Sparse-view 3D in the Wild

Jason Y. Zhang

CMU-RI-TR-24-09

May 2024

School of Computer Science
The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Thesis Committee

Deva Ramanan	(*co-chair*)
Shubham Tulsiani	(*co-chair*)
Martial Hebert	
William T. Freeman	Massachusetts Institute of Technology
Noah Snavely	Cornell Tech

Thesis proposal submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Robotics

Keywords: 3D Reconstruction, Pose Estimation, 3D Computer Vision

Abstract

Reconstructing 3D scenes and objects from images alone has been a long-standing goal in computer vision. We have seen tremendous progress in recent years, capable of producing near photorealistic renderings from any viewpoint. However, existing approaches generally rely on a large number of input images (typically 50-100) to compute camera poses and ensure view consistency. This constraint limits the applicability of these methods, as taking 100 high-quality images without motion blur can be burdensome for end users. To enable 3D reconstructions in unconstrained scenes, this thesis proposes techniques for sparse-view 3D, automatically estimating camera poses and reconstructing 3D objects in the wild from less than 10 images.

We start by exploring how implicit surfaces can be used to regularize 3D representations learned from sparse views. We demonstrate that our representation, which factors view-dependent specular effects from view-independent diffuse appearance, can robustly reconstruct 3D from as few as 4-8 images associated with noisy camera poses. However, acquiring this camera pose initialization in the first place is challenging. To address this, we propose an energy-based framework that predicts the probability distribution over relative camera rotations. These distributions are then composed into coherent sets of camera rotations given sparse image sets. We then show how leveraging a transformer-based architecture to scale our energy-based representation can effectively make use of more images. We find that additional image context allows our method to resolve ambiguities that arise from just two images. While top-down energy-based pose estimation can effectively handle pose ambiguity, it can be slow to sample poses and does not make use of level features that may provide useful cues for correspondence matching and geometric consistency. To address these issues, we propose to represent a camera as a bundle of rays passing from the camera center to the center of each image patch in 3D. We then train a diffusion-based denoising network to predict this representation. We find that this generic camera representation significantly improves pose accuracy.

Contents

Acknowledgements

This thesis could not have existed without the support of countless friends and family, of whom there are too many to list in this section. The two most important people who have made this thesis a possibility are my advisers Deva and Shubham. Deva's kindness was the first thing I had ever heard about him, and the reputation is well-deserved. From Deva, I learned to be compassionate. Even when R2 only read every other paragraph, Deva tells me to be calm and walk in the reviewer's shoes. He makes me think about where R2's misunderstanding really came from, what is the kernel of confusion that could possibly have lead to such a preposterous question. Shubham is the most brilliant person I have ever had the chance to work with. Optimization problems that I have worked on for weeks are dispatched in seconds on Shubham's whiteboard without hesitation. Shubham's always slightly cracked open door and infinite patience were extremeley helpful whenever I was stuck or had an idea. I feel blessed to have spent a PhD advised by two good people.

I am also grateful to the mentors who kicked off my research career. Anca sparked my sense of curiosity and taught me that research was about taking on challenges for which no one had the right answers. Angjoo taught me how to do research and conduct experiments. Her never-ending enthusiasm taught me that even the most challenging times need not be dull. Jitendra convinced me to not work on boring problems.

I would like to thank the members of my committee: Martial, Noah, and Bill for their infinite insights and tough questions. I also thank the mentorship of others that I have had the great fortune of working with: Hanbyul Joo, Andrea Vedaldi, and Panna Felsen among others. And I would like to thank Amy, a fantastic collaborator with an infinite enthusiasm to work on difficult problems.

My friends in Pittsburgh made this journey a blast. Shikhar and Sudeep have been like family to me. The support and friendship of many friends have made Pittsburgh a home: Peter, Alex, Pragna, Senthil, Kenny, Gaurav, and Nadine. My lab and officemates who have made research interesting and fruitful: Gengshan, Haithem, Martin, Achal, Aayush, Peiyun, Ravi, Neehar, Tarasha, ZQ, Andrew, Judy, Hanzhe, Z, Bharath, Yehonathan, and Victoria just to name a few.

Finally and most of all, I would like to thank my parents and my wife Helen for making this adventure worth it.

List of Figures

List of Tables

Chapter 1

Introduction

The goal of estimating 3D from 2D images is one of the central challenges in computer vision. Such technology has the power to reconstruct the world around us, enable robots and autonomous agents to interact freely with the world, and democratize the creation of 3D assets for creative applications such as CGI for movies or gaming. For decades, significant progress has been made in terms of better 3D representations and camera pose estimation pipelines to facilitate these.

In recent years, NeRF [89] has been one of the most significant works in terms of achieving a representation that generates near-photorealistic renderings from any viewpoint. To do this, NeRF represents the scene volumetric using a neural network that can be optimized using a differentiable raycasting operation. In some ways, NeRF and numerous follow-up works have achieved one of the holy grails in computer vision: a method that takes a set of captured images and outputs a neural plenoptic function [3]. Given any 3D position and viewing direction in the scene, we can now synthesize how the world would appear from that viewpoint. However, these existing methods that achieve impressive novel-view synthesis capabilities have a key limitation: they require an immense number of images, typically on the order of 50-100. The requirement of so many images is necessary for two reasons. First, existing camera pose estimation pipelines that rely on structure-from-motion require densely sampled views to predict the cameras that are used as input. Second, densely sampled views are necessary to effectively constrain the volumetric field, especially those conditioned on viewing directions.

The constraint of 50-100 images is prohibitive to developing general-purpose 3D algorithms that can be easily deployed in unconstrained setups. Typically, a user dedicated to taking pictures of a particularly interesting object may be willing to take several images, perhaps even 10, but is unlikely to take a hundred. Or consider the millions of listings posted on online marketplaces such as Craigslist or Amazon. Such marketplaces can be thought of as a rich source of multi-view data [25], where sellers typically take several images of an object they wish to sell (see Fig. 1.1). The scale and diversity of such data dwarf any existing 3D or multiview dataset, and being

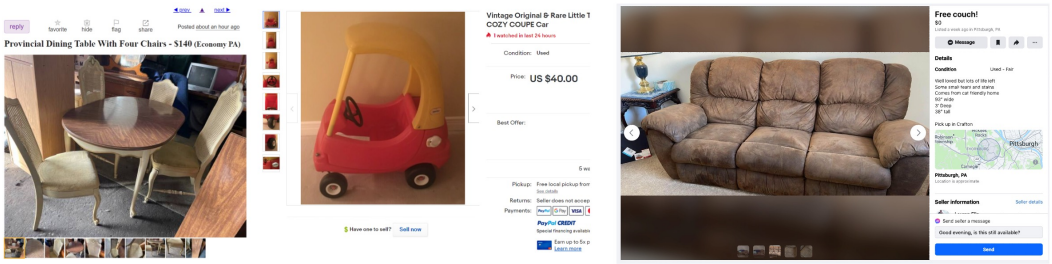


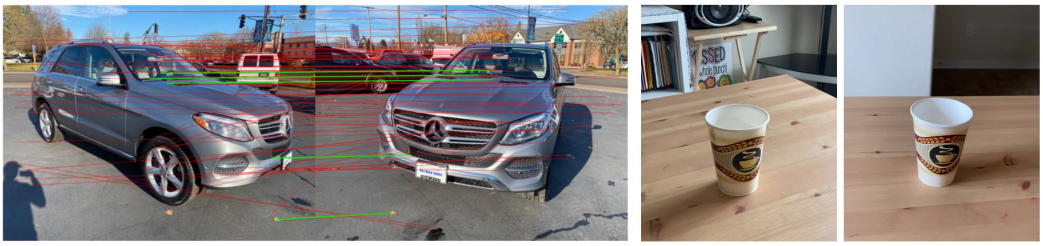
Figure 1.1: **Marketplace Listings as a Scalable Source of Multi-view Data.** Product listings on online marketplaces are a diverse and readily available source of multi-view data. Each listing is typically associated with several high-quality images of an object to sell.

able to leverage such data would significantly improve the domain generalization and diversity of representations that we can learn.

In this thesis, we take the first steps toward building a full pipeline capable of converting sparsely sampled views of an object as input into a fully textured 3D representation along with the associated illumination conditions. We start by proposing NeRS (Chapter 2), a surface-based representation that can be optimized with as few as four images using an analysis-by-synthesis framework. In contrast to existing volumetric approaches that model arbitrary geometry and view-dependent appearance, we constrain the geometry to only points that lie on a surface and constrain the view-dependent appearance using graphics-inspired rendering models. Specifically, we represent geometry using a water-tight implicit mesh. We model the illumination of the scene using a neural environment map and factor the diffuse color (albedo) from the specular lighting. Our final system can reconstruct 3D objects from as few as 4-8 images associated with noisy camera poses, and we demonstrate the scalability of our approach on hundreds of listings from an online marketplace and self-captures of everyday household objects.

NeRS assumes that each image is associated with noisy camera poses. While this constraint is already less stringent than contemporary and even follow-up approaches that require precisely aligned cameras, it is still challenging to recovery any camera parameters from sparsely sampled views consistently (see Fig. 1.2). This task is challenging because sparse views often have wide baselines, which make acquiring correspondences difficult if not impossible. Existing structure-from-motion pipelines for estimating camera poses rely on these correspondences to recover pose. Another challenge is that the poses of sparsely sampled views of objects are often ambiguous, particularly in the presence of object symmetry. A pose estimation method that effectively handles sparse view must thus be able to handle uncertainty.

To address this, we propose RelPose (Chapter 3), an energy-based model that predicts probability distributions over relative camera rotations. Specifically, we train a network that takes in pairs of images and a query rotation matrix and out-



Lack of Correspondences Ambiguities in Pose

Figure 1.2: **Challenges for Sparse-view 3D Pose Estimation.** *Left*: Sparsely sampled views often exhibit wide baselines, and wide baselines lead to larger appearance changes that make it difficult to find correspondences. Here, we show the correspondences recovered by SuperGlue [125], a state-of-the-art correspondence matcher. SuperGlue finds few inlier correspondences on the car because the front of the car is only partially visible in the left image. In sparse-view settings, it is possible to be looking at entirely opposing sides of an object, in which case correspondences are impossible to find altogether. *Right*: Given only a few images of an object, the pose may actually be ambiguous. This is especially the case for objects with symmetry.

puts a score that corresponds to how well the rotation matrix aligns with the ground truth relative rotation. To recover poses for more than two images, we solve for the set of rotations that would maximize the total pairwise sum of scores. In RelPose++ (Chapter 4), we extend our approach to handling multi-view context using a transformer and demonstrate that each additional image gives informative context that improves performance and reduces ambiguity. We also extend our approach to predicting 6-D camera pose (rotations and translations) using a new coordinate system that disentangles the ambiguity in rotation prediction from translation.

Finally, we reconsider what representation of camera should even be used for prediction. Typically, a camera is parameterized by its intrinsics ($\boldsymbol{K}$) and extrinsics ($\boldsymbol{R}$, $\boldsymbol{t}$). This compact representation, predicted using a global feature encoder that pools the spatial information present in any image, makes it challenging to reason about low-level information (*e.g.* correspondences) that is known to be important for pose estimation. Rather, in Cameras as Rays (Chapter 5), we revisit the classic parameterization of cameras as a bundle of rays [46]. This representation is both over-parameterized and generic, allowing a single representation to be used for any camera model (*e.g.* pinhole, wide-angle, fish-eye, *etc.*). We find that the set-to-set nature of transformers makes it easy to predict a ray for each patch in the image. We then propose both regression- and diffusion-based methods for predicting this distributed ray representation. We demonstrate that our method significantly outperforms prior work and exhibits much better precision.

Chapter 2

NeRS: Neural Reflectance Surfaces for Sparse-view 3D Reconstruction in the Wild

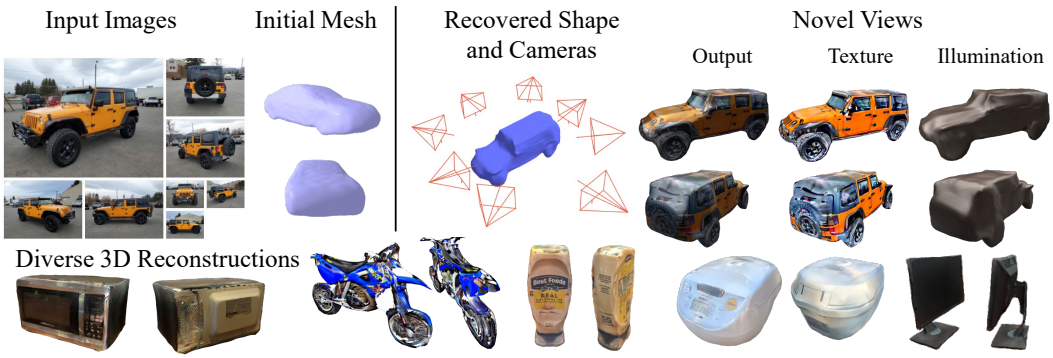


Figure 2.1: **3D view synthesis in the wild.** From several multi-view internet images of a truck and a coarse initial mesh (top left), we recover the camera poses, 3D shape, texture, and illumination (top right). We demonstrate the scalability of our approach on a wide variety of indoor and outdoor object categories (second row).

2.1 Introduction

Although we observe the surrounding world only via 2D percepts, it is undeniably 3D. The goal of recovering this underlying 3D from 2D observations has been a longstanding one in the vision community, and any computational approach aimed at this task must answer a central question about representation—how should we model the geometry and appearance of the underlying 3D structure?

An increasingly popular answer to this question is to leverage neural *volumetric* representations of density and radiance fields [89]. This allows modeling structures from rigid objects to translucent fluids, while further enabling arbitrary view-dependent lighting effects. However, it is precisely this unconstrained expressivity that makes it less robust and unsuitable for modeling 3D objects from sparse views in the wild. While these neural volumetric representations have been incredibly successful, they require hundreds of images, typically with precise camera poses, to model the full 3D structure and appearance of real-world objects. In contrast, when applied to 'in-the-wild' settings, *e.g.* a sparse set of images with imprecise camera estimates from off-the-shelf systems (see Fig. 2.1), they are unable to infer a coherent 3D representation. We argue this is because these neural volumetric representations, by allowing arbitrary densities and lighting, are *too* flexible.

Is there a robust alternative that captures real-world 3D structure? The vast majority of real-world objects and scenes comprise well-defined *surfaces*. This implies that the geometry, rather than being an unconstrained volumetric function, can be modeled as a 2D manifold embedded in Euclidean 3D space—and thus encoded via a (neural) mapping from a 2D manifold to 3D. Indeed, such meshed surface manifolds form the heart of virtually all rendering engines [38]. Moreover, instead of allowing arbitrary view-dependent radiance, the appearance of such surfaces can be described using (neural) bidirectional surface reflection functions (BRDFs), themselves developed by the computer graphics community over decades. We operationalize these insights into *Neural Reflectance Surfaces* (NeRS), a surface-based neural representation for geometry and appearance.

NeRS represents shape using a neural displacement field over a canonical sphere, thus constraining the geometry to be a watertight surface. This representation crucially associates a surface normal to each point, which enables modeling view-dependent lighting effects in a physically grounded manner. Unlike volumetric representations which allow unconstrained radiance, NeRS factorizes surface appearance using a combination of diffuse color (albedo) and specularity. It does so by learning neural texture fields over the sphere to capture the albedo at each surface point, while additionally inferring an environment map and surface material properties. This combination of a surface constraint and a factored appearance allows NeRS to learn efficiently and robustly from a sparse set of images in the wild while being able to capture varying geometry and complex view-dependent appearance.

Using only a coarse category-level template and approximate camera poses, NeRS can reconstruct instances from a diverse set of classes. Instead of evaluating in a synthetic setup, we introduce a dataset sourced from marketplace settings where multiple images of a varied set of real-world objects under challenging illumination are easily available. We show NeRS significantly outperforms neural volumetric or classic mesh-based approaches in this challenging setup, and as illustrated in Fig. 2.1, is able to accurately model the view-dependent appearance via its disentangled representation. Finally, as cameras recovered in the wild are

only approximate, we propose a new evaluation protocol for *in-the-wild* novel view synthesis in which cameras can be refined during both training *and* evaluation. We hope that our approach and results highlight the several advantages that neural surface representations offer, and that our work serves as a stepping stone for future investigations.

2.2 Related Work

Surface-based 3D Representations. As they enable efficient representation and rendering, polygonal meshes are widely used in vision and graphics. In particular, morphable models [10] allow parametrizing shapes as deformations of a canonical template and can even be learned from category-level image collections [17, 62]. With the advances in differentiable rendering [63, 71, 115], these have also been leveraged in learning-based frameworks for shape prediction [59, 43, 45] and view synthesis [119]. Whereas these approaches use an explicit discrete mesh, some recent methods have proposed using continuous neural surface parametrization like ours to represent shape [47] and texture [156, 8].

However, all of these works leverage such surface representations for (coarse) single-view 3D prediction given a category-level training dataset. In contrast, our aim is to infer such a representation given multiple images of a single instance, and without prior training. Closer to this goal of representing a single instance in detail, contemporary approaches have shown the benefits of using videos [72, 182] to recover detailed shapes, but our work tackles a more challenging setup where correspondence/flow across images is not easily available. In addition, while these prior approaches infer the surface texture, they do not enable the view-dependent appearance effects that our representation can model.

Volumetric 3D and Radiance Fields. Volumetric representations for 3D serve as a common, and arguably more flexible alternative to surface-based representations, and have been very popular for classical multi-view reconstruction approaches [40]. These have since been incorporated in deep-learning frameworks for shape prediction [42, 24] and differentiable rendering [180, 157]. Although these initial approaches used discrete volumetric grids, their continuous neural function analogs have since been proposed to allow finer shape [88, 107] and texture modeling [101].

Whereas the above methods typically aimed for category-level shape representation, subsequent approaches have shown particularly impressive results when using these representations to model a single instance from images [137, 151, 138] – which is the goal of our work. More recently, by leveraging an implicit representation in the form of a Neural Radiance Field, [89] showed the ability to model complex geometries and illumination from images. There has since been a flurry of impressive work to further push the boundaries of these representations and allow modeling deformation [114, 108], lighting variation [85], and similar to ours, leveraging insights from surface rendering to model radiance [185, 11, 102, 141, 196, 166, 184]. However, un-

like our approach which can efficiently learn from a sparse set of images with coarse cameras, these approaches rely on a dense set of multi-view images with precise camera localization to recover a coherent 3D structure of the scene. DietNeRF [55] reduces the number of images but requires precise cameras and semantic supervision. BARF [74] relaxes the constraint of precise cameras while foregoing view-dependent appearance and requiring a dense set of images. Other approaches [9, 195] that learn material properties from sparse views require specialized illumination rigs.

Multi-view Datasets. Many datasets study the longstanding problem of multi-view reconstruction and view synthesis. However, they are often captured in controlled setups, small in scale, and not diverse enough to capture the span of real-world objects. Middlebury [132] benchmarks multi-view reconstruction, containing two objects with nearly Lambertian surfaces. DTU [1] contains eighty objects with various materials but is still captured in a lab with controlled lighting. Freiburg cars [131] captures 360 degree videos of fifty-two outdoor cars for multi-view reconstruction. ETH3D [130] and Tanks and Temples [68] contain both indoor and outdoor scenes but are small in scale. Perhaps most relevant are large-scale datasets of real-world objects such as Redwood [22] and Stanford Products [103], but the data is dominated by single-views or small baseline videos. In contrast, our Multi-view Marketplace Cars (MVMC) dataset contains thousands of multi-view captures of in-the-wild objects under various illumination conditions, making it suitable for studying and benchmarking algorithms for multi-view reconstruction, view synthesis, and inverse rendering.

2.3 Method

Given a sparse set of input images of an object under natural lighting conditions, we aim to model its shape and appearance. While recent neural volumetric approaches share a similar goal, they require a dense set of views with precise camera information. Instead, our approach relies only on approximate camera pose estimates and a coarse category-level shape template. Our key insight is that instead of allowing unconstrained densities popularly used for volumetric representations, we can enforce a *surface*-based 3D representation. Importantly, this allows view-dependent appearance variation by leveraging constrained reflection models that decompose appearance into diffuse and specular components. In this section, we first introduce our (neural) surface representation that captures the object's shape and texture, and then explain how illumination and specular effects can be modeled for rendering. Finally, we describe how our approach can learn using challenging in-the-wild images.

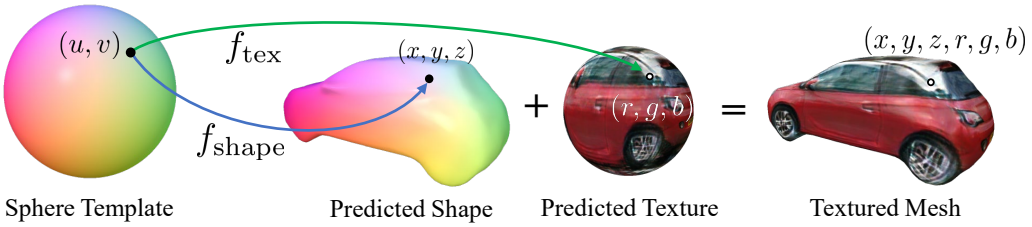


Figure 2.2: **Neural Surface Representation.** We propose an implicit, continuous representation of shape and texture. We model shape as a deformation of a unit sphere via a neural network f_{shape}, and texture as a learned per-uv color value via a neural network f_{tex}. We can discretize f_{shape} and f_{tex} to produce the textured mesh above.

2.3.1 Neural Surface Representation

We represent object shape via the deformation of a unit sphere. Previous works [59, 45] have generally modeled such deformations *explicitly*: the unit sphere is discretized at some resolution as a 3D mesh with V vertices. Predicting the shape deformation thus amounts to predicting vertex offsets $\delta \in \mathbb{R}^{V \times 3}$. Such *explicit discrete* representations have several drawbacks. First, they can be computationally expensive for dense meshes with fine details. Second, they lack useful spatial inductive biases as the vertex locations are predicted independently. Finally, the learned deformation model is fixed to a specific level of discretization, making it non-trivial, for instance, to allow for more resolution as needed in regions with richer detail. These limitations also extend to texture parametrization commonly used for such discrete mesh representations—using either per-vertex or per-face texture samples [63], or fixed resolution texture map, limits the ability to capture finer details.

Inspired by [47, 156], we address these challenges by adopting a continuous surface representation via a neural network. We illustrate this representation in Fig. 2.2. For any point u on the surface of a unit sphere $\mathbb{S}^2$, we represent its 3D deformation $x \in \mathbb{R}^3$ using the mapping $f_{\text{shape}}(u) = x$ where f_{shape} is parameterized as a multi-layer perceptron. This network, therefore, induces a deformation field over the surface of the unit sphere, and this deformed surface serves as our shape representation. We represent the surface texture in a similar manner–as a neural vector field over the surface of the sphere: $f_{\text{tex}}(u) = t \in \mathbb{R}^3$. This surface texture can be interpreted as an implicit UV texture map.

2.3.2 Modeling Illumination and Specular Rendering

Surface Rendering. The surface geometry and texture are not sufficient to infer the appearance of the object *e.g.* a uniformly red car may appear darker on one side, and lighter on the other depending on the direction of incident light. In addition,

depending on viewing direction and material properties, one may observe different appearances for the same 3D point *e.g.* shiny highlights from certain viewpoints. More formally, assuming that a surface does not emit light, the outgoing radiance L_o in direction v from a surface point x can be described by the rendering equation [57, 53]:

$$L_o(x, v) = \int_\Omega f_r(x, v, \omega) L_i(x, \omega)(\omega \cdot n) d\omega \tag{2.1}$$

where Ω is the unit hemisphere centered at surface normal n, and ω denotes the negative direction of incoming light. $f_r(x, v, \omega)$ is the bidirectional reflectance function (BRDF) which captures material properties (*e.g.* color and shininess) of surface S at x, and $L_i(x, \omega)$ is the radiance coming toward x from ω (Refer to Fig. 2.3). Intuitively, this integral computes the total effect of the reflection of every possible light ray ω hitting x bouncing in the direction v.

We thus need to infer the environment lighting and surface material properties to allow realistic renderings. However, learning arbitrary lighting L_i or reflection models f_r is infeasible given sparse views, and we need to further constrain these to allow learning. Inspired by concurrent work [176] that demonstrated its efficacy when rendering rotationally symmetric objects, we leverage the Phong reflection model [111] with the lighting represented as a neural environment map.

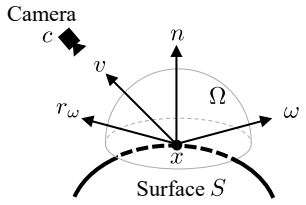


Figure 2.3: **Notation and convention for viewpoint and illumination parameterization.** The camera at c is looking at point x on the surface S. v denotes the direction of the camera w.r.t x, and n is the normal of S at x. Ω denotes the unit hemisphere centered about n. We compute the light arriving in the direction of every $\omega \in \Omega$, and r is the reflection of w about n.

Neural Environment Map. An environment map intuitively corresponds to the assumption that all the light sources are infinitely far away. This allows a simplified model of illumination, where the incoming radiance only depends on the direction ω and is independent of the position x *i.e.* $L_i(x, \omega) \equiv I_\omega$. We implement this as a neural spherical environment map f_{env} which learns to predict the incoming radiance for any query direction $L_i(x, \omega) \equiv I_\omega = f_{\text{env}}(\omega)$. Note that there is a fundamental ambiguity between material properties and illumination, *e.g.* a car that appears red could be a white car under red illumination or a red car under white illumination. To avoid this, we follow [176], and further constrain the environment illumination to be grayscale, *i.e.* $f_{\text{env}}(\omega) \in \mathbb{R}$.

Appearance under Phong Reflection. Instead of allowing an arbitrary BRDF f_r, the Phong reflection model decomposes the outgoing radiance from point x in direction v into the diffuse and specular components. The *view-independent* portion

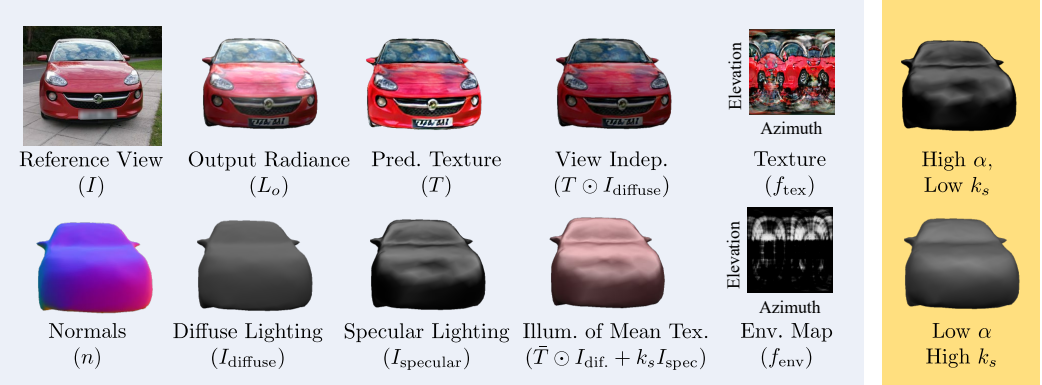


Figure 2.4: **Components of learned illumination model.** Given a query camera viewpoint (illustrated via the reference image I), we recover the radiance output L_o, computed using Phong shading [111]. Here, we show the full decomposition of learned components. From the environment map f_{env} and normals n, we compute diffuse (I_{diffuse}) and specular lighting (I_{specular}). The texture and diffuse lighting form the view-independent component ("View Indep.") and the specular lighting (weighted by the specular coefficient k_s) forms the view-dependent component of the radiance. Altogether, the output radiance $L_o = T \odot I_{\text{diffuse}} + k_s I_{\text{specularity}}$ (2.4). We also visualize the radiance using the mean texture, which is used to help learn plausible illumination. In the yellow box, we visualize the effects of the two specularity parameters. The shininess α controls the mirror-ness/roughness of the surface. The specular coefficient k_{s} controls the intensity of the specular highlights.

of the illumination is modeled by the diffuse component:

$$I_{\text{diffuse}}(x) = \sum_{\omega \in \Omega} (\omega \cdot n) I_{\omega}, \tag{2.2}$$

while the *view-dependent* portion of the illumination is modeled by the specular component:

$$I_{\text{specular}}(x, v) = \sum_{\omega \in \Omega} (r_{\omega,n} \cdot v)^{\alpha} I_{\omega}, \tag{2.3}$$

where $r_{\omega,n} = 2(\omega \cdot n)n - \omega$ is the reflection of ω about the normal n. The shininess coefficient $\alpha \in (0, \infty)$ is a property of the surface material and controls the "mirror-ness" of the surface. If α is high, the specular highlight will only be visible if v aligns closely with r_{ω}. Altogether, we compute the radiance of x in direction v as:

$$L_o(x, v) = T(x) \cdot I_{\text{diffuse}}(x) + k_s \cdot I_{\text{specular}}(x, v) \tag{2.4}$$

where the specularity coefficient k_{s} is another surface material property that controls the intensity of the specular highlight. $T(x)$ is the texture value at x computed by f_{tex}. For the sake of simplicity, α and k_s are shared across the entire instance. See Fig. 2.4 for a full decomposition of these components.

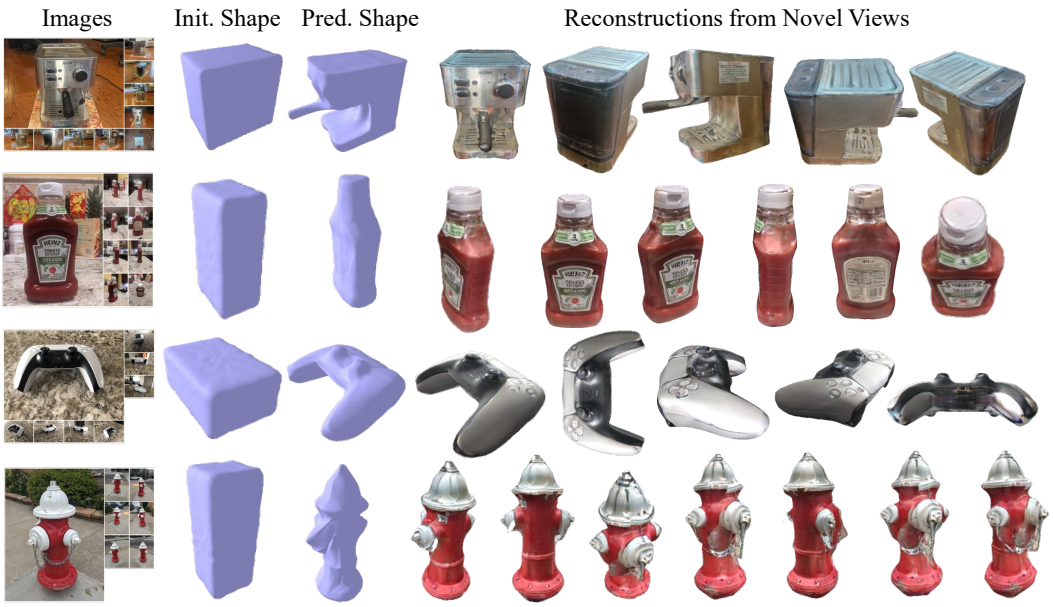


Figure 2.5: **Qualitative results on various household objects.** We demonstrate the versatility of our approach on an espresso machine, a bottle of ketchup, a game controller, and a fire hydrant. Each instance has 7-10 input views. We find that a coarse, cuboid mesh is sufficient as an initialization to learn detailed shape and texture. We initialize the camera poses by hand, roughly binning in increments of 45 degrees azimuth.

2.3.3 Learning NeRS in the Wild

Given a sparse set of images in the wild, our approach aims to infer a NeRS representation, which when rendered, matches the available input. Concretely, our method takes as input N (typically 8) images of the same instance $\{I_i\}_{i=1}^N$, noisy camera rotations $\{R_i\}_{i=1}^N$, and a category-specific mesh initialization $\mathcal{M}$. Using these, we aim to optimize full perspective cameras $\{\Pi\}_{i=1}^N$ as well as the neural surface shape f_{shape}, surface texture f_{text}, and environment map f_{env}. In addition, we also recover the material properties of the object, parametrized by a specularity coefficient k_s and shininess coefficient α.

Initialization. Note that both the camera poses and mesh initialization are only required to be coarsely accurate. We use an off-the-shelf approach [179] to predict camera rotations, and we find that a cuboid is sufficient as an initialization for several instances (See Fig. 2.5). We use off-the-shelf approaches [123, 67] to compute masks $\{M_i\}_{i=1}^N$. We assume that all images were taken with the same camera intrinsics. We initialize the shared global focal length f to correspond to a field of view of 60 degrees and set the principal point at the center of each image. We initialize the camera pose with the noisy initial rotations R_i and a translation t_i such that the

object is fully in view. We pre-train f_{shape} to output the template mesh $\mathcal{M}$.

Rendering. To render an image, NeRS first discretizes the neural shape model $f_{\text{shape}}(u)$ over spherical coordinates u to construct an explicit triangulated surface mesh. This triangulated mesh and camera Π_i are fed into PyTorch3D's differentiable renderer [115] to obtain per-pixel (continuous) spherical coordinates and associated surface properties:

$$[UV, N, \hat{M}_i] = \text{Rasterize}(\pi_i, f_{\text{shape}}) \tag{2.5}$$

where $UV[p]$, $N[p]$, and $\hat{M}[p]$ are (spherical) uv-coordinates, normals, and binary foreground-background labels corresponding to each image pixel p. Together with the environment map f_{env} and specular material parameters (α, k_s), these quantities are sufficient to compute the outgoing radiance at each pixel p under camera viewpoint Π_i using equation 2.4. In particular, denoting by $v(\Pi, p)$ the viewing direction for pixel p under camera Π, and using $u \equiv UV[p], n \equiv N[p]$ for notational brevity, the intensity at pixel p can be computed as:

$$\hat{I}[p] = f_{\text{tex}}(u) \cdot \Big(\sum_{\omega \in \Omega} (\omega \cdot n)\ f_{\text{env}}(\omega) \Big) + k_s \Big(\sum_{\omega \in \Omega} (r_{\omega,n} \cdot v(\Pi, p))^a\ f_{\text{env}}(\omega) \Big) \tag{2.6}$$

Image loss. We compute a perceptual loss [194] $L_{\text{perceptual}}(I_i, \hat{I}_i)$ that compares the distance between the rendered and true image using off-the-shelf VGG deep features. Note that being able to compute a perceptual loss is a significant benefit of surface-based representations over volumetric approaches such as NeRF [89], which operate on batches of rays rather than images, due to the computational cost of volumetric rendering. Similar to [176], we find an additional rendering loss using the mean texture (see Fig. 2.4 and Fig. 2.8 for examples) helps learn visually plausible lighting.

Mask Loss. To measure disagreement between the rendered and measured silhouettes, we compute a mask loss:

$$L_{\text{mask}} = \frac{1}{N} \sum_{i=1}^{N} \|M_i - \hat{M}_i\|_2^2, \tag{2.7}$$

a distance transform loss:

$$L_{\text{dt}} = \frac{1}{N} \sum_{i=1}^{N} D_i \odot \hat{M}_i, \tag{2.8}$$

and a 2D chamfer loss:

$$L_{\text{chamfer}} = \frac{1}{N} \sum_{i=1}^{N} \sum_{p \in E(M_i)} \min_{\hat{p} \in \hat{M}_i} \|p - \hat{p}\|_2^2. \tag{2.9}$$

D_i refers to the Euclidean distance transform of mask M_i, $E(\cdot)$ computes the 2D pixel coordinates of the edge of a mask, and $\hat{p}$ is every pixel coordinate in the predicted silhouette.

Regularization. Finally, to encourage smoother shape whenever possible, we incorporate a mesh regularization loss $L_{\text{regularize}} = L_{\text{normals}} + L_{\text{laplacian}}$ consisting of normals consistency and Laplacian smoothing losses [94, 30]. Note that such geometry regularization is another benefit of surface representations over volumetric ones. Altogether, we minimize:

$$L = \lambda_1 L_{\text{mask}} + \lambda_2 L_{\text{dt}} + \lambda_3 L_{\text{chamfer}} + \lambda_4 L_{\text{perceptual}} + \lambda_5 L_{\text{regularize}} \tag{2.10}$$

w.r.t $\Pi_i = \Big[R_i, t_i, f\Big]$, α, k_s, and the weights of f_{shape}, f_{text}, and $f_{\text{env\_map}}$.

Optimization. We optimize (2.10) in a coarse-to-fine fashion, starting with a few parameters and slowly increasing the number of free parameters. We initially optimize (2.10), w.r.t only the camera parameters Π_i. After convergence, we sequentially optimize f_{shape}, f_{tex}, and $f_{\text{env}}/\alpha/k_s$. We find it helpful to sample a new set of spherical coordinates u for each iteration when rasterizing. This helps propagate gradients over a larger surface and prevent aliasing. With 4 Nvidia 1080TI GPUs, training NeRS requires approximately 30 minutes.

2.4 Evaluation

In this section, we demonstrate the versatility of Neural Reflectance Surfaces to recover meaningful shape, texture, and illumination from in-the-wild indoor and outdoor images.

Multi-view Marketplace Dataset. To address the shortage of in-the-wild multi-view datasets, we introduce a new dataset, Multi-view Marketplace Cars (MVMC), collected from an online marketplace with thousands of car listings. Each user-submitted listing contains seller images of the same car instance. In total, we curate a subset of size 600 with at least 8 exterior views (averaging 10 exterior images per listing) along with 20 instances for an evaluation set (averaging 9.1 images per listing). We use [179] to compute rough camera poses. MVMC contains a large variety of cars under various illumination conditions (*e.g.* indoors, overcast, sunny, snowy, etc). The filtered dataset with anonymized personally identifiable information (*e.g.* license plates and phone numbers), masks, initial camera poses, and optimized NeRS cameras is available publicly.

Novel View Synthesis. Traditionally, novel view synthesis requires accurate target cameras to use as queries. Existing approaches use COLMAP [126] to recover ground truth cameras, but this consistently fails on MVMC due to specularities and limited views. On the other hand, we can use learning-based methods [179] to recover camera poses for both training and test views. However, as these are inherently approximate, this complicates training *and* evaluation. To account for this, we explore two evaluation protocols. First, to mimic the traditional evaluation setup, we obtain pseudo-ground truth cameras (with manual correction) and freeze them during training and evaluation. While this evaluates the quality of the 3D

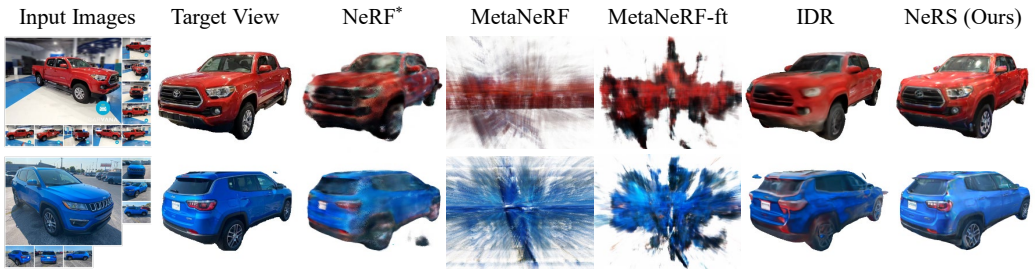


Figure 2.6: **Qualitative comparison *with fixed cameras.*** We evaluate all baselines on the task of novel view synthesis on Multi-view Marketplace Cars trained and tested with fixed, pseudo-ground truth cameras. One image is held out during training. Since we do not have ground truth cameras, we treat the optimized cameras from optimizing over all images as the ground truth cameras. We train a modified version (See Sec. 2.4) of NeRF [89] that is more competitive with sparse views (NeRF*). We also evaluate against a meta-learned initialization of NeRF with and without finetuning until convergence [146], but found poor results perhaps due to the domain shift from Shapenet cars. Finally, IDR [185] extracts a surface from an SDF representation but struggles to produce a view-consistent output given limited input views. We find that NeRS synthesizes novel views that are qualitatively closer to the target. The red truck has 16 total views while the blue SUV has 8 total views.

reconstruction, it does not evaluate the method's ability to jointly recover cameras. As a more realistic setup for evaluating view synthesis in the wild, we evaluate each method with approximate (off-the-shelf) cameras, while allowing them to be optimized.

Novel View Synthesis *with Fixed Cameras.* In the absence of ground truth cameras, we create pseudo-ground truth by manually correcting cameras recovered by jointly optimizing over all images for each object instance. For each evaluation, we treat one image-camera pair as the target and the remaining pairs for training. We repeat this process for each image in the evaluation set (totaling 182). Unless otherwise noted, qualitative results use approximate cameras and not the pseudo-ground truth.

Novel View Synthesis *in the Wild.* While the above evaluates the quality of the 3D reconstructions, it is not representative of in-the-wild settings where the initial cameras are unknown/approximate and should be optimized during training. Because even the test camera is approximate, each method is similarly allowed to refine the test camera to better match the test image while keeping the model fixed. Intuitively, this measures the ability of a model to synthesize a target view under *some* camera.

Baselines. We evaluate our approach against Neural Radiance Fields (NeRF) [89], which learns a radiance field conditioned on viewpoint and position

Figure 2.7: **Qualitative results for *in-the-wild* novel view synthesis.** Since off-the-shelf camera poses are only approximate for both training and test images, we allow cameras to be optimized during both training and evaluation (See Tab. 2.2 and Sec. 2.4). We find that NeRS generalizes better than the baselines in this unconstrained but more realistic setup.

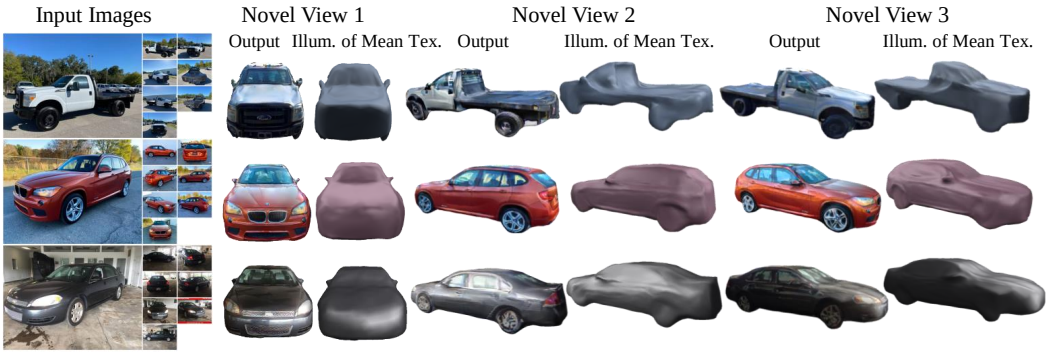


Figure 2.8: **Qualitative results on our in-the-wild Multi-view Marketplace Cars dataset.** Here we visualize the NeRS outputs as well as the illumination of the mean texture on 3 of the listings from the MVMC dataset. We find that NeRS recovers detailed textures and plausible illumination. Each instance has 8 input views.

and renders images using raymarching. We find that the vanilla NeRF struggles in our in-the-wild low-data regime. As such, we make a number of changes to make the NeRF baseline (denoted NeRF*) as competitive as possible, including a mask loss and a canonical volume. Please see the appendix for full details. We also evaluate a simplified NeRF with a meta-learned initialization for cars from multi-view images [146], denoted as MetaNeRF. MetaNeRF meta-learns an initialization such that with just a few gradient steps, it can learn a NeRF model. This allows the model to learn a data-driven prior over the shape of cars. Note that MetaNeRF is trained on ShapeNet [19] and thus has seen more data than the other test-time-optimization approaches. We find that the default number of gradient steps was insufficient for MetaNeRF to converge on images from MVMC, so we also evaluate MetaNeRF-ft, which is finetuned until convergence. Finally, we evaluate IDR [185], which represents geometry by extracting a surface from a signed distance field. IDR learns a neural renderer conditioned on the camera direction, position, and normal of the surface.

Metrics. We evaluate all approaches using the traditional image similarity met-

Method	MSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
NeRF* [89]	0.0393	16.0	0.698	0.287	231.7
MetaNeRF [146]	0.0755	11.4	0.345	0.666	394.5
MetaNeRF-ft [146]	0.0791	11.3	0.500	0.542	326.8
IDR [185]	0.0698	13.8	0.658	0.328	190.1
NeRS (Ours)	**0.0254**	**16.5**	**0.720**	**0.172**	**60.9**

Table 2.1: **Quantitative evaluation of novel-view synthesis on MVMC using *fixed pseudo-ground truth cameras*.** To evaluate novel view synthesis in a manner consistent with previous works that assume known cameras, we obtain pseudo-ground truth cameras by manually correcting off-the-shelf recovered cameras. We evaluate against a modified NeRF (NeRF*), a meta-learned initialization to NeRF with and without finetuning (MetaNeRF), and the volumetric surface-based IDR. NeRS significantly outperforms the baselines on all metrics on the task of novel-view synthesis with fixed cameras. See Fig. 2.6 for qualitative results.

Method	MSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
NeRF* [89]	0.0464	14.7	0.660	0.335	277.9
IDR [185]	0.0454	14.4	0.685	0.297	242.3
NeRS (Ours)	**0.0338**	**15.4**	**0.675**	**0.221**	**92.5**

Table 2.2: **Quantitative evaluation of *in-the-wild* novel-view synthesis on MVMC.** Off-the-shelf cameras estimated for in-the-wild data are inherently erroneous. This means that both training and test cameras are approximate, complicating training *and* evaluation. To compensate for approximate test cameras, we allow methods to refine the test camera given the test image with the model fixed. Intuitively this measures the ability of a method to synthesize a test image under *some* camera. We evaluate against NeRF and IDR, and find that NeRS outperforms the baselines across all metrics. See Fig. 2.7 for qualitative results.

rics Mean-Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index Measure (SSIM). We also compute the Learned Perceptual Image Patch Similarity (LPIPS) [194] which correlates more strongly with human perceptual distance. Finally, we compute the Fréchet Inception Distance [51] between the novel view renderings and original images as a measure of visual realism. In Tab. 2.1 and Tab. 2.2, we find that NeRS significantly outperforms the baselines in all metrics across both the fixed camera and in-the-wild novel-view synthesis evaluations. See Fig. 2.6 and Fig. 2.7 for a visual comparison of the methods.

Qualitative Results. In Fig. 2.8, we show qualitative results on our Multi-view Marketplace Cars dataset. Each car instance has between 8 and 16 views. We visualize the outputs of our reconstruction from 3 novel views. We show the rendering

for both the full radiance model and the mean texture. Both of these renderings are used to compute the perceptual loss (See Sec. 2.3.2). We find that NeRS recovers detailed texture information and plausible illumination parameters. To demonstrate the scalability of our approach, we also evaluate various household objects in Fig. 2.5. We find that a coarse, cuboid mesh is sufficient as an initialization to recover detailed shape, texture, and lighting conditions. Please refer to the project webpage for 360-degree visualizations.

2.5 Discussion

We present NeRS, an approach for learning neural surface models that capture geometry and surface reflectance. In contrast to volumetric neural rendering, NeRS enforces watertight and closed manifolds. This allows NeRS to model surface-based appearance effects, including view-dependant specularities and normal-dependant diffuse appearance. We demonstrate that such regularized reconstructions allow for learning from sparse in-the-wild multi-view data, enabling the reconstruction of objects with diverse material properties across a variety of indoor/outdoor illumination conditions. Further, the recovery of accurate camera poses in the wild (where classic structure-from-motion fails) remains unsolved and serves as a significant bottleneck for all approaches, including ours. We tackle this problem by using realistic but approximate off-the-shelf camera poses and by introducing a new evaluation protocol that accounts for this. We hope NeRS inspires future work that evaluates *in the wild* and enables the construction of high-quality libraries of real-world geometry, materials, and environments through better neural approximations of shape, reflectance, and illuminants.

Limitations. Though NeRS makes use of factorized models of illumination and material reflectance, there exist some fundamental ambiguities that are difficult from which to recover. For example, it is difficult to distinguish between an image of a gray car under bright illumination and an image of a white car under dark illumination. We visualize such limitations in the supplement. In addition, because the neural shape representation of NeRS is diffeomorphic to a sphere, it cannot model objects of non-genus-zero topologies.

Chapter 3

RelPose: Predicting Probabilistic Relative Rotation for Single Objects in the Wild

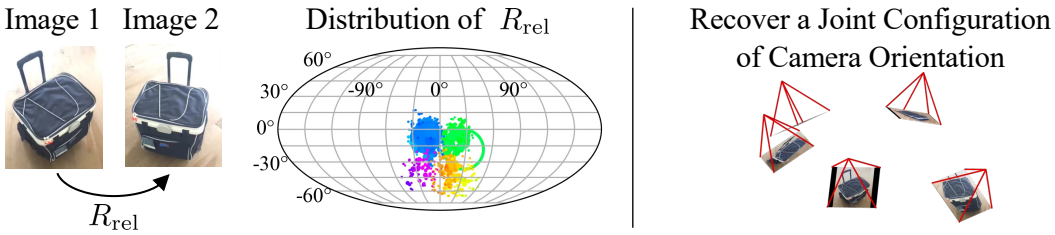


Figure 3.1: **Probabilistic Camera Rotation Estimation for Generic Objects**. *Left*: Given two images of the same object, we predict a conditional distribution of relative camera viewpoint (rotation) that effectively handles symmetries and pose ambiguities. *Right*: Given a set of images, our approach outputs a configuration of camera rotations.

3.1 Introduction

Recovering 3D from 2D images of an object has been a central task in vision for decades. Given multiple views, structure-from-motion (SfM) based methods can infer a 3D representation of the underlying instance while also associating each image with a camera viewpoint. However, these correspondence-driven methods cannot robustly handle sparsely sampled images that minimally overlap and typically require many (>20) images for a 360-degree 3D inference. Unfortunately, this requirement of densely sampled views can be prohibitive—online marketplaces often

have only a few images per instance, and a user casually reconstructing a novel object would also find capturing such views tedious. Although the recently emerging neural 3D reconstruction techniques also typically leverage similarly dense views, some works have shown promise that a far smaller number of images can suffice for high-quality 3D reconstruction. These successes have, however, still relied on precisely [144, 201, 77, 188, 175, 18] or approximately [74, 116, 192, 44, 155] known camera viewpoints for inference. To apply these methods at scale, we must therefore answer a fundamental question—*given sparsely sampled images of a generic object, how can we obtain the associated camera viewpoints*?

Existing methods do not provide a conclusive answer to this question. On the one hand, bottom-up correspondence-based techniques are not robustly applicable for sparse-view inference. On the other, recent neural multi-view methods can optimize already known approximate camera poses but provide no mechanism to obtain these to begin with. In this work, our goal is to fill this void and develop a method that, given a small number of unposed images of a generic object, can associate them with (approximate) camera viewpoints. Towards this goal, we focus on inferring the camera rotation matrices corresponding to each input image and propose a top-down approach to predict these. However, we note that the 'absolute' rotation is not well-defined given an image of a generic object—it assumes a 'canonical' pose which is not always known a-priori (e.g. what is an identity rotation for a pen? or a plant?). In contrast, the *relative* rotation between two views is well-defined even if a canonical pose for the instance is not. Thus, instead of adopting the common paradigm of single-image based pose prediction, we learn to estimate the relative pose given a pair of input images. We propose a system that leverages such pairwise predictions to then infer a consistent set of global rotations given multiple images of a generic object.

A key technical question that we consider is regarding the formulation of such pairwise pose estimation. Given two informative views of a rotationally asymmetric object, a regression-based approach may be able to accurately predict their relative transformation. The general case however, can be more challenging—given two views of a cup but with the handle only visible in one, the relative pose is ambiguous given just the two images. To allow capturing this uncertainty, we formulate an energy-based relative pose prediction network that, given two images *and* a candidate relative rotation, outputs an energy corresponding to the (unnormalized) log-probability of the hypothesis. This probabilistic estimation of relative pose not only makes the learning more stable, but more importantly, provides a mechanism to estimate a *joint distribution* over viewpoints given multiple images. We show that optimizing rotations to improve this joint likelihood yields coherent poses given multiple images and leads to significant improvements over naive approaches that do not consider the joint likelihoods.

We train our system using instances from over 40 commonplace object categories and find that not only can it infer accurate (relative) poses for novel instances of

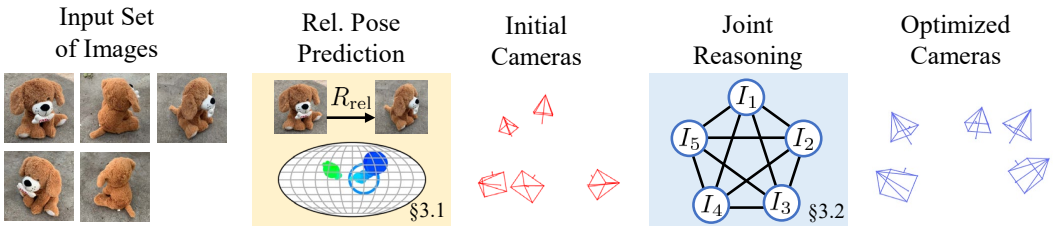


Figure 3.2: **Overview.** From a set of images, we aim to recover corresponding camera poses (rotations). To do this, we train a pairwise pose predictor that takes in two images and a candidate relative rotation and predicts energy. By repeatedly querying this network, we recover a probability distribution over conditional relative rotations (see Sec. 3.3.1). We use these pairwise distributions to induce a joint likelihood over the camera transformations across multiple images, and iteratively improve an initial estimate by maximizing this likelihood (see Sec. 3.3.2).

these classes, it even generalizes to instances from unseen categories. Our approach can thus be viewed as a stepping stone toward sparse-view 3D reconstruction of generic objects; just as classical techniques provide precise camera poses that (neural) multi-view reconstruction methods can leverage, our work provides a similar, albeit coarser, output that can be used to initialize inference in current (and future) sparse-view reconstruction methods. While our system only outputs camera rotations, we note that a reasonable corresponding translation can be easily initialized assuming object-facing viewpoints, and we show that this suffices in practice for bootstrapping sparse-view reconstruction.

3.2 Related Work

Structure-from-Motion (SfM). At a high level, structure-from-motion aims to recover 3D geometry and camera parameters from image sets. This is done classically by computing local image features [48, 79, 7, 152], finding matches across images [80], and then estimating and verifying epipolar geometry using bundle adjustment [153]. Later works have scaled up the SfM pipeline using sequential algorithms, demonstrating results on hundreds or even thousands of images [139, 39, 127, 126, 124].

The advent of deep learning has augmented various stages of the classical SfM pipeline. Better feature descriptors [31, 134, 168, 186, 34, 109, 118] and improved featured matching [125, 23, 76, 154, 35] have significantly outperformed their handcrafted counterparts. BA-Net [148] and DeepSFM [173] have even replaced the bundle-adjustment process by optimizing over a cost volume. Most recently, Pixel-Perfect SfM [75] uses a featuremetric error to post-process camera poses to achieve sub-pixel accuracy.

While these methods can achieve excellent localization, all these approaches are

bottom-up: beginning with local features that are matched across images. However, matching features requires sufficient overlap between images, which may not be possible given wide baseline views. While our work also aims to localize camera poses given image sets, our approach fundamentally differs because it is top-down and does not rely on low-level correspondences.

Simultaneous Localization and Mapping (SLAM). Related is the task of Monocular SLAM, which aims to localize and map the surroundings from a video stream. Indirect SLAM methods, similar to SfM, match local features across different images to localize the camera [122, 15, 92, 91]. Direct SLAM methods, on the other hand, define a geometric objective function to directly optimize over a photometric error [202, 129, 27, 36].

There have also been various attempts to introduce deep learning into SLAM pipelines. As with SfM, learned feature descriptors and matching have helped improve accuracy on SLAM subproblems and increased robustness. End-to-end deep SLAM methods [197, 95, 169, 171] have improved the robustness of SLAM compared to classical methods, but have generally not closed the gap on performance. One notable exception is the recent DROID-SLAM [149], which combines the robustness of learning-based SLAM with the accuracy of classical SLAM.

These approaches all assume *sequential* streams and generally rely on matching or otherwise incorporating temporal locality between neighboring frames. We do not make any assumptions about the order of the image inputs nor the amount of overlap between nearby frames.

Single-view Pose Prediction. The task of predicting a (6-DoF) pose from a single image has a long and storied history, the surface of which can barely be scratched in this section. Unlike relative pose between multiple images, the (absolute) pose given a single image is only well-defined if there exists a canonical coordinate system. Most single-view pose prediction approaches therefore deal with a fixed set of categories, each of which has a canonical coordinate system defined *a priori* [177, 150, 100, 20, 164, 54, 13, 140, 99, 93, 64, 66] or learned [143]. Other methods that are category-agnostic take in a 3D mesh or point cloud as input, which provides a local coordinate system [174, 179, 190, 104].

Perhaps most relevant to us are approaches that not only predict pose but also model inherent uncertainty in the pose prediction [12, 65, 93, 105, 26, 145, 113, 41, 28, 29, 90, 84]. Like our approach, VpDR-Net [98] uses relative poses as supervision but still predicts absolute pose (with a unimodal Gaussian uncertainty model). Implicit-PDF [93] is the most similar approach to ours and served as an inspiration. Similar to our approach, Implicit-PDF uses a neural network to implicitly represent probability using an energy-based formulation that elegantly handles symmetries and multimodal distributions. Unlike our approach, Implicit-PDF (and all other single-view pose prediction methods) predict *absolute* pose, which does not exist in general for generic or novel categories. Instead, we model probability distributions over relative pose given pairs of images.

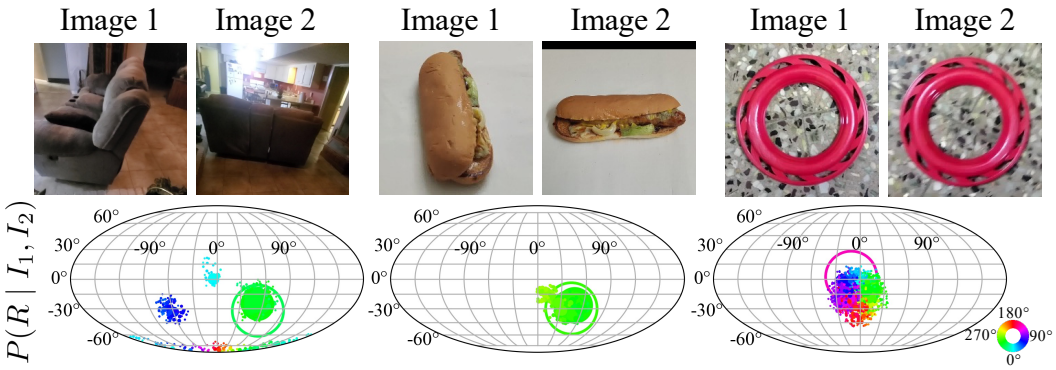


Figure 3.3: **Predicted conditional distribution of image pairs from unseen categories.** Here, we visualize the predicted conditional distribution of image pairs. Inspired by [93], we visualize the rotation distribution (Algorithm 1) by plotting yaw as latitude, pitch as longitude, and roll as the color. The size of each circle is proportional to the probability of that rotation. We omit rotations with negligible probability. The center of the open circle represents the ground truth. We can see that network predicts 4 modes for the couch images, corresponding roughly to 90-degree increments, with the greatest probability assigned to the correct 90-degree rotation. The relative pose of the hot dog is unambiguous and thus only has one mode. While the relative pose for the frisbee has close to no pitch or yaw, the roll remains ambiguous, hence the variety in colors. See the supplement for a visualization of how to interpret the relative rotations.

Learning-based Relative Pose Prediction. When considering generic scenes, prior works have investigated the task of relative pose prediction given two images. However, these supervised [159] or self-supervised [198, 187, 83, 163, 70] methods typically consider the prediction of motion between consecutive frames and are not easily adapted to wide-baseline prediction. While some approaches have investigated wide baseline prediction [87, 6, 120], regression-based inference can not effectively capture uncertainty, unlike our energy-based model. Perhaps most similar to ours is DirectionNet [21] which also predicts a camera distribution for wide baseline views. While DirectionNet only uses the expected value of the distribution and thus ignores symmetry, we take advantage of multimodal distributions to improve our joint pose estimation.

3.3 Method

Given a set of N images $\{I_1, \ldots I_N\}$ depicting a *generic* object in the wild, we aim to recover a set of N rotation matrices $\{R_1, \ldots, R_N\}$ such that rotation matrix R_i corresponds to the viewpoint of the camera used to take image i. Note that while we do not model translation, it can be easily initialized using object-facing

viewpoints for 3D object reconstruction [74, 192] or a pose graph for SLAM [16]. We are primarily interested in settings with only sparse views and wide baselines. While bottom-up correspondence-based techniques can reliably recover camera pose given dense views, they do not adapt well to sparse views with minimal overlap. We instead propose a prediction-based top-down approach that can learn and exploit the global structure directly.

The basic building block of our prediction system (visualized in figure 3.3) is a pairwise pose predictor that infers *relative* camera orientations given pairs of images. However, symmetries in objects and possibly uninformative viewpoints make this an inherently uncertain prediction task. To allow capturing this uncertainty, we propose an energy-based approach that models the *multi-modal distribution* over relative poses given two images.

Given the predicted distributions over pairwise relative rotations, we show that these can be leveraged to induce a *joint* distribution over the rotations. Starting with a greedy initialization, we present a coordinate-ascent approach that jointly reasons over and improves the set of inferred rotations. We describe our approach for modeling probability distributions over relative poses between two images in Sec. 3.3.1, and build on this in Sec. 3.3.2 to recover a joint set of poses across multiple images. Finally, we discuss implementation details in Sec. 3.3.3.

3.3.1 Estimating Pair-wise Relative Rotations

Given a pair of images depicting an arbitrary object, we aim to predict a distribution over the relative rotation corresponding to the camera transformation between the two views. As there may be ambiguities when inferring the relative pose given two images, we introduce a formulation that can model uncertainty.

```
procedure PairwiseDistribution(I_1, I_2)
    queries ←SampleRotationsUnif(50000)
    energies ← f(I_1, I_2, queries)
    probs ← SoftMax(energies)
    return queries, probs
end procedure
```

Algorithm 1: **Pseudo-code for recovering a pairwise distribution**. We describe how to recover the distribution of the relative pose given images.

Energy-based Formulation. We wish to model the conditional distribution over a relative rotation matrix R given input images I_1 and I_2: $P(R \mid I_1, I_2)$. Inspired by recent work on *implicitly* representing the distribution over rotations using a neural network [93], we propose using an energy-based relative pose estimator. More specifically, we train a network $f(R, I_1, I_2)$ that learns to predict the energy, or the unnormalized joint log-probability, $P(R, I_1, I_2) = \alpha \exp f(R, I_1, I_2)$ where α is the constant of integration. From the product rule, we can recover the conditional probability as a function of f:

$$P(R \mid I_1, I_2) = \frac{P(R, I_1, I_2)}{P(I_1, I_2)} \approx \frac{\alpha \exp f(R, I_1, I_2)}{\sum_{R'} \alpha \exp f(R', I_1, I_2)} = \frac{\exp f(R, I_1, I_2)}{\sum_{R'} \exp f(R', I_1, I_2)} \tag{3.1}$$

We marginalize over rotations to avoid having to compute α (see Algorithm 1), but note that the number of sampled rotations should be large for the approximation to be accurate. It is therefore important to use a lightweight network f since it is queried once per sampled rotation in the denominator.

Training. We train our network by maximizing the log-likelihood of the conditional distribution, or equivalently minimizing the negative log-likelihood:

$$\mathcal{L} = -\log P(R_1^\top R_2 \mid I_1, I_2) \tag{3.2}$$

where R_1 and R_2 are the ground truth poses of I_1 and I_2 respectively. Note that while the 'absolute' poses (R_1, R_2) are in an arbitrary coordinate system (depending on e.g. SLAM system outputs), the relative pose $R_1^\top R_2$ between two views is agnostic to this incidental canonical frame. Following eq. (3.1), we sample multiple candidate rotation matrices to compute the conditional probability.

Inference. Recovering the optimal transformation from the pose of I_1 to I_2 amounts to optimizing f over the space of rotations:

$$R^* = \underset{R \in \mathbf{SO}(3)}{\arg\max}\, P(R \mid I_1, I_2) = \underset{R \in \mathbf{SO}(3)}{\arg\max}\, f(R, I_1, I_2) \tag{3.3}$$

In practice, the loss landscape of f is often un-smooth, so we find that sampling and scoring rotations based on f to be more effective than gradient ascent.

We can also compute the conditional distribution of the relative rotation from I_1 to I_2 by sampling rotations over $\mathbf{SO}(3)$. The probability associated with each rotation can be computed using a softmax function, as described Algorithm 1 and derived in eq. (3.1). Inspired by [93], we can visualize the distribution of rotations by projecting the rotation matrices on a 2-sphere using pitch and yaw and coloring the rotation based on the roll. See Fig. 3.3 and the supplement for sample results.

3.3.2 Recovering Joint Poses

In the previous section, we describe an energy-based relative pose predictor conditioned on pairs of images. Using this network, we recover a coherent set of rotations when given a set of images.

Greedy Initialization. Given predictions for relative rotations between every pair of images, we aim to associate each image with an absolute rotation. However, as the relative poses are invariant up to a global rotation, we can treat the pose of the first image as the identity matrix: $R_1 = I$. We note that the rotations for the other images can be uniquely induced given any $N-1$ relative rotations that span a tree. *Sequential Chain.* Perhaps the simplest way to construct such a tree is to treat the images as part of an ordered sequence. Given $R_1 = I$, all subsequent poses can be computed by using the best scoring relative pose from the previous image: $R_i = R_{i-1} R^*_{(i-1)\to i}$, denoting $R_{i\to j}$ as the relative rotation matrix $R_i^\top R_j$. However,

```
procedure CoordAsc(Images {I_i}_N)
    {R_i}_N ← InitializeRotations({I_i}_N)
    for t ∈ 1, ..., Num Iterations do
        k ← RandomInteger(N)
        ▷ R'_k (Q×3×3): Q replacements for R_k
        R'_k ←SampleRotationsUnif(Q=250000)
        energies ← Zeros(Q)
        for i ∈ 1, ..., N and i ≠ k do
            R ← Repeat(R_i, Q)                          ▷ 3×3→Q×3×3
            energies ← energies + f(I_i, I_k, R^⊤ R'_k)
            energies ← energies + f(I_k, I_i, R'_k^⊤ R)
        end for
        R_k ← R'_k[ArgMax(energies)]
    end for
end procedure
```

Algorithm 2: **Pseudo-code for joint inference using relative pose predictor**. We describe how to recover the joint poses given n images via coordinate ascent.

this assumes that the images are captured sequentially (e.g. in a video) and may not be applicable for settings such as online marketplaces.

Maximum Spanning Tree. We improve over the naive linear chain by recognizing that some pairs of images may produce more confident predictions. Given N images, we construct a directed graph with $N \cdot (N-1)$ edges, where the weight of edge $(i,j) = P(R^*_{i\to j}|I_i, I_j)$. We then construct a Maximum Spanning Tree (MST) that covers all images with the most confident set of relative rotations.

Reasoning over all images jointly. Both of the previous methods, which select a subset of edges, do not perform any joint reasoning and discard all but the highest scoring mode for each pair of images. Instead, we can take advantage of our energy-based formulation to enforce global consistency.

Given our pairwise conditional probabilities, we can define a joint distribution over the set of rotations:

$$P\left(\{R_i\}_{i=1}^N \mid \{I_i\}_{i=1}^N\right) = \alpha \exp\Big(\sum_{(i,j)\in\mathcal{P}} f(R_{i\to j} \mid I_i, I_j)\Big) \tag{3.4}$$

where $\mathcal{P} = \{(i,j) \mid (i,j) \in [N]\times[N], i \neq j\}$ is the $N(N-1)$ set of pairwise permutations and α is the normalizing constant. Intuitively, this corresponds to the distribution modeled by a factor graph with a potential function corresponding to each pairwise edge.

We then aim to find the most likely set of rotations $\{R_1, \ldots, R_N\}$ under this conditional joint distribution (assuming $R_1 = I$). While it is not feasible to analytically obtain the global maxima, we adopt an optimization-based approach and iteratively improve the current estimate. More specifically, we initialize the set of poses with the greedy MST solution, and at each iteration, we randomly select a rotation R_k to update. Assuming fixed values for $\{R_i\}_{i\neq k}$, we then search for the rotation R_k

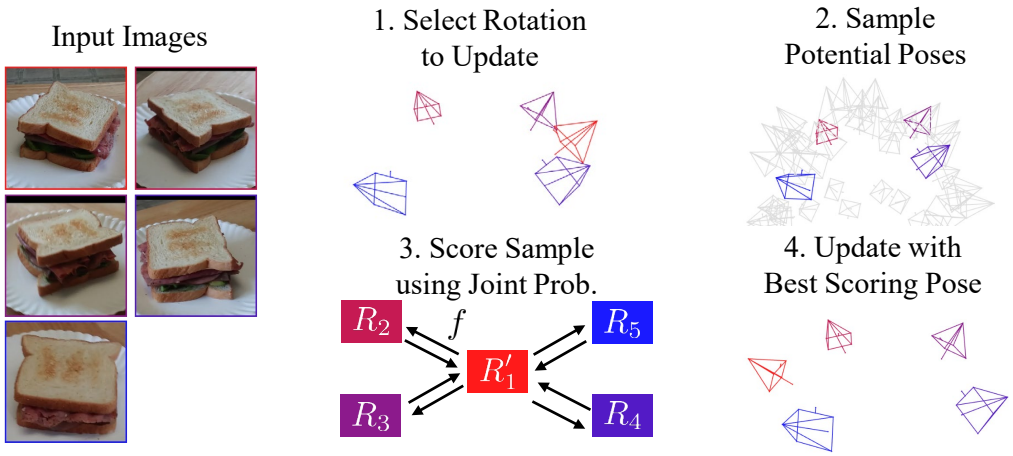


Figure 3.4: **Recovering Joint Poses with Coordinate Ascent.** Given a set of images $\{I_1, \ldots, I_N\}$, we initialize a set of corresponding poses $\{R_1, \ldots, R_N\}$. During each iteration of coordinate ascent, we: 1) randomly select one pose R_k to update (the red camera in this case); 2) sample a large number (250k) of candidate poses; 3) score each pose according to the joint distribution conditioned on the other poses and images eq. (3.5); and 4) update with the highest scoring pose. See Sec. 3.3.2 for more detail.

under the conditional distribution that maximizes the overall likelihood. We show in supplementary that this in fact corresponds to computing the most likely hypothesis under the distribution $P(R'_k \mid \{R_i\}_{i \neq k}, \{I_i\}_i)$:

$$\log P(R'_k \mid \{R_i\}_{i \neq k}, \{I_i\}_i) = \sum_{i \neq k} \left(f(R_{i \to k'}, I_i, I_k) + f(R_{k' \to i}, I_k, I_i) \right) + C \tag{3.5}$$

Analogous to our approach for finding the optimal solution for a single relative rotation, we sample multiple hypotheses for the rotation R_k, and select the hypothesis that maximizes eq. (3.5). We find that this search-based block coordinate ascent helps us consistently improve over the initial solution while avoiding the local optima that continuous optimization is susceptible to. We provide pseudo-code in Algorithm 2 and visualize one iteration of coordinate ascent in Fig. 3.4.

3.3.3 Implementation Details

Network Architecture. We use a ResNet-50 [50] with anti-aliasing [193] to extract image features. We use a lightweight 3-layer MLP that takes in a concatenation of 2 sets of image features and a rotation matrix to predict energy. We use positional encoding [89, 147] directly on flattened 3×3 rotation matrix, similar to [93]. See the supplement for architecture diagrams.

Number of Rotation Samples. We use the equivolumetric sampling in [93] to compute query rotations (37k total rotations) during training. For each iteration of coordinate ascent, we randomly sample 250k rotation matrices. For visualizing distributions, we randomly sample 50k rotations.

Runtime. We train the pairwise estimator with a batch size of 64 images for approximately 2 days on 4 NVIDIA 2080TI GPUs. Inference for 20 images takes around 1-2 seconds to construct an MST and around 2 minutes for 200 iterations of coordinate ascent on a single 2080TI. Note that the runtime of the coordinate ascent scales linearly with the number of images.

3.4 Evaluation

3.4.1 Experimental Setup

Dataset. We train and test on the Common Objects in 3D dataset (CO3D) [117], a large-scale dataset consisting of turntable-style videos of 51 common object categories. We train on the subset of the dataset that has camera poses, which were acquired by running COLMAP [126] over all frames of the video.

To train our network, we sample random frames and their associated camera poses from each video sequence. We train on 12,299 video sequences (from the `train-known` split) from 41 categories, holding out 10 categories to test generalization. We evaluate 1,711 video sequences (from the `test-known` split) over all 41 trained categories (seen) as well as the 10 held-out categories (unseen). The 10 held out categories are: `ball`, `book`, `couch`, `frisbee`, `hotdog`, `kite`, `remote`, `sandwich`, `skateboard`, and `suitcase`. We selected these categories randomly after excluding some of the categories with the most training images.

Task and Metrics. We consider the task of sparse-view camera pose estimation with $N = 3$, 5, 10, and 20 images, subsampled from a video sequence. This is highly challenging, especially when $N \leq 10$, because the ground truth camera poses have wide baselines.

We consider two possible ways to select N frames from a video sequence. First, we can randomly sample a set of N indices per video sequence (Random). Alternatively, we can use N uniformly-spaced frame indices (Uniform). We note that because CO3D video sequences are commonly taken in a turntable fashion, the uniformly spaced sampling strategy may be more representative of real-world distributions of sparse view image sets. We report metrics on both task setups.

Because the global transformation of the camera poses is ambiguous, we evaluate each pair of relative rotations. For each of the $N(N-1)$ pairs, we compare the angular difference between the relative predicted rotation and the relative ground truth rotation using Rodrigues' formula [121]. We report the proportion of relative rotations that are within 15 and 30 degrees of the ground truth. We note that

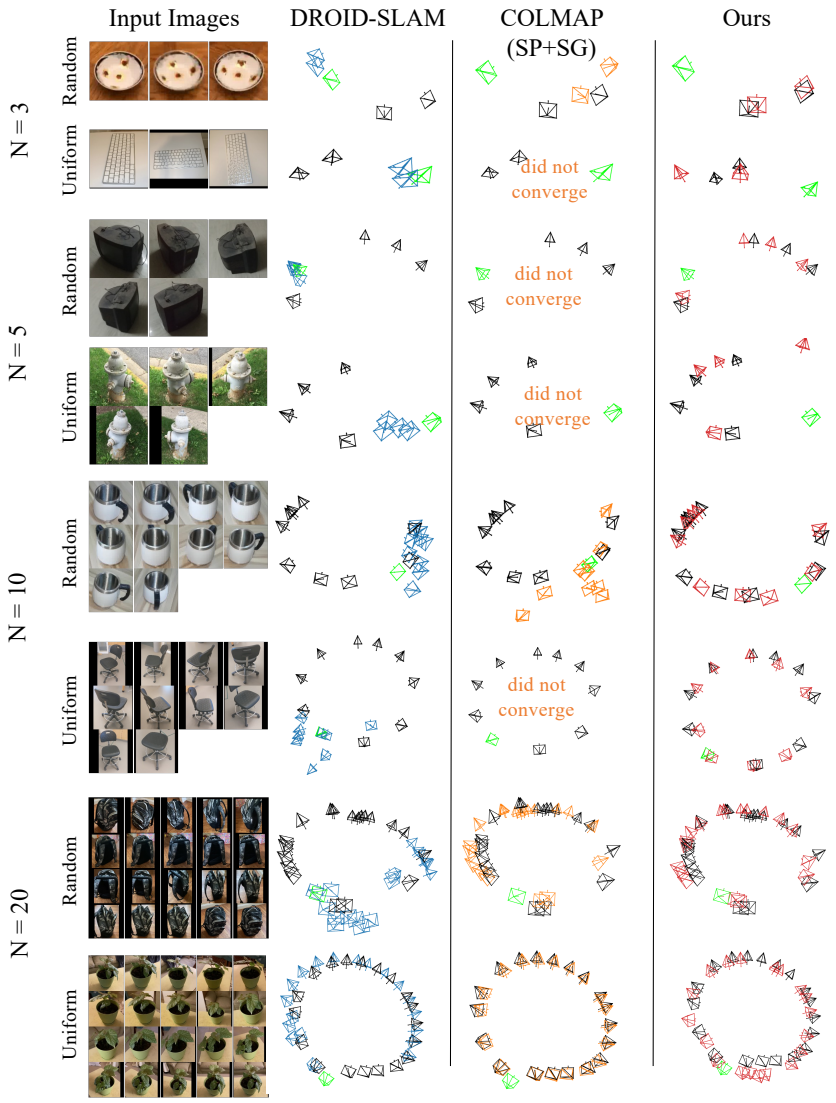


Figure 3.5: **Qualitative Comparison of Recovered Camera Poses with Baselines.** We visualize the camera poses (rotations) predicted by DROID-SLAM, COLMAP with SuperPoint/SuperGlue, and our method given sparse image frames. The black cameras correspond to the ground truth. We only visualize the rotations predicted by each method and set the translation such that the object center is a fixed distance away along the camera axis. As the poses are agnostic to a global rotation, we align the predicted cameras across all methods to the ground truth coordinate system by setting the recovered camera pose for the first image to the corresponding ground truth (visualized in green). Odd rows correspond to randomly sampled image frames, while even rows correspond to uniformly-spaced image frames.

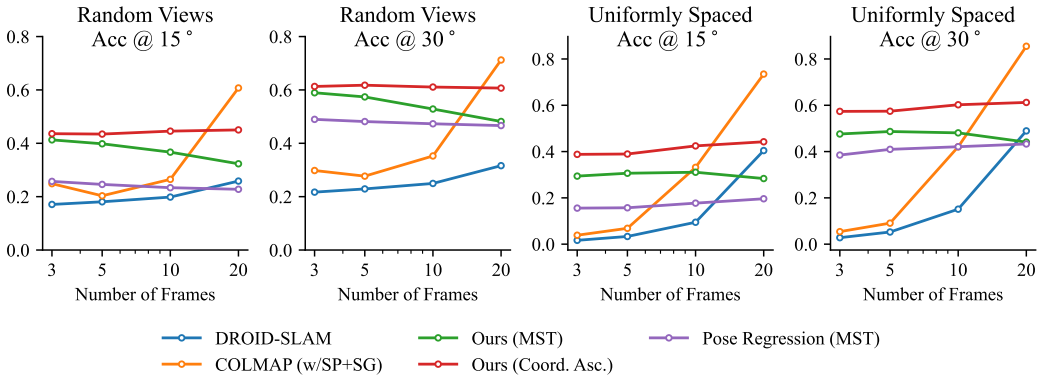


Figure 3.6: **Mean Accuracy on *Seen* Categories.** We evaluate our approach against competitive SLAM (DROID-SLAM) and SfM (COLMAP with SuperPoint + SuperGlue) baselines in sparse-view settings. We also train a direct relative rotation predictor (Pose Regression) that is not probabilistic and uses the MST generated by our method to recover joint pose. We consider both random sampling and uniformly spacing frames from a video sequence. We report the proportion of pairwise relative poses that are within 15 and 30 degrees of the ground truth, averaged over all seen categories. We find that our approach shines with fewer views because it does not rely on correspondences and thus can handle wide baseline views. The correspondence-based approaches need about 20 images to begin to work.

rotation errors within this range are relatively easy to handle by downstream 3D reconstruction tasks (See figure 3.10 for an example).

Baselines. We compare against DROID-SLAM [149], a current state-of-the-art SLAM approach that incorporates learning in an optimization framework. Note that DROID-SLAM requires trajectories and camera intrinsics. Thus, we provide the DROID-SLAM baseline with sorted frame indices and intrinsics, but do not provide these to any other method.

We also compare with a state-of-the-art structure-from-motion pipeline that uses COLMAP [126] with SuperPoint feature extraction [31] and SuperGlue matching [125]. We used the implementation provided by [124]. For instances for which COLMAP does not converge or is unable to localize some cameras, we treat the missing poses as identity rotation for evaluation. We note that DROID-SLAM also outputs approximate identity rotations when the optimization fails.

Ablations. In the spirit of learning-based solutions that directly regress pose, we train a network that predicts relative rotation directly given two images. Similar to our energy-based predictor, we pass the concatenated image features from a ResNet-50 into an MLP. We double the number of layers from 3 to 6 and add a skip connection to give this network increased capacity. Rotations are predicted using the 6D rotation representation [200]. See the supplement for additional architecture

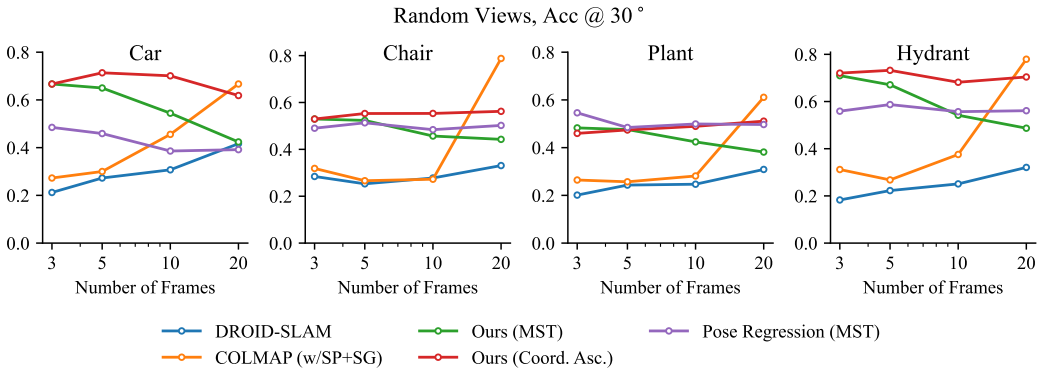


Figure 3.7: **Accuracy on Subset of *Seen* Categories.** Here we compare all approaches on a representative subset of seen categories. We find that direct regression of relative poses (purple) struggles more on categories with symmetry (Car, Hydrant) than categories without symmetry (Chair, Plant), suggesting that multimodal prediction is important for resolving ambiguity.

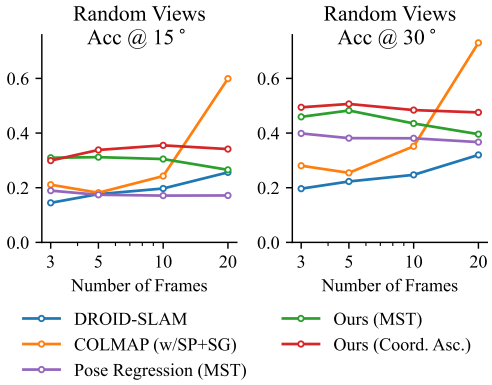


Figure 3.8: **Mean Accuracy on *Unseen* Categories.** We evaluate our approach on held-out categories from CO3D.

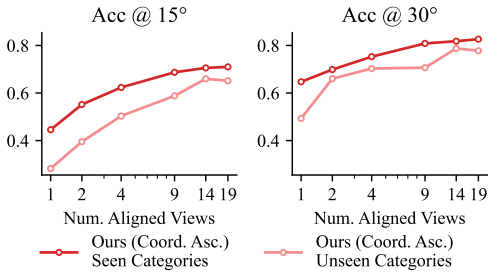


Figure 3.9: **Novel View Registration.** Here, we evaluate the task of registering a new view given previously aligned cameras. We find that adding more views improves performance, suggesting that additional views reduce ambiguity.

details. The relative pose regressor cannot directly predict poses for more than two images. To recover sets of poses from sets of images, we use the MST graph recovered by our method to link the pairs of relative rotations (we find that this performs better than linking the relative rotations sequentially).

To demonstrate the benefits of joint reasoning, we additionally report the performance of our method using the greedy Maximum Spanning Tree (MST) solution. The performance of the sequential solution is in the supplement.

3.4.2 Quantitative Evaluation

We evaluate all approaches on sparse-view camera pose estimation by averaging over all seen categories in figure 3.6. We find that our approach outperforms all baselines for $N \leq 10$ images. Correspondence-based approaches (DROID-SLAM and COLMAP) do not work until roughly 20 images, at which point image frames have a sufficient overlap for local correspondences. However, real-world multi-view data (e.g. marketplace images) typically have much fewer images. We find that coordinate ascent helps our approach scale with more image frames whereas the greedy maximum spanning tree accumulates errors with more frames.

Directly predicting relative poses does not perform well, possibly because pose regression cannot model multiple modes, which is important for symmetrical objects. We visualize the performance for four categories in figure 3.7. We find that the performance gap between our approach and direct regression is larger for objects with some symmetry (car, hydrant) than for objects without symmetry (chair, plant). Moreover, unlike our energy-based approach that models a joint distribution, a regression-based method does not allow similar joint reasoning.

We also test the generalization of our approach for *unseen* categories in figure 3.8. We still find that our method significantly outperforms all other approaches from sparse views ($N \leq 10$) even for never-before-seen object categories, indicating its ability to handle generic objects beyond training. The per-category evaluation for both seen and unseen categories are in the supplement.

Novel View Registration. In our standard SfM-inspired task setup, we aim to recover N camera poses given N images. Intuitively, adding images reduces ambiguity, but recovering additional cameras is also more challenging. To disambiguate between the two, we evaluate the task of registering new views given previously aligned images in figure 3.9. Given $N + 1$ images, of which N have aligned cameras, we use our energy-based regressor to recover the remaining camera (equivalent to one iteration of coordinate ascent). We find that adding images improves accuracy, suggesting that additional views can reduce ambiguity.

3.4.3 Qualitative Results

We show qualitative results on the outputs of our pairwise predictor in figure 3.3. The visualized distributions suggest that our model is learning useful information about symmetry and can model multiple modes even for unseen categories.

We visualize predicted camera poses for DROID-SLAM, COLMAP, and our method with coordinate ascent in figure 3.5. Unable to bridge the domain gap from narrow baseline video frames, DROID-SLAM often gets stuck in the trajectory. Although COLMAP sometimes fails to converge, it performs well for N=20. Our approach consistently outputs plausible interpretations but is unable to achieve *precise* localization. See supplementary for visualizations on randomly selected sequences and more category-specific discussion.

Figure 3.10: **Initializing 3D NeRS Reconstruction using Predicted Cameras.** NeRS [192] is a representative 3D reconstruction approach that takes noisy cameras as initialization and jointly optimizes object shape, appearance, and camera poses. We run our method with coordinate ascent on 7 input images of a fire hydrant and 4 input images of a motorbike to obtain the camera initialization (green), which we provide to NeRS. NeRS then finetunes the cameras (orange) and outputs a 3D reconstruction.

We also validate that our camera pose estimations can be used for downstream 3D reconstruction. We use our camera poses to initialize NeRS [192], a representative sparse-view surface-based approach that requires a (noisy) camera initialization. Using our cameras, we successfully reconstruct a 3D model of a fire hydrant from 7 images and a motorbike from 4 images in figure 3.10.

3.5 Discussion

We presented a prediction-based approach for estimating camera rotations given (a sparse set of) images of a generic object. Our energy-based formulation allows capturing the underlying uncertainty in relative poses, while also enabling joint reasoning over multiple images. We believe our system's robustness under sparse views can allow it to serve as a stepping stone for initializing (neural) reconstruction methods in the wild, but also note that there are several open challenges. First, our work reasoned about the joint distribution using only pairwise potentials, and developing efficient higher-order energy models may further improve performance. Moreover, while we outperform existing techniques given sparse views, the correspondence-driven methods are more accurate given a large number of views, and we hope future efforts can unify the two approaches. Finally, our approach may not be directly applicable to reasoning about camera transformations for arbitrary scenes as modeling camera translation would be more important compared to object-centric images.

Chapter 4

RelPose++: Recovering 6D Poses from Sparse-view Observations

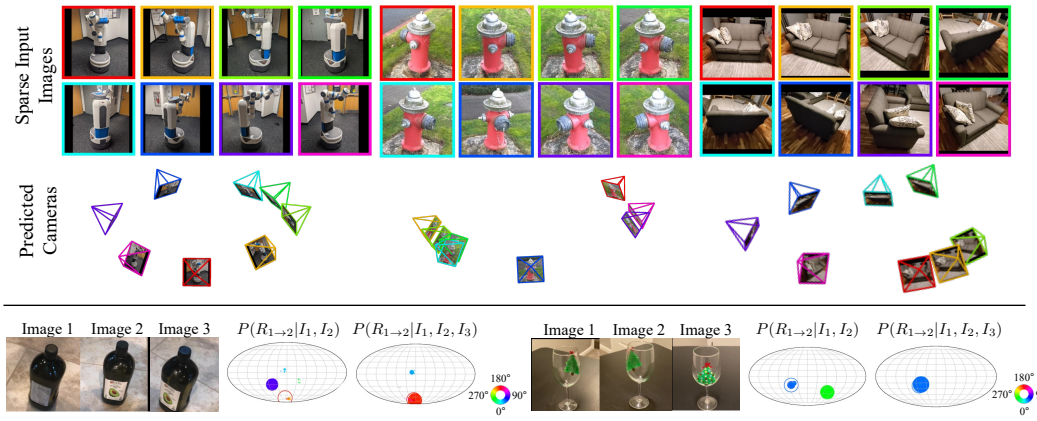


Figure 4.1: **Estimating 6D Camera Poses from Sparse Views.** We propose a framework RelPose++ that, given a sparse set of input images, can infer the corresponding 6D camera rotations and translations (**top**: the cameras are colored from red to magenta based on the image index). RelPose++ estimates a probability distribution over the relative rotations of the cameras corresponding to any 2 images, but can do so while incorporating multi-view cues. We find that the distribution improves given additional images as context (**bottom**).

4.1 Introduction

In Chapter 3, we saw that RelPose predicts distributions over pairwise relative rotations to then optimize multi-view consistent rotation hypotheses. While this optimization helps enforce multi-view consistency, RelPose's predicted distributions

only consider pairs of images, which can be limiting. As an illustration, if we consider the first two images of the bottle shown in the bottom-left of Fig. 4.1, we cannot narrow down the Y-axis rotation between the two (as the second label may be on the side or the back). However, if we consider the additional third image, we can immediately understand that the rotation between the first two should be nearly 180 degrees!

We build on this insight in our proposed framework RelPose++ and develop a method for jointly reasoning over multiple images for predicting pairwise relative distributions. Specifically, we incorporate a transformer-based module that leverages context across all input images to update the image-specific features subsequently used for relative rotation inference. RelPose++ also goes beyond predicting only rotations and additionally infers the camera translation to yield 6D camera poses. A key hurdle is that the world coordinate frame used to define camera extrinsics can be arbitrary, and naive solutions to resolve this ambiguity (*e.g.* instantiating the first camera as the world origin) end up entangling predictions of camera translations with predictions of (relative) camera rotations. Instead, for roughly center-facing images, we define a world coordinate frame centered at the intersection of cameras' optical axes. We show that this helps decouple the tasks of rotation and translation prediction, and leads to clear empirical gains.

RelPose++ is trained on 41 categories from the CO3D dataset [117] and is able to recover 6D camera poses for objects from just a few images. We evaluate on seen categories, unseen categories, and even novel datasets (in a zero-shot fashion), improving rotation prediction by 10% over prior art. We also evaluate the full 6D camera poses by measuring the accuracy of the predicted camera centers (while accounting for the similarity transform ambiguity), and demonstrate the benefits of our proposed coordinate system. We also formulate a metric that decouples the accuracy of predicted camera translations and predicted rotations, which may be generally useful for future benchmarking. Finally, we show that the 6D poses from RelPose++ can be directly useful for downstream sparse-view 3D reconstruction methods.

4.2 Related Work

Pose Estimation Using Feature Correspondences. The classic SfM and SLAM pipelines for pose estimation from sets of images or video streams involve computing matches [80] between discriminative hand-crafted local features [7, 79]. These matches are used to estimate relative camera poses [78, 97], verified via RANSAC [37], and optimized via bundle adjustment [153]. Subsequent research has explored improving each of these components. Learned feature estimation [31] and feature matching [125, 23, 76] have improved robustness. This paradigm has been scaled by efficient parallelization [127, 126] and can even run in real-time for visual odometry [91, 92, 15]. While we consider a similar task of estimating camera

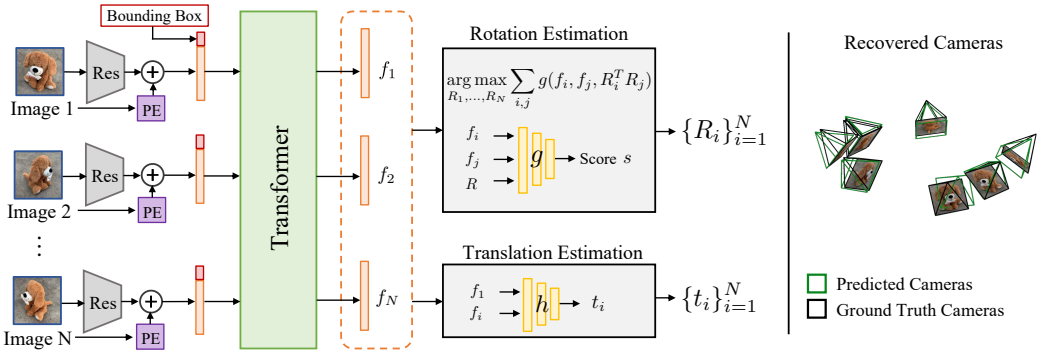


Figure 4.2: **Overview of RelPose++.** We present RelPose++, a method for sparse-view camera pose estimation. RelPose++ starts by extracting global image features using a ResNet 50. We positionally encode [162] the image index and concatenate bounding box parameters as input to a Transformer. After processing all image features jointly, we separately estimate rotations and translations. To handle ambiguities in pose, we model the distribution of rotations using an energy-based formulation, following [93, 191]. Because we predict the origin at the unique world coordinate closest to all optical axes, which is unambiguous (See Sec. 4.3.3 and Fig. 4.3), we can directly regress camera translation from the learned features. On the right, we visualize the recovered camera poses.

poses given images, our approach differs fundamentally because we do not rely on bottom-up correspondences as they cannot be reliably computed given sparse views.

Single-view 3D Pose Estimation. In the extreme case of a single image, geometric cues are insufficient for reasoning about pose, so single-view 3D pose estimation approaches rely on learned data-driven priors. A significant challenge that arises in single-view 3D is that absolute pose must be defined with respect to a coordinate system. The typical solution is to assume a fixed set of categories (*e.g.* humans [86, 58] or ShapeNet objects [19]) with pre-defined canonical coordinate systems. Related to our approach are methods that specifically handle object symmetries, which can be done by predicting multiple hypotheses [84], parameters for the antipodal Bingham distribution [105, 41], or energy [93] (similar to us). These methods predict absolute pose which is not well-defined without a canonical pose.

Because absolute poses only make sense in the context of a canonical pose, some single-view pose estimation papers have explored learning the canonical pose of objects automatically [98, 178, 143]. Other approaches bypass this issue by predicting poses conditioned on an input mesh [179, 104, 190, 5] or point-cloud [174]. In contrast, we resolve this issue by predicting relative poses from pairs of images.

Learned Multi-view Pose Estimation. Given more images, it is still possible to learn a data-driven prior rather than rely on geometric consistency cues alone.

For instance, poses can directly be predicted using an RNN [169, 149], a transformer [96], or auto-regressively [183] for SLAM and object tracking applications. However, such approaches assume temporal locality not present in sparse-view images. Other approaches have incorporated category-specific priors, particularly for human pose [60, 69, 160, 82]. In contrast, our work focuses on learning *category-agnostic* priors that generalize beyond object categories seen at training.

Most related to our approach are methods that focus on sparse-view images. Such setups are more challenging since viewpoints have limited overlap. In the case of using just 2 images for wide-baseline pose estimation, direct regression approaches [87, 120] typically do not model uncertainty or require distributions to be Gaussian [21]. [14] learns a 4D correlation volume from which distributions over relative rotations can be computed for pairs of patches. Most similar to our work is the energy-based RelPose [191], which estimates distributions over relative rotations which can be composed together given more than 2 images. We build off of this energy-based framework and demonstrate significantly improved performance by incorporating multiview context. Additionally, RelPose only predicts rotations whereas we estimate 6D pose.

To estimate poses from sparse views, FORGE [56] and SparsePose [135] both directly regress 6D poses. SparsePose also learns a bundle-adjustment procedure to refine predictions iteratively, but this refinement is complementary to our approach as it can improve any initial estimates. Similarly, the concurrent PoseDiffusion [165] models a probabilistic bundle adjustment procedure via a diffusion model in contrast to the energy-based model in our work.

4.3 Method

Given a set of N (roughly center-facing) input images $\{I_1, ..., I_N\}$ of a generic object, we wish to recover consistent 6-DoF camera poses for each image *i.e.* $\{(R_1, \mathbf{t}_1), ..., (R_N, \mathbf{t}_N)\}$, where R_i and $\mathbf{t}_i$ correspond to the rotation and translation for the i^{th} camera viewpoint.

To estimate the camera rotations, we adopt the framework proposed by RelPose [191], where a consistent set of rotations can be obtained given pairwise relative rotation distributions (Sec. 4.3.1). However, unlike RelPose which predicts these distributions using only two images, we incorporate a transformer-based module to allow the pairwise predicted distributions to capture multi-view cues (Sec. 4.3.2). We then extend this multi-view reasoning module also to infer the translations associated with the cameras, while defining a world-coordinate system that helps reduce prediction ambiguity (Sec. 4.3.3).

4.3.1 Global Rotations from Pairwise Distributions

We build on RelPose [191] for inferring consistent global rotations given a set of input images and briefly summarize the key components here. As absolute camera rotation prediction is ill-posed given the world-coordinate frame ambiguity, RelPose infers pairwise relative rotations and then obtains a consistent set of global rotations. Using an energy-based model, it first approximates the (un-normalized) log-likelihood of the pairwise relative rotations given image features f_i and f_j with an MLP $g_\theta(f_i, f_j, R_{i\to j})$ which we treat as a negative energy or score.

Given the inferred distributions over pairwise relative rotations, RelPose casts the problem of finding global rotations as that of a mode-seeking optimization. Specifically, using a greedy initialization followed by block coordinate ascent, it recovers a set of global rotations that maximize the sum of relative rotation scores:

$$\{R_1, \dots, R_N\} = \underset{\{R_i\}_{i=1}^N}{\arg\max} \sum_{i,j} g_\theta(f_i, f_j, R_i^\top R_j) \tag{4.1}$$

In RelPose, the image features are extracted using a per-frame ResNet-50 [50] encoder: $f_i = \varepsilon_\phi(I_i)$.

4.3.2 Multi-view Cues for Pairwise Distributions

Following RelPose, we similarly model the distribution of pairwise relative rotations using an energy-based model (eq. (4.1)). However, instead of only relying on the images I_i and I_j to obtain the corresponding features f_i and f_j, we propose a transformer-based module that allows for these features to depend on *other* images in the multi-view set.

Multi-view Conditioned Image Features. As illustrated in Fig. 4.2, we first use a ResNet [50] to extract per-image features. We also add an ID-specific encoding to the ResNet features and concatenate an embedding of the bounding box used to obtain the input crop from the larger image (as it may be informative about the scene scale when inferring translation). Unlike RelPose which then directly feeds these image-specific features as input to the energy prediction module, we use a transformer (similar to other recent sparse-view works [165, 96, 135]) to update these features in the context of the other images. We denote this combination of the feature extractor and transformer as a scene encoder $\mathcal{E}_\phi$, which given N input images $\{I_n\}$ outputs multi-view conditioned features $\{f_n\}$ corresponding to each image:

$$f_i = \mathcal{E}_\phi^i(I_1, \dots I_N), \quad \forall i \in \{1 \dots N\} \tag{4.2}$$

Learning Objective. Given a dataset with posed multi-view images of diverse objects, we jointly train the scene encoder $\mathcal{E}_\phi$ and the pairwise energy-based model g_θ by simply minimizing the negative log-likelihood (NLL) of the true (relative)

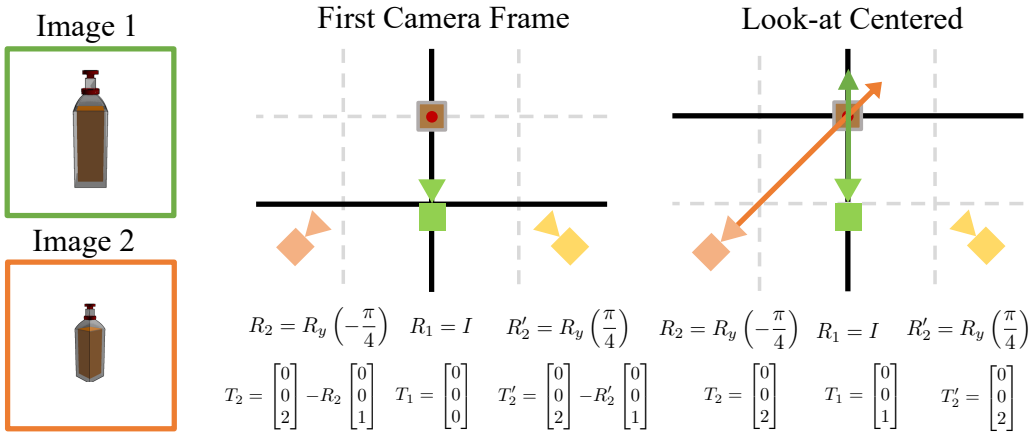


Figure 4.3: **Coordinate Systems for Estimating Camera Translation.** Given two images, consider the task of estimating their 6D poses, i.e., the R and T that transform points from the world frame to each camera's frame (**Left**). In typical SLAM setups, the world frame is centered at the first camera, but this implies the target camera translation T_2 depends on the target rotation R_2 (**Middle**). For symmetric objects where R_2 may be ambiguous, this may lead to unstable predictions for translation. Instead, for roughly center-facing cameras, a better solution is to set the world origin at the unique point closest to the optical axes of all cameras (**Right**). This helps decouple the task of predicting camera translations from rotations.

rotations [191, 93]. In particular, we randomly sample $N \in [2, 8]$ images for a training object, and minimize the NLL of the true relative rotations $R^{gt}_{i\to j}$ under our predicted distribution:

$$L_{\text{rot}} = \sum_{i,j} -\log \frac{\exp g_\theta(f_i, f_j, R^{gt}_{i\to j})}{\sum_{R'} \exp g_\theta(f_i, f_j, R')} \tag{4.3}$$

4.3.3 Predicting Camera Translations

Using the multi-view aware image features f_i, we can directly predict the per-image camera translation $\mathbf{t}_i = h_\psi(f_i)$. However, a central hurdle to learning such prediction is the inherent ambiguity in the world coordinate system. Specifically, the 'ground-truth' cameras obtained from SfM are meaningful only up to an arbitrary similarity transform [49]), and training our network to predict these can lead to incoherent training targets across each sequence. We therefore first need to define a consistent coordinate frame across training instances, so that the networks can learn to make meaningful predictions.

Geometric Interpretation of Camera Translation. Recall that the camera extrinsics $(R_i, \mathbf{t}_i)$ define a transformation of a point $\mathbf{x}^w$ in world frame to camera frame $\mathbf{x}^c_i =$

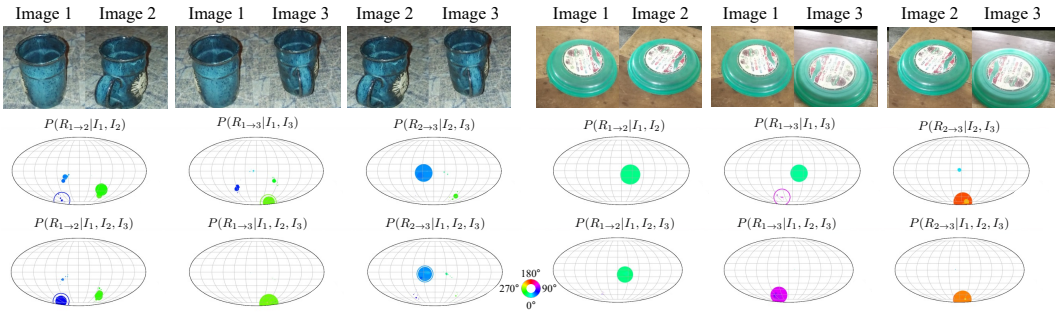


Figure 4.4: **Resolving Pose Ambiguity with More Images.** The relative rotation between only two views may be ambiguous for highly symmetric objects such as cups, frisbees, and apples. Often, seeing a third view will provide enough additional context to the scene to determine the correct relative rotation. When images are shown to the model in three separate pairs, as denoted by $P(R_{i\to j}|I_i, I_j)$, the output probability distribution may have more than one mode due to the symmetry of the object, but when shown all three images together to predict $P(R_{i\to j}|I_1, I_2, I_3)$, the model has a significantly more confident prediction. Following [191], we visualize distributions over relative rotations by projecting the rotation matrix such that the x-axis represents the yaw, the y-axis represents the pitch, and the color represents the roll. The size of each circle corresponds to probability, and rotations with negligible probability are filtered. The ground truth rotation is denoted by the unfilled circle.

$R_i\mathbf{x}^w + \mathbf{t}_i$. The translation $\mathbf{t}_i$ is therefore the location of the world origin in each camera's coordinate frame (and not the location of the camera in the world frame!). We can also see that an arbitrary rotation of the world coordinate system ($\bar{\mathbf{x}}^w = \Delta R\ \mathbf{x}^w$), does not affect the per-camera translations and that only the location of the chosen world origin (and the scaling) are relevant factors. To define a consistent coordinate frame for predicting rotations, we must therefore decide where to place the world origin and how to choose an appropriate scale.

Look-at Centered Coordinate System. One convenient choice, often also adopted by SfM/SLAM approaches [126, 27], is to define the world coordinate system as centered on the first camera (denoted as 'First Camera Frame'). Unfortunately, the per-camera translations in this coordinate frame entangle the relative rotations between cameras (as $\mathbf{t}_i$ is the location of the first camera in the i^{th} camera's frame). As illustrated in Fig. 4.3, ambiguity in estimating this relative transformation for (*e.g.*, symmetric) objects can lead to uncertainty in the translation prediction.

Instead, we argue that for roughly center-facing captures, one should define the unique point closest to the optical axes of the input cameras as the world origin. Intuitively, this is akin to setting the 'object center' as the world origin, and the translation then simply corresponds to the inference of where the object is in the

camera frame (and this remains invariant even if one is unsure of camera rotation as illustrated in Fig. 4.3). However, instead of relying on a semantically defined 'object center' which may be ambiguous given partial observations, the closest approach point across optical axes is a well-defined geometric proxy. Finally, to resolve scale ambiguity, we assume that the first camera is a unit distance away from this point.

Putting it Together. In addition to the energy-based predictor (Eq. 4.1), we also train a translation prediction module that infers the per-camera $\mathbf{t}_i = h_\psi(f_1, f_i) \in \mathbb{R}^3$ given the multi-view features. Because we normalize the scene such that $\|\mathbf{t}_1\| = 1$, we provide h_ψ with the first image feature f_1. To define the target translations for training, we use the ground-truth cameras (SfM) $\{(R_i, \mathbf{t}_i)\}$ to first identify the point $\mathbf{c}$ closest to all the optical axes. We can then transform the world coordinate to be centered at $\mathbf{c}$, thus obtaining the target translations as $\bar{\mathbf{t}}_i = s(\mathbf{t}_i - R_i\mathbf{c})$, where the scale s ensures a unit norm for $\bar{\mathbf{t}}_1$. For this training, we simply use an L1 loss between the target and predicted translations:

$$L_{\text{trans}} = \|h_\psi(f_i, f_1) - \bar{\mathbf{t}}_i\|_1 \tag{4.4}$$

Together with the optimized global rotations, these predicted translations yield 6-DoF cameras given a sparse set of input images at inference.

4.4 Evaluation

4.4.1 Experimental Setup

Dataset. We train and test our models on the CO3D [117] (v2) dataset, which consists of turntable-style video sequences across 51 object categories. Each video sequence is associated with ground truth camera poses acquired using COLMAP [126]. Following [191], we train on 41 object categories and hold out the same 10 object categories to evaluate generalization. After filtering for the camera pose score, we train on a total of 22,375 sequences with 2,212,952 images.

Task and Metrics. We randomly sample $2 \leq N \leq 8$ center-cropped images $\{I_n\}$ from each test sequence. Given these as input, each approach then infers a set of global 6-DoF camera poses $\{R_i, \mathbf{t}_i\}$ corresponding to each input image. To evaluate these predictions, we report accuracy under various complementary metrics, all of which are invariant under global similarity transforms for the prediction/ground-truth cameras. To reduce variance in metrics, we re-sample the N images from each test sequence 5 times and compute the mean.

Rotation Accuracy. We evaluate relative rotation error between every pair of predicted and ground truth rotations. Following [191, 135], we report the proportion of pose errors less than 15 degrees.

Camera Center Accuracy. Following standard benchmarks in SLAM [142] that evaluated recovered poses using camera localization error, we measure the accuracy

	# of Images	2	3	4	5	6	7	8
Seen Cate.	COLMAP (SP+SG) [124]	30.7	28.4	26.5	26.8	27.0	28.1	30.6
	RelPose [191]	56.0	56.5	57.0	57.2	57.2	57.3	57.2
	PoseDiffusion [165]	75.2	76.6	77.0	77.3	77.7	78.2	78.5
	Pose Regression	49.1	50.7	53.0	54.6	55.7	56.1	56.5
	Ours (N=2)	**81.8**	82.3	82.7	83.2	83.3	83.5	83.6
	Ours (Full)	**81.8**	**82.8**	**84.1**	**84.7**	**84.9**	**85.3**	**85.5**
Unseen Cate.	COLMAP (SP+SG) [124]	34.5	31.8	31.0	31.7	32.7	35.0	38.5
	RelPose [191]	48.6	47.5	48.1	48.3	48.4	48.4	48.3
	PoseDiffusion [165]	60.0	64.8	64.6	65.8	65.7	66.6	67.8
	Pose Regression	42.7	43.8	46.3	47.7	48.4	48.9	48.9
	Ours (N=2)	**69.8**	69.6	70.1	69.8	70.4	70.5	71.2
	Ours (Full)	**69.8**	**71.1**	**71.9**	**72.8**	**73.8**	**74.4**	**74.9**

Table 4.1: **Joint Rotation Accuracy @ 15°.** We measure the relative angular error between pairs of relative predicted and ground truth rotations . We report the proportion of angular errors within 15 degrees and report accuracies for varying thresholds in the supplement. With more images, our method surpasses the ablation that only looks at 2 images (N=2), showing the benefit of context.

	# of Images	2	3	4	5	6	7	8
Seen Cate.	COLMAP (SP+SG) [124]	100	35.8	26.1	21.6	18.9	18.3	19.2
	PoseDiffusion [165]	100	86.6	80.5	77.2	75.9	74.4	73.7
	Pose Reg. (First Fr.)	100	87.6	81.2	77.6	75.8	74.5	73.6
	Pose Reg. (Our Fr.)	100	90.3	84.6	81.5	80.0	78.5	77.7
	Ours	100	**92.3**	**89.1**	**87.5**	**86.3**	**85.9**	**85.5**
Unseen Cate.	COLMAP (SP+SG) [124]	100	37.9	29.3	24.7	23.1	23.5	25.3
	PoseDiffusion [165]	100	78.0	65.8	61.3	57.0	54.4	55.1
	Pose Reg. (First Fr.)	100	78.8	71.4	66.3	63.6	61.8	60.4
	Pose Reg. (Our Fr.)	100	**82.8**	74.0	70.0	67.8	65.8	65.3
	Ours	100	82.5	**75.6**	**71.9**	**69.9**	**68.5**	**67.5**

Table 4.2: **Camera Center Accuracy @ 0.2.** We report the proportion of camera centers that are within 20% of the scene scale to the ground truth camera centers. We align the predicted and ground truth camera centers using an optimal 7-DoF similarity transform (hence all methods are at 100% for N=2 and performance appears to drop with more images as there are more constraints).

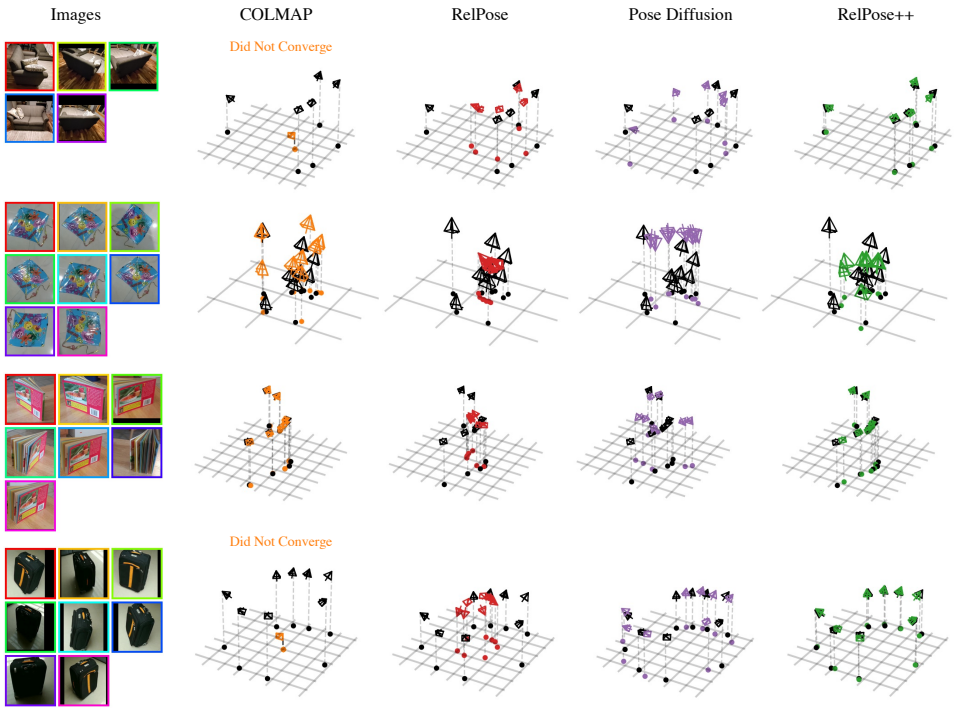


Figure 4.5: **Qualitative Results of Recovered Camera Trajectories.** We compare our approach with COLMAP, RelPose, and PoseDiffusion. Since RelPose does not predict translations, we set the translations to be unit distance from the scene center. We visualize predicted camera trajectories in color and the ground truth in black, aligned using a Procrustes optimal alignment on the camera centers. We find that COLMAP is accurate but brittle, converging only occasionally when the object has highly discriminative features and sufficient overlap between images. RelPose, while mostly accurate, usually makes 1-2 mistakes per sequence which causes misalignment. PoseDiffusion is generally accurate but struggles sometimes with symmetry. We find that our method consistently outperforms the baselines.

of the predicted camera centers. However, as the predicted centers $\mathbf{c}_i = -R_i\mathbf{t}_i$ may be in a different coordinate system from the SfM camera centers $\mathbf{c}_i^{gt}$, we first compute the optimal similarity transform to align the predicted centers with the ground-truth [158]. Following [135], we then report the proportion of predicted camera centers within 20% of the scale of the scene in Tab. 4.2, where the scale is defined as the distance from the centroid of the ground truth camera centers to the furthest camera center.

Baselines. We compare our approach with state-of-the-art correspondence-based and learning-based methods:

COLMAP (SP+SG) [127, 126]. This represents a state-of-the-art SFM pipeline

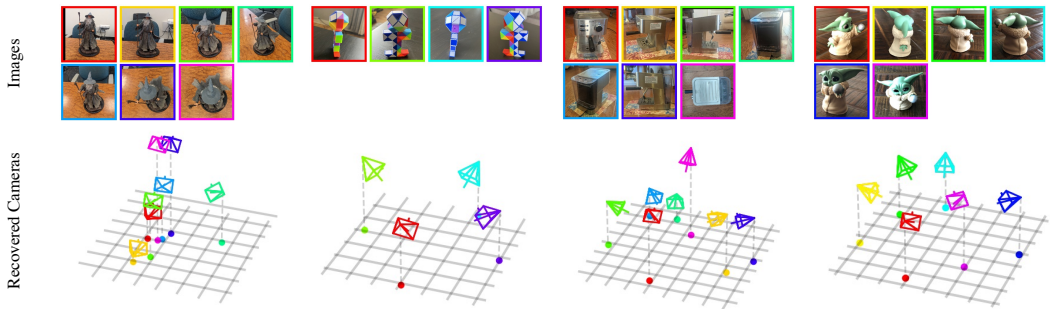


Figure 4.6: **Recovered Camera Poses from In-the-Wild Images.** We find that RelPose++ generalizes well to images outside of the distribution of CO3D object categories. Here, we demonstrate that RelPose++ can recover accurate camera poses even for self-captures of Gandalf the Grey, a Rubrik snake, an espresso machine, and Grogu. RelPose++ can capture challenging rotations and translations, including top-down poses, varying distances from the camera, and in-plane rotations (see Gandalf).

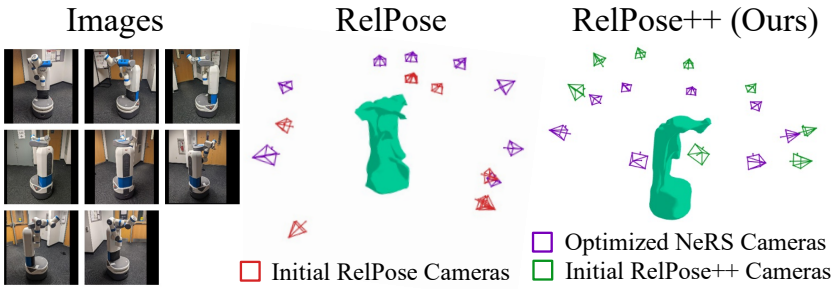


Figure 4.7: **Sparse-view 3D Reconstruction using NeRS.** We find that the camera poses estimated by our method are sufficient as initialization for 3D reconstruction. We compare our recovered cameras (green) with RelPose cameras (red) as initialization to NeRS. NeRS jointly optimizes these cameras and shape. We visualize the cameras at the end of the NeRS optimization in purple. We find that our cameras enable higher-fidelity 3D reconstruction.

(COLMAP) that uses SuperPoint features [31] with SuperGlue matching [125]. We use the implementation provided by HLOC [124].

RelPose [191]. We evaluate RelPose, which also uses a pairwise energy-based scoring network. As this only predicts rotations, we exclude it from translation evaluation.

PoseDiffusion [165]. PoseDiffusion is a concurrent work that combines diffusion with geometric constraints (from correspondences) to infer sparse-view poses probabilistically. All evaluations are with the geometry-guided sampling.

Variants. We also report comparisons to variants of our approach to highlight

		Cam. Cen.			Transl.		
	# of Images	3	5	8	3	5	8
Seen	Ours	92.3	87.5	85.5	90.5	87.8	86.2
	Constant	91.3	84.7	81.1	69.0	60.8	56.5
Uns.	Ours	82.5	71.9	67.5	79.7	74.9	73.6
	Constant	81.4	69.5	63.8	60.3	52.2	48.2

Table 4.3: **Analyzing Translation Prediction.** We quantify the improvements of our predicted translations over a naive baseline that predicts center-facing cameras located at a unit distance from the origin. Because the camera center entangles the rotation and translation prediction, we compute an additional translation accuracy that reports the fraction of translations within 0.1 of the scene scale of the ground truth translation after applying a scaling and world origin alignment (see supplement).

	Rotation			Cam. Cen.		
# of Images	3	5	8	3	5	8
MediaPipe [81]	52.3	52.8	52.7	74.5	59.1	49.9
PoseDiffusion [165]	69.2	68.0	70.0	87.2	73.8	67.2
Ours	**75.8**	**76.6**	**77.0**	**91.6**	**83.9**	**77.6**

Table 4.4: **Evaluating Zero-shot Generalization on Objectron on Rotation (@ 15°) and Camera Center (@ 0.2) Accuracy.** We evaluate our approach, trained on CO3D, on Objectron without any fine-tuning.

the benefits of the energy-based prediction, multi-view reasoning, and proposed translation coordinate frame.

Pose Regression. This corresponds to a regression approach that uses our ResNet and Transformer architecture to directly predict the global rotations (assuming the first camera has identity rotation) and translations. This rotation prediction is analogous to the initial regressor in SparsePose [135] (unfortunately, due to licensing issues, we were unable to obtain code/models for direct comparison). We consider variants that regress translations using the first camera frame and our look-at-centered coordinate frame.

Ours (N=2). This represents a variant that only has access to 2 images at a time when inferring pairwise rotation distributions. While similar to RelPose, it helps disambiguate the benefits of our transformer-based architecture.

First-frame Centered Regression. Instead of using the Look-at centered world frame, this variant defines camera translations using the first camera as world origin (while

using the same scaling as the Look-at centered system).

4.4.2 Quantitative Results

Accuracy of Recovered 6D Poses. We evaluate rotation accuracy in Tab. 4.1. We find that our approach significantly outperforms COLMAP and RelPose. While COLMAP performs well at fine error thresholds (see the supplement), it frequently does not converge in sparse-view settings because wide baselines do not provide enough overlap to compute useful correspondences. We find that the Pose Regression baseline performs poorly, suggesting that modeling uncertainty is important for sparse-view settings. The jump in performance from RelPose to our $N = 2$ variant suggests that the increased capacity of our transformer architecture is important. Finally, we find that our model starts out at a similar performance to the $N = 2$ model but quickly outperforms it for larger N, suggesting that the image context is important. Our method also consistently achieves better localization than PoseDiffusion.

We evaluate the camera center accuracy in Tab. 4.2 and also report AUC metrics in the supplement. COLMAP performs poorly since it often fails to converge. We find that the First Camera Frame Regression has the worst generalization to unseen object categories (see Tab. 4.3). This makes sense because the predicted translation must also account for any errors in the predicted rotation, which likely occur in a different distribution than seen for training categories.

Analyzing Translation Predictions. While the focus of our work is on roughly center-facing images of objects as these captures most closely resemble a typical object scanning pipeline, we do find that CO3D deviates significantly from perfectly circular trajectories. We quantify this using an additional baseline that uses our rotation predictions but always predicts a constant [0, 0, 1] translation (which would be optimal for center-facing cameras on a sphere). In addition to camera center evaluation which conflates the predicted rotation and translation, we propose a translation evaluation that computes the proportion of predicted translations that are within 10% of the ground truth translation. Similar to the camera center evaluation, we apply an optimal similarity transform that accounts for the scene scaling and world origin placement (see supplement for more details). We find that our method significantly outperforms the constant translation baseline using both the camera center and translation metrics in Tab. 4.3 (reducing translation error from 51.8% to 26.4%).

Evaluating Generalization. We evaluate zero-shot generalization on Objectron [4] in Tab. 4.4, and find that our approach outperforms PoseDiffusion [165]. We also report the accuracy of relative poses recovered from a per-frame 6D pose estimation method MediaPipe [81]. Note that both our model and PoseDiffusion are trained only on CO3D with no finetuning while MediaPipe is trained per category on Objectron. Following PoseDiffusion, we also evaluate generalization via zero-shot transfer to RealEstate10K [199] and outperform their method, but this scene-level front-facing

dataset is not an ideal testbed for testing generalization from 360-degree object-centric data as even a naive baseline (fixed identity rotation) performs well (see supplement).

4.4.3 Qualitative Results

Visualization for Co3D Predictions. We compared recovered camera poses from sparse-view images using our method with COLMAP (with SuperPoint/SuperGlue) and RelPose in Fig. 4.5. We find that our method is able to recover more accurate cameras than RelPose consistently. While COLMAP recovers highly accurate trajectories when it succeeds, it usually fails to converge for sparse images.

We also visualize the effect of increasing image context on pairwise rotation distributions in Fig. 4.4. Given just two images, the relative pose is often ambiguous, but we find that this ambiguity can be resolved by conditioning on more images using our transformer.

In-the-wild Generalization and 3D Reconstruction. We demonstrate the generalization of RelPose++ on in-the-wild captures in Fig. 4.6. These recovered cameras are sufficient to enable 3D reconstruction using NeRS [192], a representative sparse-view reconstruction method (Fig. 4.7).

4.5 Discussion

We presented RelPose++, a system for inferring a consistent set of 6D poses (rotations and translations) given a sparse set of input views. While it can robustly infer camera poses, these are not as precise as ones obtained from classical methods, and can be improved further via refinement [135, 74]. Secondly, while the energy-based models can efficiently capture uncertainty, they are inefficient to sample from and are limited to pairwise distributions, and it may be possible to instead leverage diffusion models to overcome these limitations. Lastly, while we demonstrated that our estimated poses can enable downstream 3D reconstruction, it would be beneficial to develop unified approaches that jointly tackle the tasks of reconstruction and pose inference.

Chapter 5

Cameras as Rays: Pose Estimation via Ray Diffusion

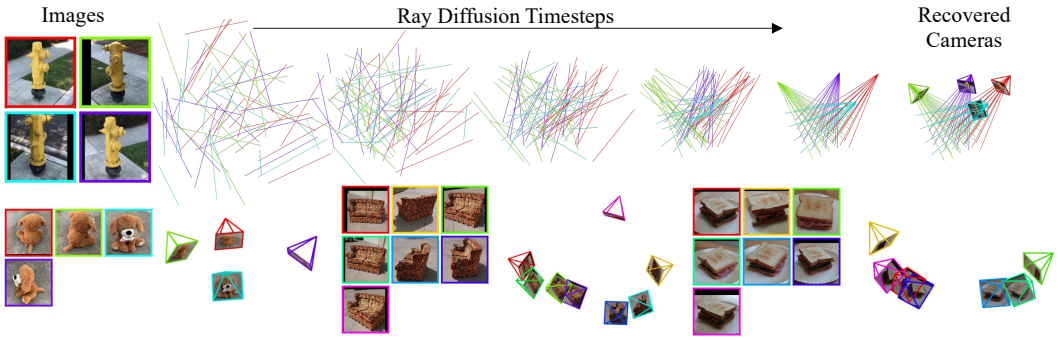


Figure 5.1: **Recovering Sparse-view Camera Parameters by Denoising Rays.** *Top:* Given sparsely sampled images, our approach learns to denoise camera rays (represented using Plücker coordinates). We then recover camera intrinsics and extrinsics from the positions of the rays. *Bottom:* We demonstrate the generalization of our approach for both seen (teddybear) and unseen object categories (couch, sandwich).

5.1 Introduction

Recent learning-based approaches have examined the task of predicting cameras given a sparse set of input images, and investigated regression [56, 135], energy-based modeling [191, 73] and denoising diffusion [165] for inference. However, while exploring a plethora of learning techniques, these methods have largely side-stepped a crucial question: *what representation of camera poses should learning-based methods predict*?

At first, there may seem to be an obvious answer. After all, every student of projective geometry is taught that (extrinsic) camera matrices are parameterized with a single rotation and a translation. Indeed, all of the above-mentioned methods adapt this representation (albeit with varying rotation parametrizations *e.g.* matrices, quaternions, or angles) for predicting camera poses. However, we argue that such a parsimonious *global* pose representation maybe suboptimal for neural learning, which often benefits from over-parameterized *distributed* representations. From a geometric perspective, classical bottom-up methods benefit from low-level correspondence across pixels/patches, while learning-based methods that predict global camera representations may not easily benefit from such (implicit or explicit) associations.

In this work, we propose an alternate camera parametrization that recasts the task of pose inference as that of patch-wise ray prediction (Fig. 5.1). Instead of predicting a global rotation and global translation for each input image, our model predicts a separate ray passing through each patch in each input image. We show that this representation is naturally suited for transformer-based set-to-set inference models that process sets of features extracted from image patches. To recover the camera extrinsics ($\boldsymbol{R}$, $\boldsymbol{t}$) and intrinsics ($\boldsymbol{K}$) corresponding to a classical perspective camera, we optimize a least-square objective given the predicted ray bundle. It is worth noting that the predicted ray bundle itself can be seen as an encoding of a *generic camera* as introduced in [46], which can capture non-perspective cameras such as catadioptric imagers or orthographic cameras whose rays may not even intersect at a center of projection.

We first illustrate the effectiveness of our distributed ray representation by training a patch-based transformer with a standard regression loss. We show that this already surpasses the performance of state-of-the-art pose prediction methods that tend to be much more compute-heavy [73, 135, 165]. However, there are natural ambiguities in the predicted rays due to symmetries and partial observations [191, 165]. We extend our regression-based method to a denoising diffusion-based probabilistic model and find that this further improves the performance and can recover distinct distribution modes. We demonstrate our approach on the CO3D dataset [117] where we systematically study performance across seen categories as well as generalization to unseen ones. Moreover, we also show that our approach can generalize even to unseen datasets and present qualitative results on in-the-wild self-captures. In summary, our contributions are as follows:

- We recast the task of pose prediction as that of inferring per-patch ray equations as an alternative to the predominant approach of inferring global camera parametrizations.

- We present a simple regression-based approach for inferring this representation given sparsely sampled views and show even this simple approach surpasses the state-of-the-art.

- We extend this approach to capture the distribution over cameras by learning a denoising diffusion model over our ray-based camera parametrization, leading to further performance gains.

5.2 Related Work

5.2.1 Structure-from-Motion and SLAM

Both Structure-from-motion and SLAM aim to recover camera poses and scene geometry from a large set of unordered or ordered images. Classic SfM [139] and indirect SLAM [91, 92, 15] methods generally rely on finding correspondences [80] between feature points [7, 79] in overlapping images, which are then efficiently optimized [126, 127] into coherent poses using Bundle Adjustment [153]. Subsequent work has improved the quality of features [31], correspondences [133, 181, 125], and the bundle adjustment process itself [148, 75]. On the contrary, rather than minimize geometric reconstruction errors, direct SLAM methods [27, 129] optimize photometric errors. While the methods described in this section can achieve (sub)pixel-perfect accuracy, their reliance on dense images is unsuitable for sparse-view pose estimation.

5.2.2 Pose Estimation from Sparsely Sampled Views

Estimating poses from sparsely sampled images (also called sparse-view or wide-baseline pose estimation in prior work) is challenging as methods cannot rely on sufficient (or even any) overlap between nearby images to rely on correspondences. The most extreme case of estimating sparse-view poses is recovering the relative pose given 2 images. Recent works have explored how to effectively regress relative poses [6, 120, 14] from wide-baseline views. Other works have explored probabilistic approaches to model uncertainty when predicting relative pose [191, 21].

Most related to our approach are methods that can predict poses given multiple images. RelPose [191] and RelPose++ [73] use energy-based models to compose relative rotations into sets of camera poses. SparsePose [135] learns to iteratively refine sparse camera poses given an initial estimate, while FORGE [56] exploits synthetic data to learn camera poses. The most comparable to us is PoseDiffusion [165], which also uses a diffusion model to denoise camera poses. However, PoseDiffusion denoises the camera parameters directly, whereas we denoise camera rays which we demonstrate to be more precise. Concurrently to our work, PF-LRM [167] and DUSt3R [170] predict sparse poses by predicting pixel-aligned pointclouds (as opposed to rays in our work) and using PnP to recover cameras.

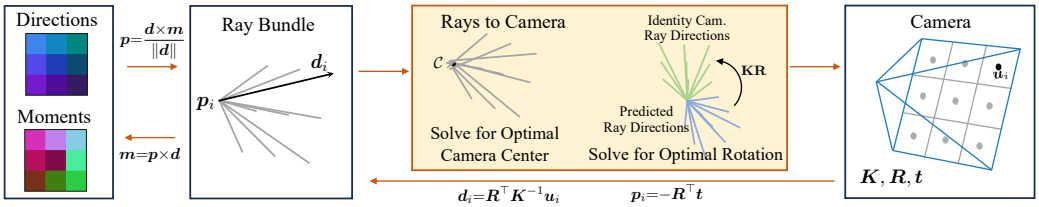


Figure 5.2: **Converting Between Camera and Ray Representations.** We represent cameras as a collection of 6-D Plücker rays consisting of directions and moments. We convert the traditional representation of cameras to the ray bundle representation by unprojecting rays from the camera center to pixel coordinates. We convert rays back to the traditional camera representation by solving least-squares optimizations for the camera center, intrinsics matrix, and rotation matrix. See Sec. 5.3.1 for more details.

5.2.3 Ray-based Camera Parameterizations

Prior work in calibrating generic camera representations has used ray-based representations of cameras, mainly for fish-eyed lenses for which the pinhole model is not a good approximation [61]. [46, 33] consider the most general camera model, where each pixel projection is modeled by its ray. Even with better algorithms [128], the large number of parameters in these camera models makes calibration difficult. Although these works also make use of ray-based camera representations, their focus is on calibration (intrinsics) and require known calibration patterns. Neural Ray Surfaces [161] considers learning the poses of generic cameras but does so from video rather than sparse views.

Parameterizing viewpoints using camera rays is also commonly used in the novel view synthesis community. Rather than render a full image at once, the pixel-wise appearance is conditioned per ray [89, 136, 172] given known cameras. In contrast, we aim to recover the camera itself.

5.3 Method

Our aim is to recover cameras from a sparse set of images $\{I_1, \ldots, I_N\}$. Rather than predict global camera parametrizations directly as done in previous work, we propose a ray-based representation that can be seamlessly converted to and from the classic representation (Sec. 5.3.1). We then describe a regression-based architecture to predict ray-based cameras in Sec. 5.3.2. We build on this architecture to introduce a probabilistic framework that estimates the rays using diffusion to handle uncertainties and symmetries that arise from sparsely sampled views in Sec. 5.3.3.

5.3.1 Representing Cameras with Rays

Distributed Ray Representation. Typically, a camera is parameterized by its extrinsics (rotation $\boldsymbol{R} \in \mathrm{SO}(3)$, translation $\boldsymbol{t} \in \mathbb{R}^3$) and intrinsics matrix $\boldsymbol{K} \in \mathbb{R}^{3\times3}$. Although this parameterization compactly relates the relationship of world coordinates to pixel coordinates using camera projection ($\boldsymbol{u} = \boldsymbol{K}[\boldsymbol{R} \mid \boldsymbol{T}]\boldsymbol{x}$), we hypothesize that it may be difficult for a neural network to directly regress this low-dimensional representation. Instead, inspired by generalized camera models [46, 128] used for calibration, we propose to *over-parameterize* a camera as a collection of rays:

$$\mathcal{R} = \{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_m\}, \tag{5.1}$$

where each ray $\boldsymbol{r}_i \in \mathbb{R}^6$ is associated with a known pixel coordinate $\boldsymbol{u}_i$. We parameterize each ray $\boldsymbol{r}$ traveling in direction $\boldsymbol{d} \in \mathbb{R}^3$ through any point $\boldsymbol{p} \in \mathbb{R}^3$ using Plücker corodinates [112]:

$$\boldsymbol{r} = \langle \boldsymbol{d}, \boldsymbol{m} \rangle \in \mathbb{R}^6, \tag{5.2}$$

where $\boldsymbol{m} = \boldsymbol{p} \times \boldsymbol{d} \in \mathbb{R}^3$ is the moment vector, and importantly, is agnostic to the specific point on the ray used to compute it. When $\boldsymbol{d}$ is of unit length, the norm of the moment $\boldsymbol{m}$ represents the distance from the ray to the origin.

Converting from Camera to Ray Bundle. Given a known camera and a set of 2D pixel coordinates $\{\boldsymbol{u}_i\}_m$, the directions $\boldsymbol{d}$ can be computed by unprojecting rays from the pixel coordinates, and the moments $\boldsymbol{m}$ can be computed by treating the camera center as the point $\boldsymbol{p}$ since all rays intersect at the camera center:

$$\boldsymbol{d} = \boldsymbol{R}^\top \boldsymbol{K}^{-1} \boldsymbol{u}, \qquad \boldsymbol{m} = (-\boldsymbol{R}^\top \boldsymbol{t}) \times \boldsymbol{d}. \tag{5.3}$$

In practice, we select the points $\{\boldsymbol{u}_i\}_m$ by uniformly sampling points on a grid across the image or image crop, as shown in Fig. 5.2. This allows us to associate each patch in the image with a ray passing through the center of the patch, which we will use later to design a patch- and ray-conditioned architecture.

Converting from Ray Bundle to Camera. Given a collection of rays $\mathcal{R} = \{\boldsymbol{r}_i\}_m$ associated with 2D pixels $\{\boldsymbol{u}_i\}_m$, we show that one can recover the camera extrinsics and intrinsics. We start by solving for the camera center $\mathbf{c}$ by finding the 3D world coordinate closest to the intersection of all rays in $\mathcal{R}$:

$$\boldsymbol{c} = \underset{\boldsymbol{p}\in\mathbb{R}^3}{\arg\min} \sum_{\langle \boldsymbol{d}, \boldsymbol{m} \rangle \in \mathcal{R}} \|\boldsymbol{p} \times \boldsymbol{d} - \boldsymbol{m}\|^2. \tag{5.4}$$

To solve for the rotation $\boldsymbol{R}$ (and intrinsics $\boldsymbol{K}$) for each camera, we can solve for the optimal homography matrix $\boldsymbol{P}$ that transforms per-pixel ray directions from the predicted ones to those of an 'identity' camera ($\boldsymbol{K} = \boldsymbol{I}$ and $\boldsymbol{R} = \boldsymbol{I}$):

$$\boldsymbol{P} = \underset{\|\boldsymbol{H}\|=1}{\arg\min} \sum_{i=1}^{m} \|\boldsymbol{H}\mathbf{d}_i \times \mathbf{u}_i\| \,. \tag{5.5}$$

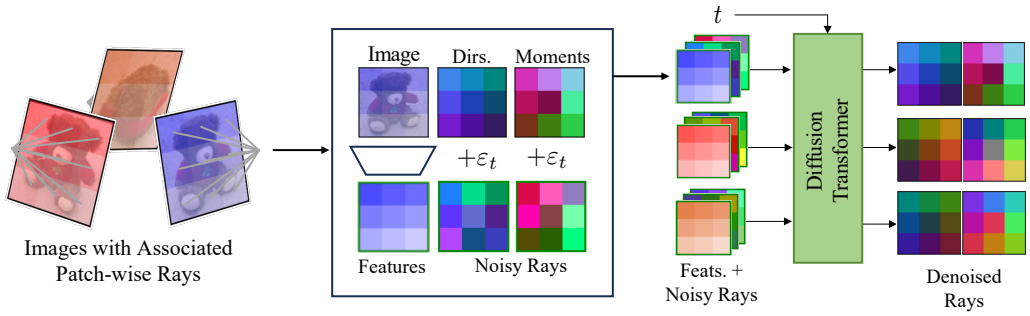


Figure 5.3: **Denoising Ray Diffuser Network**. Given a noisy ray corresponding to an image patch, our denoising ray diffusion model predicts the denoised ray. We concatenate spatial image features [106] with noisy rays, represented with 6-dimensional Plücker coordinates [112] that are visualized as 3-channel direction maps and 3-channel moment maps. We use a transformer to jointly process all image patches and associated noisy rays to predict the original denoised rays.

The matrix $\boldsymbol{P}$ can be computed via DLT [2] and can allow recovering $\boldsymbol{R}$ using RQ-decomposition as $\boldsymbol{K}$ is an upper-triangular matrix and $\boldsymbol{R}$ is orthonormal. Once the camera rotation $\boldsymbol{R}$ and camera center $\boldsymbol{c}$ are recovered, the translation $\boldsymbol{t}$ can be computed as $\boldsymbol{t} = -\boldsymbol{R}\boldsymbol{c}$.

5.3.2 Pose Estimation via Ray Regression

We now describe an approach for predicting the ray representation outlined in Sec. 5.3.1 for camera pose estimation given N images $\{I_1, \ldots, I_N\}$. Given ground truth camera parameters, we can compute the ground truth ray bundles $\{\mathcal{R}_1, \ldots, \mathcal{R}_N\}$. As shown in Fig. 5.2, we compute the rays over a uniform $p \times p$ grid over the image such that each ray bundle consists of $m = p^2$ rays (eq. (5.1)).

To ensure a correspondence between rays and image patches, we use a spatial image feature extractor and treat each patch feature as a token:

$$f_{\text{feat}}(I) = \boldsymbol{f} \in \mathbb{R}^{p \times p \times d}. \tag{5.6}$$

To make use of the crop parameters, we also concatenate the pixel coordinate $\boldsymbol{u}$ (in normalized device coordinates with respect to the uncropped image) to each spatial feature. We use a transformer-based architecture ([32, 110]) that jointly processes each of the p^2 tokens from N images, and predicts the ray corresponding to each patch:

$$\{\hat{\mathcal{R}}\}_{i=1}^{N} = f_{\text{Regress}}\left(\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^{N \cdot p^2}\right). \tag{5.7}$$

We train the network by computing a reconstruction loss on the predicted camera

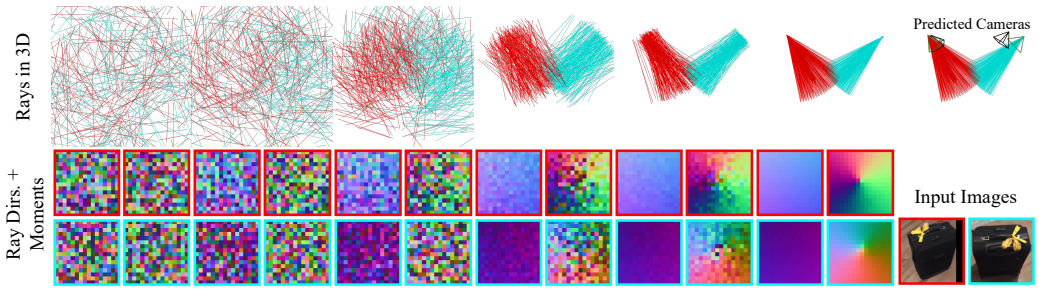


Figure 5.4: **Visualizing the Denoising Process Using Our Ray Diffuser.** Given the 2 images of the suitcase (*Bottom Right*), we visualize the denoising process starting from randomly initialized camera rays. We visualize the noisy rays using the Plücker representation (ray directions and moments) in the bottom row and their corresponding 3D positions in the top row. In the rightmost column, we recover the predicted cameras (green) and compare them to the ground truth cameras (black).

rays:

$$\mathcal{L}_{\text{recon}} = \sum_{i=1}^{N} \left\| \hat{\mathcal{R}}_i - \mathcal{R}_i \right\|_2^2. \tag{5.8}$$

5.3.3 Pose Estimation via Denoising Ray Diffusion

While the patchwise regression-based architecture described in Sec. 5.3.2 can effectively predict our distributed ray-based parametrization, the task of predicting poses (in the form of rays) may still be ambiguous given sparse views. To handle inherent uncertainty in the predictions (due to symmetries and partial observations), we extend the previously described regression approach to instead learn a diffusion-based probabilistic model over our distributed ray representation.

Denoising diffusion models [52] approximate a data likelihood function by inverting a noising process that adds time-dependent Gaussian noise to the original sample x_0:

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, \tag{5.9}$$

where $\epsilon \sim \mathcal{N}(0, \boldsymbol{I})$ and α_t is a hyper-parameter schedule of noise weights such that x_T can be approximated as a standard Gaussian distribution. To learn the reverse process, one can train a denoising network f_θ to predict the denoised sample $\boldsymbol{x}_0$ conditioned on $\boldsymbol{x}_t$:

$$\mathcal{L}(\theta) = \mathbb{E}_{t, \boldsymbol{x}_0, \epsilon} \left\| x_0 - f_\theta \left(x_t, t \right) \right\|^2. \tag{5.10}$$

We instantiate this denoising diffusion framework to model the distributions over patchwise rays conditioned on the input images. We do this by simply modifying our ray regression network from Sec. 5.3.2 to be additionally conditioned on noisy

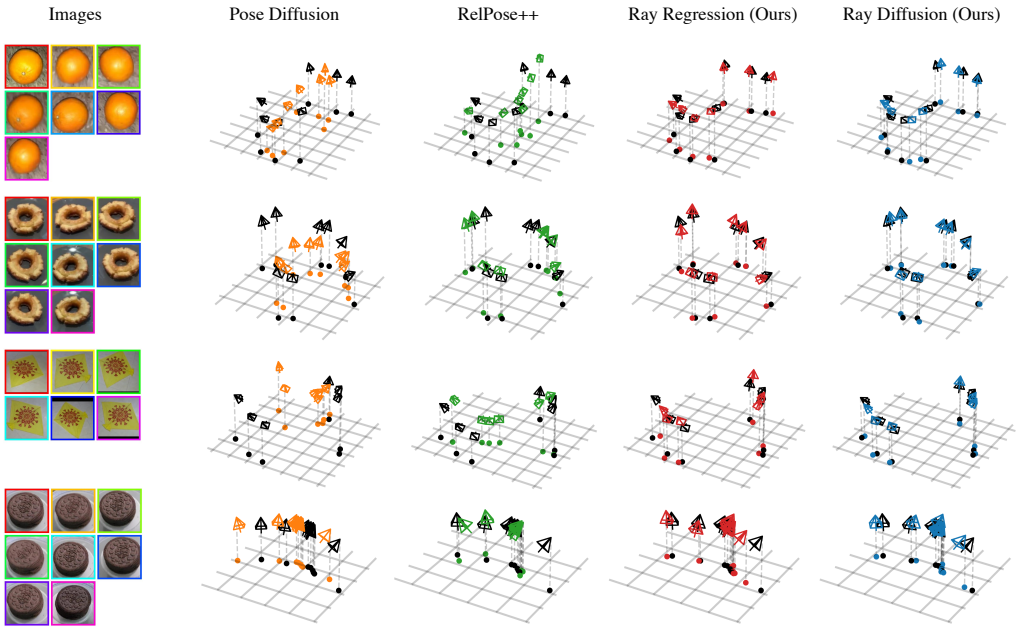


Figure 5.5: **Qualitative Comparison Between Predicted Camera Poses.** We compare the results of our regression and diffusion approaches with PoseDiffusion and RelPose++. Ground truth (black) camera trajectories are aligned to the predicted (colored) camera trajectories by performing Procrustes optimal alignment on the camera centers. The top two examples are from seen categories, and the bottom two are from held out categories.

rays (concatenated with patchwise features and pixel coordinates) and a positionally encoded [162] time embedding t:

$$\{\hat{\mathcal{R}}\}_{i=1}^{N} = f_{\text{Diffuse}}\left(\{(\boldsymbol{f}_i, \boldsymbol{u}_i, \boldsymbol{r}_{i,t})\}_{i=1}^{N\cdot p^2}, t\right), \tag{5.11}$$

where the noisy rays $\boldsymbol{r}_{i,t}$ can be computed as:

$$\boldsymbol{r}_{i,t} = \sqrt{\bar{\alpha}_t}\boldsymbol{r}_i + \sqrt{1-\bar{\alpha}_t}\epsilon. \tag{5.12}$$

Conveniently, our time-conditioned ray denoiser can be trained with the same L2 loss function (eq. (5.8)) as our ray regressor. We visualize the states of the denoised rays during backward diffusion in Fig. 5.4.

5.3.4 Implementation Details

Following [73], we place the world origin at the point closest to the optical axes of the training cameras, which represents a useful inductive bias for center-facing camera

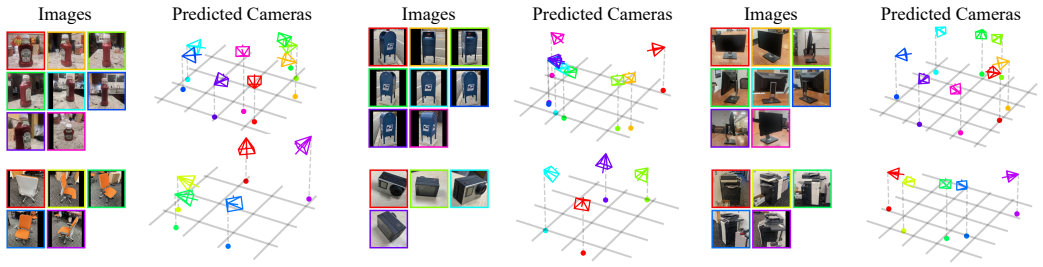


Figure 5.6: **Generalization to In-the-wild Self-captures.** We test the generalization of our ray diffusion model on a variety of *self-captured data* on objects that are not in CO3D.

setups. To handle coordinate system ambiguity, we rotate the world coordinates such that the first camera always has identity rotation and re-scale the scene such that the first camera translation has unit norm. Following prior work [191], we take square image crops tightly around the object bounding box and adjust the uniform grid of pixel coordinates associated with the rays accordingly.

We use a pre-trained, frozen DINOv2 (S/14) [106] as our image feature extractor. We use a DiT [110] with 16 transformer blocks as the architecture for both f_{Regress} (with t always set to 100) and $f_{\text{Diffusion}}$. We train our diffusion model with T=100 timesteps. When training our denoiser, we add noise to the direction and moment representation of rays. The ray regression and ray diffusion models take about 2 and 4 days respectively to train on 8 A6000 GPUs.

To predict cameras with our ray denoiser, we use DDPM [52] inference with slight modifications. Empirically, we found that removing the stochastic noise in DDPM inference and stopping the backward diffusion process early (and using the predicted x_0 as the estimate) produced better performance. We hypothesize that this is because while the earlier diffusion steps help select among distinct plausible modes, the later steps yield samples around these—and this may be detrimental to accuracy metrics that prefer distribution modes.

5.4 Evaluation

5.4.1 Experimental Setup

Dataset. Our method is trained and evaluated using CO3Dv2 [117]. This dataset contains turntable videos spanning 51 categories of household objects. Each frame is labeled with poses determined by COLMAP [127, 126]. Following [191], we train on 41 categories and hold out the remaining 10 categories for evaluating generalization.

Baselines. We evaluate our method against a handful of learning-based and correspondence-based pose estimation works.

	# of Images	2	3	4	5	6	7	8
Seen Categories	COLMAP (SP+SG) [124]	30.7	28.4	26.5	26.8	27.0	28.1	30.6
	RelPose [191]	56.0	56.5	57.0	57.2	57.2	57.3	57.2
	PoseDiffusion w/o GGS [165]	74.5	75.4	75.6	75.7	76.0	76.3	76.5
	PoseDiffusion [165]	75.7	76.4	76.8	77.4	78.0	78.7	78.8
	RelPose++ [73]	81.8	82.8	84.1	84.7	84.9	85.3	85.5
	R+T Regression [73]	49.1	50.7	53.0	54.6	55.7	56.1	56.5
	Ray Regression (Ours) [189]	88.8	88.7	88.7	89.0	89.4	89.3	89.2
	Ray Diffusion (Ours) [189]	**91.8**	**92.4**	**92.6**	**92.9**	**93.1**	**93.3**	**93.3**
Unseen Categories	COLMAP (SP+SG) [124]	34.5	31.8	31.0	31.7	32.7	35.0	38.5
	RelPose [191]	48.6	47.5	48.1	48.3	48.4	48.4	48.3
	PoseDiffusion w/o GGS [165]	62.1	62.4	63.0	63.5	64.2	64.2	64.4
	PoseDiffusion [165]	63.2	64.2	64.2	65.7	66.2	67.0	67.7
	RelPose++ [73]	69.8	71.1	71.9	72.8	73.8	74.4	74.9
	R+T Regression [73]	42.7	43.8	46.3	47.7	48.4	48.9	48.9
	Ray Regression (Ours) [189]	79.0	79.6	80.6	81.4	81.3	81.9	81.9
	Ray Diffusion (Ours) [189]	**83.5**	**85.6**	**86.3**	**86.9**	**87.2**	**87.5**	**88.1**

Table 5.1: **Camera Rotation Accuracy on CO3D (@ 15°).** Here we report the proportion of relative camera rotations that are within 15 degrees of the ground truth.

COLMAP [127, 126]. COLMAP is a standard dense correspondence-based SfM pipeline. We use an implementation [124] which uses SuperPoint features [31] and SuperGlue matching [125].

RelPose [191]. RelPose predicts relative rotations between pairs of cameras and defines evaluation procedures to optimize over a learned scoring function and determine joint rotations.

RelPose++ [73]. RelPose++ builds upon the pairwise scoring network of RelPose to incorporate multi-view reasoning via a transformer and also allows predicting camera translations.

R+T Regression [73]. To test the importance of modeling uncertainty, [73] trains a baseline that directly regresses poses. We report the numbers from [73].

PoseDiffusion [165]. PoseDiffusion reformulates the pose estimation task as directly diffusing camera extrinsics and focal length. Additionally, they introduce a geometry-guided sampling error to enforce epipolar constraints on predicted poses. We evaluate PoseDiffusion with and without the geometry-guided sampling.

5.4.2 Metrics

We evaluate sparse image sets of 2 to 8 images for each test sequence in CO3D. For an N image evaluation, we randomly sample N images and compute the accuracy of the predicted poses. We average these accuracies over 5 samples for each sequence to reduce stochasticity.

	# of Images	2	3	4	5	6	7	8
Seen Categories	COLMAP (SP+SG) [124]	100	34.5	23.8	18.9	15.6	14.5	15.0
	PoseDiffusion w/o GGS [165]	100	76.5	66.9	62.4	59.4	58.0	56.5
	PoseDiffusion [165]	100	77.5	69.7	65.9	63.7	62.8	61.9
	RelPose++ [73]	100	85.0	78.0	74.2	71.9	70.3	68.8
	R+T Regression [73]	100	58.3	41.6	35.9	32.7	31.0	30.0
	Ray Regression (Ours) [189]	100	91.7	85.7	82.1	79.8	77.9	76.2
	Ray Diffusion (Ours) [189]	100	**94.2**	**90.5**	**87.8**	**86.2**	**85.0**	**84.1**
Unseen Categs.	COLMAP (SP+SG) [124]	100	36.0	25.5	20.0	17.9	17.6	19.1
	PoseDiffusion w/o GGS[165]	100	62.5	48.8	41.9	39.0	36.5	34.8
	PoseDiffusion [165]	100	63.6	50.5	45.7	43.0	41.2	39.9
	RelPose++ [73]	100	70.6	58.8	53.4	50.4	47.8	46.6
	R+T Regression [73]	100	48.9	32.6	25.9	23.7	22.4	21.3
	Ray Regression (Ours) [189]	100	83.7	75.6	70.8	67.4	65.3	63.9
	Ray Diffusion (Ours) [189]	100	**87.7**	**81.1**	**77.0**	**74.1**	**72.4**	**71.4**

Table 5.2: **Camera Center Accuracy on CO3D (@ 0.1).** Here we report the proportion of camera centers that are within 0.1 of the scene scale. We apply an optimal similarity transform (s, $\boldsymbol{R}$, $\boldsymbol{t}$) to align predicted camera centers to ground truth camera centers (hence the alignment is perfect at $N = 2$ but worsens with more images).

Rotation Accuracy. We first compute the relative rotations between each pair of cameras for both predicted and ground truth poses. Then we determine the error between the ground truth and predicted pairwise relative rotations and report the proportion of these errors within 15 degrees.

Camera Center Accuracy. We align the ground truth and predicted poses in CO3D using the optimal similarity transform (s, $\boldsymbol{R}$, $\boldsymbol{t}$). We compare our prediction to the scene scale (the distance from the scene centroid to the farthest camera, following [135]). We report the proportion of aligned camera centers within 10 percent of the scene scale to the ground truth.

5.4.3 Evaluation

We report the camera rotation accuracy in Tab. 5.1 and camera center accuracy in Tab. 5.2 evaluated on CO3D. We find that COLMAP struggles in wide-baseline settings due to insufficient image overlap to find correspondences. We find that both the regression and diffusion versions of our method safely outperform all existing approaches, suggesting that our ray-based representation can effectively recover precise camera poses in this setup. In particular, our ray regression method significantly outperforms the baseline that regresses extrinsics R and T directly (R+T Regression). Similarly, our ray diffusion model demonstrates a large improvement over R+T Diffusion (PoseDiffusion without GGS) [165], while also outperforming

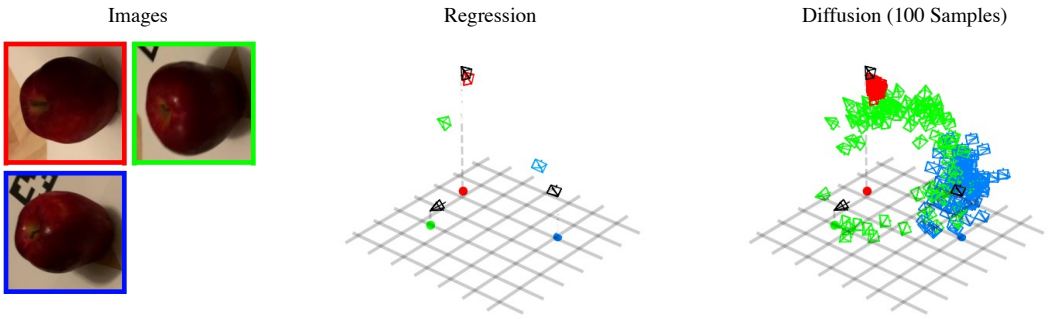


Figure 5.7: **Modeling Uncertainty Via Sampling Modes.** Sparse-view camera poses are sometimes inherently ambiguous due to symmetry. The capacity to model such uncertainty in probabilistic models such as our Ray Diffusion model is a significant advantage over regression-based models that must commit to a single mode. We thus investigate taking multiple samples from our diffusion model. We visualize the predicted cameras (colored) of both our regression- and diffusion-based approaches compared to the ground truth (black). While the regression model predicts the green camera incorrectly, we can recover better modes by sampling our diffusion model multiple times.

# of Rays	Rot@15	CC@0.01
2×2	52.5	72.5
4×4	70.3	82.6
8×8	76.1	84.8
16×16	**84.0**	**89.8**

Table 5.3: **Ray Resolution Ablation.** We evaluate various numbers of patches/rays by training a category-specific model for 2 different training categories (hydrant, wineglass) with $N = 3$ images. Performance across the 2 categories is averaged. We find that increasing the number of rays significantly improves performance. However, we found that increasing the number of rays beyond 16×16 was computationally prohibitive.

their full method (PoseDiffusion) which includes geometry-guided sampling.

We show qualitative results comparing both our Ray Regression and Diffusion methods with PoseDiffusion and RelPose++ in Fig. 5.5. We find that our ray-based representation consistently achieves finer localization. Additionally, ray diffusion achieves slightly better performance than ray regression. More importantly, it also allows recovering multiple plausible modes under uncertainty, as highlighted in Fig. 5.7.

Ablating Ray Resolution. We conduct an ablation study to evaluate how the number of camera rays affects performance in Tab. 5.3. We find that increasing the number

of camera rays significantly improves performance. Note that we kept the parameter count of the transformer constant, but more tokens incur a greater computational cost. All other experiments are conducted with 16×16 rays.

Demonstration on Self-captures. Finally, to demonstrate that our approach generalizes beyond the distribution of sequences from CO3D, we show qualitative results using Ray Diffusion on a variety of in-the-wild self-captures in Fig. 5.6.

5.5 Discussion

In this work, we explored representing camera poses using a distributed ray representation, and proposed a deterministic regression and a probabilistic diffusion approach for predicting these rays. While we examined this representation in the context of sparse views, it can be explored for single-view or dense multi-view setups. In addition, while our representation allows implicitly leveraging associations between patches, we do not enforce any geometric consistency (as done in classical pose estimation pipelines). It would be interesting to explore joint inference of our distributed ray representation and geometry in future work.

Chapter 6

Conclusions

In this thesis, we developed a pipeline for estimating camera poses of object-centric views and then recovering the textured surfaces of the object along with its illumination conditions. While we have made significant progress toward this task, a number of open challenges and future directions remain.

First, the camera pose and geometry estimation were largely done independently. In classical 3D optimization algorithms such as Bundle Adjustment [153], the joint optimization of camera poses and geometry is mutually beneficial. Subsequent work in learning-based 3D should also take advantage of the synergy of reprojecting geometry for improving camera pose. This idea is already starting to be explored in works such as [165] which incorporates 2D correspondences into the backward diffusion process. There are also recent works [170, 167] that perform that camera pose estimation by first predicting the geometry.

Second, in this thesis, we primarily considered object-centric setups. While capturing 3D objects is of utmost importance for reconstruction and scanning, we ultimately want a general purpose 3D pipeline that works in any setup: scene-centric, outdoors, *etc.*. Significant effort should be dedicated toward improving the generalization to these more diverse setups. DUSt3R [170] has already taken the first step toward training on a multitude of diverse datasets to improve this domain generalization.

Finally, ray-based representations of cameras can be extended widely to take advantage of images from a wide range of intrinsics and camera models. All images have some distortion, and being able to make use of these images directly without undistorting them would allow current methods to take greater advantage of the image pixels already present.

Bibliography

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-Scale Data for Multiple-View Stereopsis. *ICCV*, 2016. 18

[2] Yousset I Abdel-Aziz, Hauck Michael Karara, and Michael Hauck. Direct Linear Transformation from Comparator Coordinates into Object Space Coordinates in Close-Range Photogrammetry. *Photogrammetric Engineering & Remote Sensing*, 81(2):103–107, 2015. 63

[3] Edward H Adelson, James R Bergen, et al. *The Plenoptic Function and the Elements of Early Vision*, volume 2. MIT Press, 1991. 12

[4] Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. Objectron: A Large Scale Dataset of Object-Centric Videos in the Wild with Pose Annotations. In *CVPR*, 2021. 56

[5] Wang Angtian, Adam Kortylewski, and Alan Yuille. NeMo: Neural Mesh Models of Contrastive Features for Robust 3D Pose Estimation. In *ICLR*, 2021. 46

[6] Vassileios Balntas, Shuda Li, and Victor Prisacariu. RelocNet: Continuous Metric Learning Relocalisation using Neural Nets. In *ECCV*, 2018. 33, 60

[7] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded Up Robust Features. In *ECCV*, 2006. 31, 45, 60

[8] Anand Bhattad, Aysegul Dundar, Guilin Liu, Andrew Tao, and Bryan Catanzaro. View Generalization for Single Image Textured 3D Models. In *CVPR*, 2021. 17

[9] 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. 18

[10] Volker Blanz and Thomas Vetter. A Morphable Model for the Synthesis of 3D Faces. In *SIGGRAPH*, 1999. 17

[11] 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. 17

[12] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, et al. Uncertainty-Driven 6D Pose Estimation of Objects and Scenes from a Single RGB Image. In *CVPR*, 2016. 32

[13] Yannick Bukschat and Marcus Vetter. EfficientPose: An Efficient, Accurate and Scalable End-to-end 6D Multi Object Pose Estimation Approach. *arXiv:2011.04307*, 2020. 32

[14] Ruojin Cai, Bharath Hariharan, Noah Snavely, and Hadar Averbuch-Elor. Extreme Rotation Estimation using Dense Correlation Volumes. In *CVPR*, 2021. 47, 60

[15] Carlos Campos, Richard Elvira, Juan J. Gómez, José M. M. Montiel, and Juan D. Tardós. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM. *T-RO*, 37(6):1874–1890, 2021. 32, 45, 60

[16] Luca Carlone, Roberto Tron, Kostas Daniilidis, and Frank Dellaert. Initialization Techniques for 3D SLAM: A Survey on Rotation Estimation and its Use in Pose Graph Optimization. *ICRA*, 2015. 34

[17] Thomas J Cashman and Andrew W Fitzgibbon. What Shape are Dolphins? Building 3D Morphable Models from 2D Images. *TPAMI*, 35(1):232–244, 2012. 17

[18] Llukman Cerkezi and Paolo Favaro. Sparse 3D Reconstruction via Object-Centric Ray Sampling. In *3DV*, 2024. 30

[19] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An Information-rich 3D Model Repository. *arXiv:1512.03012*, 2015. 26, 46

[20] Bo Chen, Tat-Jun Chin, and Marius Klimavicius. Occlusion-Robust Object Pose Estimation with Holistic Representation. In *WACV*, 2022. 32

[21] Kefan Chen, Noah Snavely, and Ameesh Makadia. Wide-Baseline Relative Camera Pose Estimation with Directional Learning. In *CVPR*, 2021. 33, 47, 60

[22] Sungjoon Choi, Qian-Yi Zhou, Stephen Miller, and Vladlen Koltun. A Large Dataset of Object Scans. *arXiv:1602.02481*, 2016. 18

[23] Christopher B Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal Correspondence Network. *NeurIPS*, 2016. 31, 45

[24] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction. In *ECCV*, 2016. 17

[25] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, Matthieu Guillaumin, and Jitendra Malik. ABO: Dataset and Benchmarks for Real-World 3D Object Understanding. In *CVPR*, 2022. 12

[26] Enric Corona, Kaustav Kundu, and Sanja Fidler. Pose Estimation for Objects with Rotational Symmetry. In *IROS*, 2018. 32

[27] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. MonoSLAM: Real-time Single Camera SLAM. *TPAMI*, 29(6):1052–1067, 2007. 32, 50, 60

[28] Xinke Deng, Arsalan Mousavian, Yu Xiang, Fei Xia, Timothy Bretl, and Dieter Fox. PoseRBPF: A Rao-Blackwellized Particle Filter for 6D Object Pose Tracking. In *RSS*, 2019. 32

[29] Xinke Deng, Yu Xiang, Arsalan Mousavian, Clemens Eppner, Timothy Bretl, and Dieter Fox. Self-supervised 6D Object Pose Estimation for Robot Manipulation. In *ICRA*, 2020. 32

[30] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H Barr. Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow. In *SIGGRAPH*, 1999. 24

[31] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised Interest Point Detection and Description. In *CVPR-W*, 2018. 31, 40, 45, 54, 60, 67

[32] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*, 2021. 63

[33] Aubrey K Dunne, John Mallon, and Paul F Whelan. Efficient Generic Calibration Method for General Cameras with Single Centre of Projection. *Computer Vision and Image Understanding*, 114(2):220–233, 2010. 61

[34] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A Trainable CNN for Joint Detection and Description of Local Features. In *CVPR*, 2019. 31

[35] Mihai Dusmanu, Johannes L Schönberger, and Marc Pollefeys. Multi-view Optimization of Local Feature Geometry. In *ECCV*, 2020. 31

[36] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct Sparse Odometry. *TPAMI*, 2018. 32

[37] Martin A Fischler and Robert C Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, 1981. 45

[38] James D Foley, Foley Dan Van, Andries Van Dam, Steven K Feiner, John F Hughes, Edward Angel, and J Hughes. *Computer Graphics: Principles and Practice*, volume 12110. Addison-Wesley Professional, 1996. 16

[39] Yasutaka Furukawa, Brian Curless, Steven M Seitz, and Richard Szeliski. Towards Internet-scale Multi-view Stereo. In *CVPR*, 2010. 31

[40] Yasutaka Furukawa and Carlos Hernández. Multi-view Stereo: A Tutorial. *Foundations and Trends in Computer Graphics and Vision*, 9(1-2):1–148, 2015. 17

[41] Igor Gilitschenski, Roshni Sahoo, Wilko Schwarting, Alexander Amini, Sertac Karaman, and Daniela Rus. Deep Orientation Uncertainty Learning Based on a Bingham Loss. In *ICLR*, 2019. 32, 46

[42] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a Predictable and Generative Vector Representation for Objects. In *ECCV*, 2016. 17

[43] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh R-CNN. In *ICCV*, 2019. 17

[44] Shubham Goel, Georgia Gkioxari, and Jitendra Malik. Differentiable Stereopsis: Meshes from Multiple Views Using Differentiable Rendering. In *CVPR*, 2022. 30

[45] Shubham Goel, Angjoo Kanazawa, , and Jitendra Malik. Shape and Viewpoints without Keypoints. In *ECCV*, 2020. 17, 19

[46] Michael D Grossberg and Shree K Nayar. A General Imaging Model and a Method for Finding its Parameters. In *ICCV*, 2001. 14, 59, 61, 62

[47] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A Papier-mâché Approach to Learning 3D Surface Generation. In *CVPR*, 2018. 17, 19

[48] Chris Harris and Mike Stephens. A Combined Corner and Edge Detector. In *Alvey Vision Conference*, 1988. 31

[49] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. 49

[50] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016. 37, 48

[51] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans Trained by a Two Time-scale Update Rule Converge to a Local Nash Equilibrium. In *NeurIPS*, 2017. 27

[52] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. *NeurIPS*, 2020. 64, 66

[53] David S Immel, Michael F Cohen, and Donald P Greenberg. A Radiosity Method for Non-diffuse Environments. In *SIGGRAPH*, 1986. 20

[54] Shun Iwase, Xingyu Liu, Rawal Khirodkar, Rio Yokota, and Kris M. Kitani. RePOSE: Fast 6D Object Pose Refinement via Deep Texture Rendering. In *ICCV*, 2021. 32

[55] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting NeRF on a Diet: Semantically Consistent Few-Shot View Synthesis. In *ICCV*, 2021. 18

[56] Hanwen Jiang, Zhenyu Jiang, Kristen Grauman, and Yuke Zhu. Few-View Object Reconstruction with Unknown Categories and Camera Poses. In *3DV*, 2024. 47, 58, 60

[57] James T Kajiya. The Rendering Equation. In *SIGGRAPH*, 1986. 20

[58] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end Recovery of Human Shape and Pose. In *CVPR*, 2018. 46

[59] Angjoo Kanazawa, Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Learning Category-Specific Mesh Reconstruction from Image Collections. In *ECCV*, 2018. 17, 19

[60] Angjoo Kanazawa, Jason Y Zhang, Panna Felsen, and Jitendra Malik. Learning 3D Human Dynamics from Video. In *CVPR*, 2019. 47

[61] Juho Kannala and Sami S Brandt. A Generic Camera Model and Calibration Method for Conventional, Wide-Angle, and Fish-Eye Lenses. *TPAMI*, 28(8):1335–1340, 2006. 61

[62] Abhishek Kar, Shubham Tulsiani, Joao Carreira, and Jitendra Malik. Category-specific Object Reconstruction from a Single Image. In *CVPR*, 2015. 17

[63] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3D Mesh Renderer. In *CVPR*, 2018. 17, 19

[64] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. SSD-6D: Making RGB-based 3D Detection and 6D Pose Estimation Great Again. In *ICCV*, 2017. 32

[65] Alex Kendall and Roberto Cipolla. Modelling Uncertainty in Deep Learning for Camera Relocalization. In *ICRA*, 2016. 32

[66] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In *ICCV*, 2015. 32

[67] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image Segmentation as Rendering. In *CVPR*, 2020. 22

[68] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and Temples: Benchmarking Large-scale Scene Reconstruction. *TOG*, 36(4):1–13, 2017. 18

[69] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. VIBE: Video Inference for Human Body Pose and Shape Estimation. In *CVPR*, 2020. 47

[70] Zihang Lai, Sifei Liu, Alexei A Efros, and Xiaolong Wang. Video Autoencoder: Self-supervised Disentanglement of 3D Structure and Motion. In *ICCV*, 2021. 33

[71] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular Primitives for High-Performance Differentiable Rendering. *TOG*, 39(6), 2020. 17

[72] 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. In *NeurIPS*, 2020. 17

[73] Amy Lin, Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose++: Recovering 6d poses from sparse-view observations. In *3DV*, 2024. 58, 59, 60, 65, 67, 68

[74] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. BARF: Bundle-Adjusting Neural Radiance Fields. In *ICCV*, 2021. 18, 30, 34, 57

[75] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-Perfect Structure-from-Motion with Featuremetric Refinement. In *ICCV*, 2021. 31, 60

[76] Ce Liu, Jenny Yuen, and Antonio Torralba. SIFT Flow: Dense Correspondence Across Scenes and Its Applications. *TPAMI*, 33(5):978–994, 2010. 31, 45

[77] Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. SparseNeuS: Fast Generalizable Neural Surface Reconstruction from Sparse Views. In *ECCV*, 2022. 30

[78] H Christopher Longuet-Higgins. A Computer Algorithm for Reconstructing a Scene from Two Projections. *Nature*, 293(5828):133–135, 1981. 45

[79] David G Lowe. Distinctive Image Features from Scale-invariant Keypoints. *IJCV*, 60(2):91–110, 2004. 31, 45, 60

[80] Bruce D Lucas and Takeo Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *IJCAI*, 1981. 31, 45, 60

[81] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. MediaPipe: A Framework for Building Perception Pipelines. *arXiv preprint arXiv:1906.08172*, 2019. 55, 56

[82] Wei-Chiu Ma, Anqi Joyce Yang, Shenlong Wang, Raquel Urtasun, and Antonio Torralba. Virtual Correspondence: Humans as a Cue for Extreme-View Geometry. In *CVPR*, 2022. 47

[83] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised Learning of Depth and Ego-Motion From Monocular Video Using 3D Geometric Constraints. In *CVPR*, 2018. 33

[84] Fabian Manhardt, Diego Martin Arroyo, Christian Rupprecht, Benjamin Busam, Tolga Birdal, Nassir Navab, and Federico Tombari. Explaining the Ambiguity of Object Detection and 6D Pose from Visual Data. In *ICCV*, 2019. 32, 46

[85] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *CVPR*, 2021. 17

[86] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera. *TOG*, 2017. 46

[87] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Relative Camera Pose Estimation Using Convolutional Neural Networks. In *ACIVS*, 2017. 33, 47

[88] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy Networks: Learning 3D Reconstruction in Function Space. In *CVPR*, 2019. 17

[89] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*, 2020. 3, 12, 16, 17, 23, 25, 27, 37, 61

[90] David Mohlin, Josephine Sullivan, and Gérald Bianchi. Probabilistic Orientation Estimation with Matrix Fisher Distributions. *NeurIPS*, 2020. 32

[91] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *T-RO*, 31(5):1147–1163, 2015. 32, 45, 60

[92] Raúl Mur-Artal and Juan D. Tardós. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras. *T-RO*, 33(5):1255–1262, 2017. 32, 45, 60

[93] Kieran A Murphy, Carlos Esteves, Varun Jampani, Srikumar Ramalingam, and Ameesh Makadia. Implicit-PDF: Non-Parametric Representation of Probability Distributions on the Rotation Manifold. In *ICML*, 2021. 4, 6, 32, 33, 34, 35, 37, 38, 46, 49

[94] Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. Laplacian Mesh Optimization. In *GRAPHITE*, 2006. 24

[95] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. DTAM: Dense Tracking and Mapping in Real-time. In *ICCV*, 2011. 32

[96] Van Nguyen Nguyen, Yuming Du, Yang Xiao, Michael Ramamonjisoa, and Vincent Lepetit. PIZZA: A Powerful Image-only Zero-Shot Zero-CAD Approach to 6 DoF Tracking. In *3DV*, 2022. 47, 48

[97] David Nistér. An Efficient Solution to the Five-point Relative Pose Problem. *TPAMI*, 26(6):756–770, 2004. 45

[98] David Novotny, Diane Larlus, and Andrea Vedaldi. Learning 3D Object Categories by Looking Around Them. In *ICCV*, 2017. 32, 46

[99] David Novotny, Nikhila Ravi, Benjamin Graham, Natalia Neverova, and Andrea Vedaldi. C3DPO: Canonical 3D Pose Networks for Non-Rigid Structure From Motion. In *ICCV*, 2019. 32

[100] M. Oberweger, M. Rad, and V. Lepetit. Making Deep Heatmaps Robust to Partial Occlusions for 3D Object Pose Estimation. In *ECCV*, 2018. 32

[101] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture Fields: Learning Texture Representations in Function Space. In *ICCV*, 2019. 17

[102] Michael Oechsle, Songyou Peng, and Andreas Geiger. UNISURF: Unifying Neural Implicit Surfaces and Radiance Fields for Multi-View Reconstruction. In *ICCV*, 2021. 17

[103] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep Metric Learning via Lifted Structured Feature Embedding. In *CVPR*, 2016. 18

[104] Brian Okorn, Qiao Gu, Martial Hebert, and David Held. ZePHyR: Zero-shot Pose Hypothesis Scoring. In *ICRA*, 2021. 32, 46

[105] Brian Okorn, Mengyun Xu, Martial Hebert, and David Held. Learning Orientation Distributions for Object Pose Estimation. In *IROS*, 2020. 32, 46

[106] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision. *arXiv preprint arXiv:2304.07193*, 2023. 8, 63, 66

[107] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *CVPR*, 2019. 17

[108] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable Neural Radiance Fields. In *ICCV*, 2021. 17

[109] Rémi Pautrat, Viktor Larsson, Martin R. Oswald, and Marc Pollefeys. Online Invariance Selection for Local Feature Descriptors. In *ECCV*, 2020. 31

[110] William Peebles and Saining Xie. Scalable Diffusion Models with Transformers. In *ICCV*, 2023. 63, 66

[111] Bui Tuong Phong. Illumination for Computer Generated Pictures. *Communications of the ACM*, 18(6), 1975. 3, 20, 21

[112] Julius Plücker. *Analytisch-Geometrische Entwicklungen*, volume 2. GD Baedeker, 1828. 8, 62, 63

[113] Sergey Prokudin, Peter Gehler, and Sebastian Nowozin. Deep Directional Statistics: Pose Estimation with Uncertainty Quantification. In *ECCV*, 2018. 32

[114] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *CVPR*, 2020. 17

[115] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3D Deep Learning with PyTorch3D. *arXiv:2007.08501*, 2020. 17, 23

[116] Daniel Rebain, Mark Matthews, Kwang Moo Yi, Dmitry Lagun, and Andrea Tagliasacchi. LOLNeRF: Learn from One Look. In *CVPR*, 2022. 30

[117] 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*, 2021. 38, 45, 51, 59, 66

[118] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2D2: Reliable and Repeatable Detector and Descriptor. *NeurIPS*, 2019. 31

[119] Gernot Riegler and Vladlen Koltun. Free View Synthesis. In *ECCV*, 2020. 17

[120] Chris Rockwell, Justin Johnson, and David F Fouhey. The 8-Point Algorithm as an Inductive Bias for Relative Pose Prediction by ViTs. In *3DV*, 2022. 33, 47, 60

[121] Olinde Rodrigues. Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire. *Journal de Mathématiques Pures et Appliquées*, 5, 1840. 38

[122] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: an Open-Source Library for Real-Time Metric-Semantic Localization and Mapping. In *ICRA*, 2020. 32

[123] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "GrabCut" Interactive Foreground Extraction using Iterated Graph Cuts. *TOG*, 23(3):309–314, 2004. 22

[124] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In *CVPR*, 2019. 31, 40, 52, 54, 67, 68

[125] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning Feature Matching with Graph Neural Networks. In *CVPR*, 2020. 2, 14, 31, 40, 45, 54, 60, 67

[126] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *CVPR*, 2016. 24, 31, 38, 40, 45, 50, 51, 53, 60, 66, 67

[127] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise View Selection for Unstructured Multi-View Stereo. In *ECCV*, 2016. 31, 45, 53, 60, 66, 67

[128] Thomas Schops, Viktor Larsson, Marc Pollefeys, and Torsten Sattler. Why Having 10,000 Parameters in Your Camera Model is Better Than Twelve. In *CVPR*, 2020. 61, 62

[129] Thomas Schops, Torsten Sattler, and Marc Pollefeys. BAD SLAM: Bundle Adjusted Direct RGB-D SLAM. In *CVPR*, 2019. 32, 60

[130] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A Multi-view Stereo Benchmark with High-resolution Images and Multi-camera Videos. In *CVPR*, 2017. 18

[131] Nima Sedaghat and Thomas Brox. Unsupervised Generation of a Viewpoint Annotated Car Dataset from Videos. In *ICCV*, 2015. 18

[132] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A Comparison and Evaluation of Multi-view Stereo Reconstruction Algorithms. In *CVPR*, 2006. 18

[133] Xi Shen, François Darmon, Alexei A Efros, and Mathieu Aubry. RANSAC-Flow: Generic Two-stage Image Alignment. In *ECCV*, 2020. 60

[134] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Learning Local Feature Descriptors Using Convex Optimisation. *TPAMI*, 36(8):1573–1585, 2014. 31

[135] Samarth Sinha, Jason Y Zhang, Andrea Tagliasacchi, Igor Gilitschenski, and David B Lindell. SparsePose: Sparse-View Camera Pose Regression and Refinement. In *CVPR*, 2023. 47, 48, 51, 53, 55, 57, 58, 59, 60, 68

[136] Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. Light Field Networks: Neural Scene Representations with Single-evaluation Rendering. In *NeurIPS*, 2021. 61

[137] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. DeepVoxels: Learning Persistent 3D Feature Embeddings. In *CVPR*, 2019. 17

[138] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations. In *NeurIPS*, 2019. 17

[139] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo Tourism: Exploring Photo Collections in 3D. In *SIGGRAPH*. 2006. 31, 60

[140] Chen Song, Jiaru Song, and Qixing Huang. Hybridpose: 6D Object Pose Estimation under Hybrid Representations. In *CVPR*, 2020. 32

[141] 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. 17

[142] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A Benchmark for the Evaluation of RGB-D SLAM Systems. In *IROS*, 2012. 51

[143] Weiwei Sun, Andrea Tagliasacchi, Boyang Deng, Sara Sabour, Soroosh Yazdani, Geoffrey E Hinton, and Kwang Moo Yi. Canonical Capsules: Self-supervised Capsules in Canonical Pose. In *NeurIPS*, 2021. 32, 46

[144] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3D: Dataset and Methods for Single-Image 3D Shape Modeling. In *CVPR*, 2018. 30

[145] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3D Orientation Learning for 6D Object Detection from RGB Images. In *ECCV*, 2018. 32

[146] Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P. Srinivasan, Jonathan T. Barron, and Ren Ng. Learned Initializations for Optimizing Coordinate-Based Neural Representations. In *CVPR*, 2021. 3, 25, 26, 27

[147] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. In *NeurIPS*, 2020. 37

[148] Chengzhou Tang and Ping Tan. BA-Net: Dense Bundle Adjustment Network. In *ICLR*, 2019. 31, 60

[149] Zachary Teed and Jia Deng. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. *NeurIPS*, 2021. 32, 40, 47

[150] Bugra Tekin, Sudipta N. Sinha, and Pascal Fua. Real-Time Seamless Single Shot 6D Object Pose Prediction. In *CVPR*, 2018. 32

[151] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred Neural Rendering: Image Synthesis using Neural Textures. *TOG*, 38(4):1–12, 2019. 17

[152] Engin Tola, Vincent Lepetit, and Pascal Fua. Daisy: An Efficient Dense Descriptor Applied to Wide-baseline Stereo. *TPAMI*, 32(5):815–830, 2009. 31

[153] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle Adjustment—A Modern Synthesis. In *International workshop on vision algorithms*, 1999. 31, 45, 60, 71

[154] Prune Truong, Martin Danelljan, and Radu Timofte. GLU-Net: Global-Local Universal Network for Dense Flow and Correspondences. In *CVPR*, 2020. 31

[155] Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari. SPARF: Neural Radiance Fields from Sparse and Noisy Poses. In *CVPR*, 2023. 30

[156] Shubham Tulsiani, Nilesh Kulkarni, and Abhinav Gupta. Implicit Mesh Reconstruction from Unannotated Image Collections. *arXiv:2007.08504*, 2020. 17, 19

[157] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view Supervision for Single-view Reconstruction via Differentiable Ray Consistency. In *CVPR*, 2017. 17

[158] Shinji Umeyama. Least-squares Estimation of Transformation Parameters Between Two Point Patterns. *TPAMI*, 13(04):376–380, 1991. 53

[159] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox. DeMoN: Depth and Motion Network for Learning Monocular Stereo. In *CVPR*, 2017. 33

[160] Ben Usman, Andrea Tagliasacchi, Kate Saenko, and Avneesh Sud. MetaPose: Fast 3D Pose from Multiple Views without 3D Supervision. In *CVPR*, 2022. 47

[161] Igor Vasiljevic, Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Wolfram Burgard, Greg Shakhnarovich, and Adrien Gaidon. Neural Ray Surfaces for Self-supervised Learning of Depth and Ego-motion. In *3DV*, 2020. 61

[162] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All You Need. *NeurIPS*, 2017. 6, 46, 65

[163] Sudheendra Vijayanarasimhan, Susanna Ricco, Cordelia Schmid, Rahul Sukthankar, and Katerina Fragkiadaki. SfM-Net: Learning of Structure and Motion from Video. *arXiv:1704.07804*, 2017. 33

[164] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion. In *CVPR*, 2019. 32

[165] Jianyuan Wang, Christian Rupprecht, and David Novotny. PoseDiffusion: Solving Pose Estimation via Diffusion-aided Bundle Adjustment. In *ICCV*, 2023. 47, 48, 52, 54, 55, 56, 58, 59, 60, 67, 68, 71

[166] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. In *NeurIPS*, 2021. 17

[167] Peng Wang, Hao Tan, Sai Bi, Yinghao Xu, Fujun Luan, Kalyan Sunkavalli, Wenping Wang, Zexiang Xu, and Kai Zhang. PF-LRM: Pose-Free Large Reconstruction Model for Joint Pose and Shape Prediction. In *ICLR*, 2024. 60, 71

[168] Qianqian Wang, Xiaowei Zhou, Bharath Hariharan, and Noah Snavely. Learning Feature Descriptors Using Camera Pose Supervision. In *ECCV*, 2020. 31

[169] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. DeepVO: Towards End-to-End Visual Odometry with Deep Recurrent Convolutional Neural Networks. In *ICRA*, 2017. 32, 47

[170] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Revaud Jerome. DUSt3R: Geometric 3D Vision Made Easy. In *CVPR*, 2024. 60, 71

[171] Wenshan Wang, Yaoyu Hu, and Sebastian Scherer. TartanVO: A Generalizable Learning-based VO. In *CoRL*, 2020. 32

[172] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel View Synthesis with Diffusion Models. In *ICLR*, 2023. 61

[173] Xingkui Wei, Yinda Zhang, Zhuwen Li, Yanwei Fu, and Xiangyang Xue. DeepSFM: Structure From Motion Via Deep Bundle Adjustment. In *ECCV*, 2020. 31

[174] Jay M Wong, Vincent Kee, Tiffany Le, Syler Wagner, Gian-Luca Mariottini, Abraham Schneider, Lei Hamilton, Rahul Chipalkatty, Mitchell Hebert, David MS Johnson, et al. SegICP: Integrated Deep Semantic Segmentation and Pose Estimation. In *IROS*, 2017. 32, 46

[175] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P. Srinivasan, Dor Verbin, Jonathan T. Barron, Ben Poole, and Aleksander Holynski. ReconFusion: 3D Reconstruction with Diffusion Priors. In *CVPR*, 2024. 30

[176] Shangzhe Wu, Ameesh Makadia, Jiajun Wu, Noah Snavely, Richard Tucker, and Angjoo Kanazawa. De-rendering the World's Revolutionary Artefacts. In *CVPR*, 2021. 20, 23

[177] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. In *RSS*, 2018. 32

[178] Yang Xiao, Yuming Du, and Renaud Marlet. PoseContrast: Class-Agnostic Object Viewpoint Estimation in the Wild with Pose-Aware Contrastive Learning. In *3DV*, 2021. 46

[179] Yang Xiao, Xuchong Qiu, Pierre-Alain Langlois, Mathieu Aubry, and Renaud Marlet. Pose from Shape: Deep Pose Estimation for Arbitrary 3D Objects. In *BMVC*, 2019. 22, 24, 32, 46

[180] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective Transformer Nets: Learning Single-view 3D Object Reconstruction without 3D Supervision. In *NeurIPS*, 2016. 17

[181] Gengshan Yang and Deva Ramanan. Volumetric Correspondence Networks for Optical Flow. *NeurIPS*, 2019. 60

[182] Gengshan Yang, Deqing Sun, Varun Jampani, Daniel Vlasic, Forrester Cole, Huiwen Chang, Deva Ramanan, William T Freeman, and Ce Liu. LASR: Learning Articulated Shape Reconstruction from a Monocular Video. In *CVPR*, 2021. 17

[183] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry. In *CVPR*, 2020. 47

[184] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume Rendering of Neural Implicit Surfaces. In *NeurIPS*, 2021. 17

[185] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance. In *NeurIPS*, 2020. 3, 17, 25, 26, 27

[186] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. LIFT: Learned Invariant Feature Transform. In *ECCV*, 2016. 31

[187] Zhichao Yin and Jianping Shi. GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. In *CVPR*, 2018. 33

[188] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural Radiance Fields from One or Few Images. In *CVPR*, 2021. 30

[189] Jason Y. Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as Rays: Pose Estimation via Ray Diffusion. In *ICLR*, 2024. 67, 68

[190] Jason Y. Zhang, Sam Pepose, Hanbyul Joo, Deva Ramanan, Jitendra Malik, and Angjoo Kanazawa. Perceiving 3D Human-Object Spatial Arrangements from a Single Image in the Wild. In *ECCV*, 2020. 32, 46

[191] Jason Y. Zhang, Deva Ramanan, and Shubham Tulsiani. RelPose: Predicting probabilistic relative rotation for single objects in the wild. In *ECCV*, 2022. 6, 7, 46, 47, 48, 49, 50, 51, 52, 54, 58, 59, 60, 66, 67

[192] Jason Y. Zhang, Gengshan Yang, Shubham Tulsiani, and Deva Ramanan. NeRS: Neural Reflectance Surfaces for Sparse-view 3D Reconstruction in the Wild. In *NeurIPS*, 2021. 6, 30, 34, 43, 57

[193] Richard Zhang. Making Convolutional Networks Shift-Invariant Again. In *ICML*, 2019. 37

[194] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 23, 27

[195] Xiuming Zhang, Sean Fanello, Yun-Ta Tsai, Tiancheng Sun, Tianfan Xue, Rohit Pandey, Sergio Orts-Escolano, Philip Davidson, Christoph Rhemann, Paul Debevec, et al. Neural Light Transport for Relighting and View Synthesis. *TOG*, 40(1):1–17, 2021. 18

[196] 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. 17

[197] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. DeepTAM: Deep Tracking and Mapping. In *ECCV*, 2018. 32

[198] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised Learning of Depth and Ego-Motion From Video. In *CVPR*, 2017. 33

[199] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo Magnification: Learning view synthesis using multiplane images. *SIGGRAPH*, 37, 2018. 56

[200] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the Continuity of Rotation Representations in Neural Networks. In *CVPR*, 2019. 40

[201] Zhizhuo Zhou and Shubham Tulsiani. SparseFusion: Distilling View-conditioned Diffusion for 3D Reconstruction. In *CVPR*, 2022. 30

[202] Jon Zubizarreta, Iker Aguinaga, and J. M. M. Montiel. Direct Sparse Mapping. *T-RO*, 2020. 32