Unsupervised Learning of 3D Objects in the Wild



Shangzhe Wu

St Anne's College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2022

What I cannot create, I do not understand.

— Richard Feynman

Acknowledgements

None of this would have been possible without the support of so many.

First and foremost, I am extremely grateful to my advisors Andrea Vedaldi and Christian Rupprecht, who led me to the field of 3D computer vision. Their passion in pursuing challenging problems, creativity in developing new ideas and rigor in conducting scientific research have shaped me into the researcher I am today. This thesis and all of my achievements during this DPhil course would not have been possible without their constant guidance and support. I am also immensely grateful to Andrew Zisserman for his generous advice, encouragement and support, and for being a role model for so many of us in computer vision.

I am also extremely fortunate to have met the best mentors at Google Research, Jiajun Wu, Noah Snavely, Angjoo Kanazawa, Ameesh Makadia and Richard Tucker, who have continued to support me beyond the internship. I also thank my collaborators, Tomas Jakab, Zirui Wang, Weidi Xie, Felix Wimbauer, Tim Tang, Keqiang Sun, Hongsheng Li, Yunzhi Zhang and Ruining Li, for sharing valuable insights, inspirations and encouragement. A special thank-you goes to Ashish Thandavan for being a giant behind the servers, and to Jenny Hu for being an angel taking care of the lab. I am so blessed to have met so many friends in VGG and Oxford, who have made Oxford such a beautiful place, full of joy, excitement and love.

I am also indebted to Meta Research for the generous funding for my DPhil course, and to St Anne's College for providing a home in Oxford for me. Of course, I shall not forget to mention BuzzGym, Royal Oak, Victoria and Gardeners Arms, where I left both laughter and tears.

It has been a surprisingly long journey since I stepped out of the small town in China. I have been extremely fortunate to have had many great teachers and mentors, in particular Qiongsha Zhang, Changbing Cai, Shuifu Li, Chi-Keung Tang and Yu-Wing Tai, as well as many dear friends in StarCollege.

There is no way to pay back the love my parents, my brother and my family have offered me. What I shall always strive for is to make them proud of me.

Lastly, to Yiwen, for healing my soul.

Abstract

We live in a physical 3D world. Being able to perceive the world in 3D allows us to confidently navigate and perform sophisticated actions in it. Developing 3D perception systems is not only key to many AR and robotics applications, but also fundamental to the goal of visual understanding. However, most existing learning-based image understanding models process images simply as compositions of 2D patterns, ignoring the fact that images arise from a physical 3D world. Recent advances in differentiable rendering have opened new possibilities for learning physically-grounded disentangled 3D representations from raw images, marrying geometry with learning at last.

Learning 3D representations typically requires 3D supervision. This has previously come in the form of 3D labels, shape models, multi-views, or 2D geometric annotations, such as keypoints, masks and camera viewpoints. However, all of these are prohibitively expensive to collect on a large scale, which has become a major obstacle for learning 3D representations of general objects in the wild.

In this thesis, we explore the possibility of learning deformable 3D objects simply from raw casually recorded 2D image data "in the wild", such as online photos and YouTube videos, without relying on heavy manual supervision. At the core of our approach is a Photo-Geometric Autoencoding framework that decomposes images into a set of physically-grounded photometric and geometric factors, including 3D shape, pose, shiny surface material, environment lighting and camera parameters, with the objective of recomposing them to reconstruct the input images through a differentiable renderer. In order to resolve the ambiguity in this highly ill-posed task, we propose to inject generic inductive biases to exploit the symmetries and regularities of the world. The resulting model is able to learn disentangled 3D priors of various deformable object categories, allowing for single-image inference at test time.

Contents

1 Introduction **1**
- 1.1 Motivation . . . 2
- 1.2 Key Contributions . . . 4
  - 1.2.1 Learning Symmetric 3D Objects from Image Collections . . . 6
  - 1.2.2 Learning Material Decomposition from Image Collections . . . 7
  - 1.2.3 Learning Articulated 3D Objects from Video Collections . . . 8
- 1.3 List of Publications . . . 9

2 Literature Review **13**
- 2.1 3D Reconstruction . . . 13
  - 2.1.1 3D Reconstruction from Multiple Views . . . 13
  - 2.1.2 Shape from X . . . 14
  - 2.1.3 Learning 3D Reconstruction with Direct Supervision . . . 15
  - 2.1.4 Learning 3D Reconstruction with Weaker Supervision . . . 16
- 2.2 Material and Illumination Decomposition . . . 17
  - 2.2.1 Intrinsic Image Decomposition . . . 17
  - 2.2.2 Inverse Rendering from Multiple Images . . . 18
  - 2.2.3 Learning-based Single Image Inverse Rendering . . . 18

3 Unsupervised Learning of Probably Symmetric Deformable 3D Objects from Images in the Wild **21**

4 De-rendering the World's Revolutionary Artefacts **43**

5 De-rendering 3D Objects in the Wild **57**

6 DOVE: Learning Deformable 3D Objects by Watching Videos **73**

7 Summary **103**

References **105**

Appendices

A Sources of Figures **121**

Nature! She is the only artist; working-up the most uniform material into utter opposites; arriving, without a trace of effort, at perfection, at the most exact precision, though always veiled under a certain softness. Each of her works has an essence of its own; each of her phenomena a special characterisation; and yet their diversity is in unity.[a]

— Georg Christoph Tobler[b]

1 Introduction

We live in a physical 3D world. By observing 2D projections of the world, we develop a coherent, comprehensive 3D understanding of it. This prior knowledge allows us to infer 3D properties of the world from images. For instance, from a single 2D photograph of a car, we can easily imagine what the car looks like in 3D and whether it is painted with matte or shiny materials.

Being able to perceive our dynamic world in 3D from 2D visual observations is not only key to many Augmented Reality (AR) and robotics applications, but also fundamental to the goal of visual understanding. In this thesis, we aim to equip machines with a similar capability of perceiving the 3D world from only 2D observations. Specifically, we design unsupervised algorithms that allow machines to automatically learn priors about the 3D shape and appearance of object categories from unstructured 2D data "in the wild", such as online photos and YouTube videos, without external manual supervision. Once such 3D priors are learned, from only a single image of a new object at test time, the model is able to infer its full 3D shape as well as its appearance properties, such as surface color and shininess, which enables

[a] Translated by T. H. Huxley [Huxley 1869].

[b] According to Boyle 1992, this aphorism titled "Nature" was written by Georg Christoph Tobler based on many conversations with Johann Wolfgang von Goethe, but is often incorrectly attributed to Goethe.

various kinds of 3D editing applications, including object rearrangement, relighting and re-rendering, as well as robotics tasks like navigation and manipulation.

1.1 Motivation

The task of 3D reconstruction traditionally requires a set of images captured from multiple viewpoints (multi-view), with the assumption that the scene remains static, in order to apply epipolar geometry [Faugeras and Luong 2001; Hartley and Zisserman 2004]. For reconstructing deformable objects in this dynamic world, we often need specialized multi-camera devices to capture simultaneous multi-view images [Wenger et al. 2005; Kehl and Van Gool 2006; de Aguiar et al. 2008; Bradley et al. 2010; Ghosh et al. 2011; Debevec 2012; Joo et al. 2018], which is tedious and sometimes infeasible in practice. With the recent adoption of machine learning, 3D shape prediction from a single 2D image has yielded promising results, by capturing suitable 3D priors over a large collection of object instances [Choy et al. 2016; Girdhar et al. 2016; J. Wu et al. 2016; Bogo et al. 2016; Tatarchenko et al. 2017; Groueix et al. 2018; Kanazawa et al. 2018a; Kanazawa et al. 2018b; N. Wang et al. 2018; J. J. Park et al. 2019; Mescheder et al. 2019; Gkioxari et al. 2019]. However, learning such priors usually requires heavy supervision during training. Designing effective supervision mechanism for learning 3D priors is, therefore, a critical step in developing machine learning models for single-image 3D reconstruction.

In this thesis, we aim to design unsupervised methods for learning 3D object priors, in the form of 3D shape, appearance and surface material, from online images and videos. Studying this problem has both practical and scientific values from the following three perspectives.

Data Efficiency for 3D Learning. Learning 3D priors typically requires 3D supervision. This can come in the form of 3D training data, such as 3D scans [Eigen et al. 2014; Varol et al. 2018; Saito et al. 2019; Jafarian and H. S. Park 2021] and synthetic shape models [Choy et al. 2016; Tatarchenko et al. 2017; Groueix et al. 2018; J. J. Park et al. 2019; Mescheder et al. 2019; Varol et al. 2018]. However,

collecting large-scale 3D datasets with the level of diversity and photorealism that the real world presents is prohibitively expensive. Hence, this has only been done for objects that are of particular significance in applications so far, such as human bodies and faces [Paysan et al. 2009; M. Loper et al. 2015; T. Li et al. 2017].

Cheaper alternatives for geometric supervision have been explored, such as multi-view images or videos [Godard et al. 2017; Zhou et al. 2017; Novotny et al. 2017; Zhengqi Li et al. 2019; Reizenstein et al. 2021], 3D or 2D keypoint annotations [Cashman and Fitzgibbon 2012; Vicente et al. 2014; Kanazawa et al. 2018b; Novotny et al. 2019], object mask annotations [Cashman and Fitzgibbon 2012; Vicente et al. 2014; Kanazawa et al. 2018b; Ye et al. 2021], and category-specific template shapes [N. Kulkarni et al. 2020; Goel et al. 2020; F. Kokkinos and I. Kokkinos 2021]. However, these annotations still require extensive efforts to collect on a large scale. Moreover, geometric and physical quantities, including also surface material and illumination properties, are, in general, much more difficult for humans to annotate, in contrast to the recent successes in annotating properties, like object class and identity [Deng et al. 2009; Parkhi et al. 2015].

From a practical perspective, in order to enable 3D inference on various kinds of objects, we need to design data-efficient systems that can learn from as cheap sources of data as possible. Ideally, we would like to develop a 3D perception system that would automatically learn to perceive the 3D world by simply observing casually recorded images and videos, which already exist in virtually unlimited quantities on the Internet, albeit not specifically curated for the task of learning 3D representations. In this thesis, we seek to harvest 3D object priors from such unstructured, unconstrained online images and videos, without explicit external supervision.

Physical Interpretability for Image Understanding. Decomposing images into a set of explicit 3D representations of the underlying objects gives rise to a physically-grounded interpretation of images. This is particularly valuable in the context of learning. Most existing image representation learning approaches

treat images simply as a composition of 2D patterns [Geirhos et al. 2019], ignoring the fact that they are taken from a physical 3D world. The lack of 3D reasoning and physical interpretability in learning-based image understanding models could be a major obstacle in building accountable perception systems that could be safely deployed in the physical world.

On the other hand, by factoring out the effect of the extrinsic imaging conditions, such as viewpoint and illumination, the remaining variations in the intrinsic properties of the underlying objects, such as 3D shape and albedo, become significantly smaller. This disentangled space yields a much simpler distribution of object instances than that in the raw pixel space and could largely simplify the learning process.

Pursuit of Minimal Assumptions in 3D Perception. Designing unsupervised methods that allow 3D representations to emerge from raw image observations is also related to the scientific problem of seeking minimal assumptions that make 3D perception possible. Natural intelligences, including human beings and many other animals, develop a comprehensive understanding of the 3D world that enables them to navigate and act competently in it. Crucially, they do not seem to rely on extensive explicit 3D supervision (*e.g.* constantly bumping into walls) in order to build such 3D understanding. Instead, we do so primarily by simply observing the world. This begs the question of whether or not explicit 3D supervision is required to learn 3D perception, or more fundamentally: "what are the minimal assumptions for 3D perception?" Studying unsupervised algorithms for 3D perception helps draw insights into these questions, and could potentially lead to solutions closer to those of natural intelligences.

1.2 Key Contributions

Instead of relying on explicit geometric supervision for learning 3D objects, which is expensive and often infeasible at scale, we would like to design self-supervised algorithms that can automatically learn from raw, unstructured 2D images and

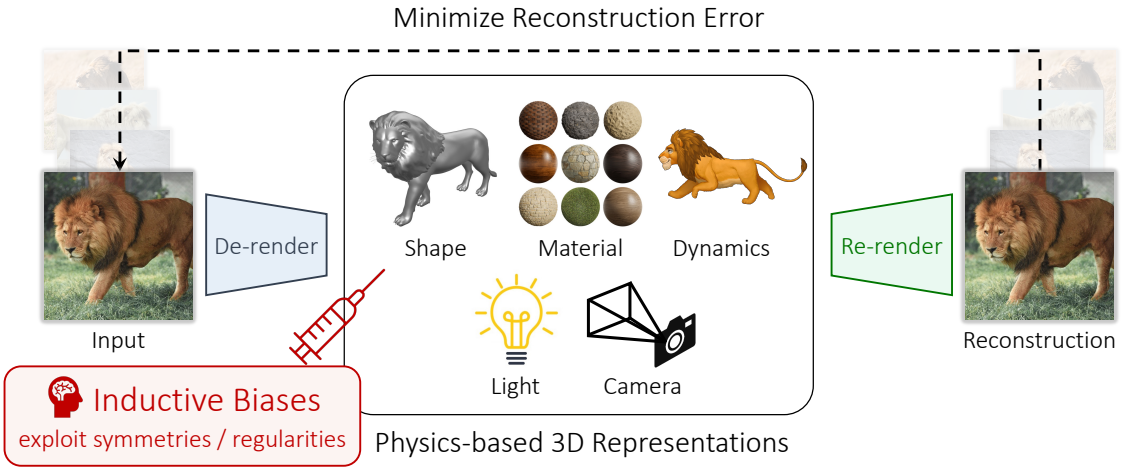


Figure 1.1: Photo-Geometric Autoencoding (PGA), a generic learning framework, in which the model learns to decompose input images into a set of physically meaningful photometric and geometric factors, including 3D shape, surface material, object dynamics, lighting and camera parameters, with the objective of reconstructing the input images. The key is to inject generic inductive biases exploiting symmetries and regularities of the world, in order to allow the internal physically-based 3D representations to automatically emerge without explicit supervision. Image sources: see Appendix A.

videos. At the core of our proposed methods is a Photo-Geometric Autoencoding (PGA) framework, illustrated in Fig. 1.1, where the model learns to decompose the input images into a set of physically meaningful photometric and geometric factors (*e.g.* 3D shape, camera viewpoint, albedo, lighting, material properties *etc.*), with the objective of recomposing them to reconstruct the input images through a differentiable renderer [M. M. Loper and Black 2014; Kato et al. 2018; S. Liu et al. 2019; Ravi et al. 2020; Jakob et al. 2022].

However, recovering 3D shape, pose and appearance of deformable objects from monocular 2D images is an extremely ill-posed problem. In order to resolve the ambiguity, we seek to exploit prior knowledge about the symmetries and regularities that are widely present in our world. Specifically, we have made the following three key contributions to demonstrate the power of this idea: learning symmetric 3D objects from image collections, learning material decomposition from image collections, and learning articulated 3D objects from video collections.

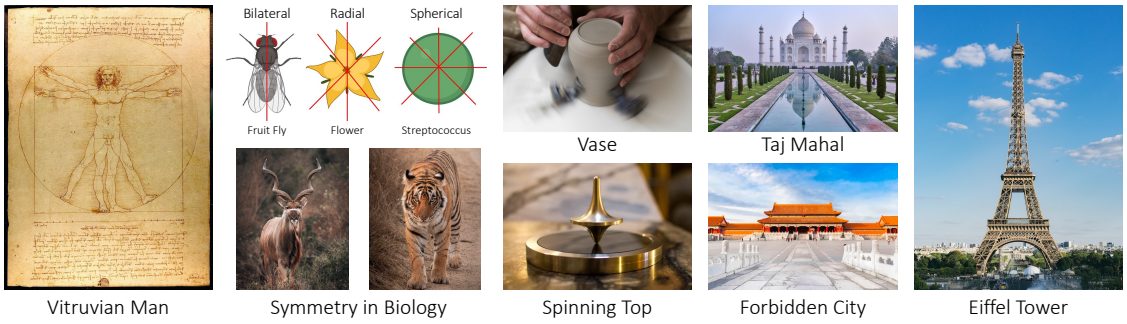


Figure 1.2: Symmetries in the World. Symmetric structures can be found everywhere in the world, ranging from natural objects, like animals and plants, to man-made artifacts and buildings. Images taken from the Internet (Wikipedia and Unsplash) with public licenses (Sources: see Appendix A).

1.2.1 Learning Symmetric 3D Objects from Image Collections

Symmetries are everywhere in the world [Weyl 1952; Tarasov 1986; Johnston et al. 2022]. Fig. 1.2 gives a few examples. Humans, many animals and man-made objects, like cars, chairs and architectures (*e.g.* Taj Mahal, Eiffel Tower, Forbidden City), have a bilaterally symmetric structure. Jellyfish, starfish, many flowers (*e.g.* clovers, buttercups) and objects like tires exhibit a radial symmetry, whereas bottles, vases, archery targets and spinning tops exhibit a continuous circular symmetry. Some other objects, on the other hand, like balls, colonial algae (*e.g.* Volvox) and our planet, are roughly spherically symmetric.

Symmetries entail invariance; a single image of a symmetric instance essentially provides "multiple views" of it, which can be used to constrain this ill-posed task of disentangling 3D shape, pose and appearance. For instance, if an object is bilaterally symmetric, flipping the image of it will generate a virtual "view" of the object from another viewpoint, mirrored about its symmetry plane, and 3D reconstruction can be achieved with two-view stereo reconstruction [Mukherjee et al. 1995; François et al. 2003; Thrun and Wegbreit 2005; Sinha et al. 2012; Gao and Yuille 2017]. Moreover, the structure of symmetry is often shared by various instances within an object category, which helps derive a canonical frame, on which all instances can be aligned, factoring out the pose of each individual image.

In this thesis, we show that these common patterns of symmetry can be leveraged as an effective constraint for learning disentangled 3D representations of natural objects from raw images. We leverage bilateral symmetry to learn 3D shapes of human faces and cat faces from a collection of single-view images in Chapter 3, as well as full bodies of animals, including birds and horses, from a collection of monocular video clips in Chapter 6. Moreover, we also show in Chapter 3 that by exploiting bilateral symmetry on the appearance of the objects, the model automatically learns to decompose the albedo and illumination from the images without any external supervision. In Chapter 4, we exploit another type of symmetry—solids of revolution—and develop an algorithm that learns the 3D shape and shiny surface material of vases also from only a single-view image collection.

1.2.2 Learning Material Decomposition from Image Collections

Not only can humans infer 3D shapes from a single image, we can also easily factor out the effects of illumination from the pixel colors and infer the intrinsic appearance of the underlying surface and how shiny it is. Recovering accurate surface materials (*e.g.* bidirectional reflectance distribution function or BRDF) is crucial for synthesizing photo-realistic images in many Visual Effects (VFX) applications [Kanamori and Endo 2018; Zhengqin Li et al. 2020; Z. Wang et al. 2021; Pandey et al. 2021; Tan et al. 2022].

Recovering surface materials typically requires a dense set of images captured with varying lighting conditions [Woodham 1980; Coleman and Jain 1982; Hernández et al. 2008; Alldrin et al. 2008; Goldman et al. 2009]. Existing methods that learn to recover surface materials from a single image rely on synthetic training data [Shi et al. 2017; G. Liu et al. 2017; Janner et al. 2017; Sengupta et al. 2018; Zhengqin Li et al. 2018b; Wenzheng Chen et al. 2019; Sang and Chandraker 2020; Zhengqin Li et al. 2020; Wenzheng Chen et al. 2021; Hou et al. 2021; Lagunas et al. 2021; Z. Wang et al. 2021], which often makes it challenging to generalize to diverse real world objects.

In this thesis, we seek to learn the decomposition of surface materials and illumination from raw image collections of real objects without external supervision. In Chapter 3, our model automatically learns to decompose albedo and diffuse shading from single-view image collections without any direct supervision, by exploiting the bilateral symmetry in the intrinsic appearance of the objects, and assuming a Lambertian illumination model and a simple directional light with an ambient component. In Chapter 4, we learn to recover shiny materials as well as 3D shape from only a single-view image collection of rotationally symmetric objects, like vases, assuming a Phong illumination model with a shininess scalar and grayscale environment map lighting. In a follow-up work in Chapter 5, we extend this framework for learning shiny materials of general objects, such as daily objects captured in the CO3D dataset [Reizenstein et al. 2021], assuming a single-view image collection with coarse estimates of the geometry (in the form of noisy normal maps) as training data.

1.2.3 Learning Articulated 3D Objects from Video Collections

Thus far, we have only discussed rigid objects, like vases and the ones in CO3D [Reizenstein et al. 2021], and mildly deformable objects, like faces. However, many objects in the wild are, in fact, much more deformable, such as animals, whose shapes tend to deform with an articulated structure. In order to model such articulated objects, we seek to leverage temporal correspondence signals to factor out the articulated pose from the canonical shape. Similar to pictures, we also have a large amount of online videos capturing various kinds of moving objects. Our next goal is, hence, to learn articulated 3D objects, from a collection of online videos, such as clips extracted from YouTube.

However, unlike the "turn-table-like" videos in CO3D, which were crowdsourced from individuals given instructions to move around the objects, specifically curated for the task of 3D reconstruction, existing online videos are casually recorded, posing several major challenges for 3D learning. First, the objects can be moving

freely, resulting in large viewpoint changes and fast motion, which causes significant challenges for disentangling pose and shape. Second, these video clips are typically only a few seconds long, capturing only a small fraction of views for each instance at a time, which are often not sufficient for reconstructing the full 3D shape. Hence, to learn a full 3D model of the objects requires aligning all clip instances and aggregating information across the entire video collection.

In Chapter 6, we propose a method that learns an articulated 3D model of a deformable object category, such as birds and horses, from scratch (by deforming a sphere), using only a collection of video clips extracted from YouTube as training data. Crucially, unlike existing methods for learning 3D shapes from image collections [Kanazawa et al. 2018b; Nguyen-Phuoc et al. 2019; Xueting Li et al. 2020b; Goel et al. 2020; F. Kokkinos and I. Kokkinos 2021; E. Chan et al. 2021], which typically require extensive geometric supervision in the form of keypoints, template shapes and known camera viewpoints, our method does not rely on any external geometric annotations, apart from object masks and optical flows, automatically obtained from off-the-shelf models.

1.3 List of Publications

- Chapter 3: "Unsupervised Learning of Probably Symmetric Deformable 3D Objects from Images in the Wild". Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021; and in: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Best Paper Award), 2020.

- Chapter 4: "De-rendering the World's Revolutionary Artefacts". Shangzhe Wu, Ameesh Makadia, Jiajun Wu, Noah Snavely, Richard Tucker, and Angjoo Kanazawa. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

- Chapter 5: "De-rendering 3D Objects in the Wild". Felix Wimbauer, Shangzhe Wu, and Christian Rupprecht. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

- Chapter 6: "DOVE: Learning Deformable 3D Objects by Watching Videos". Shangzhe Wu*, Tomas Jakab*, Christian Rupprecht, and Andrea Vedaldi. In: *arXiv preprint arXiv:2107.10844*, 2021. In submission. *The first two authors contributed equally.

Papers that I contributed to during my DPhil and are not included in this thesis:

- "Self-Supervised Localisation between Range Sensors and Overhead Imagery". Tim Y. Tang, Daniele De Martini, Shangzhe Wu, and Paul Newman. In: *Robotics: Science and Systems (RSS)*, 2020; and in: *International Journal of Robotics Research (IJRR)*, 2021.

- "NeRF−−: Neural Radiance Fields Without Known Camera Parameters". Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. In: *arXiv preprint arXiv:2102.07064*, 2021.

- "Controllable 3D Face Synthesis with Conditional Generative Occupancy Fields". Keqiang Sun, Shangzhe Wu, Zhaoyang Huang, Ning Zhang, Quan Wang, and Hongsheng Li. In: *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Table 2.1: (Part I) Related Work Comparison: Supervision, Output and Data.

Method	Supervision								Output						Data
									3D	Dynamic	View	Texture	Light	Shiny	
Static, Optimization-based															
BuildingRome [Agarwal et al. 2009]			✓						PC		✓	✓			Landmarks
COLMAP [Schönberger and Frahm 2016]			✓						PC		✓	✓			Other
NeRF [Mildenhall et al. 2020]			✓	✓					Neural			✓			Other
IDR [Yariv et al. 2020]			✓	✓		✓			Neural		✓	✓			Other
SIRFS [Barron and Malik 2015]						✓			Normal			✓	✓		Other
NeRFactor [X. Zhang et al. 2021]			✓	✓					Neural			✓	✓	✓	Other
Dynamic, Optimization-based															
NRSfM of [Bregler et al. 2000]					✓		✓		3D KP	✓	✓				Other
NRSfM of [Akhter et al. 2008]					✓		✓		3D KP	✓	✓				Other
D-NeRF [Pumarola et al. 2020]				✓			✓		Neural	✓		✓			Other
ACMR† [Xueting Li et al. 2020a]		✓		✓	✓	✓	✓		Mesh	✓	✓	✓			Animals
LASR† [G. Yang et al. 2021a]						✓	✓	✓	Mesh	✓	✓	✓			Animals, Humans
ViSER [G. Yang et al. 2021b]						✓	✓	✓	Mesh	✓	✓	✓			Animals, Humans
BANMo [G. Yang et al. 2022]				(✓)[1]	(✓)[1]	✓	✓	✓	Neural	✓	✓	✓			Animals, Humans
LASSIE [Yao et al. 2022]		✓			(✓)[2]	✓			Mesh	✓	✓	✓			Animals
Strongly-supervised Learning-based															
SingleDepth [Eigen et al. 2014]	✓								Depth						Other
3D-R2N2 [Choy et al. 2016]	✓		✓						Voxel						ShapeNet, PASCAL3D, Other
T-L Network [Girdhar et al. 2016]	✓			✓					Voxel						ShapeNet, IKEA
3D-VAE-GAN [J. Wu et al. 2016]	✓			✓					Voxel						ModelNet, IKEA
OGN [Tatarchenko et al. 2017]	✓								Voxel						ShapeNet
PointSet [Fan et al. 2017]	✓								PC						ShapeNet
AtlasNet [Groueix et al. 2018]	✓								Mesh						ShapeNet
Pixel2Mesh [N. Wang et al. 2018]	✓			✓					Mesh						ShapeNet
DeepSDF [J. J. Park et al. 2019]	✓								Neural						ShapeNet
OccNet [Mescheder et al. 2019]	✓								Neural						ShapeNet, KITTI
Mesh R-CNN [Gkioxari et al. 2019]	✓					✓			Mesh						ShapeNet, Pix3D
ShapeNet-Intrinsics [Shi et al. 2017]	✓					✓			-			✓		✓	ShapeNet, Other
SI-SVBRDF [Zhengqin Li et al. 2018b]	✓					✓			Depth			✓	✓	✓	Other
BodyNet [Varol et al. 2018]	✓		✓	✓	✓	✓			Voxel	✓	✓	✓			Humans
PIFu [Saito et al. 2019]	✓			✓					Neural			✓			Humans
(to be continued ...)															

Table 2.1: (Part II) Related Work Comparison: Supervision, Output and Data.

Method	Supervision								Output						Data
	3D	template	multi-view	viewpoint	2D kp	mask	video	optical flow	3D	Dynamic	View	Texture	Light	Shiny	
Strongly-supervised Learning-based															
(... continued)															
MoFA [Tewari et al. 2017]	(✓)[3]				✓	✓			Mesh	✓		✓			Faces
DECA [Feng et al. 2021]	(✓)[3]				✓	✓			Mesh	✓		✓			Faces
HMR [Kanazawa et al. 2018a]	(✓)[3]				✓				Mesh	✓		✓			Humans
VpNR-Net [Novotny et al. 2017]	(✓)[4]		(✓)[4]	✓					PC			✓			Cars, Chairs, PASCAL3D
FrozenPeople [Zhengqi Li et al. 2019]	(✓)[4]		(✓)[4]			✓			Depth						Humans
Weakly-supervised Learning-based															
MonoDepth [Godard et al. 2017]			✓	✓			✓		Depth						Other
SfMLearner [Zhou et al. 2017]							✓		Depth						Other
Dolphin [Cashman and Fitzgibbon 2012]		✓		✓	✓	✓			Mesh			✓			Animals
CategoryRecon [Kar et al. 2015]		(✓)[5]		(✓)[5]	✓	✓			Mesh			✓			PASCAL3D
CMR† [Kanazawa et al. 2018b]		(✓)[5]		(✓)[5]	✓	✓			Mesh			✓	(✓)[6]		Birds, PASCAL3D
A-CSM [N. Kulkarni et al. 2020]		✓				✓			Mesh	✓		✓			Animals
U-CMR† [Goel et al. 2020]		✓				✓			Mesh			✓	✓		Birds, PASCAL3D
TTP† [F. Kokkinos and I. Kokkinos 2021]		✓				✓			Mesh			✓	(✓)[6]		Birds, PASCAL3D
UMR† [Xueting Li et al. 2020b]					(✓)[7]	✓			Mesh			✓	(✓)[6]		Animals, PASCAL3D
C3DPO [Novotny et al. 2019]					✓				3D KP			✓			Humans, Other
SRN [Sitzmann et al. 2019]				✓					Neural				✓		ShapeNet
HoloGAN‡ [Nguyen-Phuoc et al. 2019]									Neural				✓		Faces, Other
GRAF‡ [Schwarz et al. 2020]									Neural				✓		Other
pi-GAN‡ [E. Chan et al. 2021]									Neural				✓		Faces, Other
pixelNeRF [A. Yu et al. 2021]			✓	✓					Neural				✓		Other
NerFormer [Reizenstein et al. 2021]			(✓)[4]	✓		✓			Neural				✓		CO3D
Our proposed, Weakly-supervised Learning-based															
Unsup3D† (Chapter 3)									Depth			✓	✓	✓	Faces, Other
SoRDerender† (Chapter 4)						✓			Mesh			✓	✓	✓	Vases
Derender (Chapter 5)	✓								Normal				✓	✓	Faces, CO3D, Other
DOVE† (Chapter 6)						✓	✓	✓	Mesh	✓		✓			Animals

Annotations: 3D ground-truth, template shape, multiple views, viewpoint, 2D keypoint, object mask, video, optical flow, [1]estimated using pretrained CSEs [Neverova et al. 2020], [2]estimated using DINO features [Caron et al. 2021], [3]parametric 3D shape models, [4]obtained from videos of static scenes, [5]estimated from 2D keypoints, [6]predicted using texture flow, [7]part segmentations from SCOPS [Hung et al. 2019], †leverages symmetry constraints, ‡GAN-based methods that require known viewpoint distribution, PC: point clouds, 3D KP: 3D keypoints, ShapeNet: Chang et al. 2015, ModelNet: Z. Wu et al. 2015, IKEA: Lim et al. 2013, KITTI: Geiger et al. 2013, PASCAL3D: Xiang et al. 2014, Pix3D: X. Sun et al. 2018, CO3D: Reizenstein et al. 2021.

2 Literature Review

The goal of this thesis is to learn 3D representations of deformable objects, including 3D shape, pose and surface material, from online images and videos without heavy manual supervision. In this section, we review the literature on two relevant topics: 3D reconstruction and material and illumination decomposition from images. Table 2.1 gives a comprehensive comparison of our proposed methods and the most relevant prior arts, in terms of supervision, output and data.

2.1 3D Reconstruction

2.1.1 3D Reconstruction from Multiple Views

Reconstructing 3D structures from images is a long-standing problem in computer vision. Structure from Motion (SfM) is an extensively studied approach for reconstructing static 3D scenes from a set of images captured from multiple viewpoints [Faugeras 1993; Faugeras and Luong 2001; Hartley and Zisserman 2004]. This typically involves detecting 2D keypoint matches between the views, estimating camera parameters for each image and lifting the 2D correspondences into 3D, based on multi-view geometry. This can be extended to Non-Rigid Structure from Motion (NRSfM) for reconstructing non-static objects from 2D correspondences [Bregler et al. 2000; Xiao et al. 2004; Akhter et al. 2008; Dai et al. 2012; Cashman and

Fitzgibbon 2012], by injecting additional constraints and assumptions about the object motion in an optimization pipeline. Nevertheless, it still remains extremely challenging in practice to solve for all kinds of motion of general deformable objects.

Recently, a line of work has shown impressive reconstruction results of deformable 3D objects, such as quadrupedal animals, from raw monocular video sequences, with only object mask and optical flow annotations [G. Yang et al. 2021a; G. Yang et al. 2021b]. It was later extended to multi-sequence optimization [G. Yang et al. 2022], leveraging Continuous Surface Embedding (CSE) [Neverova et al. 2020] predictions for viewpoint initialization and registration across sequences. A more recent piece of work, LASSIE [Yao et al. 2022], shows that it is possible to optimize deformable 3D shapes from simply an "image ensemble" (10-30 online images) of different instances in the same object category captured in various environments, by assuming a heavily constrained part-based shape representation, a known articulated 3D skeleton and part correspondences extracted from self-supervised image features [Caron et al. 2021; Amir et al. 2021].

However, all of these methods require well-captured multiple views for optimization, whereas in real life applications, we often have access to only a limited fraction of views of a scene as we navigate through it. Reasoning about the full 3D geometry of a partially observed scene, or even from only a single photograph, requires prior knowledge of the 3D world. Therefore, a learning-based approach that can capture 3D object priors is particularly convenient for few-view inference in such scenarios.

2.1.2 Shape from X

A wide range of alternative cues in images have been exploited for recovering shape, such as shading [B. K. Horn 1975; B. K. P. Horn and Brooks 1989; R. Zhang et al. 1999], shadow [Daum and Dudek 1998], defocus [Favaro and Soatto 2007], sillouettes [Baumgart 1974; Laurentini 1994], texture [Witkin 1981; Super and Bovik 1995; Lobay and Forsyth 2006], symmetry [Mukherjee et al. 1995; François et al. 2003; Thrun and Wegbreit 2005] and so on. In particular, our work in this thesis is inspired from both shape from shading and shape from symmetry.

Shape from shading aims to recover surface geometry by exploiting changes in the illumination with assumed shading and lighting models, such as a Lambertian model and a directional light. Photometric stereo reconstruction methods can also recover non-Lambertian surfaces, but requires multiple images captured under different lighting conditions. Our models leverage shading cues for learning 3D shapes by computing losses on the reconstructed images rendered with an explicit shading model, such as a Lambertian model in Chapter 3 or a Phong model [Phong 1975] in Chapter 4.

Shape from symmetry aims to utilize synthetic "multi-views" induced by symmetry for shape reconstruction. One typical example is to exploit bilateral symmetry by mirroring the image and treating it as a virtual second view of the same symmetric object. François et al. 2003 shows that if correspondences between these "two views" can be obtained, 3D reconstruction is possible with stereo geometry. In Chapter 3, we use a similar idea of exploiting bilateral symmetry for learning 3D faces from single-view image collections. Our model automatically learns to establish correspondences without explicit annotations and to discount non-symmetric regions with an uncertainty-aware reconstruction loss. In Chapter 4, we exploit another type of symmetry—rotational symmetry—for learning a 3D model of vases also from a single-view image collection.

2.1.3 Learning 3D Reconstruction with Direct Supervision

Learning-based approaches have prevailed in many computer vision tasks in the past decade, with the advantage of capturing powerful priors for accurate predictions on new test samples. 3D reconstruction can also be formulated as a supervised learning problem, where a model learns the mapping from a 2D image to the 3D shape, trained directly with 3D ground-truth labels. One popular example of this approach is supervised depth estimation [Eigen et al. 2014; Laina et al. 2016; Weifeng Chen et al. 2016], which learns to recover the geometry of visible parts of a scene in an image using a depth map.

Another group of methods learn full 3D shapes with various 3D representations, including voxels [Choy et al. 2016; Girdhar et al. 2016; J. Wu et al. 2016; Tatarchenko et al. 2017], point clouds [Fan et al. 2017; Y. Yang et al. 2018; Achlioptas et al. 2018], meshes [Groueix et al. 2018; N. Wang et al. 2018], and more recently, neural representations [J. J. Park et al. 2019; Z. Chen and H. Zhang 2019; Mescheder et al. 2019], by relying directly on dense 3D supervision, in the form of ground-truth 3D labels, parametric shape models or dense multi-views. However, such 3D labels are only available for synthetic objects and particular object categories of utmost importance in applications, such as human bodies and faces, due to the high cost of obtaining them. Therefore, in order to learn 3D models of general objects, we need to develop methods that can learn with weaker forms of supervision, ideally from raw 2D images alone.

2.1.4 Learning 3D Reconstruction with Weaker Supervision

An early example of unsupervised 3D learning from images is unsupervised depth estimation, where depth and ego-motion can be jointly learned from stereo or even monocular video sequences [Godard et al. 2017; Zhou et al. 2017].

Weaker forms of supervision have also been explored for learning full 3D shape of object categories, including object masks [Cashman and Fitzgibbon 2012; Vicente et al. 2014; Kar et al. 2015; Kanazawa et al. 2018b; Goel et al. 2020; Xueting Li et al. 2020b; Niemeyer et al. 2020; Ye et al. 2021; F. Kokkinos and I. Kokkinos 2021], 2D keypoints [Cashman and Fitzgibbon 2012; Vicente et al. 2014; Kar et al. 2015; Kanazawa et al. 2018b; Novotny et al. 2019], category-specific template shapes [N. Kulkarni et al. 2019; N. Kulkarni et al. 2020; Goel et al. 2020; Tulsiani et al. 2020; F. Kokkinos and I. Kokkinos 2021] and known camera viewpoints [Cashman and Fitzgibbon 2012; Sitzmann et al. 2019; Niemeyer et al. 2020; Reizenstein et al. 2021]. Although these annotations are easier to obtain than dense 3D labels, they still require an extensive amount of effort to collect on a large scale for training.

Another emerging paradigm is to adopt a generative adversarial framework (GAN) [Nguyen-Phuoc et al. 2019; Nguyen-Phuoc et al. 2020; Schwarz et al. 2020;

Niemeyer and Geiger 2021; E. Chan et al. 2021; Gu et al. 2022; Or-El et al. 2022; E. R. Chan et al. 2022], where 3D shapes can be learned by ensuring the 2D renderings look realistic from all viewpoints. Despite not requiring any instance annotations, this approach still relies heavily on an accurate estimation of the viewpoint distribution of the training images, and is also often fragile due to the unstable training dynamics of GANs. Hence, it has only found success in learning simple objects so far, such as faces and cars.

Our goal in this thesis is to remove the need for additional geometric supervision and develop unsupervised algorithms that can learn deformable 3D objects directly from casually-recorded online image and video collections. In Table 2.1, we compare our proposed methods and prior arts on both strongly-supervised and weakly-supervised 3D learning methods, in terms of supervision, output and data.

2.2 Material and Illumination Decomposition

2.2.1 Intrinsic Image Decomposition

One well-explored task in inverse rendering is intrinsic image decomposition, where the main goal is to factorize an image into a reflectance (or albedo) image and a shading image, *i.e.* separating the intrinsic surface color from illumination effects [Land and McCann 1971; Barrow and Tenenbaum 1978]. Since this is a highly ill-posed task, traditional methods often rely on additional heuristics and priors. The classic Retinex algorithm [Land and McCann 1971] assumes that small variations in image intensity result from shading whereas abrupt changes reveal the true surface reflectance. Many other priors have also been explored over the past few decades, such as global sparsity constraints on the reflectance [Shen et al. 2008; Rother et al. 2011; Shen and Yeo 2011; Garces et al. 2012], and explicit geometric constraints on shading assuming Lambertian surface [Barron and Malik 2015; Laffont et al. 2012]. Recently, researchers have studied learning-based approaches by training on synthetic data [Narihira et al. 2015; Shi et al. 2017; Zhengqi Li and Snavely 2018a; Y. Liu et al. 2020] or multi-illumination images [Zhengqi Li and Snavely 2018b; Ma et al. 2018; A. Liu et al. 2020]. In this work, we are not only

interested in decomposing images into two layers (albedo and shading), but also modeling the underlying 3D shape and material properties of object categories as well as the environment lighting—a more general inverse rendering task.

2.2.2 Inverse Rendering from Multiple Images

Inverse rendering aims to recover shape, material and lighting from 2D images. Photometric stereo reconstruction techniques [Woodham 1980; Coleman and Jain 1982; Hernández et al. 2008; Alldrin et al. 2008; Goldman et al. 2009] recover shape, BRDF material and lighting by solving an optimization problem, given a set of images captured under various lighting conditions and optionally from multiple viewpoints. This has been extended with learning-based approaches [Bi et al. 2020b; Boss et al. 2020] and recently with neural representations [Yariv et al. 2020; Bi et al. 2020a; Boss et al. 2021; Srinivasan et al. 2021; K. Zhang et al. 2021; X. Zhang et al. 2021; K. Zhang et al. 2022]. However, capturing multi-view and multi-illumination images is challenging for many objects "in the wild". Hence, we would like to develop learning-based methods that will enable inverse rendering from a single image.

2.2.3 Learning-based Single Image Inverse Rendering

Inverse rendering can be formulated as a supervised learning task, typically on particular object categories or scenes with direction supervision from synthetic training data, such as ShapeNet [Chang et al. 2015] objects [G. Liu et al. 2017; Wenzheng Chen et al. 2019; Wenzheng Chen et al. 2021], synthetic human faces and bodies [Shu et al. 2017; Sengupta et al. 2018; Kanamori and Endo 2018; Lagunas et al. 2021], near-planar surfaces [Xiao Li et al. 2017; Deschaintre et al. 2018; Zhengqin Li et al. 2018a], indoor scenes [Zhengqin Li et al. 2020; Zhengqin Li et al. 2021; Z. Wang et al. 2021; Yeh et al. 2022; Zhu et al. 2022; Zhengqin Li et al. 2022] or other synthetic objects [Janner et al. 2017; Zhengqin Li et al. 2018b; Sang and Chandraker 2020]. Apart from using direct ground-truth supervision, researchers have also explored training with multi-illumination images, yet allowing for single-image inference [T. D. Kulkarni et al. 2015; Nestmeyer et al. 2020; Y. Yu

et al. 2020; Tajima et al. 2021; Pandey et al. 2021]. However, obtaining large-scale multi-illumination image datasets is very challenging and often only possible with synthetic data, except for objects of significant importance in applications, like human faces and bodies captured using light stages [T. Sun et al. 2019; Nestmeyer et al. 2020; Tajima et al. 2021; Pandey et al. 2021]. Therefore, in this thesis, we are interested in designing unsupervised methods for inverse rendering of general object categories using only casually-recorded online images as training data.

3

Unsupervised Learning of Probably Symmetric Deformable 3D Objects from Images in the Wild

This work was published at the IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2021. A conference version of this work was presented at the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

Unsupervised Learning of Probably Symmetric Deformable 3D Objects From Images in the Wild (Invited Paper)

Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi

Abstract—We propose a method to learn 3D deformable object categories from raw single-view images, without external supervision. The method is based on an autoencoder that factors each input image into depth, albedo, viewpoint and illumination. In order to disentangle these components without supervision, we use the fact that many object categories have, at least approximately, a symmetric structure. We show that reasoning about illumination allows us to exploit the underlying object symmetry even if the appearance is not symmetric due to shading. Furthermore, we model objects that are probably, but not certainly, symmetric by predicting a symmetry probability map, learned end-to-end with the other components of the model. Our experiments show that this method can recover very accurately the 3D shape of human faces, cat faces and cars from single-view images, without any supervision or a prior shape model. On benchmarks, we demonstrate superior accuracy compared to another method that uses supervision at the level of 2D image correspondences.

Index Terms—Unsupervised 3D reconstruction, single-image 3D reconstruction, intrinsic image decomposition

1 Introduction

The ability to understand and reconstruct the content of images in 3D is of great importance in many computer vision applications. Yet, when it comes to learning categories of visual objects, for instance to detect and segment them, most approaches model them as 2D patterns [1], with no obvious understanding of their 3D structure. Thus, in this paper we consider the problem of learning categories of 3D deformable objects. Furthermore, we do so under two challenging conditions. The first condition is that *no 2D or 3D ground truth information* (such as keypoints, segmentation, depth maps, or prior knowledge of a 3D model) is available. Learning without external supervisions removes the need for collecting image annotations, which is often a major obstacle to deploying deep learning to new applications. The second condition is that learning can only use an *unconstrained collection of single-view images* — in particular, it does not use multiple views of the same object instance. Learning from single-view images is useful because in many applications we only have a source of independent still images to work with (for example obtained form an Internet search engine).

In more detail, we introduce a new learning algorithm that takes as input a collection of single-view images of a deformable object category and produces as output a deep network that can estimate the 3D shape of any object instance given a single image of it (Fig. 1). The algorithm is based on an autoencoder that internally decomposes the image into albedo, depth, illumination and viewpoint, *without direct supervision for any of these factors*. In general, decomposing images into these four factors is ill-posed. We thus seek for a minimal set of assumptions that makes the problem solvable. To this end, we note that many object categories are *symmetric* (e.g. almost all animals and many handcrafted objects). If an object is perfectly symmetric, mirroring any image of it results in a second virtual view of the object. Furthermore, if point correspondences between the image and its mirrored version can be established, then the 3D shape of the object can be recovered using any of a number of standard multi-view 3D reconstruction approaches [2], [3], [4], [5], [6]. Motivated by this, we seek to leverage symmetry as a cue to constrain this decomposition task.

While symmetry is a powerful cue, using it in practice is far from trivial. First, even if symmetry allows to obtain a pair of virtual views of an object, reconstruction still require to establish point correspondences between them, which can be difficult to do in an unsupervised manner. For instance, the appearance of symmetric points may still differ substantially due to asymmetric illumination. Second, specific object instances are in practice never fully symmetric, neither in shape nor appearance. Shape is non-symmetric due to variations in pose or other details (e.g. the hair style or expressions in a human face), and albedo can also be non-symmetric (e.g. asymmetries in the texture of cat's fur).

We address these issues in two ways. First, we explicitly account for the effect of illumination in the reconstruction pipeline by decomposing the appearance into albedo and shading. In this manner, the model learns to explain

• *The authors are with the Department of Engineering Science, University of Oxford, OX1 2JD Oxford, U.K. E-mail: {szwu, chrisr, vedaldi}@robots.ox.ac.uk.*

Manuscript received 20 Dec. 2020; accepted 16 Apr. 2021. Date of publication 0 . 0000; date of current version 0 . 0000.
(Corresponding author: Shangzhe Wu.)
Recommended for acceptance by Silvio Savarese and Ce Liu.
Digital Object Identifier no. 10.1109/TPAMI.2021.3076536

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

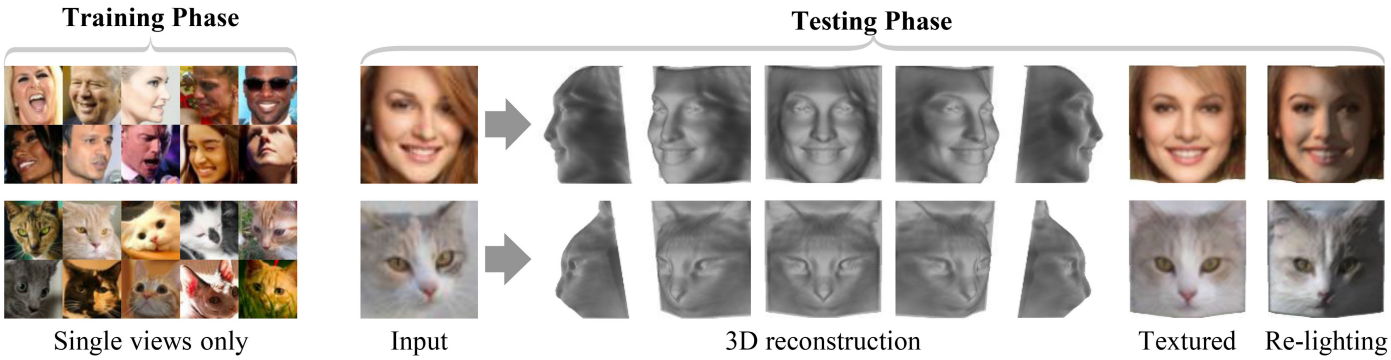


Fig. 1. *Unsupervised learning of 3D deformable objects from in-the-wild images.* Left: Training uses *only* single views of the object category with *no* additional supervision at all (i.e. no ground-truth 3D information, multiple views, or any prior model of the object). Right: Once trained, our model reconstructs the 3D pose, shape, albedo and illumination of a deformable object instance from a single image with excellent fidelity.

asymmetries in the object appearance resulting from illumination, allowing it to better understand how pairs of symmetric views of the object correspond. Moreover, since shading provides information on the surface normals and thus the 3D shape, decomposing it allows the model to explicitly use this information to constrain 3D shapes. Second, we augment the model to reason about potential lack of symmetry in the objects. To do this, the model predicts, along with the factors listed above, a dense map explaining the probability that a given pixel has a symmetric counterpart in the image.

We combine these elements in an end-to-end learning formulation, where all components, including the symmetry probability map, are learned from raw RGB images only. As a further contribution, we show that, rather than enforcing the symmetry by adding further terms to the learning objective, we can instead do so indirectly. The latter is obtained by randomly mirroring the internal representation of the object, thus encouraging the autoencoder to generate a symmetric view of the object. The advantage of this approach is that it avoids the need to introduce and thus tune additional terms in the learning objective.

We test our method on several datasets, including human faces, cat faces and synthetic cars. We provide a thorough ablation study and extensive analyses using a synthetic face dataset with the necessary 3D ground truth. On real images, we achieve higher fidelity reconstruction results compared to other methods [7], [8] that do not rely on 2D or 3D ground truth information, nor prior knowledge of a 3D model of the instance or class. In addition, our method outperforms a recent state-of-the-art method [9] that uses keypoint supervision for 3D reconstruction on real faces, while our method uses no external supervision at all. As a by-product, our method also learns intrinsic image decomposition without any external supervision. Finally, we demonstrate that our trained model generalizes to non-natural images, such as paintings and cartoon drawings, as well as video frames without any fine-tuning.

This article is an extension and archival version of our previous work [10]. In this article, we expand the literature review, provide additional technical details, and include additional experiments and discussions that reveal the important insights of the proposed algorithm, including how it works, how it may fail, and how it compares to prominent model-based methods on 3D reconstruction benchmarks. The code and pretrained models are available at https://github.com/elliottwu/unsup3d.

2 Related Work

In order to assess our contribution in relation to the vast literature on image-based 3D reconstruction, it is important to consider three aspects of each approach: which information is used, which assumptions are made, and what the output is. Below and in Table 1 we compare our contribution to prior works based on these factors.

TABLE 1
Comparison With Selected Prior Work: Supervision, Goals, and Data

Paper	Supervision	Goals	Data
[11]	3D scans	3DMM	Face
[12]	3DV, I	Prior on 3DV	ShapeNet, Ikea
[13]	3DP	Prior on 3DP	ShapeNet
[14]	3DM	Prior on 3DM	Face
[15]	3DMM, 2DKP, I	Refine 3DMM fit to I	Face
[16]	3DMM, 2DKP, I	Fit 3DMM to I+2DKP	Face
[17]	3DMM	Fit 3DMM to 3D scans	Face
[18]	3DMM, 2DKP	Pred. 3DMM	Humans
[19]	3DMM, 2DS+KP	Pred. N, A, L	Face
[20]	3DMM, I	Pred. 3DM, VP, T, E	Face
[21]	3DMM, 2DKP, I	Fit 3DMM to I	Face
[22]	2DS	Prior on 3DV	ModelNet
[23]	2DS	Pred. 3DV	ShapeNet
[24]	I, 2DS, VP	Prior on 3DV	ShapeNet, PAS3D
[25]	I, 2DS+KP	Pred. 3DM, T, VP	Birds
[26]	I, 2DS	Pred. 3DM, T, L, VP	ShapeNet, Birds
[27]	I, 2DS	Pred. 3DV, VP	ShapeNet, others
[28]*	I, 2DS, 3DTM	Fit 3DTM to I	Animals
[29]*	I, 2DS, 3DTM	Pred. 3DM, T, VP	Birds, PAS3D
[30]*	I, 2DS†	Pred. 3DM, T, VP	Birds, PAS3D
[8]	I	Prior on 3DM, T	Face
[31]	I	Prior on 3DV, T	Face, others
[32]*	I	Prior on 3DV, T	ShapeNet, others
[7]	I	Pred. 3DM, VP, T‡	Face
[33]	I	Pred. V, L, VP	ShapeNet
Ours	I	Pred. D, L, A, VP	Face, others

*I: image, 3DMM: 3D morphable model, 3DTM: 3D template model, 2DKP: 2D keypoints, 2DS: 2D silhouette, 3DP: 3D points, VP: viewpoint, E: expression, 3DM: 3D mesh, 3DV: 3D volume, D: depth, N: normals, A: albedo, T: texture, L: light, PAS3D: PASCAL 3D+ [34]. † in the form of part segmentation maps. ‡ can also recover A and L in post-processing. * appear after our original paper was published.*

Our method uses single-view images of an object category as training data, assumes that the objects belong to a specific class (e.g. human faces) which is weakly symmetric, and outputs a monocular predictor capable of decomposing any image of the category into shape, albedo, illumination, viewpoint and symmetry probability.

2.1 Structure From Motion

Traditional methods such as Structure from Motion (SfM) [35] can reconstruct the 3D structure of individual rigid scenes given as input multiple views of each scene and 2D keypoint matches between the views. This can be extended in two ways. First, *monocular reconstruction methods* can perform dense 3D reconstruction from a single image without 2D keypoints [36], [37], [38]. However, they require multiple views [38] or videos of rigid scenes for training [36]. Second, *Non-Rigid SfM* (NRSfM) approaches [39], [40] can learn to reconstruct deformable objects by allowing 3D points to deform in a limited manner between views, but require supervision in terms of annotated 2D keypoints for both training and testing. Hence, neither family of SfM approaches can learn to reconstruct deformable objects from raw pixels of a single view.

2.2 Shape From X

Many other monocular cues have been used as alternatives or supplements to SfM for recovering shape from images, such as shading [41], [42], silhouettes [43], texture [44], symmetry [2], [3] etc. In particular, our work is inspired from *shape from symmetry* and *shape from shading*. Shape from symmetry [2], [3], [4], [5] reconstructs symmetric objects from a single image by using the mirrored image as a virtual second view, provided that symmetric correspondences are available. [5] also shows that it is possible to detect symmetries and correspondences using descriptors. Shape from shading [41], [42] assumes a shading model such as Lambertian reflectance, and reconstructs the surface by exploiting the non-uniform illumination.

2.3 Category-Specific Reconstruction

Learning-based methods have recently been leveraged to reconstruct objects from a single view, either in the form of a raw image or 2D keypoints (see also Table 1). While this task is ill-posed, it has been shown to be solvable by learning a suitable object prior from the training data [11], [12], [13], [14]. A variety of supervisory signals have been proposed to learn such priors. Besides using 3D ground truth directly, authors have considered using videos [36], [45], [46], [47], [48], stereo pairs [38], [49] and multi-view images [50], [51], [52], [53], [54].

Other approaches have used single views with 2D keypoint annotations [9], [25], [55], [56] or object masks [23], [25], [26], [29]. For objects such as human bodies and human faces, some methods [16], [17], [18], [20], [21], [29], [57], [58], [59], [60], [61] have learn to reconstruct from raw images, but starting from the knowledge of a predefined shape model, such as SMPL [62] or Basel [11], or shape templates. These prior models are constructed using specialized hardware and/or other forms of supervision, which are often difficult to obtain for deformable objects in the wild, such as animals, and also limited in details of the shape.

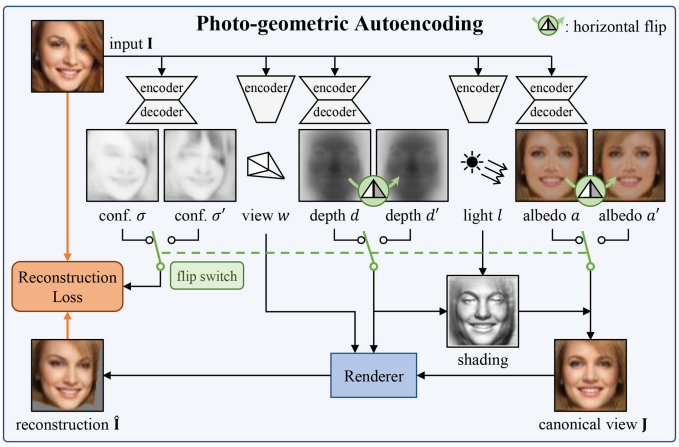


Fig. 2. *Photo-geometric autoencoding*. Our network Φ decomposes an input image $\mathbf{I}$ into depth, albedo, viewpoint and lighting, together with a pair of confidence maps. It is trained to reconstruct the input without external supervision.

Only recently have authors attempted to learn the geometry of object categories from raw, monocular views *only*. Thewlis *et al.* [63], [64] uses equivariance to learn dense landmarks, which recovers the 2D geometry of the objects. DAE [65] learns to predict a deformation field through heavily constraining an autoencoder with a small bottleneck embedding and lift that to 3D in [7] — in post processing, they further decompose the reconstruction in albedo and shading, obtaining an output similar to ours.

Adversarial learning has been proposed as a way of hallucinating new views of an object. Some of these methods start from 3D representations [12], [13], [14], [32], [66]. Kato *et al.* [24] trains a discriminator on raw images but uses viewpoint as addition supervision. HoloGAN [31] only uses raw images but does not obtain an explicit 3D reconstruction. Szabo *et al.* [8] uses adversarial training to reconstruct 3D meshes of the object, but does not assess their results quantitatively. Henzler *et al.* [27] also learns from raw images, but only experiments with images that contain the object on a white background, which is akin to supervision with 2D silhouettes. In Section 4.4, we compare to [7], [8] and demonstrate superior reconstruction results with much higher fidelity.

Since our model generates images from an internal 3D representation, one essential component is a differentiable renderer. However, with a traditional rendering pipeline, gradients across occlusions and boundaries are not defined. Several soft relaxations have thus been proposed [67], [68], [69]. Here, we use a PyTorch implementation[1] of [68].

3 Method

Our learning algorithm, illustrated in Fig. 2, takes as input a collection of independent images of objects of a certain category, such as human or cat faces. It then produces as output a model Φ that, given any new image, recovers the object's 3D shape, albedo, illumination and viewpoint.

As the algorithm has only raw images to learn from, the learning objective is reconstructive: namely, the model is trained so that the combination of the four factors gives

1. https://github.com/daniilidis-group/neural_renderer

back the input image. This results in an auto-encoding pipeline where the factors have, due to the way they are combined to generate an image, an specific photo-geometric interpretation.

Due to the lack of 2D or 3D supervision and of a 3D prior on the possible shapes of the objects, this reconstruction problem is ill-posed. In order to address this issue, we use the fact that many object categories are *bilaterally symmetric*, which provides a strong geometric cue to remove the most severe reconstruction ambiguities. In practice, the appearance of specific object instances is never exactly symmetric due to deformations of the 3D shape and asymmetric details in the shape itself as well as in the illumination and albedo. We take two measures to account for these asymmetries. First, we explicitly model asymmetric illumination. Second, our model also estimates, for each pixel in the input image, a confidence score that explains the probability of the pixel having a symmetric counterpart in the image (denoted as *conf.* σ and σ' in Fig. 2).

The following sections describe how this is done, looking first at the photo-geometric autoencoder (Section 3.1), then at how symmetries are modelled (Section 3.2), followed by details of the image formation (Section 3.3) and the optional use of a perceptual loss (Section 3.4).

3.1 Photo-Geometric Autoencoding

An image $\mathbf{I}$ is a function $\Omega \to \mathbb{R}^3$ defined on a grid $\Omega = \{0, \dots, W-1\} \times \{0, \dots, H-1\}$, or, equivalently, a tensor in $\mathbb{R}^{3\times W\times H}$. We assume that the image is roughly centered on an instance of the object of interest. The goal is to learn a function Φ, implemented as a neural network, that maps the image $\mathbf{I}$ to four factors (d, a, w, l) comprising a *depth map* $d : \Omega \to \mathbb{R}_+$, an *albedo image* $a : \Omega \to \mathbb{R}^3$, a global *light direction* $l \in \mathbb{S}^2$, and a *viewpoint* $w \in \mathbb{R}^6$ so that the image can be reconstructed from them.

The image $\mathbf{I}$ is reconstructed from the four factors in two steps, *lighting* Λ and *reprojection* Π, as follows:

$$\hat{\mathbf{I}} = \Pi(\Lambda(a, d, l), d, w). \tag{1}$$

The lighting function Λ generates a version of the object based on the depth map d, the light direction l and the albedo a as seen from a canonical viewpoint $w = 0$. The viewpoint w represents the transformation between the canonical view and the viewpoint of the actual input image $\mathbf{I}$. Then, the reprojection function Π simulates the effect of a viewpoint change and generates the image $\hat{\mathbf{I}}$ given the canonical depth d and the shaded canonical image $\Lambda(a, d, l)$. Learning uses a reconstruction loss which encourages $\mathbf{I} \approx \hat{\mathbf{I}}$ (Section 3.2).

3.1.1 Discussion

The effect of lighting could be incorporated in the albedo a by interpreting the latter as a texture rather than as the object's albedo. However, there are two good reasons to avoid this. First, the albedo a is often symmetric even if the illumination causes the corresponding appearance to look asymmetric. Separating them allows us to more effectively incorporate the symmetry constraint described below. Second, shading provides an additional cue on the underlying 3D shape [70], [71]. In particular, unlike the recent work of [65] where a shading map is predicted independently from shape, our model computes the shading based on the predicted depth, mutually constraining each other.

3.2 Probably Symmetric Objects

Leveraging symmetry for 3D reconstruction requires identifying symmetric object points in an image. Here we do so implicitly, assuming that depth and albedo, which are reconstructed in a canonical frame, are symmetric about a fixed vertical plane. An important beneficial side effect of this choice is that it helps the model discover a 'canonical view' for the object, which is important for reconstruction [40].

To do this, we consider the operator that flips a map $a \in \mathbb{R}^{C\times W\times H}$ along the horizontal axis:[2] $[\text{flip}a]_{c,u,v} = a_{c,W-1-u,v}$. We then require $d \approx \text{flip}d'$ and $a \approx \text{flip}a'$. While these constraints could be enforced by adding corresponding loss terms to the learning objective, they would be difficult to balance. Instead, we achieve the same effect indirectly, by obtaining a second reconstruction $\hat{\mathbf{I}}'$ from the flipped depth and albedo

$$\hat{\mathbf{I}}' = \Pi(\Lambda(a', d', l), d', w), \quad a' = \text{flip } a, \quad d' = \text{flip } d. \tag{2}$$

Then, we consider two reconstruction losses encouraging $\mathbf{I} \approx \hat{\mathbf{I}}$ and $\mathbf{I} \approx \hat{\mathbf{I}}'$. Since the two losses are commensurate, they are easy to balance and train jointly. Most importantly, this approach allows us to easily reason about symmetry probabilistically, as explained next.

The source image $\mathbf{I}$ and the reconstruction $\hat{\mathbf{I}}$ are compared via the loss

$$\mathcal{L}(\hat{\mathbf{I}}, \mathbf{I}, \sigma) = -\frac{1}{|\Omega|}\sum_{uv\in\Omega} \ln \frac{1}{\sqrt{2}\sigma_{uv}} \exp -\frac{\sqrt{2}\ell_{1,uv}}{\sigma_{uv}}, \tag{3}$$

where $\ell_{1,uv} = |\hat{\mathbf{I}}_{uv} - \mathbf{I}_{uv}|$ is the L_1 distance between the intensity of pixels at location uv, and $\sigma \in \mathbb{R}_+^{W\times H}$ is a *confidence map*, also estimated by the network Φ from the image $\mathbf{I}$, which expresses the *aleatoric uncertainty* of the model. The loss can be interpreted as the negative log-likelihood of a factorized Laplacian distribution on the reconstruction residuals. Optimizing likelihood causes the model to self-calibrate, learning a meaningful confidence map [72].

Modelling uncertainty is generally useful, but in our case is particularly important when we consider the "symmetric" reconstruction $\hat{\mathbf{I}}'$, for which we use the same loss $\mathcal{L}(\hat{\mathbf{I}}', \mathbf{I}, \sigma')$. Crucially, we use the network to estimate, also from the same input image $\mathbf{I}$, a *second* confidence map σ'. This confidence map allows the model to learn which portions of the input image might *not* be symmetric. For instance, in some cases hair on a human face is not symmetric, as shown in Fig. 2, and σ' can assign a higher reconstruction uncertainty to the hair region where the symmetry assumption is not satisfied. Note that this depends on the *specific* instance under consideration, and is learned by the model itself.

Overall, the learning objective is given by the combination of the two reconstruction errors

$$\mathcal{E}(\Phi; \mathbf{I}) = \mathcal{L}(\hat{\mathbf{I}}, \mathbf{I}, \sigma) + \lambda_{\text{f}}\mathcal{L}(\hat{\mathbf{I}}', \mathbf{I}, \sigma'), \tag{4}$$

2. The choice of axis is arbitrary as long as it is fixed.

where $\lambda_{\mathrm{f}} = 0.5$ is a weighing factor, $(d, a, w, l, \sigma, \sigma') = \Phi(\mathbf{I})$ is the output of the neural network, and $\hat{\mathbf{I}}$ and $\hat{\mathbf{I}}'$ are obtained according to Eqs. (1) and (2).

3.3 Image Formation Model

We now describe the functions Π and Λ in Eq. (1) in more detail. The image is formed by a camera looking at a 3D object. If we denote with $P = (P_x, P_y, P_z) \in \mathbb{R}^3$ a 3D point expressed in the reference frame of the camera, this is mapped to pixel $p = (u, v, 1)$ by the following projection:

$$p \propto KP, \quad K = \begin{bmatrix} f & 0 & c_u \\ 0 & f & c_v \\ 0 & 0 & 1 \end{bmatrix}, \quad \begin{cases} c_u = \frac{W-1}{2}, \\ c_v = \frac{H-1}{2}, \\ f = \frac{W-1}{2\tan\frac{\theta_{\mathrm{FOV}}}{2}}. \end{cases} \tag{5}$$

This model assumes a perspective camera with *field of view* (FOV) θ_{FOV}. We assume a nominal distance of the object from the camera at about $1\mathrm{m}$. Given that the images are cropped around a particular object, we assume a relatively narrow FOV of $\theta_{\mathrm{FOV}} \approx 10^\circ$.

The depth map $d : \Omega \to \mathbb{R}_+$ associates a depth value d_{uv} to each pixel $(u, v) \in \Omega$ in the canonical view. By inverting the camera model (5), we find that this corresponds to the 3D point $P = d_{uv} \cdot K^{-1}p$.

The viewpoint $w \in \mathbb{R}^6$ represents an euclidean transformation $(R, T) \in SE(3)$, where $w_{1:3}$ and $w_{4:6}$ are rotation angles and translations along x, y and z axes respectively.

The map (R, T) transforms 3D points from the canonical view to the actual view. Thus a pixel (u, v) in the canonical view is mapped to the pixel (u', v') in the actual view by the warping function $\eta_{d,w} : (u, v) \mapsto (u', v')$ given by

$$p' \propto K(d_{uv} \cdot RK^{-1}p + T), \tag{6}$$

where $p' = (u', v', 1)$.

Finally, the reprojection function Π takes as input the depth d and the viewpoint change w and applies the resulting warp to the canonical image $\mathbf{J}$ to obtain the actual image $\hat{\mathbf{I}} = \Pi(\mathbf{J}, d, w)$ as $\hat{\mathbf{I}}_{u'v'} = \mathbf{J}_{uv}$, where $(u, v) = \eta_{d,w}^{-1}(u', v')$. Note that this requires to compute the *inverse* of the warp $\eta_{d,w}$, which is detailed in Section 3.5.

The canonical image $\mathbf{J} = \Lambda(a, d, l)$ is in turn generated as a combination of albedo, normal map and light direction. To do so, given the depth map d, we derive the normal map $n : \Omega \to \mathbb{S}^2$ by associating to each pixel (u, v) a vector normal to the underlying 3D surface. In order to find this vector, we compute the vectors t_{uv}^u and t_{uv}^v tangent to the surface along the u and v directions. For example, the first one is

$$t_{uv}^u = d_{u+1,v} \cdot K^{-1}(p + e_x) - d_{u-1,v} \cdot K^{-1}(p - e_x), \tag{7}$$

where p is defined above and $e_x = (1, 0, 0)$. Then, the normal is obtained by taking the vector product $n_{uv} \propto t_{uv}^u \times t_{uv}^v$.

The normal n_{uv} is multiplied by the light direction l to obtain a value for the directional illumination and the latter is added to the ambient light. Finally, the result is multiplied by the albedo to obtain the illuminated texture, as follows:

$$\mathbf{J}_{uv} = (k_s + k_d \max\{0, \langle l, n_{uv} \rangle\}) \cdot a_{uv}. \tag{8}$$

Here k_s and k_d are the scalar coefficients weighting the ambient and diffuse terms, and are predicted by the model with range between 0 and 1 via rescaling a `tanh` output. The light direction $l = (l_x, l_y, 1)^T / (l_x^2 + l_y^2 + 1)^{0.5}$ is modeled as a spherical sector by predicting l_x and l_y with `tanh`.

3.4 Perceptual Loss

The L_1 loss function Eq. (3) is sensitive to small geometric imperfections and tends to result in blurry reconstructions. We add a *perceptual loss* term to mitigate this problem. The kth layer of an off-the-shelf image encoder e (VGG16 in our case [73]) predicts a representation $e^{(k)}(\mathbf{I}) \in \mathbb{R}^{C_k \times W_k \times H_k}$ where $\Omega_k = \{0, \dots, W_k - 1\} \times \{0, \dots, H_k - 1\}$ is the corresponding spatial domain. Note that this feature encoder does not have to be trained with supervised tasks. Self-supervised encoders can be equally effective as shown in Table 3.

Similar to Eq. (3), assuming a Gaussian distribution, the perceptual loss is given by

$$\mathcal{L}_{\mathrm{p}}^{(k)}(\hat{\mathbf{I}}, \mathbf{I}, \sigma^{(k)}) = -\frac{1}{|\Omega_k|} \sum_{uv \in \Omega_k} \ln \frac{1}{\sqrt{2\pi(\sigma_{uv}^{(k)})^2}} \exp - \frac{(\ell_{uv}^{(k)})^2}{2(\sigma_{uv}^{(k)})^2}, \tag{9}$$

where $\ell_{uv}^{(k)} = |e_{uv}^{(k)}(\hat{\mathbf{I}}) - e_{uv}^{(k)}(\mathbf{I})|$ for each pixel index uv in the kth layer. We also compute the loss for $\hat{\mathbf{I}}'$ using $\sigma^{(k)'}$. $\sigma^{(k)}$ and $\sigma^{(k)'}$ are additional confidence maps predicted by our model. In practice, we found it is good enough for our purpose to use the features from only one layer (`relu3_3`) of VGG16. We therefore shorten the notation of perceptual loss to $\mathcal{L}_{\mathrm{p}}$. With this, the loss function $\mathcal{L}$ in Eq. (4) is replaced by $\mathcal{L} + \lambda_{\mathrm{p}}\mathcal{L}_{\mathrm{p}}$ with $\lambda_{\mathrm{p}} = 1$.

3.5 Differentiable Rendering Layer

As noted in Section 3.3, the reprojection function Π *warps* the canonical image $\mathbf{J}$ to generate the actual image $\mathbf{I}$. In CNNs, image warping is usually regarded as a simple operation that can be implemented efficiently using a bilinear resampling layer [74]. However, this is true only if we can easily send pixels (u', v') in the warped image $\mathbf{I}$ back to pixels (u, v) in the source image $\mathbf{J}$, a process also known as *backward warping*. Unfortunately, in our case the function $\eta_{d,w}$ obtained by Eq. (6) sends pixels the opposite way.

Implementing a *forward warping* layer is surprisingly delicate. One way of approaching the problem is to regard this task as a special case of rendering a textured mesh. The *Neural Mesh Renderer* (NMR) of [68] is a differentiable renderer of this type. In our case, the mesh has one vertex per pixel and each group of 2×2 adjacent pixels is tessellated by two triangles. Empirically, we found the quality of the texture gradients of NMR to be poor in this case, likely caused by noisy depth map d and high frequency content in the texture image $\mathbf{J}$.

We solve the problem as follows. First, we use NMR to warp only the depth map d, obtaining a version $\bar{d}$ of the depth map as seen from the input viewpoint. This has two advantages: backpropagation through NMR is faster and second, the depth gradients are more stable than color gradients, probably also due to the comparatively smooth nature of the depth map d compared to the texture image $\mathbf{J}$.

Given the depth map $\bar{d}$, we then use the inverse of Eq. (6) to find the warp field from the observed viewpoint to the canonical viewpoint, and bilinearly resample the canonical image $\mathbf{J}$ to obtain the reconstruction (i.e. using backward warping).

4 Experiments

In this section, we first describe the experimental setup and implementation details, and then present the qualitative results on three object categories, human faces, cat faces and synthetic cars, followed by extensive ablation studies and analyses. We also report comparisons with several state-of-the-art methods both qualitatively and quantitatively. In the end, we provide a discussion on the limitations of our method.

4.1 Experimental Setup

4.1.1 Datasets

We test our method on three human face datasets: *CelebA* [75], *3DFAW* [76], [77], [78], [79] and *BFM* [11]. CelebA is a large scale human face dataset, consisting of over 200k images of real human faces in the wild annotated with bounding boxes. 3DFAW contains 23k images with 66 3D keypoint annotations, which we use to evaluate our 3D predictions in Section 4.4. We roughly crop the images around the head region using MTCNN [80] and use the official train/val/test splits. BFM (Basel Face Model) is a synthetic face model, which we use to assess the quality of the 3D reconstructions (since the in-the-wild datasets lack ground-truth). We follow the protocol of [19] to generate a dataset, sampling shapes, poses, textures, and illumination randomly. We use images from SUN Database [81] as background and save ground truth depth maps for evaluation.

We also test our method on cat faces and synthetic cars. We use two *cat datasets* [82], [83]. The first one has 10k cat images with nine keypoint annotations, and the second one is a collection of dog and cat images, containing 1.2k cat images with bounding box annotations. We combine the two datasets and crop the images around the cat heads. For cars, we render 35k images of synthetic cars from *ShapeNet* [84] with random viewpoints and illumination. We randomly split the images by 8:1:1 into train, validation and test sets.

4.1.2 Metrics

Since the scale of 3D reconstruction from projective cameras is inherently ambiguous [35], we discount it in the evaluation. Specifically, given the depth map d predicted by our model in the canonical view, we warp it to a depth map $\bar{d}$ in the actual view using the predicted viewpoint and compare the latter to the ground-truth depth map d^* using the *scale-invariant depth error* (SIDE) [85]

$$E_{\text{SIDE}}(\bar{d}, d^*) = \left(\frac{1}{WH} \sum_{uv} \Delta_{uv}^2 - \left(\frac{1}{WH} \sum_{uv} \Delta_{uv} \right)^2 \right)^{\frac{1}{2}}, \qquad (10)$$

where $\Delta_{uv} = \log \bar{d}_{uv} - \log d^*_{uv}$. We compare only valid depth pixels and erode the foreground mask by one pixel to discount rendering artefacts at object boundaries. Additionally, we report the *mean angle deviation* (MAD) between normals computed from ground truth depth and from the predicted depth, measuring how well the surface is captured.

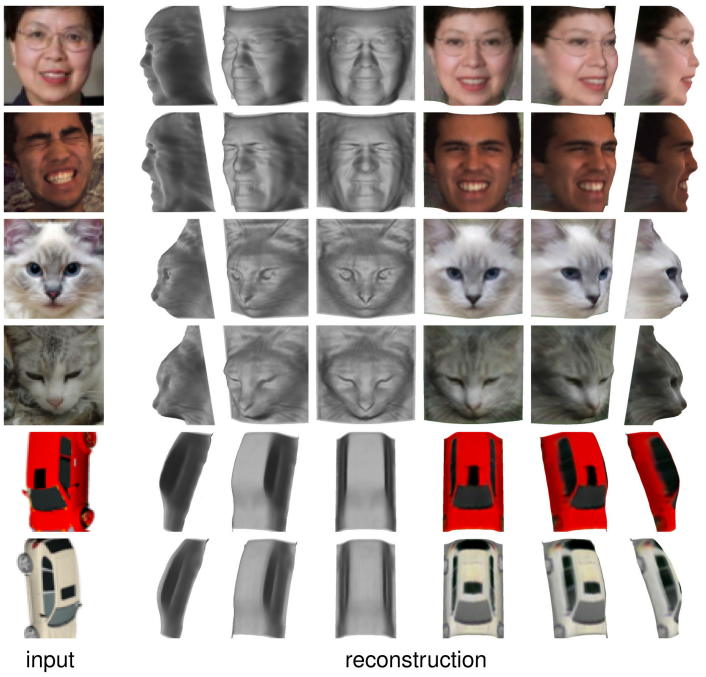


Fig. 3. *Reconstruction of faces, cats and cars.* Our unsupervised model recovers accurate 3D shape from only a single input image.

4.1.3 Implementation Details

The function $(d, a, w, l, \sigma) = \Phi(\mathbf{I})$ that preditcs depth, albedo, viewpoint, lighting, and confidence maps from the image $\mathbf{I}$ is implemented using individual neural networks. The depth and albedo are generated by encoder-decoder networks, while viewpoint and lighting are regressed using simple encoder networks. The encoder-decoders do not use skip connections because input and output images are *not* spatially aligned (since the output is in the canonical viewpoint). All four confidence maps are predicted using the same network, at different decoding layers for the photometric and perceptual losses since these are computed at different resolutions. The final activation function is `tanh` for depth, albedo, viewpoint and lighting and `softplus` for the confidence maps. The depth prediction is centered on the mean before `tanh`, as the global distance is estimated as part of the viewpoint. We do *not* use any special initialization for all predictions, except that two border pixels of the depth maps on both the left and the right are clamped at a maximal depth to avoid boundary issues.

We train using Adam over batches of 64 input images, resized to 64×64 pixels. The size of the output depth and albedo is also 64×64. We train for approximately 50k iterations. For visualization, depth maps are upsampled to 256. We include more details in the supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2021.3076536.

4.2 Qualitative Results

4.2.1 Reconstruction Results

In Fig. 3 we show reconstruction results of human faces from CelebA and 3DFAW, cat faces from [82], [83] and synthetic cars from ShapeNet. The 3D shapes are recovered

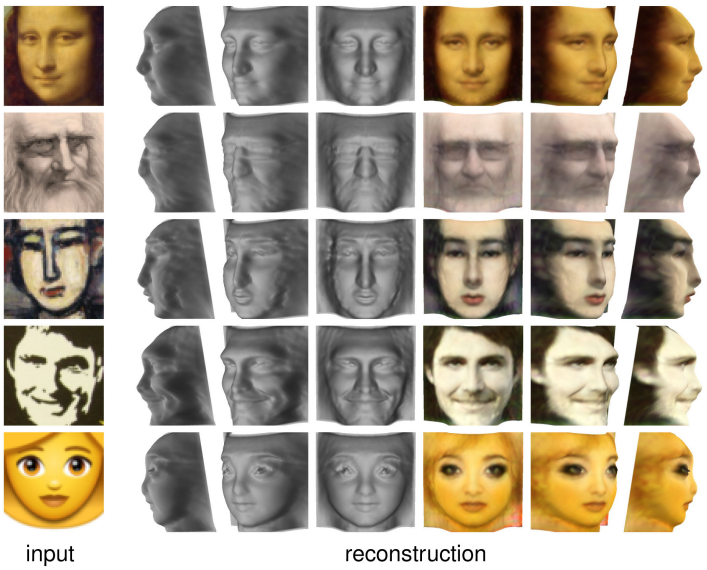


Fig. 4. *Reconstruction of faces in paintings and cartoons.* The model trained on real faces in CelebA generalizes well to abstract faces in paintings and cartoons.

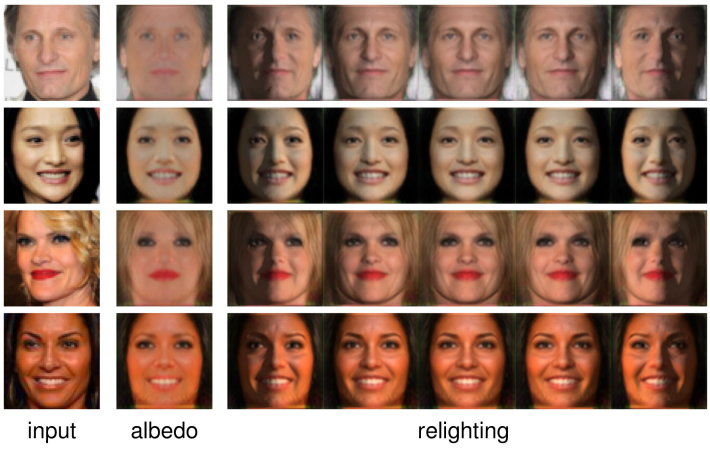


Fig. 5. *Re-lighting results.* Our model disentangles albedo and shading from a single input image, which allows us to relight the objects with novel lighting conditions.

with high fidelity. The reconstructed 3D face, for instance, contain fine details of the nose, eyes and mouth even in the presence of extreme facial expression.

4.2.2 Generalization to Paintings

To further test generalization, we applied our model trained on the CelebA dataset to a number of paintings and cartoon drawings of faces collected from [86] and the Internet. As shown in Fig. 4, our method still works well even though it has never seen such images during training. It is worth noting that since the model is trained using real face images, the reconstructions seem to also be more "realistic" faces reflecting the prior learned during training.

4.2.3 Relighting

A by-product of our reconstruction framework is that it learns to disentangle albedo and shading from a single image, without any external supervision at all. This is possible by leveraging the symmetry assumption on the albedo as well as the categorical prior imposed by training set.



Fig. 6. *Frame-by-frame reconstruction on video sequences.* Even though our model does not use videos for training, it produces temporally consistent reconstructions on video sequences.

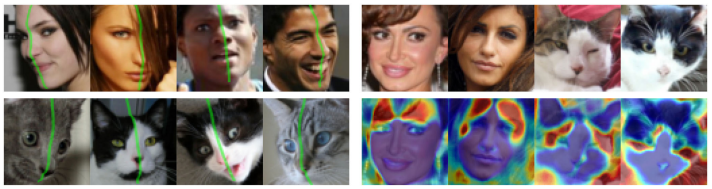


Fig. 7. *Symmetry plane and asymmetry detection.* (a): our model can reconstruct the "intrinsic" symmetry plane of an in-the-wild object even though the appearance is highly asymmetric. (b): asymmetries (highlighted in red) are detected and visualized using confidence map σ'.

Decomposing the albedo map enables realistic graphics editing applications, such as re-rendering the object under different lighting conditions, as illustrated in Fig. 5.

4.2.4 Inference on Video Frames

We can also apply our trained model on video sequences frame-by-frame. To demonstrate this, we take speech clips from VoxCeleb [87], and crop the faces using MTCNN[3] [80]. We then feed the crops to our model to produce a 3D reconstruction of the faces and render them from novel viewpoints, shown in Fig. 6. Note that our model does not use videos for training, yet it produces temporally consistent and accurate reconstruction results by simply processing the frames independently.

4.2.5 Symmetry and Asymmetry Detection

Since our model predicts a canonical view of the objects that is symmetric about the vertical center-line of the image, we can easily visualize the symmetry plane, which is otherwise non-trivial to detect from in-the-wild images. In Fig. 7, we warp the center-line of the canonical image to the predicted input viewpoint. Our method can detect symmetry planes accurately despite the presence of asymmetric texture and lighting

3. We use the implementation from https://github.com/timesler/facenet-pytorch.

TABLE 2
Comparison With Baselines

No	Baseline	SIDE ($\times 10^{-2}$) ↓	MAD (deg.) ↓
(1)	Supervised	0.410 ± 0.103	10.78 ± 1.01
(2)	Const. null depth	2.723 ± 0.371	43.34 ± 2.25
(3)	Average g.t. depth	1.990 ± 0.556	23.26 ± 2.85
(4)	Ours (unsupervised)	0.793 ± 0.140	16.51 ± 1.56

SIDE and MAD errors of our reconstructions on the BFM dataset compared against a fully-supervised and trivial baselines.

TABLE 3
Ablation Study

No	Method	SIDE ($\times 10^{-2}$) ↓	MAD (deg.) ↓
(1)	Ours full	0.793 ± 0.140	16.51 ± 1.56
(2)	w/o albedo flip	2.916 ± 0.300	39.04 ± 1.80
(3)	w/o depth flip	1.139 ± 0.244	27.06 ± 2.33
(4)	w/o light	2.406 ± 0.676	41.64 ± 8.48
(5)	w/o perc. loss	0.931 ± 0.269	17.90 ± 2.31
(6)	w/ self-sup. perc. loss	0.815 ± 0.145	15.88 ± 1.57
(7)	w/o confidence	0.829 ± 0.213	16.39 ± 2.12

Refer to Section 4.3.2 for details.

effects. We also overlay the predicted confidence map σ' onto the image, confirming that the model assigns low confidence to asymmetric regions in a sample-specific way.

4.3 Analyses and Discussions

4.3.1 Comparison With Baselines

Table 2 uses the BFM dataset to compare the depth reconstruction quality obtained by our method, a fully-supervised baseline and two other baselines. The supervised baseline is a version of our model trained to regress the ground-truth depth maps using an L_1 loss. The trivial baseline predicts a constant uniform depth map, which provides a performance lower-bound. The third baseline is a constant depth map obtained by averaging all ground-truth depth maps in the test set. Our method largely outperforms the two constant baselines and approaches the results of supervised training.

4.3.2 Ablation

To understand the influence of the individual parts of the model, we remove them one at a time and evaluate the performance of the ablated model in Table 3 and Fig. 8.

In the table, *row (1)* shows the performance of the full model (the same as in Table 2). *Row (2)* does not flip the albedo. Thus, the albedo is not encouraged to be symmetric in the canonical space, which fails to canonicalize the viewpoint of the object and to use cues from symmetry to recover shape. The performance is as low as the trivial baseline in Table 2. *Row (3)* does not flip the depth, with a similar effect to row (2). In addition, we had to add an L2 smoothness loss on the depth maps during training. Otherwise, the model tends to produce noisy depth maps without the symmetry constraint, which lead to heavy occlusion and break the training.

Row (4) predicts a shading map instead of computing it from depth and light direction. This also harms performance

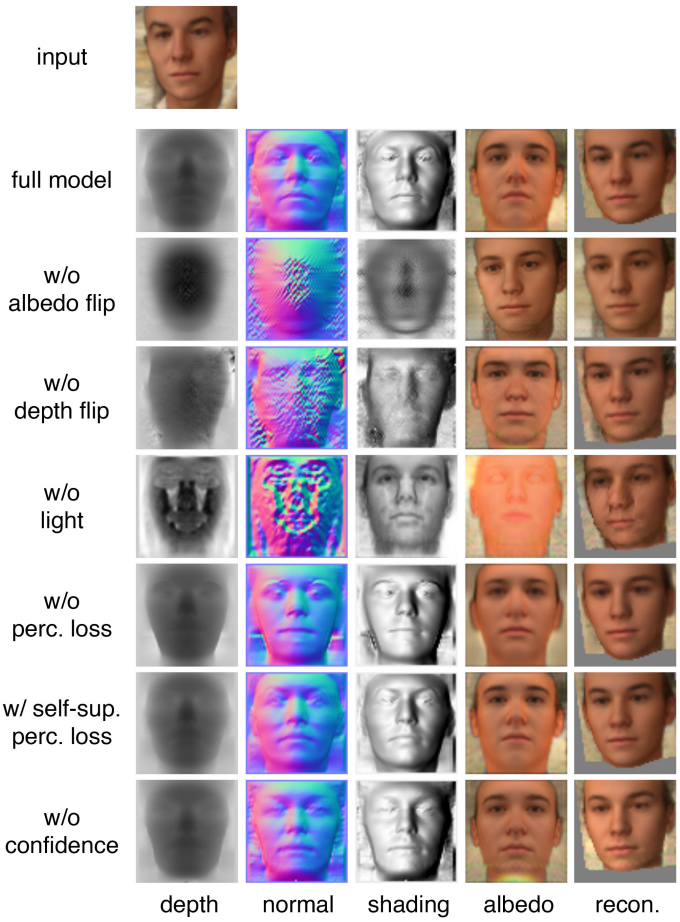


Fig. 8. *Ablation study.* Refer to Section 4.3.2 for details.

significantly because shading cannot be used as a cue to recover shape. Moreover, the training often collapses after a few epochs as the model produces spikes in the depth maps that also result in large occlusion. We therefore report the results of the latest epoch prior to collapse.

Row (5) switches off the perceptual loss, which leads to degraded image quality and hence degraded reconstruction results. *Row (6)* replaces the ImageNet pretrained image encoder used in the perceptual loss with one[4] trained through a self-supervised task [88], which shows no difference in performance.

Finally, *row (7)* switches off the confidence maps, using a fixed and uniform value for the confidence — this reduces losses (3) and (9) to the basic L_1 and L_2 losses, respectively. The accuracy does not drop significantly, as faces in BFM are highly symmetric (e.g. do not have hair), but its variance increases. To better understand the effect of the confidence maps, we specifically evaluate on partially asymmetric faces using perturbations.

4.3.3 Asymmetric Perturbation

In order to demonstrate that our uncertainty modelling allows the model to handle asymmetry, we add asymmetric perturbations to BFM. Specifically, we generate random rectangular color patches with 20 to 50 percent of the image size and blend them onto the images with α-values ranging from 0.5 to 1, as shown in Fig. 9. We then train our model

4. We use a RotNet [88] pretrained VGG16 model obtained from https://github.com/facebookresearch/DeeperCluster.

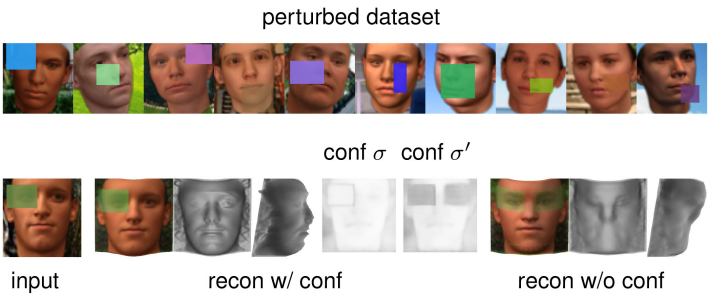


Fig. 9. *Asymmetric perturbation.* Top: examples of the perturbed dataset. Bottom: reconstructions with and without confidence maps. Confidence allows the model to correctly reconstruct the 3D shape with the asymmetric texture.

TABLE 4
Asymmetric Perturbation

	SIDE ($\times 10^{-2}$) ↓	MAD (deg.) ↓
No perturb, no conf.	0.829 ±0.213	16.39 ±2.12
No perturb, conf.	0.793 ±0.140	16.51 ±1.56
Perturb, no conf.	2.141 ±0.842	26.61 ±5.39
Perturb, conf.	0.878 ±0.169	17.14 ±1.90

We add asymmetric perturbations to BFM and show that confidence maps allow the model to reject such noise, while the vanilla model without confidence maps breaks.

with and without confidence on these perturbed images, and report the results in Table 4. Without the confidence maps, the model always predicts a symmetric albedo and geometry reconstruction often fails. With our confidence estimates, the model is able to reconstruct the asymmetric faces correctly, with very little loss in accuracy compared to the unperturbed case.

4.3.4 Training Only on Frontal Faces

Our full training data consists of single-view images of many instances, each captured from a different viewpoint, which essentially compose a large "multi-view" image set, although these are "multi-views" of different instances with different texture and shape. Nonetheless, it would be interesting to know how much this "multi-view" signal contributes to the learning, compared to other cues, such as symmetry and shading.

In order to understand this, we generate another synthetic face dataset consisting of only frontal faces with random texture and shape variations, and train a model on only frontal faces. We compare the performance of this model to our full model trained on the original dataset of images with various viewpoints in Table 5 and Fig. 10. In fact, the model trained on only frontal faces is indeed able to learn 3D shape of frontal faces, despite producing artifacts and a lower reconstruction accuracy compared to the full model. This suggests the symmetry and shading constraints can still provide powerful signals for learning shapes, even without the view variation in the training set. However, this model fails to generalize to input faces from other viewpoints.

4.3.5 Training With Fewer Images

As the symmetry assumption and shading seem to provide strong signals for learning the shape, another interesting question to ask is: does it still need to be trained on a large image collection? To answer this question, we train the model on different numbers of training images, ranging from only one single image to the entire training set of 155k images, and compare the results in Fig. 11. When training with 1 image and 100 images, we added a L2 smoothness loss on the depth maps, as the training otherwise collapses due to noisy depth maps.

As shown in Fig. 11, when trained on only 1 image, the model seems still able to pick up some shading and symmetry cues to recover the 3D shape. However, these cues alone cannot provide enough constraints on this heavily ill-posed 2D-to-3D task. Therefore, although the image reconstruction loss is low, the underlying 3D shape is poorly reconstructed. The model only starts to learn reasonable 3D faces when trained on 1000 or more images, which suggests that a sufficiently large image collection is critical for the model to learn a 3D shape prior of the object category.

TABLE 5
Training on Frontal Faces

	SIDE ($\times 10^{-2}$) ↓	MAD (deg.) ↓
Train frontal, test frontal	1.347 ±0.150	22.90 ±1.13
Train frontal, test all	1.858 ±0.429	30.80 ±3.93
Train all, test frontal	0.818 ±0.107	15.90 ±1.22
Train all, test all	0.793 ±0.140	16.51 ±1.56

We compare the model trained on only frontal faces with our full model trained on faces with all random viewpoints.

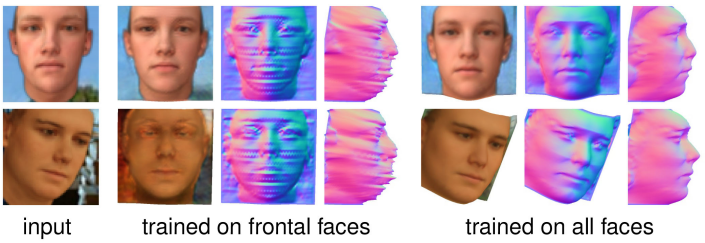


Fig. 10. *Training only on frontal faces.* The model trained on only frontal faces is still able to learn 3D shape, despite producing artifacts (first row), but it does not generalize to other views (second row).

4.3.6 Mixing Categories

In order to understand whether the model learns different priors for different categories, we further conduct experiments on cross-category inference as well as multi-category training. Fig. 12 shows some examples. We first feed images of human faces to a model trained on images of cat images and also the other way around. Unsurprisingly, the models trained on one single category learn shape priors specific to that particular category, and tend to reconstruct shapes of the training category, even if the input images depict a different category.

We further consider training the model on a mixture of images from two object categories, which turns out still capable of reconstructing both categories with similar quality compared to the models trained individually on each category. This observation shows promise of learning a general modal independent of object categories in the future.

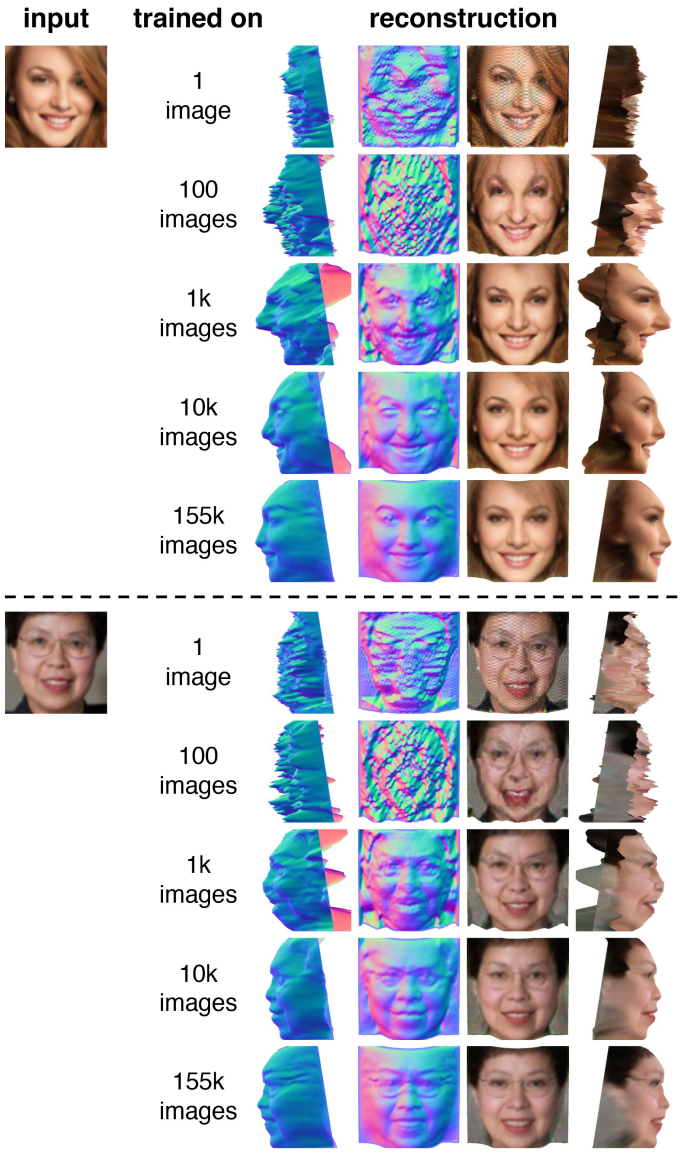


Fig. 11. *Training with fewer images*. We show a qualitative comparison of the models trained with different numbers of images, which confirms the necessity of training on a sufficiently large image collection.

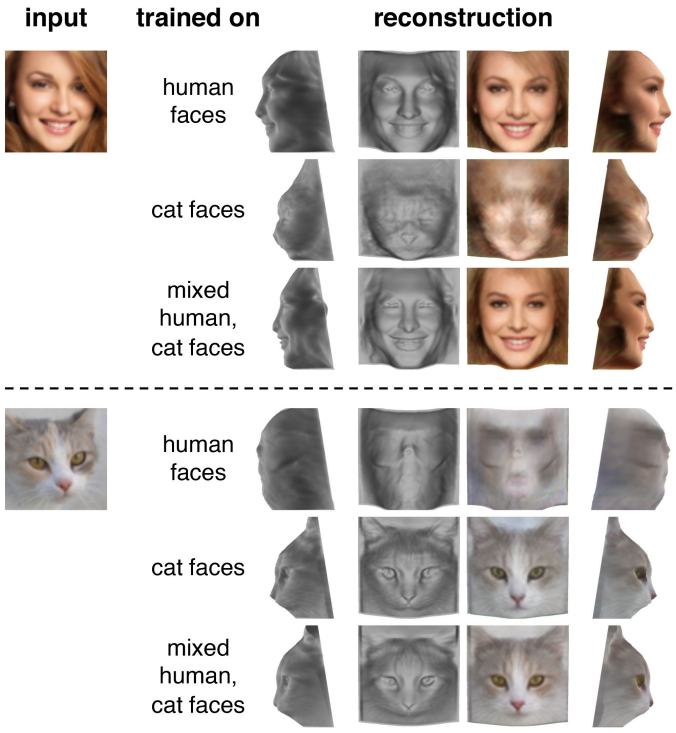


Fig. 12. *Mixing categories*. When trained on one single category, the model learns a prior specific to that particular category, whereas when trained on two categories, it is able to reconstruct both categories well.

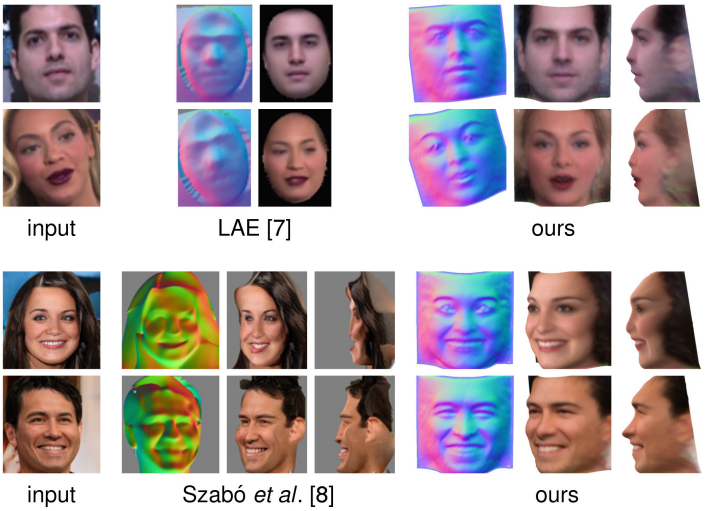


Fig. 13. *Qualitative comparison to SOTA*. Comparing to [7], [8], our method recovers higher quality shapes.

4.4 Comparison With the State of the Art

As shown in Table 1, most reconstruction methods in the literature require either image annotations, prior 3D models or both. When these assumptions are dropped, the task becomes considerably harder, and there is little prior work that is directly comparable. Of these, [33] only uses synthetic, texture-less objects from ShapeNet, [8] reconstructs in-the-wild faces but does not report any quantitative results, and [7] reports quantitative results only on keypoint regression, but not on the 3D reconstruction quality. We were not able to obtain code or trained models from [7], [8] for a direct quantitative comparison and thus compare qualitatively.

4.4.1 Qualitative Comparison

In order to establish a side-by-side comparison, we cropped the figures reported in the papers [7], [8] and compare our results with theirs (Fig. 13). Our method produces higher quality reconstructions than both methods, with fine details of the facial expression. The difference is especially noticeable in the recovery of 3D shape for [7], and the shape generation in [8]. Note that [8] uses an unconditional GAN that *generates* high resolution 3D faces from random noise, and cannot *recover* 3D shapes from images. The input images for [8] in Fig. 13 were generated by their GAN.

4.4.2 3D Keypoint Depth Evaluation

Next, we compare to the DepthNet model of [9]. This method predicts depth for selected facial keypoints, but uses 2D keypoint annotations as input — a much easier setup than the one we consider here. Still, we compare the quality of the reconstruction of these sparse point obtained by DepthNet and our method. We also compare to the baselines MOFA [90] and AIGN [89] reported in [9]. For a fair comparison, we use their public code which computes the

TABLE 6
3DFAW Keypoint Depth Evaluation

	Depth Corr. ↑
Ground truth	66
AIGN [89] (**supervised**, from [9])	50.81
DepthNetGAN [9] (**supervised**, from [9])	58.68
MOFA [90] (**model-based**, from [9])	15.97
DepthNet [9] (from [9])	26.32
DepthNet [9] (from GitHub)	35.77
Ours	48.98
Ours (w/ CelebA pre-training)	54.65

Depth correlation between ground truth and prediction evaluated at 66 facial keypoint locations.

depth correlation score (between 0 and 66) on the frontal faces. We use the 2D keypoint locations to sample our predicted depth and then evaluate the same metric. The set of test images from 3DFAW and the preprocessing are identical to [9]. Since 3DFAW is a small dataset with limited variation, we also report results with CelebA pre-training.

In Table 6 we report the results from their paper and the slightly improved results we obtained from their publicly-available implementation. The paper also evaluates a supervised model using a GAN discriminator trained with ground-truth depth information. While our method does not use any supervision, it still outperforms DepthNet and reaches close-to-supervised performance.

4.4.3 3D Face Reconstruction Benchmarks

We also evaluate the reconstructed 3D meshes and compare the performance with several recent 3DMM-based reconstruction methods [21], [57], [58], [59], [60], [61] on two 3D face reconstruction benchmarks [21], [91].

The first benchmark by Feng *et al.* [91] provides a test set, which consists of 133 ground-truth 3D scans and 2,000 test images, including 656 high-quality (HQ) images captured in a controlled environment and 1,344 low-quality (LQ) images extracted from videos. The second one, NoW benchmark [21], provides a test set of 1,702 images of 80 subjects and a ground-truth 3D scan per subject. These images are captured with a higher variety in facial expression, occlusion, and lighting, compared to the Feng *et al*. benchmark.

However, it is important to highlight that these benchmarks are designed specifically for evaluating 3DMM-based face reconstruction methods, and inherently put model-free approaches at a disadvantage. In both of these benchmark sets, only 3D scans of neutral faces are available, which are used as ground-truth for various input images that describe different viewpoints and facial expressions and may contain occlusion. This gives the 3DMM-based methods an advantage over our method, since the output of these methods is always constrained to a face model regardless of input variety, whereas our method produces instance-specific reconstructions with different expressions, which are not captured in the ground-truth scans. Our main intention with this evaluation is the establishment of a fair, quantitative evaluation of future model-free methods, since qualitative comparisons are often subjective and synthetic benchmarks are limited in terms of generalization to real data.

TABLE 7
Performance on Feng *et al.* [91] Benchmark

Methods	Median ↓		Mean ↓		Std	
	LQ	HQ	LQ	HQ	LQ	HQ
Extreme3D [58]	2.40	2.37	3.49	3.58	6.15	6.75
3DMM-CNN [57]	1.88	1.85	2.32	2.29	1.89	1.88
PRNet [59]	1.79	1.60	2.38	2.06	2.19	1.79
RingNet [21]	1.63	1.58	2.08	2.02	1.79	1.69
3DDFA-V2 [60]	1.62	1.49	2.10	1.91	1.87	1.64
DECA [61]	1.48	1.44	1.91	1.89	1.68	1.66
Const. flat plane	12.47	12.47	14.11	14.07	10.21	10.17
Ours (**model-free**)	5.58	5.54	5.74	5.68	1.47	1.89

We compare our model-free unsupervised method with several recent 3DMM-based methods.

For both datasets, we detect faces and crop the images using MTCNN [80] and obtain 3D mesh reconstructions from the depthmaps predicted by our model trained on CelebA. We then use the same evaluation protocol in both benchmarks [21], [91], which align the predicted meshes with the ground-truth meshes with a rigid transformation based on 7 pre-defined keypoints and compute the scan-to-mesh distances. We obtain these keypoints on our predicted meshes by applying a facial keypoint detector [92] on the reconstructed canonical images. The average keypoints are used when the keypoint detector fails.

We report the statistics of the distances and compare them with other methods in Tables 7 and 8. Although our model-free unsupervised method does not perform as well as the model-based methods on these benchmarks, it is significantly better than a flat shape baseline as shown in Table 7. Since the NoW dataset provides attributes for the images, we select a subset of the test set that contains 91 frontal neutral faces, which better match with the ground-truth scans, and include the results in Table 8. The results in this subset further reduce the gap towards model-based methods.

4.5 Limitations

While our unsupervised method is robust in many challenging scenarios (e.g., extreme facial expression, drawings), we do observe limitations as shown in Fig. 14.

TABLE 8
Performance on NoW *et al.* [21] Benchmark

Methods	Median ↓	Mean ↓	Std
3DMM-CNN [57]	1.84	2.33	2.05
PRNet [59]	1.50	1.98	1.88
RingNet [21]	1.21	1.54	1.31
3DDFA-V2 [60]	1.23	1.57	1.39
DECA [61]	1.09	1.38	1.18
Ours (**model-free**)	2.64	3.29	2.86
3DMM-CNN [57] (frontal)	1.88	2.36	2.07
PRNet [59] (frontal)	1.38	1.79	1.67
RingNet [21] (frontal)	1.16	1.48	1.28
Ours (**model-free**, frontal)	2.25	2.80	2.44

We compare our model-free unsupervised method with several recent 3DMM-based methods. The bottom half of the table reports the results on a subset of frontal neutral faces, indicated by "frontal".

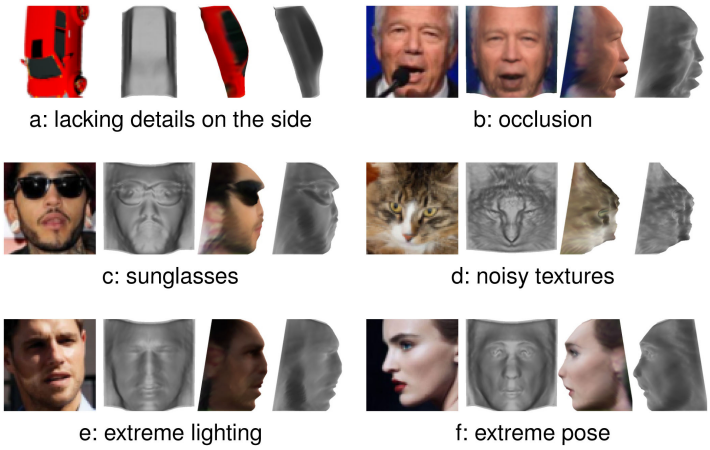


Fig. 14. *Limitations.* See Section 4.5 for details.

First and foremost, our model relies on the assumption that object category has weakly symmetric 3D shape as well as weakly symmetric albedo. Extending the key insights in this work, including leveraging category priors, other forms of symmetry and shape from shading in a learning framework, to general objects will require future work.

In this work, we represent shape using a depth map in the canonical (symmetric) viewpoint, which cannot describe the full 3D shape in 360 degrees. Thus, the reconstructed shapes often lack details on the sides. This is particularly evident for the cars, as illustrated in Fig. 14a. One would need to consider using other 3D representations to capture full 3D objects from 360 degrees.

Our model also tends to ignore occluders (Fig. 14b), since the training set does not contain many examples with occlusion. Disentangling dark textures and shading is often difficult. Therefore, the model fails to accurately reconstruct sunglasses (Fig. 14c) and may produce bumpy surfaces when the texture is noisy (Fig. 14d). During training, we assume a simple Lambertian shading model, ignoring shadows and specularity, which leads to inaccurate reconstructions under extreme lighting conditions (Fig. 14e) or highly non-Lambertian surfaces. The reconstruction quality is also lower for extreme poses (Fig. 14f), partly due to poor supervisory signal from the reconstruction loss of side images. This may be improved by imposing constraints from accurate reconstructions of frontal poses.

5 Conclusion

We have presented a method that can learn a 3D model of a deformable object category from an unconstrained collection of single-view images of the object category. The model is able to obtain high-fidelity monocular 3D reconstructions of individual object instances. This is trained based on a reconstruction loss without any supervision, resembling an autoencoder. We have shown that symmetry and illumination are strong cues for shape and help the model to converge to a meaningful reconstruction. Our model outperforms a current state-of-the-art 3D reconstruction method that uses 2D keypoint supervision. As for future work, the model currently represents 3D shape from a canonical viewpoint using a depth map, which is sufficient for objects such as faces that have a roughly convex shape and a natural canonical viewpoint. For more complex objects, it may be possible to extend the model to use either multiple canonical views or a different 3D representation, such as a mesh or a voxel map.

Acknowledgments

The authors would like to thank Soumyadip Sengupta for sharing with us the code to generate synthetic face datasets and Mihir Sahasrabudhe for sending us the reconstruction results of Lifting AutoEncoders. The authors would also like to thank the members of Visual Geometry Group for insightful discussions. This work was supported in part by the Facebook Research, in part by the ERC Horizon 2020 Research, and in part by the Innovation Programme under Grant IDIU 638009.

References

[1] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness," in *Proc. Int. Conf. Learn. Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=Bygh9j09KX

[2] D. P. Mukherjee, A. Zisserman, and J. M. Brady, "Shape from symmetry: Detecting and exploiting symmetry in affine images," *Philos. Trans. Roy. Soc. London*, vol. 351, pp. 77–106, 1995.

[3] A. R. J. François, G. G. Medioni, and R. Waupotitsch, "Mirror symmetry ⇒ 2-view stereo geometry," *Image Vis. Comput.*, vol. 21, pp. 137–143, 2003.

[4] S. Thrun and B. Wegbreit, "Shape from symmetry," in *Proc. 10th IEEE Int. Conf. Comput. Vis.*, 2005, pp. 1824–1831.

[5] S. N. Sinha, K. Ramnath, and R. Szeliski, "Detecting and reconstructing 3D mirror symmetric objects," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 586–600.

[6] Y. Gao and A. L. Yuille, "Exploiting symmetry and/or manhattan properties for 3D object structure estimation from single and multiple images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7408–7417.

[7] M. Sahasrabudhe, Z. Shu, E. Bartrum, R. A. Guler, D. Samaras, and I. Kokkinos, "Lifting autoencoders: Unsupervised learning of a fully-disentangled 3D morphable model using deep non-rigid structure from motion," in *Proc. Int. Conf. Comput. Vis. Workshops*, 2019, pp. 4054–4064.

[8] A. Szabó, G. Meishvili, and P. Favaro, "Unsupervised generative 3D shape learning from natural images," 2019, *arXiv:1910.00287*.

[9] J. R. A. Moniz, C. Beckham, S. Rajotte, S. Honari, and C. Pal, "Unsupervised depth estimation, 3D face rotation and replacement," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 9759–9769.

[10] S. Wu, C. Rupprecht, and A. Vedaldi, "Unsupervised learning of probably symmetric deformable 3D objects from images in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1–10.

[11] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3D face model for pose and illumination invariant face recognition," in *Proc. IEEE Int. Conf. Adv. Video Signal Based Surveill.*, 2009, pp. 296–301.

[12] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 82–90.

[13] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3D point clouds," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 40–49.

[14] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black, "Generating 3D faces using convolutional mesh autoencoders," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 725–741.

[15] Z. Geng, C. Cao, and S. Tulyakov, "3D guided fine-grained face manipulation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9813–9822.

[16] B. Gecer, S. Ploumpis, I. Kotsia, and S. Zafeiriou, "GANFIT: Generative adversarial network fitting for high fidelity 3D face reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1155–1164.

[17] T. Gerig *et al.*, "Morphable face models - An open framework," in *Proc. Int. Conf. Autom. Face Gesture Recognit.*, 2018, pp. 75–82.

[18] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7122–7131.

[19] S. Sengupta, A. Kanazawa, C. D. Castillo, and D. W. Jacobs, "SfSNet: Learning shape, refectance and illuminance of faces in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6296–6305.

[20] M. Wang, Z. Shu, S. Cheng, Y. Panagakis, D. Samaras, and S. Zafeiriou, "An adversarial neuro-tensorial approach for learning disentangled representations,"*Int. J. Comput. Vis* , vol. 127, no. 6–7, pp. 743–762, 2019.

[21] S. Sanyal, T. Bolkart, H. Feng, and M. Black, "Learning to regress 3D face shape and expression from an image without 3D supervision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7763–7772.

[22] M. Gadelha, S. Maji, and R. Wang, "3D shape induction from 2D views of multiple objects," in *Proc. Int. Conf. 3D Vis.*, 2017, pp. 402–411.

[23] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee, "Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 1704–1712.

[24] H. Kato and T. Harada, "Learning view priors for single-view 3D reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9778–9787.

[25] A. Kanazawa, S. Tulsiani, A. A. Efros, and J. Malik, "Learning category-specific mesh reconstruction from image collections," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 371–386.

[26] W. Chen *et al.*, "Learning to predict 3D objects with an interpolation-based differentiable renderer," in *Proc. Conf. Neural Inf. Process. Syst.*, 2019, pp. 9605–9616. [Online]. Available: https://dblp.org/rec/conf/nips/ChenLGSLJF19.html?view=bibtex

[27] P. Henzler, N. Mitra, and T. Ritschel, "Escaping plato's cave using adversarial training: 3D shape from unstructured 2D image collections," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 9984–9993.

[28] N. Kulkarni, A. Gupta, D. F. Fouhey, and S. Tulsiani, "Articulation-aware canonical surface mapping," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 452–461.

[29] S. Goel, A. Kanazawa, and J. Malik, "Shape and viewpoints without keypoints," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 88–104.

[30] X. Li *et al.*, "Self-supervised single-view 3D reconstruction via semantic consistency," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 677–693.

[31] T. Nguyen-Phuoc , C. Li, L. Theis, C. Richardt, and Y. -L. Yang, "Hologan: Unsupervised learning of 3D representations from natural images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7588–7597.

[32] K. Schwarz, Y. Liao, M. Niemeyer, and A. Geiger, "GRAF: Generative radiance fields for 3D-aware image synthesis," in *Proc. Conf. Neural Inf. Process. Syst.*, 2020, pp. 20154–20166. [Online]. Available: https://papers.nips.cc/paper/2020/hash/e92e1b476bb5262d793fd40931e0ed53-Abstract.html

[33] P. Henderson and V. Ferrari, "Learning single-image 3D reconstruction by generative modelling of shape, pose and shading," *Int. J. Comput. Vis.*, vol. 128, pp. 835–854, 2019.

[34] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond Pascal: A benchmark for 3D object detection in the wild," in *Proc.IEEE Winter Conf. Appl. Comput. Vis.*, 2014, pp. 75–82.

[35] O. Faugeras, Q.-T. Luong, and T. Papadopoulo, *The Geometry of Multiple Images*. Cambridge, MA, USA: MIT Press, 2001.

[36] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6612–6619.

[37] B. Ummenhofer *et al.*, "Demon: Depth and motion network for learning monocular stereo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5622–5631.

[38] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 270–279.

[39] C. Bregler, A. Hertzmann, and H. Biermann, "Recovering non-rigid 3D shape from image streams," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2000, pp. 690–696 .

[40] D. Novotny, N. Ravi, B. Graham, N. Neverova, and A. Vedaldi, "C3DPO: Canonical 3D pose networks for non-rigid structure from motion," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7687–7696.

[41] B. K. P. Horn and M. J. Brooks, *Shape from Shading*. Cambridge, MA, USA: MIT Press, 1989.

[42] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah, "Shape-from-shading: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 8, pp. 690–706, Aug. 1999.

[43] J. J. Koenderink, "What does the occluding contour tell us about solid shape?," *Perception*, vol. 13, no. 3, pp. 321–330, 1984.

[44] A. P. Witkin, "Recovering surface shape and orientation from texture," *Artif. Intell.*, vol. 17, no. 1–3, pp. 17–45, 1981.

[45] P. Agrawal, J. Carreira, and J. Malik, "Learning to see by moving," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 37–45.

[46] D. Novotny, D. Larlus, and A. Vedaldi, "Learning 3D object categories by looking around them," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5228–5237.

[47] C. Wang, J. M. Buenaposada, R. Zhu, and S. Lucey, "Learning depth from monocular videos using direct methods," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2022–2030.

[48] X. Li *et al.*, "Online adaptation for consistent mesh reconstruction in the wild," in *Proc. Neural Inf. Process. Syst.*, 2020, pp. 15009–15019. [Online]. Available: https://papers.nips.cc/paper/2020/hash/aba3b6fd5d186d28e06ff97135cade7f-Abstract.html

[49] Y. Luo *et al.*, "Single view stereo matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 155–163.

[50] V. Sitzmann, M. Zollhöfer, and G. Wetzstein, "Scene representation networks: Continuous 3D-structure-aware neural scene representations," in *Proc. Neural Inf. Process. Syst.*, 2019, pp. 1119–1130.

[51] O. Wiles, G. Gkioxari, R. Szeliski, and J. Johnson, "SynSin: End-to-end view synthesis from a single image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7467–7477.

[52] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 405–421. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-58452-8_24

[53] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, "Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3504–3515.

[54] L. Yariv *et al.*, "Multiview neural surface reconstruction by disentangling geometry and appearance," in *Proc. Neural Inf. Process. Syst.*, 2020, pp. 2492–2502.

[55] S. Suwajanakorn, N. Snavely, J. Tompson, and M. Norouzi, "Discovery of latent 3D keypoints via end-to-end geometric reasoning," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 2063–2074. [Online]. Available: https://dblp.org/rec/conf/nips/SuwajanakornSTN18.html?view=bibtex

[56] C.-H. Chen *et al.*, "Unsupervised 3D pose estimation with geometric self-supervision," in *Proc.IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5714–5724.

[57] A. T. Tran, T. Hassner, I. Masi, and G. Medioni, "Regressing robust and discriminative 3D morphable models with a very deep neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5163–5172.

[58] A. T. Tran, T. Hassner, I. Masi, E. Paz, Y. Nirkin, and G. Medioni, "Extreme 3D face reconstruction: Seeing through occlusions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3935–3944.

[59] Y. Feng, F. Wu, X. Shao, Y. Wang, and X. Zhou, "Joint 3D face reconstruction and dense alignment with position map regression network," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 557–574.

[60] J. Guo, X. Zhu, Y. Yang, F. Yang, Z. Lei, and S. Z. Li, "Towards fast, accurate and stable 3 D dense face alignment," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 152–168.

[61] Y. Feng, H. Feng, M. J. Black, and T. Bolkart, "Learning an animatable detailed 3D face model from in-the-wild images," 2020, *arXiv:2012.04012*.

[62] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll , and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graph.*, vol. 34, no. 6, 2015, Art. no. 248.

[63] J. Thewlis, H. Bilen, and A. Vedaldi, "Unsupervised learning of object frames by dense equivariant image labelling," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 844–855. [Online]. Available: https://dblp.org/rec/conf/nips/ThewlisBV17.html?view=bibtex
[64] J. Thewlis, H. Bilen, and A. Vedaldi, "Modelling and unsupervised learning of symmetric deformable object categories," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 8189–8200.
[65] Z. Shu, M. Sahasrabudhe, A. Guler, D. Samaras, N. Paragios, and I. Kokkinos, "Deforming autoencoders: Unsupervised disentangling of shape and appearance," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 650–665.
[66] J.-Y. Zhu *et al.*, "Visual object networks: Image generation with disentangled 3D representations," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 118–129. [Online]. Available: https://dblp.org/rec/conf/nips/ZhuZZ00TF18.html?view=bibtex
[67] M. M. Loper and M. J. Black, "OpenDR: An approximate differentiable renderer," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 154–169.
[68] H. Kato, Y. Ushiku, and T. Harada, "Neural 3D mesh renderer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3907–3916.
[69] S. Liu, T. Li, W. Chen, and H. Li, "Soft rasterizer: A differentiable renderer for image-based 3D reasoning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7707–7716.
[70] B. Horn, "Obtaining shape from shading information, " in, *The Psychology of Computer Vision*. New York, NY, USA: McGraw-Hill, 1975.
[71] P. N. Belhumeur, D. J. Kriegman, and A. L. Yuille, "The bas-relief ambiguity," *Int. J. Comput. Vis.*, vol. 35, pp. 33–44, 1999.
[72] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 5574–5584. [Online]. Available: https://dblp.org/rec/conf/nips/KendallG17.html?view=bibtex
[73] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015. [Online]. Available: https://dblp.org/rec/journals/corr/SimonyanZ14a.html?view=bibtex
[74] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.
[75] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3730–3738.
[76] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-PIE," *Image Vis. Comput.*, vol. 28, no. 5, pp. 807–813, 2010.
[77] L. A. Jeni, J. F. Cohn, and T. Kanade, "Dense 3D face alignment from 2D videos in real-time," in *Proc. Int. Conf. Autom. Face Gesture Recognit.*, 2015, pp. 1–8.
[78] X. Zhang *et al.*, "BP4D-Spontaneous: A high-resolution spontaneous 3D dynamic facial expression database," *Image Vis. Comput.*, vol. 32, no. 10, pp. 692–706, 2014.
[79] L. Yin, X. Chen, Y. Sun, T. Worm, and M. Reale, "A high-resolution 3D dynamic facial expression database," in *Proc. Int. Conf. Autom. Face Gesture Recognit.*, 2008, pp. 1–6.
[80] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016.
[81] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3485–3492.
[82] W. Zhang, J. Sun, and X. Tang, "Cat head detection - How to effectively exploit shape and texture features," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 802–816.
[83] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar, "Cats and dogs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3498–3505.
[84] A. X. Chang *et al.*, "Shapenet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*.
[85] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2366–2374.
[86] E. J. Crowley, O. M. Parkhi, and A. Zisserman, "Face painting: Querying art with photos," in *Proc. Brit. Mach. Vis. Conf.*, 2015, pp. 1–13.
[87] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Proc. INTERSPEECH*, 2018, pp. 1086–1090.
[88] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *Proc. Int. Conf. Learn. Representations*, 2018. [Online]. Available: https://dblp.org/rec/conf/iclr/GidarisSK18.html?view=bibtex
[89] H.-Y. F. Tung, A. W. Harley, W. Seto, and K. Fragkiadaki, "Adversarial inverse graphics networks: Learning 2D-to-3D lifting and image-to-image translation from unpaired supervision," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4364–4372.
[90] A. Tewari *et al.*, "MoFA: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3735–3744.
[91] Z.-H. Feng *et al.*, "Evaluation of dense 3D reconstruction from 2D face images in the wild," in *Proc. Int. Conf. Autom. Face Gesture Recognit.*, 2018, pp. 780–786.
[92] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: Database and results," *Image Vis. Comput.*, vol. 47, pp. 3–18, 2016.



Shangzhe Wu received the bachelor's degree in computer science from the Hong Kong University of Science and Technology, where he worked with Chi-Keung Tang and Yu-Wing Tai on image translation. He is currently working toward the DPhil degree with the Visual Geometry Group, University of Oxford, supervised by Andrea Vedaldi. His research focuses on unsupervised 3D understanding. He was the recipient of the Best Paper Award at CVPR 2020.



Christian Rupprecht received the PhD degree from the Technical University of Munich, Germany, advised by Nassir Navab and Gregory D. Hager (JHU). He is currently a postdoctoral researcher with the Visual Geometry Group, University of Oxford. For six months, he was with Chris Pal, Mila Institute, Montreal, working on AI safety. His research interests include self-supervised and minimally supervised learning for computer vision.



Andrea Vedaldi is currently a professor of computer vision and machine learning with the University of Oxford, where he has been co-leading Visual Geometry Group since 2012. He is also a research scientist with Facebook AI Research, London. He has authored or coauthored more than 130 peer-reviewed publications in the top machine vision and artificial intelligence conferences and journals. His research interests include unsupervised learning of representations and geometry in computer vision. He was the recipient of the Mark Everingham Prize for selfless contributions to the computer vision community, the Open Source Software Award by the ACM, and the Best Paper Award from the Conference on Computer Vision and Pattern Recognition.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.

Supplementary Material: Unsupervised Learning of Probably Symmetric Deformable 3D Objects from Images in the Wild

Shangzhe Wu Christian Rupprecht Andrea Vedaldi

Visual Geometry Group, University of Oxford

{szwu, chrisr, vedaldi}@robots.ox.ac.uk

1. Technical Details

We report the training details including all hyper-parameter settings in Table 1, and detailed network architectures in Tables 2 to 4. We use standard encoder networks for both viewpoint and lighting predictions, and encoder-decoder networks for depth, albedo and confidence predictions. In order to mitigate checkerboard artifacts [3] in the predicted depth and albedo, we add a convolution layer after each deconvolution layer and replace the last deconvolotion layer with nearest-neighbor upsampling, followed by 3 convolution layers. Abbreviations of the operators are defined as follows:

- Conv$(c_{in}, c_{out}, k, s, p)$: convolution with c_{in} input channels, c_{out} output channels, kernel size k, stride s and padding p.
- Deconv$(c_{in}, c_{out}, k, s, p)$: deconvolution [6] with c_{in} input channels, c_{out} output channels, kernel size k, stride s and padding p.
- Upsample(s): nearest-neighbor upsampling with a scale factor of s.
- GN(n): group normalization [5] with n groups.
- LReLU(α): leaky ReLU [2] with a negative slope of α.

2. Additional Qualitative Results

We provide more qualitative results in the following and 3D animations in the supplementary video[1]. Fig. 1 shows reconstruction results on human faces from CelebA and 3DFAW. We also show reconstruction results on face paintings and drawings collected from [1] and the Internet in Figs. 2 and 3. Figs. 4 to 6 show results on real cat faces from [7, 4], abstract cats collected from the Internet and synthetic cars rendered using ShapeNet.

[1] https://www.youtube.com/watch?v=5rPJyrU-WE4

Parameter	Value/Range
Optimizer	Adam
Learning rate	1×10^{-4}
Number of epochs	30
Batch size	64
Loss weight λ_{f}	0.5
Loss weight λ_{p}	1
Input image size	64×64
Output image size	64×64
Depth map	$(0.9, 1.1)$
Albedo	$(0, 1)$
Light coefficient k_s	$(0, 1)$
Light coefficient k_d	$(0, 1)$
Light direction l_x, l_y	$(-1, 1)$
Viewpoint rotation $w_{1:3}$	$(-60°, 60°)$
Viewpoint translation $w_{4:6}$	$(-0.1, 0.1)$
Field of view (FOV)	10

Table 1: Training details and hyper-parameter settings.

Encoder	Output size
Conv(3, 32, 4, 2, 1) + ReLU	32
Conv(32, 64, 4, 2, 1) + ReLU	16
Conv(64, 128, 4, 2, 1) + ReLU	8
Conv(128, 256, 4, 2, 1) + ReLU	4
Conv(256, 256, 4, 1, 0) + ReLU	1
Conv(256, c_{out}, 1, 1, 0) + Tanh $\rightarrow$ *output*	1

Table 2: Network architecture for viewpoint and lighting. The output channel size c_{out} is 6 for viewpoint, corresponding to rotation angles $w_{1:3}$ and translations $w_{4:6}$ in x, y and z axes, and 4 for lighting, corresponding to k_s, k_d, l_x and l_y.

References

[1] Elliot J. Crowley, Omkar M. Parkhi, and Andrew Zisserman. Face painting: querying art with photos. In *Proc. BMVC*, 2015. 1

[2] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML*, 2013. 1

Encoder	Output size
Conv(3, 64, 4, 2, 1) + GN(16) + LReLU(0.2)	32
Conv(64, 128, 4, 2, 1) + GN(32) + LReLU(0.2)	16
Conv(128, 256, 4, 2, 1) + GN(64) + LReLU(0.2)	8
Conv(256, 512, 4, 2, 1) + LReLU(0.2)	4
Conv(512, 256, 4, 1, 0) + ReLU	1
Decoder	**Output size**
Deconv(256, 512, 4, 1, 0) + ReLU	4
Conv(512, 512, 3, 1, 1) + ReLU	4
Deconv(512, 256, 4, 2, 1) + GN(64) + ReLU	8
Conv(256, 256, 3, 1, 1) + GN(64) + ReLU	8
Deconv(256, 128, 4, 2, 1) + GN(32) + ReLU	16
Conv(128, 128, 3, 1, 1) + GN(32) + ReLU	16
Deconv(128, 64, 4, 2, 1) + GN(16) + ReLU	32
Conv(64, 64, 3, 1, 1) + GN(16) + ReLU	32
Upsample(2)	64
Conv(64, 64, 3, 1, 1) + GN(16) + ReLU	64
Conv(64, 64, 5, 1, 2) + GN(16) + ReLU	64
Conv(64, c_{out}, 5, 1, 2) + Tanh $\rightarrow$ $output$	64

Table 3: Network architecture for depth and albedo. The output channel size c_{out} is 1 for depth and 3 for albedo.

Encoder	Output size
Conv(3, 64, 4, 2, 1) + GN(16) + LReLU(0.2)	32
Conv(64, 128, 4, 2, 1) + GN(32) + LReLU(0.2)	16
Conv(128, 256, 4, 2, 1) + GN(64) + LReLU(0.2)	8
Conv(256, 512, 4, 2, 1) + LReLU(0.2)	4
Conv(512, 128, 4, 1, 0) + ReLU	1
Decoder	**Output size**
Deconv(128, 512, 4, 1, 0) + ReLU	4
Deconv(512, 256, 4, 2, 1) + GN(64) + ReLU	8
Deconv(256, 128, 4, 2, 1) + GN(32) + ReLU	16
↳ Conv(128, 2, 3, 1, 1) + SoftPlus $\rightarrow$ $output$	16
Deconv(128, 64, 4, 2, 1) + GN(16) + ReLU	32
Deconv(64, 64, 4, 2, 1) + GN(16) + ReLU	64
Conv(64, 2, 5, 1, 2) + SoftPlus $\rightarrow$ $output$	64

Table 4: Network architecture for confidence maps. The network outputs two pairs of confidence maps at different spatial resolutions for photometric and perceptual losses.

[3] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016. 1

[4] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *Proc. CVPR*, 2012. 1

[5] Yuxin Wu and Kaiming He. Group normalization. In *Proc. ECCV*, 2018. 1

[6] Matthew D. Zeiler, Graham W. Taylor, and Rob Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *Proc. ICCV*, 2011. 1

[7] Weiwei Zhang, Jian Sun, and Xiaoou Tang. Cat head detection - how to effectively exploit shape and texture features. In *Proc. ECCV*, 2008. 1



Figure 1: **Reconstruction of human faces.**



Figure 2: **Reconstruction of face paintings.**



Figure 3: **Reconstruction of abstract faces.**

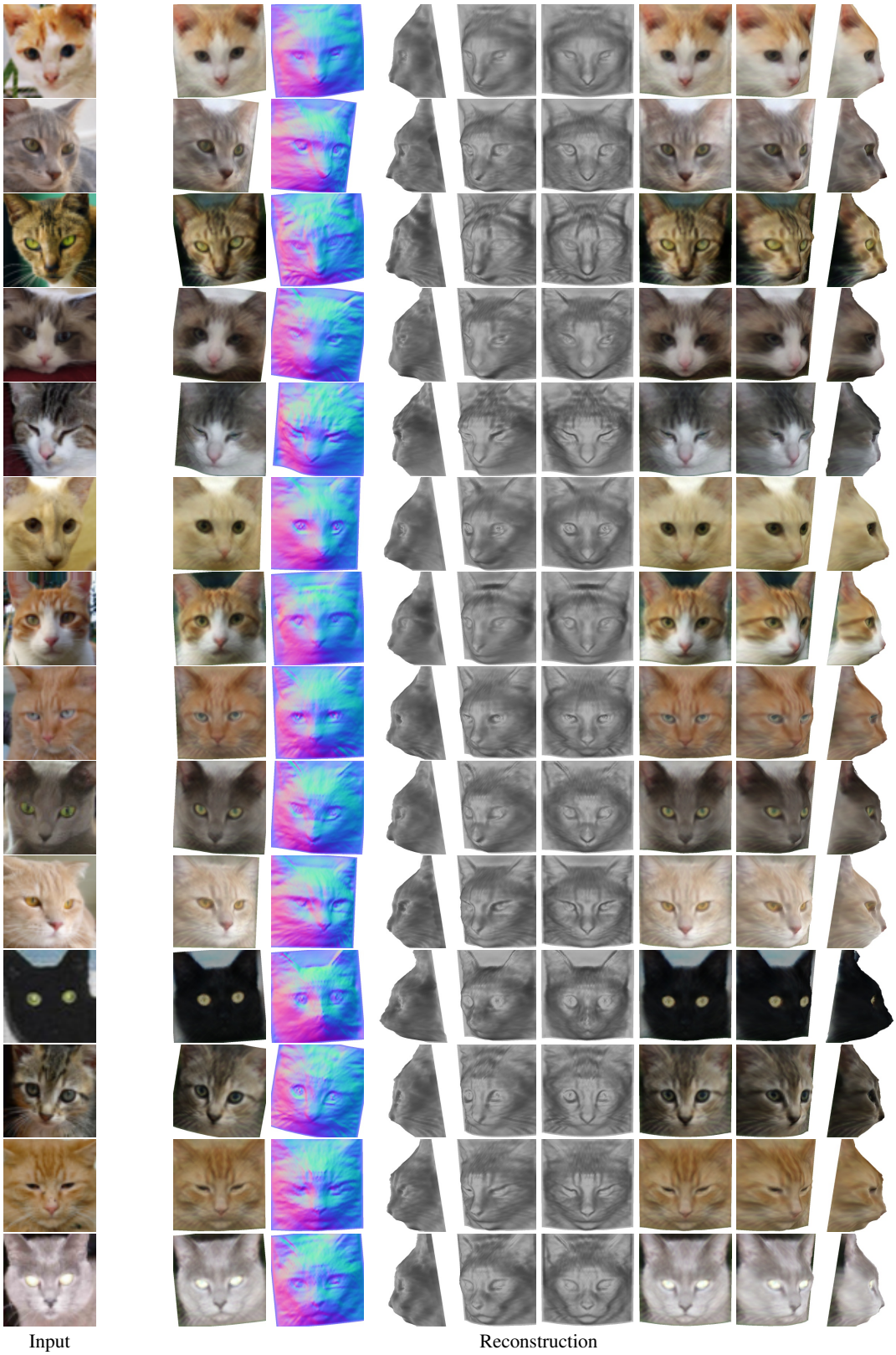


Figure 4: **Reconstruction of cat faces.**

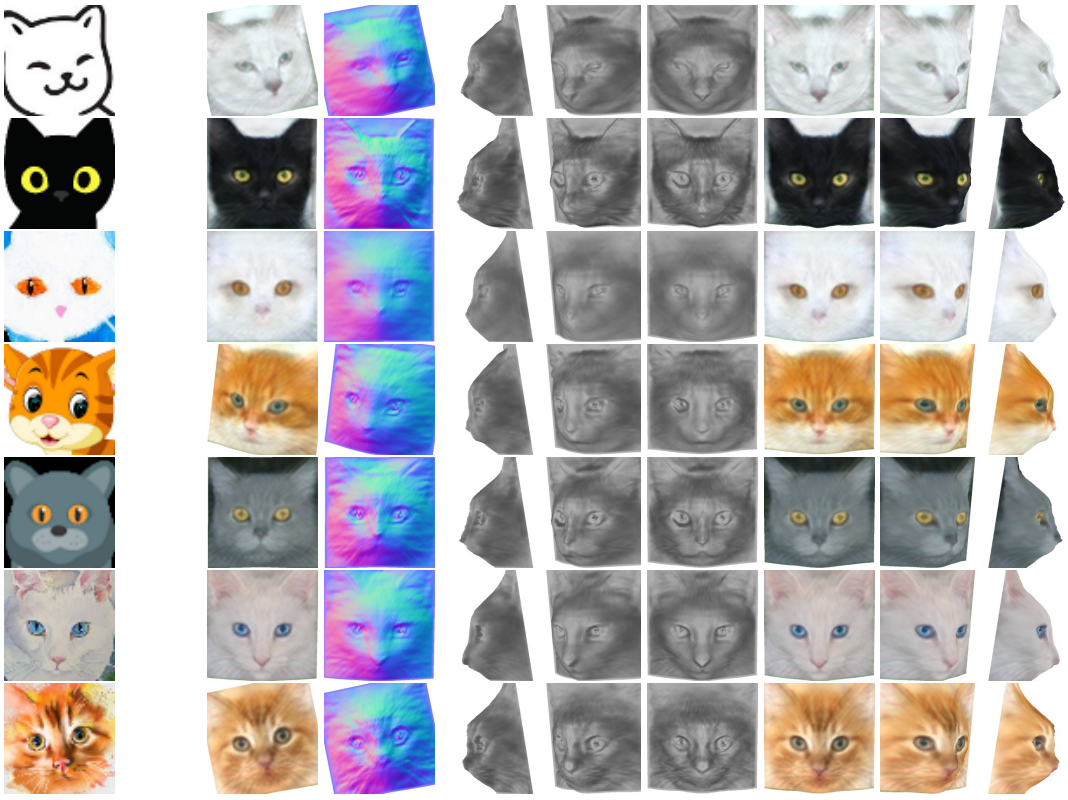


Figure 5: **Reconstruction of abstract cats.**

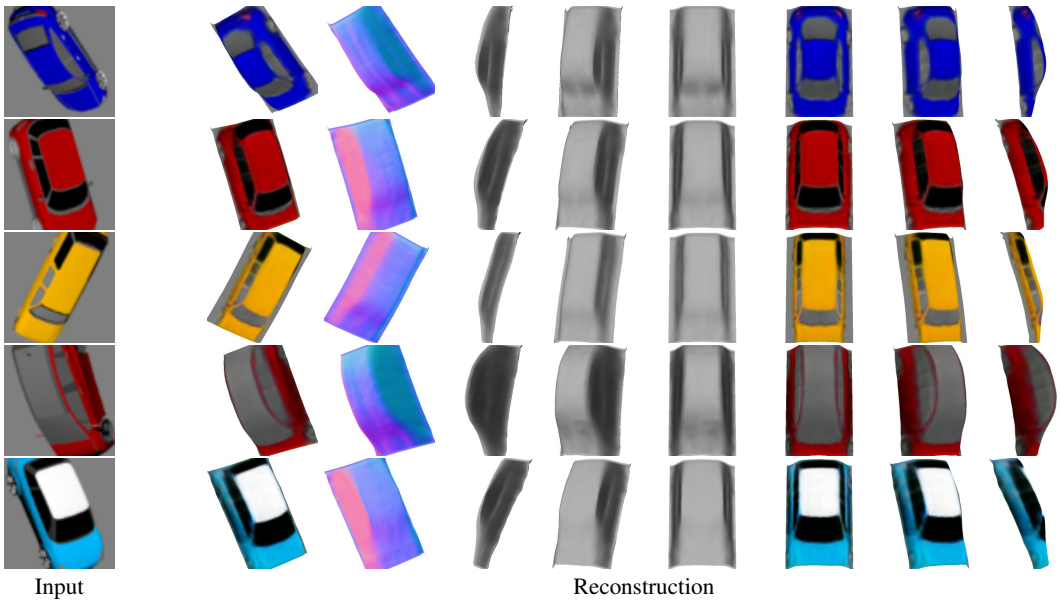


Figure 6: **Reconstruction of synthetic cars.**

4

De-rendering the World's Revolutionary Artefacts

This work was presented at the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

arXiv:2104.03954v2 [cs.CV] 31 Aug 2021

De-rendering the World's Revolutionary Artefacts

Shangzhe Wu[1,4*] Ameesh Makadia[4] Jiajun Wu[2]
Noah Snavely[4] Richard Tucker[4] Angjoo Kanazawa[3,4]

[1]University of Oxford [2]Stanford University
[3]University of California, Berkeley [4]Google Research

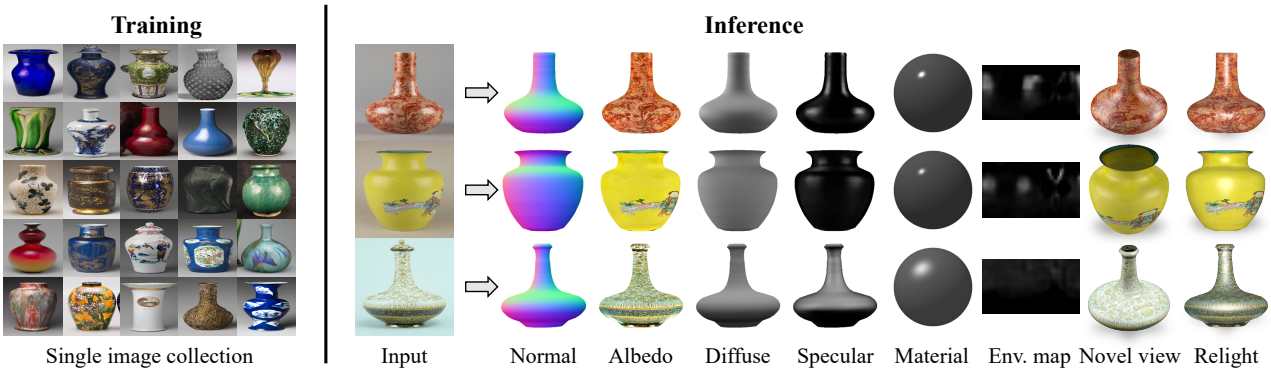


Figure 1: **De-rendering from single images.** From only a real single-view image collection of "revolutionary" (i.e., solid of revolution) artefacts with known silhouettes as training data (left), our framework learns to de-render a single image into shape, albedo and complex lighting and material components, suitable for applications such as novel-view synthesis and relighting (right).

Abstract

Recent works have shown exciting results in unsupervised image de-rendering—learning to decompose 3D shape, appearance, and lighting from single-image collections without explicit supervision. However, many of these assume simplistic material and lighting models. We propose a method, termed RADAR, that can recover environment illumination and surface materials from real single-image collections, relying neither on explicit 3D supervision, nor on multi-view or multi-light images. Specifically, we focus on rotationally symmetric artefacts that exhibit challenging surface properties including specular reflections, such as vases. We introduce a novel self-supervised albedo discriminator, which allows the model to recover plausible albedo without requiring any ground-truth during training. In conjunction with a shape reconstruction module exploiting rotational symmetry, we present an end-to-end learning framework that is able to de-render the world's revolutionary artefacts. We conduct experiments on a real vase dataset and demonstrate compelling decomposition results, allowing for applications including free-viewpoint rendering and relighting. More results and code at: https://sorderender.github.io/.

1. Introduction

Consider one of the vases shown in Fig. 1. From just a single image, we can tell a lot about the underlying properties of that vase. Despite the image's flatness, we can perceive an instance of a 3D surface with various lights cast upon it. We can distinguish between areas where the underlying color of the vase changes and regions that reflect light, revealing the glossiness of the surface and its local geometry.

We introduce a model that aims to *de-render* a single image into these factors—geometry, material, and illumination—which we call ***RADAR*** (***R***evolutionary ***A***rtefact ***D***e-rendering ***A***nd ***R***e-rendering). In particular, our approach can decompose real images of vase-like objects under complex illumination and with glossy materials. Notably, our approach can learn this ability just from collections of single images (i.e., where each object is pictured once), without explicit 3D supervision or multiple images. This allows us to analyze images obtained in real world settings, such as artefact collections in museums, and subsequently apply modifications including relighting, as illustrated in Fig. 1.

Making de-rendering tractable involves simplifying assumptions. In some methods, this means requiring explicit supervision, e.g., with synthetic [25, 27] or specially captured data [26]. An alternative to direct supervision is to

*The work was primarily done during an internship at Google Research.

observe an object under multiple viewpoints [6, 47] or multiple lights [46, 7], but for many existing image collections, such multiple views are unavailable. Hence, learning to de-render from single image collections has been of growing interest [44, 37]. However, these approaches assume simplistic shading or lighting models, such as Lambertian, and are not applicable to realistic scenarios with complex illumination effects.

In contrast, our objective in this paper is to explore unsupervised de-rendering in the presence of more complex illumination effects. To make our task tractable, we consider simplifying assumptions on the 3D shape. We draw inspiration from recent work [44] that leverages symmetry priors for self-supervised decomposition. Specifically, we focus on de-rendering objects whose shapes are described by solids of revolution (SoRs, or "*revolutionary*" objects)—such objects include many categories of man-made objects such as vases. This allows us to derive a simple yet effective method for recovering the 3D geometry and camera viewpoint from only single images with 2D silhouettes.

Our model de-renders a single image of a revolutionary object into 3D geometry, viewpoint, albedo, material shininess, and environment lighting. Even with this strong assumption on SoR shape and inductive bias on the rendering process, this is still an extremely under-constrained problem. As with most ill-posed inverse problems, we must prevent degenerate solutions where the model learns no disentanglement at all. Another major challenge is to predict realistic diffuse albedo in regions saturated by specular reflections.

To ensure realistic disentanglement, we incorporate novel components into our model. In particular, we propose a new adversarial module that we call a *Self-supervised Albedo Discriminator* (SAD). The key insight is that the distribution of diffuse albedo patches should be independent of observed specular effects—it should not be possible to tell from the albedo alone whether a particular surface region exhibits a specular reflection or not. Unlike existing adversarial frameworks, a key feature of SAD is that the discriminator always takes its inputs from the predicted albedo and never requires a 'real' albedo, hence the label *self-supervised*.

In summary, we propose ***RADAR***, an end-to-end framework for de-rendering single images into shape, complex lighting, and materials, learning only from single-image collections with 2D silhouettes. We evaluate our approach numerically on a synthetic dataset, and demonstrate effective results on real images of revolutionary artefacts from museum collections, where our approach allows for applications such as free-viewpoint rendering and relighting.

2. Related Work

There is a vast literature on intrinsic image decomposition and de-rendering. Many methods build upon some physical model of the image formation process and complement such models with representations learned from data. Existing methods can be roughly divided into three categories: optimization-based, learning from annotated or synthetic images, and learning from unannotated image collections. We focus on single-image decomposition methods.

Optimization-based approaches. Traditional approaches derive heuristic physical priors and rely on optimization with such priors to decompose images [16, 5, 24, 4, 13]. In particular, SIRFS [4] is an extension of classic shape-from-shading that recovers shape as well as reflectance and illumination, but does not handle non-Lambertian reflectance. While these methods work well in specific domains, it turns out to be challenging to design general priors for real images with complex intrinsic albedo and BRDFs.

Learning from annotated images. Leveraging advances in deep learning, researchers have explored learning-based intrinsic image decomposition. Shi *et al*. [40] use synthetic ShapeNet objects for training; Liu *et al*. [29] extend this framework for material editing. Others attempt to decompose general objects under flash illumination [27, 38] or general indoor scenes [25], similarly with synthetic data. However, models trained purely only synthetic data often generalize poorly to real scenes due to the domain gap. Some methods pre-train models on synthetic data and then fine-tune them on real data [17, 39] for better generalization. Tremendous efforts are still required to generate large-scale realistic synthetic data that allows easy fine-tuning.

A few works have also studied learning from controlled data, such as multi-view or multi-light images. Many of them also require multiple images during inference [46, 6, 47, 7]. Kulkarni *et al*. [21] and Ma *et al*. [30] leverage training pipelines that allow for single-image inference. However, the complexity in acquiring controlled multiple images of the same real-world object has led these models to be trained again only on synthetic data. Some recent works leverage photo collections of real scenes [23, 49, 48, 28], but are often restricted to famous landmarks or street view imagery.

Learning from unannotated image collections. As explicit or indirect supervision is rarely available for real-world objects and synthetic datasets often lack sufficient realism, a few recent papers have attempted to learn image decomposition directly from unannotated real image collections [18, 9, 44, 37], but none of them can recover complex material and lighting effects, such as specular reflection.

Our method follows a similar setup, and is able to recover environment illumination and glossy material properties from a single image. Inspired by Wu *et al*. [44], which leverages a bilateral symmetry assumption to recover shape, albedo and diffuse lighting, our model also embraces a rotational symmetry prior to obtain the shape of sufficient quality, allowing us to start to reason about complex material and illumination in real images. Rotational symmetry has been exploited for shape recovery in prior work [11, 8, 33, 10],

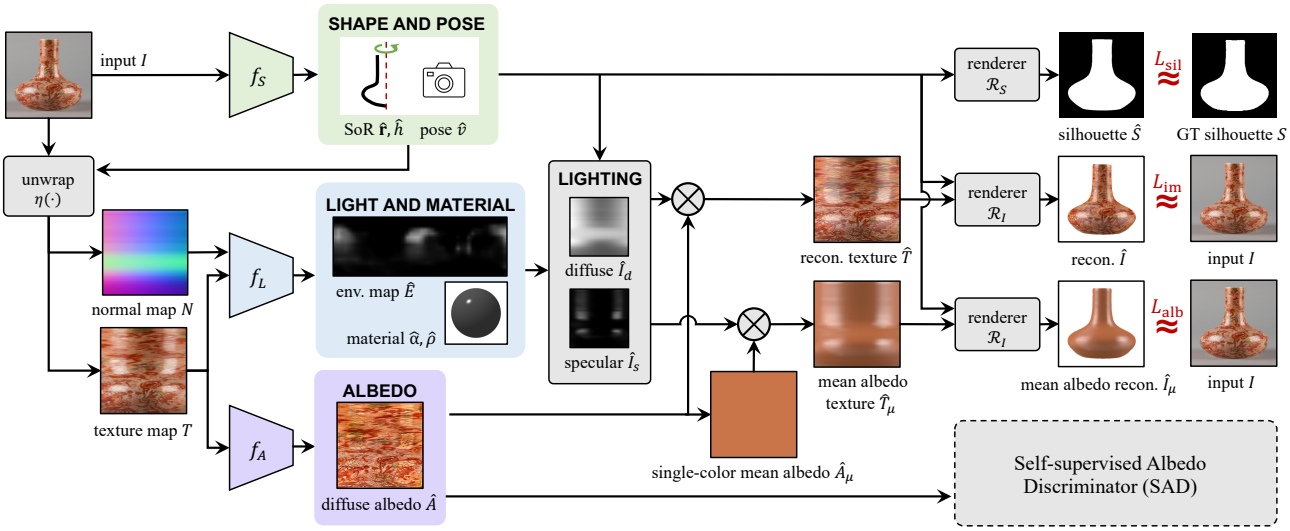


Figure 2: ***RADAR* training pipeline.** Given a single image of a vase, our model first predicts shape and pose with the shape network f_S, which is used to unwrap the surface of the object. The lighting network f_L and albedo network f_A then take in the unwrapped textures and predict environment lighting, surface material and diffuse albedo, which are recomposed to render the input image. A self-supervised albedo discriminator is proposed to encourage the decomposition of albedo and lighting, illustrated in Fig. 4. The whole pipeline is trained end-to-end without any external supervision except for the silhouettes.

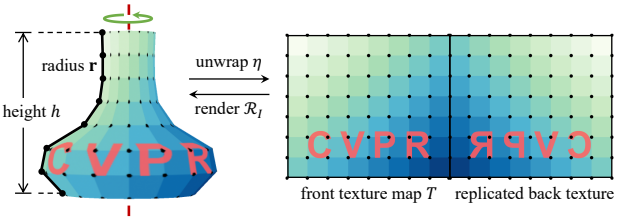


Figure 3: **Surface of revolution.** We represent the surface using a vertex grid $V \in \mathbb{R}^{L \times K \times 3}$ generated by rotating a discretized radius curve $\mathbf{r} \in \mathbb{R}^L$ around the axis of revolution.

but our method also recovers material and lighting.

3. Method

Given a collection of single-view images of revolutionary artefacts, such as vases, our goal is to learn a de-rendering function Φ, which takes in a single image I and predicts the 3D shape of the object, its surface material properties, and the environment illumination. Making this even more challenging, we do not want to rely on explicit 3D supervision or multi-view images, as obtaining such supervision is not only expensive but often simply intractable for precaptured image collections.

In general, recovering shape, material, and lighting without direct supervision is an extremely ill-posed inverse problem. In this paper, we consider this task for objects whose shapes roughly observe *solids of revolution (SoRs)*, and assume that only minimal indirect training supervision is available, in the form of reasonable silhouettes which can be obtained using off-the-shelf object detectors. SoRs describe a large subset of real world objects. In particular, we focus on vases, which are made of materials exhibiting complex lighting effects such as specular reflections.

Fig. 2 shows an overview of our training pipeline. In the following sections, we present the main components of model, including three sub-networks that recover the shape (f_S), lighting (f_L), and diffuse albedo (f_A) from a single image, along with the reconstruction losses used to train them. We then describe additional components we introduce to encourage realistic disentanglement of lighting and albedo.

3.1. SoR Shape and Texture

Vases made on spinning wheels have 'revolutionary' shapes known as solids of revolution (SoRs). SoRs are generated by a plane curve (the generatrix) rotated about a straight line (the axis of revolution). We model the shape of a vase as an SoR, parameterized by a vector $\mathbf{r} \in \mathbb{R}^L$ giving the radius (i.e. the perpendicular distance from the axis to the generatrix) at L evenly-spaced points along the axis of revolution, together with the axis height h, as illustrated in Fig. 3.

A complete discretization of the SoR shape is obtained by rotating the resulting sampled curve $\mathbf{r}$ about the axis to obtain sample points at K evenly-spaced rotation angles in $[0°, 360°)$. This produces a regular sampling of the surface in height and angle, and we denote the resulting vertex map as $V \in \mathbb{R}^{L \times K \times 3}$. To recover the shape from a single image, we define a shape network f_S, which takes an image I and predicts the radius column $\hat{\mathbf{r}}$ and its height $\hat{h}$, as well as the camera pose $\hat{v} \in \mathbb{R}^4$, which specifies pitch and roll Euler

angles and translation in the X and Y axes:

$$(\hat{\mathbf{r}}, \hat{h}, \hat{v}) = f_S(I). \tag{1}$$

As described above, the vertex map $\hat{V}$ can be constructed from the predicted radius column $\hat{\mathbf{r}}$ and height $\hat{h}$. We use a differentiable renderer $\mathcal{R}_S$ [19] to render a silhouette of the predicted SoR mesh with vertices $\hat{V}$ and camera pose $\hat{v}$:

$$\hat{S} = \mathcal{R}_S(\hat{V}, \hat{v}). \tag{2}$$

For simplicity, we fix the camera intrinsics for all rendering operations in our model. We can train the network by minimizing the silhouette loss:

$$L_{\text{sil}} = \lambda_{\text{s}} \|S - \hat{S}\|_2^2 + \lambda_{\text{dt}} \|\texttt{dt}(S) \odot \hat{S}\|_1, \tag{3}$$

where $\odot$ is the Hadamard product and S the target silhouette, obtained with an off-the-shelf object segmentation technique (details in Sec. 4.1). $\texttt{dt}(\cdot)$ is the distance transform of the mask, and λ_{s}, λ_{dt} are weights balancing the terms.

Texture representation and unwrapping. As described above, our SoR representation allows us to unwrap the surface into a regular 2D grid, which can be easily triangulated for rendering. To render textures, we define a 2D texture map $T \in \mathbb{R}^{H_T \times W_T \times 3}$ in *the unwrapped space* aligned with the vertex map, which is interpolated and mapped onto the surface during rendering with a differentiable renderer $\mathcal{R}_I$:

$$I = \mathcal{R}_I(V, v, T). \tag{4}$$

We denote by η the inverse mapping of this texture rendering operation, which unwraps the textures of a SoR surface into a texture map T from an image I:

$$T = \eta(V, v, I). \tag{5}$$

As explained in the next section, we decompose material and lighting in this unwrapped space, since a 2D convolution on the unwrapped texture map will behave closer to an intrinsic convolution on the SoR surface, and it is also viewpoint and shape invariant. See Fig. 3 for an illustration.

3.2. Unsupervised De-rendering

We first describe our lighting model, then the network architecture that can produces these components. Because this is still ill-posed, we discuss our methods to encourage proper disentaglement without explicit supervision.

Lighting model. We use a Phong illumination model [34] with a normalized specular term and a single-channel environment map $E \in \mathbb{R}^{H_E \times W_E}$ to represent the environment lighting. Each vase is modeled as having a diffuse albedo texture $A \in \mathbb{R}^{H_T \times W_T \times 3}$ in the unwrapped space, a constant shininess scalar α, and a constant specular albedo scalar ρ, since specular reflections are often due to a layer of glaze on the vase's surface. Note that for simplicity, we assume gray illumination leaving the color information to the albedo, and ignore global illumination. The rendered texture T is given by the tone-mapped sum of diffuse and specular terms:

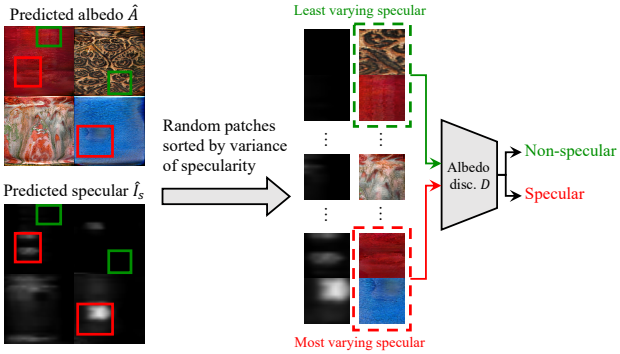


Figure 4: **Self-supervised albedo discriminator.** We randomly sample patches from the predicted albedo map and sort them by the variance of the corresponding specular patches. We feed the two groups of albedo patches with the lowest and the highest specular variance to a discriminator, and train our model to prevent it from telling the two groups apart.

$$T = \tau(A \odot I_d + \rho I_s), \tag{6}$$

where the tone-mapper is the inverse gamma function $\tau(I) = I^{1/\gamma}$ with $\gamma = 2.2$. $I_d, I_s \in \mathbb{R}^{H_T \times W_T}$ are the diffuse and specular lighting factors also in the unwrapped space, which are computed as follows.

We treat each pixel i in E as a directional light with intensity I_i and lighting direction $\vec{L_i}$. Since E is an equirectangular projection of the sphere, $\vec{L_i}$ can be determined directly from the pixel coordinates. I_d and I_s are then given by

$$I_{dj} = \sum_{i \in E} I_i (\vec{L_i} \cdot \vec{N_j}), \tag{7}$$

$$I_{sj} = \frac{\alpha + 1}{2\pi} \sum_{i \in E} I_i (\vec{R}_{i,j} \cdot \vec{P_j})^{\alpha}. \tag{8}$$

Here the subscript j denotes a pixel in the unwrapped space, N is the surface normal map, and P the view direction map. Finally, $\vec{R}_{i,j}$ is the reflected light direction computed from the environment light direction at pixel i and the surface normal at pixel j: $\vec{R}_{i,j} = 2(\vec{L_i} \cdot \vec{N_j})\vec{N_j} - \vec{L_i}$. Inspired by energy-conserving Phong models [3], we include a normalization term in the specular component, which essentially ensures that the cosine lobe integrates to a constant. We find it helpful during training in preventing the specular component from vanishing when the shininess α gets large.

Predicting materials and lighting. Recall that our objective is to de-render an image into various components, specified by V, v, A, E, α, and ρ. Our model takes the predicted shape and camera pose ($\hat{V}$ and $\hat{v}$) as inputs when recovering the remaining terms (see Fig. 2).

In order to decompose surface materials and illumination, we first unwrap the texture of the frontal (visible) half of the vase from the input image I into a texture map $T \in \mathbb{R}^{H_T \times W_T \times 3}$ using Eq. (5) with the predicted shape and pose: $T = \eta(\hat{V}_\text{f}, \hat{v}, I)$, where $\hat{V}_\text{f}$ denotes the vertices corresponding to the frontal half of the vase.

We design two networks, f_A and f_L, to predict albedo and lighting. The albedo network takes in the unwrapped texture map T and predicts a diffuse albedo map: $\hat{A} = f_A(T)$. The lighting network takes in an additional normal map $\hat{N}$ concatenated along channel dimension and predicts the environment map, shininess, and specular albedo: $(\hat{E}, \hat{\alpha}, \hat{\rho}) = f_L(T, \hat{N})$. Note that the surface normals $\hat{N} \in \mathbb{R}^{H_T \times W_T \times 3}$ are computed from the predicted vertices $\hat{V}$ and upsampled.

We then apply Eqs. (7) and (8) to generate predicted lighting factors $\hat{I}_d, \hat{I}_s$, and render a reconstruction of the input image using the differentiable renderer $\mathcal{R}_I$:

$$\hat{T} = \tau(\hat{A} \odot \hat{I}_d + \hat{\rho}\hat{I}_s), \tag{9}$$

$$\hat{I} = \mathcal{R}_I(\hat{V}_\text{f}, \hat{v}, \hat{T}). \tag{10}$$

We can then train the networks with a reconstruction loss:

$$L_\text{im} = \|\tilde{S} \odot (I - \hat{I})\|_1, \tag{11}$$

where $\tilde{S}$ is the intersection of the ground-truth silhouette S and the rendered silhouette of the frontal visible part $\hat{S}_\text{f}$.

3.3. Disentangling Lighting and Albedo

Thus far, we have introduced three networks (f_S, f_A, f_E) to de-render an image into its shape, material, and lighting components. While the loss L_im ensures these components combine to faithfully reproduce the input image, recovering the individual terms correctly remains underconstrained.

From the lighting model (Eq. (6)) we can identify two prominent failure modes when training only with the reconstruction loss. First, the model can always predict little or no specularity and leave all the specular reflections in the albedo map. Second, a non-empty specular map is still insufficient to ensure accurate albedo, as there is no incentive for the model to disentangle these components correctly, or to reconstruct realistic albedo in regions saturated by specularity. In the following, we introduce additional components to our model to prevent these failure modes.

Single-color albedo rendering. To encourage the model to utilize the lighting components, we replace the predicted diffuse albedo map $\hat{A}$ with a single average color of it $\hat{A}_\mu$, and obtain a second reconstructed image $\hat{I}_\mu$ with this single-color albedo. We then define another reconstruction loss L_alb similar to Eq. (11):

$$L_\text{alb} = \|\tilde{S} \odot (I - \hat{I}_\mu)\|_1. \tag{12}$$

This auxiliary loss encourages the lighting network to make a coarse lighting prediction, such that the reconstructed image rendered with single-color albedo can still recover some color variation in the input image resulting from the lighting alone. However, this does not guarantee correct lighting or albedo predictions, as the single-color approximation does not reflect the color diversity in real vases, or provide a useful signal to reconstruct albedo in saturated specular regions. Hence, we address these limitations next.

Self-supervised albedo discriminator. In order to successfully recover the diffuse material, we must incentivize the model to predict a realistic albedo map free of specular effects. This is particularly challenging for large patches saturated by specular reflections, which require an inpainting-like solution to recover the underlying albedo. To this end, we propose a novel specularity-guided Self-supervised Albedo Discriminator (SAD).

Starting with the weak assumption that our model can predict a moderately reasonable specular map, we make two key insights. First, the distribution of patches in the true diffuse albedo is independent of the specular map, i.e., it should not be possible to predict specular reflection from the albedo alone. Second, the accuracy of the predicted albedo for an image patch is generally inversely related to the amount of specularity in the patch. This follows from the observation that where the specularity is low, the input texture map is much closer to the true albedo compared to image patches saturated by specular reflections.

From these observations, it follows that we can improve the albedo prediction in highly specular regions by making their distribution indistinguishable from that of the albedo patches in low specular regions. We realize this idea with an adversarial framework [14]. As illustrated in Fig. 4, for each iteration of during training, we randomly sample patches from the predicted tone-mapped diffuse albedo maps $\{\tau(\hat{A}^{(i)})\}_{i=1}^{B}$ in a batch, and separate them into two groups according to the variance of their specularity values: one group $\mathcal{P}_\text{nonspec}$ with low specularity variance ("real"), and the other $\mathcal{P}_\text{spec}$ with high specularity variance ("fake").

We then have a discriminator network D that tries to tell apart these two groups of albedo patches, and introduce an additional GAN loss to our decomposition model:

$$L_\text{SAD} = \mathbb{E}_{p_1 \in \mathcal{P}_\text{nonspec}}[\log D(p_1)] + \mathbb{E}_{p_2 \in \mathcal{P}_\text{spec}}[\log(1 - D(p_2))]. \tag{13}$$

We label our discriminator as self-supervised, as no "real" albedo data is necessary in training our framework. We show in Fig. 7 that SAD significantly improves the quality of the albedo prediction, especially in saturated specular regions.

3.4. End-to-end Training

After combining all the components of our model, there remains an inherent ambiguity between the intensity level of the light and the brightness level of the albedo. Thus, we add a consistency regularizer on the diffuse map $\hat{I}_d$, encouraging

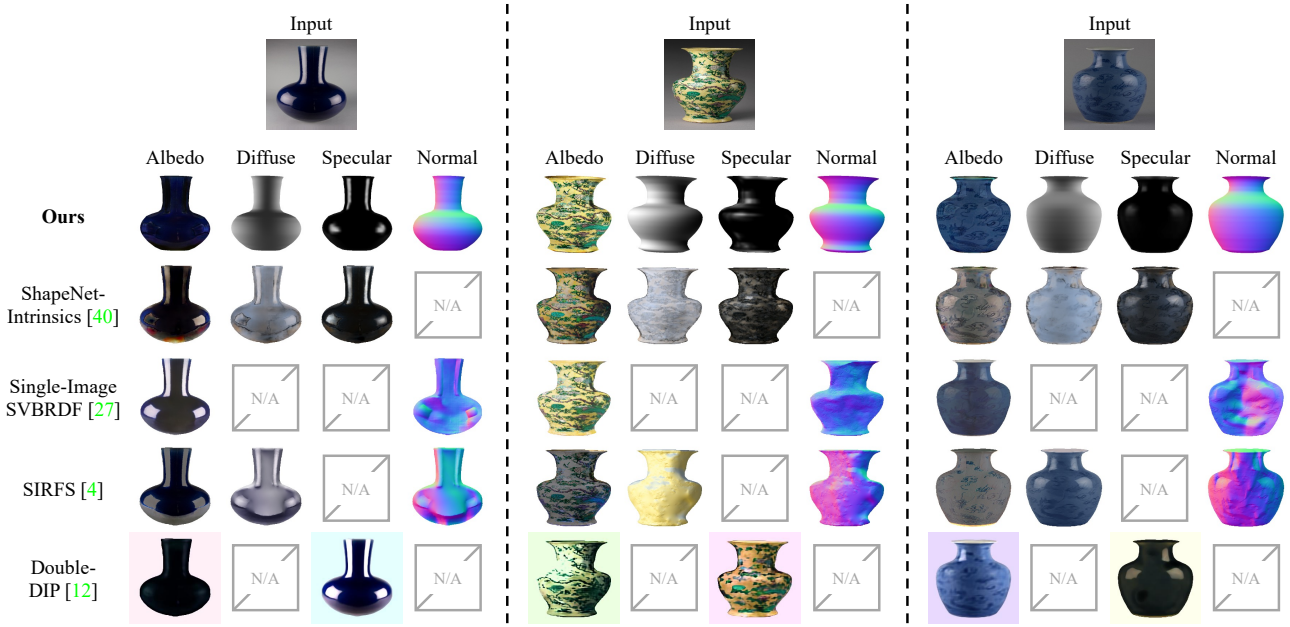


Figure 5: **Qualitative comparison.** We compare the decomposition results of our model against several prior methods. Our method recovers accurate geometry and achieves significantly better decomposition compared to other methods, including supervised models trained on synthetic objects or specially captured data [40, 27].

Method	Albedo[§] $(\times 10^{-2})\downarrow$	Normal[‡] $\downarrow$
Ours	**0.71** ± 0.92	**5.81** ± 0.51
ShapeNet-Intrinsics [40]	3.24 ± 3.24	-
Single-Image SVBRDF [27]	3.34 ± 2.48	36.39 ± 6.92
SIRFS [4]	2.74 ± 3.28	35.85 ± 11.15

Table 1: **Quantitative comparison on synthetic vases.** We evaluate different methods quantitatively on our synthetic vase dataset. Our method significantly outperforms other prior methods. Error metrics: [§]scale-invariant MSE, following Grosse *et al.* [15], [‡]angular deviation in degrees.

its average brightness to reside in a specified interval:

$$L_{\text{diff}} = \max\Big(\Big|\frac{1}{HW}\sum_i \hat{I}_{d,i} - \xi\Big| - \Delta, 0\Big)^2, \qquad (14)$$

where ξ specifies the target brightness level and Δ is the margin, which are respectively set to 0.5 and 0.1 in our experiments. The total loss used to train our model end-to-end is a weighted sum of the five loss terms:

$$L_{\text{total}} = L_{\text{sil}} + \lambda_{\text{im}} L_{\text{im}} + \lambda_{\text{alb}} L_{\text{alb}} + \lambda_{\text{SAD}} L_{\text{SAD}} + \lambda_{\text{diff}} L_{\text{diff}}. \qquad (15)$$

4. Experiments

4.1. Datasets and Implementation Details

Metropolitan Museum vases. We collected a dataset of real vase images from the Metropolitan Museum of Art Collection via the open-access API [2]. We first obtain 5,061 images with the query keyword "vase", and pass them through PointRend [20] to generate bounding boxes and segmentation masks for each vase instance. The images are then cropped and resized to 256×256, and GrabCut [36] is applied to refine the masks. We roughly filter out vases with non-SoR shapes manually, and split the remaining images into 1,888 training images and 526 testing images.

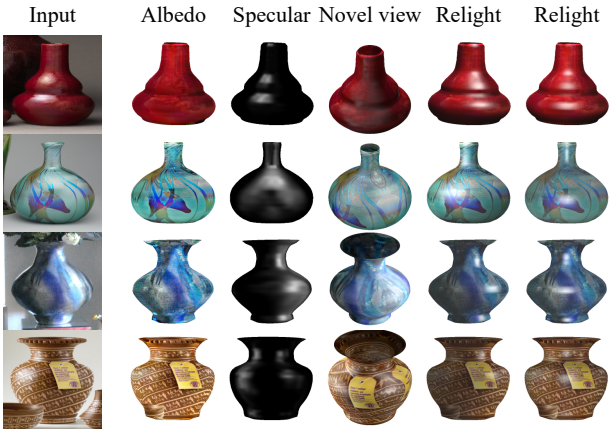


Figure 6: **Novel view and relighting.** Our method recovers accurate geometry and surface material, allowing us to render the vases from novel viewpoints and with new environment lighting. Note that the last two examples are taken from Open Images [22], which shows that the model trained on museum images generalizes well to diverse input images beyond the training distribution.

Method	Pose† ↓	Normal‡ ↓	Albedo§ $(\times 10^{-2})$ ↓	Shininess† ↓	Spec. albedo† ↓	Env. map§ ↓
Supervised	0.16 ± 0.18	5.91 ± 0.56	0.48 ± 0.59	43.61 ± 34.35	0.15 ± 0.12	0.75 ± 0.51
No decomposition	-	-	3.08 ± 2.39	-	-	-
Ours full	0.43 ± 0.40	5.81 ± 0.51	0.71 ± 0.92	42.04 ± 34.83	0.23 ± 0.21	0.41 ± 0.37
w/o L_{alb}	2.77 ± 4.64	8.23 ± 4.54	2.83 ± 2.28	70.53 ± 58.51	0.35 ± 0.25	0.91 ± 0.50
w/o L_{SAD}	0.48 ± 0.49	5.83 ± 0.48	0.75 ± 1.09	43.57 ± 33.59	0.21 ± 0.18	0.41 ± 0.39
w/o L_{diff}	0.45 ± 0.42	5.78 ± 0.49	0.82 ± 0.98	43.27 ± 34.00	0.33 ± 0.21	0.46 ± 0.35

Table 2: **Baselines and ablations.** We evaluate the predictions against the ground-truth on the synthetic vase dataset. The performance of our model approaches the supervised baseline trained with full supervision, and the accuracy of the albedo prediction is clearly higher than lower-bound with no decomposition. The ablation studies validate the effectiveness of each component. Error metrics: †RMSE, ‡angular deviation in degrees, §scale-invariant MSE, following Grosse *et al*. [15].

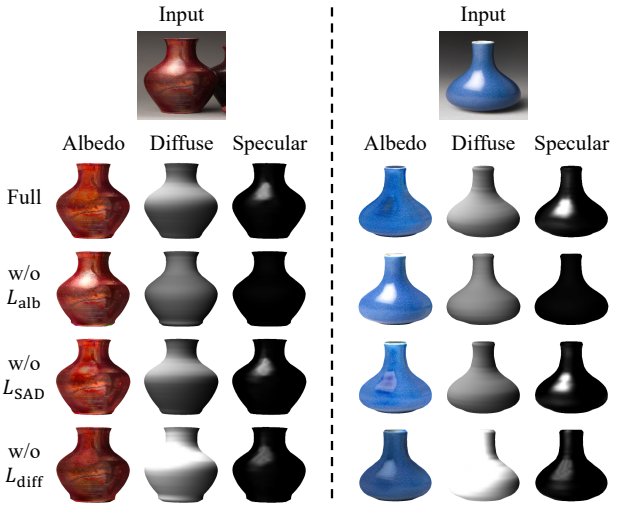


Figure 7: **Qualitative ablation.** We visualize the decomposition results of our full model and its variants. Our full model recovers more realistic albedo and lighting compared to other variants.

Synthetic vases. Since we do not have ground-truth for the decomposed components on the real vases, in order to assess the performance of our model quantitatively, we prepare a dataset of synthetic vases. We randomly generate vase-like SoR curves using combinations of sine curves, and take albedo maps from a public material dataset (CC0 Textures [1]) with various augmentations. We then render the synthetic vases from random elevation angles in $(0, 20°)$, assuming a Phong illumination model with random shininess values in $(1, 200)$ and spherical Gaussian environment lighting [43, 25] with 3 Gaussian lobes randomly sampled from the front upper hemisphere. We generated 4,115 training images and 460 testing images. See the supplementary material for some examples.

Implementation details. The shape network f_S consists of an encoder and two decoder branches. The first decoder branch uses 1D upsampling convolutions to produce an $L \times 1$ radius column $\hat{\mathbf{r}}$, exploiting the structural prior of convolutions to obtain a smooth curve. The second branch is simply 2 FC layers that predict the height $\hat{h}$ and pose $\hat{v}$. The albedo network f_A is a U-Net [35] with 6 downsampling and 6 upsampling layers. The lighting network f_L is similar to f_S, except that it predicts an environment map $\hat{E}$ with 2D upsampling convolutions, and a shininess scalar $\hat{\alpha}$ and a specular albedo scalar $\hat{\rho}$ with 2 FC layers. We use a Least Square GAN [32] for SAD, and the discriminator D is a simple encoder network comprised of 5 downsampling convolution layers. All networks are trained with Adam with a learning rate of 0.0002 and a batch size of 24 for approximately 40k iterations.

Both input images and unwrapped frontal texture maps are 256×256. We use a projective camera with a narrow fixed field of view of $10°$, since the images are cropped around the objects. In practice, we only unwrap the frontal one third of the whole $360°$ circular texture map to ignore the back of the vase and compensate for perspective projection. The sizes of the vertex maps and the environment maps are 32×96 and 16×48 respectively. For visualization, we replicate the texture maps three-fold, and use dimmed textures for the inside of the vase. More details are included in the supplementary material.

4.2. Qualitative Results on Real Vases

Our method recovers geometry, specular material and lighting from a single image, and assumes no ground-truth labels except for object silhouettes during training. To the best of our knowledge, no prior work tackles this problem under such a setting. Nevertheless, we have identified several closest methods, and show a comparison in Fig. 5.

SIRFS [4] is an optimization-based method for decomposing albedo and diffuse shading from a single image, without considering specular materials. ShapeNet-Intrinsics [40] predicts albedo, diffuse shading and specular shading from a single image without explicitly modeling lighting. It is trained on synthetic ShapeNet objects with full supervision. Single-Image SVBRDF [27] is another supervised method that predicts spatially-varying BRDF and environment lighting from a single input image, but assumes that images are captured under camera flash. We also compare to Double-DIP [12], an unsupervised method that decomposes a single image into multiple layers by exploiting the internal image

statistics using "Deep-Image-Prior" networks [42], without requiring training data. It achieves impressive decomposition results in several tasks, including reflection separation, motivating us to test it on the task of specularity separation.

Our method recovers accurate geometry and plausible disentanglement of material and lighting, whereas all other methods fail to decompose these components accurately. Since ShapeNet-Intrinsics and Single-Image SVBRDF are trained with synthetic objects and real objects captured under camera flash respectively, they do not generalize well to these real vases under various lighting environments. SIRFS results in poor decomposition in the presence of specularity, as it assumes only diffuse shading. It is worth noting that Double-DIP in fact achieves plausible decomposition of albedo and specularity in scenarios where the surface textures are simple, although it fails when textures are more complicated. However, it does not consider 3D geometry and thus does not allow for realistic 3D editing.

Novel views and relighting. Since our model recovers the 3D shape, surface material and lighting from a single image, we can easily render the object from arbitrary viewpoints under different lighting conditions, as shown in Fig. 6.

Generalization. We further apply the trained model to diverse input images taken from the Open Images dataset [22], shown in the last two row in Fig. 6. The model generalizes reasonably well to images beyond the training distribution of museum images, where the environment lighting may be more complicated or the vase may be partially occluded.

4.3. Quantitative Comparisons on Synthetic Data

To quantify the prediction accuracy, we evaluate it on our synthetic vase dataset, and report a numerical comparison of different methods in Table 1. We measure the accuracy of the albedo in the predicted region using a scale-invariant mean square error metric [15], since the scales of the albedo intensity and the lighting intensity are ambiguous (one can trade off one for the other), and measure the accuracy of normal maps in degrees of angular deviation. It is evident in Table 1 that our method outperforms other methods for both albedo and shape predictions.

4.4. Baselines and Ablations

We conduct a thorough evaluation of all the predictions of our model on the synthetic dataset and compare the results with two baselines and various ablated models in Table 2 and Fig. 7. The first baseline is a supervised model trained with ground-truth labels on all predictions, which gives an performance upper-bound. We also report a performance lower-bound on the albedo decomposition, obtained by simply evaluating the albedo error metric on the input image without any decomposition. The error of our predicted albedo is clearly much lower than the original undecomposed image, and our model overall achieves high performance close to the supervised baseline on all metrics.

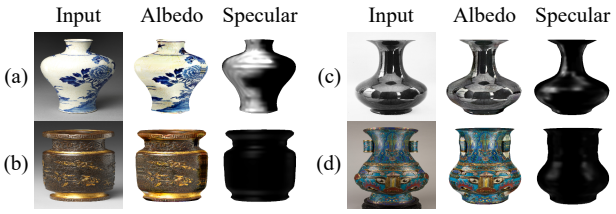


Figure 8: **Limitations.** (a) Incorrect environment lighting and specularity prediction in the presence of high contrast textures. (b) Spatially-varying material properties. (c) Complicated environment lighting. (d) Non-revolutionary shapes.

Comparing our full model to the ablated models, it is evident that without the single-color albedo rendering loss L_{alb}, the model fails to learn various components. The diffuse regularizer L_{diff} largely improves the lighting prediction and consequently other predicted components as well. The albedo discriminator loss L_{SAD} also improves accuracy of albedo prediction, and more importantly, it helps inpaint the albedo in the specular regions as visualized in Fig. 7.

5. Conclusion

We introduce an end-to-end framework for de-rendering a single image into shape, lighting, and surface material components, learning only from single-image collections with 2D silhouettes. Our method works well on both synthetic and real images of revolutionary artefacts and enables applications such as free-view rendering and relighting.

Limitations and future work. Fig. 8 illustrates limitations of our method. First, it tends to predict specularity in bright texture regions, which could lead to unrealistic environment lighting in the presence of high-contrast textures. This could be improved by adding constraints on the lighting model. Second, since we use a Phong model with a single shininess constant for each vase and a low-resolution environment illumination map, our model cannot handle objects with spatially-varying material properties or complex lighting. We intend to incorporate more sophisticated graphics models in future work. Last, as a first step to tackle this extremely challenging problem, we assume revolutionary objects, and hence our model does not work well on objects whose shapes are not revolutionary. However, the proposed components for disentangling lighting and albedo, including the self-supervised discriminator, are not specific to revolutionary objects and it would be interesting to extend these ideas to general real-world objects.

Acknowledgements We would like to thank Christian Rupprecht, Soumyadip Sengupta, Manmohan Chandraker and Andrea Vedaldi for insightful discussions.

References

[1] CC0 Textures. https://cc0textures.com. Accessed: 2020-06. 7, 11

[2] The Metropolitan Museum of Art Open Access. https://metmuseum.github.io. Accessed: 2020-06. 6, 12, 13

[3] James Arvo. Applications of Irradiance Tensors to the Simulation of Non-Lambertian Phenomena. In *SIGGRAPH*, 1995. 4

[4] Jonathan T. Barron and Jitendra Malik. Shape, Illumination, and Reflectance from Shading. *IEEE TPAMI*, 2015. 2, 6, 7

[5] Harry G. Barrow and Jay M. Tenenbaum. Recovering Intrinsic Scene Characteristics from Images. *Computer Vision Systems*, 1978. 2

[6] Sai Bi, Zexiang Xu, Kalyan Sunkavalli, David Kriegman, and Ravi Ramamoorthi. Deep 3D Capture: Geometry and Reflectance from Sparse Multi-View Images. In *CVPR*, 2020. 2

[7] Mark Boss, Varun Jampani, Kihwan Kim, Hendrik P.A. Lensch, and Jan Kautz. Two-shot Spatially-varying BRDF and Shape Estimation. In *CVPR*, 2020. 2

[8] Tao Chen, Zhe Zhu, Ariel Shamir, Shi-Min Hu, and Daniel Cohen-Or. 3-Sweep: Extracting Editable Objects from a Single Photo. In *SIGGRAPH Asia*, 2013. 2

[9] Wenzheng Chen, Jun Gao, Huan Ling, Edward Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to Predict 3D Objects with an Interpolation-based Differentiable Renderer. In *NeurIPS*, 2019. 2

[10] Xin Chen, Yuwei Li, Xi Luo, Tianjia Shao, Jingyi Yu, Kun Zhou, and Youyi Zheng. AutoSweep: Recovering 3D Editable Objects from a Single Photograph. *IEEE TVCG*, 2020. 2

[11] Carlo Colombo, Alberto Del Bimbo, and Federico Pernici. Metric 3D Reconstruction and Texture Acquisition of Surfaces of Revolution from a Single Uncalibrated View. *IEEE TPAMI*, 2005. 2

[12] Yossi Gandelsman, Assaf Shocher, and Michal Irani. "Double-DIP": Unsupervised Image Decomposition via Coupled Deep-Image-Priors. In *CVPR*, 2019. 7

[13] Dan B Goldman, Brian Curless, Aaron Hertzmann, and Steven M Seitz. Shape and Spatially-Varying BRDFs from Photometric Stereo. *IEEE TPAMI*, 2009. 2

[14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. In *NeurIPS*, 2014. 5

[15] Roger Grosse, Micah K Johnson, Edward H Adelson, and William T Freeman. Ground Truth Dataset and Baseline Evaluations for Intrinsic Image Algorithms. In *ICCV*, 2009. 6, 7, 8

[16] Berthold KP Horn. Obtaining Shape from Shading Information. *The Psychology of Computer Vision*, 1975. 2

[17] Michael Janner, Jiajun Wu, Tejas Kulkarni, Ilker Yildirim, and Joshua B Tenenbaum. Self-Supervised Intrinsic Image Decomposition. In *NeurIPS*, 2017. 2

[18] Angjoo Kanazawa, Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Learning Category-Specific Mesh Reconstruction from Image Collections. In *ECCV*, 2018. 2

[19] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3D Mesh Renderer. In *CVPR*, 2018. 4

[20] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. PointRend: Image Segmentation as Rendering. In *CVPR*, 2020. 6

[21] Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep Convolutional Inverse Graphics Network. In *NeurIPS*, 2015. 2

[22] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The Open Images Dataset V4: Unified Image Classification, Object Detection, and Visual Relationship Detection at Scale. *IJCV*, 2020. 6, 8, 12, 13

[23] Pierre-Yves Laffont, Adrien Bousseau, Sylvain Paris, Frédo Durand, and George Drettakis. Coherent Intrinsic Images from Photo Collections. *ACM TOG*, 2012. 2

[24] Edwin H Land and John J McCann. Lightness and Retinex Theory. *Journal of the Optical Society of America*, 1971. 2

[25] Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Inverse Rendering for Complex Indoor Scenes: Shape, Spatially-Varying Lighting and SVBRDF from a Single Image. In *CVPR*, 2020. 1, 2, 7, 11

[26] Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. Materials for Masses: SVBRDF Acquisition with a Single Mobile Phone Image. In *ECCV*, 2018. 1

[27] Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Learning to Reconstruct Shape and Spatially-Varying Reflectance from a Single Image. In *SIGGRAPH Asia*, 2018. 1, 2, 6, 7

[28] Andrew Liu, Shiry Ginosar, Tinghui Zhou, Alexei A. Efros, and Noah Snavely. Learning to Factorize and Relight a City. In *ECCV*, 2020. 2

[29] Guilin Liu, Duygu Ceylan, Ersin Yumer, Jimei Yang, and Jyh-Ming Lien. Material Editing Using a Physically Based Rendering Network. In *ICCV*, 2017. 2

[30] Wei-Chiu Ma, Hang Chu, Bolei Zhou, Raquel Urtasun, and Antonio Torralba. Single Image Intrinsic Decomposition without a Single Intrinsic Image. In *ECCV*, 2018. 2

[31] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier Nonlinearities Improve Neural Network Acoustic Models. In *ICML*, 2013. 10

[32] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley. Least Squares Generative Adversarial Networks. In *ICCV*, 2017. 7

[33] Cody J. Phillips, Matthieu Lecce, and Kostas Daniilidis. Seeing Glassware: from Edge Detection to Pose Estimation and Shape Recovery. In *Robotics: Science and Systems*, 2016. 2

[34] Bui Tuong Phong. Illumination for Computer Generated Pictures. *Commun. ACM*, 1975. 4

[35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI*, 2015. 7

[36] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "GrabCut" — Interactive Foreground Extraction Using Iterated Graph Cuts. In *SIGGRAPH*, 2004. 6

[37] Mihir Sahasrabudhe, Zhixin Shu, Edward Bartrum, Riza Alp Guler, Dimitris Samaras, and Iasonas Kokkinos. Lifting AutoEncoders: Unsupervised Learning of a Fully-Disentangled 3D Morphable Model using Deep Non-Rigid Structure from Motion. In *CVPR Workshops*, 2019. 2

[38] Shen Sang and Manmohan Chandraker. Single-Shot Neural Relighting and SVBRDF Estimation. In *ECCV*, 2020. 2

[39] Soumyadip Sengupta, Angjoo Kanazawa, Carlos D. Castillo, and David W. Jacobs. SfSNet: Learning Shape, Refectance and Illuminance of Faces in the Wild. In *CVPR*, 2018. 2

[40] Jian Shi, Yue Dong, Hao Su, and Stella X. Yu. Learning Non-Lambertian Object Intrinsics across ShapeNet Categories. In *CVPR*, 2017. 2, 6, 7

[41] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance Normalization: The Missing Ingredient for Fast Stylization. *arXiv preprint arXiv:1607.08022*, 2016. 10

[42] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep Image Prior. In *CVPR*, 2018. 8

[43] Jiaping Wang, Peiran Ren, Minmin Gong, John Snyder, and Baining Guo. All-Frequency Rendering of Dynamic, Spatially-Varying Reflectance. *ACM TOG*, 2009. 7, 11

[44] Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised Learning of Probably Symmetric Deformable 3D Objects from Images in the Wild. In *CVPR*, 2020. 2

[45] Yuxin Wu and Kaiming He. Group Normalization. In *ECCV*, 2018. 10

[46] Zexiang Xu, Sai Bi, Kalyan Sunkavalli, Sunil Hadap, Hao Su, and Ravi Ramamoorthi. Deep View Synthesis from Sparse Photometric Images. *ACM TOG*, 2019. 2

[47] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance. In *NeurIPS*, 2020. 2

[48] Ye Yu, Abhimetra Meka, Mohamed Elgharib, Hans-Peter Seidel, Christian Theobalt, and Will Smith. Self-supervised Outdoor Scene Relighting. In *ECCV*, 2020. 2

[49] Ye Yu and William AP Smith. InverseRenderNet: Learning Single Image Inverse Rendering. In *CVPR*, 2019. 2

6. Supplementary Material

6.1. Training Details

All hyper-parameter settings are specified in Table 3 and the network architectures in Tables 4 to 7. Abbreviations of the components are defined as follows:

- Conv$(c_{in}, c_{out}, k, s, p)$: 2D convolution with c_{in} input channels, c_{out} output channels, kernel size k, stride s and padding p.
- Deconv$(c_{in}, c_{out}, k, s, p)$: 2D deconvolution with c_{in} input channels, c_{out} output channels, kernel size k, stride s and padding p.
- Upsample(s): 2D nearest-neighbor upsampling with a scale factor s.
- Linear(c_{in}, c_{out}): linear layer with c_{in} input channels and c_{out} output channels.
- GN(n): group normalization [45].
- IN: instance normalization [41] with n groups.
- LReLU(p): leaky ReLU [31] with a slope p.
- Conv1D and Upsample1D are similarly defined.

6.2. Synthetic Vases

We generate a synthetic vase dataset in order to conduct quantitative assessment of our de-rendering results. Examples of the synthetic vases are shown in Fig. 10. The detailed procedure to generate this dataset is described in the following.

SoR shapes. We simulate vase-like SoR curves $\mathbf{r} \in \mathbb{R}^L$ using a combination of two sine curves, where L is set to be 32, and each entry r_i is given by:

$$\begin{aligned} r_i &= t + f_1(i) + f_2(i) \\ f_1(i) &= a_1 \cdot (1 + \sin(\frac{L-i}{L} \cdot p_1 + \frac{i}{L} \cdot q_1)) \\ f_2(i) &= a_2 \cdot (1 + \sin(p_2 + \frac{i}{L} \cdot q_2)), \end{aligned} \tag{16}$$

where the random variables are $t \sim \mathcal{U}(0.1, 0.3)$, $a_1 \sim \mathcal{U}(0, 0.3)$, $p_1 \sim \mathcal{U}(-\pi, 0)$, $q_1 \sim \mathcal{U}(\frac{\pi}{2}, 2\pi)$, $a_2 \sim \mathcal{U}(0, 0.1)$, $p_2 \sim \mathcal{U}(0, 2\pi)$ and $q_2 \sim \mathcal{U}(\frac{\pi}{2}, 2\pi)$.

We then render the vases with random elevation angles between 0° and 20°, using a projective camera with a field of view of 10°.

Parameter	Value/Range
Optimizer	Adam
Learning rate	2×10^{-4}
Number of iterations	40k
Batch size	24
Loss weight λ_{s}	10
Loss weight λ_{dt}	100
Loss weight λ_{im}	1
Loss weight λ_{alb}	1
Loss weight λ_{SAD}	0.01
Loss weight λ_{diff}	1
Input image size	256×256
Whole unwrapped image size	256×768
Frontal unwrapped image size	256×256
Vertex grid size	32×96
Environment map size	16×48
Field of view (FOV)	10°
Radius $\hat{\mathbf{r}}$	$(0.05, 0.9)$
Radius column height $\hat{h}$	$(0.5, 0.95)$
Pitch angles	$(0^{\circ}, 20^{\circ})$
Roll angles	$(-10^{\circ}, 10^{\circ})$
Translation in X, Y axes	$(-0.2, 0.2)$
Albedo $\hat{A}$	$(0, 1)$
Shininess $\hat{\alpha}$	$(1, 196)$
Specular albedo $\hat{\rho}$	$(0, 2)$
Environment map $\hat{E}$	$(0, 1)$

Table 3: Training details and hyper-parameter settings.

Encoder	Output size
Conv(3, 64, 4, 2, 1) + ReLU	128×128
Conv(64, 128, 4, 2, 1) + ReLU	64×64
Conv(128, 256, 4, 2, 1) + ReLU	32×32
Conv(256, 512, 4, 2, 1) + ReLU	16×16
Conv(512, 512, 4, 2, 1) + ReLU	8×8
Conv(512, 512, 4, 2, 1) + ReLU	4×4
Conv(512, 128, 4, 1, 0) + ReLU	1×1
Decoder	**Output size**
Upsample1D(2) + Conv1D(128, 128, 3, 1, 1) + ReLU	2
Upsample1D(2) + Conv1D(128, 128, 3, 1, 1) + ReLU	4
Upsample1D(2) + Conv1D(128, 128, 3, 1, 1) + ReLU	8
Upsample1D(2) + Conv1D(128, 128, 3, 1, 1) + ReLU	16
Upsample1D(2) + Conv1D(128, 128, 3, 1, 1)	32
↳ Sigmoid $\rightarrow$ output $\hat{\mathbf{r}}$	32
Linear(128, 128) + ReLU	1
Linear(128, 5)	1
Sigmoid $\rightarrow$ output $\hat{h}, \hat{v}$	1

Table 4: Architecture of the shape network f_S. The network outputs radius column $\hat{\mathbf{r}}$, height $\hat{h}$ and camera pose $\hat{v}$ from two branches.

Material. We generate random diffuse albedo maps using texture images from a public material dataset (CC0 Textures [1]), with random augmentations in brightness, contrast and hue. Shininess constant α is randomly sampled between 1 and 196 and specular albedo constant ρ is sampled between 0.1 and 1.

Encoder	Output size
Conv(3, 64, 4, 2, 1) + GN(16) + LReLU(0.2)	128×128
Conv(64, 128, 4, 2, 1) + GN(32) + LReLU(0.2)	64×64
Conv(128, 256, 4, 2, 1) + GN(64) + LReLU(0.2)	32×32
Conv(256, 512, 4, 2, 1) + GN(128) + LReLU(0.2)	16×16
Conv(512, 512, 4, 2, 1) + GN(128) + LReLU(0.2)	8×8
Conv(512, 512, 4, 2, 1) + LReLU(0.2)	4×4
Conv(512, 128, 4, 1, 0) + ReLU	1×1
Decoder	**Output size**
Deconv(128, 512, (2,6), 1, 0) + ReLU	2×6
Upsample(2) + Conv(512, 256, 3, 1, 1) + GN(64) + ReLU	4×12
Upsample(2) + Conv(256, 128, 3, 1, 1) + GN(32) + ReLU	8×24
Upsample(2) + Conv(128, 64, 3, 1, 1) + GN(16)	16×48
↳ Sigmoid $\rightarrow$ output $\hat{E}$	16×48
Linear(128, 128) + ReLU	1
Linear(128, 2)	1
Sigmoid $\rightarrow$ output $\hat{\alpha}, \hat{\rho}$	1

Table 5: Architecture of the light network f_L. The network outputs environment map $\hat{E}$ and specular albedo $\hat{\rho}$ from two branches.

Encoder	Output size
Conv(3, 64, 4, 2, 1) + IN + LReLU(0.2)	128×128
Conv(64, 128, 4, 2, 1) + IN + LReLU(0.2)	64×64
Conv(128, 256, 4, 2, 1) + IN + LReLU(0.2)	32×32
Conv(256, 512, 4, 2, 1) + IN + LReLU(0.2)	16×16
Conv(512, 512, 4, 2, 1) + IN + LReLU(0.2)	8×8
Conv(512, 512, 4, 2, 1) + IN + LReLU(0.2)	4×4
Decoder	**Output size**
Upsample(2) + Conv(512, 512, 3, 1, 1) + IN + SC + ReLU	8×8
Upsample(2) + Conv(512, 256, 3, 1, 1) + IN + SC + ReLU	16×16
Upsample(2) + Conv(512, 256, 3, 1, 1) + IN + SC + ReLU	32×32
Upsample(2) + Conv(256, 128, 3, 1, 1) + IN + SC + ReLU	64×64
Upsample(2) + Conv(128, 64, 3, 1, 1) + IN + SC + ReLU	128×128
Upsample(2) + Conv(64, 3, 3, 1, 1)	256×256
Tanh $\rightarrow$ output $\hat{A}$	256×256

Table 6: Architecture of the albedo network f_A. The network follows a U-Net structure with skip-connections and replaces deconvolution with nearest neighbor upsampling followed by convolution.

Encoder	Output size
Conv(3, 64, 4, 2, 1) + IN + LReLU(0.2)	32×32
Conv(64, 128, 4, 2, 1) + IN + LReLU(0.2)	16×16
Conv(128, 256, 4, 2, 1) + IN + LReLU(0.2)	8×8
Conv(256, 512, 4, 2, 1) + LReLU(0.2)	4×4
Conv(512, 1, 4, 1, 0) $\rightarrow$ output scalar	1×1

Table 7: Architecture of the discriminator network D. The network outputs a single scalar for each input patch.

Lighting. We synthesize environment illumination using 3 random spherical Gaussian lobes [43, 25]:

$$L(\eta) = \sum_{k=1}^{3} \sqrt{\lambda_k} F_k G(\eta; \xi_k, \lambda_k), \;\; G(\eta; \xi, \lambda) = e^{-\lambda(1-\eta \cdot \xi)}, \tag{17}$$

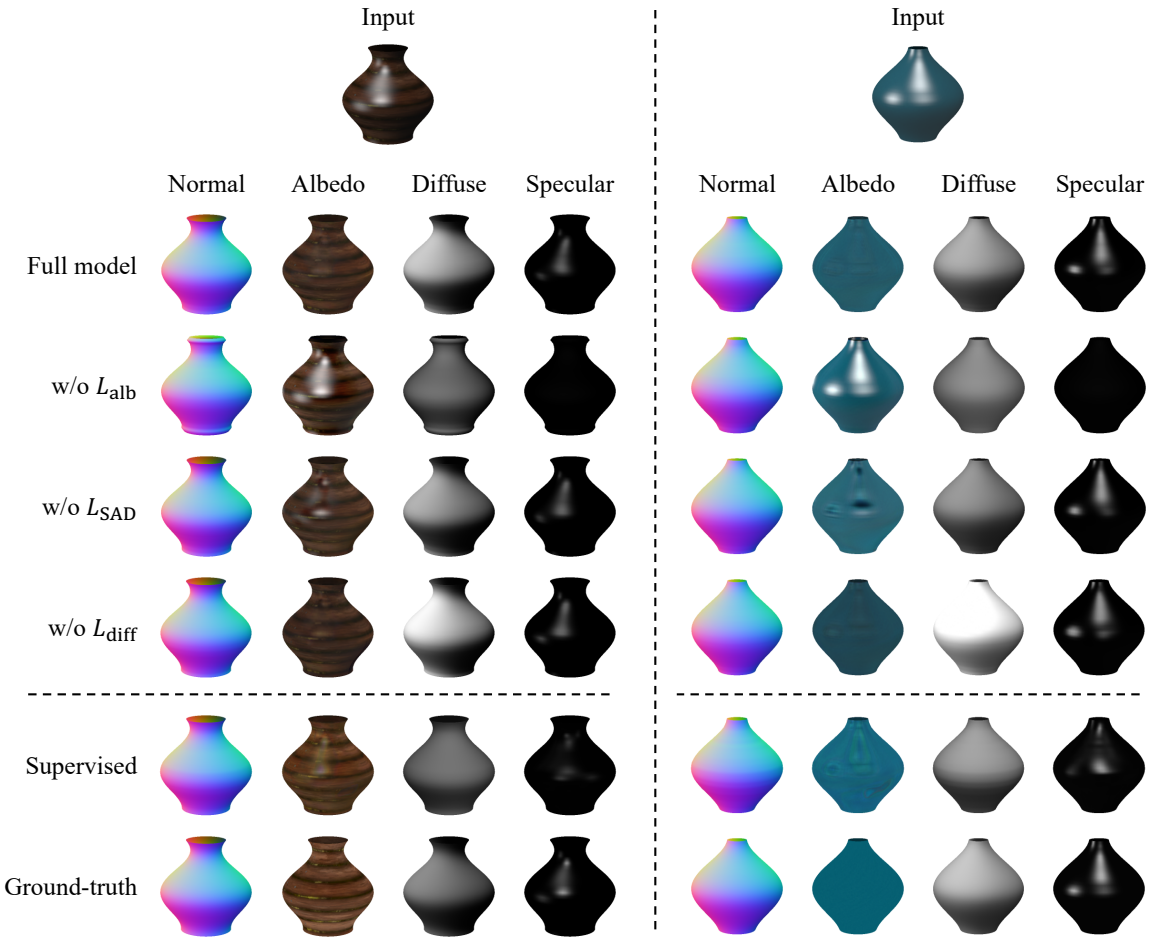


Figure 9: Qualitative comparison of the ablation experiments and the supervised baseline.



Figure 10: Examples of the synthetic vases.

where ξ_k controls the direction of each lobe and is a unit vector randomly sampled from the upper-front quarter of the sphere, $\lambda_k \sim \mathcal{U}(10, 30)$ controls the bandwidth, and $F_k \sim \mathcal{U}(0.1, 0.3)$ controls the intensity.

6.3. Additional Results

Fig. 9 shows a visual comparison of the results obtained from the ablation experiments as well as a supervised baseline, corresponding to the numerical results reported in Table 2.

Additional decomposition and relighting results of real vases are shown in Fig. 11 (from Metropolitan Museum collection [2]) and in Fig. 12 (from Open Images [22]). See the video for more visual results, including animations of rotating vases as well as relighting effects.

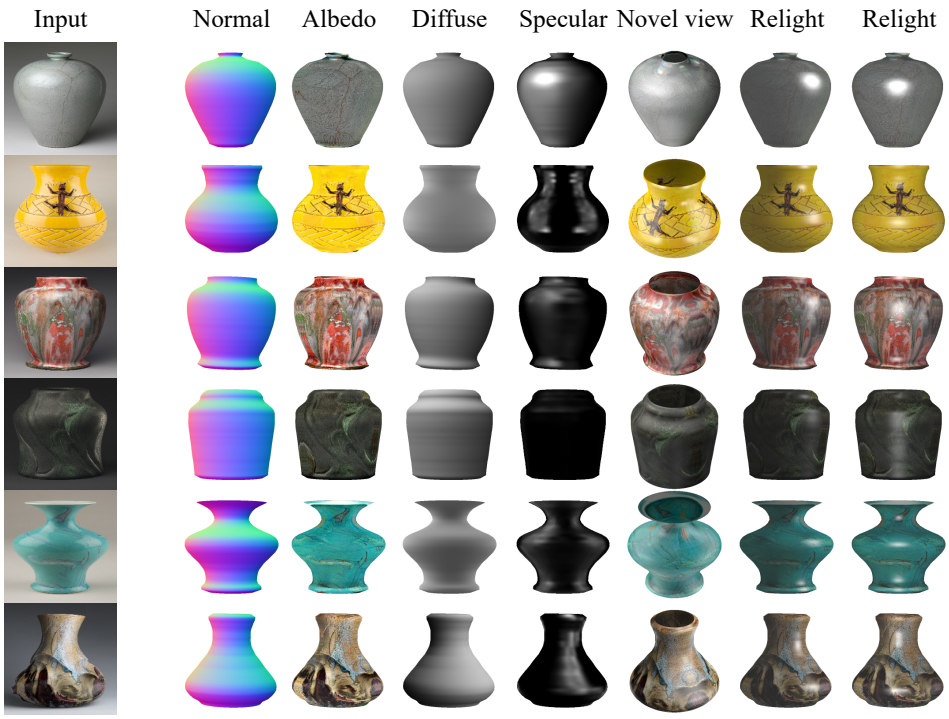


Figure 11: Additional results on Metropolitan Museum collection [2].

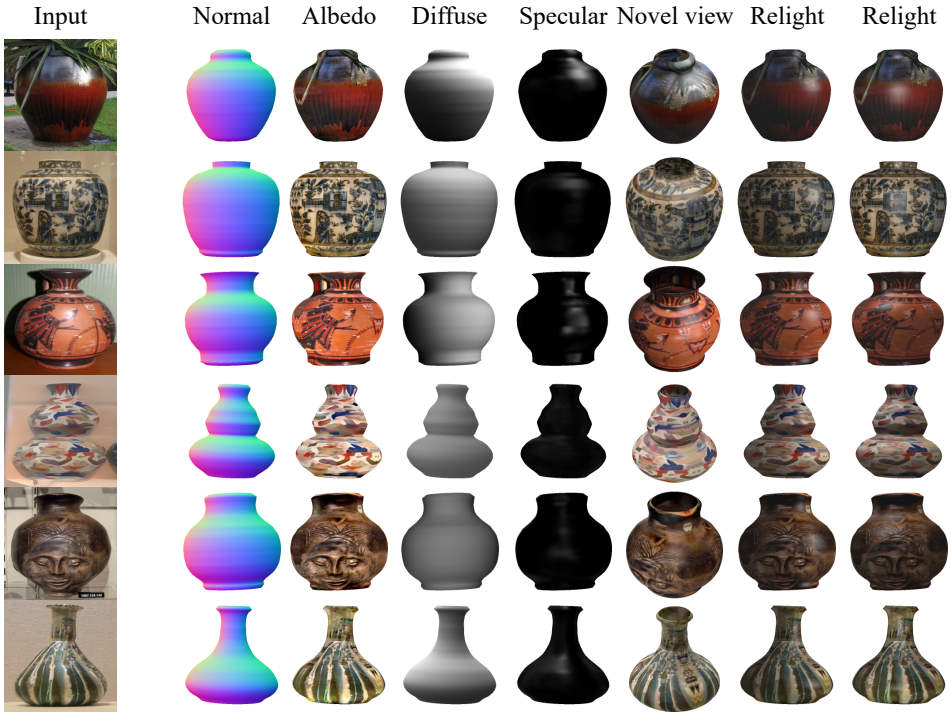


Figure 12: Additional results on Open Images vases [22].

5

De-rendering 3D Objects in the Wild

This work was presented at the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022.

De-rendering 3D Objects in the Wild

Felix Wimbauer[1,2] Shangzhe Wu[2] Christian Rupprecht[2]
[1] Technical University Munich, [2] University of Oxford
wimbauer@in.tum.de {szwu, chrisr}@robots.ox.ac.uk

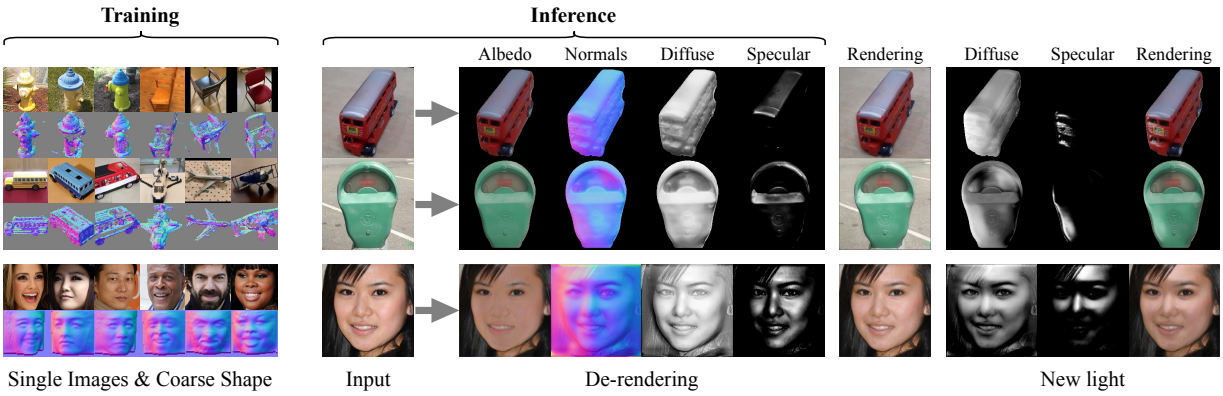


Figure 1. **De-rendering 3D objects from in-the-wild images.** Left: Our method is trained on unconstrained images with coarse shape estimates which can be obtained using off-the-shelf methods or classical structure from motion. Right: Our method de-renders an image into precise shape, material (diffuse albedo, specularity, specular intensity), and lighting (direction, intensity). The image can then be re-rendered and re-lit based on our illumination model. The model generalizes to objects beyond the training categories (*e.g.* parkingmeter).

Abstract

With increasing focus on augmented and virtual reality (XR) applications comes the demand for algorithms that can lift objects from images into representations that are suitable for a wide variety of related 3D tasks. Large-scale deployment of XR devices and applications means that we cannot solely rely on supervised learning, as collecting and annotating data for the unlimited variety of objects in the real world is infeasible. We present a weakly supervised method that is able to decompose a single image of an object into shape (depth and normals), material (albedo, reflectivity and shininess) and global lighting parameters. For training, the method only relies on a rough initial shape estimate of the training objects to bootstrap the learning process. This shape supervision can come for example from a pretrained depth network or—more generically—from a traditional structure-from-motion pipeline. In our experiments, we show that the method can successfully de-render 2D images into a decomposed 3D representation and generalizes to unseen object categories. Since in-the-wild evaluation is difficult due to the lack of ground truth data, we also introduce a photo-realistic synthetic test set that allows for quantitative evaluation. Please find our project page at: https://github.com/Brummi/derender3d

1. Introduction

From a single 2D image, humans can easily reason about the underlying 3D properties of an object, such as the 3D shape, the surface material and its illumination properties. Being able to infer "object intrinsics" from a single image has been a long standing goal in Computer Vision and is often referred to as "inverse rendering" or "*de-rendering*" as it reverses the well-known rendering step of Computer Graphics, where an image is generated from a similar set of object and material descriptors.

De-rendering an image into its physical components, not only plays an important role for general image understanding, but is also key to many applications, such as Augmented/Virtual Reality (XR) and Visual Effects (VFX). In these applications, a decomposed 3D representation can be used to increase the realism by enabling post-processing steps, such as relighting or changing the texture or material properties, which further blurs the line between real and synthetic objects in these environments.

As XR is moving from research and commercial use to consumer devices, a de-rendering method should work on a wide variety of images in the wild to allow a broad adoption of these technologies. While the history of image decomposition literature is long [13, 14], recent learning-based ap-

proaches have demonstrated this capability on specific categories, such as human portraits [38] and synthetic ShapeNet objects [51], by training on ground-truth data, often obtained using synthetic models or sophisticated light stage capturing systems. However, obtaining large-scale ground-truth material and illumination annotations for general objects "in the wild" is much more challenging and infeasible to collect for *all* objects. Models trained on synthetic data often lack sufficient realism, resulting in poor transfer to real images. Models trained on real data usually focus on a single category (*e.g*. faces or birds [11, 17, 24, 25, 57]) and do not generalize to new classes.

On the other hand, another line of research that has recently gained interest, aims to learn 3D objects in an unsupervised or weakly-supervised fashion, without relying on explicit 3D ground-truth [17, 19, 25, 36, 57]. Although impressive results have been demonstrated in reconstructing 3D shapes of simple objects, few of the methods have considered also recovering specular surface materials as this introduces even more ambiguities to the model. Furthermore, they are generally restricted to a single category.

In this paper, we explore the problem of learning non-Lambertian intrinsic decomposition from in-the-wild images without relying on explicit ground-truth annotations. In particular, we introduce a method that capitalizes on the coarse 3D shape reconstructions obtained from unsupervised methods and learns to predict a refined shape as well as further decomposes the material into albedo and specular components, given a collection of single-view images.

At the core of the method lies an image formation process that renders the image from its individual components. The model then learns to decompose the image through a reconstruction objective. Since this formulation is highly ambiguous, the model relies on several additional cues to enable learning a meaningful decomposition. We bootstrap the training using a coarse estimate of the initial shape. This estimate can come from a variety of sources. For datasets such as Co3D [41], where multi-view information is available, we rely on traditional structure-from-motion pipelines (*e.g*. COLMAP [46]). For specific categories such as faces, existing specialized unsupervised methods can be used to obtain a coarse initial shape estimate. We present a simple method that estimates initial material and light properties using the coarse shape, the input image and a simple lighting model. We can then facilitate learning by using the coarse estimates as initial supervisory signals, which avoids many degenerate solutions that would fulfill the reconstruction objective alone. Finally, to further improve the quality of the decomposition, we introduce a third objective, where the image is rendered with randomized light parameters, and a discriminator helps to ensure realistic reconstructions.

While we do need (pseudo) supervision of the coarse shape during training, the final model can directly decompose an input image without any other input. We show that our model produces accurate and convincing image decompositions that improve the state of the art and even generalizes beyond the categories of objects it was trained on. In our experiments, we show that the model works on a wide variety of objects from different datasets. However, as this is the first method to tackle de-rendering in the wild, there is currently no suitable benchmark to quantitatively evaluate the quality of the decomposition. We thus also introduce a synthetic benchmark dataset, using photo-realistic rendering of 10 objects from several viewpoints. Each image is associated with ground truth per-pixel material properties and lighting information that allows us to directly evaluate the decomposition. The new dataset, code and trained models will be published together with the paper.

2. Related Work

This work studies the problem of learning to de-render images of general objects "in the wild", which lies in the intersection of several fields of Computer Vision and Computer Graphics. In this section, we will first discuss the relevant work on Intrinsic Image Decomposition and Inverse Rendering from multiple images, and direct supervision as well as recent unsupervised approaches.

Instrinsic Image Decomposition. Intrinsic Image Decomposition is a classic task, where the main goal is to factorize an image into a reflectance image and a shading image, *i.e*. separating the true surface color from lighting effects. Since this is a highly ill-posed task, traditional methods often rely on additional heuristics and priors. The classic Retinex algorithm [23] assumes that small variations in image intensity result from shading whereas abrupt changes reveal the true reflectance. Many other priors have also been explored over the past few decades, such as global sparsity constraints on the reflectance [10, 43, 48, 49], and explicit geometric constraints on shading assuming Lambertian surface [2, 21]. Recently, researchers have also studied learning-based approaches, by training on synthetic data [16, 33] or multi-illumination images [27, 31]. In this work, we borrow ideas from this area to constrain the albedo extraction, but aim at decomposing the image into explicit material, shape and lighting factors rather than a single shading map, as this allows for relighting and re-rendering.

Supervised Inverse Rendering. Next, we will focus on inverse rendering methods that recover shape, material and illumination from images. Classical Shape-from-Shading approaches assume Lambertian surface properties [13, 14]. Photometric Stereo techniques [1, 12] recover shape, BRDF material and lighting by solving an optimization problem, given multiple images of a scene captured under various lighting conditions and/or from multiple viewpoints. This has been extended with learning-based approaches [3, 4,

6, 61–63] and recently with implicit (neural) representations [5, 52, 59, 64]. While most of these methods still require multiple images at inference, some have learned priors from multiple views that can be used for single image inference [20, 34]. However, capturing multiple images with controlled lighting for either training or inference is challenging and difficult to apply to objects "in the wild", which is main target of this paper.

The de-rendering task can also be learned with direct supervision, often using synthetic data, like ShapeNet [7] objects [6, 8, 9, 32, 51], synthetic faces/bodies [15, 22, 47], near-planar surfaces [28], indoor scenes [26] or other synthetic objects [16, 29, 45]. However, generating large-scale realistic synthetic data that captures the level of complexity of the real world is challenging, and hence it remains questionable how well these methods generalize to real images. As inverse rendering and relighting of faces and persons is particularly useful, relighting datasets for real faces have been collected using light stage setups [37, 38, 53, 54]. This approach, however, is not feasible for general objects.

Unsupervised Inverse Rendering. Recently, there has been an increasing interest in developing unsupervised or weakly supervised methods for inverse rendering tasks. Several works have attempted to learn 3D shapes of object categories, such as faces and birds, from only single-view image collections [11, 17, 25, 57, 60], with weak supervision such as 2D keypoints, masks, category template shapes or assumptions like symmetry. Most of these focus on shape learning and do not tackle material and lighting decomposition specifically or assume a simple Lambertian shading model. Wu *et al*. [56] recovers shape, shiny material and environment lighting, but focuses only on a single specific type of object—vases—and assumes rotational symmetry.

Unlike all these approaches, this work aims at recovering specular material and illumination on *general objects* from images in the wild, with only coarse geometry estimations during training, that can be obtained from existing methods.

3. Method

In this section we will describe the model and training scheme of our method. Fig. 2 shows an overview of the decomposition, training procedure and losses.

3.1. Rendering - Image Formation Model

Essential to our method is modeling the image formation process, *i.e*. rendering an image from its intrinsic components. Our method learns to invert this process—*de-rendering* the image—by extracting the intrinsic components from an input image $\mathbf{I}_{\text{in}} \in [0,1]^{3\times H\times W}$.

While the rendering process is usually deterministic, it is however surjective. This means that due to the highly complex nature of the image formation process, the inverse is ambiguous and many different combinations of intrinsic materials map to the same image.

We deal with the highly ambiguous inverse rendering step in three ways. First, we make reasonable assumptions about the object's material that drastically simplify the rendering process. Second, we leave enough flexibility in the rendering process, such that the model can *learn* to overcome the approximations used in the first step. Finally, we use coarse shape supervision using traditional methods or existing, object-specific solutions to bootstrap the learning process and to avoid degenerate solutions.

Shape. As the shape of an object has a strong influence on its shading, we will directly link these two components. To compute per-pixel shading, we require a per-pixel normal map $N \in [-1,1]^{3\times H\times W}$. Given an image, directly predicting a normal map with a neural network is problematic, as there is no incentive to adhere to a global shape. Thus, we compute the normal map N_D from a depth map $D \in [d_{\min}, d_{\max}]^{H\times W}$. However, fine geometric details (*e.g*. scratches or small reliefs) have a strong influence on the normals, but little on the global shape. Thus, we predict both a depth map D (and compute the corresponding normal map N_{D}) and a refinement normal map N_{ref} from the image, and combine them:

$$N = \frac{N_{\text{D}} + N_{\text{ref}}}{\|N_{\text{D}} + N_{\text{ref}}\|} \tag{1}$$

Light & Material. Using a very expressive lighting model would allow us to capture highly complex effects for photo-realistic rendering. However, we find that such models add significant difficulty to the learning of inverse tasks without further supervision. To model the lighting, we hence rely on Phong Illumination [40], which considers ambient, diffuse, and specular light components. Additionally, we make the following assumptions. We can observe during training that the shading from the one dominant light source (*e.g*. the sun) is a very important hint for the model. Further, multiple light sources would introduce more ambiguity, potentially harming the correctness of the predicted geometry. Therefore, we model the light as a single directional light source and a global ambient light, both emitting perfectly white light. It is parameterized by ambient and directional strength $s_{\text{amb}}, s_{\text{dir}} \in [0,1]$, and a light direction $l \in \mathbb{S}^2$. For both terms, we use a combined per-pixel albedo map $A \in [0,1]^{3\times H\times W}$. Specularity is a very complex lighting effect and therefore difficult to extract from a single image. To keep the complexity tractable, we use a global shininess value $\alpha \in [0, \alpha_{\max}]$ and a global specularity intensity $a_{\text{spec}} \in [0,1]$ for the whole object. Summarizing, we represent the lighting as $L = (s_{\text{amb}}, s_{\text{diff}}, l)$ and the intrinsic material properties as $(A, \alpha, a_{\text{spec}})$.

We obtain an image $\hat{\mathbf{I}}$ from shape, material, and lighting through the following rendering equation, where $u \in \Omega =$

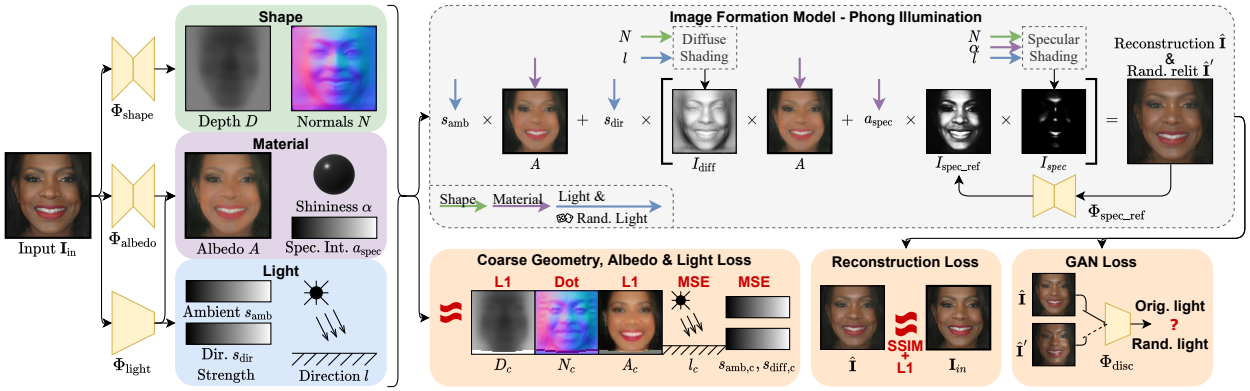


Figure 2. **Model architecture.** Several networks decompose the input image $\mathbf{I}_{\text{in}}$ into shape (depth D and normals N), material (albedo A, shininess α and specular intensity a_{spec}), and light (ambient and directional strength $s_{\text{amb}}, s_{\text{dir}}$ and light direction l). To allow for complex lighting effects, we also introduce a specularity refinement step. Training combines three different loss terms: 1. a loss on the decomposition using coarse estimates 2. a reconstruction loss, and 3. a discriminator loss for a randomly relit image.

$\{1,\dots,H\}\times\{1,\dots,W\}$ represents a pixel location.

$$\hat{\mathbf{I}}_u = \tau\big(\underbrace{s_{\text{amb}}A_u}_{\text{ambient term}} + s_{\text{dir}}\big(\underbrace{N_u^T l A_u}_{\text{diffuse term}} + \underbrace{a_{\text{spec}}\left(r^T v\right)^{\alpha}}_{\text{specular term}}\big)\big), \quad (2)$$

where r is the reflected light direction, and gamma function $\tau(\mathbf{I}_u) = \mathbf{I}_u^{1/\gamma}$, $\gamma = 2.2$ approximates tone-mapping, commonly used to ensure a more even brightness distribution.

3.2. De-rendering

Our network architecture is composed of several subnetworks, that predict the different shape, material, and lighting properties of an input image. From these predictions, we can reconstruct the image with the image formation model described above. However, as mentioned before, due to the plethora of ambiguities in the rendering function, a simple reconstruction objective alone (*e.g.* $\|\mathbf{I}-\hat{\mathbf{I}}\|_2^2$) is not sufficient to learn a meaningful decomposition. To overcome this challenge, we propose a training scheme, with two additional objectives that regularize the learning problem and prevent degenerate solutions. As training data, we use an unconstrained set of images with associated coarse geometry estimates. We use the coarse geometry to generate further coarse estimates for the intrinsic components, which we use in auxiliary loss terms. These coarse constraints force the model to predict a semantically correct disentanglement on a global level.

Extracting Coarse Light & Albedo. Because coarse shape (depth map D_c and its normal map N_c) alone does not suffice to constrain the decomposition, we also compute coarse light and albedo estimates from the geometry information through two optimization steps.

As we only need coarse estimates of the intrinsic components, we can make the simplifying assumption that the per-pixel coarse brightness $B \in [0,1]^{H\times W}$ (computed in HSV color space) of the input image is proportional to the combination of ambient and diffuse shading, and discard specular lighting effects. This translates to an albedo map with constant brightness. Given light information, we can obtain the relevant shading map from the coarse geometry. Therefore, we optimize the coarse light parameters $L_c = (s_{\text{amb},c}, s_{\text{dir},c}, l_c)$ such that the aggregated shading map corresponds to the brightness of the input image.

$$\arg\min_{L_c} \sum_{u\in\Omega} \left(2B_u - \left(s_{\text{amb},c} + s_{\text{dir},c} N_{c,u}^T l_c\right)\right)^2 \quad (3)$$

We fix the albedo brightness to $\frac{1}{2}$ to avoid color saturation effects and consequently add a scaling factor of 2 for B. Here, N_c is the coarse normal map. With this light estimate, an initial albedo estimate $\tilde{A}_c$ can be obtained by inverting the shading equation:

$$\tilde{A}_{c,u} = \mathbf{I}_u \left(s_{\text{amb},c} + s_{\text{dir},c} N_{c,u}^T l_c\right)^{-1} \quad (4)$$

However, because of the coarseness of the geometry and no modeling of specularity effects, an estimate using this formulation alone will contain many artifacts. To regularize the estimate $\tilde{A}_c$ we refine it using another optimization step. Similar to the constraints used in the intrinsic image decomposition literature [50, 58], we apply total variation regularization (TV) on the albedo as well as a data term that retains the image gradients (*i.e.* edges):

$$\begin{aligned}\arg\min_{A_c} \|\delta_x A_c - \delta_x \tilde{A}_c\|_2 + \|\delta_y A_c - \delta_y \tilde{A}_c\|_2 + \\ \lambda_{\text{TV}} \left(\|\delta_x A_c\|_1 + \|\delta_y A_c\|_1\right) .\end{aligned} \quad (5)$$

We use δ_x and δ_y to signify the computation of image gradients, which can, for example, be obtained by applying the Sobel operator to the image. We obtain L_c and A_c by op-

timizing Eq. (3) and Eq. (5) respectively using gradient descent, which takes less than a second and can be precomputed for each image (see Fig. 3).

Learning to De-render. We use three different neural networks to predict the intrinsic components from the input image $\mathbf{I}_{\text{in}}$. A shape network Φ_{shape} predicts both the depth map $D, D_u \in [d_{\min}, d_{\max}]$ and the normal refinement map N_{ref}, which is normalized after prediction and used to obtain the final normal map N with Eq. (1). The albedo network Φ_{albedo} predicts the albedo map $A, A_u \in [0, 1]$, and the light network Φ_{light} predicts the light parameters $s_{\text{amb}}, s_{\text{dir}} \in [0, 1]$, as well as, shininess $\alpha \in [0, \alpha_{\max}]$ and specular intensity $a_{\text{spec}} \in [a^{\text{spec}}_{\min}, a^{\text{spec}}_{\max}]$.

We train our model using complementary losses on the decomposition and on the rendered image. This makes the network adhere to globally accurate components, while achieving more detailed reconstructions. The loss is computed using the (precomputed) coarse shape, albedo, and light information as pseudo supervision.

$$\begin{aligned}\mathcal{L}_c = \sum_{u\in\Omega} \lambda_D \|D_u - D_{c,u}\|_1 - \lambda_N N_u^T N_{c,u} + \\ \lambda_A \|A_u - A_{c,u}\|_1 + \lambda_L \|L - L_c\|_2^2\end{aligned} \tag{6}$$

Additionally, there are two losses on the rendered image. First, we apply a reconstruction loss between the rendered and the input image to train our model to capture all local details in the decomposition. Specifically, this loss term is computed from the combination of a per-pixel L1 loss and the patch-based structural similarity score $\text{SSIM}(\mathbf{I}, \hat{\mathbf{I}})$ [55].

$$\mathcal{L}_{\text{rec}} = \frac{1}{|\Omega|} \sum_{u\in\Omega} \|\mathbf{I}_u - \hat{\mathbf{I}}_u\|_1 + \frac{1}{2}\left(1 - \text{SSIM}(\mathbf{I}, \hat{\mathbf{I}})\right) \tag{7}$$

While the reconstruction loss gives a very strong training signal, there often remains some ambiguity, in that, given fixed light, certain details can be modeled either by the material (light independent) or the shape component (light dependent). Such mistakes only become apparent when we render the image under new lighting conditions L' (mainly influenced by the direction l') .

To ensure that we achieve a semantically correct decomposition, we therefore also introduce an adversarial formulation. Concretely, we render two images in each forward pass: one with the predicted lighting conditions, denoted as $\hat{\mathbf{I}}$, which is also used in the reconstruction loss term, and one with randomly sampled lighting conditions, denoted as $\hat{\mathbf{I}}'$. We then train a discriminator network $\Phi_{\text{disc}} \in \mathbb{R}$ to score, whether an image was rendered using the original lighting conditions or whether it was re-lit. For this we use the discriminator from LSGAN [35]. Using the reconstructed image $\hat{\mathbf{I}}$ instead of the original image $\mathbf{I}_{\text{in}}$ as positive example when training Φ_{disc}, has the advantage that the network cannot use artifacts from the image formation models as hints as to whether the image was re-lit or not, as both, real and fake examples come from the same pipeline. The loss term on the relit image is computed as: $\mathcal{L}_{\text{gan}} = (1 - \Phi_{\text{disc}}(\hat{\mathbf{I}}'))^2$. We can then train the whole model end-to-end using

$$\mathcal{L} = \mathcal{L}_c + \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{gan}}\mathcal{L}_{\text{gan}} , \tag{8}$$

to learn to de-render an image into its intrinsic components.

Refinement. While simplifying the specularity model to two scalars allows for stable training, it can be limiting when there are large differences in material properties across the object. To alleviate this issue, similar to the way we allow the normals to deviate from the underlying shape via a refinement map, we predict a per-pixel specularity refinement map $I_{\text{spec_ref}}(\hat{\mathbf{I}})$ from the *output* image. We then multiply $I_{\text{spec_ref}}$ with the specularity term and re-compose the image.

4. Experiments

We conduct extensive experiments to evaluate our method and its individual components.

4.1. Datasets and Metrics

We use three different datasets to cover a wide variety of objects: faces, a collection of common objects in the wild, and a new synthetic and photo-realistic test set with ground truth annotations. Please see the supplement for all details.

CelebA-HQ [18] is a large-scale human face dataset, consisting of 30k high-resolution portrait pictures of celebrities. We roughly crop out the face area and use the corresponding train/val/test split of the CelebA dataset. To obtain the rough initial geometry estimate D_c, we use [57] at a reduced resolution of 64×64.

Co3D [41] is a collection of nearly 19,000 videos capturing objects from 50 MS-COCO [30] categories, that come with per-frame depth, camera pose data, and reconstructed sparse point clouds. First, we use the Point Cloud Library [44] to compute surface normals from the point clouds. The resulting depth and normal maps are very sparse (see Fig. 3). We select a subset of the categories and obtain 23895 training and 2817 testing images.

COSy (Common Objects Synthetic) is a test set we have created to allow for quantitatively evaluation of image decomposition methods. This is necessary as there does not exist a dataset that combines photorealistic images with precise image decomposition ground truth. We hand-select 10 freely available and photorealistic 3D scenes for the Blender 3D modeling software[1] and define 4 different camera views for each. Additionally to the rendered image, we also save the diffuse albedo map, normal map, and foreground mask. We do not use this dataset of 40 images only for testing.

[1] https://blender.org

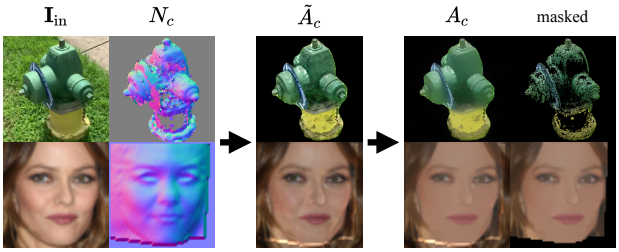


Figure 3. **Coarse Albedo Optimization.** For training, we precompute coarse albedo estimates A_c from the input image $\mathbf{I}_{\text{in}}$ and the coarse shape estimate N_c. We first approximate the light by assuming a Lambertian shading model and inverting it. Subsequently, we apply a smoothing optimization to remove artifacts.

Post-processing for Training. For every image, we compute the normal map from the rough initial depth estimate and optimize a light and albedo approximation (see Fig. 3).

We train one model for the **CelebA-HQ** dataset and one model for the **Co3D** dataset. All hyper-parameters are the same for both models (see supplementary material) except for $\lambda_L = 0$ on **Co3D**, as the geometry is very sparse and the light estimate is often not accurate enough, hindering the convergence of the training.

Metrics. As other methods use different image formation models, and thus obtain a different representation for shading, it is not possible to directly compare shading maps. This constrains us to quantitative evaluation of normal and albedo map only. Aside from the common $\mathcal{L}_1$, $\mathcal{L}_2$, and SSIM error metrics, we use mean angle deviation in degrees $\text{DIA}(N, \hat{N}) = \frac{1}{|\Omega|}\sum_{u\in\Omega}\cos^{-1}(N_u^T\hat{N}_u)$ for normals, and the scale-invariant error $\text{SIE}(A, \hat{A}) = \frac{1}{|\Omega|}\sum_{u\in\Omega}\|A_u - \mu_A - (\hat{A} - \mu_{\hat{A}})\|_2^2$ for albedo, as it can be estimated only up to a constant scale factor. Here, μ_A is the average albedo across the whole image $\mu_A = \frac{1}{|\Omega|}\sum_{u\in\Omega} A_u$.

4.2. Results

Qualitative Evaluation. To demonstrate the capabilities of our method, we first evaluate it on a diverse selection of samples, as shown in Fig. 4. Regardless of the category and background, we obtain globally correct results with a very high fidelity. Critically, even though the COSy dataset object categories are *not* part of the training categories, we observe the same level of detail. This demonstrates our method's generalization capabilities to novel objects and categories. In addition to the decomposition results, we also show that our method produces realistic images and shading maps when changing the light.

Further, we compare our results with state-of-the-art methods for intrinsic image decomposition [2, 29, 45, 51] in Fig. 5. All methods are able to predict reasonable albedo maps, that capture the major color components. However, the albedo maps of [2, 29, 45] still contain color gradients and lighting effects around the edges and corners. [51] is able to remove almost all specular components of the roof but introduces artifacts, for example, at the top of the roof. Our method successfully removes shading effects and does not contain artifacts. For normal prediction, [2] does not capture the shape of the object, nor fine details. Although [29] and [45] predict seemingly detailed normal maps, closer inspection shows that they are not physically grounded (*e.g.* the normals on the windows point upwards). Our normal map is both detailed and it adheres to the global shape. Finally, our shading maps are computed from the normal map and additional material properties. This is why they are similarly detailed and based on physical properties. Diffuse and specular effects are captured correctly. The diffuse shading map predicted by [51] is very detailed, but does not capture the light direction.

	Normal N		Albedo A	
Model	**MSE** ↓	**DIA** ↓	**SIE** ↓	**SSIM** ↑
SIRFS [2]	0.331	52.994	0.113	0.724
ShapeNet-Intr. [51]	N/A	N/A	0.114	0.726
SISaSVBRDF [29]	0.288	42.801	0.112	0.752
Neur. Rel. [45]	0.228	41.603	0.093	0.723
Ours	**0.173**	**37.807**	**0.075**	**0.760**

Table 1. **Comparison with the state of the art.** We show good improvements over previous methods on the COSy dataset. [29] and [45] were trained using flash photographs.

	Normal N		Albedo A		Specular I_{spec}	
Model	**MSE** ↓	**DIA** ↓	**SIE** ↓	**SSIM** ↑	**MSE** ↓	**L1** ↓
No Albedo	0.162	36.5	0.088	0.750	0.124	0.077
No Shape	0.506	68.7	0.079	0.757	0.108	0.058
No GAN	0.169	37.2	0.075	0.762	0.123	0.073
Ours	0.173	37.8	0.075	0.760	0.112	0.059

Table 2. **Ablation.** Results on the COSy dataset when deactivating components of our model. Concretely, we set the $\lambda = 0$ coefficient for the respective loss term and then do a full training run.

Quantitative Evaluation. As neither **CelebA-HQ** nor **Co3D** contain explicit, dense ground truth for the different intrinsic image components, we apply the **Co3D** model on our newly introduced **COSy** dataset to perform quantitative evaluations. We compare against state-of-the-art image decomposition methods [2, 29, 45, 51], as shown in Tab. 1. Across all metrics, we achieve best accuracy on normal and albedo extraction. The fact, that our method (like the others) was not trained on this test set, highlights its strong generalization capabilities.

Single Image Relighting. To demonstrate the usefulness of de-rendering, we perform relighting on the **CelebA-HQ** dataset. Fig. 6 shows comparison of our method with state-of-the-art face relighting methods [65] and [15]. As a re-

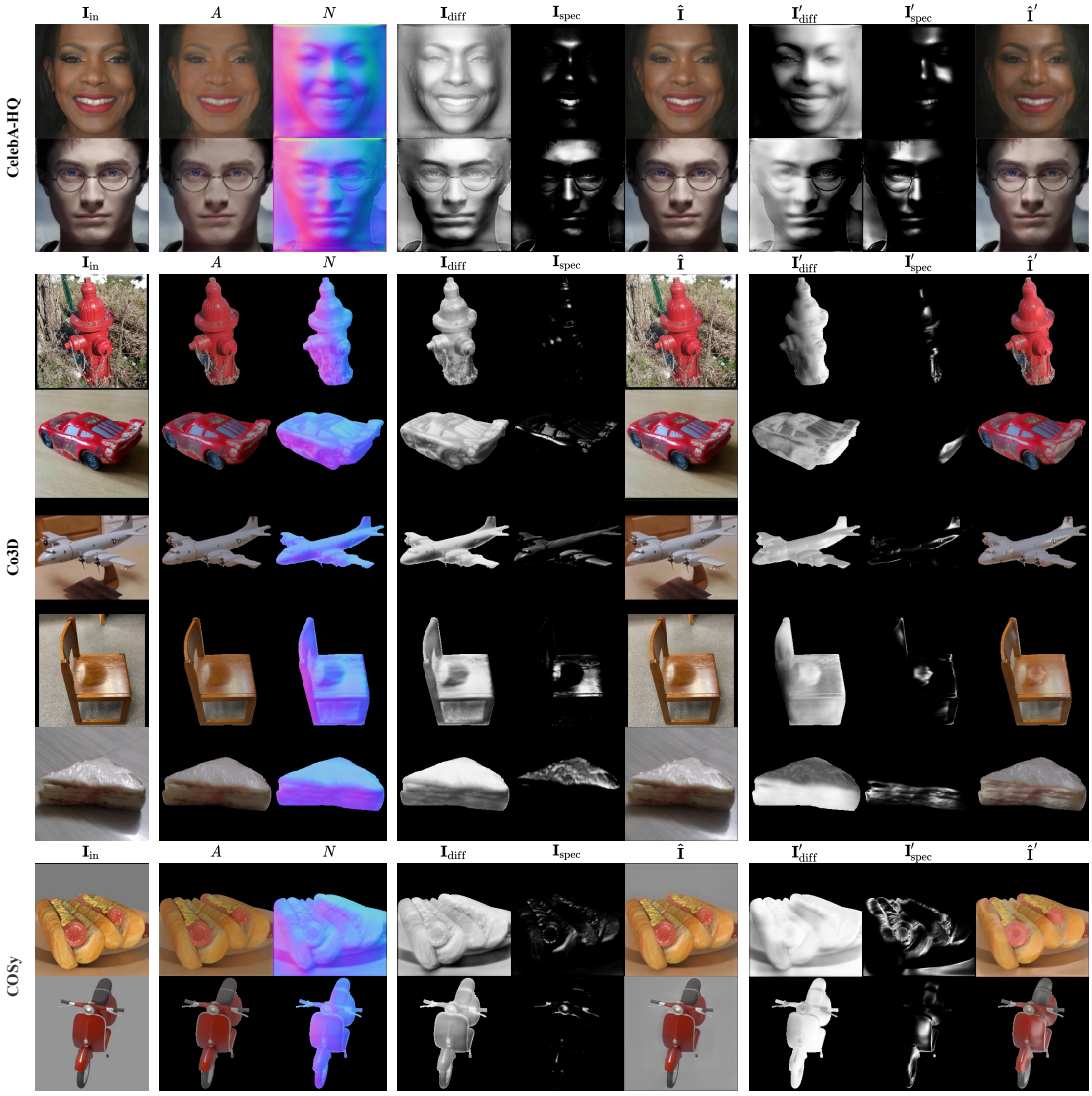


Figure 4. **Qualitative results.** We train a model on each of the two datasets **CelebA-HQ** and **Co3D**, and show the respective decomposition results from the test sets. To highlight the generalization capabilities, we also apply the **Co3D** model to samples from our synthetic test set **COSy**. Every row contains the input image $\mathbf{I}_{\text{in}}$, predicted albedo A and normals N, diffuse shading map $\mathbf{I}_{\text{diff}}$, specular shading map $\mathbf{I}_{\text{spec}}$ and reconstructed image $\hat{\mathbf{I}}$. Further, we also show the shading maps ($\mathbf{I}'_{\text{diff}}$, $\mathbf{I}'_{\text{spec}}$) and reconstructed image $\hat{\mathbf{I}}'$ under new lighting conditions. Our model achieves a high level of detail on shape and material reconstruction and convincing relighting results.

sult of the underlying explicit image formation model, our method produces visually correct relighting results, which are more color accurate than [65] and contain much fewer artifacts than [15]. This demonstrates that our method can not only perform very well in intrinsic image decomposition, but also compete with methods, that were specifically designed for certain sub-tasks.

4.3. Ablation Study and Analysis

We also conduct several ablation studies and further analyses of the impact of the individual model components.

Loss Components. At the heart of our method is the combination of three losses: a coarse loss on the different intrinsic components, a reconstruction loss, and a discriminator loss. We deactivate each component and then evaluate the resulting models on COSy, as shown in Tab. 2.

When deactivating the albedo and shape losses ($\lambda_A = 0$ and $\lambda_D = \lambda_N = 0$ respectively), the predictions of the respective components become significantly worse. The discriminator loss does not have a large influence on the quality of the albedo and normal accuracy, however, it stabilizes the accuracy of the specular shading map.

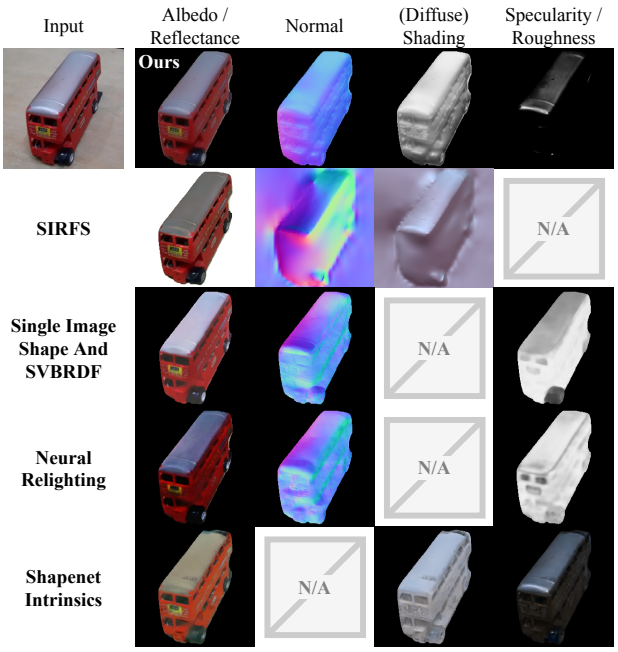


Figure 5. **Qualitative comparison with state of the art.** We show superior image decomposition results compared to SIRFS [2], Single Image Shape and SVBRDF [29], Neural Relighting [45], and ShapeNet Intrinsics [51].

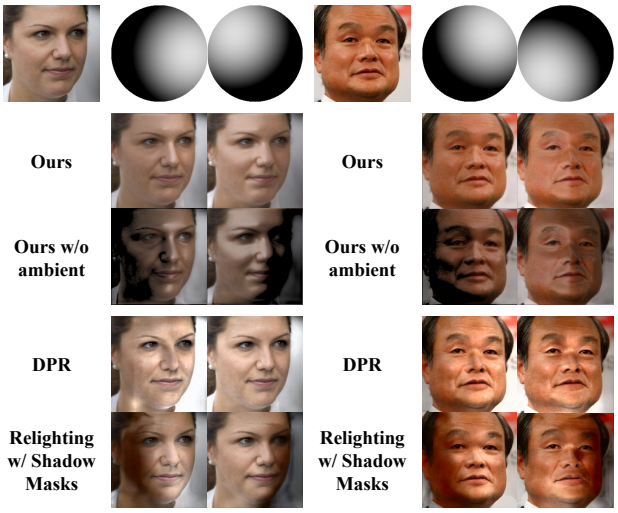


Figure 6. **Qualitative comparison with face relighting methods**. We use our model to relight images from the **CelebA-HQ** test set and compare with state-of-the-art face relighting methods. Our method shows better color accuracy and more robustness.

Geometry & Albedo Improvement. Fig. 3 compares on two (test-set) examples the coarse input during training and the prediction of the trained model. This is to verify that albedo and normal map predictions achieve a significantly higher level of detail and completeness compared to their initial, coarse counterparts that are used to supervise the training process. This is a result of the reconstruction and GAN losses and the explicit image formation model.

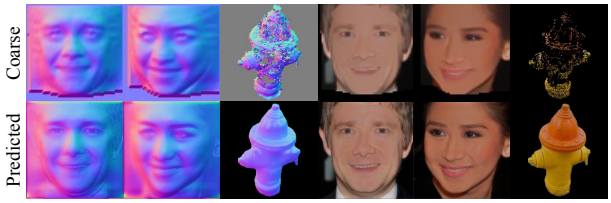


Figure 7. **Coarse estimates vs. prediction.** We compare the rough shape and albedo estimates that we use during training with predictions from the final model and show large improvements. Note that the rough estimates are not available during test time.

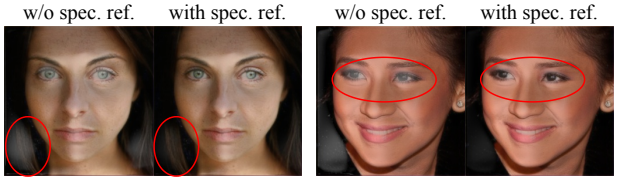


Figure 8. **Specular Refinement.** The specular refinement network improves regions where the complexity of underlying material is difficult to describe explicitly with the lighting model.

Specular Refinement. Fig. 8 demonstrates the specular refinement on two re-lit portrait images. The assumption of shared specular parameters for the entire image, can sometimes lead to specular artifacts in complex regions, which is especially important during relighting. The network effectively removes artifacts both on the hair and around the eyes, leading to a more realistic output.

5. Conclusions

We have presented a method that can factorize in-the-wild images of objects into their intrinsic components shape, material, and lighting. Our proposed learning pipeline does not rely synthetic datasets and only uses sparse geometry estimates during training, which can be obtained using off-the-shelf unsupervised methods. Through a series of ablation studies, we have demonstrated the importance of the different components of our method, particularly the coarse losses. The proposed method achieves high accuracy for all intrinsic components, both on in- and out-of-distribution images, which we measure on our newly introduced synthetic image decomposition test set that we hope will become a new benchmark for de-rendering images in the wild.

Acknowledgments Shangzhe Wu is supported by Meta Research. Christian Rupprecht is supported by Innovate UK (project 71653) on behalf of UK Research and Innovation (UKRI) and the Department of Engineering Science at the University of Oxford.

References

[1] Neil Alldrin, Todd Zickler, and David Kriegman. Photometric stereo with non-parametric and spatially-varying reflectance. In *CVPR*, pages 1–8. IEEE, 2008. 2

[2] Jonathan T. Barron and Jitendra Malik. Shape, Illumination, and Reflectance from Shading. *IEEE TPAMI*, 2015. 2, 6, 8

[3] Sai Bi, Zexiang Xu, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. Deep reflectance volumes: Relightable reconstructions from multi-view photometric images. In *ECCV*, pages 294–311. Springer, 2020. 2

[4] Sai Bi, Zexiang Xu, Kalyan Sunkavalli, David Kriegman, and Ravi Ramamoorthi. Deep 3D Capture: Geometry and Reflectance from Sparse Multi-View Images. In *CVPR*, 2020. 2

[5] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P.A. Lensch. Nerd: Neural reflectance decomposition from image collections. In *ICCV*, 2021. 3

[6] Mark Boss, Varun Jampani, Kihwan Kim, Hendrik P.A. Lensch, and Jan Kautz. Two-shot Spatially-varying BRDF and Shape Estimation. In *CVPR*, 2020. 2, 3

[7] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 3

[8] Wenzheng Chen, Huan Ling, Jun Gao, Edward Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. In *NeurIPS*, 2019. 3

[9] Wenzheng Chen, Joey Litalien, Jun Gao, Zian Wang, Clement Fuji Tsang, Sameh Khamis, Or Litany, and Sanja Fidler. Dib-r++: Learning to predict lighting and material with a hybrid differentiable renderer. In *NeurIPS*, 2021. 3

[10] Elena Garces, Adolfo Munoz, Jorge Lopez-Moreno, and Diego Gutierrez. Intrinsic images by clustering. *Computer Graphics Forum*, page 1415–1424, 2012. 2

[11] Shubham Goel, Angjoo Kanazawa, and Jitendra Malik. Shape and viewpoint without keypoints. In *ECCV*, 2020. 2, 3

[12] Dan B Goldman, Brian Curless, Aaron Hertzmann, and Steven M Seitz. Shape and Spatially-Varying BRDFs from Photometric Stereo. *IEEE TPAMI*, 2009. 2

[13] Berthold KP Horn. Obtaining Shape from Shading Information. *The Psychology of Computer Vision*, 1975. 1, 2

[14] Berthold K. P. Horn and Michael J. Brooks. *Shape from Shading*. MIT Press, Cambridge Massachusetts, 1989. 1, 2

[15] Andrew Hou, Ze Zhang, Michel Sarkis, Ning Bi, Yiying Tong, and Xiaoming Liu. Towards high fidelity face relighting with realistic shadows. In *CVPR*, pages 14719–14728, 2021. 3, 6, 7

[16] Michael Janner, Jiajun Wu, Tejas Kulkarni, Ilker Yildirim, and Joshua B Tenenbaum. Self-Supervised Intrinsic Image Decomposition. In *NeurIPS*, 2017. 2, 3

[17] Angjoo Kanazawa, Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018. 2, 3

[18] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018. 5, 11

[19] Nilesh Kulkarni, Abhinav Gupta, and Shubham Tulsiani. Canonical surface mapping via geometric cycle consistency. In *CVPR*, pages 2202–2211, 2019. 2

[20] Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep Convolutional Inverse Graphics Network. In *NeurIPS*, 2015. 3

[21] Pierre-Yves Laffont, Adrien Bousseau, Sylvain Paris, Frédo Durand, and George Drettakis. Coherent Intrinsic Images from Photo Collections. *ACM TOG*, 2012. 2

[22] Manuel Lagunas, Xin Sun, Jimei Yang, Ruben Villegas, Jianming Zhang, Zhixin Shu, Belen Masia, and Diego Gutierrez. Single-image full-body human relighting. In *Eurographics Symposium on Rendering (EGSR)*. The Eurographics Association, 2021. 3

[23] Edwin H. Land and John J. McCann. Lightness and retinex theory. *Journal of the Optical Society of America*, 61(1):1–11, 1971. 2

[24] 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, 2020. 2

[25] 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*, 2020. 2, 3

[26] Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Inverse Rendering for Complex Indoor Scenes: Shape, Spatially-Varying Lighting and SVBRDF from a Single Image. In *CVPR*, 2020. 3

[27] Zhengqi Li and Noah Snavely. Learning intrinsic image decomposition from watching the world. In *CVPR*, 2018. 2

[28] Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. Materials for Masses: SVBRDF Acquisition with a Single Mobile Phone Image. In *ECCV*, 2018. 3

[29] Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Learning to Reconstruct Shape and Spatially-Varying Reflectance from a Single Image. In *SIGGRAPH Asia*, 2018. 3, 6, 8

[30] 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. Springer, 2014. 5

[31] Andrew Liu, Shiry Ginosar, Tinghui Zhou, Alexei A. Efros, and Noah Snavely. Learning to Factorize and Relight a City. In *ECCV*, 2020. 2

[32] Guilin Liu, Duygu Ceylan, Ersin Yumer, Jimei Yang, and Jyh-Ming Lien. Material Editing Using a Physically Based Rendering Network. In *ICCV*, 2017. 3

[33] Yunfei Liu, Yu Li, Shaodi You, and Feng Lu. Unsupervised learning for intrinsic image decomposition from a single image. In *CVPR*, 2020. 2

[34] Wei-Chiu Ma, Hang Chu, Bolei Zhou, Raquel Urtasun, and Antonio Torralba. Single Image Intrinsic Decomposition without a Single Intrinsic Image. In *ECCV*, 2018. 3

[35] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *ICCV*, pages 2794–2802, 2017. 5

[36] Joel Ruben Antony Moniz, Christopher Beckham, Simon Rajotte, Sina Honari, and Christopher Pal. Unsupervised depth estimation, 3d face rotation and replacement. In *NeurIPS*, 2018. 2

[37] Thomas Nestmeyer, Jean-François Lalonde, Iain Matthews, and Andreas Lehrmann. Learning physics-guided face relighting under directional light. In *CVPR*, pages 5124–5133, 2020. 3

[38] Rohit Pandey, Sergio Orts-Escolano, Chloe LeGendre, Christian Haene, Sofien Bouaziz, Christoph Rhemann, Paul Debevec, and Sean Fanello. Total Relighting: Learning to Relight Portraits for Background Replacement. *ACM TOG*, 2021. 2, 3

[39] 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*, 2019. 11

[40] Bui Tuong Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975. 3

[41] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *ICCV*, pages 10901–10911, 2021. 2, 5

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015. 11, 15

[43] Carsten Rother, Martin Kiefel, Lumin Zhang, Bernhard Schölkopf, and Peter Gehler. Recovering intrinsic images with a global sparsity prior on reflectance. In *NeurIPS*, 2011. 2

[44] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *ICRA*, Shanghai, China, May 9-13 2011. 5

[45] Shen Sang and Manmohan Chandraker. Single-Shot Neural Relighting and SVBRDF Estimation. In *ECCV*, 2020. 3, 6, 8

[46] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 2

[47] Soumyadip Sengupta, Angjoo Kanazawa, Carlos D. Castillo, and David W. Jacobs. SfSNet: Learning Shape, Refectance and Illuminance of Faces in the Wild. In *CVPR*, 2018. 3

[48] Li Shen, Ping Tan, and Stephen Lin. Intrinsic image decomposition with non-local texture cues. In *CVPR*, 2008. 2

[49] Li Shen and Chuohao Yeo. Intrinsic images decomposition using a local and global sparse representation of reflectance. In *CVPR*, 2011. 2

[50] Li Shen, Chuohao Yeo, and Binh-Son Hua. Intrinsic image decomposition using a sparse representation of reflectance. *IEEE TPAMI*, 35(12):2904–2915, 2013. 4

[51] Jian Shi, Yue Dong, Hao Su, and Stella X. Yu. Learning Non-Lambertian Object Intrinsics across ShapeNet Categories. In *CVPR*, 2017. 2, 3, 6, 8

[52] Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *CVPR*, 2021. 3

[53] Tiancheng Sun, Jonathan T Barron, Yun-Ta Tsai, Zexiang Xu, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul E Debevec, and Ravi Ramamoorthi. Single image portrait relighting. *ACM TOG*, 38(4):79–1, 2019. 3

[54] Daichi Tajima, Yoshihiro Kanamori, and Yuki Endo. Relighting humans in the wild: Monocular full-body human relighting with domain adaptation. In *Computer Graphics Forum*, pages 205–216. Wiley Online Library, 2021. 3

[55] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 5

[56] Shangzhe Wu, Ameesh Makadia, Jiajun Wu, Noah Snavely, Richard Tucker, and Angjoo Kanazawa. De-rendering the world's revolutionary artefacts. In *CVPR*, 2021. 3

[57] Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised Learning of Probably Symmetric Deformable 3D Objects from Images in the Wild. In *CVPR*, 2020. 2, 3, 5

[58] Li Xu, Qiong Yan, Yang Xia, and Jiaya Jia. Structure extraction from texture via relative total variation. *ACM TOG*, 31(6):1–10, 2012. 4

[59] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance. In *NeurIPS*, 2020. 3

[60] Yufei Ye, Shubham Tulsiani, and Abhinav Gupta. Shelf-supervised mesh prediction in the wild. In *CVPR*, 2021. 3

[61] Ye Yu, Abhimetra Meka, Mohamed Elgharib, Hans-Peter Seidel, Christian Theobalt, and Will Smith. Self-supervised Outdoor Scene Relighting. In *ECCV*, 2020. 2

[62] Ye Yu and William AP Smith. InverseRenderNet: Learning Single Image Inverse Rendering. In *CVPR*, 2019. 2

[63] 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. 2

[64] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. NeRFactor: Neural Factorization of Shape and Reflectance Under an Unknown Illumination. In *SIGGRAPH Asia*, 2021. 3

[65] Hao Zhou, Sunil Hadap, Kalyan Sunkavalli, and David W Jacobs. Deep single-image portrait relighting. In *ICCV*, pages 7194–7202, 2019. 6, 7

6. Supplementary Material

6.1. Additional Results

We show some additional qualitaive results. Figure 9 shows a large number of additional decomposition results. Figure 10 shows a large number of additional visual comparisons with other state-of-the-art methods. Tab. 3 analyses the effects of bias in the training data, which might get introduced as a result of our coarse shape, light, and material estimation.

6.2. Limitations

While our proposed method achieves good results on a wide variety of data, it also comes with some limitations.

First, when there is no coarse light supervision, the training can become unstable and might, after a long training, suddenly diverge to a state where it ignores specular effects. This usually shows in α taking maximum value and the a_{spec} taking minimum value.

Secondly, even though our image formation model is already fairly complex, it can sometimes still not capture all possible lighting effects that can appear in in-the-wild images. In extreme cases, the model tends to bake those shading effects into either the albedo or the shading map, as shown for example in Fig. 11. Additionally, hard shadows can sometimes have an influence on the normal map, even though the shape underneath is of course unaffected by the shadow of another object falling onto it (see second row of Fig. 11). We expect that an explicit modeling of shadows could alleviate this problem.

6.3. Broader Impact & Ethics

The goal of our method is to learn a model that lifts images of objects from 2D into a disentangled 3D representation. We expect this method to be useful for the increasing amount of XR and VFX application in consumer devices. Additionally, learning to de-render images is in itself a challenging computer vision task as the model needs to discover robust and general visual representations.

Alongside the paper, we are releasing a synthetic dataset of 3D objects for future benchmarking under an open, non-commerical license. We expect this to contribute to the field as it establishes a test set with ground truth annotations that was previously not available to researchers.

As the main focus of this work concerns general objects, potential negative societal implications are low. However, we include an evaluation on CelebA-HQ [18], which is a dataset of images of faces of celebrities. The copyright situation is unclear as the dataset was scraped from the internet and naturally contains person identifying information (faces). The dataset contains biases in many forms (celebrities are not representative of the general population, most images are professional photographs, general focus on celebrities from western countries etc.). Models trained on this data can reflect these biases and thus should only be used for research purposes. We include these results to show a comparison to methods that are specialized for human faces while our method can be trained on an object-centric dataset.

As described in the previous section, our method still has some limitations and thus should not be used in critical applications.

	Normal N		Albedo A		Specular I_{spec}	
Add. bias	**MSE** ↓	**DIA** ↓	**SIE** ↓	**SSIM** ↑	**MSE** ↓	**L1** ↓
None	0.173	37.8	0.075	0.760	0.112	0.059
Normal $+15°$	0.184	39.4	0.077	0.759	0.124	0.077
Brightn. -0.1	0.171	37.7	0.075	0.766	0.115	0.059
Brightn. $+0.1$	0.183	39.1	0.077	0.716	0.122	0.065

Table 3. **Effects of bias in training data** Artificial bias introduced for normal angles and the brightness of the coarse albedo estimates.

6.4. Technical Details

3D Geometry. The coordinate system used for representing the normal maps has its x-axis pointing left, y-axis pointing upwards, and z-axis pointing towards the viewer. We visualize normal maps by first scaling and translating the value range from $[-1, -1]$ to $[0, 1]$ and then directly converting the xyz coordinates to rgb colors.

The lighting direction is modeled by a vector pointing from the model to the light source. Note, that the lighting is modeled to be relative to the camera position Therefore, the viewing direction, relevant for specular effects, is always constant at $v = (0, 0, 1)^T$.

To obtain the normal map N_{D} from the depth map D, which is used in Eq. 1, we compute the 3D tangent plane of a point using the 4-neighborhood of this point in image space. The normal of the tangent plant is then used as the normal at this point.

The coarse normal map N_c is the a key component in the optimization to obtain the coarse albedo and light estimate. It is usually either given at a lower resolution than the input image, or it is sparse. In the case of lower resolution, we downsample the input image to match the resolution of the coarse normal map. When computing the coarse losses, we upsample the coarse normal and albedo map again.

In the case that N_c is sparse, we fill it in using nearest neighbor interpolation before computing the initial albedo estimate $\tilde{A}_c$. However, during albedo and light optimization, as well as during the computation of the coarse losses, we only consider valid pixels.

Light Prediction and Sampling. When predicting the light direction, we fix the $z = 1$ (facing from the camera side to the object) and only predict the $x, y \in [-1, 1]$ components. Afterwards, we normalize the vector. The reasoning behind this is that there only exist meaningful lighting effects, when the light is not coming from behind the object. Further, we apply a custom scaling function $f(x) = (\frac{x+1}{2}(\alpha_{\max} - 1) + 1)^2$ to obtain α. We also limit the range of a_{spec} to $[a_{\text{spec,min}}, a_{\text{spec,max}}]$

The adversarial loss requires images that are relit under random lighting conditions. To also cover unusual light directions $l' = (x', y', z')$, we again fix $z' = 1$, sample $x', y' \sim \mathcal{N}(0, \sigma_l)$ and then normalize l'. As strength of ambient and directional lighting only really influences the overall brightness of the image, which is a very easy cue to pick up on for the discriminator, they have to remain within the distribution of the training data. Therefore, we compute the mean μ_{amb} of s_{amb} over the training batch and then sample $s'_{\text{amb}} \sim \mathcal{N}(\mu_{\text{amb}}, 0.1)$ (same for s'_{dir}).

Implementation. All models and data processing steps (except for the Point Cloud Library) are implemented in PyTorch [39]. The Image-to-image networks (shape, albedo, specular refinement) are implemented as auto-encoders with skip-connections, inspired by the U-Net [42]. The light network follows a classical encoder architecture. All of them use the tanh activation function and the respective outputs gets scaled to the corresponding value range (*e.g.* to $[0, 1]$ for colors). Sec. 6.4 and Sec. 6.4 give an overview over the different network configurations.

Training. For all training runs, we use a batch size of 12 on a GPU with 24GB VRAM. We train the **CelebA-HQ** model for 30 epochs (approx. 60k iterations) and the **Co3D** model for 10 epochs (approx. 20k iterations). We only use specular refinement for the CelebA-HQ model and train it in a second training stage for 5 more epochs. Here, we freeze all other network weights and only use the adversarial loss with a weight of $\lambda_{\text{gan}} = 0.1$. Tab. 4 shows the exact hyperparameters used for training. On Co3D, we use the objects masks obtained from SfM to mask the prediction during training and inference.

COSy Dataset. The **COSy** dataset is built from ten publicly available 3D scenes for the Blender 3D modeling software[2]. The 3D models can be downloaded from the following links and are available under variants of the Creative Commons license. Hot Dog[3], Accordion[4], Wall-Phone[5],

[2] https://www.blender.org/
[3] https://blendswap.com/blend/23962
[4] https://blendswap.com/blend/17099
[5] https://blendswap.com/blend/19579

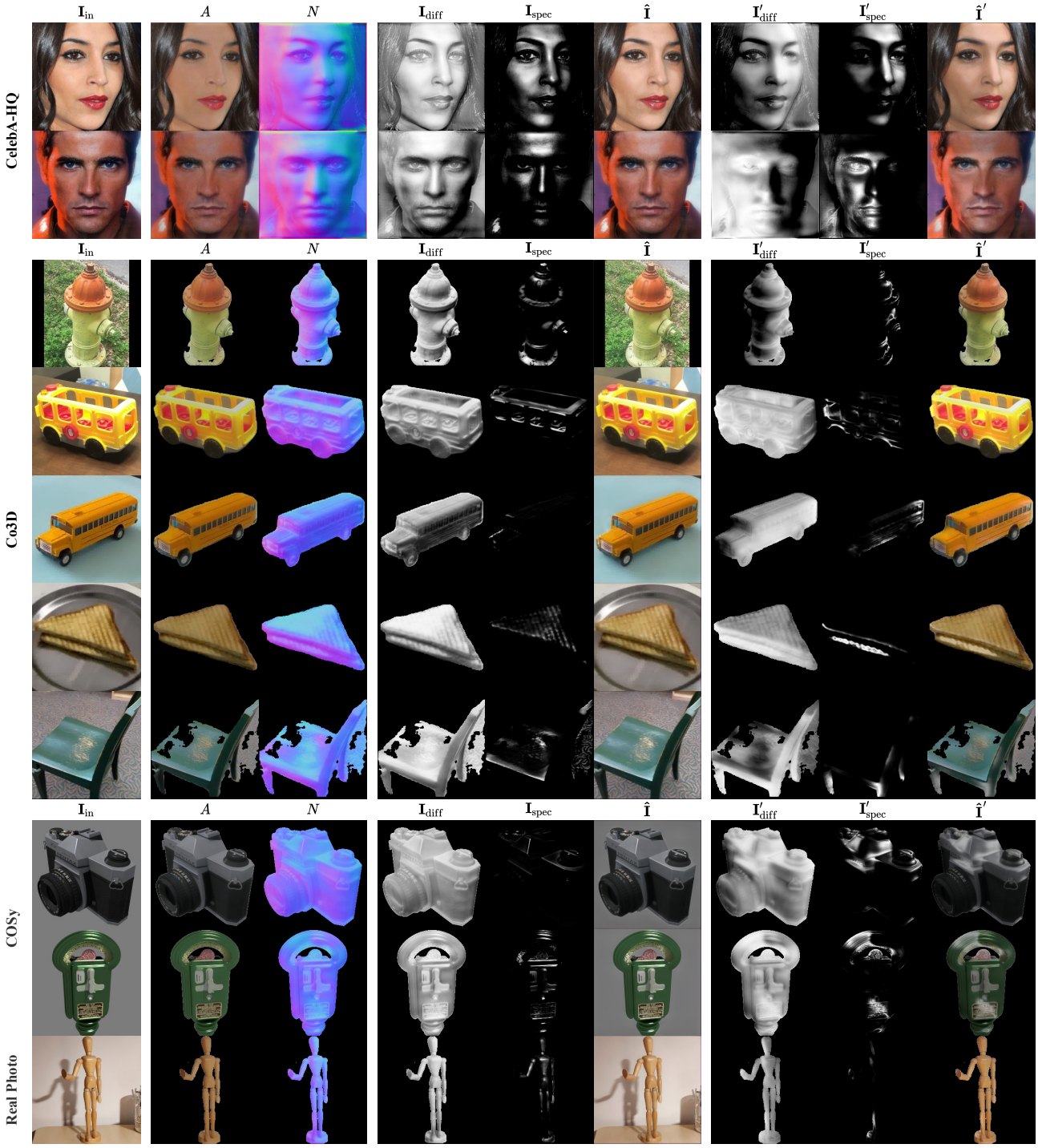


Figure 9. **Additional qualitative results.** Corresponding to Fig. 4 in the main paper.

Hydrant[6], Wingback Chair[7], Camera[8], Toaster[9], Scooter[10], Microphone[11], Parkingmeter[12]. We adapt the models to fit our requirements, for example,

[6] https://blendswap.com/blend/8443
[7] https://blendswap.com/blend/12555
[8] https://blendswap.com/blend/15833
[9] https://blendswap.com/blend/6231
[10] https://blendswap.com/blend/5256
[11] https://blendswap.com/blend/4145
[12] https://blendswap.com/blend/7714

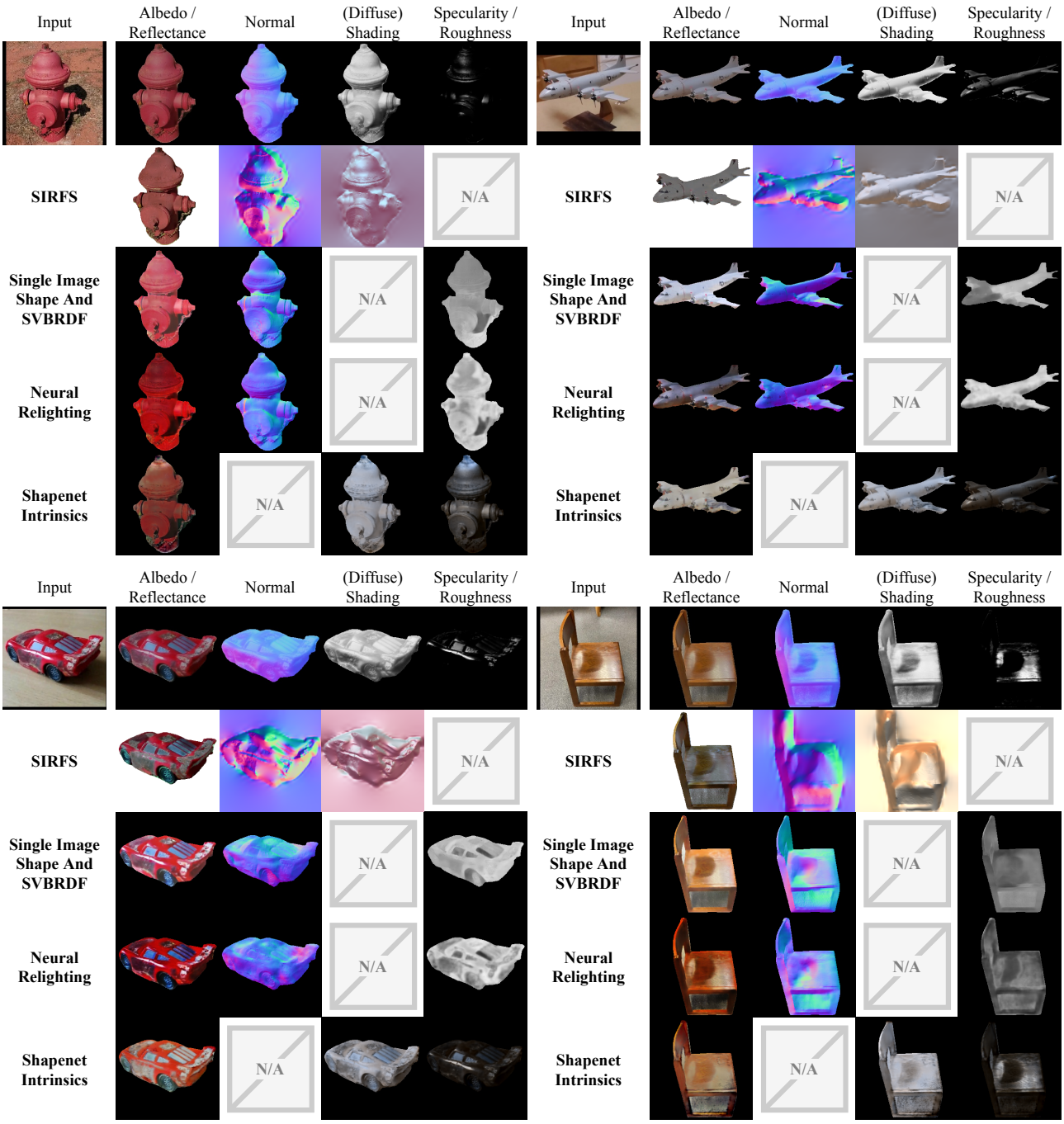


Figure 10. **Additional qualitative comparison with state of the art.** Corresponding to Fig. 6 in the main paper.

define camera views and modify the material so that we can extract the diffuse albedo.

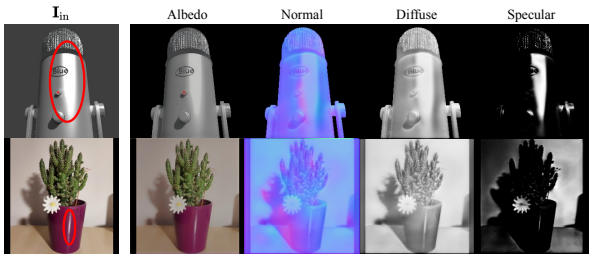


Figure 11. **Limitations.** Extreme lighting effects with out-of-distribution examples. Strong specular reflections can sometimes bleed into the albedo map. Strong shadows can have an effect on the normal map.

Model Configuration		
Param.	CelebA-HQ	CO3D
D	$[0.9, 1.1]$	$[0.9, 1.1]$
a_{spec}	$[0.0, 0.5]$	$[0.1, 0.5]$
α_{max}	64	64

Training Configuration			
Param.	Value	Param.	Value
n	12	λ_A	1
η	$1e^{-4}$	λ_L	$1_{\text{(CelebA-HQ)}}$ / $0_{\text{(Co3D)}}$
λ_D	0.5	λ_{rec}	0.5
λ_N	1	λ_{gan}	0.01

Optimization Configuration			
CelebA-HQ		**CO3D**	
Param.	Value	Param.	Value
i	100	i	100
η_{light}	0.01	η_{light}	0.01
η_{albedo}	0.04	η_{albedo}	0.01
λ_{TV}	5	λ_{TV}	20

Table 4. **Model Configuration and Hyperparameters.** n denotes batch size. η denotes the learning rate. i denotes the number of iterations for optimization.

Table 5. **Encoder Architecture.** Architecture of the shape f_S and pose network f_P. The network follows a convolutional encoder structure. n is the number of parameters predicted by each network.

Encoder	Output size
Conv(3, 64, 4, 2) + ReLU()	128×128
Conv(64, 128, 4, 2) + ReLU()	64×64
Conv(128, 256, 4, 2) + ReLU()	32×32
Conv(256, 512, 4, 2) + ReLU()	16×16
Conv(512, 512, 4, 2) + ReLU()	8×8
Conv(512, 512, 4, 2) + ReLU()	4×4
Conv(512, 512, 4, 2) + ReLU()	2×2
Conv(512, 512, 4, 1) + ReLU	1×1
Conv(512, c_{out}, 1, 1) $\rightarrow$ output	1×1

Table 6. **Auto-Encoder Architecture.** Architecture of Φ_{shape}, Φ_{albedo}, and $\Phi_{\text{spec_ref}}$. The network follows a U-Net structure [42]. All convolution operators zero-pad the input such that the output has the same resolution.

Encoder	Output size
Conv(3, 64, 3, 1) + LReLU(0.1) + Conv(64, 64, 3, 1) + LReLU(0.1) + MaxPool(2)	128 × 128
Conv(64, 128, 3, 1) + LReLU(0.1) + Conv(128, 128, 3, 1) + LReLU(0.1) + MaxPool(2)	64 × 64
Conv(128, 256, 3, 1) + LReLU(0.1) + Conv(256, 256, 3, 1) + LReLU(0.1) + MaxPool(2)	32 × 32
Conv(256, 512, 3, 1) + LReLU(0.1) + Conv(512, 512, 3, 1) + LReLU(0.1) + MaxPool(2)	16 × 16
Conv(512, 1024, 3, 1) + LReLU(0.1) + Conv(1024, 1024, 3, 1) + LReLU(0.1) + MaxPool(2)	8 × 8
Conv(1024, 1024, 3, 1) + LReLU(0.1) + Conv(1024, 1024, 3, 1) + LReLU(0.1) + MaxPool(2)	4 × 4
Conv(1024, 1024, 3, 1) + LReLU(0.1) + Conv(1024, 1024, 3, 1) + LReLU(0.1) + MaxPool(2)	2 × 2
Conv(1024, 1024, 3, 1) + LReLU(0.1) + Conv(1024, 1024, 3, 1) + LReLU(0.1)	2 × 2
Decoder	**Output size**
Conv(1024, 1024, 3, 1) + LReLU(0.1) + Conv(1024, 1024, 3, 1) + LReLU(0.1) + Upsample(2)	4 × 4
Conv(1024, 1024, 3, 1) + LReLU(0.1) + Conv(1024, 1024, 3, 1) + LReLU(0.1) + Upsample(2)	8 × 8
Conv(1024, 1024, 3, 1) + LReLU(0.1) + Conv(1024, 1024, 3, 1) + LReLU(0.1) + Upsample(2)	16 × 16
Conv(1024, 512, 3, 1) + LReLU(0.1) + Conv(512, 512, 3, 1) + LReLU(0.1) + Upsample(2)	32 × 32
Conv(512, 256, 3, 1) + LReLU(0.1) + Conv(256, 256, 3, 1) + LReLU(0.1) + Upsample(2)	64 × 64
Conv(256, 128, 3, 1) + LReLU(0.1) + Conv(128, 128, 3, 1) + LReLU(0.1) + Upsample(2)	128 × 128
Conv(128, 64, 3, 1) + LReLU(0.1) + Conv(64, 64, 3, 1) + LReLU(0.1) + Upsample(2)	256 × 256
Conv(64, c_{out}, 3, 1) + LReLU(0.1) + Conv(c_{out}, c_{out}, 3, 1)	256 × 256

6

DOVE: Learning Deformable 3D Objects by Watching Videos

This work is currently under review and this preprint is public on arXiv.

DOVE: Learning Deformable 3D Objects by Watching Videos

Shangzhe Wu[1*†], Tomas Jakab[1*†], Christian Rupprecht[1] and Andrea Vedaldi[1]

[1]Visual Geometry Group, University of Oxford.

*Corresponding author(s). E-mail(s): szwu@robots.ox.ac.uk; tomj@robots.ox.ac.uk;
Contributing authors: chrisr@robots.ox.ac.uk; vedaldi@robots.ox.ac.uk;
†These authors contributed equally to this work.

Abstract

Learning deformable 3D objects from 2D images is often an ill-posed problem. Existing methods rely on explicit supervision to establish multi-view correspondences, such as template shape models and keypoint annotations, which restricts their applicability on objects "in the wild". A more natural way of establishing correspondences is by watching videos of objects moving around. In this paper, we present DOVE, a method that learns textured 3D models of deformable object categories from monocular videos available online, without keypoint, viewpoint or template shape supervision. By resolving symmetry-induced pose ambiguities and leveraging temporal correspondences in videos, the model automatically learns to factor out 3D shape, articulated pose and texture from each individual RGB frame, and is ready for single-image inference at test time. In the experiments, we show that existing methods fail to learn sensible 3D shapes without additional keypoint or template supervision, whereas our method produces temporally consistent 3D models, which can be animated and rendered from arbitrary viewpoints. Project page: https://dove3d.github.io/.

Keywords: Deformable 3D Objects, Unsupervised 3D Learning

1 Introduction

In applications, we often need to obtain accurate 3D models of deformable objects from just one or a few pictures of them. This is the case in traditional applications such as robotics, but also, increasingly, in consumer applications, such as content creation for virtual and augmented reality — using everyday pictures and videos taken with a cellphone.

3D reconstruction from a single image, or even a small number of views, is generally very ambiguous and only solvable by leveraging powerful statistical priors of the 3D world. Learning such priors is however very challenging. One approach is to use training data specifically collected for this purpose, for example by using 3D scanners and domes [1–10] or shape models [11–15]. This is expensive and can be justified only for a few categories such as human bodies and faces that are of particular significance in applications. However, scanning is not a viable approach to cover the huge diversity of objects that exist in the real world.

We thus need to develop methods that can learn 3D deformable objects from as cheap supervision as possible, such as leveraging casually-collected images and videos found on the Internet, or crowd-sourced datasets such as CO3D [16]. Ideally, our system should take as input a collection of such casual images and videos and learn a model capable of reconstructing the 3D shape, appearance and deformation of a new object from a single image of it.

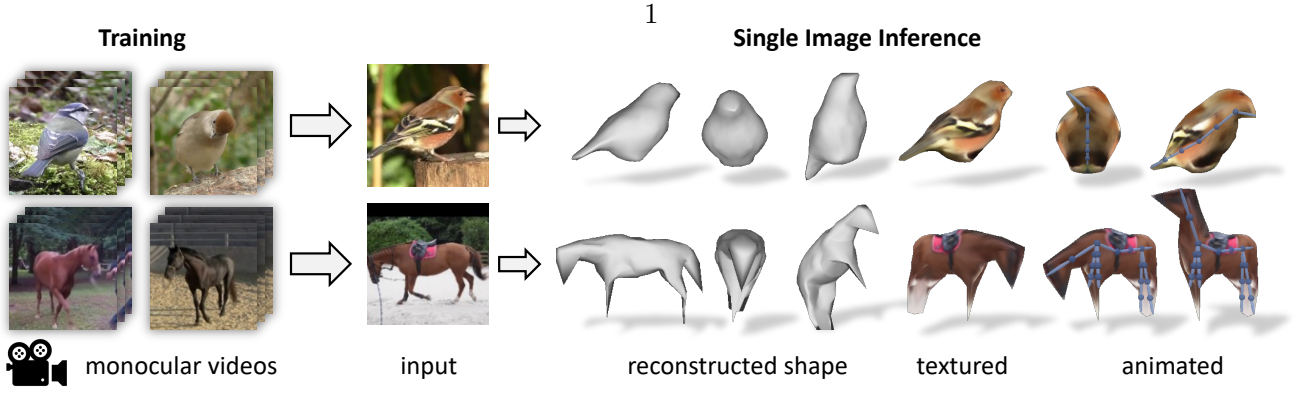


Fig. 1: **DOVE - Deformable Objects from VidEos.** Given a collection of video clips of an object category as training data, we learn a model that is able to predict a textured, articulated 3D mesh of the object from a single input image.

While several authors have looked at this problem before [1, 2, 4–6, 10], so far it has always been necessary to make additional simplifying assumptions compared to the ideal unsupervised setting described above. These assumptions usually come in the form of additional geometric supervision. The most common one is to require 2D masks for the objects, either obtained manually or via a pre-trained segmentation network such as [17, 18]. On top of this, there is usually at least one more *additional* form of geometric supervision, such as providing an initial approximate 3D template of the object, 2D keypoint detections, or approximate 3D camera parameters [15, 19–25]. There is a small number of works that require no masks or geometric supervision [26], but they come with other limitations such as relying on limited viewpoint range.

Our aim in this paper is to learn 3D deformable objects from complex *casual videos* while only using 2D masks and removing the additional geometric supervision that require expensive manual annotations used in prior works (keypoints, viewpoint, and templates). In order to compensate for this lack of geometric information, we propose to learn from casual videos rather than still images, unlike most prior works. While this adds some complexity to the method, using videos has the key advantage to allow one to establish correspondences between different images, for instance by using an off-the-shelf optical flow algorithm. While this information is weaker than externally-provided information such as keypoints, nevertheless it is very helpful in recovering the objects' viewpoint. Note, though, that videos are only used for supervision: our goal is still to learn a model that can reconstruct a new object instance from a single image.

In order to use videos effectively, we make a number of technical contributions. The first one addresses the challenge of estimating the viewpoint of the 3D objects. Prior works addressed this issue by sampling a large number of possible views [23, 27], an approach that [23] calls a *camera multiplex*. We find that this is unnecessary. While viewpoint estimation is ambiguous, we show that the ambiguity is mostly restricted to a small space of symmetries induced by the 2D projection of the 3D objects onto the image. The result is that, as the model is learned, only a very small number of alternative viewpoints need to be explored in order to escape form the ambiguity-induced local optima: from, e.g., 40 in [23] to just two per iteration, which largely reduces memory and time requirements for training.

Our second contribution is the design of the object model. We propose a hierarchical shape representation that explicitly disentangles category-dependent prior shape, instance-dependent deformation, as well as time-dependent articulated and rigid pose. In this way, we automatically factor shape and pose variations at different levels in the video dataset, and leverage instance-specific correspondences within a video and instance-agnostic correspondences across multiple videos. We also enforce a bilateral symmetry on the predicted

canonical shape and texture, similar to previous methods [19, 23, 26, 28]. However, differently from these approaches, which assume symmetry at the level of the object instances, here we assume the canonical (pose free) shapes are symmetric, but individual articulations can be asymmetric [29, 30], which is much more realistic.

We also address the challenge of evaluating these reconstruction methods. Prior works in this area generally lack data with 3D ground truth. Instead, they resort to indirect evaluation by measuring the quality of the 2D correspondences that are established by the 3D models. To address this problem, we create a dataset of views of real-life animal models (toy birds). The data is designed to resemble a subset of the images as found in existing datasets such as CUB [31]; however, it additionally comes with 3D scans of the objects, which can be used to test the quality of the 3D reconstructions directly. We use this data to evaluate our and several state-of-the-art algorithms without the need for proxy metrics such as keypoint re-projection error that are insufficient to assess the quality of a 3D reconstruction. We hope that this data will be useful for future work in this area.

Overall, our method can successfully learn good 3D shape predictors from videos of animals such as birds and horses. Compared to prior work, our method produces better 3D shape reconstructions, as measured on the new benchmark, when not using additional geometric supervision.

2 Related Work

We divide the vast literature of related work into two parts. The first one focuses on learning based approaches for 3D reconstruction with limited supervision. The second parts highlights related work for 3D reconstruction from images and video.

2.1 Unsupervised and Weakly-supervised 3D Reconstruction

A primary motivation for introducing machine learning in 3D reconstruction is to enable reconstruction from single views, which necessitates learning suitable shape priors. In particular, we focus the discussion on unsupervised and weakly-supervised methods that do not require explicit 3D ground-truth for training. Early unsupervised work include monocular depth predictors trained from egocentric videos of rigid scenes [38, 39].

Others have explored weakly-supervised methods for learning full 3D meshes of object categories [4, 6, 19–21, 23, 24, 28, 40–43]. Many of these methods learn from still images and generally require masks and other additional supervision or prior assumptions, summarized in Table 1. In particular, **CMR** [19] uses 2D keypoint annotations (in addition to masks) to initialize shape and viewpoint using Structure-from-Motion (SfM). This is extended in the follow-up works in various ways. **U-CMR** [23], **TTP** [24] and **IMR** [37] replace the keypoint annotations with a category-specific template shape beforehand. With the template shape, extensive viewpoint sampling (camera multiplex) can be done to search for the best camera viewpoint for each training image [23]. **UMR** [28] instead uses part segmentations from SCOPS [33], which also requires supervised ImageNet pretraining. **VMR** [20] extends CMR with asymmetric deformation, and introduces a test-time adaptation procedure on individual videos by enforcing temporal consistency on the predictions produced by a pre-trained CMR model. Note that we use videos to learn a 3D shape model *from scratch*, whereas VMR starts with a pre-trained model and only performs online adaptation on videos. **CSM** [22] and **articulated CSM** [27] learn to pose an externally-provided (articulated) 3D template of an object category to images. Unsup3D [26] learns symmetric objects, like faces, without masks, but only with limited viewpoint variation.

Adversarial learning has also been explored to replace the need of multi-views for training [44–54]. The idea is to use a discriminator network to tell whether or not arbitrarily generated views of the learned 3D model are plausible, which provides signals to learn the geometry. Although this approach does not require viewpoint annotations for individual images, it does rely on a reasonable approximation of the viewpoint distribution in the training data, from which random views are generated. Overall, promising results can be achieved on synthetic data as well as a few real object categories, but general methods usually recover only coarse 3D shapes or 3D feature volumes that are difficult to extract.

Table 1: Related Work Overview. Annotations: template shape, viewpoint, 2D keypoint, object mask, optical flow, video, *optimizes a single object instance over a single or a few sequences, [1]shape bases initialized from CMR [19], [2]outputs texture flow, [3]obtained from DensePose [32], [4]obtained from keypoints using SfM, †UMR [28] relies on part segmentations from SCOPS [33].

	Supervision						Output				
Method							3D	2.5D	Motion	Viewpoint	Texture
VMR* [20]	(✓)[1]			✓		✓	✓		✓	✓	(✓)[2]
LASR* [34]				✓	✓	✓	✓		✓	✓	✓
ViSER* [35]				✓	✓	✓	✓		✓	✓	✓
BANMo* [36]		(✓)[3]	(✓)[3]	✓	✓	✓	✓		✓	✓	✓
Unsup3D [26]								✓		✓	✓
CSM [22]	✓			✓						✓	
CMR [19]	(✓)[4]	(✓)[4]	✓	✓			✓			✓	(✓)[2]
U-CMR [23]	✓			✓			✓			✓	✓
IMR [37]	✓			✓			✓			✓	(✓)[2]
TTP [24]	✓			✓			✓			✓	(✓)[2]
UMR† [28]				✓			✓			✓	(✓)[2]
VMR [20]	(✓)[1]	(✓)[4]	✓	✓			✓		✓	✓	(✓)[2]
Ours				✓	✓	✓	✓		✓	✓	✓

2.2 Reconstruction from Multiple Views and Videos

Most works using multiple views and videos focus on reconstructing individual instances of an object. Classic SfM methods [55, 56] use multiple views of a rigid scene, with pipelines such as KinectFusion [57] and DynamicFusion [58] integrating depth sensors for reconstructing dense static and deformable surfaces. Neural implicit surface representations have recently emerged for multi-view reconstruction [59–61]. NeRF [62] and its deformable extensions [63–68] synthesize novel views from densely sampled multi-views of a static or mildly dynamic scene using a Neural Radiance Field, from which explicit coarse 3D geometry can be further extracted. A more recent line of work, such as LASR [34] and ViSER [35], optimizes a single 3D deformable model on an individual video sequence, using mask and optical flow supervision. BANMo [36] further extends the pipeline and optimizes over a few video sequences of the same object instance, with the help of a pretrained DensePose [32] model. However, these optimization-based models are typically sensitive to the quality of the sequences and tends to fail when only limited views are observed (see Fig. 6). In contrast, by learning priors over a video dataset, our model can perform inference on a single image.

Other works that learn 3D categories form videos typically require some shape prior, such as a parametric shape model [11, 69], and hence mostly focus on reconstruction of human bodies or faces [15, 21, 70–76]. [77] and [78] consider turntable like videos to learn to reconstruct rigid object categories. In contrast, our method learns a 3D shape model of a deformable object category *from scratch* using videos.

3 Method

Our goal is to learn a function $(V, \xi, T) = f(I)$ that, given a *single image* $I \in \mathbb{R}^{3\times H\times W}$ of an object, predicts its 3D shape V (a mesh), its pose ξ (either a rigid transformation or full articulation parameters) and its texture T (an image). We describe below the key ideas in our method and refer the reader to the sup. mat. for details.

While the predictor f is monocular, we supervise it by making use of video sequences $\mathcal{I} = \{I_t\}_{t=1,\dots,|\mathcal{I}|}$, where t denotes time. For this, we use a *photo-geometric auto-encoding approach*. Let $M \in \{0,1\}^{H\times W}$ be the 2D mask of the object in image I, which we assume to be given. The model $(V, \xi, T) = f(I)$ encodes the image as a set of

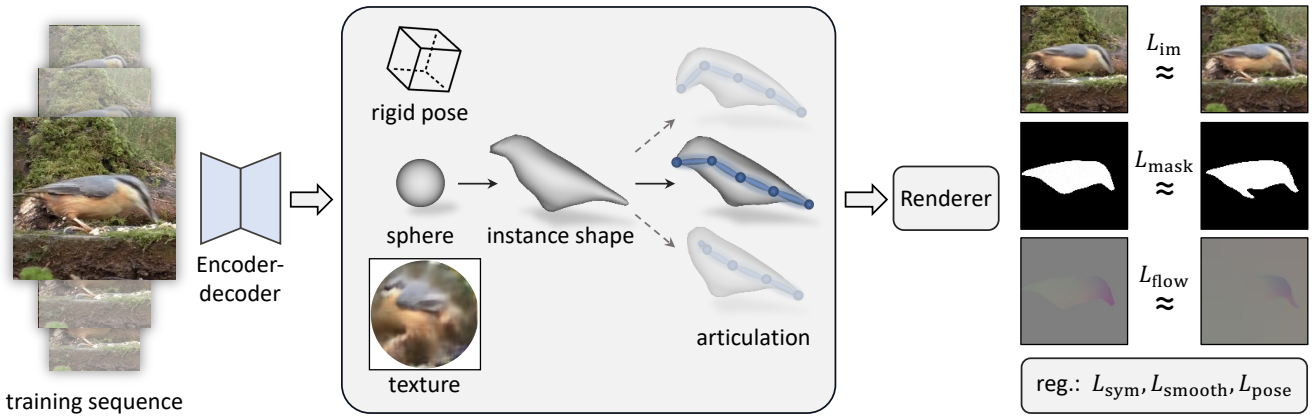


Fig. 2: **Training Pipeline.** From a single frame of a video, we predict the 3D pose, shape and texture of the object. The shape is further disentangled into category shape, instance shape and deformation using linear blend skinning. Using a differentiable rendering step, we can train the model end-to-end by reconstructing the image and by enforcing temporal consistencies.

photo-geometric parameters; from these, an hand-crafted rendering function $(\hat{I}, \hat{M}) = \mathcal{R}(V, \xi, T)$ reconstructs the image $\hat{I}$ and the mask $\hat{M}$. For supervision, the rendered image and the rendered mask is compared to the given ones via two losses:

$$L_{\text{im}} = \lambda_{\text{im}} \|\hat{M} \odot (\hat{I} - I)\|_1, \tag{1}$$

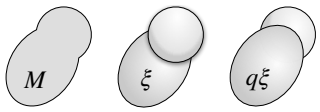
$$L_{\text{mask}} = \lambda_{\text{mask}} \|\hat{M} - M\|_2^2, \tag{2}$$

where λ_{im} and λ_{mask} weigh each loss. Note that the image loss is restricted to the predicted region as the model only represents the object but not the background. Figure 2 gives an overview of the training pipeline.

3.1 Solving the Viewpoint Ambiguity

Decomposing a single image into shape, pose and appearance is ambiguous, which is a major challenge that can easily result in poor reconstructions. Some prior works have addressed this issue by sampling a large number of viewpoints during training, thus giving the optimizer a chance to avoid local optima. However, this is a slow process that requires testing a large number of hypotheses at every iteration (*e.g.* 40 in [23]) and requires a precise template shape to understand the differences between small viewpoint changes.

Here we note that this is likely unnecessary. The key observation is that the ambiguities arising from image-based reconstruction are not arbitrary; instead, they tend to concentrate around specific symmetries induced by the projection of a 3D object onto an image.



The image to the right illustrates this idea. Here, given only the mask M, one is unable to choose between the object pose ξ or its mirrored variant $q\xi$, where q is a suitable 'mirror mapping' that rotates the pose back to front (see sup. mat. for details). We argue that, before developing a more nuanced understanding of appearance, the model f is similarly undecided about the pose of the 3D object; however, the number of choices is very limited: either the current prediction $\xi = f_\xi(I)$ is correct, or its mirrored version $q\xi$ is.

Concretely, during training we evaluate the loss $L(V, \xi, T)$ for the model prediction and the loss $L(V, q\xi, T)$ for the mirrored pose. We find the better of the two poses $\xi^* = \arg\min_{\hat{\xi} \in \{\xi, q\xi\}} L(V, \hat{\xi}, T)$ and optimize the loss:

$$L_{\text{pose}} = \lambda_{\text{pose}} \|\xi - \xi^*\|_2^2. \tag{3}$$

In this way, the model is encouraged to flip its prediction when $L(V, q\xi, T) < L(V, \xi, T)$. This assures that the model eventually learns the correct pose and does not rely on the flipping towards the end of the training.

3.2 Learning from Videos

We exploit the information in videos by noting that the shape V and texture T of an object are invariant over time, with any time-dependent change limited to the pose ξ. Hence, given a sequence of images $\mathcal{I} = \{I_t\}_{t=0,\dots,|\mathcal{I}|}$ of the same object and corresponding frame-based predictions $(V_t, \xi_t, T_t) = \Phi(I_t)$, we feed the rendering function $(\hat{I}_t, \hat{M}_t) = R(\bar{V}, \xi_t, \bar{T})$ with the shape and texture averages $\bar{V} = \frac{1}{|\mathcal{I}|}\sum_{t=1}^{|\mathcal{I}|} V_t$ and $\bar{T} = \frac{1}{|\mathcal{I}|}\sum_{t=1}^{|\mathcal{I}|} T_t$. The idea is that, unless shape and texture agree across predictions, their averages would be blurry and result in poor renderings. Hence, minimizing the rendering loss indirectly encourages these quantities to be consistent over time.

Furthermore, while the pose ξ_t does vary over time, pose changes must be compatible with image-level correspondences. Specifically, let $F_t \in \mathbb{R}^{H\times W\times 2}$ be the optical flow measured between frames I_t and I_{t+1} by an off-the-shelf method such as RAFT [79]. We can render the flow $\hat{F}_t = \mathcal{R}(V, \xi_t, \xi_{t+1})$ by computing the displacement of the object vertices V as a pose change from ξ_t to ξ_{t+1}. We can then add the flow reconstruction loss

$$L_{\text{flow}}(\hat{F}_t, F_t) = \lambda_{\text{flow}} \|M_t \odot (\hat{F}_t - F_t)\|_2^2, \qquad (4)$$

to encourage consistent motion of the object. Its influence is controlled by the weight λ_{flow}.

3.3 Hierarchical Shape Model

Next, we flesh out the shape model. The shape $V \in \mathbb{R}^{3\times K}$ is given by K mesh vertices and represents the shape of a specific object instance in a *canonical* pose. It is obtained by the predictor $f_V(I) = V_{\text{base}} + \Delta V_{\text{tmpl}} + \Delta V(I)$ where: V_{base} is an initial fixed shape (a sphere), ΔV_{tmpl} is a learnable matrix (initialized as zeros) such that $V_{\text{tmpl}} = V_{\text{base}} + \Delta V_{\text{tmpl}}$ gives an average shape for the category (template), and $\Delta V(I)$ is a neural network further deforming the this template into the specific shape of the object seen in image I. We further restrict V, which is the rest pose, to be bilaterally symmetric by only predicting half of the vertices and obtaining the remaining half via mirroring along the x axis. Note that, while in many prior works the category-level template V_{tmpl} is given to the algorithm, here this is learned automatically from a sphere.

Finally, the shape V is transformed into the actual mesh observed in the image by a *posing function* $g(V, \xi)$. We work with two kinds of such functions. The first one is a simple rigid motion $g(V, \xi) = g_\xi V$, $g_\xi \in SE(3)$. This is used in an initial warm-up phase for the model to allow it to learn a first version of the template V automatically.

In a second learning phase, we further enrich the model to capture complex articulations of the shape. There are a number of possible parameterizations that could be used for this purpose. For instance, [21] automatically initializes a set of keypoints via spectral analysis of the mesh. Here, we initialize instead a traditional skinning model, given by a system of bones $b \in \{1, \dots, B\}$, ensuring inelastic deformations.

The skinning model is specified by: the bone topology (a tree), the joint location $\mathbf{J}_b \in \mathbb{R}^3$ of each bone with respect to the parent bone, the relative rotation $\xi_b \in SO(3)$ of that bone with respect to the parent, and a row-stochastic matrix of weights $w \in [0, 1]^{K\times B}$ specifying the strength of association of each mesh vertex to each bone. Of these, only the topology is chosen manually (e.g, to account for a different number of legs for objects in the category). The joint locations $\mathbf{J}_b$ and the skinning weights w are set automatically based on a simple heuristic (described in sup. mat.).

While topology, $\mathbf{J}_b$ and w are fixed, the joint rotation $\xi_b \in SO(3), b = 2, \dots, B$ and the rigid pose $\xi_1 \in SE(3)$ are output by the predictor f to express the deformation of the object as it changes from image to image.

3.4 Appearance Model and Rendering

We model the appearance of the object using a texture map $T \in \mathbb{R}^{3\times H_T\times W_T}$. The vertices of the base mesh V_{base} are assigned to fixed texture uv-coordinates and the texture inherits the symmetry of the base mesh. Given the posed mesh $g(V, \xi)$ and the texture T, we *render* an image $(\hat{I}, \hat{M}) = \mathcal{R}(V, \xi, T)$ of the object using standard perspective-correct texture mapping with barycentric coordinates using the PyTorch3D differentiable mesh renderer [80].

3.5 Symmetry and Geometric Regularizers

An important property of object categories is that they are often symmetric. This does not mean that individual object instances are symmetric, but that the space of objects is [29]. In other words, if image I contains a valid object, so does the mirrored image mI. Furthermore, given the photo-geometric parameters $(V, \xi, T) = f(I)$ for I, the parameters for mI must be given by $f(mI) = (mV, m\xi, Tm)$ where $mV = V$ (because the rest shape is assumed symmetric), Tm is the flipped texture image and $m\xi$ is a mirrored version of the pose. Hence, we additionally enforce the pose predictor to satisfy this structure by minimizing the loss $L_{\text{sym}} = \lambda_{\text{sym}} \| f_\xi(mI) - m(f_\xi(I)) \|_2^2$, weighted by λ_{sym}.

Note the relationship between the mirroring operators $q\xi$ in Section 3.1 and $m\xi$ here: they are the same, up to a further rigid body rotation. The effect is that q appears to rotate the object back to front, and m left to right. This is developed formally in the sup. mat.

We further regularise learning the mesh V via a loss $L_{\text{smooth}}(V, V_{\text{tmpl}})$ which includes: the ARAP loss [81] between V and the template V_{tmlp}, ensuring that they do not diverge too much, and a Laplacian and mesh normal smoothers for V.

3.6 Learning Formulation

Given a video $\mathcal{I} = \{I_t\}_{t=1,\dots,|\mathcal{I}|}$, the overall learning loss is thus:

$$L = L_{\text{im}} + L_{\text{mask}} + L_{\text{flow}} + L_{\text{sym}} + L_{\text{smooth}} + L_{\text{pose}}.$$

In practice, we found it important to warm up the model, activating increasingly more refined model components as training progresses. This can be seen as a sort of coarse-to-fine or paced learning strategy.

Learning thus uses the following schedule in three phases: (1) *shape learning*: the basic model with no instance-specific deformation (i.e., $V = V_{\text{tmpl}}$), no bone articulation and only the mask loss is optimized in order to obtain an initial template V_{tmpl}; (2) *ambiguity resolution*: the pose rectification loss L_{pose} is activated to resolve the front-to-back ambiguity of Section 3.1; (3) *full model*: the bones are instantiated, and the instance deformation, skinning models and appearance loss are also activated in order to learn the full model.



Fig. 3: Examples of the 3D Toy Bird Dataset. Each bird toy was 3D scanned and the photographed "*in the wild*".

4 Experiments

We perform an extensive set of experiments to evaluate our method and compare to the state of the art. To this end, we collect three datasets, two of real animals, birds and horses, and one using toy birds of which we can obtain ground truth 3D scans.

4.1 Dataset and Implementation Details

Video Datasets.

We experiment with two types of objects: birds and horses. For each category, we extract a collection of short video clips from YouTube. The exact links to these videos and the preprocessing details are included in the sup. mat. We use the off-the-shelf PointRend model [18] to detect and segment the object instances, remove the frames where the object is static, and automatically split the remaining frames into short clips, each containing one single object. The frames and the masks are then cropped around the objects and resized to 128×128 for training. We also run the off-the-shelf RAFT model [79] on the full frames to estimate optical flow between consecutive frames, and account for the cropping and resizing to obtain the correct optical flow for the crops. This procedure creates $1,962$ and 114 short clips of birds and horses respectively, each containing 16 to a few hundred frames with paired image, mask and flow. We randomly split them into $1,767/195$ and $103/11$ training/testing sequences for birds and horses respectively.

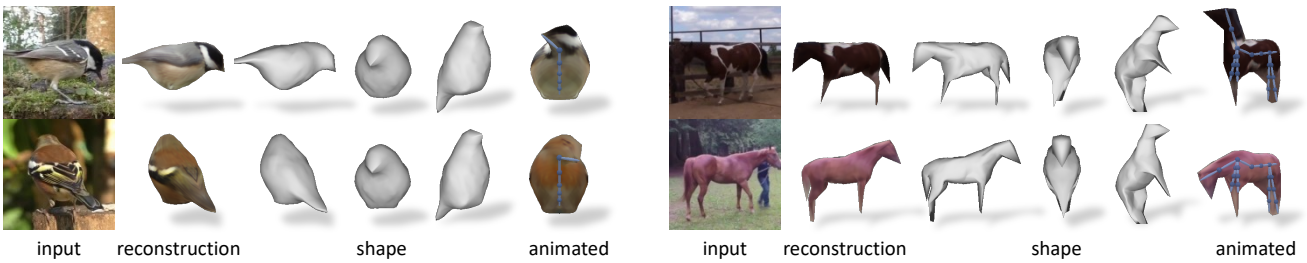


Fig. 4: **Qualitative Examples.** We show multiple views of the reconstructed mesh together with a textured view and animated version of the bird that we obtained by rotating the learned bones. We find that the model is able to recover the shape well even when seen from novel viewpoints. The animation is able to generate believable poses.

Table 2: **Evaluation on Toy Bird Scans.** Shape reconstruction quality measured by bi-directional Chamfer Distance between predicted shape and ground-truth scans. The lower the better. template shape, viewpoint, 2D keypoint, mask, optical flow, video. "finetuned" indicates pretrained models finetuned on our video dataset.

	Supervision	Chamfer Distance (cm) ↓
CMR [19] (finetuned)		1.35 ± 0.81
U-CMR [23] (finetuned)		1.82 ± 0.93
VMR [20] (finetuned)		1.28 ± 0.69
UMR [28] (finetuned)	+SCOPS	1.24 ± 0.75
CMR [19]		5.94 ± 10.33
U-CMR [23]		4.36 ± 1.56
VMR [20]		1.90 ± 0.96
UMR [28]		2.26 ± 1.12
UMR [28]	+SCOPS	1.82 ± 0.93
Ours		**1.51** ± 0.89

3D Toy Bird Dataset.

In order to properly evaluate and compare the quality of the reconstructed 3D shapes produced by different methods, we introduce a 3D Toy Bird Dataset, which consists of ground-truth 3D scans of realistic toy bird models and photographs of them taken in real world environments. Fig. 3 shows examples of the dataset. Specifically, we obtain 23 toy bird models, and used Apple RealityKit Object Capture API [82] to capture accurate 3D scans from turn-table videos. For each model, we then take 5 photographs from different viewpoints in 3 different outdoor scenes, resulting in a total of 345 images. We will release the dataset and ground-truth for future benchmarking.

Implementation Details.

Our reconstruction model is implemented using three neural networks (f_V, f_ξ, f_T) as well as a set of of trainable parameters for the categorical prior shape ΔV_{tmpl}. The shape network f_V and the rigid pose network f_ξ are simple encoders with downsampling convolutional layers that take in an image and predict vertex deformations ΔV_{ins}, skinning parameters $\xi_{2:B}$, and rigid pose ξ_1 and $\mathbf{J}_1$ as flattened vectors. The texture network f_T is an encoder-decoder that predicts the texture map T from an image. We use Adam optimizers with a learning rate of 10^{-4} for all networks, and a learning rate 0.01 for the category shape parameters ΔV_{tmpl}. We use a symmetric ico-sphere as the initial mesh. For each training iteration, we randomly sample 8 consecutive frames from 8 sequences. The models are

trained in three phases described in Section 3.6. All details are included in the sup. mat.

4.2 Qualitative Results

Figure 4 shows qualitative 3D reconstruction results obtained from our model. Note that videos are no longer needed during inference and the shown predictions come from a single frame. Despite not requiring any explicit 3D, viewpoint or keypoint supervision, our model learns to reconstruct accurate 3D shapes from only monocular training videos. The reconstructed 3D meshes can be animated with our skinning model by transforming the bones of the learned shape. This animation can also be transferred between instances.

4.3 Comparisons with State-of-the-Art Methods

We compare our model with a number of state-of-the-art learning-based reconstruction methods, including CMR [19], U-CMR [23], UMR [28] and VMR [20]. CMR requires 2D keypoint annotations for initializing the 3D shape and viewpoints and also for the training loss. U-CMR removes keypoint supervision but requires a 3D template shape, and UMR replaces that with part segmentation maps from SCOPS [33] which relies on supervised ImageNet pretraining. VMR [20] allows for deformations but it requires the same level of supervision as CMR. All of them rely on external geometric supervision to establish correspondences for learning 3D shapes. We train all these methods on our video dataset with only mask supervision and show that without the additional supervision, all these methods reconstruct poor shapes. We also finetune their models pre-trained on CUB [31] with the required keypoint, camera view or template shape supervision on our bird video dataset. Finally, we also train UMR from scratch on our bird video dataset with SCOPS predictions obtained from the pre-trained SCOPS model.

On 3D Toy Bird Scans.

Our toy scan dataset allows for a direct evaluation of shape prediction. We first scale the predicted shapes to match the volume of the scans and roughly align the canonical pose of each method to the scans manually. Each individual predicted shape is further aligned to the ground-truth scan using Iterative Closest Point (ICP) [83] and the symmetric (average of scan-to-object and object-to-scan) Chamfer distance is reported in centimeters, Table 2, by assuming the width of each bird to be 10 cm. While the reconstruction quality of other methods is good when trained with more geometric supervision, it degrades strongly without this training signal resulting in worse reconstructions when compared to our method. Note that this metric evaluates the individual shape predictions regardless the viewpoints. Next we evaluate the consistency across views.

On Bird Video Dataset.

Since we do not have ground-truth 3D shape and viewpoints for direct evaluation on our video test set, we measure reconstruction quality via a mask *forward* projection accuracy from one frame to another, using the object masks predicted by PointRend [18] as the pseudo ground-truth. This evaluates the shape and viewpoint quality as the object from a past frame is projected to a future frame which can only align when both shape and pose are estimated correctly, but cannot account for non-rigid deformation between frames. For each test sequence, we predict the shape at frame t and render the object mask from the pose at frame $t + \Delta t$ with an offset Δt of 0, 5 and 20 frames. We then compute the mean Intersection over Union (mIoU) between the rendered masks and the ground-truth masks at $t + \Delta t$. Table 3 summarizes the results, which suggest that our model achieves both better shape reconstruction and viewpoint consistency. We also compute the metrics on our model with frame-specific deformations predicted at frame $t + \Delta t$ applied to the shape predicted at frame t. This further improves the mask reprojection IoU, confirming that our model learns correct frame-specific deformations. Other methods tend to overfit the shape to the image, resulting in a larger decrease in reprojection accuracy with increasing Δt.

We also compare the distribution of estimated viewpoints/object poses by plotting the elevation and azimuth predicted on the test set in Fig. 5. Our method is able to learn the full azimuth range, while other methods, with the exception of U-CMR, only predict limited range of views (azimuth) without additional geometric supervision.

Table 3: **Mask Forward Projection IoU.** Shape reconstruction quality and temporal consistency measured by projecting the shape predicted at frame t to a different pose at a *future* frame $t + \Delta t$ and comparing the masks at $t + \Delta t$. The higher the better. "finetuned" indicates pretrained models finetuned on our video dataset.

Frame offset	Supervision	$\Delta t = 0$	$\Delta t = 5$	$\Delta t = 20$
CMR [19] (finetuned)	📷 🔑 ✂	0.770 ±0.13	0.722 ±0.13	0.712 ±0.13
U-CMR [23] (finetuned)	🧍 ✂	0.790 ±0.06	0.761 ±0.07	0.758 ±0.07
VMR [20] (finetuned)	🧍 📷 🔑 ✂	0.807 ±0.08	0.752 ±0.08	0.737 ±0.09
UMR [28] (finetuned)	✂ + SCOPS	0.847 ±0.05	0.782 ±0.07	0.772 ±0.07
CMR [19]	✂	0.634 ±0.06	0.605 ±0.11	0.596 ±0.11
U-CMR [23]	✂	0.725 ±0.05	0.714 ±0.06	0.700 ±0.06
VMR [20]	✂	0.777 ±0.05	0.720 ±0.07	0.700 ±0.09
UMR [28]	✂	0.853 ±0.04	0.798 ±0.06	0.788 ±0.06
UMR [28]	✂ + SCOPS	0.830 ±0.04	0.766 ±0.07	0.753 ±0.07
Ours (articulation fixed)	✂ ⏩ 📹	**0.855** ±0.07	0.810 ±0.07	0.805 ±0.07
Ours (articulation transferred)	✂ ⏩ 📹	**0.855** ±0.07	**0.844** ±0.07	**0.845** ±0.07

Table 4: **Ablation Studies with 3D Toy Bird Scans.** Every component of our model helps to improve the final performance.

	Chamfer Distance (cm)
Full model	**1.51** ±0.89
w/o front-back hypothesis	2.52 ±1.41
w/o symmetry	2.19 ±1.24
w/o video training	2.20 ±1.03
w/o learned prior shape	3.92 ±1.47

Qualitative Comparisons.

Fig. 5 shows a qualitative comparison of different methods. When methods relying on more geometric supervision (CMR, U-CMR, VMR) are trained without this learning signal, they fail to produce reasonable shape reconstructions. UMR trained without SCOPS part segmentations overfits to the input views producing inaccurate 3D shapes. Our method reconstructs accurate shape and pose, despite not using keypoint or template supervision. We refer the reader to the sup. mat. for more results. Note that our model is trained on 128×128 images, whereas other methods train on 256×256 images and, except U-CMR, sample the texture directly from the input image, explaining the difference in the texture quality.

On Horse Video Dataset.

For horses, we compare qualitatively with LASR [34] in Fig. 6, which is an optimization-based method for single video sequences. While their reconstruction appears to be convincing in the original viewpoint, the actual mesh often does not resemble the shape of a horse. Running LASR on such a sequence takes over four hours.

4.4 Ablation and Analysis

We ablate the different components of our method quantitatively on our toy bird dataset in Table 4 and Fig. 7. We find that all components are necessary for the final performance. The pose distribution in Table 4 shows that the model only learns the full 360-degree (azimuth) view of the object when all components are active. Especially the two-view-ambiguity resolution and the shape symmetry are important to learn the pose while video training helps to discover the backside of the object. Without a good pose prediction the reconstructions look reasonable in the input view, reveal to be degenerate from other directions.

The model without symmetry produces unrealistic shapes indicating that symmetry is a useful prior, even when learning deformable shapes. Similarly, the shape prior is important to discover fine details (*e.g.* beak and tail) that are not visible in every image. The full model predicts a full range of

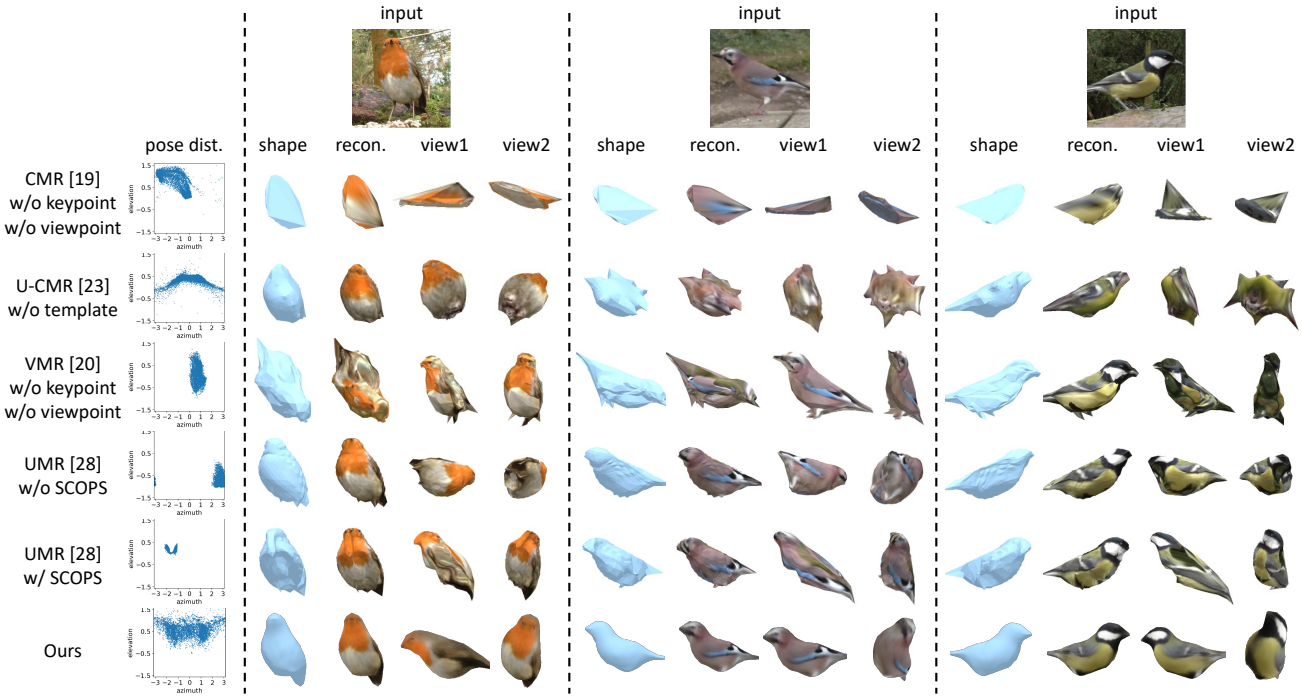


Fig. 5: **Visual Comparison.** We compare to state-of-the-art methods trained without external geometric supervision in the form of 2D keypoints, viewpoint, or template shape. As UMR leverages weak-supervision using part segmentation maps from SCOPS [33], we show versions trained with and without SCOPS. Our method consistently reconstructs reasonable 3D shapes and the predictions cover full 360-degree (azimuth) view, whereas other methods produce poor reconstructions and their viewpoint predictions collapse to only a limited range with the exception of U-CMR. Other methods, except for U-CMR directly copy the texture from the input image using texture flow. Hence, although the texture appears sharper from the input view, they are often incorrect as seen from other views. See the sup. mat. for extended results.

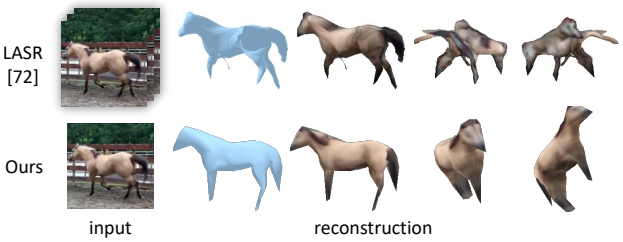


Fig. 6: **Comparison with LASR [34].** While the rendering in the original viewpoint looks convincing, the shape produced by LASR is distorted and does not resemble the actual shape of a horse. Since our method trains on multiple sequences it can learn a consistent shape.

viewpoints (Fig. 7) and the most consistent shape (Table 4).

We train another model without the learned category prior shape, predicting individual shapes for each bird. The resulting reconstructions are inconsistent across different instances, shown in Fig. 7. This suggests that the full model is able to leverage shape prior of the whole category, which is a major benefit of *learning* in a reconstruction pipeline.

5 Limitations and Future Work

Our method still requires segmentation masks obtained from the off-the shelf model as supervision for training. Moreover, their quality affects the fidelity of our reconstructions. Thus, similar to comparable methods, our reconstructions do not capture fine details well, such as legs and the beak. The texture prediction sometimes results in low quality reconstructions especially when the input image is affected by motion blur. Currently, we have to handcraft a structure for various types of animals, for example different structures for horses (quadrupeds) and birds. How to automatically discover plausible bone structures is also an interesting question to explore for future work.

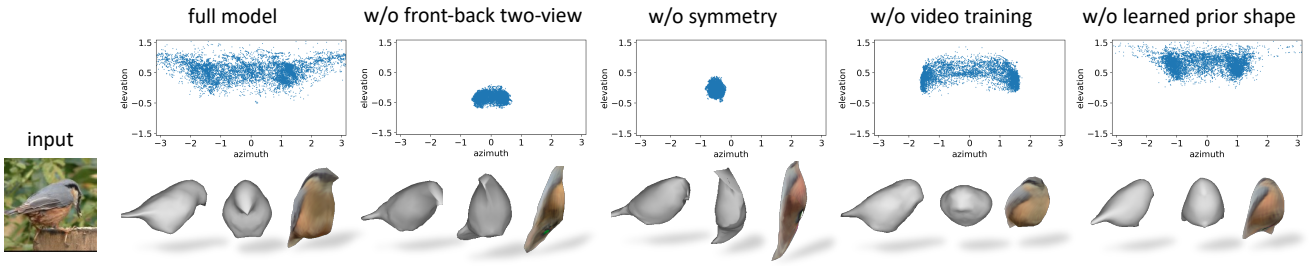


Fig. 7: **Ablation Studies.** We train our model without some of the key components and plot the distribution of the predicted poses. Without 2-view ambiguity resolution or symmetry constraint, the pose prediction collapses. Video training and learning a shape prior also help improve the poses and shapes.

6 Conclusions

We have presented a method to learn articulated 3D representations of deformable objects from monocular videos without explicit geometric supervision, such as keypoints, viewpoint or template shapes. The resulting 3D meshes are temporally consistent and can be animated. The method can be trained from videos and only needs off-the-shelf object detection and optical flow models for preprocessing. For reproducibility, comparison and benchmarking, the dataset, code and models will be released with the paper.

Acknowledgments

We thank Zirui Wang for insightful discussions and Xueting Li for sharing the code for VMR with us. Shangzhe Wu is supported by Meta Research. Tomas Jakab is supported by Clarendon Scholarship. Christian Rupprecht is supported by Innovate UK (project 71653) on behalf of UK Research and Innovation (UKRI) and the Department of Engineering Science at the University of Oxford. Andrea Vedaldi is supported by EPSRC Grant (Visual AI EP/T028572/1).

References

[1] Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: ECCV (2016)

[2] Girdhar, R., Fouhey, D., Rodriguez, M., Gupta, A.: Learning a predictable and generative vector representation for objects. In: ECCV (2016)

[3] Wu, J., Zhang, C., Xue, T., Freeman, W.T., Tenenbaum, J.B.: Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In: NeurIPS, pp. 82–90 (2016)

[4] Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.-G.: Pixel2mesh: Generating 3d mesh models from single rgb images. In: ECCV (2018)

[5] Groueix, T., Fisher, M., Kim, V.G., Russell, B., Aubry, M.: AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In: CVPR (2018)

[6] Kato, H., Ushiku, Y., Harada, T.: Neural 3d mesh renderer. In: CVPR (2018)

[7] Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: CVPR (2019)

[8] Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: CVPR (2019)

[9] Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., Li, H.: Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In: ICCV (2019)

[10] Gkioxari, G., Malik, J., Johnson, J.: Mesh r-cnn. In: ICCV (2019)

[11] Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: a skinned multi-person linear model. ACM TOG (2015)

[12] Kanazawa, A., Black, M.J., Jacobs, D.W., Malik, J.: End-to-end recovery of human shape and pose. In: CVPR (2018)

[13] Zuffi, S., Kanazawa, A., Jacobs, D., Black, M.J.: 3D menagerie: Modeling the 3D shape and pose of animals. In: CVPR (2017)

[14] Sanyal, S., Bolkart, T., Feng, H., Black, M.: Learning to regress 3D face shape and expression from an image without 3D supervision. In: CVPR (2019)

[15] Zuffi, S., Kanazawa, A., Berger-Wolf, T., Black, M.J.: Three-d safari: Learning to estimate zebra pose, shape, and texture from images "in the wild". In: ICCV (2019)

[16] Reizenstein, J., Shapovalov, R., Henzler, P., Sbordone, L., Labatut, P., Novotny, D.: Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In: ICCV, pp. 10901–10911 (2021)

[17] He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. In: ICCV (2017)

[18] Kirillov, A., Wu, Y., He, K., Girshick, R.: PointRend: Image Segmentation as Rendering. In: CVPR (2020)

[19] Kanazawa, A., Tulsiani, S., Efros, A.A., Malik, J.: Learning category-specific mesh reconstruction from image collections. In: ECCV (2018)

[20] Li, X., Liu, S., Mello, S.D., Kim, K., Wang, X., Yang, M., Kautz, J.: Online adaptation for consistent mesh reconstruction in the wild. In: NeurIPS (2020)

[21] Kokkinos, F., Kokkinos, I.: Learning monocular 3d reconstruction of articulated categories from motion. In: CVPR (2021)

[22] Kulkarni, N., Gupta, A., Tulsiani, S.: Canonical surface mapping via geometric cycle consistency. In: ICCV (2019)

[23] Goel, S., Kanazawa, A., Malik, J.: Shape and viewpoint without keypoints. In: ECCV (2020)

[24] Kokkinos, F., Kokkinos, I.: To the point: Correspondence-driven monocular 3d category reconstruction. In: NeurIPS (2021)

[25] Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In: CVPR (2020)

[26] Wu, S., Rupprecht, C., Vedaldi, A.: Unsupervised learning of probably symmetric deformable 3D objects from images in the wild. In: CVPR (2020). https://elliottwu.com/projects/unsup3d/

[27] Kulkarni, N., Gupta, A., Fouhey, D.F., Tulsiani, S.: Articulation-aware canonical surface mapping. In: CVPR, pp. 449–458 (2020)

[28] Li, X., Liu, S., Kim, K., Mello, S.D., Jampani, V., Yang, M., Kautz, J.: Self-supervised single-view 3D reconstruction via semantic consistency. In: ECCV (2020)

[29] Thewlis, J., Bilen, H., Vedaldi, A.: Modelling and unsupervised learning of symmetric deformable object categories. In: NeurIPS (2018)

[30] Fernandez-Labrador, C., Chhatkuli, A., Paudel, D.P., Guerrero, J.J., Demonceaux, C., Gool, L.V.: Unsupervised learning of category-specific symmetric 3D keypoints from point sets. In: ECCV (2020)

[31] Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology (2011)

[32] Neverova, N., Novotný, D., Vedaldi, A.: Continuous surface embeddings. In: NeurIPS (2020)

[33] Hung, W., Jampani, V., Liu, S., Molchanov, P., Yang, M., Kautz, J.: SCOPS: self-supervised co-part segmentation. In: CVPR

(2019)

[34] Yang, G., Sun, D., Jampani, V., Vlasic, D., Cole, F., Chang, H., Ramanan, D., Freeman, W.T., Liu, C.: LASR: Learning articulated shape reconstruction from a monocular video. In: CVPR (2021)

[35] Yang, G., Sun, D., Jampani, V., Vlasic, D., Cole, F., Liu, C., Ramanan, D.: Viser: Video-specific surface embeddings for articulated 3d shape reconstruction. In: NeurIPS (2021)

[36] Yang, G., Vo, M., Natalia, N., Ramanan, D., Andrea, V., Hanbyul, J.: Banmo: Building animatable 3d neural models from many casual videos. In: CVPR (2022)

[37] Tulsiani, S., Kulkarni, N., Gupta, A.: Implicit mesh reconstruction from unannotated image collections. arXiv preprint arXiv:2007.08504 (2020)

[38] Garg, R., Kumar, B.G.V., Carneiro, G., Reid, I.D.: Unsupervised CNN for single view depth estimation: Geometry to the rescue. In: ECCV (2016)

[39] Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: CVPR (2017)

[40] Liu, S., Li, T., Chen, W., Li, H.: Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In: ICCV (2019)

[41] Kato, H., Harada, T.: Learning view priors for single-view 3d reconstruction. In: CVPR (2019)

[42] Henderson, P., Ferrari, V.: Learning single-image 3d reconstruction by generative modelling of shape, pose and shading. IJCV, 1–20 (2019)

[43] Wu, S., Makadia, A., Wu, J., Snavely, N., Tucker, R., Kanazawa, A.: De-rendering the world's revolutionary artefacts. In: CVPR (2021)

[44] Kudo, Y., Ogaki, K., Matsui, Y., Odagiri, Y.: Unsupervised adversarial learning of 3D human pose from 2D joint locations. arXiv preprint arXiv:1803.08244 (2018)

[45] Chen, C., Tyagi, A., Agrawal, A., Drover, D., MV, R., Stojanov, S., Rehg, J.M.: Unsupervised 3d pose estimation with geometric self-supervision. In: CVPR (2019)

[46] Henzler, P., Mitra, N.J., Ritschel, T.: Escaping plato's cave using adversarial training: 3D shape from unstructured 2D image collections. In: ICCV (2019)

[47] Nguyen-Phuoc, T., Li, C., Theis, L., Richardt, C., Yang, Y.-L.: Hologan: Unsupervised learning of 3d representations from natural images. In: ICCV (2019)

[48] Nguyen-Phuoc, T., Richardt, C., Mai, L., Yang, Y., Mitra, N.J.: Blockgan: Learning 3d object-aware scene representations from unlabelled images. In: NeurIPS (2020)

[49] Ye, Y., Tulsiani, S., Gupta, A.: Shelf-supervised mesh prediction in the wild. In: CVPR (2021)

[50] Schwarz, K., Liao, Y., Niemeyer, M., Geiger, A.: Graf: Generative radiance fields for 3d-aware image synthesis. In: NeurIPS (2020)

[51] Niemeyer, M., Geiger, A.: Giraffe: Representing scenes as compositional generative neural feature fields. In: CVPR (2021)

[52] Zhang, Y., Chen, W., Ling, H., Gao, J., Zhang, Y., Torralba, A., Fidler, S.: Image gans meet differentiable rendering for inverse graphics and interpretable 3d neural rendering. In: ICLR (2021)

[53] Chan, E., Monteiro, M., Kellnhofer, P., Wu, J., Wetzstein, G.: pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In: CVPR (2021)

[54] Pan, X., Dai, B., Liu, Z., Loy, C.C., Luo, P.: Do 2d gans know 3d shape? unsupervised 3d shape reconstruction from 2d image gans. In: ICLR (2021)

[55] Faugeras, O., Luong, Q.-T.: The Geometry of

Multiple Images. MIT Press, ??? (2001)

[56] Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, ISBN: 0521540518, ??? (2004)

[57] Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohli, P., Shotton, J., Hodges, S., Fitzgibbon, A.W.: KinectFusion: Real-time dense surface mapping and tracking. In: Proc. ISMAR (2011)

[58] Newcombe, R.A., Fox, D., Seitz, S.M.: DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In: CVPR (2015)

[59] Yariv, L., Kasten, Y., Moran, D., Galun, M., Atzmon, M., Ronen, B., Lipman, Y.: Multiview neural surface reconstruction by disentangling geometry and appearance. In: NeurIPS (2020)

[60] Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In: NeurIPS (2021)

[61] Oechsle, M., Peng, S., Geiger, A.: Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In: ICCV (2021)

[62] Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020)

[63] Park, K., Sinha, U., Barron, J.T., Bouaziz, S., Goldman, D.B., Seitz, S.M., Martin-Brualla, R.: Nerfies: Deformable neural radiance fields. In: ICCV (2021)

[64] Gafni, G., Thies, J., Zollhöfer, M., Nießner, M.: Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In: CVPR (2021)

[65] Tretschk, E., Tewari, A., Golyanik, V., Zollhöfer, M., Lassner, C., Theobalt, C.: Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In: ICCV (2021)

[66] Raj, A., Zollhoefer, M., Simon, T., Saragih, J., Saito, S., Hays, J., Lombardi, S.: Pva: Pixel-aligned volumetric avatars. In: CVPR (2021)

[67] Noguchi, A., Sun, X., Lin, S., Harada, T.: Neural articulated radiance field. In: ICCV (2021)

[68] Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-NeRF: Neural Radiance Fields for Dynamic Scenes. In: CVPR (2020)

[69] Paysan, P., Knothe, R., Amberg, B., Romdhani, S., Vetter, T.: A 3D face model for pose and illumination invariant face recognition. In: Advanced Video and Signal Based Surveillance (2009)

[70] Tung, H., Tung, H., Yumer, E., Fragkiadaki, K.: Self-supervised learning of motion capture. In: NIPS (2017)

[71] Arnab, A., Doersch, C., Zisserman, A.: Exploiting temporal context for 3d human pose estimation in the wild. In: CVPR (2019)

[72] Doersch, C., Zisserman, A.: Sim2real transfer learning for 3d human pose estimation: motion to the rescue. NeurIPS (2019)

[73] Kanazawa, A., Zhang, J.Y., Felsen, P., Malik, J.: Learning 3d human dynamics from video. In: CVPR (2019)

[74] Zhang, J.Y., Felsen, P., Kanazawa, A., Malik, J.: Predicting 3d human dynamics from video. In: ICCV (2019)

[75] Feng, Y., Wu, F., Shao, X., Wang, Y., Zhou, X.: Joint 3d face reconstruction and dense alignment with position map regression network. In: ECCV (2018)

[76] Tran, L., Liu, X.: On learning 3d face morphable model from in-the-wild images. IEEE TPAMI **43**(1), 157–171 (2019)

[77] Novotný, D., Larlus, D., Vedaldi, A.: Learning 3D object categories by looking around them. In: ICCV (2017)

[78] Henzler, P., Reizenstein, J., Labatut, P., Shapovalov, R., Ritschel, T., Vedaldi, A., Novotny, D.: Unsupervised learning of 3d object categories from videos in the wild. In: CVPR (2021)

[79] Teed, Z., Deng, J.: RAFT: recurrent all-pairs field transforms for optical flow. In: ECCV (2020)

[80] Ravi, N., Reizenstein, J., Novotny, D., Gordon, T., Lo, W.-Y., Johnson, J., Gkioxari, G.: Accelerating 3d deep learning with pytorch3d. arXiv preprint arXiv:2007.08501 (2020)

[81] Sorkine, O., Alexa, M.: As-rigid-as-possible surface modeling. In: Symposium on Geometry Processing, vol. 4, pp. 109–116 (2007)

[82] Apple Reality Kit Object Capture API. https://developer.apple.com/augmented-reality/object-capture/. Accessed: 2021-10

[83] Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. IEEE TPAMI **14**(2) (1992)

[84] Wu, Y., He, K.: Group normalization. In: ECCV (2018)

[85] Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: ICML (2013)

7 Appendices

7.1 Additional Results

On top of the material in this pdf, please see the **supplementary video** for more qualitative results.

7.1.1 Additional Comparisons with State-of-the-Art Methods

Comparison of Learned Rigid Pose Distributions.

Estimating the viewpoint/object pose is a key factor of learning consistent 3D shapes. We plot the elevation and azimuth of the rigid poses (viewpoint) predicted by various methods (CMR [19], U-CMR [23], UMR [28] and VMR [20]) and compare them in Fig. 9. As shown in the plots, without additional geometric supervision (keypoint, viewpoint or template shape), existing methods are not able to learn correct poses and the predicted poses tend to collapse to only half of the azimuth, resulting in poor shape reconstructions that are not consistent across different inputs, as shown in Fig. 13 as well as in the main paper. The only exception is U-CMR [23] which explicitly uses an extensive viewpoint search (camera multiplex) to encourage diverse viewpoint predictions. However, without a good shape template, the resulting viewpoints are also not correct leading to poor shapes.

Additional Qualitative Comparisons.

Fig. 13 provides a few more examples comparing the reconstruction results of our model and several state-of-the-art methods. We also show shape reconstructions rendered with a static synthetic texture, shown in Fig. 8, to clearly illustrate the orientation of the predicted shape. Our model learns more accurate 3D shapes, despite not requiring explicit geometric supervision from keypoints, viewpiont or template shapes. More comparisons on entire video sequences are provided in the supplementary video.

Fig. 14 shows more qualitative comparisons to LASR [34] on horses.

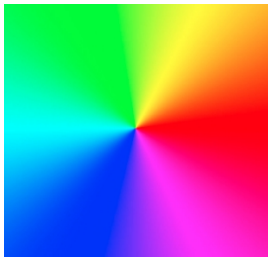


Fig. 8: Synthetic texture used for visualization in Fig. 13.

7.1.2 PCA Analysis on Learned Shape Space

We analyze the learned articulated shape space across the dataset using Principal Component Analysis (PCA). Figure 11 visualizes the first 6 principal components. Each principal component corresponds to a typical bird movement, which means that the model learns meaningful articulations that are reflected in the skeleton and not in the shape deformation component.

7.1.3 Additional Reconstruction Results

Figure 15 shows more bird reconstructions from various viewpoints. The model is robust against various input images, including frontal views and blurry images.

7.1.4 Texture Swapping and Animation

Our model reconstructs the birds in the canonical pose, where the shape and texture of different birds are aligned in the canonical representation. This allows us to easily edit the texture, for example swapping the texture with another bird, as shown in Fig. 16.

Moreover, with the learned articulation model, we can also easily animate the reconstructed birds in 3D, by rotating the bones, also illustrated in Fig. 16.

7.2 Datasets

We collected videos containing birds and horses from YouTube and automatically pre-processed them as follows. We use PointRend model [18] to obtain detection and segmentation of the object instances.

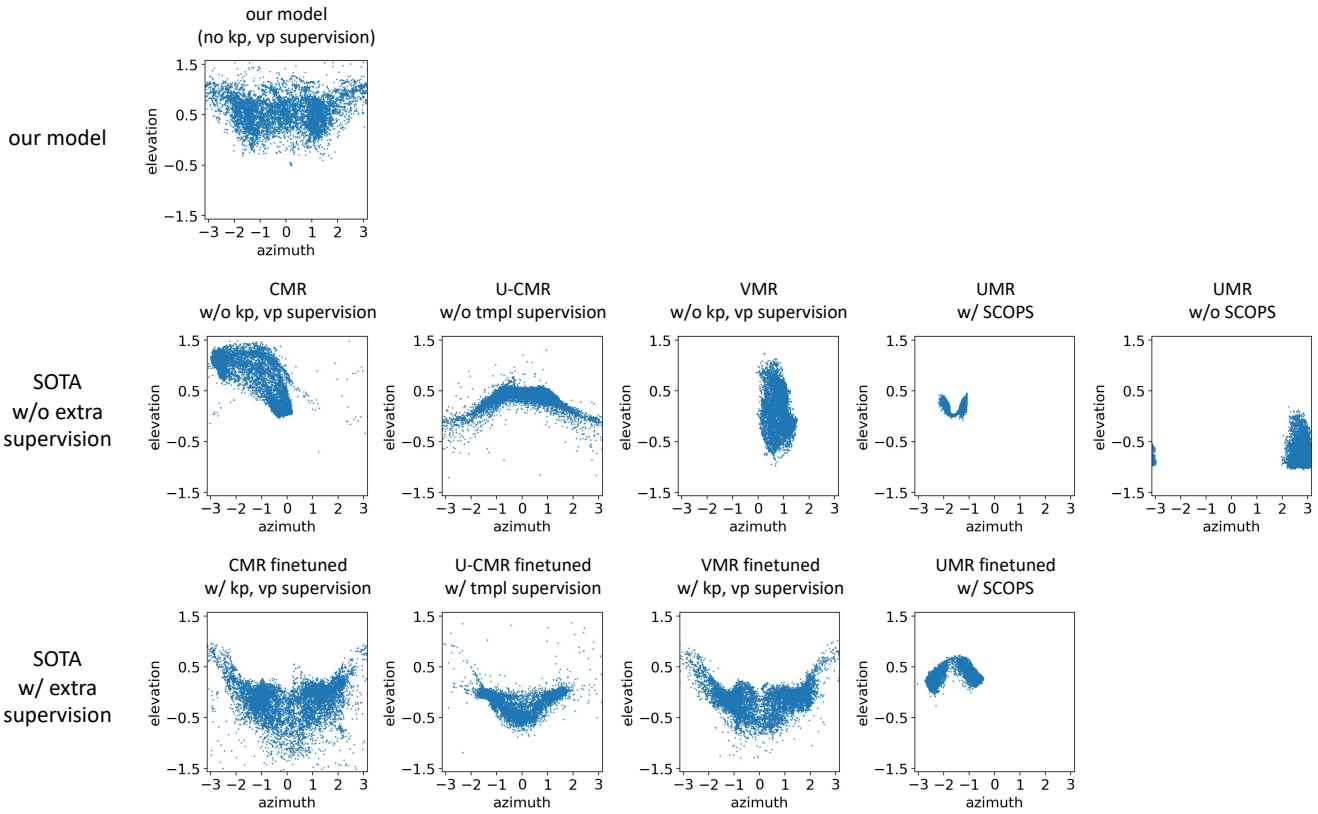


Fig. 9: **Comparison of the Learned Rigid Pose Distributions.** We plot the distributions of the rigid poses predicted by various methods. Without additional geometric supervision ("kp" for keypoint, "vp" for viewpoint and "tmpl" for template shape), the rigid poses (viewpoint) predicted by existing methods collapse to only a limited range, hence resulting in poor, inconsistent shape predictions. For example, UMR is able to predict only frontal poses.

Each video is then split into short clips containing a single object while we also remove frames that would contain humans after final cropping. We also filter out static frames from the clips using optical flow computed by off-the-shelf RAFT model [79]. The optical flow is then recomputed one more time to account for the removed frames. Finally, we crop the frames, segmentation masks and optical flow around the detected object bounding boxes and resize them to 128×128 for training.

Bird Videos.

We downloaded a single four-hour long video of birds with static camera from YouTube[1].

Horse Videos

We searched YouTube for terms such as *horse training*, *horse training in pen*, *leading horse* and selected 12 videos [2] where a horse is trained without a mounted rider. As the content in these videos can be varied, we manually roughly split the videos into clips where the horse is the main focus before applying the automatic pre-processing.

---

[1] https://youtu.be/xbs7FT7dXYc

[2] https://youtu.be/YM_jw4g0urc, https://youtu.be/0MSifR_4BAw, https://youtu.be/qincEZod6mQ, https://youtu.be/nyBMMoeg4CU, https://youtu.be/_BN-0e1r89E, https://youtu.be/Bb8nquNkruY, https://youtu.be/wq4vryHoe-0, https://youtu.be/asE5y2qO5dw, https://youtu.be/4ZuKilrSgXI, https://youtu.be/dJtyHyt6bOk, https://youtu.be/0YUAArukMFc, https://youtu.be/U61UqV0x4QQ

Table 5: Architecture of the shape f_S and pose network f_P. The network follows a convolutional encoder structure. n is the number of parameters predicted by each network.

Encoder	Output size
Conv(3, 64, 4, 2, 1) + GN(16) + LReLU(0.2)	64 × 64
Conv(64, 128, 4, 2, 1) + GN(32) + LReLU(0.2)	32 × 32
Conv(128, 256, 4, 2, 1) + GN(64) + LReLU(0.2)	16 × 16
Conv(256, 512, 4, 2, 1) + GN(128) + LReLU(0.2)	8 × 8
Conv(512, 512, 4, 2, 1) + LReLU(0.2)	4 × 4
Conv(512, 256, 4, 1, 0) + ReLU	1 × 1
Conv(256, n, 1, 1, 0) → output	1 × 1

Table 6: Architecture of the texture network f_T. The network follows an encoder-decoder structure.

Encoder	Output size
Conv(3, 64, 4, 2, 1) + GN(16) + LReLU(0.2)	64 × 64
Conv(64, 128, 4, 2, 1) + GN(32) + LReLU(0.2)	32 × 32
Conv(128, 256, 4, 2, 1) + GN(64) + LReLU(0.2)	16 × 16
Conv(256, 512, 4, 2, 1) + GN(128) + LReLU(0.2)	8 × 8
Conv(512, 512, 4, 2, 1) + LReLU(0.2)	4 × 4
Conv(512, 256, 4, 1, 0) + ReLU	1 × 1

Decoder	Output size
Deconv(256, 512, 4, 1, 0) + ReLU	4 × 4
Upsample(2) + Conv(512, 512, 3, 1, 1) + GN(128) + ReLU	8 × 8
Upsample(2) + Conv(512, 256, 3, 1, 1) + GN(64) + ReLU	16 × 16
Upsample(2) + Conv(256, 128, 3, 1, 1) + GN(32) + ReLU	32 × 32
Upsample(2) + Conv(128, 64, 3, 1, 1) + GN(16) + ReLU	64 × 64
Upsample(2) + Conv(64, 64, 3, 1, 1) + GN(16) + ReLU	128 × 128
Conv(64, 3, 5, 1, 2) + Sigmoid → output T	128 × 128

7.2.1 3D Toy Bird Dataset

Figure 12 shows renderings of six 3D models in our 3D toy bird dataset and the corresponding photographs of the toys "in the wild". The visual appearance of the birds in the photos is close to the real one is CUB200 or our video dataset. The 3D reconstructions are of high quality and allow for precise quantitative evaluation. For future benchmarking and comparisons, we will release the dataset together with the paper.

7.3 Broader Impact

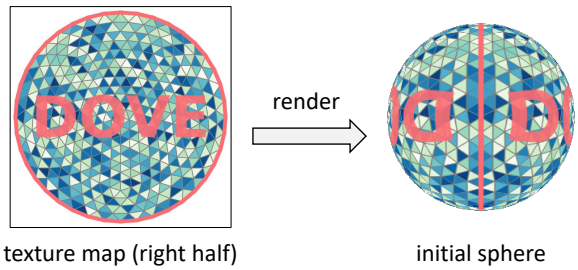
Our work focuses on 3D reconstruction of deformable objects from monocular videos. We expect this work to be most useful for object categories that do not have sophisticated 3D ground-truth annotations and 3D shape models. This is mainly the case for animals, as shown for birds and horses here, and thus the work can potentially be of use in behavioral research of animals in the wild. Our 3D Toy Bird dataset does not contain any humans, only toy birds in nature background and was captured by the authors. Thus, the copyright of the dataset is with the authors and it does not violate the personal privacy of individuals. Overall, we expect this work to impact mostly the research community with very little impact on society in the short term.

In the long term, we expect the task of obtaining 3D models and algorithms that can lift objects from images into 3D to become more and more important; for example in XR and VFX applications. Our method shows that obtaining these models can be achieved without external geometric supervision, which is often difficult to collect for arbitrary object categories.

Table 7: Training details and hyper-parameters.

Parameter	Value/Range
Optimizer	Adam
Learning rate (f_V, f_ξ, f_T)	1×10^{-4}
Learning rate (ΔV_{tmpl})	1×10^{-2}
Number of epochs	16
Number of sequences per batch	4
Number of frames per sequence	8
Loss weight λ_{im}	1
Loss weight λ_{mask}	2
Loss weight λ_{ARAP}	50
Loss weight λ_{Lap}	1
Loss weight λ_{nrm}	1
Loss weight λ_{flow}	100
Loss weight λ_{sym}	0.05
Loss weight λ_{pose}	0.1
Input image size	128×128
Texture image size	128×128
Field of view (FOV)	25°
Camera location	$(0, 0, 10)$
Initial mesh center	$(0, 0, 0)$
Number of bones (bird)	6
Number of bones (horse)	$6 + 4 \times 4$

Fig. 10: **Texture Mapping.** The texture is mapped from a circle in the texture to both, left and right, sides of the initial sphere.



7.4 Mathematical Details

In this section we expand the underlying math of shape, pose and symmetries to provide a detailed mathematical foundation of the underlying principles.

7.4.1 Shape and Pose

Recall that we represent the shape of each object instance using a hierarchical shape model. The model first predicts the shape of each instance V at rest pose as:

$$V = V_{\text{base}} + \Delta V_{\text{tmpl}} + \Delta V, \tag{5}$$

where V_{base} is the initial fixed shape parametrized by a symmetric ico-sphere mesh with 642 vertices and 1,280 faces. ΔV_{tmpl} is a set of trainable parameters initilized as zeros and directly optimized during training, such that $V_{\text{tmpl}} = V_{\text{base}} + \Delta V_{\text{tmpl}}$ represents the category-specific template shape shared across all instances of the category. ΔV is instance-specific deformation predicted from each input image by the shape network f_V.

This rest pose shape V is further transformed into the actual mesh observed in the image by a posing (or skinning) function $g(V, \xi)$, described in Section 7.4.2, where the pose parameter ξ consists of the rigid pose $\xi_1 \in SE(3)$ and a set of rotations of the bones $\xi_b \in SO(3), b = 2, ..., B$.

The rigid pose is predicted by the pose network f_ξ as a 3D rotation parametrized by a forward vector and 3D translations in xyz axes. Specifically, we predict a forward vector and define $(0, 1, 0)^T$ as the up direction and obtain a basis for the rotation matrix from these vectors by two consecutive cross products. This effectively disables the in-plane rotation as the objects tend to stay mostly upright. The translations are capped at a range roughly corresponding to 0.4 of the image size after projection.

The bone rotations are predicted by the shape network f_V simply as a set of Euler angles.

7.4.2 Skinning Equation

Each bone b is rigidly attached to a parent bone $\pi(b)$ forming a kinematic tree. Vector $\mathbf{J}_b$ is the location of the joint between b and $\pi(b)$ expressed relative to the parent. If V_{ib} are the coordinates of mesh vertex i expressed relative to bone b, we can find the location in world space as:

$$V_{i0} = G_b(\xi)V_{ib}, \quad G_b = G_{\pi(b)} \circ g_b, \quad g_b(\xi) = \begin{bmatrix} R_{\xi_b} & \mathbf{J}_b \\ 0 & 1 \end{bmatrix}.$$

Given an initial mesh V at rest, then posed version is found by first expressing the vertices relative to each bone b in the rest configuration ξ_0 using transformations $G_b(\xi_0)^{-1}$ and then posing the vertex by applying transformation $G_b(\xi)$. This is weighed by the strength of association of each vertex to a bone, resulting in the *skinning equation* [11]:

$$V_i(\xi) = \left(\sum_{b=1}^{B} w_{ib} G_b(\xi) G_b(\xi_0)^{-1} \right) V_i. \tag{6}$$

Skinning Weights.

As described in the main paper, we estimate the bone structure using the two most extreme points of the mesh at its rest pose. The one with positive z coordinate is selected as the head end and the other one the tail end, ensuring consistent orientation of the bone structure. The rotation of individual bones is represented using Euler angles about the xyz-axes in its local coordinate frame, where the center of rotation is specified by the joint location.

Recall Eq. (2) of the main paper where we use skinning weights w_{bi} that associate each mesh vertex with the bones. We softly assign each mesh vertex with the bones based on its distances to the bones. The weights are defined as the inverse of the distance between a vertex $[V_{\text{ins}}]_i$ and a bone b at their rest pose. We normalize this distance for each vertex over all the bones with softmax function with temperature:

$$w_{bi} = \frac{e^{d_{bi}/T}}{\sum_{k=1}^{B} e^{d_{ki}/T}}, \qquad d_{bi} = 1/(\min_{r \in [0,1]} \|[V_{\text{ins}}]_i - r\mathbf{s}_{b1} - (1-r)\mathbf{s}_{b2}\|_2^2 + \epsilon), \tag{7}$$

where T is the temperature parameter, $(\mathbf{s}_{b1}, \mathbf{s}_{b2})$ is the line segment defining the bone b at its rest pose and ϵ is a small number to avoid division by zero.

Bone Structure Initialization.

We estimate the bone structure using a simple heuristics. Given the shape corresponding to the rest pose, we define a fixed number of bones inside the mesh forming a 'spine' which we set to lie on two line segments going from the two most extreme point of the mesh to the center. We then divide each line segment into equally-sized parts that define the origin and the length of each bone.

For quadruped animals, we also define bones forming four legs. We divide the space into quadrants along the x and z axis and then we found the lowest point on the mesh along the y axis in each quadrant. These lowest points are supposed to correspond to feet. We define a line segment between each of the lowest points, and the closest joint on the spine. As done with the spine, we then divide each line segment into equally-sized parts forming the individual bones.

7.4.3 Symmetry

We first define the action of the mirror mapping m on several objects. First, m acts on 3D points as the matrix that flips the x axis:

$$m = \begin{bmatrix} -1 & & \\ & 1 & \\ & & 1 \end{bmatrix}. \tag{8}$$

Note that $m^{-1} = m$.

Next, let $V = \{V_i \in \mathbb{R}^3\}_{i=1,\dots,|V|}$ be the vertices of a mesh. The mesh is posed by the transformation $g \in SE(3)$, which we also write as $g(\cdot, \xi)$ for a certain vector of pose parameters ξ. The *mirror-symmetric pose* of g is the *conjugate transformation* $mgm \in SE(3)$. This also defines the meaning of the symbol $m\xi$ as the parameters of the conjugate:

$$g(\cdot, m\xi) \equiv mg(\cdot, \xi)m.$$

Thus, given a point $X \in \mathbb{R}^3$, this definition satisfies the equation $mgX = (mgm)mX$ or:

$$mg(X, \xi) = g(mX, m\xi).$$

This means that posing the point X and mirroring the result is the same as applying the mirror pose to the mirror point mX.

Now let T_i be the color of vertex V_i. This determines the color $I(u) = T_i$ of pixel $u \in \mathbb{R}^2 \times \{1\}$ where

$$u \propto gV_i$$

is the perspective projection of the posed vertex V_i onto the image plane. If we mirror pixel u, we obtain:

$$mu \propto mgV_i = (mgm)(mV_i).$$

If we define the *mirror image* $(mI)(u) = I(mu)$ as the flipped copy of image I, and if I is the image of object (V, g, T), then mI is the image of the mirrored object (mV, mgm, T).

Next, we assume that the mesh is symmetric. This means that every vertex V_i has a symmetric counterpart $V_{m(i)}$, where m acts as a permutation on the vertex indices:

$$\forall i \in \{1, \dots, |V|\} : \quad mV_i = V_{m(i)}.$$

In fact this is not an arbitrary permutation: it consists of swaps, meaning that $m(m(i)) = i$, so that we have $m^{-1} = m$, just as before.

If is convenient to define the right action m on the vertex collection V as applying this permutation. Using matrix notation, if we interpret V as a $3 \times |V|$ matrix, this means that in the expression mV the operator m acts as the axis-flipping operator Eq. (8), and in the expression Vm it acts a $|V| \times |V|$ permutation matrix. With this convention, the mesh is symmetric if, and only if,

$$mV = Vm.$$

Mirror Equivariance.

To summarise, if I is the image of (V, g, T), them mI is the image of (mV, mgm, T) in general, and of (Vm, mgm, T) if the mesh is symmetric. Because the order of the mesh vertices is irrelevant for rendering an image, mI is also the image of object $(Vmm^{-1}, mgm, Tm^{-1}) = (V, mgm, Tm)$. In other words, we obtain the mirror image mI by viewing the same symmetric shape V under a mirror pose and texture. If the texture is also symmetric (i.e, $Tm = T$), then we only need to mirror the pose.

We conclude that, if the network makes the prediction $(V, g, T) = f(I)$ for image I, then it mus make the prediction $V, mgm, Tm) = f(mI)$ for the mirror image.

Front-to-Back Ambiguity.

Next, instead of a perspective camera, consider an orthographic projection

$$\Pi = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 0 \end{bmatrix}$$

so that vertex V_i projects to point $u = \Pi V_i$ in the image. The same consideration as before apply without changes. Additionally, consider the rotation matrix

$$r = \begin{bmatrix} -1 & & \\ & 1 & \\ & & -1 \end{bmatrix}.$$

which consists of a 180 degree rotation around the y axis. Given a symmetric mesh $mV = Vm$, consider objects (V, g, T) and $(Vm, rmgm, T)$. If we pose the second, we see that:

$$(rmgm)(Vm) = (rm)(gV) = \begin{bmatrix} 1 & & \\ & 1 & \\ & & -1 \end{bmatrix} (gV).$$

This means that the posed points are exactly the same, except for the fact that the z (not x!) axis is inverted. If we image them using the orographic projection Π, this axis is removed, so the image points are exactly the same in the two cases. Naturally, because the depth changes, the images are different; however, the masks are identical, causing the ambiguity.

Ton conclude, this means that the network $(V, g, T) = f(I)$ is likely to be confused between pose g and pose $rmgm$. Hence, we define $q\xi$ so that:

$$rmg(\dot{,}\xi)m = g(\cdot, q\xi).$$

Extension to Articulated Pose.

Fortunately, the discussion above generalizes to articulated poses. Posing uses in fact a chain of $SE(3)$ transformations of the type:

$$G_b(\xi)G_b(\xi)^{-1}, \quad \text{where} \;\; G_b(\xi) = G_{\pi(b)}(\xi) \cdot g_b(\xi_b).$$

The conjugate $mG_b(\xi)G_b(\xi)^{-1}m$ is found by interjecting factors $mm = 1$ in the chain:

$$m(G_{\pi(b)}(\xi) \cdot g_b(\xi_b))m = (mG_{\pi(b)}(\xi)m) \cdot (mg_b(\xi_b)m)$$

Expanding the last term we get that:

$$(mg_b(\xi)m)(X) = m \begin{bmatrix} R_{\xi_b} & \mathbf{J}_b \\ 0 & 1 \end{bmatrix} (mX) = mR_{\xi_b} mmX + m\mathbf{J}_b.$$

We assume that the bone structure is also symmetric, so that we can define for each bone b a symmetric counter-part $m(b)$ (another permutation) such that:

$$m\mathbf{J}_b = \mathbf{J}_{m(b)}.$$

With this, we have that:

$$(mg_b(\xi)m)(X) = g_{m(b)}(m\xi_{m(b)})(mX) = g_{m(b)}((m\xi m)_b)(mX).$$

Likewise, we assume the symmetry of the skinning weights and of the base vertices:

$$w_{ib} = w_{m(i),m(b)}, \qquad mV_i = V_{m(i)}.$$

With this:

$$\begin{aligned} mV_i(\xi) &= m\left(\sum_{b=1}^{B} w_{ib} G_b(\xi) G_b(\xi_0)^{-1}\right) mmV_i \\ &= \left(\sum_{b=1}^{B} w_{ib} (mG_b(\xi)m)(mG_b(\xi_0)m)^{-1}\right) V_{m(i)} \\ &= \left(\sum_{b=1}^{B} w_{m(i),m(b)} G_{m(b)}(m\xi m) G_{m(b)}(m\xi_0 m)^{-1}\right) V_{m(i)} \\ &= \left(\sum_{b=1}^{B} w_{m(i)b} G_b(m\xi m) G_b(m\xi_0 m)^{-1}\right) V_{m(i)} \\ &= V_{m(i)}(m\xi m). \end{aligned}$$

The equation $mV_i(\xi) = V_{m(i)}(m\xi m)$ is a generalization of the equation $mgV_i = (mgm)V_{m(i)}$ found before.

7.5 Implementation Details

7.5.1 Texture

We assume the texture of the objects is symmetric, and predict a texture map T for each object. Each vertex is associated with a fixed uv texture coordinate defined on the initial sphere, such that the texture is mapped from a circle in the texture map to both, left and right, sides of the sphere, as illustrated in Fig. 10. Mathematically, this is computed as:

$$u = \frac{2\arccos|x|}{\pi}\frac{z}{\sqrt{z^2+y^2}}, \qquad v = \frac{2\arccos|x|}{\pi}\frac{y}{\sqrt{z^2+y^2}}, \tag{9}$$

where x, y and z are the 3D coordinates of the vertices of a unit sphere.

7.5.2 Renderer Details

We use Pytorch3D differentiable renderer [80]. We use a perspective camera with a FOV of 25°, that is assumed to be stationary at $(0, 0, 10)$ looking at $(0, 0, 0)$. The initial sphere is a unit sphere centered at $(0, 0, 0)$. If the background image can be easily extracted from the video, *e.g.* videos with a static camera, we overlay the rendered mesh onto the extracted background, and compute the photometric loss described below on the entire image. Otherwise, we compute it only on the foreground pixels.

7.5.3 Geometric Smoothing Terms

In order to prevent the instance-specific shapes V to deviate too much from the category prior V_{tmpl}, we use the As-Rigid-As-Possible (ARAP) regularizer [81]:

$$L_{\text{ARAP}} = \sum_{i=1}^{|V|} \min_{R \in SO(3)} \sum_{j \in \mathcal{N}_i} \|(V_i - V_j) - R(V_{\text{tmpl},i} - V_{\text{tmpl},j})\|^2, \tag{10}$$

where $\mathcal{N}_k$ is the fan of vertices around vertex k. ARAP thus encourages triangle fans to deform rigidly. We further apply Laplacian and normal smoothing losses to the posed mesh V_ξ:

$$L_{\text{Lap}} = \sum_{i=1}^{|V_\xi|} \left\| V_{\xi i} - \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} V_{\xi j} \right\|^2, \quad L_{\text{nrm}} = \sum_{f \in \mathcal{F}} \sum_{f' \in \mathcal{N}_f} \left(1 - \frac{\mathbf{n}_{if} \cdot \mathbf{n}_{if'}}{\|\mathbf{n}_{if}\| \|\mathbf{n}_{if'}\|} \right), \tag{11}$$

where $\mathbf{n}_{if}$ is the normal of face f in V_ξ and $\mathcal{N}_f$ are the faces adjacent to f in the mesh. We sum the three geometric losses in the shape regularizer $L_{\text{smooth}}(V_{\text{tmpl}}, V, V_\xi) = \lambda_{\text{ARAP}} L_{\text{ARAP}} + \lambda_{\text{Lap}} L_{\text{Lap}} + \lambda_{\text{nrm}} L_{\text{nrm}}$.

7.5.4 Network Architectures

All the network architectures are described in in Tables 5 and 6. Abbreviations of the components are defined as follows:

- Conv$(c_{in}, c_{out}, k, s, p)$: 2D convolution with c_{in} input channels, c_{out} output channels, kernel size k, stride s and padding p;
- Deconv$(c_{in}, c_{out}, k, s, p)$: 2D deconvolution with c_{in} input channels, c_{out} output channels, kernel size k, stride s and padding p;
- Upsample(s): 2D nearest-neighbor upsampling with a scale factor s;
- GN(n): group normalization [84] with n groups;
- LReLU(p): leaky ReLU [85] with a slope p;

7.5.5 Training Details

All hyper-parameters are specified in Table 7. The model is trained for 20 epochs, which takes three days on one NVIDIA RTX-6000 GPU. For each epoch, we iterate through the training set by densely sampling 4 short sequences in a batch for each iteration. Each sequence contains 8 consecutive frames.

We train the model is three phases. (1) In the first phase, the goal is to learn a category template shape as well as rough as rough rigid poses of the objects. Therefore, we train the basic model with no instance-specific deformation, no bone articulation and no texture loss for the first epochs. (2) During the second phase, once the initial shape and rigid pose is learned, we activate the pose rectification loss L_{pose} rectify incorrect front/back poses, for one epoch. (3) Finally, we train the full model with instance deformation, skinning model with the bones instantiated, as well as texture loss for another 13 epochs. Furthermore, we gradually decay the weight of the Laplacian and normal smoothness regularizers by 0.7 every epoch from 1 to the 0.05, to allow the model to learn the geometric details in a coarse-to-fine manner.

To better enforce temporal consistency, we randomly replace both the instance-specific rest-pose shape and texture with their sequence averages with a 50% probability.

We use Adam optimizers with a learning rate of 0.0001 for the networks and 0.01 for the trainable category template shape parameters ΔV_{tmpl}. The learning rates are decayed by a factor of 0.7 after each epoch starting after 8 epochs.

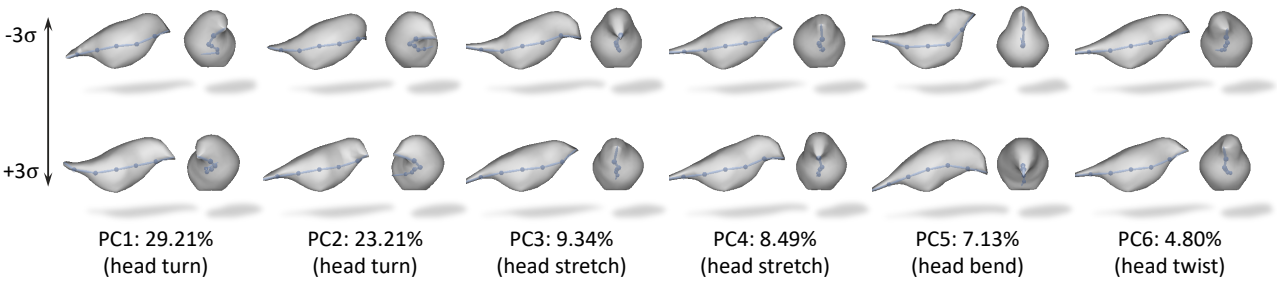


Fig. 11: **PCA on Learned Articulations.** Our model learns meaningful articulations with the bone-based skinning model, including typical head and tail movements of birds.

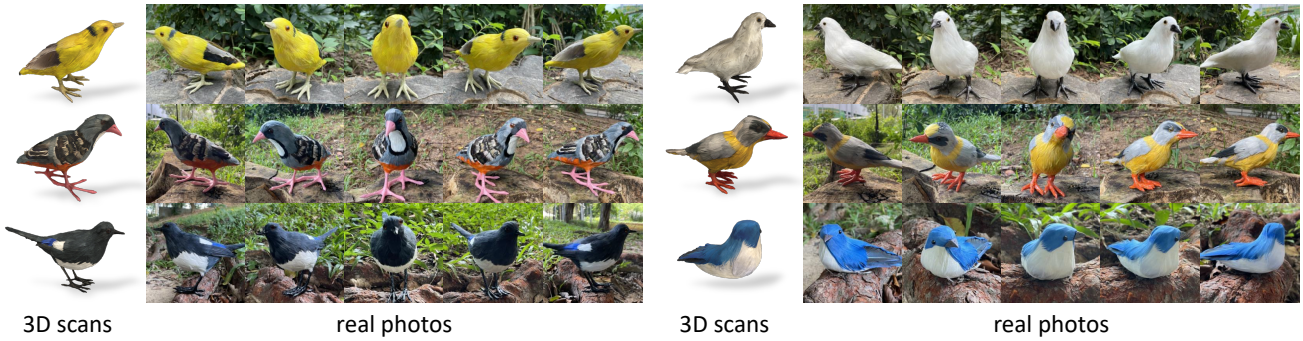


Fig. 12: **More examples of the 3D Toy Bird Dataset.**

Technical Details for SOTA Baselines.

We used the open-sourced implementations and pre-trained models of CMR [19], U-CMR [23], UMR [28]. We used a code and models of VMR [20] provided by the authors upon request. We finetuned their published models pre-trained on CUB dataset by training them on our bird video dataset. In the case of UMR, we also trained a version from scratch using only our birds dataset with SCOPS model pre-trained on CUB dataset.

We also trained all the baselines from scratch on our birds dataset without the additional supervisions that they need to use by default. We initialized CMR, U-CMR and VMR with a sphere base shape instead of their template shapes.

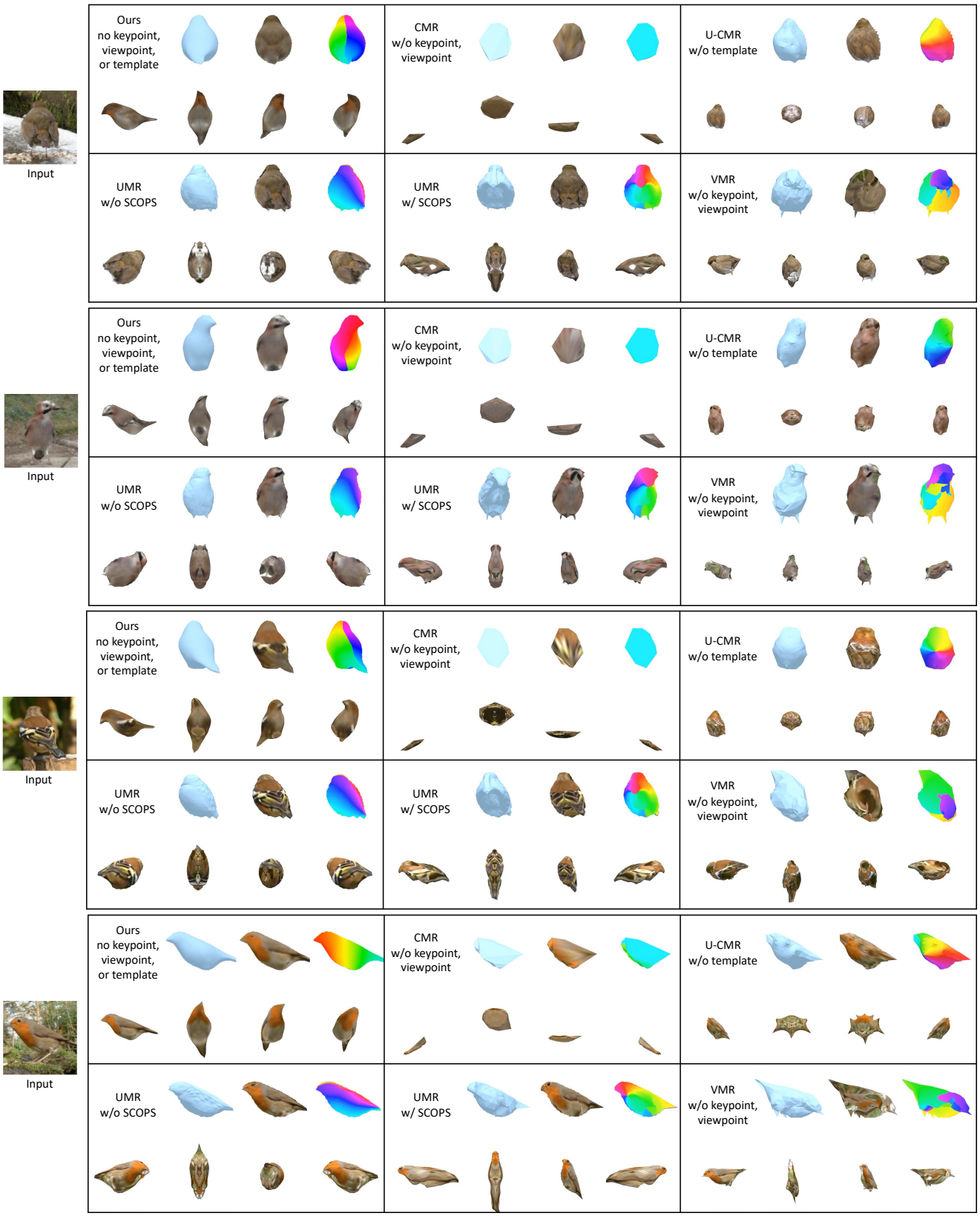


Fig. 13: **Comparisons against Existing Methods.** Without additional geometric supervision, SOTA methods fail to learn correct poses and hence produces poor shape reconstructions, whereas our model learns more plausible 3D shapes and accurate poses.

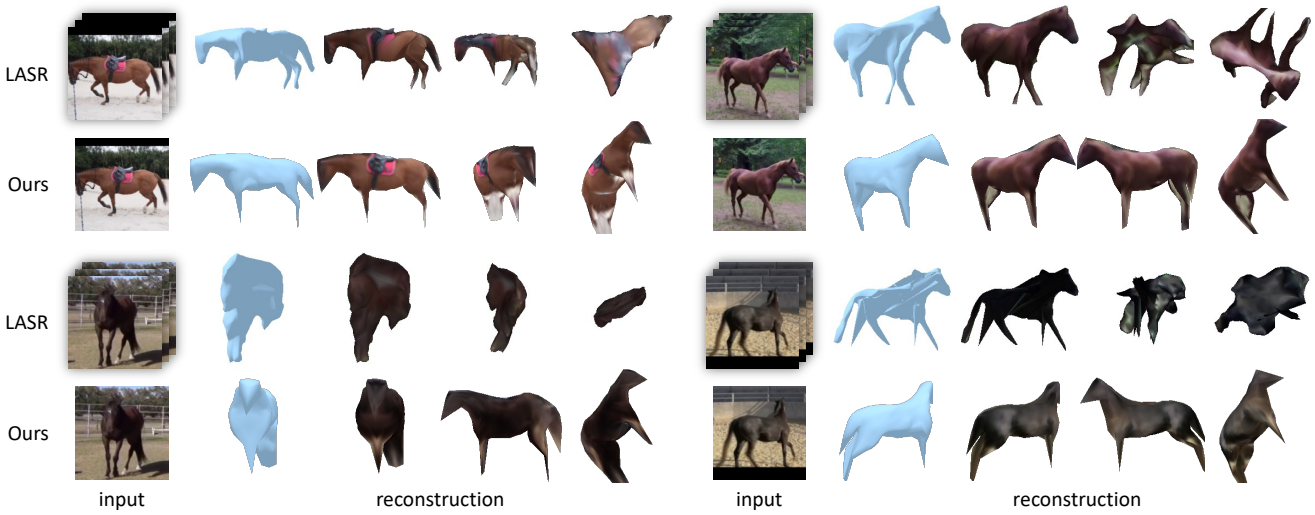


Fig. 14: **Additional qualitative comparison with LASR [34].** The shape produced by LASR is distorted and does not resemble the actual shape of a horse. Since our method trains on multiple sequences it learns consistent shapes of horses.

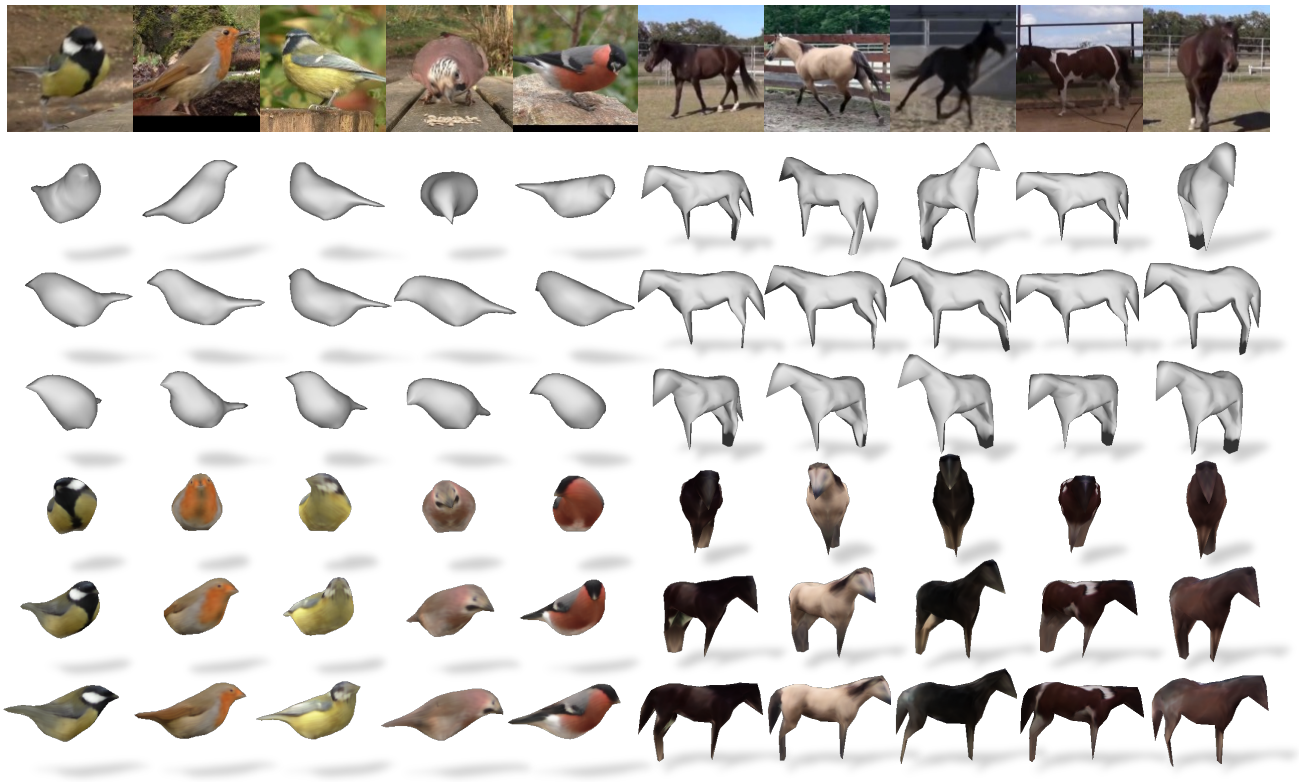


Fig. 15: **Additional Reconstruction Results.** We show the reconstructed objects from various viewpoints.



Fig. 16: **Texture Swapping and Animation.** Top: since our model learns a canonical representation for all objects, we can easily swap the texture across different instances. Bottom: we can also easily animate the 3D birds using our learned articulation model.

7 Summary

In this thesis, we have demonstrated an unsupervised framework for learning 3D representations of deformable object categories from casually recorded images and videos available online. At the core of this approach is a Photo-Geometric Autoencoding pipeline that exploits symmetries and regularities of the world, allowing physically-based disentangled 3D representations (including 3D shape, surface material, object motion, environment lighting and camera viewpoints) to automatically emerge from raw pixels through a simple reconstruction objective, without relying on explicit supervision for training.

In Chapter 3, we show that this approach can learn high-fidelity symmetric 3D faces from only a single-view collection of face images. In Chapter 4, we extend this framework for learning rotationally symmetric 3D vases as well as recovering shiny surface materials, also from a single-view image collection. In Chapter 5, we demonstrate that it is also possible to learn shiny material decomposition of general objects, given a single-view image collection with coarse geometry estimates. Finally, in Chapter 6, we extend the framework for learning more general, articulated 3D objects from collections of casually recorded monocular videos available online. Table 7.1 compares the key assumptions, outputs and types of data being used in these proposed methods. In particular, for shiny object categories, such as vases, we use a more expressive lighting representation (*e.g.* environment maps) and a

Table 7.1: Summary of Proposed Methods.

	Supervision					Output				
Method	🧊	✂	📹	⏩	Symmetry	3D Rep.	View	Material	Light	Data
Unsup3D (Chapter 3)					Bilateral	Depth	✓	Lambertian	Directional	Faces, Cars
SoRDerender (Chapter 4)		✓			Rotational	Mesh	✓	Phong	Env. Map	Vases
Derender (Chapter 5)	(✓)[1]					Normal		Phong	Directional	Faces, CO3D, Other
DOVE (Chapter 6)		✓	✓	✓	Bilateral	Art. Mesh	✓			Birds, Horses

Annotations: 🧊 *3D ground-truth,* ✂ *object mask,* 📹 *video,* ⏩ *optical flow,* [1] *coarse normal estimates, Art. Mesh: articulated mesh, Env. Map: environment map, CO3D: Reizenstein et al. 2021.*

Phong illumination model with shininess parameters. For object categories that exhibit less specular effects like cat faces, we restrict our model to a dominant directional light with Lambertian surfaces.

With the recent advances in differentiable rendering, geometric reasoning and representation learning can finally meet. It is now possible to learn physically-grounded 3D object priors from large collections of raw images, rather than treating them simply as 2D patterns. Unsupervised learning of 3D objects, as an emerging area, has thus been attracting growing interest in the community. Although promising results have been made in the past few years, it still remains unclear how we can learn general dynamic 3D objects and scenes in the wild. The quality of the reconstruction results our current models produce is still far from satisfactory for many real world AR and VFX applications. Integrating with powerful neural representations that have been developed recently, such as Neural Radiance Fields (NeRFs) [Mildenhall et al. 2020], could potentially lead to higher-fidelity reconstructions. We shall also see whether symmetries and regularities will automatically emerge along with the disentangled 3D representations even without manually injecting any inductive biases, as the scale of training continues to grow.

References

Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas (2018). "Learning Representations and Generative Models for 3D Point Clouds". In: *Proceedings of the International Conference on Machine Learning (ICML)*.

Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski (2009). "Building Rome in a Day". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Ijaz Akhter, Yaser Sheikh, Sohaib Khan, and Takeo Kanade (2008). "Nonrigid Structure from Motion in Trajectory Space". In: *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Neil Alldrin, Todd Zickler, and David Kriegman (2008). "Photometric Stereo with Non-Parametric and Spatially-Varying Reflectance". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel (2021). "Deep ViT Features as Dense Visual Descriptors". In: *arXiv preprint arXiv:2112.05814*.

Jonathan T. Barron and Jitendra Malik (2015). "Shape, Illumination, and Reflectance from Shading". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Harry G. Barrow and Jay M. Tenenbaum (1978). "Recovering Intrinsic Scene Characteristics from Images". In: *Computer Vision Systems*.

Bruce G. Baumgart (1974). "Geometric Modeling for Computer Vision". PhD thesis. Stanford University.

Sai Bi, Zexiang Xu, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi (2020a). "Deep Reflectance Volumes: Relightable Reconstructions from Multi-View Photometric Images". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Sai Bi, Zexiang Xu, Kalyan Sunkavalli, David Kriegman, and Ravi Ramamoorthi (2020b). "Deep 3D Capture: Geometry and Reflectance from Sparse Multi-View Images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black (2016). "Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Mark Boss, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P.A. Lensch (2021). "NeRD: Neural Reflectance Decomposition from Image Collections". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Mark Boss, Varun Jampani, Kihwan Kim, Hendrik P.A. Lensch, and Jan Kautz (2020). "Two-shot Spatially-varying BRDF and Shape Estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Nicholas Boyle (1992). *Goethe: The Poet and the Age*. Volume 1. Oxford University Press, p. 339.

Derek Bradley, Wolfgang Heidrich, Tiberiu Popa, and Alla Sheffer (2010). "High Resolution Passive Facial Performance Capture". In: *Proceedings of the ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques (SIGGRAPH)*.

Christoph Bregler, Aaron Hertzmann, and Henning Biermann (2000). "Recovering Non-Rigid 3D Shape from Image Streams". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin (2021). "Emerging Properties in Self-Supervised Vision Transformers". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Thomas J. Cashman and Andrew W. Fitzgibbon (2012). "What Shape are Dolphins? Building 3D Morphable Models from 2D Images". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein (2021). "pi-GAN: Periodic Implicit Generative Adversarial Networks for 3D-Aware Image Synthesis". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein (2022). "Efficient Geometry-aware 3D Generative Adversarial Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu (2015). "ShapeNet: An Information-Rich 3D Model Repository". In: *arXiv preprint arXiv:1512.03012*.

Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng (2016). "Single-Image Depth Perception in the Wild". In: *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Wenzheng Chen, Huan Ling, Jun Gao, Edward Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler (2019). "Learning to Predict 3D Objects with an

Interpolation-based Differentiable Renderer". In: *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Wenzheng Chen, Joey Litalien, Jun Gao, Zian Wang, Clement Fuji Tsang, Sameh Khamis, Or Litany, and Sanja Fidler (2021). "DIB-R++: Learning to Predict Lighting and Material with a Hybrid Differentiable Renderer". In: *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Zhiqin Chen and Hao Zhang (2019). "Learning Implicit Fields for Generative Shape Modeling". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese (2016). "3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

E. North Coleman and Ramesh Jain (1982). "Obtaining 3-Dimensional Shape of Textured and Specular Surfaces Using Four-Source Photometry". In: *Computer Graphics and Image Processing*.

Yuchao Dai, Hongdong Li, and Mingyi He (2012). "A Simple Prior-free Method for Non-Rigid Structure-from-Motion Factorization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

M. Daum and G. Dudek (1998). "On 3-D Surface Reconstruction Using Shape from Shadows". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Edilson de Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun (2008). "Performance Capture from Sparse Multi-view Video". In: *ACM Transactions on Graphics (TOG)*.

Paul Debevec (2012). "The Light Stages and Their Applications to Photoreal Digital Actors". In: *Proceedings of the ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques in Asia (SIGGRAPH Asia)*.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009). "ImageNet: A large-scale hierarchical image database". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Valentin Deschaintre, Miika Aittala, Fredo Durand, George Drettakis, and Adrien Bousseau (2018). "Single-Image SVBRDF Capture with a Rendering-Aware Deep Network". In: *Proceedings of the ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques (SIGGRAPH)*.

David Eigen, Christian Puhrsch, and Rob Fergus (2014). "Depth Map Prediction from a Single Image Using a Multi-scale Deep Network". In: *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman (2022). "StyleSDF: High-Resolution 3D-Consistent Image and Geometry Generation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Haoqiang Fan, Hao Su, and Leonidas Guibas (2017). "A Point Set Generation Network for 3D Object Reconstruction from a Single Image". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Olivier Faugeras (1993). *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press.

Olivier Faugeras and Quang-Tuan Luong (2001). *The Geometry of Multiple Images*. MIT Press.

Paolo Favaro and Stefano Soatto (2007). *3-D Shape Estimation and Image Restoration: Exploiting Defocus and Motion-Blur*. Springer London.

Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart (2021). "Learning an Animatable Detailed 3D Face Model from In-the-Wild Images". In: *Proceedings of the ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques (SIGGRAPH)*.

Alexandre R. J. François, Gérard G. Medioni, and Roman Waupotitsch (2003). "Mirror Symmetry ⇒ 2-View Stereo Geometry". In: *Image Vision Computing*.

Yuan Gao and Alan L. Yuille (2017). "Exploiting Symmetry and/or Manhattan Properties for 3D Object Structure Estimation from Single and Multiple Images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Elena Garces, Adolfo Munoz, Jorge Lopez-Moreno, and Diego Gutierrez (2012). "Intrinsic Images by Clustering". In: *Computer Graphics Forum*, pp. 1415–1424.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun (2013). "Vision meets Robotics: The KITTI Dataset". In: *International Journal of Robotics Research (IJRR)*.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel (2019). "ImageNet-trained CNNs are Biased Towards Texture; Increasing Shape Bias Improves Accuracy and Robustness". In: *Proceedings of the International Conference on Machine Learning (ICML)*.

Abhijeet Ghosh, Graham Fyffe, Borom Tunwattanapong, Jay Busch, Xueming Yu, and Paul Debevec (2011). "Multiview Face Capture using Polarized Spherical Gradient Illumination". In: *Proceedings of the ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques in Asia (SIGGRAPH Asia)*.

Rohit Girdhar, David Fouhey, Mikel Rodriguez, and Abhinav Gupta (2016). "Learning a Predictable and Generative Vector Representation for Objects". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Georgia Gkioxari, Jitendra Malik, and Justin Johnson (2019). "Mesh R-CNN". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow (2017). "Unsupervised Monocular Depth Estimation with Left-Right Consistency". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shubham Goel, Angjoo Kanazawa, and Jitendra Malik (2020). "Shape and Viewpoints without Keypoints". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Dan B Goldman, Brian Curless, Aaron Hertzmann, and Steven M Seitz (2009). "Shape and Spatially-Varying BRDFs from Photometric Stereo". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry (2018). "AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt (2022). "StyleNeRF: A Style-based 3D Aware Generator for High-resolution Image Synthesis". In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Richard Hartley and Andrew Zisserman (2004). *Multiple View Geometry in Computer Vision*. Second. Cambridge University Press, ISBN: 0521540518.

Carlos Hernández, George Vogiatzis, and Roberto Cipolla (2008). "Multiview Photometric Stereo". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Berthold K. P. Horn and Michael J. Brooks (1989). *Shape from Shading*. Cambridge Massachusetts: MIT Press.

Berthold KP Horn (1975). "Obtaining Shape from Shading Information". In: *The Psychology of Computer Vision*.

Andrew Hou, Ze Zhang, Michel Sarkis, Ning Bi, Yiying Tong, and Xiaoming Liu (2021). "Towards High Fidelity Face Relighting with Realistic Shadows". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wei-Chih Hung, Varun Jampani, Sifei Liu, Pavlo Molchanov, Ming-Hsuan Yang, and Jan Kautz (2019). "SCOPS: Self-Supervised Co-Part Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

T. H. Huxley (1869). "Nature: Aphorisms by Goethe". In: *Nature*.

Yasamin Jafarian and Hyun Soo Park (2021). "Learning High Fidelity Depths of Dressed Humans by Watching Social Media Dance Videos". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, Merlin Nimier-David, Delio Vicini, Tizian Zeltner, Baptiste Nicolet, Miguel Crespo, Vincent Leroy, and Ziyi Zhang (2022). *Mitsuba 3 renderer*. Version 3.0.1. https://mitsuba-renderer.org.

Michael Janner, Jiajun Wu, Tejas Kulkarni, Ilker Yildirim, and Joshua B Tenenbaum (2017). "Self-Supervised Intrinsic Image Decomposition". In: *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Iain G. Johnston, Kamaludin Dingle, Sam F. Greenbury, Chico Q. Camargo, Jonathan P. K. Doye, Sebastian E. Ahnert, and Ard A. Louis (2022). "Symmetry and

Simplicity Spontaneously Emerge from the Algorithmic Nature of Evolution". In: *Proceedings of the National Academy of Sciences (PNAS)* 119.11.

Hanbyul Joo, Tomas Simon, and Yaser Sheikh (2018). "Total Capture: A 3D Deformation Model for Tracking Faces, Hands, and Bodies". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Yoshihiro Kanamori and Yuki Endo (2018). "Relighting Humans: Occlusion-Aware Inverse Rendering for Full-Body Human Images". In: *ACM Transactions on Graphics (TOG).*

Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik (2018a). "End-to-end Recovery of Human Shape and Pose". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Angjoo Kanazawa, Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik (2018b). "Learning Category-Specific Mesh Reconstruction from Image Collections". In: *Proceedings of the European Conference on Computer Vision (ECCV).*

Abhishek Kar, Shubham Tulsiani, João Carreira, and Jitendra Malik (2015). "Category-Specific Object Reconstruction from a Single Image". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada (2018). "Neural 3D Mesh Renderer". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Roland Kehl and Luc Van Gool (2006). "Markerless Tracking of Complex Human Motions from Multiple Views". In: *Computer Vision and Image Understanding (CVIU).*

Filippos Kokkinos and Iasonas Kokkinos (2021). "To The Point: Correspondence-Driven Monocular 3D Category Reconstruction". In: *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS).*

Nilesh Kulkarni, Abhinav Gupta, David F Fouhey, and Shubham Tulsiani (2020). "Articulation-aware Canonical Surface Mapping". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Nilesh Kulkarni, Abhinav Gupta, and Shubham Tulsiani (2019). "Canonical Surface Mapping via Geometric Cycle Consistency". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV).*

Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum (2015). "Deep Convolutional Inverse Graphics Network". In: *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS).*

Pierre-Yves Laffont, Adrien Bousseau, Sylvain Paris, Frédo Durand, and George Drettakis (2012). "Coherent Intrinsic Images from Photo Collections". In: *ACM Transactions on Graphics (TOG).*

Manuel Lagunas, Xin Sun, Jimei Yang, Ruben Villegas, Jianming Zhang, Zhixin Shu, Belen Masia, and Diego Gutierrez (2021). "Single-image Full-body Human Relighting". In: *Eurographics Symposium on Rendering (EGSR).*

Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab (2016). "Deeper Depth Prediction with Fully Convolutional Residual Networks". In: *Proceedings of the International Conference on 3D Vision (3DV).*

Edwin H. Land and John J. McCann (1971). "Lightness and Retinex Theory". In: *Journal of the Optical Society of America* 61.1, pp. 1–11.

Aldo Laurentini (1994). "The Visual Hull Concept for Silhouette-based Image Understanding". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI).*

Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero (2017). "Learning a Model of Facial Shape and Expression from 4D Scans". In: *Proceedings of the ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques in Asia (SIGGRAPH Asia).*

Xiao Li, Yue Dong, Pieter Peers, and Xin Tong (2017). "Modeling Surface Appearance from a Single Photograph using Self-augmented Convolutional Neural Networks". In: *Proceedings of the ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques (SIGGRAPH).*

Xueting Li, Sifei Liu, Shalini De Mello, Kihwan Kim, Xiaolong Wang, Ming-Hsuan Yang, and Jan Kautz (2020a). "Online Adaptation for Consistent Mesh Reconstruction in the Wild". In: *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS).*

Xueting Li, Sifei Liu, Kihwan Kim, Shalini De Mello, Varun Jampani, Ming-Hsuan Yang, and Jan Kautz (2020b). "Self-supervised Single-view 3D Reconstruction via Semantic Consistency". In: *Proceedings of the European Conference on Computer Vision (ECCV).*

Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T. Freeman (2019). "Learning the Depths of Moving People by Watching Frozen People". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Zhengqi Li and Noah Snavely (2018a). "CGIntrinsics: Better Intrinsic Image Decomposition through Physically-Based Rendering". In: *Proceedings of the European Conference on Computer Vision (ECCV).*

Zhengqi Li and Noah Snavely (2018b). "Learning Intrinsic Image Decomposition from Watching the World". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker (2020). "Inverse Rendering for Complex Indoor Scenes: Shape, Spatially-Varying Lighting and SVBRDF from a Single Image". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Zhengqin Li, Jia Shi, Sai Bi, Rui Zhu, Kalyan Sunkavalli, Milos Hasan, Zexiang Xu, Ravi Ramamoorthi, and Manmohan Chandraker (2022). "Physically-Based Editing of

Indoor Scene Lighting from a Single Image". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker (2018a). "Materials for Masses: SVBRDF Acquisition with a Single Mobile Phone Image". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker (2018b). "Learning to Reconstruct Shape and Spatially-Varying Reflectance from a Single Image". In: *Proceedings of the ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques in Asia (SIGGRAPH Asia)*.

Zhengqin Li, Ting-Wei Yu, Shen Sang, Sarah Wang, Meng Song, Yuhan Liu, Yu-Ying Yeh, Rui Zhu, Nitesh Gundavarapu, Jia Shi, Sai Bi, Hong-Xing Yu, Zexiang Xu, Kalyan Sunkavalli, Milos Hasan, Ravi Ramamoorthi, and Manmohan Chandraker (2021). "OpenRooms: An Open Framework for Photorealistic Indoor Scene Datasets". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Joseph J. Lim, Hamed Pirsiavash, and Antonio Torralba (2013). "Parsing IKEA Objects: Fine Pose Estimation". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Andrew Liu, Shiry Ginosar, Tinghui Zhou, Alexei A. Efros, and Noah Snavely (2020). "Learning to Factorize and Relight a City". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Guilin Liu, Duygu Ceylan, Ersin Yumer, Jimei Yang, and Jyh-Ming Lien (2017). "Material Editing Using a Physically Based Rendering Network". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Shichen Liu, Tianye Li, Weikai Chen, and Hao Li (2019). "Soft Rasterizer: A Differentiable Renderer for Image-based 3D Reasoning". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Yunfei Liu, Yu Li, Shaodi You, and Feng Lu (2020). "Unsupervised Learning for Intrinsic Image Decomposition from a Single Image". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Anthony Lobay and D. A. Forsyth (2006). "Shape from Texture without Boundaries". In: *International Journal of Computer Vision (IJCV)*.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black (2015). "SMPL: A skinned Multi-Person Linear Model". In: *ACM Transactions on Graphics (TOG)*.

Matthew M. Loper and Michael J. Black (2014). "OpenDR: An Approximate Differentiable Renderer". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Wei-Chiu Ma, Hang Chu, Bolei Zhou, Raquel Urtasun, and Antonio Torralba (2018). "Single Image Intrinsic Decomposition without a Single Intrinsic Image". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger (2019). "Occupancy Networks: Learning 3D Reconstruction in Function Space". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng (2020). "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis". In: *Proceedings of the European Conference on Computer Vision (ECCV).*

Dipti P. Mukherjee, Andrew Zisserman, and J. Michael Brady (1995). "Shape from Symmetry – Detecting and Exploiting Symmetry in Affine Images". In: *Philosophical Transactions of the Royal Society of London* 351, pp. 77–106.

Takuya Narihira, Michael Maire, and Stella X. Yu (2015). "Direct Intrinsics: Learning Albedo-Shading Decomposition by Convolutional Regression". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV).*

Thomas Nestmeyer, Jean-François Lalonde, Iain Matthews, and Andreas Lehrmann (2020). "Learning Physics-guided Face Relighting under Directional Light". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5124–5133.

Natalia Neverova, David Novotný, and Andrea Vedaldi (2020). "Continuous Surface Embeddings". In: *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS).*

Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang (2019). "HoloGAN: Unsupervised Learning of 3D Representations from Natural Images". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV).*

Thu Nguyen-Phuoc, Christian Richardt, Long Mai, Yong-Liang Yang, and Niloy Mitra (2020). "BlockGAN: Learning 3D Object-aware Scene Representations from Unlabelled Images". In: *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS).*

Michael Niemeyer and Andreas Geiger (2021). "GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger (2020). "Differentiable Volumetric Rendering: Learning Implicit 3D Representations without 3D Supervision". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

David Novotny, Diane Larlus, and Andrea Vedaldi (2017). "Learning 3D Object Categories by Looking Around Them". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV).*

David Novotny, Nikhila Ravi, Benjamin Graham, Natalia Neverova, and Andrea Vedaldi (2019). "C3DPO: Canonical 3D Pose Networks for Non-Rigid Structure From Motion". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV).*

Rohit Pandey, Sergio Orts-Escolano, Chloe LeGendre, Christian Haene, Sofien Bouaziz, Christoph Rhemann, Paul Debevec, and Sean Fanello (2021). "Total Relighting: Learning to Relight Portraits for Background Replacement". In: *ACM Transactions on Graphics (TOG)*.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove (June 2019). "DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman (2015). "Deep Face Recognition". In: *Proceedings of the British Machine Vision Conference (BMVC)*.

Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter (2009). "A 3D Face Model for Pose and Illumination Invariant Face Recognition". In: *Proceedings of the IEEE International Conference on Advanced Video and Signal-Based Surveillances*.

Bui Tuong Phong (1975). "Illumination for Computer Generated Pictures". In: *Communications of the ACM*.

Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer (2020). "D-NeRF: Neural Radiance Fields for Dynamic Scenes". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari (2020). "Accelerating 3D Deep Learning with PyTorch3D". In: *arXiv:2007.08501*.

Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny (2021). "Common Objects in 3D: Large-Scale Learning and Evaluation of Real-life 3D Category Reconstruction". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Carsten Rother, Martin Kiefel, Lumin Zhang, Bernhard Schölkopf, and Peter Gehler (2011). "Recovering Intrinsic Images with a Global Sparsity Prior on Reflectance". In: *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li (2019). "PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Shen Sang and Manmohan Chandraker (2020). "Single-Shot Neural Relighting and SVBRDF Estimation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Johannes Lutz Schönberger and Jan-Michael Frahm (2016). "Structure-from-Motion Revisited". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger (2020). "GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis". In: *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Soumyadip Sengupta, Angjoo Kanazawa, Carlos D. Castillo, and David Jacobs (2018). "SfSNet: Learning Shape, Refectance and Illuminance of Faces in the Wild". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Li Shen, Ping Tan, and Stephen Lin (2008). "Intrinsic Image Decomposition with Non-Local Texture Cues". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Li Shen and Chuohao Yeo (2011). "Intrinsic Images Decomposition Using a Local and Global Sparse Representation of Reflectance". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Jian Shi, Yue Dong, Hao Su, and Stella X. Yu (2017). "Learning Non-Lambertian Object Intrinsics across ShapeNet Categories". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Zhixin Shu, Ersin Yumer, Sunil Hadap, Kalyan Sunkavalli, Eli Shechtman, and Dimitris Samaras (2017). "Neural Face Editing with Intrinsic Image Disentangling". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Sudipta N. Sinha, Krishnan Ramnath, and Richard Szeliski (2012). "Detecting and Reconstructing 3D Mirror Symmetric Objects". In: *Proceedings of the European Conference on Computer Vision (ECCV).*

Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein (2019). "Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations". In: *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS).*

Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron (2021). "NeRV: Neural Reflectance and Visibility Fields for Relighting and View Synthesis". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Tiancheng Sun, Jonathan T Barron, Yun-Ta Tsai, Zexiang Xu, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul E Debevec, and Ravi Ramamoorthi (2019). "Single Image Portrait Relighting". In: *ACM Transactions on Graphics (TOG).*

Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman (2018). "Pix3D: Dataset and Methods for Single-Image 3D Shape Modeling". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Boaz J. Super and Alan C. Bovik (1995). "Shape from Texture Using Local Spectral Moments". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI).*

Daichi Tajima, Yoshihiro Kanamori, and Yuki Endo (2021). "Relighting Humans in the Wild: Monocular Full-Body Human Relighting with Domain Adaptation". In: *Computer Graphics Forum.* Wiley Online Library, pp. 205–216.

Feitong Tan, Sean Fanello, Abhimitra Meka, Sergio Orts-Escolano, Danhang Tang, Rohit Pandey, Jonathan Taylor, Ping Tan, and Yinda Zhang (2022). "VoLux-GAN: A Generative Model for 3D Face Synthesis with HDRI Relighting". In: *Proceedings of the ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques (SIGGRAPH)*.

L. V. Tarasov (1986). *This Amazingly Symmetrical World*. Mir Publishers.

Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox (2017). "Octree Generating Networks: Efficient Convolutional Architectures for High-resolution 3D Outputs". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Ayush Tewari, Michael Zollhöfer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Pérez, and Christian Theobalt (2017). "MoFA: Model-based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Sebastian Thrun and Ben Wegbreit (2005). "Shape From Symmetry". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Shubham Tulsiani, Nilesh Kulkarni, and Abhinav Gupta (2020). "Implicit Mesh Reconstruction from Unannotated Image Collections". In: *arXiv preprint arXiv:2007.08504*.

Gül Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid (2018). "BodyNet: Volumetric Inference of 3D Human Body Shapes". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Sara Vicente, João Carreira, Lourdes Agapito, and Jorge Batista (2014). "Reconstructing PASCAL VOC". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang (2018). "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Zian Wang, Jonah Philion, Sanja Fidler, and Jan Kautz (2021). "Learning Indoor Inverse Rendering with 3D Spatially-Varying Lighting". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Andreas Wenger, Andrew Gardner, Chris Tchou, Jonas Unger, Tim Hawkins, and Paul Debevec (2005). "Performance Relighting and Reflectance Transformation with Time-Multiplexed Illumination". In: *Proceedings of the ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques (SIGGRAPH)*.

Hermann Weyl (1952). *Symmetry*. Princeton University Press.

Andrew P. Witkin (1981). "Recovering Surface Shape and Orientation from Texture". In: *Artificial Intelligence*.

Robert J. Woodham (1980). "Photometric Method For Determining Surface Orientation From Multiple Images". In: *Optical Engineering*.

Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T. Freeman, and Joshua B. Tenenbaum (2016). "Learning a Probabilistic Latent Space of Object

Shapes via 3D Generative-Adversarial Modeling". In: *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao (2015). "3D ShapeNets: A Deep Representation for Volumetric Shapes". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese (2014). "Beyond PASCAL: A Benchmark for 3D Object Detection in the Wild". In: *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*.

Jing Xiao, Jin-xiang Chai, and Takeo Kanade (2004). "A Closed-Form Solution to Non-Rigid Shape and Motion Recovery". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Gengshan Yang, Deqing Sun, Varun Jampani, Daniel Vlasic, Forrester Cole, Huiwen Chang, Deva Ramanan, William T. Freeman, and Ce Liu (2021a). "LASR: Learning Articulated Shape Reconstruction from a Monocular Video". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gengshan Yang, Deqing Sun, Varun Jampani, Daniel Vlasic, Forrester Cole, Ce Liu, and Deva Ramanan (2021b). "ViSER: Video-Specific Surface Embeddings for Articulated 3D Shape Reconstruction". In: *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Gengshan Yang, Minh Vo, Neverova Natalia, Deva Ramanan, Vedaldi Andrea, and Joo Hanbyul (2022). "BANMo: Building Animatable 3D Neural Models from Many Casual Videos". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian (2018). "FoldingNet: Point Cloud Auto-encoder via Deep Grid Deformation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chun-Han Yao, Wei-Chih Hung, Yuanzhen Li, Michael Rubinstein, Ming-Hsuan Yang, and Varun Jampani (2022). "LASSIE: Learning Articulated Shapes from Sparse Image Ensemble via 3D Part Discovery". In: *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman (2020). "Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance". In: *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Yufei Ye, Shubham Tulsiani, and Abhinav Gupta (2021). "Shelf-Supervised Mesh Prediction in the Wild". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yu-Ying Yeh, Zhengqin Li, Yannick Hold-Geoffroy, Rui Zhu, Zexiang Xu, Miloš Hašan, Kalyan Sunkavalli, and Manmohan Chandraker (2022). "PhotoScene: Photorealistic Material and Lighting Transfer for Indoor Scenes". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa (2021). "pixelNeRF: Neural Radiance Fields from One or Few Images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Ye Yu, Abhimetra Meka, Mohamed Elgharib, Hans-Peter Seidel, Christian Theobalt, and Will Smith (2020). "Self-supervised Outdoor Scene Relighting". In: *Proceedings of the European Conference on Computer Vision (ECCV).*

Kai Zhang, Fujun Luan, Zhengqi Li, and Noah Snavely (2022). "IRON: Inverse Rendering by Optimizing Neural SDFs and Materials from Photometric Images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely (2021). "PhySG: Inverse Rendering with Spherical Gaussians for Physics-based Material Editing and Relighting". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah (1999). "Shape-from-shading: a survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI).*

Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron (2021). "NeRFactor: Neural Factorization of Shape and Reflectance Under an Unknown Illumination". In: *Proceedings of the ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques in Asia (SIGGRAPH Asia).*

Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe (2017). "Unsupervised Learning of Depth and Ego-Motion from Video". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Rui Zhu, Zhengqin Li, Janarbek Matai, Fatih Porikli, and Manmohan Chandraker (2022). "IRISformer: Dense Vision Transformers for Single-Image Inverse Rendering in Indoor Scenes". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Appendices

A

Sources of Figures

- "Lion Photo 1" in Fig. 1.1: by Francesco, source from https://unsplash.com/photos/ZxNKxnR32Ng, under free-to-use Unsplash License.
- "Lion Photo 2" in Fig. 1.1: by jean wimmerlin, source from https://unsplash.com/photos/FC4GY9nQuu0, under free-to-use Unsplash License.
- "Lion Photo 3" in Fig. 1.1: by Jonathan Ridley, source from https://unsplash.com/photos/ZU3-BIjVuK8, under free-to-use Unsplash License.
- "Lion Photo 4" in Fig. 1.1: by Mika Brandt, source from https://unsplash.com/photos/UlipBbZpweg, under free-to-use Unsplash License.
- "Lion Mesh" in Fig. 1.1: by daniel.jibi, source from https://sketchfab.com/3d-models/lion-f94d39b60c5846148b63b523c8e85ed4, under free-to-use CC BY 4.0 License.
- "Material Samples" in Fig. 1.1: by ambientCG, source from https://ambientcg.com/list?type=Atlas,Decal,Material, under free-to-use CC0 License.
- "Running Lion Animation" in Fig. 1.1: by Slava Borovik and Marcel Hampel, source from https://www.behance.net/gallery/71805451/Animals-Run-Cycle-Animations, under free-to-use CC BY-NC-ND 4.0 License.
- "Light Bulb Icon" in Fig. 1.1: by King god, source from https://www.vecteezy.com/vector-art/2915286-light-bulb-vector-bulb-light-icon-bulb-light-vector, under free-to-use Vecteezy License.
- "Vitruvian Man" in Fig. 1.2: by Leonardo da Vinci, credit to Luc Viatour, source from https://commons.wikimedia.org/wiki/File:Da_Vinci_Vitruve_Luc_Viatour.jpg, under free-to-use Public Domain License.
- "Diagram comparing bilateral, radial, and spherical symmetry" in Fig. 1.2: by Charl Hutchings, source from https://commons.wikimedia.org/wiki/File:Diagram_comparing_bilateral,_radial,_and_spherical_symmetry.jpg, under free-to-use CC BY 4.0 License.

- "Kudu" in Fig. 1.2: by Matthew Mitchell, source from https://unsplash.com/photos/UXpGLd6qQg0, under free-to-use Unsplash License.
- "Tiger" in Fig. 1.2: by Kartik Iyer, source from https://unsplash.com/photos/tnebcEDLcQs, under free-to-use Unsplash License.
- "Throwing Pottery" in Fig. 1.2: by Courtney Cook, source from https://unsplash.com/photos/AZmiWbWMgzU, under free-to-use Unsplash License.
- "Spinnig Top" in Fig. 1.2: by Ash from Modern Afflatus, source from https://unsplash.com/photos/iiRQxPCDQ_Y, under free-to-use Unsplash License.
- "Taj Mahal" in Fig. 1.2: by Julian Yu, source from https://unsplash.com/photos/_WuPjE-MPHo, under free-to-use Unsplash License.
- "Forbidden City" in Fig. 1.2: by __ drz __, source from https://unsplash.com/photos/wXyf-U8HqiI, under free-to-use Unsplash License.
- "Eiffel Tower" in Fig. 1.2: by Filip Andrejevic, source from https://unsplash.com/photos/FJYS6vuf39k, under free-to-use Unsplash License.