3DFIRES: Few Image 3D REconstruction for Scenes with Hidden Surfaces

Linyi Jin[1], Nilesh Kulkarni[1], David F. Fouhey[2]
University of Michigan[1], New York University[2]
{jinlinyi,nileshk}@umich.edu, david.fouhey@nyu.edu

Abstract

This paper introduces 3DFIRES, a novel system for scene-level 3D reconstruction from posed images. Designed to work with as few as one view, 3DFIRES reconstructs the complete geometry of unseen scenes, including hidden surfaces. With multiple view inputs, our method produces full reconstruction within all camera frustums. A key feature of our approach is the fusion of multi-view information at the feature level, enabling the production of coherent and comprehensive 3D reconstruction. We train our system on non-watertight scans from large-scale real scene dataset. We show it matches the efficacy of single-view reconstruction methods with only one input and surpasses existing techniques in both quantitative and qualitative measures for sparse-view 3D reconstruction. Project page: https://jinlinyi.github.io/3DFIRES/

1. Introduction

Consider two views of the scene in Fig. 1. Part of the bedroom in View 1 is occluded by the wall, and so you may be uncertain what is behind it, although you might guess the wall continues. Now consider adding in View 2. You can see a bedside table, but little else. However, you can fuse these pieces together to create a consistent 3D sense of the scene viewed by the images, including both the visible and invisible parts. We use this sense when shopping for real estate or looking at a friend's photos. We estimate the structure of the scene from parts that are visible to all views; integrate information across images for parts that visible in one view but not others; and take educated guesses for completely occluded regions. Importantly, as the available data increases from one camera to a handful, we can seamlessly integrate the evidence across views.

This task poses a challenge for current computer vision since it requires making judgments about visible and occluded 3D structures and integrating information across images with large pose change. These abilities are usually independently investigated in two separate strands of research. With single image reconstruction techniques [15, 20, 26, 41, 43], one can predict both visible and occluded 3D structure from an image, but stacking such outputs from multiple images can produce inconsistent outputs. When handled independently, methods cannot identify the best view to reason about an occluded region. Non-line-of-sight imaging involves transmitting and receiving signals to reveal hidden scenes, incompatible with standard camera images [14]. Sparse view reconstruction methods [1, 17, 39] can create consistent reconstructions from two views; however, these approaches are limited to the visible parts of the scene that can decomposed into planes. Moreover, these methods are usually specialized to a particular number of images that can be accepted.

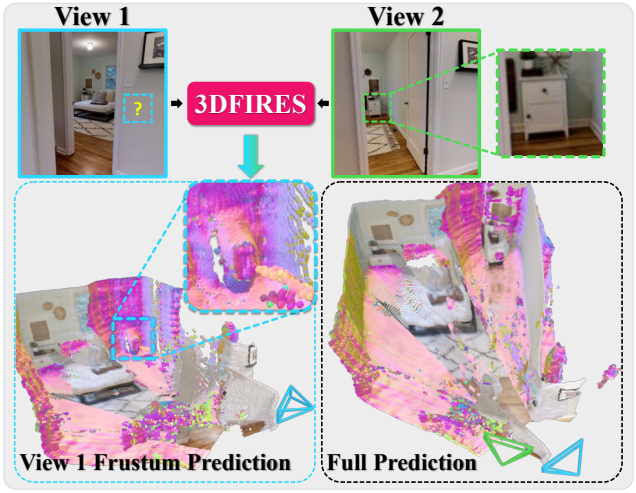


Figure 1. **Reconstructing 3D from sparsely posed images**. Given a sparse set of posed image views, our method is able to reconstruct the full 3D of the scene. On the top, we show two sparse views of the scene in View 1 and View 2. On the bottom left is the 3D reconstruction from our network in the frustum of View 1. We show that our method can generate the occluded side table (zoom in). On the bottom right is the full reconstruction. We color occluded surfaces with surface normals.

Recently, there has been considerable progress in generalized radiance fields, which produce full 3D representations. This occupancy representation and per-scene optimization has shown promising results by optimizing for

novel view synthesis on single scenes from posed images sets [7, 23, 36, 40]. Extending this line of work, methods like [32, 46] have shown an ability to predict novel views for unseen scenes from a few images. However, since these methods optimize for perceptual quality, the underlying geometry often has artifacts. Like them we also require one or more image views at input, but instead we predict an implicit function [20] that can reliably reconstruct both visible and occluded parts of previously unseen scenes.

We propose 3DFIRES, **F**ew **I**mage 3D-**RE**construction of **S**cenes, which integrates information from a variable number of images to produce a *full reconstruction* of the scene. 3DFIRES integrates information in the features space across a varying number of images, enabling it to identify how to best use the available image data to produce an accurate reconstruction at a point. As output, 3DFIRES produces a pixel-aligned implicit field based on a generalization of the Directed Ray Distance Function [20, 21], which enables high quality reconstructions. Thanks to integration in feature space, the results are more consistent than handling images independently: this is what enables reconstructing the bed-side table in Fig. 1, even though it is hidden by the wall in one image. We found and document several design decisions in terms of training and network architecture needed to produce these results.

We evaluate our method on complex interior scenes from Omnidata [8, 33] dataset collected with a real scanner. We compare 3DFIRES with the point-space fusion of state-of-the-art methods for scene-level full 3D reconstruction methods from a single image [21, 43]. Our experiments show several key results. First, 3DFIRES produces more accurate results compared to existing works. The improvements are larger in hidden regions, and especially substantial when measuring consistency of prediction from multiple views. Second, ablative analysis reveals the key design decisions responsible for 3DFIRES's success. Third, 3DFIRES can generalize to variable views: we train on 1, 2, and 3 views and generalize to 5 views. Finally, 3DFIRES can reconstruct when given LoFTR [37] estimated poses with known translation scale.

2. Related Works

We aim to produce a coherent 3D scene reconstruction given a single or a few images with wide baselines.

3D from Single Image. Predicting a complete 3D scene from a single image is inherently ambiguous. Recently different 3D representations have been proposed to reconstruct complete 3D scenes (including occluded surfaces) such as layered depth [35], voxels [3, 11, 19, 41], planes [16], point-clouds [9, 43], meshes [10, 12, 25], or implicit representation for objects [22, 26] and scenes [2, 4, 20, 21, 36]. While they have strong performance on single image, they do not necessarily produce coherent results when required to infer on multiple images of the same scene [21]. Our method can reconstruct hidden geometry from at least a single image using implicit representation from [20]. Instead of naively fusing point clouds from different images, we fuse features when predicting a multi-view consistent point cloud with few input images.

3D from dense views. Traditional multi-view 3D reconstruction methods can produce accurate and coherent point-clouds from pixel correspondences [33]. Classical methods in computer vision use approaches like Multi-view stereo (MVS) to construct only visible parts of the scene in all the images. There is a long line of work in trying to reconstruct scenes from video sequences [6, 34] where they reconstruct visible scenes and camera poses. Learning-based methods for MVS estimate geometry for scenes [18, 24, 38, 45] also require an input video to explicitly predict scene geometry. Instead of requiring high overlap inputs such as video frames, our method works on wide-baseline images.

3D from sparse view inputs. Our approach operates in a multi-view setting with a sparse set of views. We have a similar setting as wide-baseline reconstruction [27]. Associative3D [28] reconstructs the whole scene but requires voxelized scenes to train, our method works on non-watertight scene data. Prior work also explores planar representation [1, 17, 39] for coherent 3D surfaces in non-watertight scenes. They use feed-forward networks to predict visible 3D surfaces for each view and merge them using predicted correspondences. Our approach leverages an implicit representation that accommodates non-watertight data, enabling the reconstruction of both visible and occluded surfaces. We fuse deep features from multiple views to predict DRDF representation from Kulkarni *et al*. [20], producing a coherent reconstruction.

Novel view synthesis. NeRF [23] and its extensions [42, 46, 48] optimizes per-scene radiance fields for novel-view synthesis, this requires many views and test-time optimization. Due to its occupancy-based representation, extracting geometry often requires thresholding the density function, which leads to cloudy geometry with sparse input views. Our method directly predicts geometry from unseen images without the need for test-time optimization. PixelNerf [46] or SRT [32] can generalize to new scenes but their objectives optimize for photometric losses.

3. Method

Our goal is to predict an accurate and consistent 3D reconstruction from one or more sparsely spaced camera views and known poses. With one image, the method should predict all surfaces in the camera frustum, including visible and occluded regions. With more images, the method should predict the surfaces in the union of the frustum.

We tackle this problem with 3DFIRES, a simple and effective approach designed for this setting. We first discuss

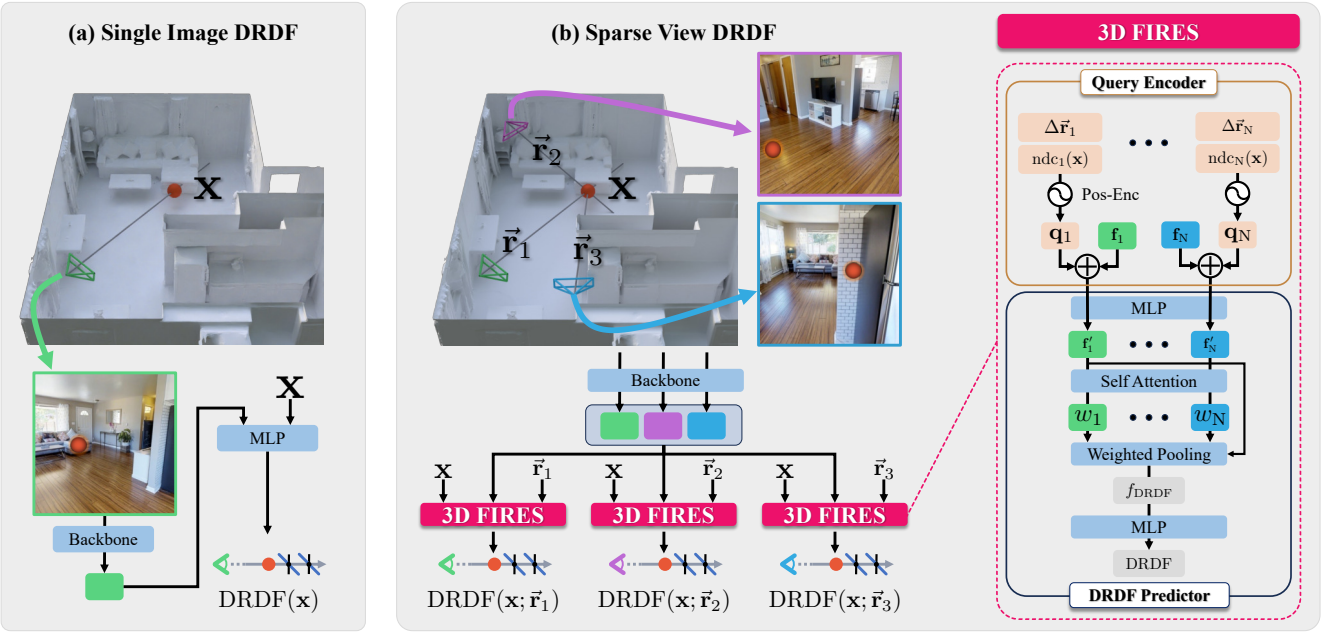
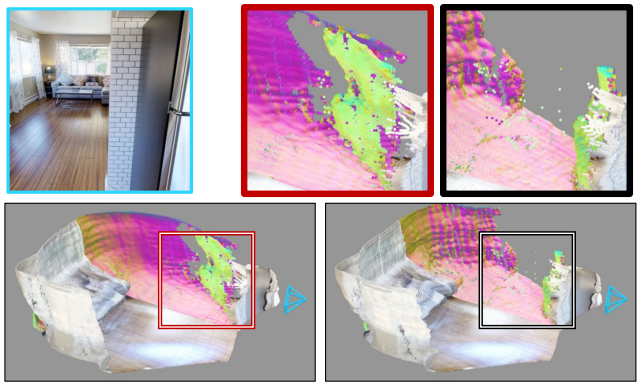


Figure 2. (a) Architecture for single view DRDF [20]. Given an image and a query pixel location, it predicts DRDF along the ray from the query pixel. (b) we extend (a) to work on sparse views. Middle: Given N images, a query point $\mathbf{x}$, and a query direction $\vec{\mathbf{r}}_q$, we aggregate features from multiple images and output DRDF along the query ray. Right: We show detailed network architecture of 3DFIRES which consists of a Query Encoder and a DRDF Predictor.



SV-DRDF 3DFIRES

Figure 3. **Predictions in the blue camera frustum.** Occluded surfaces are colored with surface normals. A single image to 3D method like DRDF [20] is unable to reconstruct the parts of the scene behind the wall with certainty and hence erroneously adds a full wall in front of the hallway (red box). 3DFIRES which fuses features from multiple views (Green and Purple camera in Fig. 2) predicts empty space for the entrance (black box).

tackling scene reconstruction in a single image case in §3.1 using the Directed Ray Distance Function (DRDF) [20] and scale this approach to multiple image views in §3.2. In §3.3, we show how we can operationalize our multi-view reconstruction goal with an attention-based model architecture.

3.1. Background Single View Reconstruction

We begin by revisiting the DRDF formulation for a single image reconstruction. Consider a single image $\mathcal{I}$, a single view implicit reconstruction method aims to produce the full 3D reconstruction for the scene from this image. At inference, when conditioned on image features, the method outputs a distance function for a pre-defined set of 3D points in the camera frustum. It then decodes this predicted distance function to a surface to recover the 3D geometry of the scene. For instance, if the predicted 3D distance function is an unsigned distance function [2], the points on the surface are with distances close to zero.

Kulkarni *et al.* [20] solve the single image 3D reconstruction with the DRDF function and show that using the DRDF outperforms the standard unsigned distance function. The DRDF is a ray-based distance function measuring the distance of a point $\mathbf{x}$ to the nearest intersection with a surface along a ray $\vec{\mathbf{r}}$. In [20], the ray on which distances are measured is the ray from the camera center $\mathbf{c}$ to $\mathbf{x}$.

Fig. 2 (a) shows the DRDF for one such ray. Now, any 3D point $\mathbf{x}$ can be represented as its distance towards the camera times a unit ray direction, or $z\vec{\mathbf{r}}$, where $z \in \mathbb{R}$ and $\vec{\mathbf{r}} = \text{norm}(\mathbf{x} - \mathbf{c})$ where $\text{norm}(\mathbf{p}) = \mathbf{p}/||\mathbf{p}||$. The DRDF, $d_{\text{DR}}(z\vec{\mathbf{r}})$, furthermore includes a sign that determines for the point the direction along the ray towards the nearest intersection (i.e., forwards or backwards). Therefore $(z + d_{\text{DR}}(z\vec{\mathbf{r}}))\vec{\mathbf{r}}$ corresponds to a point on the surface.

The DRDF can be used to create a system that infers single image 3D by pairing the distance function at a point $\mathbf{x}$ with pixel-aligned features. At inference time, as shown in Fig. 2 (a), given a point $\mathbf{x}$ in the camera frustum we can extract corresponding pixel-aligned image features using an

image backbone $\text{BB}[\pi(\mathbf{x})]$, and use an MLP to predict the DRDF value corresponding to the point $\mathbf{x}$ along the $\vec{\mathbf{r}}$. Since DRDF is a ray-based function, its value only depends on the intersections along the ray. For any ray corresponding to a pixel on the image, the prediction of DRDF for the point depends on the image features, and the location of the point on the ray. This parameterization allows DRDF to learn sharp 3D reconstructions of the scene from a single RGB image. At training time, we train a model to predict the DRDF by supervising it with the ground-truth DRDF values computed using the mesh geometry.

3.2. Extending DRDFs to Multiple Views

Now, with multiple views we have: N images $\{\mathcal{I}_i\}_{i=1}^{\text{N}}$, relative camera transforms $\{\pi_i\}_{i=1}^{\text{N}}$, and corresponding camera centers $\{\mathbf{c}_i\}_{i=1}^{\text{N}}$, our goal is to reconstruct the 3D of the full scene. While the task could perhaps be accomplished by simply predicting individual 3D for each camera, and assembling them together. Our insight is that if the camera frustums have considerable overlap, for overlapping regions we can achieve a better and more consistent reconstruction by allowing the network to reason about which camera provides the best view for each point. This can be achieved by allowing the network to fuse features across cameras for the points in *feature* space rather than by concatenating in *point* space. We propose to improve the feature quality of any point $\mathbf{x}$ by fusing the features from multiple cameras. Since we are now dealing with the multi-view settings, a multi-view DRDF formulation is necessary to allow us to predict the DRDF value along each of the query rays, $\vec{\mathbf{r}}_q$, originating from the respective camera centers.

In the case of multiple views, the image feature corresponding to a point $\mathbf{x}$ should be a fusion of features $\{\mathbf{f}_\theta[\pi_i(\mathbf{x})]\}_{i=1}^{\text{N}}$. The feature should support predicting the N DRDF values along all the camera directions as $\{d_{\text{DR}}(z_i\vec{\mathbf{r}}_i)\}_{i=1}^{\text{N}}$. The intuition of our key idea is that multiple-image views provide more information about the 3D scene and hence potentially better features. We can learn these better features by fusing features to predict a consistent output. This requires a novel architecture that attends to features and rays, $\{\vec{\mathbf{r}}_i\}_{i=1}^{\text{N}}$, originating from all the available image views. Under this formulation single view DRDF is a special case of our formulation where N is 1.

3.3. Network Architecture

Towards the goal of predicting DRDFs along multiple query rays $\vec{\mathbf{r}}_q \in \{\vec{\mathbf{r}}_i\}_{i=1}^{\text{N}}$, we present a simple and effective network 3DFIRES that accomplishes this task. 3DFIRES consists of three modules: The first module is a *Backbone Feature Extractor* that obtains pixel-aligned appearance features; by projecting the query point $\mathbf{x}$ onto the camera, we can obtain a per-point and per-camera appearance feature as in [20, 23, 31, 42, 46]. Since the appearance feature is per-image, the model must learn to aggregate information across cameras. This is done with our second component *Query Encoder* that provides geometric information for aggregating appearance features. Specifically, the query encoder uses the information about the relative positions of query point $\mathbf{x}$ and query direction $\vec{\mathbf{r}}_q$ w.r.t. cameras $\{\pi_i\}_{i=1}^{\text{N}}$. The final module is the *DRDF Predictor* that takes appearance and query features to produces a DRDF value along the query direction $\vec{\mathbf{r}}_q$ by incorporating the appearance features (evidence for geometry) and query encoder features (evidence that relates different features). Fig. 3 shows an example on how integrating information across multiple views leads to better prediction for occluded parts of the scene.

Backbone Feature Extractor. Our backbone features extractor $\text{BB}(\cdot)$ aims to create appearance features from an image. It accepts an image $\mathcal{I}_i \in \mathbb{R}^{H\times W\times 3}$ and produces a grid of D-dimensional features $\mathbf{F}_i \in \mathbb{R}^{H'\times W'\times D_{\text{img}}}$. We use a pre-trained depth estimating vision transformer [29]. Feature extraction for each image proceeds independently using the same network. With extracted per-camera backbone features, $\mathbf{f}_i$, for point $\mathbf{x}$ by interpolating features in $\{\mathbf{F}_i\}_{i=1}^{\text{N}}$ at the projection $\{\pi_i(\mathbf{x})\}_{i=1}^{\text{N}}$ correspondingly.

Query Encoder. Our query encoder $q(\cdot)$ aims to enable a predictor to decide how to aggregate information across images. As input, the encoder takes a query 3D point $\mathbf{x}$ and a query direction $\vec{\mathbf{r}}_q$. It additionally considers the backbone features, camera centers $\{\mathbf{c}_i\}_{i=1}^{\text{N}}$ and transforms $\{\pi_i\}_{i=1}^{\text{N}}$. Our query encoding is the concatenation of: (i) the relative viewing direction in camera i's space $\Delta\vec{\mathbf{r}}_i(\vec{\mathbf{r}}_q) = [\vec{\mathbf{r}}_q - \text{norm}(\mathbf{x}-\mathbf{c}_i), \vec{\mathbf{r}}_q \cdot \text{norm}(\mathbf{x}-\mathbf{c}_i)] \in \mathbb{R}^4$; and (ii) the normalized device coordinates (NDC), coordinates of point $\mathbf{x}$ in the camera frame $\text{ndc}_i(\mathbf{x}) \in \mathbb{R}^3$. Intuitively this query representation, $\mathbf{q}_i = \{\Delta\vec{\mathbf{r}}_i, \text{ndc}_i(\mathbf{x})\} \in \mathbb{R}^7$ enables reasoning such as: information about surfaces near $\mathbf{x}$ in direction $\vec{\mathbf{r}}_q$ is likely not visible in camera i due to either angle or distance, so this feature ought to be weighted low. The ray query vector is encoded in a positional encoding layer [40] with output dimension D_{query}.

DRDF Predictor. For a query ray and point tuple, $\{\vec{\mathbf{r}}_q, \mathbf{x}\}$, this model considers the image features $\{\mathbf{f}_i\}_{i=1}^{\text{N}}$, and query features $\{\mathbf{q}_i\}_{i=1}^{N}$ yielding a joint camera specific feature, $\{\mathbf{f}_i, \mathbf{q}_i\}_{i=1}^{\text{N}}$, of dimension $D_{\text{img}} + D_{\text{query}}$. Our self-attention attends over all these features to produce a weight w_i per feature. We aggregate the features using this weight to produce a fused feature for the point $\mathbf{x}$. We then use the fused feature to predict a DRDF value between $[-1, 1]$ with the help of an MLP. This is akin to selecting cameras that are likely to contain the geometry information about the ray point tuple and predicting the geometry information.

3.4. Training 3DFIRES

The effectiveness of 3DFIRES is improved by getting details right during training. One observation is that sampling

points near intersections gives improvements over uniform sampling because the scene-level space is predominantly empty. By increasing the density of sampled points near surface, the network can better learn the scene structure. We sample points along the ray as per a Gaussian distribution centered at the intersection. Prior work [42] involves applying ray attention which allows for samples along a ray to attend with each other before the final prediction. This has been shown to be effective. However, combining ray attention with Gaussian sampling during training enables the network to 'cheat'. Ray Attention exploits a train-time shortcut (query point density) to infer intersections. At inference as point density is uniform and this shortcut fails. Empirically we find Gaussian sampling alone to be more effective than ray attention.

3.5. Implementation Details

Training. Our image feature backbone is vision transformer [29] dpt_beit_large_384 pretrained by MiDaS [30]. We use ℓ_1 loss on log-space truncated DRDF [5, 20, 38]. During training, we randomly sample 1, 2, 3 views with 80 rays per image and 512 points along each ray. Our method is trained for 300K iteration on NVIDIA A100 GPU with batch size of 1. More details in supp.

Inference. Given N images, we extract backbone features for each image. We generate $n_{\text{ray}} = 128 \times 128$ query rays from each camera. Along each ray, we sample $n_{\text{pt}} = 256$ points that have uniformly spaced depth from 0 to 8m. In total, we get N $\times n_{\text{ray}} \times n_{\text{pt}}$ query pairs $\{\mathbf{x}, \vec{\mathbf{r}}_q\}$, which are fed to 3DFIRES in parallel to get DRDF value. We calculate positive-to-negative zero-crossings along each ray [20] to get a 3D point and aggregate the results.

4. Experiment

In this section, we present the experimental framework for 3DFIRES, our system designed to reconstruct full scene geometry from wide-baseline, sparse images. Considering the novelty of our problem, there is no prior work that does this exact setting. To address this, we curated a dataset and developed testing metrics specifically tailored to the problem's requirements. We conduct comprehensive evaluations of 3DFIRES using real scene images, comparing its performance against alternative methods in the field.

4.1. Dataset

Following [21], we use the dataset from the Gibson database [44], which contains real images of complex and diverse scenes such as multi-floor villas and expansive warehouses. The scale of the assets in the dataset presents challenging reconstruction problem, which is desirable for evaluating the ability to recover occluded surfaces. We use the images sampled by Omnidata [8] for a diverse set of camera poses from the Taskonomy [47] Medium subset, including 98/20/20 training/validation/test buildings. Since our multiview setting is different from the single-view setting of [21], the precise samples are different. Our setting is also similar to [17, 39] in that images have wide baselines (median 2.8m translation, 63.9° rotation), unlike methods using video frames [38] where images have high overlap. Our approach diverges from [17, 39] in also reconstructing occluded regions and using real (not rendered) images.

To curate our image sets, we use a sampling process like [17]. For a set of k images, after picking an image at random, each new image is selected to have at most 70% overlap with any existing image in the set, and at least 30% overlap with at least one other image in the set. The process balances diversity and coherence in the viewpoints. We crop images to a fixed field of view. We collect 3781 training sets among $\geq$ 10K images. We also sample 300 sets of 3-view images and 100 sets of 5-view images for evaluation from the held-out test scenes. See the supplementary for dataset generation details. The 3 view and 5 view test set contain considerable occluded 3D geometry (41.9% and 43.7% respectively).

4.2. Baselines

To the best of our knowledge, no prior work reconstructs occluded regions from sparse-view images at scene scale. We thus create strong baselines from existing methods that handle parts of our setting. Each method is the strongest in its line of work.

For instance, the visible surface upper-bound includes all methods that reconstruct visible surfaces from sparse views [17, 38, 39]. The DRDF method [20, 21] has been shown to be more effective for scene-level 3D reconstruction compared to many other implicit functions like density [46], occupancy [31], unsigned distance functions on scenes and rays [2]. MCC [43] is likewise SOTA for point cloud completion.

Depth Only [8, 29] Prior state-of-the-art works on sparse scene reconstruction [38, 39] predict visible surfaces from multiple views, but cannot recover hidden surfaces. To show the near-oracle reconstruction of visible surfaces, we use MiDaS [29] depth model trained on Omnidata [8] *with ground-truth scale and shift*. This baseline is an upper bound on the performance of methods like [1, 17, 38, 39].

Multiview Compressive Coding (MCC) [43] This method predicts occupancy probability from RGB-D partial pointclouds. MCC works on scene-level reconstructions including non-watertight meshes. We train MCC on the same training set as ours. This method requires depth as input and at inference we provide it with ground truth depth. Since MCC only works on a single point cloud, to produce predictions from multiple images, we infer each image independently and aggregate the predicted point cloud in point

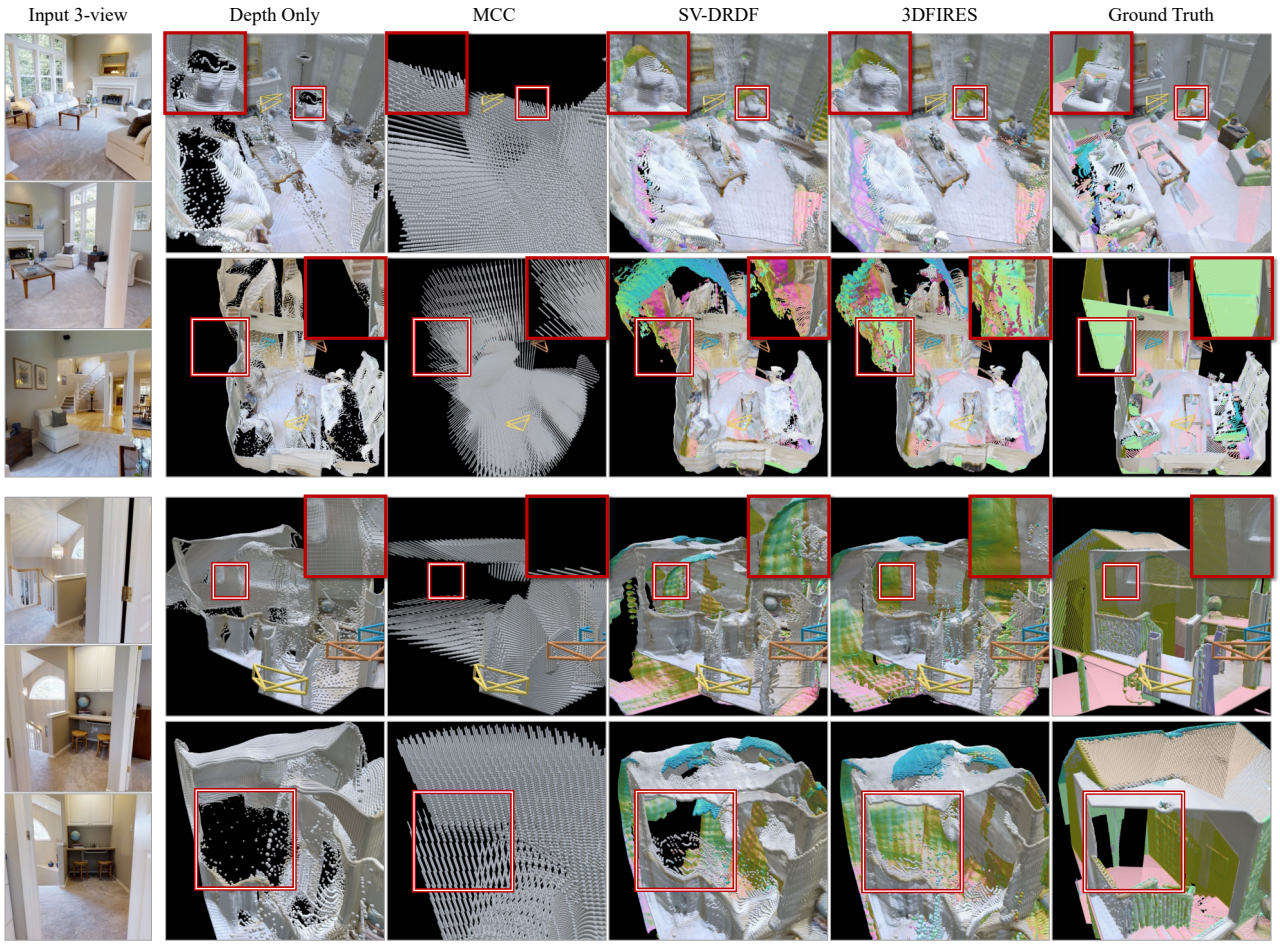


Figure 4. Comparison between different methods on held-out test scene. Occluded surfaces are colored with the computed surface normals. "Depth only" leaves holes with sparse input views, *e.g.* absent floors and walls. Occupancy-based method MCC [43] produces cloudy results, failing to get the details like pillow, tables. Concatenation of single view DRDF (SV-DRDF) [20] produces inconsistent results, *e.g.* missing wall in row 2, the double wall in row 3. Our method produces more consistent predictions across different views and also recovers the hidden surface, resulting in a complete mesh. We urge the reader to see results provided in the supplementary videos.

cloud space.

Single-view DRDF (SV-DRDF) [20] This method reconstructs both visible and hidden surfaces from a single input image. We use this baseline to show the benefit of our proposed multi-view feature aggregation. For a fair comparison, we upgrade the original backbone from ResNet34 [13] to the same BEiT [29] and use the same training strategy such as Gaussian sampling of points. Both improve results. Since this baseline only supports single image reconstruction, we produce predictions independently from each input image and aggregate all the point clouds.

4.3. Evaluation Metrics

We use two metrics to evaluate our system.

Scene F score. Following [20, 43], we compute the scene accuracy (fraction of predicted points within ρ of a ground truth point), completeness (fraction of ground truth points within ρ from a predicted point), and their F-score (F1). This gives an overall summary of scene-level reconstruction. We classify the scene into (1) visible: points that are visible from any one of the input views; and (2) hidden: points that are hidden from all of the input views. Due to the space limit, we only show F-score at $\rho = 0.2$. A full table with accuracy, completeness, F-score at different ρ is in the supp. Trends are the same across values of ρ and there is no significant accuracy/completeness imbalance for the baselines (MCC, SV-DRDF).

Multiview consistency. Only measuring the F-score does not measure the consistency of 3D reconstruction when generating results from multiple views. Doubled predictions of surfaces do not change the Scene F score results if they are within ρ. Prior work [17] used a detection-based method that penalized double surfaces on planar predictions, but their metric is not applicable since it requires pla-

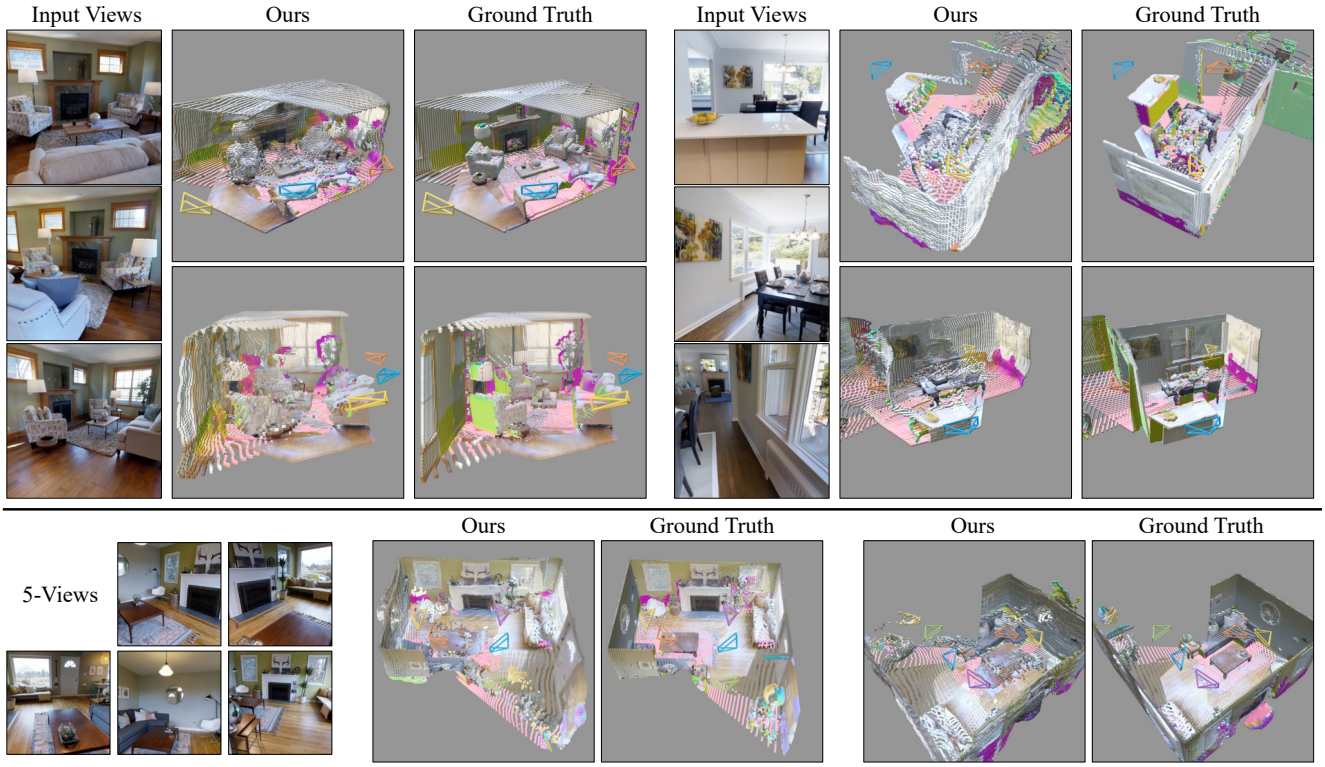


Figure 5. Qualitative results on held-out test scenes. Top row: Reconstruction from 3 images and compared with ground truth. Our method can reconstruct a complete scene structure within all the camera frustums, including the occluded surfaces. Bottom row: Predictions from 5 input images compared with ground truth. For the 2nd and 3rd examples, ceilings are removed to reveal the details of the scene.

nar instances. We require a metric that can measure the consistency of 3D reconstruction of points in individual frustums. Specifically, we would like to ensure that points $\mathbf{P}_i$ generated from all query rays originating from $\mathbf{c}_i$ of π_i are consistent with points, $\mathbf{P}_j$, generated from by ray queries from $\mathbf{c}_j$ of π_j at the intersection of frustums of both the cameras. For every point, $\mathbf{p} \in \mathbf{P}_j$ and within the field of view of camera i, we compute their minimum distance to points in $\mathbf{P}_i$. Our metric measures percent of points in the set $\mathbf{P}_j$ that have minimum distance within the threshold of ρ. We evaluate this metric bidirectionally to ensure complete results.

4.4. Results

Qualitative Results. Fig. 3 shows reconstruction from using query rays from the blue camera in Fig. 2. Occluded surfaces are colored with surface normals. DRDF [20] is unable to reconstruct the parts of the scene behind the wall with certainty and erroneously adds a full wall in front of the hallway. 3DFIRES fuses features from multiple images (Green and Purple camera in Fig. 2) accurately predicts the empty space.

Fig. 4 shows results unseen test scenes, and compares reconstruction of baselines. Red box crop show highlighted differences and provide a zoomed-in view for detailed examination. Depth only (MiDaS with ground truth scale and shift) reconstructs *only* visible regions this leaves holes such as the missing surfaces behind chairs in Row 1; and absent floor sections in Row 4. MCC [43] tends to produce cloudy volumes and misses details like pillows and tables. Single-view DRDF (SV-DRDF) produces occluded regions and sharp surfaces but lacks consistency when aggregating results from multiple views. This is noticeable in its inability to reconstruct the occluded wall in Row 2, the creation of a doubled ceiling in Row 3 due to occlusions. 3DFIRES, effectively merges observations from multiple images, resulting in sharp and accurate reconstructions of both visible and hidden surfaces. By fusing information across views in the feature space, our method overcomes the limitations of other approaches. This ensures comprehensive and consistent scene-level reconstruction from few sparse views.

In Fig. 5 we show additional alongside the ground truth. 3DFIRES successfully reconstructs large occluded areas, floors hidden by foreground objects (colored in pink), and unseen sides of objects such as the back of chairs in the first example and the kitchen islands in the second example. The reconstruction from multiple views demonstrates consistency and coherent surfaces in overlapping regions.

While our method is trained with up to three views, it seamlessly extends to five views. This adaptability stems from our architecture's inherent flexibility to the number of input views. With increasing views it predicts clean and coherent reconstructions within all the camera frustums.

Quantitative Results. We evaluate our method on sets of 1,

Table 1. Quantitative results on Scene F-score ($\rho = 0.2$) for Hidden points, Visible points, All points. For 3 and 5 views, we evaluate Consistency. Depth only: visible surface upperbound is separated to indicate it has oracle information. Despite accurate reconstructions on visible surfaces, these lines of work cannot recover hidden surfaces, causing low overall performance. With 1 view, 3DFIRES is comparable to single view DRDF. With more views, 3DFIRES outperforms all the other baselines in F-score. There is large improvement in consistency metric compared to single view DRDF, showing that aggregating features produces a more coherent reconstruction. Full tables showing accuracy and completeness are in the supplemental.

	1 view			3 views				5 views			
	Hidden ↑	Visible ↑	All ↑	Hidden ↑	Visible ↑	All ↑	Consistency ↑	Hidden ↑	Visible ↑	All ↑	Consistency ↑
Depth only	-	**85.31**	60.12	-	**87.84**	63.90	72.79	-	**91.29**	69.40	72.57
MCC	40.27	56.40	50.25	42.91	62.02	54.78	70.20	38.51	64.44	55.94	66.57
SV-DRDF	**53.36**	73.45	65.21	48.02	76.19	65.61	76.44	47.51	81.31	70.54	78.13
3DFIRES	53.34	74.29	**65.71**	**49.99**	76.74	**66.56**	**85.48**	**49.52**	81.74	**71.41**	**85.92**

Table 2. Ablation study on training strategies. GS: Gaussian sampling near intersection along the ray during training. Ray Attn: points along a query ray attend to each other.

	Hidden	Visible	All	Consistency
-GS	43.07	77.05	64.81	83.45
+Ray Attn. -GS	47.09	**77.60**	65.58	83.27
+Ray Attn. +GS	14.85	3.36	13.29	33.56
Ours	**50.20**	77.30	**66.46**	**85.45**

Table 3. Quantitative results on noisy camera poses generated by LoFTR, evaluated on 3 view cases at $\rho = 0.2$. 3DFIRES assumes accurate pixel-aligned features but still produces more consistent reconstructions compared to not aggregating features.

3-View	Hidden	Visible	All	Consistency
SV-DRDF	37.39	**62.71**	52.93	57.65
Ours	**38.85**	62.40	**53.19**	**65.71**

3, 5 views respectively, as detailed in Tab. 1. Our approach, designed for flexible input views, matches prior works in single-view scene reconstruction and achieves state-of-the-art results with multiple input views. In single-image cases, it is comparable to the single-view DRDF baseline.

For 3-view sets, our method outperforms MCC [43] or DRDF [21]. Although MiDaS with ground truth scale and shift demonstrates optimal visible surface reconstruction, it falls short in overall scene reconstruction because of no reconstruction on occluded surfaces. When evaluated on scene consistency, 3DFIRES shows a large absolute improvement of $> 9\%$, over the second-best baseline, showing 3DFIRES's ability to aggregate features across views to produce consistent results.

The trend persists with 5-view inputs, where our method has the highest F score and consistency. Our method is not trained on 5-views subset but still remains robust to more input views enhancing the reconstruction quality in both visible and hidden surface reconstructions.

4.5. Ablations and Analysis

Ablation study on training strategy. We conduct an ablation study (Tab. 2) to investigate the effectiveness of different training strategies for our method. Without Gaussian sampling or ray attention (-GS), the method has degraded performance (-7% in hidden F score). With ray attention only (+Ray Attn. -GS), the method is able to better reconstruct the hidden surface but is still worse than ours (-3%). With both ray attention and Gaussian sampling (+Ray Attn. +GS), the network finds shortcut during training and does not work during testing. With Gaussian sampling strategy, our method performs the best.

Robustness with noisy camera poses. Our method requires accurate camera poses to aggregate pixel-aligned features. This setting is challenging with sparse view data since camera estimation can be noisy. We test if the misalignment of image features caused by noisy camera projection matrices degrades our system. We use LoFTR [37] to estimate the camera rotation and translation angle and evaluate the reconstruction within all the camera frustums. Since LoFTR does not provide a translation scale, we use ground truth instead. Tab. 3 shows results on 3-view cases. Our method still has significantly higher consistency over single view DRDF baseline. We provide an analysis with synthetic Gaussian camera noise in the supplementary.

5. Conclusions

We present 3DFIRES, a scene-level 3D reconstruction method that requires only one or a few posed images of a scene. Our method takes in an arbitrary number of input views, fuses multi-view information in the features space and predicts DRDF given a 3D point and query direction. We train our method on a large-scale scene dataset and show its strong ability to reconstruct both visible and hidden surfaces coherently within all the camera frustums on challenging wide-baseline images. Currently, our methods requires pose input from off-the-shelf estimation methods, solving for 3D reconstruction and adapting the poses is a challenging next step and left to future work.

Acknowledgments. Thanks to Mohamed Banani, Richard Higgins, Ziyang Chen for their helpful feedback. Thanks to UM ARC for computing support. Toyota Research Institute provided funds to support this work.

References

[1] Samir Agarwala, Linyi Jin, Chris Rockwell, and David F Fouhey. Planeformers: From sparse view planes to 3d reconstruction. In *ECCV*, 2022. 1, 2, 5

[2] Julian Chibane, Aymen Mir, and Gerard Pons-Moll. Neural unsigned distance fields for implicit function learning. In *NeurIPS*, 2020. 2, 3, 5

[3] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV*, 2016. 2

[4] Manuel Dahnert, Ji Hou, Matthias Nießner, and Angela Dai. Panoptic 3d scene reconstruction from a single rgb image. *NeurIPS*, 2021. 2

[5] Angela Dai, Christian Diller, and Matthias Nießner. Sg-nn: Sparse generative neural networks for self-supervised scene completion of rgb-d scans. In *CVPR*, 2020. 5

[6] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *TPAMI*, 2007. 2

[7] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *CVPR*, 2022. 2

[8] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *ICCV*, 2021. 2, 5

[9] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, 2017. 2

[10] Justin Johnson Georgia Gkioxari, Nikhila Ravi. Learning 3d object shape and layout without 3d supervision. *CVPR*, 2022. 2

[11] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *ECCV*, 2016. 2

[12] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *ICCV*, 2019. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[14] Mariko Isogawa, Dorian Chan, Ye Yuan, Kris M. Kitani, and Matthew O'Toole. Efficient non-line-of-sight imaging from transient sinograms. In *ECCV*, 2020. 1

[15] Hamid Izadinia, Qi Shan, and Steven M. Seitz. Im2cad. In *CVPR*, 2017. 1

[16] Ziyu Jiang, Buyu Liu, Samuel Schulter, Zhangyang Wang, and Manmohan Chandraker. Peek-a-boo: Occlusion reasoning in indoor scenes with plane representations. In *CVPR*, 2020. 2

[17] Linyi Jin, Shengyi Qian, Andrew Owens, and David F Fouhey. Planar surface reconstruction from sparse views. In *ICCV*, 2021. 1, 2, 5, 6

[18] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. *NeurIPS*, 2017. 2

[19] Nilesh Kulkarni, Ishan Misra, Shubham Tulsiani, and Abhinav Gupta. 3d-relnet: Joint object and relational network for 3d prediction. In *ICCV*, 2019. 2

[20] Nilesh Kulkarni, Justin Johnson, and David F Fouhey. Directed ray distance functions for 3d scene reconstruction. In *ECCV*, 2022. 1, 2, 3, 4, 5, 6, 7

[21] Nilesh Kulkarni, Linyi Jin, Justin Johnson, and David F Fouhey. Learning to predict scene-level implicit 3d from posed rgbd data. In *CVPR*, 2023. 2, 5, 8

[22] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 2019. 2

[23] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2, 4

[24] Zak Murez, Tarrence Van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-end 3d scene reconstruction from posed images. In *ECCV*, 2020. 2

[25] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In *CVPR*, 2020. 2

[26] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 1, 2

[27] Philip Pritchett and Andrew Zisserman. Wide baseline stereo matching. In *ICCV*, 1998. 2

[28] Shengyi Qian, Linyi Jin, and David F Fouhey. Associative3d: Volumetric reconstruction from sparse views. In *ECCV*, 2020. 2

[29] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *ICCV*, 2021. 4, 5, 6

[30] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *TPAMI*, 2022. 5

[31] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *CVPR*, 2020. 4, 5

[32] Mehdi S. M. Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lucic, Daniel Duckworth, Alexey Dosovitskiy, Jakob Uszkoreit, Thomas Funkhouser, and Andrea Tagliasacchi. Scene Representation Transformer: Geometry-Free Novel View Synthesis Through Set-Latent Scene Representations. *CVPR*, 2022. 2

[33] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 2002. 2

[34] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 2

[35] Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski. Layered depth images. In *Siggraph*, 1998. 2

[36] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *NeurIPS*, 2020. 2

[37] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. *CVPR*, 2021. 2, 8

[38] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. Neuralrecon: Real-time coherent 3d reconstruction from monocular video. In *CVPR*, 2021. 2, 5

[39] Bin Tan, Nan Xue, Tianfu Wu, and Gui-Song Xia. Nope-sac: Neural one-plane ransac for sparse-view planar 3d reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 1, 2, 5

[40] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 2020. 2, 4

[41] Shubham Tulsiani, Saurabh Gupta, David F Fouhey, Alexei A Efros, and Jitendra Malik. Factoring shape, pose, and layout from the 2d image of a 3d scene. In *CVPR*, 2018. 1, 2

[42] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021. 2, 4, 5

[43] Chao-Yuan Wu, Justin Johnson, Jitendra Malik, Christoph Feichtenhofer, and Georgia Gkioxari. Multiview compressive coding for 3D reconstruction. *CVPR*, 2023. 1, 2, 5, 6, 7, 8

[44] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *CVPR*, 2018. 5

[45] Yiming Xie, Matheus Gadelha, Fengting Yang, Xiaowei Zhou, and Huaizu Jiang. Planarrecon: Real-time 3d plane detection and reconstruction from posed monocular videos. In *CVPR*, 2022. 2

[46] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *CVPR*, 2021. 2, 4, 5

[47] Amir R. Zamir, Alexander Sax, William Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, 2018. 5

[48] Zhizhuo Zhou and Shubham Tulsiani. Sparsefusion: Distilling view-conditioned diffusion for 3d reconstruction. In *CVPR*, 2023. 2