FvOR: Robust Joint Shape and Pose Optimization for Few-view Object Reconstruction

Zhenpei Yang[1] Zhile Ren[2] Miguel Angel Bautista[2]
Zaiwei Zhang[1] Qi Shan[2] Qixing Huang[1]
[1]The University of Texas at Austin [2]Apple

Abstract

Reconstructing an accurate 3D object model from a few image observations remains a challenging problem in computer vision. State-of-the-art approaches typically assume accurate camera poses as input, which could be difficult to obtain in realistic settings. In this paper, we present FvOR, a learning-based object reconstruction method that predicts accurate 3D models given a few images with noisy input poses. The core of our approach is a fast and robust multi-view reconstruction algorithm to jointly refine 3D geometry and camera pose estimation using learnable neural network modules. We provide a thorough benchmark of state-of-the-art approaches for this problem on ShapeNet. Our approach achieves best-in-class results. It is also two orders of magnitude faster than the recent optimization-based approach IDR [67]. Our code is released at https://github.com/zhenpeiyang/FvOR/.

1. Introduction

Reconstructing the 3D shape of objects solely from unregistered RGB inputs is a long-standing problem in computer vision. One popular pipeline is to integrate Structure-from-Motion (SfM) and Multi-view Stereo (MVS) [24,35]. A common principle of this popular pipeline is to recover relative camera poses, establish pixel correspondences (either explicitly or implicitly), and solve triangulation to obtain a dense reconstruction. The success of this paradigm relies on dense image coverage to obtain accurate camera poses and correspondences [1, 18, 19, 49]. Enabled by the emergence of large scale 3D datasets that provide shape priors about 3D objects, a recent line of works focus on learning monocular 3D reconstruction [9, 13, 14, 21, 59, 60]. The general idea is to learn multi-scale correlation priors among different regions of geometric shapes, which are used to infer complete geometry from partial observations.

Acquiring dense input views is crucial for achieving good 3D reconstruction quality on current pipelines, but it is also a very tedious and not user-friendly process. For instance, a casual non-expert user that just began using 3D reconstruction applications (such as creating 3D models of their house), may overlook the strict requirements of capturing high-quality dense views.

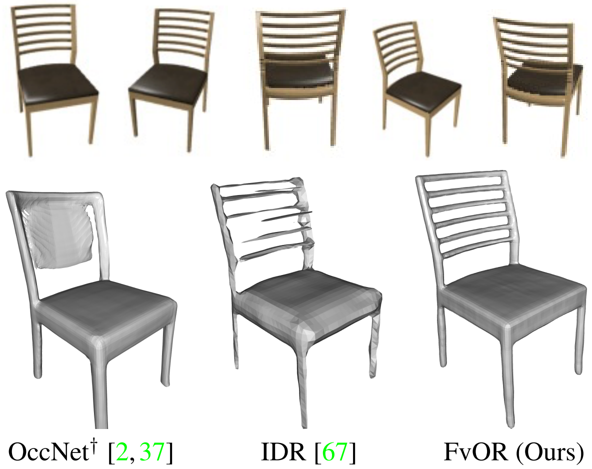

Figure 1. Our approach FvOR outperforms state-of-the-art approaches of few-view 3D reconstruction.

In this paper, we study the setting of few-view reconstruction [9], which sits between dense-view reconstruction and single-view reconstruction. The promise of this setting is that the input views cover the most of underlying object, and one only needs to fill in a small portion of missing regions, a task that is easier to achieve than single-view reconstruction. The ultimate goal is to match the quality of dense reconstruction while significantly reducing the number of inputs. While both few-view reconstruction and single-view reconstruction fall into the category of learning-based approaches, the performance of few-view reconstruction relies on accurate image poses, which could be challenging to estimate from the input images themselves in realistic scenarios. In dense-view reconstruction, the SfM pipeline estimates image poses by first predicting relative camera poses using feature correspondences and then performing synchronization [6, 11] to extract absolute camera poses. However, this pipeline does not apply to few-view reconstruction as there are only a few images, which makes the

[*] Experiments are conducted by Z. Yang at the University of Texas at Austin. Email: yzp@utexas.edu

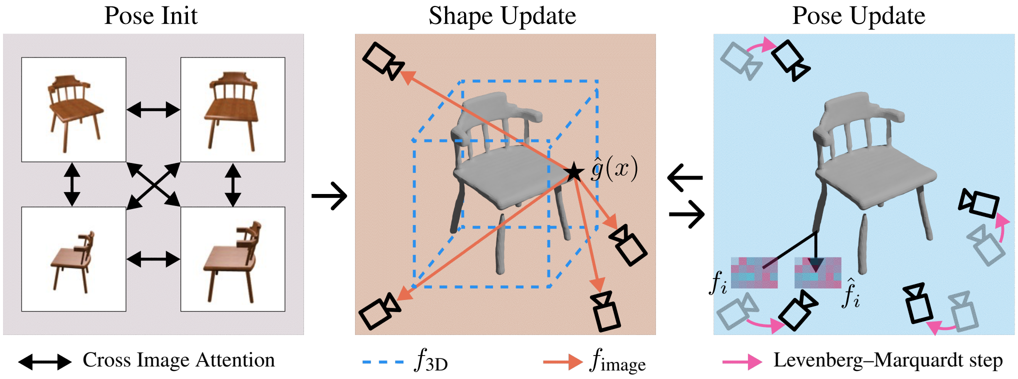

Figure 2. Our approach consists of two stages. The first stage is *pose initialization* which predicts an initial pose for each input image. The second stage alternates between *shape update* and *pose update* to give an accurate reconstruction with jointly improved camera poses.

accurate pose prediction using correspondence difficult.

This paper introduces a novel learning-based approach for joint optimization of the shape reconstruction and the camera poses associated with the input images. The core of our approach consists of a pose initialization module, a shape module, and a pose refinement module. The pose initialization module computes an initial camera pose for each input image. The shape and pose refinement modules are alternated to improve the shape reconstruction and the camera poses jointly. We design the pose initialization module using a geometric approach, aiming to reduce outlier predictions of camera poses that are difficult to rectify in pose refinement. The shape module combines the strengths of per-view image features and 3D convolutional features to obtain an accurate implicit 3D reconstruction with shape details. The pose refinement module performs geometric alignments between rendered images and real images in a learned feature space. Both the shape and pose modules are end-to-end trainable. Compared to existing learning-based image pose estimation techniques, our approach uses a dynamically changing 3D reconstruction and geometric constraints, both of which are unavailable in standard end-to-end pose estimation approaches [16, 27, 54, 62].

Our approach achieves state-of-the-art results on ShapeNet. The shape reconstruction module also improves upon state-of-the-art approaches under the setting of known camera poses. Due to the efficiency of our neural network modules, our approach is two orders of magnitude faster than the recent optimization-based approach IDR [67].

2. Related Work

Single-view Object Reconstruction Typical single-view approaches use an image encoder to estimate a latent code, which is then decoded into 3D shape representations such as voxels [20], point-clouds [15], meshes [22, 56], skeletons [30, 61], or implicit functions [37]. Although this methodology has shown promising results, they are inherently limited to large uncertainties in the invisible regions given the partial visible observations (c.f. [52]).

Dense-view Object Reconstruction Traditional approaches for reconstructing an object usually involve dense scanning around the object, followed by SfM(Structure from Motion) [11, 48, 50, 58] or SLAM(Simultaneous Localization and Mapping) [12, 66] approaches for reconstruction and camera pose estimation. Popular software includes COLMAP [46] and OpenMVG [39]. Deep learning counterparts, for example DeepV2D [3, 53], have also been proposed in recent years. [34] proposed an approach to recover object shape from posed video frames by combing a deep shape prior network with photometric optimization. Recently, the seminar work NeRF [38] inspire many research learning implicit 3D representation from images. Some recent works [33, 36, 67] consider inaccurate camera poses as input and optimize the shape reconstruction [67](or neural radiance field [33, 36]) and poses jointly. This type of approach requires the most expensive capture efforts, but usually gives good performance.

Few-view Object Reconstruction Such a task aims at reconstructing the underlying object given several images. In general, there are two approaches to solving this task: whether they model camera poses explicitly during the inference time, or not. One pioneering work of the pose-free approach is 3D-R2N2 [9], which uses a 3D convolutional LSTM to aggregate multi-view information sequentially. A recent example of this type of approach is Pix2Vox++ [63]. The other approaches model camera poses directly. Many of these approach assume ground truth camera poses as input [2, 29, 41, 43, 57, 64, 65, 68]. For example, [29] use ground truth camera pose to build a volumetric feature representation, which is then decoded into discrete voxel. [65] proposed learning a shape prior during training, and optimizing the shape code to minimize silhouette loss during testing to recover the shape. [64] directly uses predicted camera pose obtained from pre-trained network. Recently, NeRS [69] proposed a NeRF-style few-view reconstruction method using a neural surface representation. Our approach innovates in learning shape reconstruction and pose estimation using deep learning. As a result, our approach does not require object masks [67, 69] or category-specific mesh initialization [69]. Moreover, our approach requires only a few updates, and the running time is significantly faster than IDR [67] and NeRS [69].

3. Approach

We first introduce the problem statement and an overview of our approach in Sect. 3.1. We then elaborate on the technical contributions from Sect. 3.2 to Sect. 3.5.

3.1. Few-view Object Reconstruction

Problem statement. Given a set of RGB images $\mathcal{I} = \{I_i \,|i = 0, \ldots, k-1\}$ observing a single object, where k is the number of observations, we aim to recover the 3D mesh model S of the underlying object, up to a global similarity transformation. We assume the camera intrinsic matrix K is known and fixed across all views.

Approach overview. Fig. 2 is an overview of FvOR. It starts with a pose initialization module that predicts camera poses for each image. This module gives us initial pose estimates with acceptable accuracy. We then alternate between reconstructing the shape from input images with current poses and performing image-shape alignment to refine the poses of each input image. For the shape reconstruction module, we combine a two-stream network that integrates image-based features with 3D features. Image-shape alignment is performed in a learned feature space between input images and corresponding rendered images of the predicted shape. Both modules are end-to-end differentiable. We alternate between the shape and pose modules to reconstruct an accurate 3D model from few-view inputs.

A common approach for training the alternating mechanism is to stitch the alternating shape, pose modules together, and enforce a loss on the final output. We found that this strategy is challenging to train and is not very flexible. In the same spirit as the gradient operators in alternating minimization, this paper trains each module in isolation while forcing them to make progress under different inputs. For example, the pose module is learned to recover the underlying ground truth under randomly perturbed poses. This methodology offers excellent flexibility in developing training losses and instilling training data.

3.2. Pose Initialization Module

The goal of the pose initialization module is to provide initial camera poses for subsequent shape and pose optimization steps. As the camera poses can be refined later, we design a pose initialization module to reduce the number of pose outliers, which are hard to rectify later in the pose refinement stage. For each pixel of each input image, we predict its 3D coordinate in a world coordinate system (scene coordinate) of the underlying geometry [47, 55]. The pose is then obtained by performing global matching between 2D image pixels and the corresponding 3D points via RANSAC [17]. Our approach exhibits three advantages compared to existing regressing and classification based pose estimation approaches [16, 27]. First, reconstructing the 3D coordinates of each image uses information from all input images during testing, meaning the camera poses are jointly predicted. Second, pose regression enforces geometric constraints between correspondences. Third, RANSAC can efficiently deal with incorrect 3D coordinates.

Scene coordinate prediction. Our model first encodes a 2D feature map for each input image independently. We then use a multi-image attention module to aggregate features from all input images. Inspired by [31, 51], the multi-image attention module is composed by alternating between self-attention and cross-attention blocks. The final output is a 3D coordinate $\hat{p}_{i,j}$ for each pixel. The detailed network design can be found in the supp. Network training minimizes the l_2 distances between the predicted and ground-truth scene coordinates. The loss for one set of input images is given by

$$\mathcal{L}_{\text{init}} = \sum_{i=0}^{k-1} \sum_{j=0}^{h\times w-1} w_{i,j} \| \hat{p}_{i,j} - d_{i,j}^{\text{gt}} (R_i^{\text{gt}} K^{-1} \begin{pmatrix} u_{i,j} \\ v_{i,j} \\ 1 \end{pmatrix} + t_i^{\text{gt}}) \|_2$$

where $(u_{i,j}, v_{i,j})$ is the pixel coordinate of the j-th pixel of the i-th input image; $d_{i,j}^{\text{gt}}$ is its ground-truth depth; $T_i^{\text{gt}} := (R_i^{\text{gt}} | t_i^{gt}) \in SE(3)$ is the ground-truth camera to world pose of the i-th image. $w_{i,j}$ is a binary weight indicating whether or not this pixel has G.T. depth.

Pose regression. After we acquired scene coordinate estimates, we use an off-the-shelf RANSAC PnP approach to recover the pose estimates for each input view (details is in the supp.). This initial camera pose serves as input for the subsequent shape optimization module.

3.3. Shape Optimization Module

The shape optimization module takes as input the input images and their pose estimates $\{(I_i, \hat{T}_i^t) | i = 0, \ldots, k-1\}$ and outputs a shape reconstruction. Motivated by the success of implicit shape representations [7, 42], we encode the 3D reconstruction as a deep signed distance function $\hat{g}^t : \mathcal{R}^3 \rightarrow \mathcal{R}$ [42] that outputs the signed distance of any query point in the space.

Our approach innovates computing the implicit function value $\hat{g}^t(\mathbf{x})$ by fusing features from two sources:

$$\hat{g}^t(\mathbf{x}) = g_\Theta\big(\mathbf{f}_{\text{image}}^t(\mathbf{x}), \mathbf{f}_{\text{3D}}^t(\mathbf{x})\big),$$

where g_Θ is a multi-layer fully connected network. Similar to [2, 26, 32, 44, 45, 64, 68], $\mathbf{f}_{\text{image}}^t(\mathbf{x})$ is given by the features extracted from projecting $\mathbf{x}$ onto the input images:

$$\mathbf{f}_{\text{image}}^t(\mathbf{x}) = \text{Pooling}\big(F_i(\mathcal{P}_i(\mathbf{x}, T_i))\big),$$

where $\mathcal{P}_i(\mathbf{x}, T_i^t)$ is the projection of $\mathbf{x}$ on image I_i given the current camera pose T_i^t; F_i is augmented ResNet18 [25] that takes each image as input and outputs the pixel-wise feature map; Pooling is the average pooling function. In addition, $\mathbf{f}_{\text{3D}}^t(\mathbf{x})$ represent features obtained from a 3D feature volume:

$$\mathbf{f}_{\text{3D}}^t(\mathbf{x}) = V(\mathbf{x}|\mathbf{f}_{\text{image}}^t, \Phi),$$

where $V \in \mathcal{R}^{c \times d \times d \times d}$ is a 3D volume produced by a convolutional 3D U-Net [10] with trainable parameters Φ. The input to this 3D U-Net is an initial volume built by evaluating $\mathbf{f}_{\text{image}}^t(\mathbf{x})$ where $\mathbf{x}$ is the coordinates at $d \times d \times d$ grid position (See the supp.).

Network training. In addition to supervising the implicit shape reconstruction using ground-truth signed distance values, we also force the gradient field of the implicit representation to match the corresponding ground truth:

$$\min_g \sum_{x_i \in \mathcal{S}_0} \|g(x_i) - s_i^{\text{gt}}\|_1 + \lambda_{\text{grad}} \sum_{x_i \in \mathcal{S}_1} \Big\| \frac{\partial g}{\partial x_i} / \| \frac{\partial g}{\partial x_i} \| - n_i^{\text{gt}} \Big\|_2$$

where $\mathcal{S}_0$ are points sampled in the 3D space as done in DeepSDF [42]; $\mathcal{S}_1$ are points in on the surface of the underlying object.

3.4. Pose Optimization Module

We now describe the module for updating the camera pose estimates given the current 3D reconstruction. Specifically, the input consists of the input images $\mathcal{I}$, the current implicit shape representation $\hat{g}^t$, and the current camera poses $\{\hat{T}_i^t | i = 0, \ldots, k-1\}$. The output of this module is the pose updates $\Delta\hat{T}^t = \{\Delta\hat{T}_i^t\}_{0 \leq i < k}$ of the corresponding camera poses. Our key idea is to perform geometric alignment between the 3D reconstruction and the input images on a learned feature representation. This is achieved by rendering the 3D reconstruction and then aligning features extracted from the rendered images and the corresponding input images. The training objective enforces that the pose updates derived from these modules match the underlying ground truth. We now describe the technical details.

Efficient renderer. The efficiency of end-to-end learning of the pose module depends on the efficiency of the implicit function renderer. Therefore, unlike IDR [67] that repeatedly evaluates the implicit function to find the accurate intersection point, we use a volumetric grid of size $d \times d \times d$ ($d = 64$ in our implementation) to discretize the implicit function and then render the discretized implicit function. Rendering is achieved using sphere tracing [23, 28]. Such discretization allows us to render 224×224 images at 79.1 FPS (IDR's speed is only 0.32) using a Nvidia V100 GPU. The output of this module is a depth map which is converted into a 2D object mask $\hat{M}^t$ and a set of 3D points that represent the visible region of the current 3D reconstruction.

Learning feature alignment. The goal of this sub-module is to find an incremental pose update $\Delta T_i^t \in SE(3)$ to better align the rendered object mask $\hat{M}_i$ and the input image I_i. The goal is to ensure that this sub-module is end-to-end trainable while utilizing as much information as possible. Instead of directly aligning $\hat{M}_i$ and I_i, we use a neural network to compute a dense feature space to align $\hat{M}_i$ and I_i. Since the underlying pose between $\hat{M}_i$ and I_i is expected to be small, we found that it is sufficient to enforce the loss on pose update derived from aligning the corresponding points in the feature space.

Specifically, let f_Θ denote the network that computes the dense image descriptor, and $\hat{f}_i = f_\Theta(\hat{M}_i)$ and $f_i = f_\Theta(I_i)$ be the resulting feature map. Denote current camera pose for image i as $T_i^0 := (R_i^0 | t_i^0)$, and its corresponding G.T. as $T_i^{\text{gt}} := (R_i^{\text{gt}} | t_i^{\text{gt}})$. We employ the exponential map parametrization of the pose correction $\Delta T_i^t = \begin{pmatrix} e^{[\Delta c_i]_\times} & \Delta t_i \\ 0 & 1 \end{pmatrix}$. Let $\mathcal{P}_i = \{p_j\}$ collect the 3D points of I_i derived from rendering. We propose to compute $\mathbf{c}_i$ and $\mathbf{t}_i$ by solving following non-linear least square problem:

$$\min_{\Delta c_i, \Delta t_i} \sum_{p_j \in \mathcal{P}_i} \|f_i(P(K\mathbf{p}_{j'})) - \hat{f}_i(u_j, v_j)\|_{\mathcal{F}}^2, \qquad (1)$$

$$\begin{pmatrix} \mathbf{p}_j' \\ 1 \end{pmatrix} = \begin{pmatrix} I + [\Delta c_i]_\times & \Delta t_i \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{p}_j \\ 1 \end{pmatrix},$$

$$\mathbf{p}_j = d_j K^{-1}(u_j, v_j, 1)^T.$$

Here (u_j, v_j) denotes the pixel coordinates of p_j in the rendered image. $P(K\mathbf{p}_{j'})$ denotes the corresponding pixel coordinates on the input image after applying the pose update.

To obtain an explicit expression of Δc_i and Δt_i, we use a linear approximation. This leads to the following expression, which applies one-step of Levenberg–Marquardt [40]:

$$(\Delta c_i, \Delta t_i)^T = -(J^T J + \lambda I)^{-1}(J^T r), \qquad (2)$$

where $r = \sum_j r_j$;$J = \sum_j J_j$;I is the identity matrix;λ is a constant. Moreover, r_j and J_j are given by

$$\begin{aligned} r_j &= f_i(P(Kp_j')) - \hat{f}_i(u_j, v_j) \qquad (3) \\ J_j &= \frac{\partial r_j}{\partial(\Delta c_i, \Delta t_i)^T} \\ &= \frac{\partial r_j}{\partial f_i} \frac{\partial f_i}{\partial(u_i', v_i')} \frac{\partial(u_i', v_i')}{\partial P} \frac{\partial P}{\partial p_j'} K \frac{\partial p_j'}{(\Delta c_i, \Delta t_i)^T} \end{aligned}$$

Training Details For training the pose refinement module, we simulate input poses by adding random perturbations to the ground truth camera poses [33]. We then train the pose refinement module by forcing it to recover from the perturbations. We then minimize the following loss:

$$\mathcal{L}_{\text{pose\_refine}} = \sum_i \Big\| \begin{pmatrix} I + [\Delta c_i]_\times & \Delta t_i \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} R_i^0 & t_i^0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} R_i^{\text{gt}} & t_i^{\text{gt}} \\ 0 & 1 \end{pmatrix} \Big\|_{\mathcal{F}}^2$$

Algorithm 1 FvOR 3D Recon Algorithm

```
Input: $\mathcal{I}$, $\hat{\mathcal{T}}_0$
$\hat{g}^0 \leftarrow$ shape_update$(\mathcal{I}, \hat{\mathcal{T}}^0)$
**for** $t = 0 : n_1$ **do**
    **for** $j = 0 : n_2$ **do**
        $\hat{M}^t \leftarrow$ render$(\hat{g}^t, \hat{\mathcal{T}}^t)$
        $\hat{\mathcal{T}}^t \leftarrow$ pose_update$(\mathcal{I}, \hat{g}^t, \hat{\mathcal{T}}^t, \hat{M}^t)$
    **end for**
    $\hat{g}^{t+1} \leftarrow$ shape_update$(\mathcal{I}, \hat{\mathcal{T}}^t)$
    $\hat{\mathcal{T}}^{t+1} \leftarrow \hat{\mathcal{T}}^t$
**end for**
Return $\hat{g}^{n_1}, \hat{\mathcal{T}}^{n_1}$
```

3.5. Alternating Shape and Pose Optimization

We first train the shape module 3.3 using GT poses. Then, we add noise to the GT poses to fine-tune the shape module and use the shape module's prediction on the fly to train the pose update module 3.4 with a single step update. At inference time, we make two modifications: first, we alternate between shape update and pose update for multiple iterations instead of the single iteration used in training. Second, we add a regularization term that penalizes very large deviations from initial pose estimates. The complete algorithm is shown in Algorithm 1. We set $n_1 = 3$ and $n_2 = 5$ in our experiments.

4. Experimental Results

In this section, we present our experimental results. We first describe the datasets used for evaluation in Sect. 4.1. Then, we introduce the baselines for camera pose estimation (Sect. 4.2) and 3D reconstruction (Sect. 4.3). In Sect. 4.4 we discuss the evaluation metrics. Finally, we provide an analysis of results in Sect. 4.5

4.1. Datasets

ShapeNet. This dataset was introduced by 3D-R2N2 [8] based on ShapeNet [5] and has become a widely used benchmark for single/multi-view 3D reconstruction. It contains objects from 13 categories from ShapeNet v1 [5]. For each object, it contains 24 views from a camera pointing to the origin, and has large azimuth variation but small elevation variation. We follow the training/test splits and evaluation protocol in [2] and randomly sample 5 views out of the 24 views to form an input set.

4.2. Baselines for Pose Initialization

In this section, we describe different baseline approaches for initializing the pose estimates.

DISN [64] parametrizes camera poses by orthogonal vectors, and introduces a novel loss for regression-based pose estimation. We implement DISN's camera pose estimation algorithm based on our framework. Note that for baseline comparison, we report results of our updated implementation, which improve from the original results in [64].

	DISN [64]	Cai *et al.* [4]	FvOR-Quat	FvOR
Base	3.66	5.10	4.46	3.82
Base + Cross	**2.46**	**2.25**	**3.06**	**1.40**

Table 1. Ablation study of the pose initialization module on ShapeNet. The results are the Pixel Error↓. We can see that cross-attention can help predict more accurate image poses. "Base" means per-image prediction is used. "Base+Cross" means that each image's feature map interacts with other images through cross-image attention.

Metrics	All	w/o f_{image}	w/o f_{3D}	w/o $\mathcal{L}_{\text{grad}}$
IoU↑	**0.783**	0.782	0.718	0.759
Chamfer-L1↓	**0.058**	0.060	0.082	0.066

Table 2. Ablation study of the decoder design and the gradient loss. We use ShapeNet and ground truth poses in this experiment. The results are averaged across all 13 categories.

Cai *et al.* [4] Extreme-Rot is a recent deep learning method for estimating pair-wise relative rotations between two images with little overlap. It estimates the rotation by predicting a distribution over discretized Euler angle bins. We implement Extreme-Rot based on our framework and modify it to predict the absolute pose for each image (details are in the supp. material).

FvOR-Quat. The above two methods use the continuous rotation matrix and discrete Euler angle representations. In addition, we add a baseline that predicts a quaternion/translation vector. It shares the same backbone as our method. But instead of predicting per-pixel scene coordinates, it averages the feature map of each image to a single vector and regresses the quaternion and translation.

4.3. Baselines for 3D Reconstruction

We now describe the 3D reconstruction baselines that we use to evaluate the effectiveness of our approach.

OccNet [37] is a top performing method for single-view 3D reconstruction . We follow the practice of 3D43D [2] which provides a multi-view augmented version of OccNet denoted as **OccNet**† [2, 37]. For Tab. 3, we use the evaluation results provided in 3D43D [2].

Pix2Vox++ [63] is a recent work that provides an improved framework to 3D-R2N2 [9] with multi-scale context-aware fusion. We train their model on ShapeNet using our settings (5 views). We evaluate their prediction against continuous mesh instead of the discretized version [37].

3D43D [2] is a recent work that uses pixel-aligned feature representations and multi-view images with ground-truth camera poses for object 3D reconstruction.

IDR [67] is an optimization based algorithm that does not learn a prior from training data. IDR achieves good performance when reconstructing objects with tens of images paired with ground truth object masks. We run IDR [67]

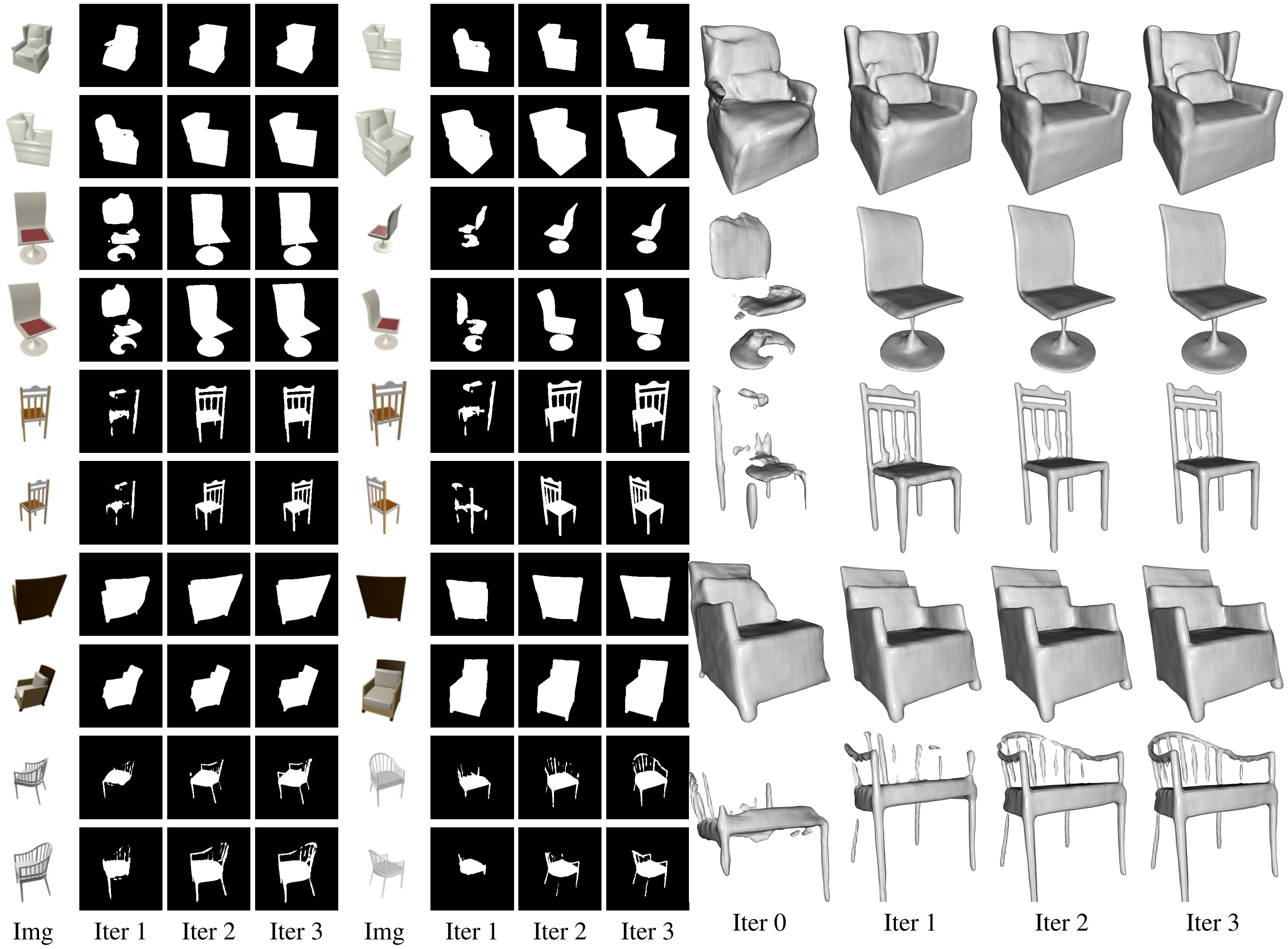

Figure 3. Visualizing the intermediate results of FvOR. On the left we visualize 4 input views (out of 5 views used) and the back-projected masks in each iteration. On the right we show the progression of 3D reconstructions through alternating shape and pose optimizations.

Category	w/o GT Pose				w/ GT Pose	
	OccNet [37]	OccNet† [2]	Pix2Vox++	FvOR	3D43D [2]	FvOR w/ GT Pose
plane	0.591 / 0.134 / 0.845	0.600 / 0.096 / 0.853	0.366/-/-	0.726 / 0.0541 / 0.905	0.736 / 0.021 / 0.899	0.802 / 0.032 / 0.930
bench	0.492 / 0.150 / 0.814	0.547 / 0.176 / 0.834	0.328/-/-	0.654 / 0.0530 / 0.904	0.663 / 0.027 / 0.881	0.710 / 0.047 / 0.913
cabinet	0.750 / 0.153 / 0.884	0.770 / 0.125 / 0.893	0.601/-/-	0.844 / 0.0773 / 0.931	0.831 / 0.073 / 0.925	0.83 / 0.067 / 0.936
car	0.746 / 0.149 / 0.852	0.759 / 0.109 / 0.861	0.581/-/-	0.768 / 0.0953 / 0.874	0.797 / 0.090 / 0.873	0.800 / 0.088 / 0.882
chair	0.530 / 0.206 / 0.829	0.568 / 0.1870.846	0.430/-/-	0.690 / 0.0730 / 0.918	0.716 / 0.063 / 0.911	0.746 / 0.057 / 0.932
display	0.518 / 0.258 / 0.857	0.593 / 0.168 / 0.884	0.443/-/-	0.754 / 0.0769 / 0.935	0.752 / 0.089 / 0.935	0.794 / 0.058 / 0.950
lamp	0.400 / 0.368 / 0.751	0.415/ 1.083 / 0.764	0.277/-/-	0.599 / 0.116 / 0.867	0.625 / 0.256 / 0.858	0.682 / 0.069 / 0.893
speaker	0.677 / 0.266 / 0.848	0.699 / 0.360 / 0.856	0.588/-/-	0.793 / 0.109 / 0.908	0.807 / 0.143 / 0.912	0.807 / 0.089 / 0.919
rifle	0.480 / 0.143 / 0.783	0.466 / 0.112 / 0.789	0.338/-/-	0.705 / 0.0476 / 0.913	0.745 / 0.012 / 0.903	0.823 / 0.0269 / 0.944
sofa	0.693 / 0.181 / 0.867	0.731 / 0.171 / 0.886	0.554/-/-	0.804 / 0.0748 / 0.938	0.809 / 0.054 / 0.927	0.834 / 0.063 / 0.943
table	0.542 / 0.182 / 0.860	0.569 / 0.588 / 0.873	0.373/-/-	0.654 / 0.0726 / 0.923	0.689 / 0.058 / 0.921	0.706 / 0.060 / 0.934
phone	0.740 / 0.127 / 0.939	0.785 / 0.103 / 0.948	0.589/-/-	0.855 / 0.0434 / 0.978	0.861 / 0.017 / 0.971	0.875 / 0.039/ 0.977
boat	0.547 / 0.201 / 0.797	0.592 / 0.163 / 0.818	0.437/-/-	0.712 / 0.0816 / 0.884	0.708 / 0.053 / 0.868	0.763 / 0.064 / 0.906
Mean	0.593 / 0.194 / 0.840	0.621 / 0.265 / 0.854	0.455/-/-	**0.735 / 0.075 / 0.914**	0.749 / 0.073 / 0.906	**0.783 / 0.058 / 0.928**

Table 3. Quantitative results of few-view 3D reconstruction on the ShapeNet dataset. The numbers in each cell is (IoU / Chamfer-L1 / F-score). OccNet [37] uses a single view. The rest of the methods use 5 views. The last two columns show methods that use GT camera poses. We do not factor out similarity because we obtained results for OccNet/OccNet†/3D43D directly from original papers. Chamfer-L1 is multiplied by 10 [37].

on each test inputs for 1000 epochs on ShapeNet for best results.

FvOR w/ GT Pose is our standalone 3D reconstruction module trained with ground truth camera poses. This setting is also used in 3D43D [2], but we differ from 3D43D in network architecture. Although DISN [64] also shown in qualitative results for multi-view reconstruction in their paper, we do not find their official implementation for multi-view reconstruction. Instead, we compare a variant of our method w/o f_{image} which is similar to DISN, in Tab. 2.

FvOR + Noise@L{1,2,3}. For this baseline, we train our standalone 3D reconstruction module with noisy input poses. In order to do this, we add Gaussian noise to the camera poses at 3 different levels of standard deviation ($\sigma \in \{0.75\text{e}{-}2, 1.5\text{e}{-}2, 2.25\text{e}{-}2\}$) [33].

FvOR w/o Joint is our proposed approach without performing iterative refinement during inference. We use this baseline to demonstrate the importance of pose optimization for improving robustness to noisy poses.

4.4. Evaluation Metrics

Pose Initialization. We evaluate the camera pose estimation accuracy using three metrics. The first metric is *Pixel-Error*, which is calculated by first projecting the object's surface point into each view using predicted pose and GT pose, and then calculating the corresponding distance in the pixel space. The other two metrics are *Rotation Error* and *Translation error*. *Pixel-Error* is a more reasonable metric for evaluating poses for multi-view 3D reconstruction, as it reflects both the rotation and translation errors [64].

3D Reconstruction We measure the distance between a predicted 3D mesh $\hat{S}$ and a ground truth 3D mesh S using common metrics [2, 37, 64] including *IoU*, *Chamfer-L1 distance*, and *normal consistency*. To eliminate the influence of predicted poses on the final mesh reconstruction, for each method we factor out the similarity transformation between the reconstructed mesh and the underlying ground-truth mesh for evaluation in Tab. 4(details in the supp). However, to provide a direct comparisons with previous approaches, we do not perform this alignment for results reported in Tab. 3).

4.5. Results Analysis

Pose initialization In Tab. 1, we report results for the different pose initialization baselines on the ShapeNet. The per-category results on ShapeNet can be found in the supp. We make two key observations from these results. First, our approach is the top-performing approach on both datasets. In particular, our method has only a 1.40 pixel error on ShapeNet. Given that all the baselines for pose estimation share the same backbone, these results demonstrate the benefits of our scene-coordinate representation of camera poses. Second, we also observe from Tab. 1 that removing the cross-view attention module reduces the accuracy of the pose estimates across the board. This illustrates the importance of aggregating information across multiple views.

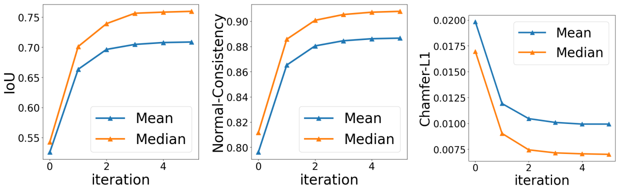

Figure 4. Per iteration results of the refinement approach on ShapeNet under Noise@L3. Our iterative refinement approach has improved 3D reconstruction scores. The refinement process also converges quickly in 3∼ 4 iterations.

Few-view 3D reconstruction w/ GT pose. To validate our 3D reconstruction module design, we perform multi-view 3D reconstruction experiments on the ShapeNet dataset using ground truth camera poses. The results can be found in Tab. 3 (FvOR †) and Tab. 2. Tab. 3 shows that our method is the top performer, achieving a 0.783 IoU, which is a 4.5% relative improvement compared with the previous state-of-the-art. In addition, the ablation results in Tab. 2 show that removing the pixel-aligned feature (f_{image}), 3D convolutional feature (f_{3D}) or removing the gradient loss ($\mathcal{L}_{\text{grad}}$) have negative impacts on the reconstruction accuracy, both in terms of IoU and Chamfer-L1.

Robustness of few-view 3D reconstruction under noisy poses. We experiment two settings of camera poses. In the first setting, a Gaussian noise is applied to the ground truth camera poses, following the practice of BARF [33]. The pose perturbation magnitude is controlled by $\sigma \in \{0.75\text{e}{-}2, 1.5\text{e}{-}2, 2.25\text{e}{-}2\}$ with three values (called L1, L2, and L3 respectively). The corresponding average pixel errors are $\{2.29, 4.58, 6.88\}$ on ShapeNet. In Tab. 4, we show results on ShapeNet. The first observation is that FvOR trained with ground truth poses (FvOR w/ GT Pose) is highly sensitive to noisy pose initialization at inference time. In particular, the average IoU drops from 0.806 to 0.667 with L1 noise, and drops further to 0.441 with L3 noise. A second observation is that FvOR trained with noisy poses (FvOR w/ Noise@L$\{1, 2, 3\}$) (rows 3–5 in Tab. 4) gains robustness at the trained noise level, as expected. For example, FvOR w/ Noise@L1 achieves an IoU of 0.743 at test time with noise level L1, far exceeding the IoU of 0.667 achieved with FvOR w/ GT Pose. On the other hand, the robustness of these models (FvOR w/ Noise@L$\{1, 2, 3\}$) comes at a cost of decreased performance when accurate camera poses are given, which is expected as they simply fit the network to noisy pose without explicitly modeling(*e.g.* the first column in Tab. 4).

In contrast, FvOR with joint shape and pose iterative refinement is a lot more robust to noisy poses, while retaining high performance when the poses become accurate. In

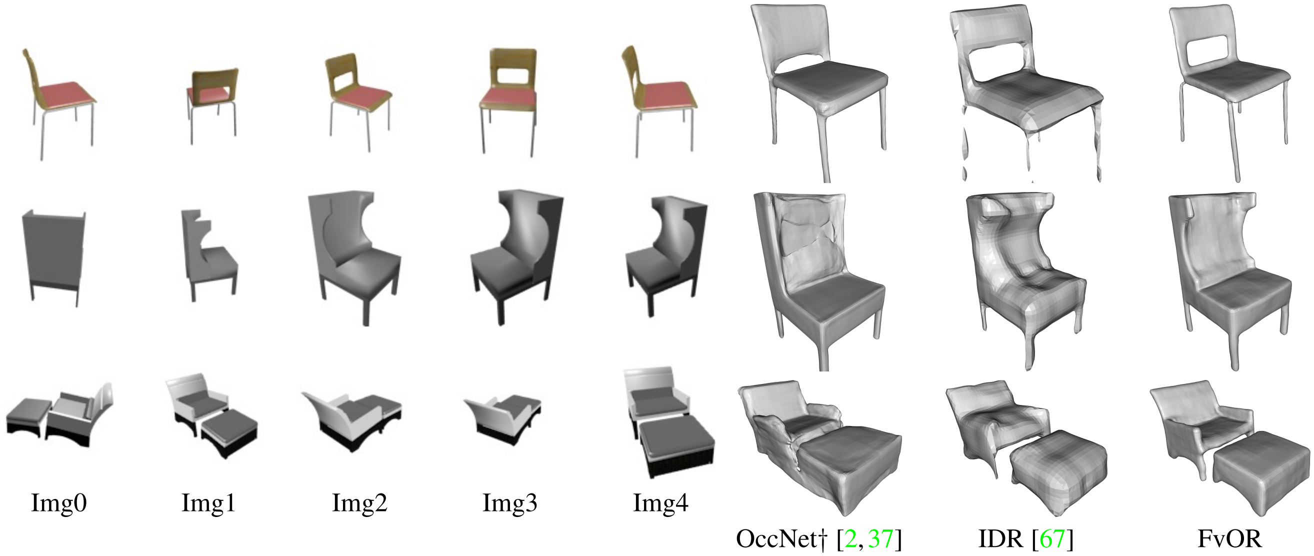

Figure 5. Qualitative comparison on ShapeNet. On the left we show the five input images. On the right we show the prediction of OccNet† [2, 37], IDR [67] and FvOR(ours). IDR and FvOR use our predicted camera poses.

Method	GT		Noise@L1		Noise@L2		Noise@L3		Predict	
	IoU↑	Chamfer-L1↓	IoU↑	Chamfer-L1↓	IoU↑	Chamfer-L1↓	IoU↑	Chamfer-L1↓	IoU↑	Chamfer-L1↓
OccNet† [2, 37]	0.667/0.702	1.22/0.989	0.667/0.702	1.22/0.989	0.667/0.702	1.22/0.989	0.667/0.702	1.22/0.989	0.667/0.702	1.22/0.989
IDR [67]	0.392/0.358	4.1/2.4	0.419/0.444	4.0/2.2	0.393/0.364	4.3/2.6	0.415/0.396	3.9/2.2	0.392/0.370	4.3/2.4
FvOR w/ Noisy@L1	0.760/0.795	0.744/0.642	0.743/0.777	0.811/0.694	0.688/0.722	1.07/0.899	0.601/0.631	1.64/1.33	0.751/0.787	0.796/0.659
FvOR w/ Noisy@L2	0.725/0.757	0.899/0.741	0.718/0.748	0.935/0.777	0.703/0.733	0.998/0.851	0.676/0.710	1.14/0.964	0.721/0.752	0.920/0.756
FvOR w/ Noisy@L3	0.704/0.741	0.977/0.828	0.698/0.731	1.01/0.858	0.689/0.724	1.06/0.906	0.677/0.712	1.13/0.966	0.700/0.739	1.02/0.834
FvOR w/ GT Pose	**0.806/0.841**	**0.605/0.498**	0.667/0.699	1.17/0.985	0.531/0.557	2.05/1.70	0.441/0.443	2.89/2.29	**0.785/0.825**	**0.677/0.543**
FvOR w/o Joint	0.786/0.820	0.658/0.554	0.749/0.779	0.777/0.667	0.645/0.677	1.27/1.09	0.533/0.551	2.06/1.78	0.775/0.812	0.702/0.573
FvOR	0.783/0.818	0.664/0.561	**0.779/0.814**	**0.676/0.571**	**0.766/0.803**	**0.735/0.600**	**0.721/0.768**	**0.988/0.707**	0.773/0.812	0.708/0.576

Table 4. Evaluating the robustness of few-view 3D reconstruction baselines on ShapeNet (mean/median, top-2 results highlighted). We report the results using ground truth poses, perturbed poses with different perturbation levels, and predicted poses from our pose estimation module. Chamfer-L1 is multiplied by 100. Our approach(FvOR) can strike a balance between being robust to noisy poses and obtaining high reconstruction accuracy. The details of three noise levels can be found in Section 4.5. Note that differently from Table 3, here we pre-align the predicted shape with ground truth shape before evaluation to focus on accessing shape quality. Since OccNet† always predicts a unit scale shape. We've factored out the shape scale when computing the Chamfer-L1 metric of OccNet† for a fair comparison.

Metric	IoU↑	Chamfer-L1↓	Inference Speed(s)↓
IDR [67]	0.392	4.33	1.0×10^3
FvOR	**0.773**	**0.708**	**9.8**

Table 5. Inference speed for ShapeNet dataset. As an optimization based approach, our method is significant faster than IDR [67].

Fig. 4 we show how reconstruction metrics improve as a function of the number of Levenberg–Marquardt updates in the refinement process. Fig. 3 shows how the rendered masks and geometry improve at each iteration.

In the second setting, we use predicted poses produced by our pose initialization method during inference. This is shown in the last column of Tab. 4. We observe that our iterative refinement approach (FvOR) does not provide further gains(i.e. FvOR w/o joint) in this case(last 2 rows of Tab. 4), which is also expected because on ShapeNet dataset the predicted pose are already fairly close to G.T. 1, and our pose update module are designed to address considerable pose error. Qualitative comparison between FvOR and existing methods on the ShapeNet dataset can be found in Fig. 5. FvOR outperforms existing methods significantly.

Computational speed. We found our approach typically converges after 3 shape and pose updates, while IDR [67] requires thousands of updates. The inference speed can be found in the Tab. 5.

5. Conclusions and Limitations

Conclusions. This paper studied the problem of reconstructing a 3D object from a few observations. We proposed a joint pose and shape refinement approach that strikes a balance between being robust to noisy camera poses and producing accurate 3D reconstructions.

Limitations. A limitation of our approach is that separate training of shape and pose module may result in sub-optimal performance. Another limitation is the pose optimization module requires a reasonable initial shape prediction. We plan to address these limitations in future work.

References

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, and Richard Szeliski. Building rome in a day. *Commun. ACM*, 54(10):105–112, 2011. 1

[2] Miguel Ángel Bautista, Walter Talbott, Shuangfei Zhai, Nitish Srivastava, and Joshua M. Susskind. On the generalization of learning-based 3D reconstruction. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021. 1, 2, 3, 5, 6, 7, 8

[3] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J Davison. CodeSLAM—learning a compact, optimisable representation for dense visual slam. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2560–2568, 2018. 2

[4] Ruojin Cai, Bharath Hariharan, Noah Snavely, and Hadar Averbuch-Elor. Extreme rotation estimation using dense correlation volumes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14566–14575, 2021. 5

[5] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *CoRR*, abs/1512.03012, 2015. 5

[6] Avishek Chatterjee and Venu Madhav Govindu. Efficient and robust large-scale rotation averaging. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 521–528. IEEE Computer Society, 2013. 1

[7] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5939–5948. Computer Vision Foundation / IEEE, 2019. 3

[8] Sungjoon Choi, Q. Y. Zhou, and V. Koltun. Robust reconstruction of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5556–5565, June 2015. 5

[9] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 1, 2, 5

[10] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3D U-Net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016. 4

[11] David Crandall, Andrew Owens, Noah Snavely, and Daniel Huttenlocher. SfM with MRFs: Discrete-continuous optimization for large-scale structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35:2841–53, 12 2013. 1, 2

[12] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006. 2

[13] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2650–2658. IEEE Computer Society, 2015. 1

[14] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2366–2374, 2014. 1

[15] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3D object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 605–613, 2017. 2

[16] Zhaoxin Fan, Yazhi Zhu, Yulin He, Qi Sun, Hongyan Liu, and Jun He. Deep learning on monocular object pose detection and tracking: A comprehensive overview. *CoRR*, abs/2105.14291, 2021. 2, 3

[17] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. 3

[18] Yasutaka Furukawa, Brian Curless, Steven M. Seitz, and Richard Szeliski. Towards internet-scale multi-view stereo. In *CVPR*, pages 1434–1441. IEEE Computer Society, 2010. 1

[19] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(8):1362–1376, 2010. 1

[20] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 484–499. Springer, 2016. 2

[21] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6602–6611. IEEE Computer Society, 2017. 1

[22] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3D surface generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 216–224, 2018. 2

[23] John C Hart. Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer*, 12(10):527–545, 1996. 4

[24] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, USA, 2 edition, 2003. 1

[25] 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 (CVPR)*, pages 770–778. IEEE Computer Society, 2016. 4

[26] Philipp Henzler, Jeremy Reizenstein, Patrick Labatut, Roman Shapovalov, Tobias Ritschel, Andrea Vedaldi, and David Novotny. Unsupervised learning of 3d object categories from videos in the wild. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4700–4709, 2021. 3

[27] Sabera Hoque, Md. Yasir Arafat, Shuxiang Xu, Ananda Maiti, and Yuchen Wei. A comprehensive review on 3d object detection and 6d pose estimation with deep learning. *IEEE Access*, 9:143746–143770, 2021. 2, 3

[28] Yue Jiang, Dantong Ji, Zhizhong Han, and Matthias Zwicker. SDFDiff: Differentiable rendering of signed distance fields for 3d shape optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1251–1261, 2020. 4

[29] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 364–375, 2017. 2

[30] Abhishek Kar, Shubham Tulsiani, Joao Carreira, and Jitendra Malik. Category-specific object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1966–1974, 2015. 2

[31] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020. 3

[32] Youngjoong Kwon, Dahun Kim, Duygu Ceylan, and Henry Fuchs. Neural human performer: Learning generalizable radiance fields for human performance rendering. *Advances in Neural Information Processing Systems*, 34, 2021. 3

[33] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. BARF: Bundle-adjusting neural radiance fields. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 2, 4, 7

[34] Chen-Hsuan Lin, Oliver Wang, Bryan C Russell, Eli Shechtman, Vladimir G Kim, Matthew Fisher, and Simon Lucey. Photometric mesh optimization for video-aligned 3d object reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[35] Yi Ma, Stefano Soatto, Jana Košecká, and Shankar Sastry. *An invitation to 3-D vision: from images to geometric models*, volume 26. Springer, 2004. 1

[36] Quan Meng, Anpei Chen, Haimin Luo, Minye Wu, Hao Su, Lan Xu, Xuming He, and Jingyi Yu. GNeRF: GAN-based neural radiance field without posed camera. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[37] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4460–4470, 2019. 1, 2, 5, 6, 7, 8

[38] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 405–421. Springer, 2020. 2

[39] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. OpenMVG: Open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 60–74. Springer, 2016. 2

[40] Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer series in operations research and financial engineering. Springer, New York, NY, 2. ed. edition, 2006. 4

[41] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[42] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019. 3, 4

[43] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[44] Amit Raj, Michael Zollhofer, Tomas Simon, Jason Saragih, Shunsuke Saito, James Hays, and Stephen Lombardi. Pixel-aligned volumetric avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11733–11742, 2021. 3

[45] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2304–2314, 2019. 3

[46] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[47] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2930–2937, 2013. 3

[48] Noah Snavely. Bundler: Structure from motion (sfm) for unordered image collections. `http://www.cs.cornell.edu/~snavely/bundler/`. 2

[49] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: Exploring photo collections in 3D. *ACM Transactions on Graphics (SIGGRAPH)*, 25(3):835–846, July 2006. 1

[50] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM siggraph 2006 papers*, pages 835–846. 2006. 2

[51] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8922–8931, 2021. 3

[52] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do

single-view 3D reconstruction networks learn? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3405–3414, 2019. 2

[53] Zachary Teed and Jia Deng. DeepV2D: Video to depth with differentiable structure from motion. *International Conference on Learning Representations (ICLR)*, 2019. 2

[54] Shubham Tulsiani and Jitendra Malik. Viewpoints and keypoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1510–1519. IEEE Computer Society, 2015. 2

[55] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J. Guibas. Normalized object coordinate space for category-level 6D object pose and size estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3

[56] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3D mesh models from single RGB images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–67, 2018. 2

[57] 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. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2

[58] Changchang Wu et al. Visualsfm: A visual structure from motion system. 2011. 2

[59] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, Bill Freeman, and Josh Tenenbaum. Marrnet: 3d shape reconstruction via 2.5d sketches. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 540–550, 2017. 1

[60] Jiajun Wu, Tianfan Xue, Joseph J. Lim, Yuandong Tian, Joshua B. Tenenbaum, Antonio Torralba, and William T. Freeman. Single image 3d interpreter network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 9910, pages 365–382. Springer, 2016. 1

[61] Jiajun Wu, Tianfan Xue, Joseph J Lim, Yuandong Tian, Joshua B Tenenbaum, Antonio Torralba, and William T Freeman. Single image 3D interpreter network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 365–382. Springer, 2016. 2

[62] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. In Hadas Kress-Gazit, Siddhartha S. Srinivasa, Tom Howard, and Nikolay Atanasov, editors, *Robotics: Science and Systems XIV, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, June 26-30, 2018*, 2018. 2

[63] Haozhe Xie, Hongxun Yao, Shengping Zhang, Shangchen Zhou, and Wenxiu Sun. Pix2vox++: multi-scale context-aware 3d object reconstruction from single and multiple images. *International Journal of Computer Vision (IJCV)*, 128(12):2919–2935, 2020. 2, 5

[64] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. DISN: Deep implicit surface network for high-quality single-view 3d reconstruction. *Advances in Neural Information Processing Systems (NeurIPS)*, pages 492–502, 2019. 2, 3, 5, 7

[65] Mingyue Yang, Yuxin Wen, Weikai Chen, Yongwei Chen, and Kui Jia. Deep optimized priors for 3d shape modeling and reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3269–3278, 2021. 2

[66] Shichao Yang and Sebastian Scherer. Cubeslam: Monocular 3D object slam. *IEEE Transactions on Robotics*, 35(4):925–938, 2019. 2

[67] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020. 1, 2, 3, 4, 5, 8

[68] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4578–4587, 2021. 2, 3

[69] Jason Y Zhang, Gengshan Yang, Shubham Tulsiani, and Deva Ramanan. Ners: Neural reflectance surfaces for sparse-view 3d reconstruction in the wild. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2, 3