novel instance-specific optimization strategies that allow reconstructions to faithful fit on each instance while preserving the class-specific priors

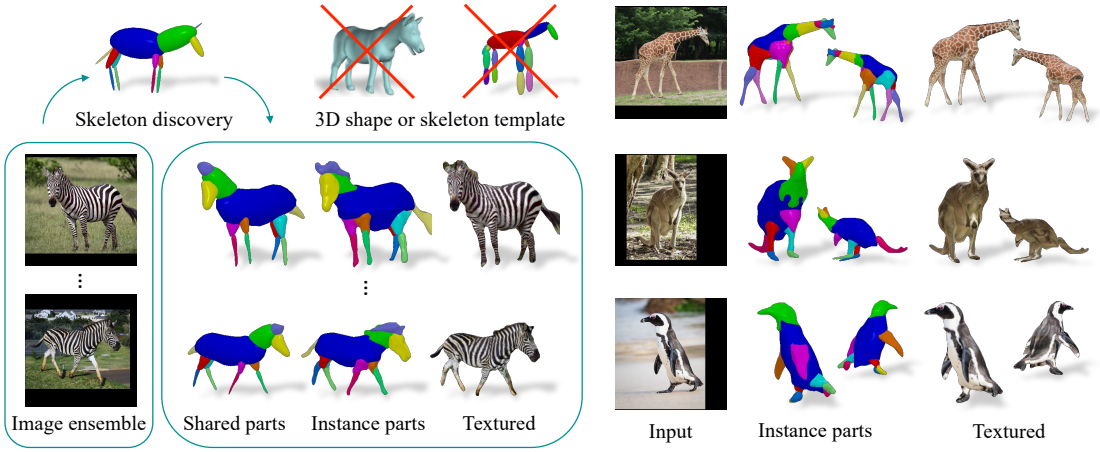

Figure 5.1: **Hi-LASSIE overview and sample reconstructions.** Given 20-30 images of an articulated animal class, we first discover a generic 3D skeleton, then jointly optimize the camera viewpoints, skeleton articulations, as well as shared and instance-specific neural part shapes. Hi-LASSIE is able to produce high-fidelity shapes and texture without any pre-defined shape model or 3D skeleton annotations. The part-based representation also allows applications like animation and motion re-targeting.

learned across all images. Experiments on in-the-wild image ensembles show that Hi-LASSIE obtains higher fidelity state-of-the-art 3D reconstructions despite requiring minimum user input.

5.2 Introduction

3D assets of articulated animals enable numerous applications in games, movies, AR/VR, etc. However, building high-fidelity 3D models of articulated shapes like animal bodies is labor intensive either via manual creation or 3D scanning. Recent advances in deep learning and 3D representations have significantly improved the quality of 3D reconstruction from images. Much of the success depends on the availability of either rich 3D annotations or multi-view captures, both of which are not always available in a real-world scenario. A more practical and scalable alternative is automatic 3D reconstruction from online im-

ages as it is straightforward to obtain image ensembles of any animal category (*e.g.*, image search results). In this chapter, we tackle a practical problem setting introduced in a recent work, LASSIE [141], where the aim is to automatically estimate articulated 3D shapes with only a few (20-30) in-the-wild images of an animal species, without any image-level 2D or 3D annotations.

This problem is highly under-constrained and challenging due to a multitude of variations within image ensembles. In-the-wild images usually have diverse backgrounds, lighting, and camera viewpoints. Moreover, different animal instances can have distinct 2D appearances due to pose articulations, shape variations, and surface texture variations (skin colors, patterns, and lighting). As shown in Fig. 5.2, early approaches in this space try to simplify the problem with some user-defined 3D templates, hurting the generalization of those techniques to classes where such templates are not always readily available. In addition, most of these methods (except LASSIE [141]) assume either large-scale training images of each animal species [62, 61, 154] or per-image 2D keypoint annotations [155], which also limits their scalability.

In this chapter, we propose a more practical technique that does not require any 3D shape or skeleton templates. Instead, as illustrated in Fig. 5.2, the user simply has to select a reference image in the ensemble where all the animal parts are visible. We achieve this by providing two key technical advances over LASSIE: 1) 3D skeleton discovery and 2) instance-specific optimization strategies. Our overall framework, named Hi-LASSIE, can produce **Hi**gher-fidelity articulated shapes than **LASSIE** [141] while requiring minimal human input. Our key insight is to exploit the 2D part-level correspondences for 3D skeleton discovery. Recent works [2, 125] observe that the deep features extracted from a self-supervised vision transformer (ViT) [22] like DINO-ViT [9] can provide good co-part segmentation across images. We further exploit such features to reason about part visibility and their symmetry. At a high level, we first obtain a 2D skeleton using the animal silhouette and part clusters [2] obtained from DINO features [9]. We then uplift this 2D skeleton into 3D using symmetric part information that is present in the deep DINO fea-

tures. Fig. 5.1 shows the skeleton for zebra images discovered by Hi-LASSIE. Similar to LASSIE [141], we leverage 3D part priors (learned from and shared across instances) and the discovered 3D skeleton to regularize the articulated shape learning. Furthermore, we design three novel modules to increase the quality of output shapes: 1) High-resolution optimization by zooming in on individual parts, 2) Surface feature MLPs to densely supervise the neural part surface learning, and 3) Frequency-based decomposition of part surfaces for shared and instance-specific components. Note that Hi-LASSIE can generalize to diverse animal species easily as it does not require any image annotations or category-specific templates.

We conduct extensive experiments on the Pascal-Part [12] and LASSIE [141] image ensembles, which contain in-the-wild images of various animal species like horse, elephant, and penguin. Compared with LASSIE and other baselines, we achieve higher reconstruction accuracy in terms of keypoint transfer, part transfer, and 2D IOU metrics. Qualitatively, Hi-LASSIE reconstructions show considerable improvement on 3D geometric and texture details as well as faithfulness to input images. Finally, we demonstrate several applications like animation and motion re-targeting enabled by our 3D part representation. Fig. 5.1 (right) shows some Hi-LASSIE 3D reconstructions for different animal species. The main contributions of this work are:

- To our best knowledge, Hi-LASSIE is the first approach to discover 3D skeletons of articulated animal bodies from in-the-wild image ensembles without using any image-level annotations. We show that the discovered 3D skeleton can faithfully fit all instances in the same class and effectively regularize the 3D shape optimization.
- Hi-LASSIE includes several novel optimization strategies that makes the output shapes richer in 3D details and more faithful to each image instance than prior methods.
- Extensive results on multiple animal classes and datasets demonstrate the state-of-the-art performance of Hi-LASSIE while requiring less user inputs than prior works.

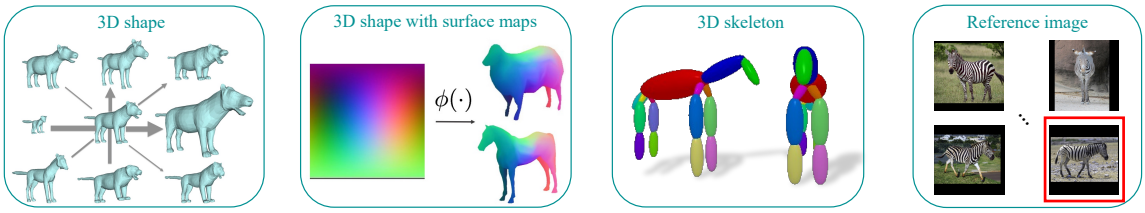

Figure 5.2: **User inputs across different techniques for articulated animal reconstruction.** Contrary to prior methods that leverage detailed 3D shapes or skeletons, Hi-LASSIE only requires the user to select a reference image where most animal body parts are visible.

5.3 Related Work

Animal pose and shape estimation. 3D pose and shape estimation of animal bodies from in-the-wild images is quite challenging considering the diverse 2D appearance across different instances, viewpoints, and articulations, among others. Most mesh reconstruction methods [62, 69, 33, 122, 145] mainly deal with objects with simple or rigid shapes (*e.g.*, birds and cars), which cannot be applied to the highly-articulated animal bodies. Recent articulation-aware approaches address a different (often simplified) scenario by leveraging a pre-defined statistical model [156, 155, 154, 110], 3D shape/skeleton templates [61, 141], or dense temporal correspondence in videos [137, 136, 138]. Without assuming any of these annotations/data, Hi-LASSIE introduces a novel and practical framework to estimate high-fidelity animal bodies by discovering 3D skeleton from a reference image.

Skeleton extraction and part discovery. 2D skeleton/outline extraction has been widely studied and used as geometric context for shape/pattern recognition [4, 133, 3, 115, 41]. However, these methods often fail to identify separate skeleton bones/parts when they overlap or self-occlude each other in an image. In this chapter, we propose to lift 2D skeletons to 3D for articulated shape learning by jointly considering the geometric and semantic cues in 2D images. For part discovery, deep feature factorization (DFF) [18] and follow-up works [44, 65, 16, 2] show that one could automatically obtain 2D corresponding part

segments by clustering deep semantic features. In the 3D domain, the object parts can be discovered by using volumetric cuboids [123], clustering 3D point clouds [79, 94, 95, 80], or learning part prior [140]. These methods mainly assume some form of supervision like 3D shapes or camera viewpoints. In this chapter, we discover 3D parts of articulated animals from image ensembles without any 2D or 3D annotations/templates, which, to the best of our knowledge, is unexplored in the literature.

3D reconstruction from sparse images. Optimizing a 3D scene or object from multi-view images is a fundamental problem in computer vision. The majority of recent breakthroughs are based on the powerful Neural Radiance Field (NeRF) [85] representation. Given a set of multi-view images, NeRF learns a neural volume from which one can render high-quality novel views. NeRS [148] introduces a neural surface representation to learn compact 3D shapes from a sparse image collection, which we find suitable to model individual animal body parts. Another line of work [6, 149, 8] propose neural reflectance decomposition on image collections captured in varying illuminations, but they operate on rigid objects with ground-truth segmentation and known camera poses. SAMURAI [7] jointly reasons about camera pose, shape, and materials from image collections of a single rigid object. In contrast, our input consists of in-the-wild animal images with varying textures, viewpoints, and pose articulations captured in different environments.

5.4 Approach

Given a sparse image ensemble of an animal species, Hi-LASSIE first discovers a class-specific 3D skeleton that specifies the initial 3D joint coordinates and part connectivity. Then, it jointly optimizes the camera viewpoint, pose articulation, and part shapes for each instance. Before describing our approach, we briefly review the LASSIE [141] method, where several parts are adopted in this chapter.

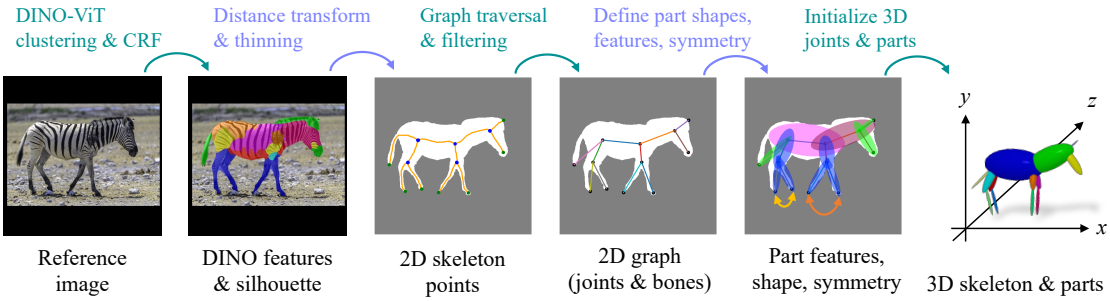

Figure 5.3: **3D Skeleton Discovery.** Given a reference image and its rough silhouette, we first extract, filter, and connect the 2D candidate points to form a 2D skeleton graph. Then, we find and split the symmetric parts by leveraging the 2D geometry and semantic cues. Finally, we design a simple heuristic to initialize the 3D joints, part shapes, and surface features.

5.4.1 Preliminaries: 3D Skeleton & Parts in LASSIE

Given a user-provided 3D skeleton template that specifies the 3D joints and bones, LASSIE [141] represents an overall articulated shape by assembling several 3D parts. In particular, each part is defined as a deformable neural surface [148] wrapped around a skeleton bone. The part surfaces are parameterized by multi-layer perceptron networks (MLPs) which predict the deformation of any 3D point on a surface template (*e.g.*unit sphere). Formally, let $X \in \mathbb{R}^{3\times m}$ denote the m uniformly sampled 3D points on a unit sphere, one can deform the 3D surface of the i-th part in the canonical space using the part MLP as $X \mapsto \mathcal{F}_i(X)$. The part surface is then rigidly transformed using the optimized 3D skeleton with scaling $s_i \in \mathbb{R}$, rotation $R_i \in \mathbb{R}^{3\times3}$, and translation $t_i \in \mathbb{R}^3$. The final part surface points V_i in the global coordinate frame can be expressed as:

$$V_i = s_i R_i \mathcal{F}_i(X) + t_i. \tag{5.1}$$

The rigid transformation of each part is defined by its corresponding bone length (scaling), orientation (rotation), and centroid (translation). This skeleton-based representation ensures that the connectivity of 3D parts under arbitrary articulations and easy to repose.

Compared to explicit mesh representation, the neural surfaces also enable efficient mesh regularization while producing high-resolution surfaces since the MLPs take continuous surface coordinates as input. A key innovation of LASSIE is the use of part prior within the part MLPs $\{\mathcal{F}_i\}$ that regularizes the part shapes to be close to convex primitive geometric shapes like spheres, cones, etc. Please refer to [141] for further details. To take advantage of these properties, we adopt a similar part-based representation in this chapter. A key difference from LASSIE is that we alleviate the need for user-provided 3D skeleton template and instead discover the 3D skeleton automatically from a reference image chosen from the ensemble. In addition, we propose several improvements to the neural surfaces and the optimization process to address the limitations of LASSIE resulting in higher-fidelity reconstructions.

5.4.2 Discovering 3D Skeleton

2D skeleton extraction. Fig. 5.3 illustrates our 3D skeleton discovery process from a reference image. We first extract a set of 2D skeleton points from a reference silhouette using a skeletonize/morphological thinning algorithm [152]. Specifically, we iteratively remove pixels from the borders until none can be removed without altering the connectivity. These 2D points can be seen as candidates of underlying skeleton joints and bones through camera projection. Similar to prior 2D shape matching methods [133, 3], we categorize the skeleton points into junctions, endpoints, and connection points. Assuming that each skeleton curve is one pixel wide, a skeleton point is called an *endpoint* if it has only one adjacent point; a *junction* if it has three or more adjacent points; and a *connection point* if it is neither an endpoint nor a junction. We sort the junctions and endpoints by distance transform (2D distance to closest border) and identify the 'root' junction with the largest distance to the border. Next, we find the shortest path (sequence of skeleton points) between root and each endpoint by traversing though the graph. From these paths, we can build a skeleton tree with 2D joints (junctions or endpoints) and bones (junction-junction

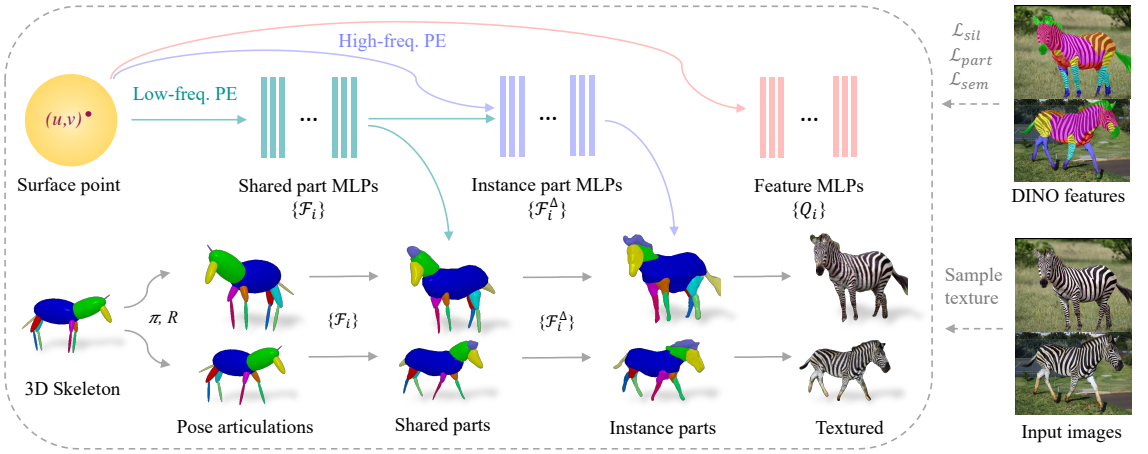

Figure 5.4: **Hi-LASSIE optimization framework.** Based on the discovered 3D skeleton, we reconstruct an articulated shape by optimizing the camera viewpoints, pose articulations, and part shapes. We represent the 3D part shapes as neural surfaces, which are decomposed into shared (low-frequency) and instance-specific (high-frequency) components via positional encoding (PE) of input surface coordinates. The only image-level annotation is from the self-supervisory DINO features, and the surface texture is sampled from the input images.

connection or junction-endpoint connection). Since the silhouettes and skeleton points are sometimes noisy, we filter those noisy 2D joints using non-maximal suppression. That is, we remove a point is if it lies within the coverage (radius calculated by distance transform) of its parent joint.

3D joint and part initialization. From the extracted 2D skeleton, one can roughly initialize the simple part shapes and surface features using the reference image. However, all the joints and bones would be initialized on a 2D plane (image plane) which is not sufficient for later 3D shape and camera pose optimization across instances. To obtain a better 3D skeleton initialization, we propose to find symmetric parts and separate them in the 3D space w.r.t. the symmetry plane. Our key insight is that symmetric parts (*e.g.*left and right legs/ears) share similar geometric and semantic features. Therefore, we design a heuristic to calculate the symmetry score of each pair of joints/bones based on the their geometric

and semantic feature distance. Specifically, we compute the following features for each joint/bone: length, average radius, and the average DINO features [9] along the paths to their common ancestor in the skeleton tree. Then, we identify pairs of joints that share similar features, which usually correspond to the symmetric animal parts (e.g., left and right leg). To uplift our 2D skeleton onto 3D, we set the z-coordinates of the each joint pairs to be on opposite sides of the symmetry plane ($z = 0$) and offset by their average radius (so that the initial 3D parts do not overlap). To better deal with overlapping parts in 2D silhouettes, we also split the parent of two symmetric parts if it has only two children. See Fig. 5.3 (right) for an example of estimated 3D skeleton.

5.4.3 Learning High-fidelity Articulated Shapes

Optimization setting. Our optimization framework takes a sparse set of n (typically 20-30) in-the-wild images $\{I_j\}_{j=1}^{n}$ as well as the discovered 3D skeleton $P \in \mathbb{R}^{p\times 3}$ with p joints and b bones/parts as input. The only image-level annotations like silhouettes and semantic clusters are obtained from the clustering results of self-supervisory DINO [9] features. All the images in an ensemble contain instances of the same species, but each instance could vary in pose, shape, texture, camera viewpoint, etc. We assume that each animal body is mostly visible in an image without severe truncation or occlusions by other objects although self-occlusion is allowed. We use j to denote the index over images $j \in \{1, ., n\}$ and i the index over parts $i \in \{1, ., b\}$. For each instance, Hi-LASSIE optimizes the camera viewpoint $\pi^j = (R_0, t_0)$, part rotations $R^j \in \mathbb{R}^{b\times 3\times 3}$, and neural part deformation MLPs. The overall framework is shown in Fig. 5.4, where we first perform pose articulation then progressively add more geometric details (from the shared part shapes to instance-specific deformations).

Per-instance deformation by frequency decomposition. Unlike LASSIE where the part shapes are shared across all instances, which limits the shape fidelity to input images, we propose to learn instance-specific part shapes that can better account for instance-varying

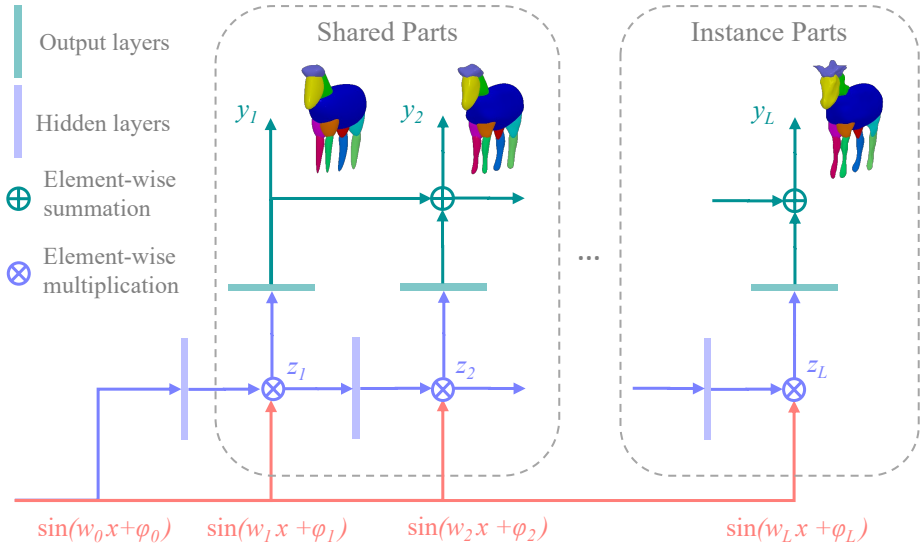

Figure 5.5: **Frequency-decomposed part MLP for per-instance shape deformation.** We design a MLP network to represent the neural surfaces composed of varying amount of details. By increasing the frequencies of input positional encoding, the outputs of deeper layers can express more detailed deformation. The output and hidden layers are linear layers.

and articulation-dependent deformations. Intuitively, instances of the same species should share similar base part shapes and only deviate in high-frequency details (*e.g.*ears and tail). Such detailed difference is usually caused by part articulations or instance variance. We observe that naively fine-tuning the part surfaces on one instance tends to overfit to that instance and nullify the 3D shape prior learned across all instances. For instance, the part shapes can overfit to the target silhouette (input view) but look unrealistic in novel views. To address this, we propose to decompose the surface deformations in the spatial frequency space so that each instance can have high-frequency detail variations while sharing the same low-frequency base part shapes. As shown in Fig. 5.5, we achieve this by designing a part deformation MLP with frequency-decomposed input and output components. In a forward pass through the MLP, we first apply positional encoding (PE) to an input coordinate $x \in \mathbb{R}^3$ using sines with varying frequencies ω and phases ϕ: $\text{PE}_i(x)$

$= \sin(\omega_i x + \phi_i)$, where $i = 0, ..., L$, and L is the number of layers in the network. The hidden activations z_i and final outputs y_i of layer i are computed as:

$$z_i = \text{PE}_i(x) \otimes (W_i^{\text{h}} z_{i-1} + b_i^{\text{h}}) \text{ with } z_0 = \text{PE}_0(x); \tag{5.2}$$

$$y_i = y_{i-1} + (W_i^{\text{o}} z_i + b_i^{\text{o}}) \text{ with } y_0 = \mathbf{0}; \tag{5.3}$$

where $\otimes$ denotes the Hadamard (element-wise) product, $(W^{\text{h}}, b^{\text{h}})$ parametrize the hidden linear layers, and $(W^{\text{o}}, b^{\text{o}})$ represent the output linear layers. This formulation leverages a useful property that repeated Hadamard product of sines are equivalent to a sum of sines with varying amplitude, frequency, and phase [26]. Therefore, with increasing PE frequencies from shallow to deep layers, we can produce output shapes with increasing amount of high-frequency details. This allows us to perform instance-specific optimization while preserving the common base shapes by sharing the shallow layers of part MLPs across instances and optimizing the deep layers to be instance-specific as shown in Fig. 5.5. Note that our MLP network is inspired from band-limited coordinate networks (BACON) [75]. However, we add the cumulative sum of all previous outputs to the current output of each layer since the high-frequencies details should be deformations from low-frequency based shapes. Moreover, unlike BACON, that learns the frequencies of sine functions in PE together with other parameters, we pre-define and fix the input frequencies to improve the optimization stability in our ill-posed problem and better control the separation of shared and instance components.

Zoom in to parts for higher details. Similar to LASSIE, we perform analysis-by-synthesis to supervise the overall shape reconstruction since we do not have access to any form of 3D supervision. We render the part surfaces via a differentiable renderer [76] and compare them with the pseudo ground-truth masks obtained from DINO feature clustering. The silhouette loss $\mathcal{L}_{sil}$ is written as: $\mathcal{L}_{sil} = \sum_j \|M^j - \hat{M}^j\|^2$, where M^j and $\hat{M}^j$ are the rendered silhouette and pseudo ground-truth of instance j, respectively. To capture more shape details in the images, we propose to render and compare the high-resolution silhouettes by zooming in on individual parts during optimization. Concretely, we crop the 2D

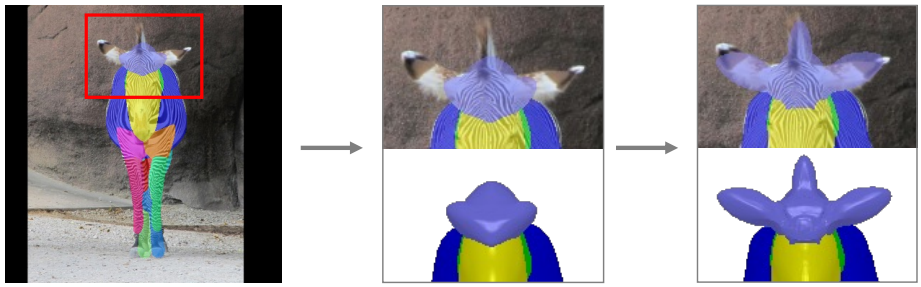

Figure 5.6: **High-resolution rendering and optimization by zooming in on parts.** Based on the initial estimates of 2D part localization, we crop and upsample each part region to perform shape optimization at higher resolution.

part masks estimated by Hi-LASSIE, upsample the cropped regions to higher resolution, and calculate the part silhouette loss $\mathcal{L}_{part}$ on the zoomed-in part masks as:

$$\mathcal{L}_{part} = \sum_i \sum_j \left\| {\Gamma_i}^j(M^j) - {\Gamma_i}^j(\hat{M}^j) \right\|^2, \tag{5.4}$$

where ${\Gamma_i}^j$ denotes the crop-and-upsample operation for part i on instance j. We show some example outputs before and after zoomed-in optimization in Fig. 5.6.

2D-3D semantic consistency via feature MLPs. To densely enforce the 2D-3D semantic consistency of part surfaces, we further introduce feature MLPs to improve the semantic loss proposed in LASSIE. For each instance j, the semantic consistency loss is defined as the Chamfer distance between the foreground pixels $\{p|\hat{M}^j(p) = 1\}$ and 3D surface points $\{x \in X\}$ in a high-dimensional space:

$$\mathcal{L}_{sem} = \sum_j \Big(\sum_p \min_x \mathcal{D}(p,x) + \sum_x \min_p \mathcal{D}(p,x) \Big). \tag{5.5}$$

The high-dimensional distance $\mathcal{D}$ is defined as:

$$\mathcal{D}(p,x) = \underbrace{\|\pi^j(x) - p\|^2}_{\text{Geometric distance}} + \alpha \underbrace{\|Q(x) - K^j(p)\|^2}_{\text{Semantic distance}}, \tag{5.6}$$

where α is a scalar weighting for semantic distance, $Q(x)$ denotes the 3D surface features of point x, and $K(p)$ is the 2D image features at pixel p. In effect, $\mathcal{L}_{sem}$ optimizes the

3D surface coordinates such that the aggregated 3D point features would project closer to the similar pixel features in the image ensemble. In LASSIE, the 3D surface features are maintained and updated for only a sparse set of surface points, which limits the capabilities of $\mathcal{L}_{sem}$ to supervise part localization. Instead, we represent the 3D surface feature of each part with an MLP (Fig 5.4), which is similar to the part shape MLPs shared by all instances. Consequently, we can obtain the semantic features $Q(x)$ given an arbitrary surface coordinate x. We update the part surface and feature MLPs alternatively in an EM-style optimization. That is, we update the feature MLPs by sampling 3D surface points and projecting them onto 2D images in the E-step. In the M-step, we use the updated features to optimize 3D surface MLPs via minimizing the semantic consistency loss.

Pose and shape regularizations. To constrain the output articulations and part shapes, we apply the following pose and shape regularizations. First, we impose a part rotation loss $\mathcal{L}_{rot}$ to limit the angle offsets from resting pose as: $\mathcal{L}_{rot} = \sum_j \|R^j - \bar{R}\|^2$, where R^j is the part rotations of instance j and $\bar{R}$ denotes the part rotations of shared resting pose. Moreover, we remove the side-way rotation constraint on animal legs in LASSIE since the leg parts are not specified in our self-discovered skeletons. Instead, we propose a more general regularization based on 3D symmetry prior. We define the symmetry loss $\mathcal{L}_{sym}$ on the 3D joints to prevent overlapping parts or irregular poses as: $\mathcal{L}_{sym} = \sum_i \|J_i - \Psi(J_i{}^\star)\|^2$ where $J_i{}^\star$ is the symmetric joint of J_i and Ψ is the reflection operation w.r.t. the symmetry plane. Finally, to encourage smooth part surfaces, we apply common mesh regularizations like Laplacian loss $\mathcal{L}_{lap}$ and surface normal loss $\mathcal{L}_{norm}$. $\mathcal{L}_{lap}$ encourages smooth 3D surfaces by pulling each vertex towards the center of its neighbors, and $\mathcal{L}_{norm}$ enforces neighboring mesh faces to have similar normal vectors. Note that all the pose and shape regularizations are generic and applicable to a wide range of articulated shapes.

Optimization and texture sampling. The overall optimization objective is given by the weighted sum of aforementioned losses ($\mathcal{L}_{sil}$, $\mathcal{L}_{part}$, $\mathcal{L}_{sem}$, $\mathcal{L}_{rot}$, $\mathcal{L}_{sym}$, $\mathcal{L}_{lap}$, $\mathcal{L}_{norm}$). We optimize the camera, pose, and shape parameters in a multi-stage manner. Superficially, we first estimate the camera viewpoints and fix the rest. Then, we optimize the part trans-

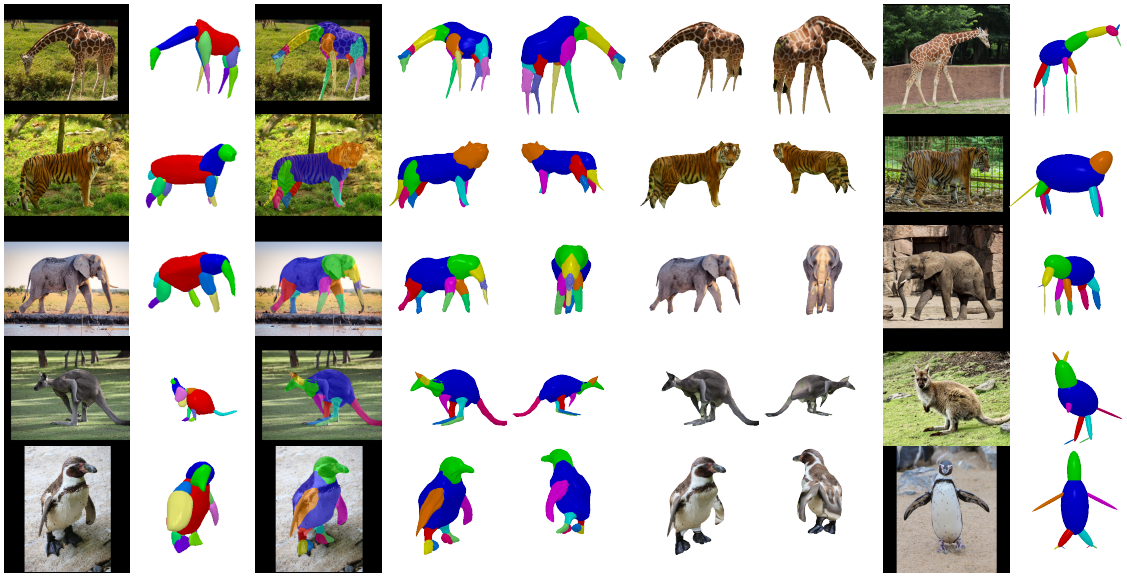

Figure 5.7: **Qualitative results on in-the-wild images.** We show example results of prior arts and Hi-LASSIE on the LASSIE [141] animal image ensembles as well as the reference image and 3D skeleton discovered by Hi-LASSIE. The results demonstrate the effectiveness of our 3D skeleton discovery and the high-fidelity shape/texture reconstruction across diverse animal classes.

formations and shared part MLPs along with cameras until convergence. Finally, we freeze shared part MLPs (shallow layers) and fine-tune the instance-specific part deformations (deep layers) on each instance individually. Note that the surface feature MLPs are also updated during all optimization stages in an EM-style. The final textured outputs are generated by densely sampling the colors of visible surface points from individual images. The invisible (self-occluded) surfaces, on the other hand, are textured by their symmetric surfaces or nearest visible neighbors.

5.5 Experiments

Datasets and baselines. We follow the same evaluation protocols as in LASSIE [141].

That is, we optimize and evaluate Hi-LASSIE on individual image ensembles in the Pascal-Part [12] and LASSIE [141] datasets. From the Pascal-Part dataset, we select the same set of images of horse, cow, and sheep as in LASSIE. These image are annotated with 2D part segmentation masks, which we use to automatically find 2D keypoints for evaluation. The LASSIE dataset contains sparse image ensembles (with CC-licensed web images) of some diverse animals (zebra, tiger, giraffe, elephant, kangaroo, and penguin) with 2D keypoints annotations for evaluation. Considering the novelty of this problem setting (sparse image optimization for articulated animal shapes), we mainly compare Hi-LASSIE with LASSIE as well as some learning-based mesh reconstruction methods. Most recent mesh reconstruction methods either cannot handle articulations [62, 69, 33, 122, 145] or leverage different inputs [68, 136, 138]. 3D Safari [154] and A-CSM [61], on the other hand, are more comparable to Hi-LASSIE since they explicitly model articulations for animal classes of our interest. Since both 3D Safari and A-CSM require large-scale image sets for training (not available in our setting), we use their released models of the closest animal classes to evaluate on our datasets. For instance, we use the zebra model of 3D Safari and the horse model of A-CSM to evaluate on similar quadrupeds.

Visual comparisons. Fig. 5.7 shows the qualitative results of Hi-LASSIE and prior methods on LASSIE dataset images. LASSIE results can fit the object silhouettes quite well but lack shape details. Our results demonstrate that Hi-LASSIE can effectively discover a good 3D skeleton that well explains other instance in the image ensemble. Moreover, the output articulated parts are detailed, high-fidelity, and faithful to input images.

Keypoint transfer. Without ground-truth 3D annotations in our datasets, we follow a common practice [154, 61] to evaluate 3D reconstruction by transferring 2D keypoints from source to target images. That is, we map a set of 2D keypoints on a source image onto the canonical 3D part surfaces, and project them to a target image via the estimated camera, pose, and shape. Since the keypoints are transferred from 2D-to-3D and from 3D-to-2D, a successful transfer indicates accurate 3D reconstruction on both the source and target images. In Table 5.1, we report the percentage of correct keypoints (PCK) under a

Table 5.1: **Keypoint transfer evaluations on the Pascal-Part [12] and LASSIE [141] image ensembles.** For all pairs of images in each animal class, we report the average percentage of correct keypoints (PCK@0.05).

Method	Horse	Cow	Sheep	Zebra	Tiger	Giraffe	Elephant	Kangaroo	Penguin
3D Safari [154]	57.1	50.3	50.5	62.1	50.3	32.5	29.9	20.7	28.9
A-CSM [61]	55.3	60.5	54.7	60.3	55.7	52.2	39.5	26.9	33.0
LASSIE [141]	58.0	62.4	55.5	63.3	62.4	60.5	40.3	31.5	40.6
Hi-LASSIE w/o inst. MLPs	56.8	57.7	53.6	59.7	63.0	59.9	40.8	31.5	41.2
Hi-LASSIE w/o feat. MLPs	57.5	62.2	**56.3**	64.1	62.0	60.8	**42.7**	30.3	41.5
Hi-LASSIE w/o $\mathcal{L}_{part}$	58.8	62.8	55.9	63.8	62.8	61.1	41.5	33.3	42.5
Hi-LASSIE	**59.6**	**63.1**	56.2	**64.2**	**63.1**	**61.6**	**42.7**	**35.0**	**44.4**

Table 5.2: **Quantitative evaluations on the Pascal-part images.** We report the overall 2D IOU, part mask IOU, and percentage of correct pixels (PCP) under dense part mask transfer between all source-target image pairs.

	Overall IOU			Part IOU			Part transfer (PCP)		
Method	Horse	Cow	Sheep	Horse	Cow	Sheep	Horse	Cow	Sheep
SCOPS [44]	62.9	67.7	63.2	23.0	19.1	26.8	-	-	-
DINO clustering [2]	81.3	85.1	83.9	26.3	21.8	30.8	-	-	-
3D Safari [154]	72.2	71.3	70.8	-	-	-	71.7	69.0	69.3
A-CSM [61]	72.5	73.4	71.9	-	-	-	73.8	71.1	72.5
LASSIE	81.9	87.1	85.5	38.2	35.1	**43.7**	78.5	77.0	74.3
Hi-LASSIE w/o inst. MLPs	80.4	80.0	79.5	30.2	29.3	33.8	76.4	74.9	71.1
Hi-LASSIE w/o feat. MLPs	82.5	87.6	85.9	34.9	32.4	39.7	74.7	72.4	74.9
Hi-LASSIE w/o $\mathcal{L}_{part}$	81.6	84.7	83.8	38.6	35.2	43.6	79.2	77.4	72.0
Hi-LASSIE	**83.4**	**88.1**	**86.3**	**39.0**	**35.3**	43.4	**79.9**	**77.8**	**75.5**

tight threshold $0.05\times\max(h, w)$, where h and w are image height and width, respectively. The results show that Hi-LASSIE achieves higher PCK on most animal image ensembles compared to the baselines while requiring minimal user inputs. We also show ablative results of Hi-LASSIE without the instance part MLPs (frequency decomposition), feature MLPs, or zoomed-in part silhouette loss $\mathcal{L}_{part}$. All the proposed modules can effectively increase the accuracy of keypoint transfer demonstrating their use.

2D overall/part IOU. In addition to keypoint transfer accuracy, we compare Hi-LASSIE with the baselines using different segmentation metrics (Overall/Part IOU) in Table 5.2. For Hi-LASSIE and prior 2D co-part segmentation methods like SCOPS [44] and DINO clustering [2], we manually assign each discovered part to the best matched part in the Pascal-part annotations. Hi-LASSIE can produce accurate overall and part masks in 2D by learning high-fidelity 3D shapes. Compared to prior methods, Hi-LASSIE outputs match the Pascal-part segmentation better and achieves consistently higher overall IOU.

Part transfer. Finally, we show the results of part transfer accuracy in Table 5.2 using the percentage of correct pixels (PCP) metric proposed in LASSIE [141]. The PCP metric is designed similarly as PCK for keypoint transfer, but it uses 2D part segmentations to more densely evaluate 3D reconstruction. In short, we densely transfer the part segmentation from source to target images through mapping 2D pixels and 3D canonical surfaces. A correct transfer is done when a pixel is mapped to the same 2D part in both the source and target images. The PCP results further demonstrate the favorable performance of Hi-LASSIE against prior arts.

Applications. Hi-LASSIE not only produces high-fidelity 3D shapes but also enables various part-based applications due to explicit skeleton and part-based representation. For instance, we can easily transfer/interpolate the 3D skeleton transformations to repose or animate the output 3D shapes. Likewise, we can transfer the surface texture or part deformation from one animal species/instance to another. Due to their explicit nature, Hi-LASSIE 3D shapes can also be used by graphics artists for downstream applications in AR/VR/games.

5.6 Conclusion

We propose Hi-LASSIE, a technique for 3D articulated shape reconstruction from sparse image ensemble without using any 2D/3D annotations or templates. Hi-LASSIE automatically discovers 3D skeleton based on a single reference image from the input en-

semble. We further design several optimization strategies to reconstruct high-resolution and instance-varying details of 3D part shapes across the given ensemble. Our results on Pascal-Part and LASSIE image ensembles demonstrate the favorable reconstructions of Hi-LASSIE against prior arts despite using minimal user annotations. In future, we hope to apply Hi-LASSIE on more general articulated objects in-the-wild.

Chapter 6

ARTIC3D: Learning Robust Articulated 3D Shapes from Noisy Web Image Collections

6.1 Overview

Estimating 3D articulated shapes like animal bodies from monocular images is inherently challenging due to the ambiguities of camera viewpoint, pose, texture, lighting, etc. We propose ARTIC3D [143], a self-supervised framework to reconstruct per-instance 3D shapes from a sparse image collection in-the-wild. Specifically, ARTIC3D is built upon a skeleton-based surface representation and is further guided by 2D diffusion priors from Stable Diffusion. First, we enhance the input images with occlusions/truncation via 2D diffusion to obtain cleaner mask estimates and semantic features. Second, we perform diffusion-guided 3D optimization to estimate shape and texture that are of high-fidelity and faithful to input images. We also propose a novel technique to calculate more stable image-level gradients via diffusion models compared to existing alternatives. Finally, we produce realistic animations by fine-tuning the rendered shape and texture under rigid

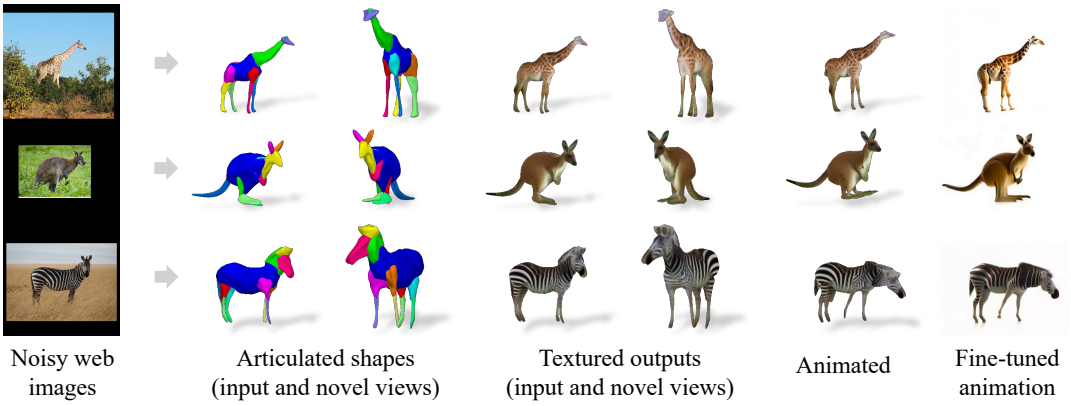

Figure 6.1: **Learning articulated 3D shapes from noisy web images.** We propose ARTIC3D, a diffusion-guided optimization framework to estimate the 3D shape and texture of articulated animal bodies from sparse and noisy image in-the-wild. Results show that ARTIC3D outputs are detailed, animatable, and robust to occlusions or truncation.

part transformations. Extensive evaluations on multiple existing datasets as well as newly introduced noisy web image collections with occlusions and truncation demonstrate that ARTIC3D outputs are more robust to noisy images, higher quality in terms of shape and texture details, and more realistic when animated.

6.2 Introduction

Articulated 3D animal shapes are widely used in applications such as AR/VR, gaming, and content creation. However, the articulated models are usually hard to obtain as manually creating them is labor intensive and 3D scanning real animals in the lab settings is highly infeasible. In this chapter, we aim to automatically estimate high-quality 3D articulated animal shapes directly from sparse and noisy web image collections. This is a highly ill-posed problem due to the variations across images with diverse backgrounds, lighting, camera viewpoints, animal poses, shapes, and textures, etc. In addition, we do not assume access to any 3D shape models or per-image annotations like keypoints and camera viewpoints in our in-the-wild setting.

While several recent methods [141, 131, 142] can produce animatable 3D shapes using a skeleton-based neural surface or pre-defined mesh template, the success is largely dependent on large-scale image datasets or manually-filtered clean images for training or optimization. Moreover, the output 3D shapes and textures are usually unrealistic when viewed from novel viewpoints or pose articulations. On the other hand, recent success of generative diffusion models [105, 112, 109] shows that one can generate high-quality images for a given text prompt. Several recent works [101, 71, 84, 103] further demonstrate the possibility to produce 3D objects/scenes simply using 2D diffusion as multi-view supervision. In this chapter, we leverage the powerful 2D diffusion prior to learn 3D articulated shapes, aiming to reconstruct and animate 3D animals from sparse noisy online images without any 2D or 3D annotations. Intuitively, one can improve the quality of 3D reconstructions by utilizing a diffusion prior similar to the score distillation sampling (SDS) loss proposed in DreamFusion [101]. In our experiments, nonetheless, we observe that naively applying the SDS loss on 3D surface optimization leads to unstable and inefficient training, producing undesirable artifacts like noisy surfaces or ambiguous texture.

In this chapter, we present ARTIC3D (ARTiculated Image Collections in 3D), a diffusion-guided optimization framework to learn articulated 3D shapes from sparse noisy image collections. We use the articulated part surface and skeleton from Hi-LASSIE [142], which allows explicit part manipulation and animation. We propose a novel Decoder-based Accumulative Score Sampling (DASS) module that can effectively leverage 2D diffusion model priors from Stable Diffusion [109] for 3D optimization. In contrast to existing works that back-propagate image gradients through the latent encoder, we propose a decoder-based multi-step strategy in DASS, which we find to provide more stable gradients for 3D optimization. To deal with noisy input images, we propose an input preprocessing scheme that use diffusion model to reason about occluded or truncated regions. In addition, we also propose techniques to create realistic animations from pose articulations.

We analyze ARTIC3D on the Pascal-Part [12] and LASSIE [141] datasets. To better demonstrate the robustness to noisy images, we extend LASSIE animal dataset [141] with

noisy web animal images where animals are occluded and truncated. Both qualitative and quantitative results show that ARTIC3D produces 3D shapes and textures that are detailed, faithful to input images, and robust to partial observations. Moreover, our 3D articulated representation enables explicit pose transfer and realistic animation which are not feasible for prior diffusion-guided methods with neural volumetric representations. Fig. 6.1 shows sample 3D reconstructions and applications from ARTIC3D. The main contributions of this work are:

- We propose a diffusion-guided optimization framework called ARTIC3D, where we reconstruct 3D articulated shapes and textures from sparse noisy online images without using any pre-defined shape templates or per-image annotations like camera viewpoint or keypoints.
- We design several strategies to efficiently incorporate 2D diffusion priors in 3D surface optimization, including input preprocessing, decoding diffused latents as image targets, pose exploration, and animation fine-tuning.
- We introduce E-LASSIE, an extended LASSIE dataset [141], by collecting and annotating noisy web images with occlusions or truncation to evaluate model robustness. Both qualitative and quantitative results show that ARTIC3D outputs have higher-fidelity compared to prior arts in terms of 3D shape details, texture, and animation.

6.3 Related Work

Animal shape and pose estimation. Earlier techniques on animal shape estimation used statistical body models [156, 154] that are learned either using animal figurines or a large number of annotated animal images. Some other works [137, 136, 138, 139], use video inputs to learn articulated shapes by exploiting dense correspondence information in video. However, these methods rely on optical flow correspondences between video frames, which are not available in our problem setting. Other techniques [62, 61] leverage a parametric mesh model and learn a linear blend skinning from images to obtain a posed

mesh for different animal categories. Most related to our work are LASSIE [141] and Hi-LASSIE [142] which tackle the same problem setting of recovering 3D shape and texture from a sparse collection of animal images in the wild using either a manually annotated skeleton template, or by discovering category specific template from image collections. MagicPony [131] learns a hybrid 3D representation of the animal instance from category specific image collections. However, these approaches require carefully curated input data and fail to handle image collections with partial occlusions, truncation or noise. By leveraging recent advances in diffusion models, we support reconstruction on a wider variety of input images.

3D reconstruction from sparse images. Several recent works [120, 148, 146, 130, 104, 6, 149, 7] have used implicit representations [85] to learn geometry and appearance from sparse image collections either by training in a category specific manner or assuming access to multi-view consistent sparse images during inference. However, most of these approaches demonstrate compelling results only on rigid objects. Zhang et al. [148] is another closely related work that finds a neural surface representation from sparse image collections but requires coarse camera initialization. By learning a part based mesh shape and texture, our framework naturally lends itself to modeling and animating articulated categories such as animals in the wild without any additional requirements on camera parameters.

Diffusion prior for 3D. Diffusion models [109, 112, 150] have recently gained popularity for generating high resolution images guided by various kinds of conditioning inputs. Diffusion models capture the distribution of real data which can be used as score function to guide 3D generation with score-distillation sampling (SDS) loss as first described in DreamFusion [101]. Several recent approaches [84, 71, 82, 117, 107, 103] leverage the SDS loss to generate 3D representations from either text or single or sparse image collections. Drawing inspiration from these lines of work, we propose a novel Decoder-based accumulative Score Sampling (DASS) that exploits the high quality images synthesized by the decoder and demonstrate improved performance over naive SDS loss.

6.4 Approach

Given 10-30 noisy web images of an animal species, ARTIC3D first preprocesses the images via 2D diffusion to obtain cleaner silhouette estimates, semantic features, and 3D skeleton initialization. We then jointly optimizes the camera viewpoint, pose articulation, part shapes and texture for each instance. Finally, we animate the 3D shapes with rigid bone transformations followed by diffusion-guided fine-tuning. Before introducing our diffusion-based strategies to improve the quality of 3D outputs, we briefly review the skeleton-based surface representation similar to [141, 142], as well as Stable Diffusion [109] that we use as diffusion prior.

6.4.1 Preliminaries

While most 3D generation methods optimizes a volumetric neural field to represent 3D rigid objects/scenes, we aim to produce 3D shapes that are articulated and animatable. To enable explicit part manipulation and realistic animation, we adopt a skeleton-based surface representation as in LASSIE [141] and Hi-LASSIE [142]. Unlike [141, 142] which directly sample surface texture from images, we optimize per-part texture images to obtain realistic instance textures from novel views.

3D Skeleton. Given a user-specified reference image in the collection, Hi-LASSIE [142] automatically discovers a 3D skeleton based on the geometric and semantic cues from DINO-ViT [9] feature clusters. The skeleton initializes a set of 3D joints and primitive part shapes, providing a good constraint of part transformation and connectivity. In our framework, we obtain cleaner feature clusters by diffusing input images, then applying Hi-LASSIE as an off-the-shelf skeleton discovery method. For a fair comparison with existing works, we use the same reference image for skeleton discovery as in [142] in our experiments. Please refer to [142] for further details on skeleton discovery.

Neural part surfaces. Following [142], using the discovered 3D skeleton, we reconstruct a 3D part corresponding to each skeleton bone via a deformable neural surface [148]. The

neural surfaces are parameterized by multi-layer perceptron networks (MLPs), mapping 3D surface points on a unit sphere to their xyz deformation. Given m uniformly sampled 3D points $X \in \mathbb{R}^{3\times m}$ on a spherical surface, we can deform the 3D shape of the i-th part through the part MLP as $X \mapsto \mathcal{F}_i(X)$. Then, the part surfaces are rigidly transformed by the scaling $s_i \in \mathbb{R}$, rotation $R_i \in \mathbb{R}^{3\times 3}$, and translation $t_i \in \mathbb{R}^3$ of each skeleton part i. The transformed part surface points V_i in the global coordinate can be written as: $V_i = s_i R_i \mathcal{F}_i(X) + t_i$. Please refer to [142] for further details.

Stable Diffusion architecture. Stable Diffusion (SD) [109] is a state-of-the-art text-to-image generative model that can synthesize high-quality images given a text prompt. SD mainly consists of 3 components: An image encoder $\mathcal{E}$ that encodes a given image x into a latent code z; a decoder network $\mathcal{D}$ that converts the latent code back to image pixels; and a U-Net denoiser ϵ_ϕ that can iteratively denoise a noisy latent code. We use SD as a diffusion prior in our framework.

6.4.2 Decoder-based Accumulative Score Sampling (DASS)

To leverage the 2D diffusion prior for 3D shape learning, DreamFusion [101] proposes a score distillation sampling (SDS) loss to distill the images rendered from random views and propagate the image-level gradients to Neural Radiance Field (NeRF) parameters. To reduce the computational cost and improve training stability, recent works like Latent-NeRF [84] and Magic3D [71] perform distillation on the low-resolution latent codes in SD and back-propagate the gradients through the SD image encoder $\mathcal{E}$. Formally, let x be a rendered image from 3D model and z denote its latent codes from the SD image encoder $\mathcal{E}$. At each score distillation iteration, the latent codes z are noised to a random time step t, denoted as z_t, and denoised by the U-Net denoiser ϵ_ϕ of the diffusion model. The image-level SDS gradients can then be expressed as: $\nabla_x \mathcal{L}_{\text{SDS}} = w_t(\epsilon_\phi(z_t; y, t) - \epsilon)\frac{\partial z}{\partial x}$, where y denotes the guiding text embedding, ϵ is the random noise added to the latent codes, and w_t is a constant multiplier which depends on diffusion timestep t. The denoiser ϵ_ϕ uses

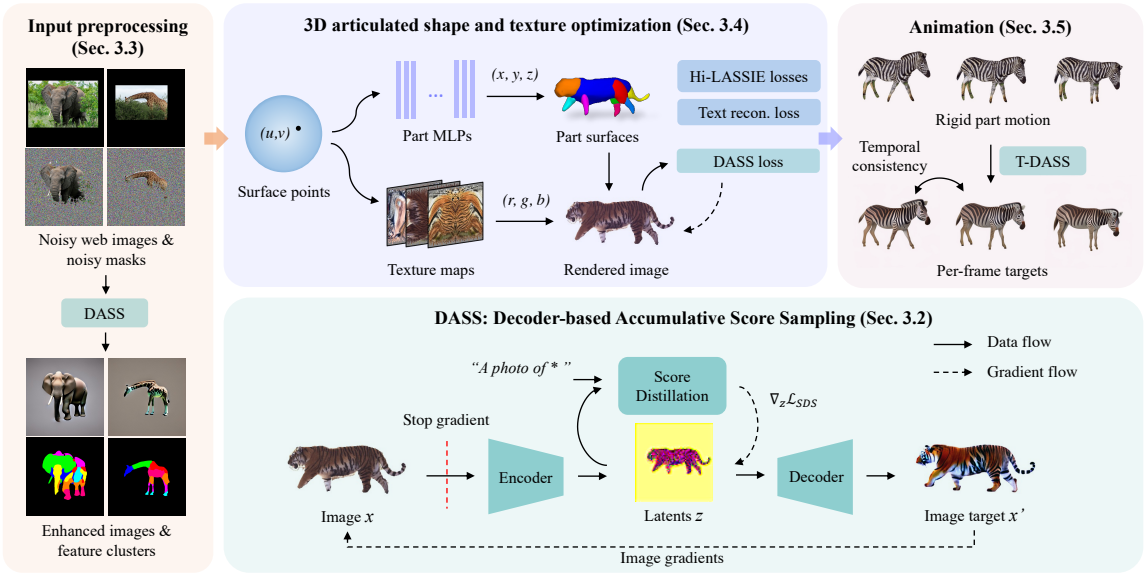

Figure 6.2: **ARTIC3D overview.** Given sparse web images of an animal species, ARTIC3D estimates the camera viewpoint, articulated pose, 3D part shapes, and surface texture for each instance. We propose a novel DASS module to efficiently compute image-level gradients from stable diffusion, which are applied in 1) input preprocessing, 2) shape and texture optimization, and 3) animation.

a guidance scale w_g to balance the text guidance and a classifier-free guidance [43] of an unconditional model.

Although this common SDS loss is effective in generating NeRFs from text, we observe that naively applying it in our framework leads to unstable and inefficient training. As shown in Fig. 6.3 (b), the SDS gradients back-propagated through the encoder are often quite noisy, causing undesirable artifacts on 3D shapes and texture. Moreover, it requires the extra computation and memory usage for gradient back propagation, limiting the training batch size and thus decreasing stability.

To mitigate these issues, we propose a novel Decoder-based Accumulative Score Sampling (DASS) module, an alternative to calculate pixel gradients that are cleaner and more efficient. Fig. 6.2 illustrates the proposed DASS module. At a high level, given an input image x, we obtain a denoised image x' from the decoder as a reconstruction target, based

on our observation that decoded outputs are generally less noisy. As shown in Fig. 6.2, we pass a rendered image through the encoder $\mathcal{E}$ to obtain low-resolution latent codes, update the latents for n steps via score distillation, then decode the updated latents with the decoder $\mathcal{D}$ as an image target. Formally, instead of explicitly calculating the partial derivative $\partial z/\partial x$, we use $x - \mathcal{D}(z - \nabla z)$ as a proxy to ∇x, where ∇z is the accumulated latent gradients over n steps. This makes a linear assumption on $\mathcal{D}$ around latents z, which we empirically find effective to approximate the pixel gradients. The target image $x' = \mathcal{D}(z - \nabla z)$ can be directly used as an updated input (Section 6.4.3) or to compute a pixel-level DASS loss $\mathcal{L}_{dass} = \|(x - \mathcal{D}(z - \nabla z))\|^2$ in 3D optimization (Section 6.4.4). Since the DASS module only involves one forward pass of the encoder and decoder, it costs roughly half the memory consumption during training compared to the original SDS loss. The visualizations in Fig. 6.3 demonstrate that DASS produces cleaner images than the original SDS loss in one training step (b), and that the accumulated gradients can effectively reduce noise and fill in the missing parts (c). Moreover, we show that adding random noise to the background pixels can facilitate the shape completion by DASS (a). We also perform ablative analyses on other diffusion parameters like noise timestep (d) and guidance weight (e) in Fig. 6.3. In general, ARTIC3D favors moderate accumulation steps $n \in (3, 10)$ and lower timestep $t \in (0.2, 0.5)$ since higher variance can lead to 3D results that are not faithful to the input images. Also, we use a lower guidance weight $w_g \in (10, 30)$ so that our results do not suffer from over saturation effects common in prior works due to high guidance scale in SDS loss.

6.4.3 Input preprocessing for noisy images

Animal bodies in real-world images often have ambiguous appearance caused by noisy texture, dim lighting, occlusions, or truncation, as shown in Fig. 6.4. To better deal with noisy or partial observations, we propose a novel method to enhance the image quality and complete the missing parts. Given a sparse image collection $\{I_j \in \mathbb{R}^{H \times W \times 3}\}$ ($j \in$

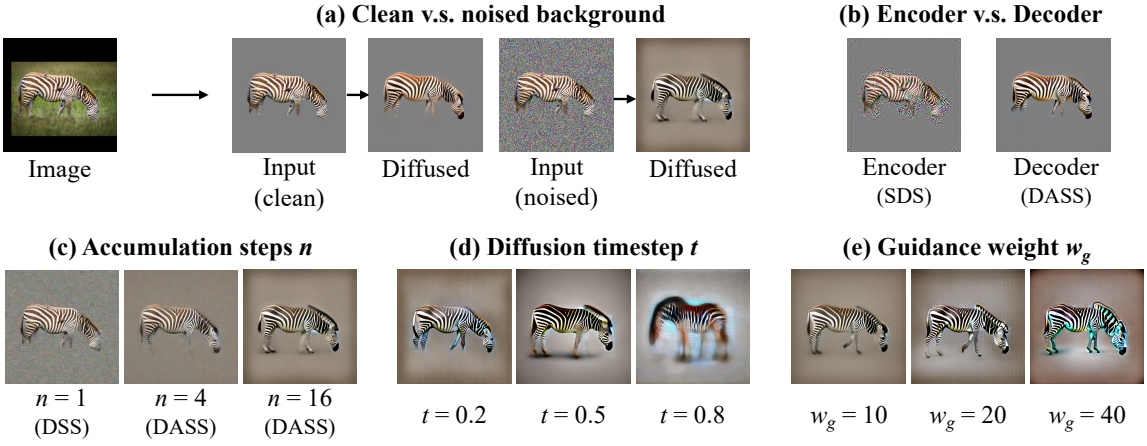

Figure 6.3: **Ablative visualizations of the DASS method.** From the example input image (top left), we show the updated image after one optimization iteration using various ways to obtain image-level gradients or parameter settings: (a) shows that noised background in the input image encourages DASS to hallucinate the missing parts; (b) compares the standard SDS (back-propagate gradients through encoder) and our DASS (decoder-based) losses; (c) justifies our accumulating latent gradient approach as it leads to cleaner decoded output; (d) indicates that small timestep mostly modifies the texture, whereas large timestep changes the geometry more (sometimes removes or creates body parts); (e) demonstrates high-contrast colors and slightly disproportioned body with higher guidance weight (diffusion prior is biased towards larger heads and frontal views). Note that (b) uses the clean input in (a) for better visualization, whereas (c),(d),(e) are obtained from the noised input.

$\{1, ..., N\}$ and N is typically between 10-30) of an animal species, we aim to obtain accurate silhouettes estimates $\{\hat{M}_j \in \mathbb{R}^{H \times W}\}$ and clean semantic features $\{K_j \in \mathbb{R}^{h \times w \times d}\}$ for each instance. As shown in Fig. 6.2, we roughly estimate the foreground masks via clustering salient features extracted by a trained DINO-ViT [9] network. Then, we apply DASS to diffuse the background-masked images, resulting in animal bodies with cleaner texture and complete shapes. Formally, we obtain an updated image I' by $\mathcal{D}(z - \nabla z)$, where $z = \mathcal{E}(I)$. Here, DASS serves as an image denoising and inpainting module, which can effectively generate a high-quality version of a noisy input via n latent updates and a single forward pass of $\mathcal{D}$. Following the noise-and-denoise nature of diffusion models, we

show in Fig. 6.3 (a) that manually adding Gaussian noise to the background pixels in an input image encourages DASS to hallucinate the occluded parts while mostly preserving the visible regions. Finally, we re-apply DINO-ViT feature extraction and clustering [2] on the diffused images to obtain cleaner and more complete masks as well as semantic features. Fig. 6.2 (left) shows sample noisy input images and the corresponding output enhanced images and feature clusters. Note that Farm3D [47] uses SD [109] to generate animal images from text for 3D training, which, however, often contain irregular shapes (*e.g.*, horses with 5 legs). On the contrary, our preprocessed images are more suitable for the sparse-image optimization framework since our goal is to reconstruct 3D shape and texture that are realistic and faithful to the input images.

6.4.4 Diffusion-guided optimization of shape and texture

Given the preprocessed images, silhouette estimates, and semantic features, we jointly optimize the camera viewpoint, pose articulation, 3D part shapes, and texture. Since we do not assume any 2D or 3D annotations, we follow Hi-LASSIE [142] and adopt an analysis-by-synthesis approach to reconstruct 3D shape and texture that are faithful to the input images. That is, we render the 3D part using a differentiable renderer [76] and compare them with the 2D images, pseudo ground-truth silhouettes, and DINO-ViT features. Fig. 6.2 (top) illustrates the shape and texture optimization.

LASSIE and Hi-LASSIE losses. Given the rendered silhouette $\tilde{M}^j$ and pseudo ground-truth $\hat{M}^j$ of instance j, the silhouette loss $\mathcal{L}_{sil}$ can be written as: $\mathcal{L}_{sil} = \sum_j \|\tilde{M}^j - \hat{M}^j\|^2$. LASSIE [141] and Hi-LASSIE [142] further leverage the 2D correspondence of DINO features between images of the same animal class to define a semantic consistency loss $\mathcal{L}_{sem}$. $\mathcal{L}_{sem}$ can be interpreted as the Chamfer distance between 3D surface points and 2D pixels, enforcing the aggregated 3D point features to project closer to the similar pixel features in all images. To regularize the pose articulations and part shapes, [141, 142] also apply a part rotation loss $\mathcal{L}_{rot}$, Laplacian mesh regularization $\mathcal{L}_{lap}$, and surface normal

loss $\mathcal{L}_{norm}$. The part rotation loss $\mathcal{L}_{rot} = \sum_j \|R^j - \bar{R}\|^2$ limits the angle offsets from resting pose, where R^j is the part rotations of instance j and $\bar{R}$ denotes the part rotations of shared resting pose. $\mathcal{L}_{lap}$ and $\mathcal{L}_{norm}$ encourage smooth 3D surfaces by pulling each vertex towards the center of its neighbors and enforcing neighboring faces to have similar normals, respectively. We omit the details and refer the readers to [141, 142]. Considering that the reconstruction ($\mathcal{L}_{sil}$, $\mathcal{L}_{sem}$) and regularization ($\mathcal{L}_{rot}$, $\mathcal{L}_{lap}$, $\mathcal{L}_{norm}$) losses are generic and effective on articulated shapes, we use them in ARTIC3D along with novel texture reconstruction and DASS modules.

Texture reconstruction. Both [141, 142] directly sample texture from input RGB, resulting in unrealistic textures in occluded regions. To obtain more realistic textures, we also optimize a texture image T_i for each part. The vertex colors $C \in \mathbb{R}^{3\times m}$ are sampled via the pre-defined UV mapping $\mathcal{S}$ of surface points X. Formally, the surface color sampling of part i can be expressed as $C_i = T_i(\mathcal{S}(X))$. The sampled surface texture are then symmetrized according to the symmetry plane defined in the 3D skeleton. Note that the texture images are optimized per instance since the animals in web images can have diverse texture. Similar to the $\mathcal{L}_{lap}$, we enforce the surface texture to be close to input image when rendered from the estimated input view. The texture reconstruction loss is defined as: $\mathcal{L}_{text} = \sum_j \|\hat{M}^j \odot (\hat{I}^j - \tilde{I}^j)\|^2$ where $\hat{I}^j$ denotes the clean input image of instance j after input preprocessing and $\hat{M}^j$ denotes the corresponding animal mask; $\tilde{I}^j$ is the rendered RGB image from the estimated 3D shape and texture; and $\odot$ denotes element-wise product. The reconstruction loss is masked by the estimated foreground silhouette so that the surface texture optimization is only effected by the visible non-occluded animal pixels.

Distilling 3D reconstruction. In addition to the aforementioned losses, we propose to increase the shape and texture details by distilling 3D reconstruction. Here, we use DASS as a critic to evaluate how well a 3D reconstruction looks in its 2D renders, and calculate pixel gradients from the image target. Similar to prior diffusion-based methods [101, 84, 71], we render the 3D surfaces with random viewpoints, lighting, and background colors during training. Moreover, we design a pose exploration scheme to densify the articulation

space in our sparse-image scenario. In particular, we randomly interpolate the estimated bone rotation (R^{j_1}, R^{j_2}) of two instances (j_1, j_2), and generate a new instance with novel pose $R' = \alpha R^{j_1} + (1-\alpha)R^{j_2}$ for rendering, where $\alpha \in (0, 1)$ is a random scalar. As such, we can better constrain the part deformation by diffusion prior and prevent irregular shape or disconnection between parts. As shown in Fig. 6.2, we then diffuse the latent codes of rendered images and obtain pixel gradients from the DASS module. The resulting gradients are back-propagated to update the part surface texture, deformation MLP, bone transformation, and camera viewpoints. In our experiments, we observe that the RGB gradients do not propagate well through the SoftRas [76] blending function, and we thus modify it with a layered blending approach proposed in [87].

Optimization details. The overall optimization objective can be expressed as the weighted sum of all the losses $\mathcal{L} = \sum_{l \in \mathfrak{L}} \alpha_l \mathcal{L}_l$, where $\mathfrak{L} = \{sil, sem, rot, lap, norm, text, dass\}$ as described above. We optimize the shared and instance-specific shapes in two stages. That is, we first update the shared part MLPs along with camera viewpoints and pose parameters. Then, we fine-tune the instance-specific part MLPs and optimize texture images for each instance. All model parameters are updated using an Adam optimizer [53]. We render the images at 512×512 resolution and at 128×128 for the part texture images.

6.4.5 Animation fine-tuning

One can easily animate the resulting 3D articulated animals by gradually rotating the skeleton bones and their corresponding parts surfaces. However, the rigid part transformations often result in disconnected shapes or texture around the joints. To improve the rendered animation in 2D, one can naively use DASS frame-by-frame on a sequence of articulated shapes. However this can produce artifacts like color flickering and shape inconsistency across the frames. As a remedy, we further propose a fine-tuning step, called Temporal-DASS (T-DASS), to generate high-quality and temporally consistent 2D animations based on the ARTIC3D outputs. Given a sequence of part transformations from

simple interpolation across instances or motion re-targeting, we render the 3D surfaces as video frames $\{J_k \in \mathbb{R}^{H\times W\times 3}(k \in \{1, ..., K\})\}$ and encode them into latent codes $\{z_k \in \mathbb{R}^{h\times w\times 3}\}$ through the SD encoder $\mathcal{E}$. Then, we design a reconstruction loss $\mathcal{L}_{recon}$ and temporal consistency loss $\mathcal{L}_{temp}$ to fine-tune the animation in the latent space. Similar to DASS, we obtain the reconstruction targets $\{z'_k\}$ by accumulating latent SDS gradients ∇z_k for multiple steps: $z'_k = z_k - \nabla z_k$. The reconstruction loss can then be written as: $\mathcal{L}_{recon} = \sum_t \|(z_k - z'_k)\|^2$. To enforce temporal consistency, we exploit our 3D surface outputs and calculate accurate 2D correspondences across neighboring frames. Specifically, for each latent pixel in frame z_k, we find the closest visible 3D surfaces via mesh rasterization, then backtrack their 2D projection in frame z_{k-1}, forming a dense 2D flow field $F_k \in \mathbb{R}^{h\times w\times 2}$. Intuitively, the corresponding pixels should have similar latent codes. Hence, we use F_k to perform temporal warpping on the latent codes z_{k-1}, denoted as: warp(z_{k-1}, F_k), and define $\mathcal{L}_{temp}$ as: $\mathcal{L}_{temp} = \sum_{k=2}^{K} \|(z_k - \text{warp}(z_{k-1}, F_k)\|^2$. We fine-tune the latent codes $\{z_k\}$ with $\mathcal{L}_{recon}$ and $\mathcal{L}_{temp}$, where $\{F_k\}$ are pre-computed and $\{z'_k\}$ are updated in each iteration. Finally, we can simply obtain the RGB video frames by passing the optimized latent codes through the SD decoder $\{\mathcal{D}(z_k)\}$. The proposed $\mathcal{L}_{recon}$ encourages better shape and texture details in each frame, and $\mathcal{L}_{temp}$ can effectively regularize latent updates temporally. Note that T-DASS optimizes the latent codes and takes temporal consistency into account, which is different from DASS which operates on each image individually.

6.5 Experiments

Datasets. Following [141, 142], we evaluate ARTIC3D on the Pascal-Part [12] and LASSIE [141] images. From Pascal-Part, we obtain images of horse, cow, and sheep, as well as their 2D keypoints automatically computed using the ground-truth 2D part masks. The LASSIE dataset includes web images of other animal species (zebra, tiger, giraffe, elephant, kangaroo, and penguin) and 2D keypoint annotations. Each image collection contains roughly

Figure 6.4: **E-LASSIE samples.** We extend LASSIE [141] image sets with 15 occluded or truncated images per animal class and annotate the 2D keypoints for evaluation. These noisy images pose great challenges to sparse-image optimization since the per-instance 3D shapes can easily overfit to the visible parts and ignore the rest.

30 images of different instances with diverse appearances, which are manually filtered so that the animal bodies are fully visible in the images. To evaluate the model robustness in a more practical setting, we extend the LASSIE image sets with several noisy images where the animals are occluded or truncated. In particular, we collect 15 additional web images (CC-licensed) per class and annotate the 2D keypoints for evaluation. We call the extended image sets E-LASSIE and show some examples in Fig. 6.4. For the experiments on E-LASSIE, we optimize and evaluate on all the 45 images in each set.

Baselines. We mainly compare ARTIC3D with LASSIE [141] and Hi-LASSIE [142] as we deal with the same problem setting, namely sparse image optimization for articulated animal shapes. For reference, we also compare the results with several learning-based methods like A-CSM [61], MagicPony [71], and Farm3D [47]. Note that these approaches are not directly comparable to ARTIC3D since they train a feedforward network on large-scale image sets (not available in our scenario). Although related, some other recent works on 3D surface reconstruction either cannot handle articulations [62, 69, 33, 122, 145] or require different inputs [68, 136, 138]. As a stronger baseline, we implement Hi-LASSIE+, incorporating the standard SDS loss as in [109, 71, 84] (back-propagate latent gradients through encoder) during Hi-LASSIE [142] optimization for shape and texture.

Evaluation metrics. Considering the lack of ground-truth 3D annotations in our datasets, we follow a common practice [154, 61, 141, 142] to use keypoint transfer accuracy as a quantitative metric to evaluate 3D reconstruction. For each pair of images, we map the annotated 2D keypoints on source image onto the canonical 3D surfaces, re-project them to the target image via the estimated camera, pose, and shape, and compare the transferred

keypoints with target annotations. To further evaluate the quality of textured outputs, we compute CLIP [102] features of the 3D output renders under densely sampled viewpoints, and calculate the feature similarity against text prompt as well as input images. While most prior arts on 3D shape generation [101, 103] only evaluate the image-text similarity, we also evaluate the image-image similarity since our outputs should be faithful to both the category-level textual description as well as instance-specific input images. We use a text prompt: "A photo of *" for each animal class "*" in our experiments. A CLIP ViT-B/32 model is used to compute the average feature similarity over 36 uniformly sampled azimuth renders at a fixed elevation of 30 degrees.

Qualitative results. Fig. 6.1 shows some sample outputs of ARTIC3D. In Fig. 6.5, we compare the visual results of Hi-LASSIE, Hi-LASSIE+, and ARTIC3D on the E-LASSIE images. Both Hi-LASSIE and Hi-LASSIE+ produce irregular pose and shape for the invisible parts. Regarding surface texture, Hi-LASSIE reconstructs faithful texture from the input view but noisy in novel views, since it naively samples vertex colors from the input images. The output texture of Hi-LASSIE+ is generally less noisy thanks to the SDS loss. By comparison, ARTIC3D accurately estimates the camera viewpoint, pose, shape, and texture even with the presence of occlusions or truncation. The ARTIC3D outputs are detailed, faithful to input images, and realistic from both input and novel views.

Quantitative comparisons. We show comparisons of the keypoint transfer accuracy (PCK) in Tables 6.1. On both LASSIE and E-LASSIE image sets, Hi-LASSIE+ produces a marginal PCK gain from Hi-LASSIE [142] by naively applying the SDS loss. ARTIC3D, on the other hand, achieves consistently higher PCK than the baselines, especially on the noisy E-LASSIE images. The results demonstrate that our diffusion-guided strategies can effectively learn more detailed, accurate, and robust 3D shapes. The Pascal-Part results in Tab 6.2 further show that ARTIC3D performs favorably against the state-of-the-art optimization-based methods and are comparable to learning-based approaches. In Table 6.3, we show the CLIP similarity comparisons on the E-LASSIE images, which indicate that our textured outputs are more faithful to both the input images (instance-level)

Figure 6.5: **Visual comparison of ARTIC3D and other baselines.** For each input image, we show the 3D textured outputs from input (upper) and novel (lower) views. The results demonstrate that ARTIC3D is more robust to noisy images with occlusions or truncation, producing 3D shape and texture that are detailed and faithful to the input images.

Table 6.1: **Keypoint transfer evaluations on the LASSIE [141] and E-LASSIE image sets.** We report the average PCK@0.05 (↑) on all pairs of images. ARTIC3D performs favorably against the optimization-based prior arts on all animal classes. The larger performance gap in the E-LASSIE demonstrates that ARTIC3D is robust to noisy images.

Method	Image set	Elephant	Giraffe	Kangaroo	Penguin	Tiger	Zebra
LASSIE [141]	LASSIE	40.3	60.5	31.5	40.6	62.4	63.3
Hi-LASSIE [142]	LASSIE	42.7	61.6	35.0	44.4	63.1	64.2
Hi-LASSIE+	LASSIE	43.3	61.5	35.5	44.6	63.4	64.0
ARTIC3D	LASSIE	**44.1**	**61.9**	**36.7**	**45.3**	**64.0**	**64.8**
Hi-LASSIE [142]	E-LASSIE	37.6	54.3	31.9	41.7	57.4	60.1
Hi-LASSIE+	E-LASSIE	38.3	54.8	32.8	41.8	57.7	61.3
ARTIC3D	E-LASSIE	**39.8**	**58.0**	**35.3**	**43.8**	**59.3**	**63.0**

Table 6.2: **Keypoint transfer results on Pascal-Part [24].** We report the mean PCK@0.1 (↑) on all pairs of images. * indicates learning-based models which are trained on a large-scale image set.

Method	Horse	Cow	Sheep
UMR* [69]	24.4	-	-
A-CSM* [61]	32.9	26.3	28.6
MagicPony* [131]	**42.9**	**42.5**	26.2
Farm3D* [47]	42.5	40.2	**32.8**
LASSIE [141]	42.2	37.5	27.5
Hi-LASSIE [142]	43.7	42.1	29.9
Hi-LASSIE+	43.3	42.3	30.5
ARTIC3D	**44.4**	**43.0**	**31.9**

and text prompt (class-level) for most animal classes.

Animation and texture transfer. In Fig. 6.6, we compare the animations before and af-

Table 6.3: **CLIP similarity (↑) evaluations on the E-LASSIE images.** For each animal class, we calculate cosine similarities $s1/s2$, where $s1$ is the image-image similarity (against masked input image) and $s2$ is the image-text similarity (against text prompt).

Method	Elephant	Giraffe	Kangaroo	Penguin	Tiger	Zebra
Hi-LASSIE [142]	80.0 / 26.3	85.2 / 29.6	77.4 / 25.6	**85.8** / 30.8	79.7 / 25.6	83.8 / 27.4
Hi-LASSIE+	79.0 / 27.7	84.7 / 30.2	78.3 / 29.1	82.9 / 32.3	75.3 / 25.3	81.9 / 27.6
ARTIC3D	**82.6 / 28.4**	**85.3 / 30.7**	**81.6 / 29.9**	85.5 / **33.1**	**80.0 / 27.8**	**84.1 / 29.4**

Figure 6.6: **Animation fine-tuning.** Compared to the original animated outputs via rigid transformation (top), our animation fine-tuning (bottom) effectively improves the shape and texture details, especially around animal joints.

Figure 6.7: **Texture transfer.** Our part surface representation enables applications like pose or texture transfer. Given a source shape and target texture, we show the transferred texture between instances (left) and animal species (right).

ter our fine-tuning step via T-DASS. While the skeleton-based representation allows easy animation via rigid part transformation, the output part shapes and texture are often disconnected and irregular around the joints. The results show that T-DASS can effectively produce high-quality animations that are detailed in shape and texture and temporally consistent between frames. In addition to animation, our 3D part surfaces also enables convenient controllable syntheses like texture transfer and pose transfer between different instance or animal classes. Several examples of texture transfer are shown in Fig. 6.7.

Limitations. ARTIC3D relies on the 3D skeleton discovered by Hi-LASSIE [142] to initialize the parts. If the animal bodies are occluded or truncated in most images, the skeleton initialization tends to be inaccurate, and thus limiting ARTIC3D's ability to form realistic parts. Although our input preprocessing method can mitigate this issue to some extent,

fluffy animals (*e.g.*sheep) with ambiguous skeletal configuration can still pose challenges in skeleton discovery. In addition, the front-facing bias in diffusion models sometimes lead to unrealistic texture like multiple faces, which also affects our reconstruction quality.

6.6 Conclusion

We propose ARTIC3D, a diffusion-guided framework to reconstruct 3D articulated shapes and texture from sparse and noisy web images. Specifically, we design a novel DASS module to efficiently calculate pixel gradients from score distillation for 3D surface optimization and use it in the input preprocessing of noisy images; Shape and texture optimization; as well as the animation fine-tuning. Results on both the existing datasets as well as newly introduced noisy web images demonstrate that ARTIC3D produces more robust, detailed, and realistic reconstructions against prior arts.

Chapter 7

Conclusion and Future Work

7.1 Summary

In this thesis, we introduce several methods to learn the 3D shapes of rigid objects, human bodies, and articulated animal bodies from monocular images using weak supervisory signals (*e.g.*, 3D part/skeleton prior, visibility, semantic image features, 2D diffusion prior, etc) Specifically, we propose:

- LPD [140]: a network that discovers and reconstructs 3D parts of general rigid objects given single-view images for training. Our key idea is to learn a simple shape prior from 3D primitives and use it to regularize the output part shapes. The part-based representation not only improve the overall reconstruction accuracy but enables applications like shape interpolation and generation.

- VisDB [144]: a model that learns robust human body estimation from mixed 2D-3D data. We leverage the canonical surface mapping from DensePose as a weak supervision for 2D vertex localization and dense surface visibility, which is used to resolve truncation and occlusion ambiguities.

- LASSIE [141], Hi-LASSIE [142], and ARTIC3D [143]: a series of optimization

frameworks that estimate the 3D pose, shape, and texture of articulated animal bodies from sparse images in-the-wild. Our main contributions include 1) the new problem setting and datasets of sparse animal images online, 2) a novel skeleton-based part surface representation, 3) several effective optimization strategies using geometric priors and semantic consistency, and 4) incorporating 2D diffusion prior for realistic 3D reconstruction and animation.

7.2 Future work

The five frameworks discussed in this thesis open up the possibilities of learning general articulated shapes from monocular images, which can be further explored in the following future directions.

7.2.1 Generalization to diverse animal and articulated objects

As LASSIE, Hi-LASSIE, and ARTIC3D focus on common animal bodies, it would be of great interest to generalize these methods to more diverse animal classes and articulated objects whose 3D data is hard to obtain. Currently, these methods sometimes suffer from 2D ambiguities caused by noisy semantic features and partially visible skeletons on fluffy animals or thin body parts. To facilitate the generalization, the key next steps include automatic filtering of web images, robust 3D skeleton discovery by jointly considering all images, and obtaining higher-quality dense correspondence between noisy images.

7.2.2 Towards realistic and animatable 3D articulated shapes

Another crucial future goal is to further improve the quality of 3D articulated shapes and animations. To achieve this, one potential research direction is to combine implicit and explicit/model-based representations if we want to generate high-resolution shapes that are easily animatable and controllable. Also, we need to find a better balance between

the realism of shape/texture and faithfulness to the input image, by carefully integrate 3D geometric and 2D generative priors.

Bibliography

[1] Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt, and Marcus Magnor. Tex2shape: Detailed full human body geometry from a single image. In *ICCV*, 2019.

[2] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep ViT features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2021.

[3] Xiang Bai and Longin Jan Latecki. Path similarity skeleton graph matching. *PAMI*, 30(7):1282–1292, 2008.

[4] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24(4):509–522, 2002.

[5] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *ECCV*, pages 561–578, 2016.

[6] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. NeRD: Neural reflectance decomposition from image collections. In *CVPR*, pages 12684–12694, 2021.

[7] Mark Boss, Andreas Engelhardt, Abhishek Kar, Yuanzhen Li, Deqing Sun, Jonathan T Barron, Hendrik Lensch, and Varun Jampani. SAMURAI: Shape and material from unconstrained real-world arbitrary image collections. *NeurIPS*, 2022.

[8] Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan Barron, and Hendrik Lensch. Neural-PIL: Neural pre-integrated lighting for reflectance decomposition. *NeurIPS*, 34, 2021.

[9] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021.

[10] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[11] Wenzheng Chen, Jun Gao, Huan Ling, Edward J Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3D objects with an interpolation-based differentiable renderer. *arXiv preprint arXiv:1908.01210*, 2019.

[12] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *CVPR*, pages 1971–1978, 2014.

[13] Zhiqin Chen, Kangxue Yin, Matthew Fisher, Siddhartha Chaudhuri, and Hao Zhang. BAE-NET: Branched autoencoder for shape co-segmentation. In *ICCV*, pages 8490–8499, 2019.

[14] Hongsuk Choi, Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Beyond static features for temporally consistent 3d human pose and shape from a video. In *CVPR*, pages 1964–1973, 2021.

[15] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *ECCV*, pages 769–787, 2020.

[16] Subhabrata Choudhury, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Unsupervised part discovery from contrastive reconstruction. *NeurIPS*, 34:28104–28118, 2021.

[17] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In *ECCV*, pages 628–644, 2016.

[18] Edo Collins, Radhakrishna Achanta, and Sabine Susstrunk. Deep feature factorization for concept discovery. In *ECCV*, pages 336–352, 2018.

[19] Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey Hinton, and Andrea Tagliasacchi. CvxNet: Learnable convex decomposition. In *CVPR*, pages 31–44, 2020.

[20] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[21] Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. Learning elementary structures for 3D shape generation and matching. *arXiv preprint arXiv:1908.04725*, 2019.

[22] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[23] Sai Kumar Dwivedi, Nikos Athanasiou, Muhammed Kocabas, and Michael J Black. Learning to regress bodies from images using differentiable semantic rendering. In *ICCV*, pages 11250–11259, 2021.

[24] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes (VOC) challenge. *IJCV*, 88(2):303–338, 2010.

[25] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3D object reconstruction from a single image. In *CVPR*, pages 605–613, 2017.

[26] Rizal Fathony, Anit Kumar Sahu, Devin Willmott, and J Zico Kolter. Multiplicative filter networks. In *ICLR*, 2020.

[27] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.

[28] Lin Gao, Jie Yang, Tong Wu, Yu-Jie Yuan, Hongbo Fu, Yu-Kun Lai, and Hao Zhang. SDM-NET: Deep generative network for structured deformable mesh. *TOG*, 38(6):1–15, 2019.

[29] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3D shape. In *CVPR*, pages 4857–4866, 2020.

[30] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *ICCV*, pages 7154–7164, 2019.

[31] Ross Girshick. Fast R-CNN. In *ICCV*, 2015.

[32] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh R-CNN. In *ICCV*, pages 9785–9795, 2019.

[33] Shubham Goel, Angjoo Kanazawa, and Jitendra Malik. Shape and viewpoint without keypoints. In *ECCV*, pages 88–104, 2020.

[34] Riza Alp Guler and Iasonas Kokkinos. Holopose: Holistic 3d human reconstruction in-the-wild. In *CVPR*, pages 10884–10894, 2019.

[35] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *CVPR*, pages 7297–7306, 2018.

[36] Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T Freeman. Unsupervised semantic segmentation by distilling feature correspondences. In *ICLR*, 2022.

[37] Zekun Hao, Hadar Averbuch-Elor, Noah Snavely, and Serge Belongie. DualSDF: Semantic shape manipulation using a two-level representation. In *CVPR*, pages 7631–7641, 2020.

[38] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J Black. Resolving 3d human pose ambiguities with 3d scene constraints. In *ICCV*, pages 2282–2292, 2019.

[39] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017.

[40] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[41] Xingzhe He, Bastian Wandt, and Helge Rhodin. Autolink: Self-supervised learning of human skeletons and object outlines by linking keypoints. *arXiv preprint arXiv:2205.10636*, 2022.

[42] Paul Henderson, Vagia Tsiminaki, and Christoph H Lampert. Leveraging 2D data to learn textured 3D mesh generation. In *CVPR*, pages 7498–7507, 2020.

[43] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[44] Wei-Chih Hung, Varun Jampani, Sifei Liu, Pavlo Molchanov, Ming-Hsuan Yang, and Jan Kautz. SCOPS: Self-supervised co-part segmentation. In *CVPR*, pages 869–878, 2019.

[45] Eldar Insafutdinov and Alexey Dosovitskiy. Unsupervised learning of shape and pose with differentiable point clouds. In *NeurIPS*, pages 2802–2812, 2018.

[46] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3D human sensing in natural environments. *PAMI*, 36(7):1325–1339, 2013.

[47] Tomas Jakab, Ruining Li, Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Farm3d: Learning articulated 3d animals by distilling 2d diffusion. *arXiv preprint arXiv:2304.10535*, 2023.

[48] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, pages 7122–7131, 2018.

[49] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *NeurIPS*, pages 365–376, 2017.

[50] Hiroharu Kato and Tatsuya Harada. Learning view priors for single-view 3D reconstruction. In *CVPR*, pages 9778–9787, 2019.

[51] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3D mesh renderer. In *CVPR*, pages 3907–3916, 2018.

[52] Yuki Kawana, Yusuke Mukuta, and Tatsuya Harada. Neural star domain as primitive representation. *arXiv preprint arXiv:2010.11248*, 2020.

[53] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[54] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[55] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. VIBE: Video inference for human body pose and shape estimation. In *CVPR*, pages 5253–5263, 2020.

[56] Muhammed Kocabas, Chun-Hao P. Huang, Otmar Hilliges, and Michael J. Black. PARE: Part attention regressor for 3D human body estimation. In *ICCV*, pages 11127–11137, 2021.

[57] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3D human pose and shape via model-fitting in the loop. In *ICCV*, pages 2252–2261, 2019.

[58] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *CVPR*, pages 4501–4510, 2019.

[59] Nikos Kolotouros, Georgios Pavlakos, Dinesh Jayaraman, and Kostas Daniilidis. Probabilistic modeling for human mesh recovery. In *ICCV*, 2021.

[60] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *NeurIPS*, pages 1097–1105, 2012.

[61] Nilesh Kulkarni, Abhinav Gupta, David F Fouhey, and Shubham Tulsiani. Articulation-aware canonical surface mapping. In *CVPR*, pages 452–461, 2020.

[62] Nilesh Kulkarni, Abhinav Gupta, and Shubham Tulsiani. Canonical surface mapping via geometric cycle consistency. In *ICCV*, pages 2202–2211, 2019.

[63] K L Navaneet, Priyanka Mandikal, Varun Jampani, and Venkatesh Babu. DIFFER: Moving beyond 3D reconstruction with differentiable feature rendering. In *CVPR Workshops*, pages 18–24, 2019.

[64] Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J Black, and Peter V Gehler. Unite the people: Closing the loop between 3d and 2d human representations. In *CVPR*, pages 6050–6059, 2017.

[65] Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, Nicu Sebe, et al. Motion-supervised co-part segmentation. *arXiv preprint arXiv:2004.03234*, 2020.

[66] Ke Li, Bharath Hariharan, and Jitendra Malik. Iterative instance segmentation. In *CVPR*, pages 3659–3667, 2016.

[67] Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. Synthesizing 3D shapes from silhouette image collections using multi-projection generative adversarial networks. In *CVPR*, pages 5535–5544, 2019.

[68] Xueting Li, Sifei Liu, Shalini De Mello, Kihwan Kim, Xiaolong Wang, Ming-Hsuan Yang, and Jan Kautz. Online adaptation for consistent mesh reconstruction in the wild. *NeurIPS*, 33:15009–15019, 2020.

[69] Xueting Li, Sifei Liu, Kihwan Kim, Shalini De Mello, Varun Jampani, Ming-Hsuan Yang, and Jan Kautz. Self-supervised single-view 3D reconstruction via semantic consistency. In *ECCV*, pages 677–693, 2020.

[70] Yichen Li, Kaichun Mo, Lin Shao, Minhyuk Sung, and Leonidas Guibas. Learning 3D part assembly from a single image. In *ECCV*, pages 664–682, 2020.

[71] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. *arXiv preprint arXiv:2211.10440*, 2022.

[72] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *CVPR*, pages 1954–1963, 2021.

[73] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *ICCV*, 2021.

[74] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, pages 740–755, 2014.

[75] David B Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. Bacon: Band-limited coordinate networks for multiscale scene representation. In *CVPR*, pages 16252–16262, 2022.

[76] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3D reasoning. In *CVPR*, pages 7708–7717, 2019.

[77] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *ECCV*, pages 21–37, 2016.

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

[79] Tiange Luo, Kaichun Mo, Zhiao Huang, Jiarui Xu, Siyu Hu, Liwei Wang, and Hao Su. Learning to group: A bottom-up framework for 3D part discovery in unseen categories. *arXiv preprint arXiv:2002.06478*, 2020.

[80] Priyanka Mandikal, Navaneet KL, and R Venkatesh Babu. 3D-PSRNet: Part segmented 3D point cloud reconstruction from a single image. In *ECCV*, pages 0–0, 2018.

[81] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Srinath Sridhar, Gerard Pons-Moll, and Christian Theobalt. Single-shot multi-person 3D pose estimation from monocular RGB. In *3DV*, pages 120–130, 2018.

[82] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Realfusion: 360 {\deg} reconstruction of any object from a single image. *arXiv preprint arXiv:2302.10663*, 2023.

[83] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *CVPR*, pages 4460–4470, 2019.

[84] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. *arXiv preprint arXiv:2211.07600*, 2022.

[85] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, pages 405–421, 2020.

[86] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *CVPR*, pages 909–918, 2019.

[87] Tom Monnier, Matthew Fisher, Alexei A Efros, and Mathieu Aubry. Share with thy neighbors: Single-view reconstruction by cross-instance consistency. In *ECCV*, pages 285–303, 2022.

[88] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *CVPR*, pages 5079–5088, 2018.

[89] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Camera distance-aware top-down approach for 3d multi-person pose estimation from a single rgb image. In *ICCV*, pages 10133–10142, 2019.

[90] Gyeongsik Moon and Kyoung Mu Lee. I2L-MeshNet: Image-to-lixel prediction network for accurate 3D human pose and mesh estimation from a single RGB image. In *ECCV*, pages 752–768, 2020.

[91] Lea Muller, Ahmed AA Osman, Siyu Tang, Chun-Hao P Huang, and Michael J Black. On self-contact and human pose. In *CVPR*, pages 9990–9999, 2021.

[92] KL Navaneet, Priyanka Mandikal, Mayank Agarwal, and R Venkatesh Babu. CAPNet: Continuous approximation projection for 3D point cloud reconstruction using 2D supervision. In *AAAI*, pages 8819–8826, 2019.

[93] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In *3DV*, pages 484–494, 2018.

[94] Despoina Paschalidou, Luc Van Gool, and Andreas Geiger. Learning unsupervised hierarchical part decomposition of 3D objects from a single RGB image. In *CVPR*, pages 1060–1070, 2020.

[95] Despoina Paschalidou, Angelos Katharopoulos, Andreas Geiger, and Sanja Fidler. Neural parts: Learning expressive 3D shape abstractions with invertible neural networks. In *CVPR*, pages 3204–3215, 2021.

[96] Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. Superquadrics revisited: Learning 3D shape parsing beyond cuboids. In *CVPR*, pages 10344–10353, 2019.

[97] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035, 2019.

[98] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3D hands, face, and body from a single image. In *CVPR*, pages 10975–10985, 2019.

[99] Georgios Pavlakos, Nikos Kolotouros, and Kostas Daniilidis. Texturepose: Supervising human mesh estimation with texture consistency. In *ICCV*, pages 803–812, 2019.

[100] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3d human pose and shape from a single color image. In *CVPR*, pages 459–468, 2018.

[101] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[102] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021.

[103] Amit Raj, Srinivas Kaza, Ben Poole, Michael Niemeyer, Nataniel Ruiz, Ben Mildenhall, Shiran Zada, Kfir Aberman, Michael Rubinstein, Jonathan Barron, et al. Dreambooth3d: Subject-driven text-to-3d generation. *arXiv preprint arXiv:2303.13508*, 2023.

[104] Amit Raj, Michael Zollhofer, Tomas Simon, Jason Saragih, Shunsuke Saito, James Hays, and Stephen Lombardi. Pixel-aligned volumetric avatars. In *CVPR*, pages 11733–11742, 2021.

[105] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, pages 8821–8831, 2021.

[106] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015.

[107] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. *arXiv preprint arXiv:2302.01721*, 2023.

[108] Chris Rockwell and David F Fouhey. Full-body awareness from partial observations. In *ECCV*, pages 522–539, 2020.

[109] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022.

[110] Nadine Rueegg, Silvia Zuffi, Konrad Schindler, and Michael J Black. BARC: Learning to regress 3D dog shape from images by exploiting breed information. *arXiv preprint arXiv:2203.15536*, 2022.

[111] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[112] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, 35:36479–36494, 2022.

[113] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *ICCV*, pages 2304–2314, 2019.

[114] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *CVPR*, pages 84–93, 2020.

[115] Wei Shen, Yuan Jiang, Wenjing Gao, Dan Zeng, and Xinggang Wang. Shape recognition by bag of skeleton-associated contour parts. *Pattern Recognition Letters*, 83:321–329, 2016.

[116] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[117] Uriel Singer, Shelly Sheynin, Adam Polyak, Oron Ashual, Iurii Makarov, Filippos Kokkinos, Naman Goyal, Andrea Vedaldi, Devi Parikh, Justin Johnson, et al. Text-to-4d dynamic scene generation. *arXiv preprint arXiv:2301.11280*, 2023.

[118] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *ECCV*, pages 529–545, 2018.

[119] Yu Sun, Qian Bao, Wu Liu, Yili Fu, Michael J Black, and Tao Mei. Monocular, one-stage, regression of multiple 3d people. In *ICCV*, pages 11179–11188, 2021.

[120] Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari. Sparf: Neural radiance fields from sparse and noisy poses. *arXiv preprint arXiv:2211.11738*, 2022.

[121] Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Multi-view consistency as supervisory signal for learning shape and pose prediction. In *CVPR*, pages 2897–2905, 2018.

[122] Shubham Tulsiani, Nilesh Kulkarni, and Abhinav Gupta. Implicit mesh reconstruction from unannotated image collections. *arXiv preprint arXiv:2007.08504*, 2020.

[123] Shubham Tulsiani, Hao Su, Leonidas J Guibas, Alexei A Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *CVPR*, pages 2635–2643, 2017.

[124] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *CVPR*, pages 2626–2634, 2017.

[125] Narek Tumanyan, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Splicing ViT features for semantic appearance transfer. *arXiv preprint arXiv:2201.00424*, 2022.

[126] Gul Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. Bodynet: Volumetric inference of 3d human body shapes. In *ECCV*, pages 20–36, 2018.

[127] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[128] Timo von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3D human pose in the wild using IMUs and a moving camera. In *ECCV*, pages 601–617, 2018.

[129] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2Mesh: Generating 3D mesh models from single RGB images. In *ECCV*, pages 52–67, 2018.

[130] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, pages 4690–4699, 2021.

[131] Shangzhe Wu, Ruining Li, Tomas Jakab, Christian Rupprecht, and Andrea Vedaldi. Magicpony: Learning articulated 3d animals in the wild. *arXiv preprint arXiv:2211.12497*, 2022.

[132] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond PASCAL: A benchmark for 3D object detection in the wild. In *WACV*, pages 75–82, 2014.

[133] Jun Xie, Pheng-Ann Heng, and Mubarak Shah. Shape matching and modeling using skeletal context. *Pattern Recognition*, 41(5):1756–1767, 2008.

[134] Yuanlu Xu, Song-Chun Zhu, and Tony Tung. Denserac: Joint 3d pose and shape estimation by dense render-and-compare. In *ICCV*, pages 7760–7770, 2019.

[135] Kohei Yamashita, Shohei Nobuhara, and Ko Nishino. 3D-GMNet: Learning to estimate 3D shape from a single image as a Gaussian mixture. *arXiv preprint arXiv:1912.04663*, 2019.

[136] Gengshan Yang, Deqing Sun, Varun Jampani, Daniel Vlasic, Forrester Cole, Huiwen Chang, Deva Ramanan, William T Freeman, and Ce Liu. LASR: Learning articulated shape reconstruction from a monocular video. In *CVPR*, pages 15980–15989, 2021.

[137] Gengshan Yang, Deqing Sun, Varun Jampani, Daniel Vlasic, Forrester Cole, Ce Liu, and Deva Ramanan. ViSER: Video-specific surface embeddings for articulated 3d shape reconstruction. *NeurIPS*, 34, 2021.

[138] Gengshan Yang, Minh Vo, Natalia Neverova, Deva Ramanan, Andrea Vedaldi, and Hanbyul Joo. BANMo: Building animatable 3D neural models from many casual videos. *arXiv preprint arXiv:2112.12761*, 2021.

[139] Gengshan Yang, Chaoyang Wang, N Dinesh Reddy, and Deva Ramanan. Reconstructing animatable categories from videos. *arXiv preprint arXiv:2305.06351*, 2023.

[140] Chun-Han Yao, Wei-Chih Hung, Varun Jampani, and Ming-Hsuan Yang. Discovering 3d parts from image collections. In *ICCV*, pages 12981–12990, 2021.

[141] 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. *NeurIPS*, 35:15296–15308, 2022.

[142] Chun-Han Yao, Wei-Chih Hung, Yuanzhen Li, Michael Rubinstein, Ming-Hsuan Yang, and Varun Jampani. Hi-lassie: High-fidelity articulated shape and skeleton discovery from sparse image ensemble. In *CVPR*, pages 4853–4862, 2023.

[143] Chun-Han Yao, Amit Raj, Wei-Chih Hung, Yuanzhen Li, Michael Rubinstein, Ming-Hsuan Yang, and Varun Jampani. Artic3d: Learning robust articulated 3d shapes from noisy web image collections. *NeurIPS*, 2023.

[144] Chun-Han Yao, Jimei Yang, Duygu Ceylan, Yi Zhou, Yang Zhou, and Ming-Hsuan Yang. Learning visibility for robust dense human body estimation. In *ECCV*, pages 412–428. Springer, 2022.

[145] Yufei Ye, Shubham Tulsiani, and Abhinav Gupta. Shelf-supervised mesh prediction in the wild. In *CVPR*, pages 8843–8852, 2021.

[146] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, pages 4578–4587, 2021.

[147] Wang Zeng, Wanli Ouyang, Ping Luo, Wentao Liu, and Xiaogang Wang. 3d human mesh regression with dense correspondence. In *CVPR*, pages 7054–7063, 2020.

[148] Jason Zhang, Gengshan Yang, Shubham Tulsiani, and Deva Ramanan. NeRS: Neural reflectance surfaces for sparse-view 3D reconstruction in the wild. *NeurIPS*, 34, 2021.

[149] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. PhySG: Inverse rendering with spherical Gaussians for physics-based material editing and relighting. In *CVPR*, pages 5453–5462, 2021.

[150] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023.

[151] Tianshu Zhang, Buzhen Huang, and Yangang Wang. Object-occluded human shape and pose estimation from a single color image. In *CVPR*, pages 7376–7385, 2020.

[152] Tongjie Y Zhang and Ching Y. Suen. A fast parallel algorithm for thinning digital patterns. *CACM*, 27(3):236–239, 1984.

[153] Xiaowei Zhou, Menglong Zhu, Georgios Pavlakos, Spyridon Leonardos, Konstantinos G Derpanis, and Kostas Daniilidis. Monocap: Monocular human motion capture using a cnn coupled with a geometric prior. *PAMI*, 41(4):901–914, 2018.

[154] 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 *ICCV*, pages 5359–5368, 2019.

[155] Silvia Zuffi, Angjoo Kanazawa, and Michael J Black. Lions and tigers and bears: Capturing non-rigid, 3D, articulated shape from images. In *CVPR*, pages 3955–3963, 2018.

[156] Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *CVPR*, pages 6365–6373, 2017.