

Photometric Mesh Optimization for Video-Aligned 3D Object Reconstruction

Chen-Hsuan Lin^{1,2*} Oliver Wang² Bryan C. Russell² Eli Shechtman²
Vladimir G. Kim² Matthew Fisher² Simon Lucey¹

¹The Robotics Institute, Carnegie Mellon University ²Adobe Research
chlin@cmu.edu {owang,brussell,elishe,vokim,matfishe}@adobe.com slucey@cs.cmu.edu

<https://chenhsuanlin.bitbucket.io/photometric-mesh-optim/>

Abstract

In this paper, we address the problem of 3D object mesh reconstruction from RGB videos. Our approach combines the best of multi-view geometric and data-driven methods for 3D reconstruction by optimizing object meshes for multi-view photometric consistency while constraining mesh deformations with a shape prior. We pose this as a piecewise image alignment problem for each mesh face projection. Our approach allows us to update shape parameters from the photometric error without any depth or mask information. Moreover, we show how to avoid a degeneracy of zero photometric gradients via rasterizing from a virtual viewpoint. We demonstrate 3D object mesh reconstruction results from both synthetic and real-world videos with our photometric mesh optimization, which is unachievable with either naïve mesh generation networks or traditional pipelines of surface reconstruction without heavy manual post-processing.

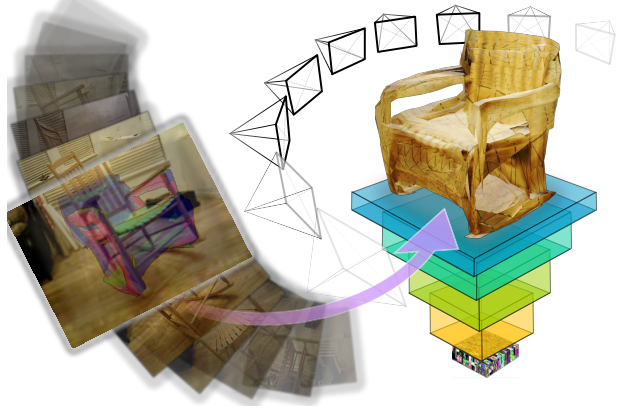


Figure 1: Our video-aligned object mesh reconstruction enforcing multi-view consistency while constraining shape deformations with shape priors, generating an output mesh with improved geometry with respect to the input views.

1. Introduction

The choice of 3D representation plays a crucial role in 3D reconstruction problems from 2D images. Classical multi-view geometric methods, most notably structure from motion (SfM) and SLAM, recover point clouds as the underlying 3D structure of RGB sequences, often with very high accuracy [10, 30]. Point clouds, however, lack inherent 3D spatial structure that is essential for efficient reasoning. In many scenarios, *mesh representations* are more desirable – they are significantly more compact since they have inherent geometric structures defined by point connectivity, while they also represent continuous surfaces necessary for many applications such as robotics (e.g., accurate localization for autonomous driving), computer graphics (e.g., physical simulation, texture synthesis), and virtual/augmented reality.

Another drawback of classical multi-view geometric methods is reliance on hand-designed features and can be

fragile when their assumptions are violated. This happens especially in textureless regions or when there are changes in illumination. Data-driven approaches [5, 15], on the other hand, learn priors to tackle ill-posed 3D reconstruction problems and have recently been widely applied to 3D prediction tasks from single images. However, they can only reliably reconstruct from the space of training examples it learns from, resulting in limited ability to generalize to unseen data.

In this work, we address the problem of 3D mesh reconstruction from image sequences by bringing together the best attributes of multi-view geometric methods and data-driven approaches (Fig. 1). Focusing on object instances, we use *shape priors* (specifically, neural networks) to reconstruct geometry with incomplete observations as well as *multi-view geometric constraints* to refine mesh predictions on the input sequences. Our approach allows dense reconstruction with object semantics from learned priors, which is not possible from the traditional pipelines of surface meshing [22] from multi-view stereo (MVS). Moreover, our approach general-

*Work done during CHL’s internship at Adobe Research.

izes to unseen objects by utilizing multi-view geometry to enforce observation consistency across viewpoints.

Given only RGB information, we achieve mesh reconstruction from image sequences by photometric optimization, which we pose as a piecewise image alignment problem of individual mesh faces. To avoid degeneracy, we introduce a novel virtual viewpoint rasterization to compute photometric gradients with respect to mesh vertices for 3D alignment, allowing the mesh to deform to the observed shape. A main advantage of our photometric mesh optimization is its non-reliance on any a-priori known depth or mask information [20, 35, 38] – a necessary condition to be able to reconstruct objects from real-world images. With this, we take a step toward practical usage of prior-based 3D mesh reconstruction aligned with RGB sequences.

In summary, we present the following contributions:

- We incorporate multi-view photometric consistency with data-driven shape priors for optimizing 3D meshes using 2D photometric cues.
- We propose a novel photometric optimization formulation for meshes and introduce a virtual viewpoint rasterization step to avoid gradient degeneracy.

Finally, we show 3D object mesh reconstruction results from both synthetic and real-world sequences, unachievable with either naïve mesh generators or traditional MVS pipelines without heavy manual post-processing.

2. Related Work

Our work on object mesh reconstruction touches several areas, including multi-view object reconstruction, mesh optimization, deep shape priors, and image alignment.

Multi-view object reconstruction. Multi-view calibration and reconstruction is a well-studied problem. Most approaches begin by estimating camera coordinates using 2D keypoint matching, a process known as SLAM [10, 29] or SfM [12, 32], followed by dense reconstruction methods such as MVS [13] and meshing [22]. More recent works using deep learning have explored 3D reconstruction from multiple-view consistency between various forms of 2D observations [24, 34, 35, 38, 41]. These methods all utilize forms of 2D supervision that are easier to acquire than 3D CAD models, which are relatively limited in quantity. Our approach uses both geometric and image-based constraints, which allows it to overcome common multi-view limitations such as missing observations and textureless regions.

Mesh optimization. Mesh optimization dates back to classical works of Active Shape Models [7] and Active Appearance Models [6, 28], which uses 2D meshes to fit facial landmarks. In this work, we optimize for 3D meshes using 2D photometric cues, a significantly more challenging prob-

lem due to the inherent ambiguities in the task. Similar approaches for mesh refinement have also been explored [8, 9]; however, a sufficiently good initialization is required with very small vertex perturbations allowed. As we show in our experiments, we are able to handle larger amount of noise perturbation by optimizing over a latent shape code instead of mesh vertices, making it more suitable for practical uses.

Several recent methods have addressed learning 3D reconstruction with mesh representations. AtlasNet [15] and Pixel2Mesh [36] are examples of learning mesh object reconstructions from 3D CAD models. Meanwhile, Neural Mesh Renderer [21] suggested a method of mesh reconstruction via approximate gradients for 2D mask optimization, and Kanazawa *et al.* [20] further advocated learning mesh reconstruction from 2D supervision of textures, masks, and 2D keypoints. Our approach, in contrast, does *not* assume any availability of masks or keypoints and operates purely via photometric cues across viewpoints.

Shape priors. The use of neural networks as object priors for reconstruction has recently been explored with point clouds [42]. However, it requires object masks as additional constraints during optimization. We eliminate the need for mask supervision by regularizing the latent code. Shape priors have also been explored for finding shape correspondences [14], where the network learns the deformation field from a template shape to match 3D observations. In our method, we directly optimize the latent shape code to match 2D cues from multiple viewpoints and do not require a known shape template for the object. A plane and primitive prior has been used for the challenging task of multi-view scene reconstruction [18]. Although the primitive prior does not need to be learned from an object dataset, the resulting reconstruction can differ significantly from the target geometry when it is not well represented by the chosen primitives.

Image alignment. The most generic form of image alignment refers to prediction of inherent geometric misalignment between a pair of images. Image alignment using simple warping functions can be dated back to the seminal Lucas-Kanade algorithm [27] and its recent variants [1, 26]. Recent work has also explored learning a warp function to align images from neural networks for applications such as novel view synthesis [39, 40] and learning invariant representations [19, 25]. In this work, we pose our problem of mesh optimization as multiple image alignment problems of mesh faces, and solve it by optimizing over a latent code from a deep network rather than the vertices themselves.

3. Approach

We seek to reconstruct a 3D object mesh from an RGB sequence $\{(\mathcal{I}_f, \Omega_f)\}$, where each frame $\mathcal{I}_f$ is associated with a camera matrix Ω_f . In this work, we assume that the camera matrices $\{\Omega_f\}$ can be readily obtained from off-

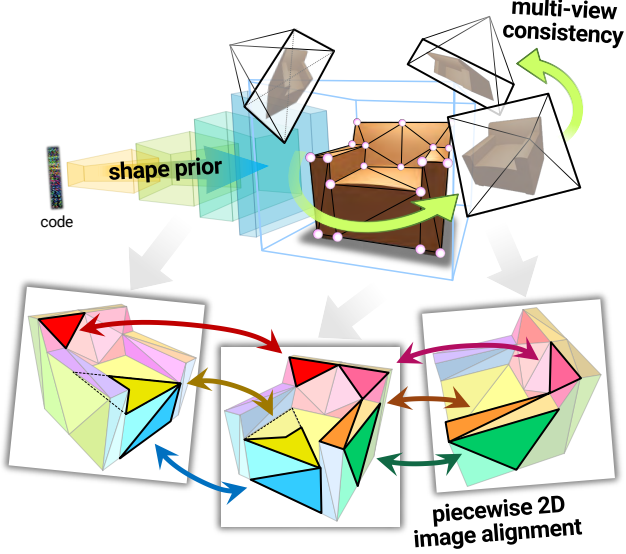


Figure 2: Overview. We perform 3D mesh reconstruction via *piecewise image alignment* of triangles to achieve per-triangle visibility-aware photometric consistency across multiple views, with mesh vertices optimized over the latent code of a shape prior learned by deep neural networks.

the-shelf SfM methods [32]. Fig. 2 provides an overview – we optimize for object meshes that maximize multi-view photometric consistency over a shape prior, where we use a pretrained mesh generator. We focus on triangular meshes here although our method is applicable to any mesh type.

3.1. Mesh Optimization over Shape Prior

Direct optimization on a 3D mesh $\mathcal{M}$ with N vertices involves solving for $3N$ degrees of freedom (DoFs) and typically becomes underconstrained when N is large. Therefore, reducing the allowed DoFs is crucial to ensure mesh deformations are well-behaved during optimization. We wish to represent the mesh $\mathcal{M} = \mathcal{G}(\mathbf{z})$ as a differentiable function $\mathcal{G}$ of a reduced vector representation $\mathbf{z}$.

We propose to use an off-the-shelf generative neural network as the main part of $\mathcal{G}$ and reparameterize the mesh with an associated latent code $\mathbf{z} \in \mathbb{R}^K$, where $K \ll 3N$. The network serves as an object shape prior whose efficacy comes from pretraining on external shape datasets. Shape priors over point clouds have been previously explored [42]; here, we extend to mesh representations. We use AtlasNet [15] here although other mesh generators are also applicable. The shape prior allows the predicted 3D mesh to deform within a learned shape space, avoiding many local minima that exist with direct vertex optimization. To utilize RGB information from the given sequence for photometric optimization, we further add a 3D similarity transform to map the generated mesh to world cameras recovered by SfM (see Sec. 3.4).

We define our optimization problem as follows: given the RGB image sequence and cameras $\{(\mathcal{I}_f, \Omega_f)\}$, we optimize a regularized cost consisting of a photometric loss $\mathcal{L}_{\text{photo}}$ for all pairs of frames over the representation $\mathbf{z}$, formulated as

$$\min_{\mathbf{z}} \sum_{a \neq b} \mathcal{L}_{\text{photo}}(\mathcal{I}_a, \mathcal{I}_b, \Omega_a, \Omega_b; \mathcal{G}(\mathbf{z})) + \mathcal{L}_{\text{reg}}(\mathbf{z}), \quad (1)$$

where $\mathcal{L}_{\text{reg}}$ is a regularization term on $\mathbf{z}$. This objective allows the generated mesh to deform with respect to an effective shape prior. We describe each term in detail next.

3.2. Piecewise Image Alignment

Optimizing the mesh $\mathcal{M}$ with the photometric loss $\mathcal{L}_{\text{photo}}$ is based on the assumption that a dense 2D projection of the individual triangular faces of a 3D mesh should be globally consistent across multiple viewpoints. Therefore, we cast the problem of 3D mesh alignment to the input views as a collection of *piecewise 2D image alignment* subproblems of each projected triangular face (Fig. 2).

To perform piecewise 2D image alignment between $\mathcal{I}_a$ and $\mathcal{I}_b$, we need to establish pixel correspondences. We first denote $\mathbf{V}_j(\mathbf{z}) \in \mathbb{R}^{3 \times 3}$ as the 3D vertices of triangle j in mesh $\mathcal{M} = \mathcal{G}(\mathbf{z})$, defined as column vectors. From triangle j , we can sample a collection of 3D points $\mathcal{P}_j = \{\mathbf{p}_i(\mathbf{z})\}$ that lie within triangle j , related via $\mathbf{p}_i(\mathbf{z}) = \mathbf{V}_j(\mathbf{z})\alpha_i$ through the barycentric coordinates α_i . For a camera Ω , let $\pi: \mathbb{R}^3 \rightarrow \mathbb{R}^2$ be the projection function mapping a world 3D point $\mathbf{p}_i(\mathbf{z})$ to 2D image coordinates. The pixel intensity error between the two views Ω_a and Ω_b can be compared at the 2D image coordinates corresponding to the projected sampled 3D points. We formulate the photometric loss $\mathcal{L}_{\text{photo}}$ as the sum of ℓ_1 distances between pixel intensities at these 2D image coordinates over all triangular faces,

$$\begin{aligned} & \mathcal{L}_{\text{photo}}(\mathcal{I}_a, \mathcal{I}_b, \Omega_a, \Omega_b; \mathcal{G}(\mathbf{z})) \\ &= \sum_j \sum_{i: \mathbf{p}_i \in \mathcal{P}_j} \|\mathcal{I}_a(\pi(\mathbf{p}_i(\mathbf{z}); \Omega_a)) - \mathcal{I}_b(\pi(\mathbf{p}_i(\mathbf{z}); \Omega_b))\|_1. \end{aligned} \quad (2)$$

As such, we can optimize the photometric loss $\mathcal{L}_{\text{photo}}$ with pixel correspondences established as a function of $\mathbf{z}$.

Visibility. As a 3D point $\mathbf{p}_i$ may not be visible in a given view due to possible object self-occlusion, we handle visibility by constraining $\mathcal{P}_j$ to be the set of samples in triangle j whose projection is visible in both views. We achieve this by returning a mesh index map using mesh rasterization, a standard operation in computer graphics, for each optimization step. The photometric gradients of each sampled point $\frac{\partial \mathcal{L}}{\partial \mathbf{V}_j} = \frac{\partial \mathcal{L}}{\partial \mathbf{x}_i} \frac{\partial \mathbf{x}_i}{\partial \mathbf{p}_i} \frac{\partial \mathbf{p}_i}{\partial \mathbf{V}_j}$ in turn backpropagate to the vertices $\mathbf{V}_j$. We obtain $\frac{\partial \mathcal{L}}{\partial \mathbf{x}_i}$ through differentiable image sampling [19], $\frac{\partial \mathbf{x}_i}{\partial \mathbf{p}_i}$ by taking the derivative of the projection π , and $\frac{\partial \mathbf{p}_i}{\partial \mathbf{V}_j}$ by associating with the barycentric coordinates α_i . We note that the entire process is differentiable and does not resort to approximate gradients [21].

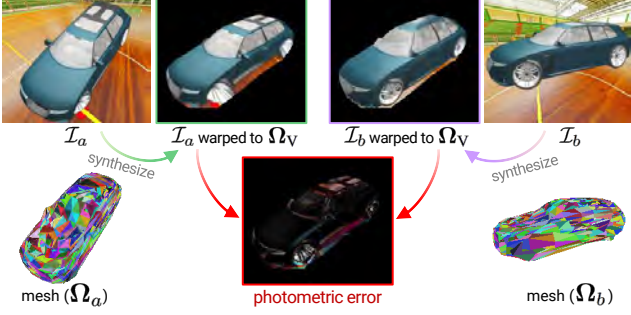


Figure 3: Visualization of the photometric loss $\mathcal{L}_{\text{photo}}$ between the synthesized appearances at virtual viewpoints Ω_V starting from input images I_a and I_b . The photometric loss $\mathcal{L}_{\text{photo}}$ encourages consistent appearance syntheses from both input viewpoints Ω_a and Ω_b .

3.3. Virtual Viewpoint Rasterization

We can efficiently sample a large number of 3D points $\mathcal{P}_j$ in triangle j by rendering the depth of $\mathcal{M}$ from a given view using mesh rasterization (Sec. 3.2). If the depth were rasterized from either input view Ω_a or Ω_b , however, we would obtain zero photometric gradients. This degeneracy arises due to the fact that ray-casting from one view and projecting back to the same view results in $\frac{\partial \mathcal{I}}{\partial \mathbf{V}_j} = \mathbf{0}$.

To elaborate, we first note that depth rasterization of triangle j is equivalent to back-projecting regular grid coordinates $\bar{\mathbf{x}}_i$ to triangle j . We can express each depth point from camera $\Omega \in \{\Omega_a, \Omega_b\}$ as $\mathbf{p}_i(\mathbf{z}) = \pi^{-1}(\bar{\mathbf{x}}_i; \mathbf{V}_j(\mathbf{z}), \Omega)$, where $\pi^{-1} : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ is the inverse projection function realized by solving for ray-triangle intersection with $\mathbf{V}_j(\mathbf{z})$. Combining with the projection equation, we have

$$\mathbf{x}_i(\mathbf{z}, \Omega) = \pi(\pi^{-1}(\bar{\mathbf{x}}_i; \mathbf{V}_j(\mathbf{z}), \Omega); \Omega) = \bar{\mathbf{x}}_i \quad \forall \bar{\mathbf{x}}_i, \quad (3)$$

becoming the identity mapping and losing the dependency of $\mathbf{x}_i$ on $\mathbf{V}_j(\mathbf{z})$, which in turn leads to $\frac{\partial \mathbf{x}_i}{\partial \mathbf{V}_j} = \mathbf{0}$. This insight is in line with the recent observation from Ham *et al.* [16].

To overcome this degeneracy, we rasterize the depth from a *third* virtual viewpoint $\Omega_V \notin \{\Omega_a, \Omega_b\}$. This step allows correct gradients to be computed in both viewpoints Ω_a and Ω_b , which is essential to maintain stability during optimization. We can form the photometric loss by synthesizing the image appearance at Ω_V using the pixel intensities from both Ω_a and Ω_b (Fig. 3). We note that Ω_V can be arbitrarily chosen. In practice, we choose Ω_V to be the bisection between Ω_a and Ω_b by applying Slerp [33] on the rotation quaternions and averaging the two camera centers.

3.4. Implementation Details

Coordinate systems. Mesh predictions from a generative network typically lie in a canonical coordinate system [15, 36] independent of the world cameras recovered



Figure 4: Sample sequences composited from ShapeNet renderings (top: car, bottom: airplane) and SUN360 scenes.

by SfM. Therefore, we need to account for an additional 3D similarity transform $\mathcal{T}(\cdot)$ applied to the mesh vertices. For each 3D vertex $\mathbf{v}'_k$ from the prediction, we define the similarity transform as

$$\mathbf{v}_k = \mathcal{T}(\mathbf{v}'_k; \boldsymbol{\theta}) = \exp(s) \cdot \mathcal{R}(\boldsymbol{\omega}) \mathbf{v}'_k + \mathbf{t} \quad \forall k, \quad (4)$$

where $\boldsymbol{\theta} = [s; \boldsymbol{\omega}; \mathbf{t}] \in \mathbb{R}^7$ are the parameters and $\mathcal{R}$ is a 3D rotation matrix parameterized with the $\mathfrak{so}(3)$ Lie algebra. We optimize for $\mathbf{z} = [\mathbf{z}'; \boldsymbol{\theta}]$ together, where $\mathbf{z}'$ is the latent code associated with the generative network.

Since automated registration of noisy 3D data with unknown scales is still an open problem, we assume a coarse alignment of the coordinate systems can be computed from minimal annotation of rough correspondences (see Sec. 4.3 for details). We optimize for the similarity transform to more accurately align the meshes to the RGB sequences.

Regularization. Despite neural networks being effective priors, the latent space is only spanned by the training data. To avoid meshes from reaching a degenerate solution, we impose an extra penalty on the latent code $\mathbf{z}'$ to ensure it stays within a trust region of the initial code $\mathbf{z}_0$ (extracted from a pretrained image encoder), defined as $\mathcal{L}_{\text{code}} = \|\mathbf{z}' - \mathbf{z}_0\|_2^2$. We also add a scale penalty $\mathcal{L}_{\text{scale}} = -s$ that encourages the mesh to expand, since the mesh shrinking to infinitesimal is a trivial solution with zero photometric error. The regularization $\mathcal{L}_{\text{reg}}$ in cost (1) is written as

$$\mathcal{L}_{\text{reg}}(\mathbf{z}) = \lambda_{\text{code}} \cdot \mathcal{L}_{\text{code}}(\mathbf{z}') + \lambda_{\text{scale}} \cdot \mathcal{L}_{\text{scale}}(\boldsymbol{\theta}) \quad (5)$$

where λ_{code} and λ_{scale} are the penalty weights.

4. Experiments

We evaluate the performance of our method on a single (Sec. 4.1) and multiple (Sec. 4.2) object categories with synthetic data as well as real-world videos (Sec. 4.3).

Data preparation. We create datasets of 3D CAD model renderings for training a mesh generation network and evaluating our optimization framework. Our rendering pipeline aims to create realistic images with complex backgrounds so they could be applied to real-world video sequences. We use

ShapeNet [3] for the object dataset and normalize all objects to fit an origin-centered unit sphere. We render RGB images of each object using perspective cameras at 24 equally spaced azimuth angles and 3 elevation angles.

To simulate realistic backgrounds, we randomly warp and crop spherical images from the SUN360 database [37] to create background images of the same scene taken at different camera viewpoints. By compositing the foreground and background images together at corresponding camera poses, we obtain RGB sequences of objects composited on realistic textured backgrounds (Fig. 4). Note that we do not keep any mask information that was accessible in the rendering and compositing process as such information is typically not available in real-world examples. All images are rendered/cropped at a resolution of 224×224 .

Shape prior. We use AtlasNet [15] as the base network architecture for mesh generation, which we retrain on our new dataset. We use the same 80%-20% training/test split from Groueix *et al.* [15] and additionally split the SUN360 spherical images with the same ratio. During training, we augment background images at random azimuth angles.

Initialization. We initialize the code $\mathbf{z}_0$ by encoding an RGB frame with the AtlasNet encoder. For ShapeNet sequences, we choose frames with objects facing 45° sideways. For real-world sequences, we manually select frames where objects are center-aligned to the images as much as possible to match our rendering settings. We initialize the similarity transform parameters to $\theta = \mathbf{0}$ (identity transform).

Evaluation criteria. We evaluate the result by measuring the 3D distances between the sampled 3D points from the predicted meshes and the ground-truth point clouds [15]. We follow Lin *et al.* [24] by reporting the 3D error between the predicted and ground-truth point clouds as $\eta(\mathcal{S}_1, \mathcal{S}_2) = \sum_{i: \mathbf{v}_i \in \mathcal{S}_1} \min_{\mathbf{v}_j \in \mathcal{S}_2} \|\mathbf{v}_i - \mathbf{v}_j\|_2$ for some source and target point sets $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. This metric measures the prediction shape accuracy when $\mathcal{S}_1$ is the prediction and $\mathcal{S}_2$ is the ground truth, while it indicates the prediction shape coverage when vice versa. We report quantitative results in both directions separately averaged across all instances.

4.1. Single Object Category

We start by evaluating our mesh alignment in a category-specific setting. We select the car, chair, and plane categories from ShapeNet, consisting of 703, 1356, and 809 objects in our test split, respectively. For each object, we create an RGB sequence by overlaying its rendering onto a randomly paired SUN360 scene with the cameras in correspondence. We retrain each category-specific AtlasNet model on our new dataset using the default settings for 500 epochs. During optimization, we use the Adam optimizer [23] with a constant learning rate of 0.003 for 100 iterations. We manually set the penalty factors to be $\lambda_{\text{code}} = 0.05$ and $\lambda_{\text{scale}} = 0.02$.

One challenge is that the coordinate system for a mesh generated by AtlasNet is independent of the recovered world cameras $\{\Omega_f\}$ for a real-world sequence. Determining such coordinate system mapping (defined by a 3D similarity transform) is required to relate the predicted mesh to the world. On the other hand, for the synthetic sequences, we know the exact mapping as we can render the views for AtlasNet and the input views $\{\mathcal{I}_f\}$ in the same coordinate system.

For our first experiment, we simulate the possibly incorrect mapping estimates by perturbing the ground-truth 3D similarity transform by adding Gaussian noise $\varepsilon \sim \mathcal{N}(0, \sigma \mathbf{I})$ to its parameters, pre-generated per sequence for evaluation. We evaluate the 3D error metrics under such perturbations. Note that our method utilizes no additional information other than the RGB information from the given sequences.

We compare our mesh reconstruction approach against three baseline variants of AtlasNet: (a) mesh generations from a single-image feed-forward initialization, (b) generation from the mean latent code averaged over all frames in the sequence, and (c) the mean shape where vertices are averaged from the mesh generation across all frames.

We show qualitative results in Fig. 5 (compared under perturbation $\sigma = 0.12$). Our method is able to take advantage of multi-view geometry to resolve large misalignments and optimize for more accurate shapes. The high photometric error from the background between views discourages mesh vertices from staying in such regions. This error serves as a natural force to constrain the mesh within the desired 3D regions, eliminating the need of depth or mask constraints during optimization. We further visualize our mesh reconstruction with textures that are estimated from all images (Fig. 6). Note that the fidelity of mean textures increases while variance in textures decrease after optimization.

We evaluate quantitatively in Fig. 7, where we plot the average 3D error over mapping noise. This result demonstrates how our method handles inaccurate coordinate system mappings to successfully match the meshes against RGB sequences. We also ablate optimizing the latent code $\mathbf{z}$, showing that allowing shape deformation improves reconstruction quality over a sole 3D similarity transform (“fixed code” in Fig. 7). Note that our method is slightly worse in shape coverage error (GT→pred.) when evaluated at the ground-truth mapping. This result is attributed to the limitation of photometric optimization that opts for degenerate solutions when objects are insufficiently textured.

4.2. Multiple Object Categories

We extend beyond a model that reconstructs a single object category by training a single model to reconstruct multiple object categories. We take 13 commonly chosen CAD model categories from ShapeNet [5, 11, 15, 24]. We follow the same settings as in Sec. 4.1 except we retrain AtlasNet longer for 1000 epochs due to a larger training set.

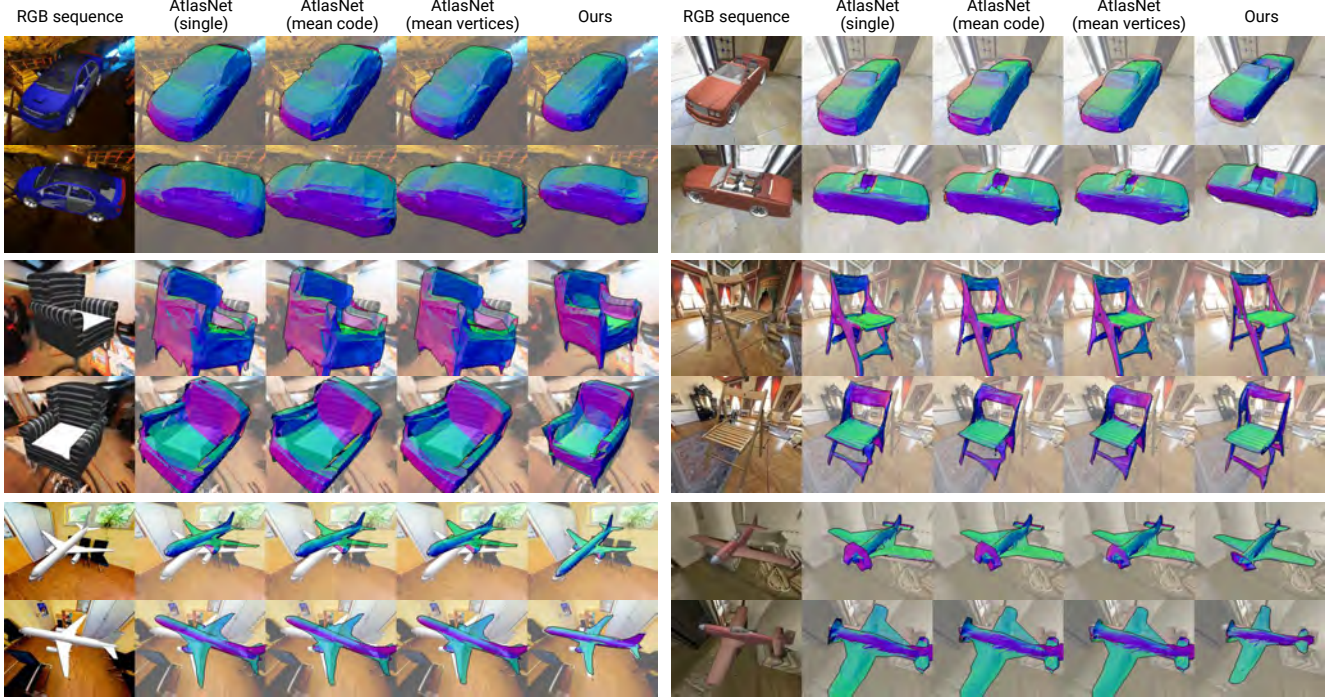


Figure 5: Qualitative results from category-specific models, where we visualize two sample frames from each test sequence. Our method better aligns initial meshes to the RGB sequences while optimizing for more subtle shape details (*e.g.*, car spoilers and airplane wings) over baselines. The meshes are color-coded by surface normals with occlusion boundaries drawn.

Category	plane	bench	cabin.	car	chair	monit.	lamp	speak.	fire.	couch	table	cell.	water.	mean
AtlasNet (single)	3.872	4.931	5.708	4.269	4.869	4.687	8.684	7.245	3.864	5.017	4.964	4.571	4.290	5.152
AtlasNet (mean code)	3.746	4.496	5.600	4.286	4.571	4.634	7.366	6.976	3.632	4.798	4.903	4.286	3.860	4.858
AtlasNet (mean shape)	3.659	4.412	5.382	4.192	4.499	4.424	7.200	6.683	3.547	4.606	4.860	4.196	3.742	4.723
Ours	0.704	1.821	2.850	0.597	1.441	1.115	8.855	3.430	1.255	0.983	1.725	1.599	1.743	2.163

(a) 3D error: prediction $\rightarrow$ ground truth (shape accuracy).

Category	plane	bench	cabin.	car	chair	monit.	lamp	speak.	fire.	couch	table	cell.	water.	mean
AtlasNet (single)	4.430	4.895	5.024	4.461	4.896	4.640	8.906	6.994	4.407	4.613	5.350	4.254	4.263	5.164
AtlasNet (mean code)	4.177	4.507	4.962	4.384	4.635	4.143	7.292	6.990	4.307	4.463	5.084	4.036	3.718	4.823
AtlasNet (mean shape)	4.464	4.915	5.150	4.521	4.940	4.560	8.159	7.308	4.528	4.707	5.255	4.299	4.183	5.153
Ours	2.237	3.215	1.927	0.734	2.377	2.119	10.764	4.152	2.583	1.735	6.126	1.851	2.926	3.288

(b) 3D error: ground truth $\rightarrow$ prediction (shape coverage).

Table 1: Average 3D test error for general object categories (numbers scaled by 10^3). The mean is taken across categories. Our optimization method is effective on most object categories. Note that our method improves on accuracy of the table category despite worsening in shape coverage due to insufficient textures in object samples.

We show visual results in Fig. 8 on the efficacy of our method for multiple object categories (under perturbation $\sigma = 0.12$). Our results show how we can reconstruct a shape that better matches our RGB observations (*e.g.*, refining hollow regions, as in the bench backs and table legs). We also show category-wise quantitative results in Table 1, compared under perturbation noise $\sigma = 0.06$. We find photometric

optimization to perform effectively across most categories except lamps, which consist of many examples where optimizing for thin structures is hard for photometric loss.

4.3. Real-world Videos

Finally, we demonstrate the efficacy of our method on challenging real-world video sequences orbiting an object.

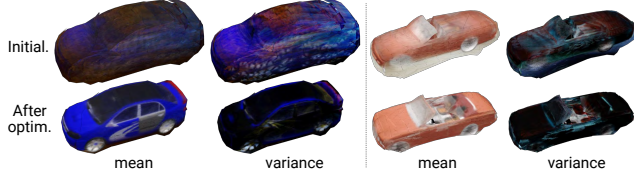


Figure 6: Mesh visualization with textures computed by averaging projections across all viewpoints. Our method successfully reduces variance and recovers dense textures that can be embedded on the surfaces.

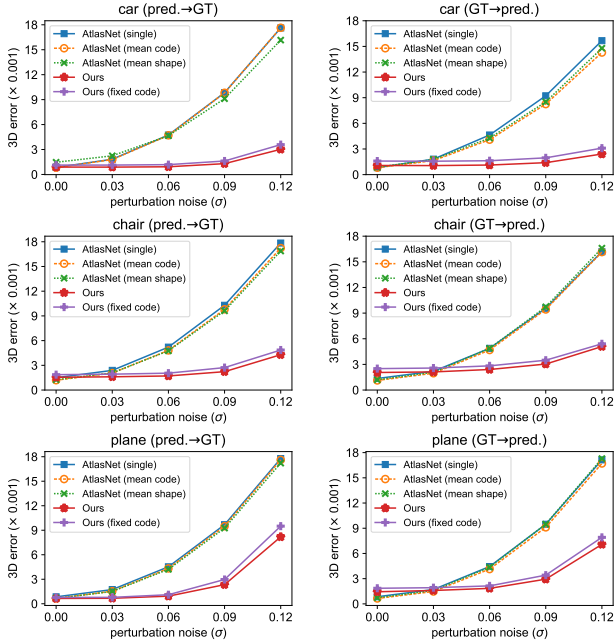


Figure 7: Category-specific performance to noise in coordinate system mapping. Our method is able to resolve for various extents of mesh misalignments from the sequence.

We use a dataset of RGB-D object scans [4], where we use the chair model to evaluate on the chair category. We select the subset of video sequences that are 3D-reconstructible using traditional pipelines [32] and where SfM extracts at least 20 reliable frames and 100 salient 3D points. We retain 82 sequences with sufficient quality for evaluation. We rescale the sequences to 240×320 and skip every 10 frames.

We compute the camera extrinsic and intrinsic matrices using off-the-shelf SfM with COLMAP [32]. For evaluation, we additionally compute a rough estimate of the coordinate system mapping by annotating 3 corresponding points between the predicted mesh and the sparse points extracted from SfM (Fig. 9), which allows us to fit a 3D similarity transform. We optimize using Adam with a learning rate of $2e-3$ for 200 iterations, and we manually set the penalty factors to be $\lambda_{\text{code}} = 0.05$ and $\lambda_{\text{scale}} = 0.01$.

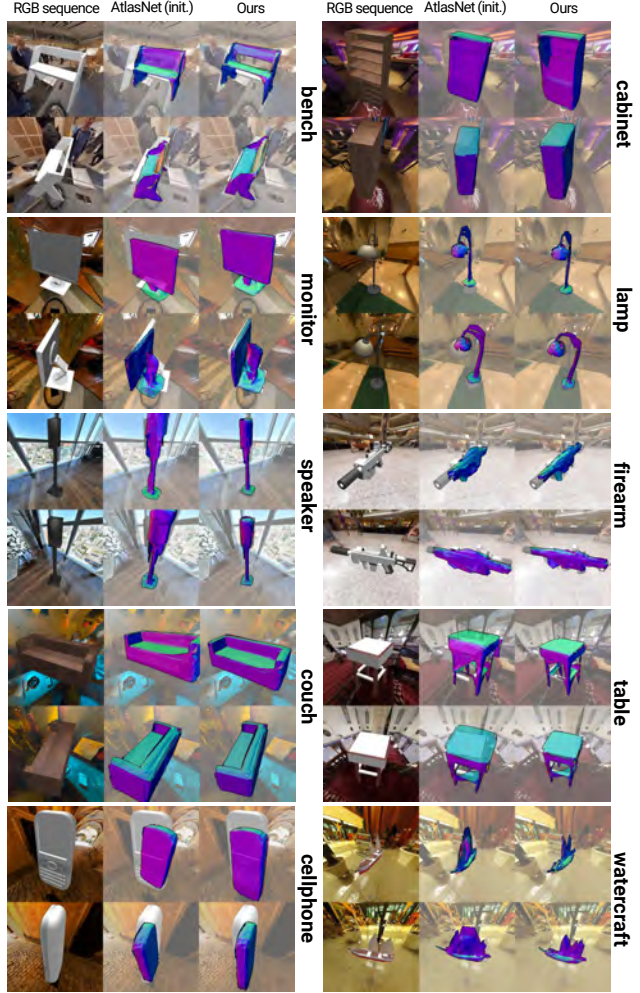


Figure 8: Qualitative results for general object categories. Our optimization method recovers subtle details such as back of benches, watercraft sails, and even starts to reveal cabinet open spaces which were initially occluded. Our method tends to fail more frequently with textureless objects (*e.g.*, cellphone and firearm).

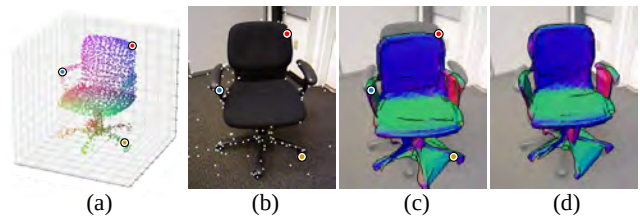


Figure 9: We select 3 correspondences between (a) the mesh vertices and (b) the SfM points to find (c) an estimated coordinate system mapping by fitting a 3D similarity transform. (d) Alignment result after our photometric optimization.

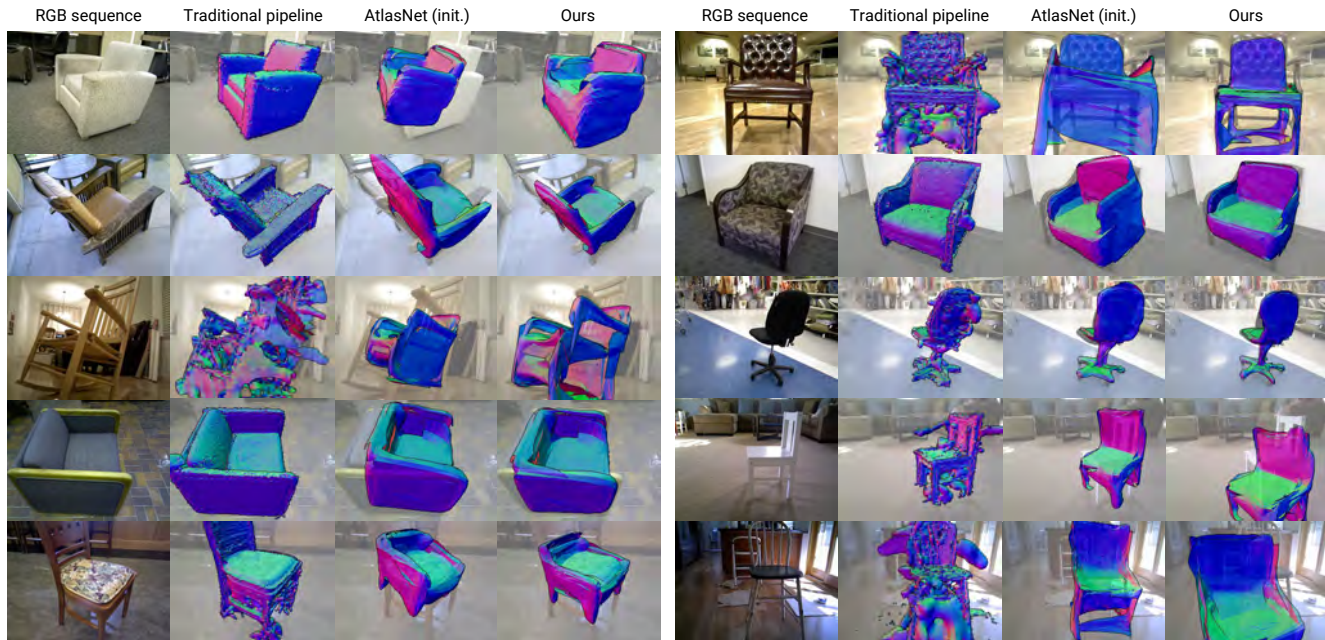


Figure 10: Qualitative results on real-world sequences. Given an initialization, our method accurately aligns a generated mesh to an RGB video. Even when the initial mesh is an inaccurate prediction of the real object, our method is still able to align the semantic parts (bottom left). We show failure cases in the last two examples in the bottom right, where there is insufficient background texture as photometric cues and where the initial mesh is insufficient to capture the thin structures. We also show the result of a traditional reconstruction pipeline [32] after manual cleanup. Due to the difficulty of the problem these meshes still often have many undesirable artifacts.

Dist.	Initial.	Optim.
1	6.504	4.990
2	9.064	6.979
3	10.984	8.528
4	12.479	9.788
6	14.718	11.665

Table 2: Average pixel reprojection error (scaled by 100) from real-world videos as a function of frame distances.

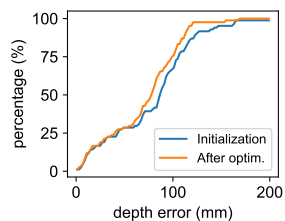


Figure 11: Metric-scale depth error before and after optimization (with SfM world cameras rescaled).

We demonstrate how our method is applicable to real-world datasets in Fig. 10. Our method is able to refine shapes such as armrests and office chair legs. Note that our method is sensitive to the quality of mesh initialization from real images, mainly due to the domain mismatch between synthetic and real data during the training/test phases of the shape prior. Despite this, it is still able to straighten and align to the desired 3D location. In addition, we report the average pixel reprojection error in Table 2 and metric depth error in Fig. 11 to quantify the effect of photometric optimization, which shows further improvement over coarse initializations.

Finally, we note that surface reconstruction is a challenging post-processing procedure for traditional pipelines. Fig. 10 shows sample results for SfM [32], PatchMatch Stereo [2], stereo fusion, and Poisson mesh reconstruction [22] from COLMAP [32]. In addition to the need of accurate object segmentation, the dense meshing problem with traditional pipelines typically yields noisy results without laborious manual post-processing.

5. Conclusion

We have demonstrated a method for reconstructing a 3D mesh from an RGB video by combining data-driven deep shape priors with multi-view photometric consistency optimization. We also show that mesh rasterization from a virtual viewpoint is critical for avoiding degenerate photometric gradients during optimization. We believe our photometric mesh optimization technique has merit for a number of practical applications. It enables the ability to generate more accurate models of real-world objects for computer graphics and potentially allows automated object segmentation from video data. It could also benefit 3D localization for robot navigation and autonomous driving, where accurate object location, orientation, and shape from real-world cameras is crucial for more efficient understanding.

References

- [1] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International journal of computer vision*, 56(3):221–255, 2004. [2](#)
- [2] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo-stereo matching with slanted support windows. In *British Machine Vision Conference*, 2011. [8](#)
- [3] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. [5](#), [11](#)
- [4] Sungjoon Choi, Qian-Yi Zhou, Stephen Miller, and Vladlen Koltun. A large dataset of object scans. *arXiv preprint arXiv:1602.02481*, 2016. [7](#)
- [5] 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 *European conference on computer vision*, pages 628–644. Springer, 2016. [1](#), [5](#)
- [6] Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2001. [2](#)
- [7] Timothy F Cootes and Christopher J Taylor. Active shape modelssmart snakes. In *BMVC92*, pages 266–275. Springer, 1992. [2](#)
- [8] Amaël Delaunoy and Marc Pollefeys. Photometric bundle adjustment for dense multi-view 3d modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1486–1493, 2014. [2](#)
- [9] Amaël Delaunoy and Emmanuel Prados. Gradient flows for optimizing triangular mesh-based surfaces: Applications to 3d reconstruction problems dealing with visibility. *International journal of computer vision*, 95(2):100–123, 2011. [2](#)
- [10] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European Conference on Computer Vision*, pages 834–849. Springer, 2014. [1](#), [2](#)
- [11] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [5](#)
- [12] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial Intelligence Review*, 43(1):55–81, 2015. [2](#)
- [13] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2010. [2](#)
- [14] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 230–246, 2018. [2](#)
- [15] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. Atlasnet: A papier-mâché approach to learning 3d surface generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [1](#), [2](#), [3](#), [4](#), [5](#), [11](#)
- [16] Christopher Ham, Simon Lucey, and Surya Singh. Proxy templates for inverse compositional photometric bundle adjustment. *arXiv preprint arXiv:1704.06967*, 2017. [4](#)
- [17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. [11](#)
- [18] Jingwei Huang, Angela Dai, Leonidas Guibas, and Matthias Nießner. 3dlite: Towards commodity 3d scanning for content creation. *ACM Transactions on Graphics 2017 (TOG)*, 2017. [2](#)
- [19] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015. [2](#), [3](#)
- [20] Angjoo Kanazawa, Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018. [2](#)
- [21] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [2](#), [3](#)
- [22] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):29, 2013. [1](#), [2](#), [8](#)
- [23] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. [5](#), [11](#)
- [24] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018. [2](#), [5](#)
- [25] Chen-Hsuan Lin and Simon Lucey. Inverse compositional spatial transformer networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [2](#)
- [26] Chen-Hsuan Lin, Rui Zhu, and Simon Lucey. The conditional lucas & kanade algorithm. In *European Conference on Computer Vision (ECCV)*, pages 793–808. Springer International Publishing, 2016. [2](#)
- [27] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI’81*, pages 674–679, 1981. [2](#)
- [28] Iain Matthews and Simon Baker. Active appearance models revisited. *International journal of computer vision*, 60(2):135–164, 2004. [2](#)
- [29] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. [2](#)
- [30] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *2011 international conference on computer vision*, pages 2320–2327. IEEE, 2011. [1](#)
- [31] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy,

- Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 11
- [32] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 3, 7, 8
- [33] Ken Shoemake. Animating rotation with quaternion curves. In *ACM SIGGRAPH computer graphics*. ACM, 1985. 4
- [34] Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Multi-view consistency as supervisory signal for learning shape and pose prediction. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 2
- [35] Shubham Tulsiani, Tinghui Zhou, Alexei A. Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 2
- [36] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. *arXiv preprint arXiv:1804.01654*, 2018. 2, 4
- [37] Jianxiong Xiao, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Recognizing scene viewpoint using panoramic place representation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2695–2702. IEEE, 2012. 5, 11
- [38] Xinchun Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *Advances in Neural Information Processing Systems*, pages 1696–1704, 2016. 2
- [39] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017. 2
- [40] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *European Conference on Computer Vision*, pages 286–301. Springer, 2016. 2
- [41] Rui Zhu, Hamed Kiani Galoogahi, Chaoyang Wang, and Simon Lucey. Rethinking reprojection: Closing the loop for pose-aware shape reconstruction from a single image. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 57–65. IEEE, 2017. 2
- [42] Rui Zhu, Chaoyang Wang, Chen-Hsuan Lin, Ziyang Wang, and Simon Lucey. Object-centric photometric bundle adjustment with deep shape prior. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 894–902. IEEE, 2018. 2, 3

A. Appendix

A.1. Architectural and Pretraining Details

We use AtlasNet [15] as the base network architecture for our experiments. Following Groueix *et al.* [15], the image encoder is the ResNet-18 [17] architecture where the last fully-connected layer is replaced with one with an output dimension of 1024, which is the size of the latent code. We use the 25-patch version of the AtlasNet mesh decoder, where each deformable patch is an open triangular mesh with $5^2 \times 2 = 50$ triangles on a 5×5 regular grid. We redirect the readers to Groueix *et al.* [15] for more details.

In the stage of pretraining AtlasNet on ShapeNet [3] with textured background from SUN360 [37], we train all networks using the Adam optimizer [23] with a constant learning rate of 10^{-4} . We set the batch size for all experiments to be 32. We initialize the AtlasNet encoder with the pretrained ResNet-18 on ImageNet [31] except for the last modified layer (before the latent code), and we initialize the decoder with that pretrained from a point cloud autoencoder from Groueix *et al.* [15].

A.2. Warp Parameterization Details

We parameterize the rotation component of 3D similarity transformations with the $\mathfrak{so}(3)$ Lie algebra. Given a warp parameter vector $\omega = [\omega_1, \omega_2, \omega_3]^\top \in \mathfrak{so}(3)$, the rotation matrix $\mathcal{R}(\omega) \in \mathbb{SO}(3)$ can be written as

$$\mathcal{R}(\omega) = \exp \left(\begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \right), \quad (6)$$

where $\exp$ is the exponential map (*i.e.* matrix exponential). $\mathcal{R}$ is the identity transformation when ω is an all-zeros vector. The exponential map is Taylor-expandable as

$$\mathcal{R}(\omega) = \exp(\omega_\times) = \lim_{K \rightarrow \infty} \sum_{k=0}^K \frac{\omega_\times^k}{k!}. \quad (7)$$

We implement the $\mathfrak{so}(3)$ parameterization using the Taylor approximation expression with $K = 20$. We have also tried parametrizing the 3D similarity transformations with the self-contained Lie group $\text{Sim}(3)$, where the scale is incorporated into the exponential map; we find it to yield almost identical results. We also take the exponential on the scale s to ensure positivity; the resulting scale does not change when $s = 0$.

A.3. SUN360 Background Data Generation

The background images from SUN360 [37] are cropped from spherical images with a resolution of 1024×512 , using a field of view of 90° . Fig. 12 illustrates an example of the original spherical image and its generated crops.

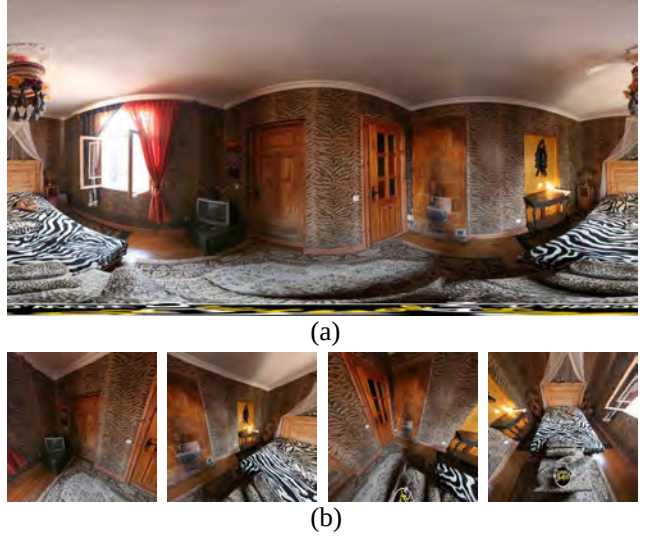


Figure 12: (a) Example panoramic (spherical) image and (b) sample cropped images at different camera viewpoints.