Addressing Low-Shot MVS by Detecting and Completing Planar Surfaces

Rajbir Kataria[1] Zhizhong Li[2] Joseph DeGol[3] Derek Hoiem[1]

[1]University of Illinois Urbana-Champaign
{rk2,dhoiem}@illinois.edu

[2]Amazon
zli@amazon.com

[3]Microsoft
jodegol@microsoft.com

Abstract

Multiview stereo (MVS) systems typically require at least three views to reconstruct each scene point. This requirement increases the burden of image captures and leads to incomplete reconstructions. Our main idea to address this ***low-shot MVS*** *problem is to detect planar surfaces in depth maps generated by any MVS system and complete these surfaces by reformulating the MVS depth prediction task to a simpler planar surface assignment problem. We use single and multi-view cues (when available) and employ the DeepLabv3 architecture to infer the extent of planar regions and accurately complete missing surfaces. We show that our approach reconstructs portions of surfaces viewed by only one image, yielding denser models than existing MVS systems.*

1. Introduction

Multiview stereo (MVS) involves producing dense reconstructions from a collection of overlapping images with known camera intrinsics and extrinsics. The prevalence of drones, 360 cameras, and camera phones has enabled widespread use of MVS algorithms to model buildings, bridges, and cities for inspection and maintenance [1–3]. Currently, MVS algorithms aim to reconstruct portions of the scene viewed by at least three images (or two minimally), and neighboring photos are recommended to overlap in viewed surfaces by 60-80%. In practice, this guideline is difficult to follow, especially when taking photos by hand, and resulting scene models are often incomplete, which can cause delay and additional cost in critical applications such as bridge inspection. We introduce the problem of ***low-shot MVS***, aiming to reconstruct portions of the scene observed by at least one image.

Incomplete regions are caused by (1) inaccurate and inconsistent depth estimates on textureless surfaces and (2) unreliable or rejected depth estimates on surfaces viewed by too few images. Current state-of-the-art traditional and deep learning approaches address the first problem explicitly [8, 29, 37, 38, 40], but still cannot reconstruct portions of scene surfaces viewed by only one image.

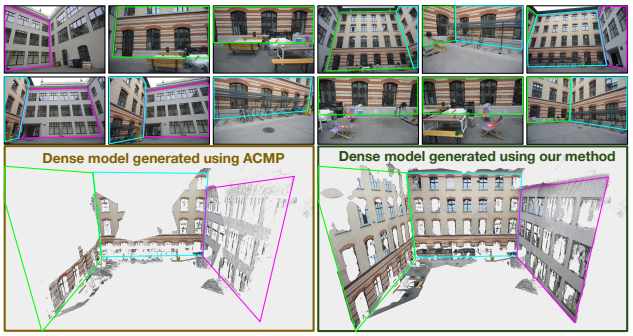

Figure 1. Insufficiently captured regions generate severely incomplete models in current MVS systems (e.g. [39]). In our approach, we detect planar regions in depth maps and infer their extent, resulting in more complete models.

Our approach focuses on addressing low-shot MVS, but also is capable of completing textureless surfaces. The key idea is that parametric surfaces can be fit to MVS-generated points in well-viewed portions of the scene, and the parametric surfaces can be completed by recognizing to which surface each pixel belongs. In this paper, we apply this idea to planar surfaces, which are most common for indoor and urban scenes, but the idea is applicable to other surfaces with parametric models, such as cones (e.g. power plant chimneys) and hyperbaloids (e.g. cooling towers). Our method is to detect planes and generate features based on available depth, number of views, and monocular surface normal prediction and RGB values. Based on these features, a segmentation model assigns each pixel in each image to the most likely plane (or no plane), and the corresponding depth estimates are fused with MVS depth estimates into a 3D model.

Our experiments show that our proposed system leads to more complete 3D models, while retaining good accuracy for scenes in the ETH3D [32] and Tanks and Temples [19] datasets, especially in the low-shot case. To show the effects of limited overlap in scenes, we evaluate MVS completeness and accuracy while varying the number of images. Furthermore, we also improve completeness for textureless surfaces (where we have a high degree of overlap).

In summary, the **contributions** of this paper are: (1) an

introduction to the low-shot MVS problem and its impact on completeness; (2) a system that addresses the low-shot MVS problem by detecting planar surfaces in reconstructed portions of the scene and inferring their extents to complete regions with insufficient views; and (3) experiments that validate our approach by demonstrating improved F1 scores for the low-shot and full-shot cases over COLMAP [31] and ACMP [39] systems on the ETH3D and Tanks and Temples datasets.

2. Related Work

Shape detection: The detection of primitive shapes is a well studied problem in computer vision [14, 17, 21, 30]. RANSAC [14], in particular, is robust to outliers and has proven to be effective in many applications. Schnabel et al. [30] developed a version called Efficient RANSAC that detects primitive shapes with a high probability in point clouds. Sampling minimal sets naively in the presence of millions of points is infeasible, and they address this issue by employing a localized sampling strategy and scoring scheme with a connectivity measure to limit the number of minimal sets to a smaller subset thereby increasing the probability of detecting shapes. Dimitrov et al. [11], alternatively, use a user-defined radius to compute multi-scale features that determine roughness of surfaces. They use a clustering approach to initialize and grow segments, ensuring that each point in the segment has similar surface roughness. We use a modified version of Efficient RANSAC [30] to detect planar regions, primarily due to its efficacy and simplicity. Although we made some small tweaks to the base algorithm, we claim no novelty in this regard.

Segmentation networks: An essential component of our approach is to accurately determine full extents of detected planes. To accomplish this task, we formulate extent prediction as a binary segmentation problem (whether or not a pixel lies on a detected plane). We limited our review of existing segmentation literature to approaches adept at capturing long-range dependencies because the boundaries of plane extents can appear far (in terms of pixels) compared to the pixels that detect the planar surfaces. Recently, vision transformers [12] have been used in semantic segmentation networks [25, 26, 34, 43] where they have demonstrated the ability to capture long-range dependencies. However, these networks require large training sets, which can be difficult to acquire for 3D modeling. Alternatively, convolutional architectures use atrous (dilated) convolutions to increase the filter field of view and capture long-range dependencies. Unlike transformer-based networks, these architectures do not require vast amounts of data for training. We adapt the DeepLabv3 [7] segmentation architecture for our approach, as it is a state-of-the-art network that uses atrous convolution.

Plane-informed Multi-view stereo: The use of planar surfaces in MVS systems has been around for over a decade. Earlier approaches [15, 16, 33, 35] explicitly detect planar surfaces to complete textureless surfaces in architectural scenes. Furukawa et al. [15], under the Manhattan-world assumption, detect planar surfaces in dominant directions using histogram binning of normals. They then recover the depths for textureless surfaces by formulating the problem as an MRF, which they solve using graph cuts. Sinha et al. [33] detect salient planes using the sparse point cloud from structure from motion (SfM) and vanishing point cues. They formulate the depth recovery problem as a multi-label MRF, which assigns each pixel to a candidate plane, and solve it using graph cuts. Gallup et al. [16] present a technique that detects and segments piecewise planar regions using image depth maps and segment them using a planar versus non-planar classifier. They ensure plane consistency across overlapping views by linking planes using common reconstructed points.

More recently, PatchMatch-based methods [31, 42] have gained popularity due to their efficiency and scalability in reconstructing large scenes. In these methods, depths and normal maps are estimated by employing PatchMatch [6] through an iterative series of search and propagation steps. Romanoni et al. [29] augment the set of hypothesis for PatchMatch by modelling textureless regions as piecewise planar surfaces. They accomplish this by segmenting the image into superpixels and propagate the hypotheses from superpixels with good inlier ratios to their respective neighbors. Xu et al. [39] propose a planar prior assisted multi-step PatchMatch framework in which they triangulate points using sparse correspondences to produce planar models. The planar models are used alongside photometric consistencies in a probabilistic graphical model to derive matching costs.

Monocular depth estimation: Monocular depth estimation approaches have shown promise in completing surfaces viewed just once. Methods such as [22–24] use plane priors to reconstruct surfaces from a single RGB image. While these approaches address the low-shot problem, their accuracy is significantly lower than that of traditional methods [31, 39].

Piecewise planar reconstructions: Planar reconstruction methods such as [5, 9, 27, 28, 36] use a sequence of images and automatically recover the camera poses along with a piecewise planar reconstruction. These systems demonstrate the benefits of detecting and estimating planar surfaces but require sequence information and do not address the low-shot problem.

Sparse view setting: Few methods address the sparsity of images for reconstruction. The method proposed in [41] improves reconstructions for different levels of sampling for scenes in the DTU dataset [4]. Their experiments under the extreme sparsity setting do not imply a low-shot setting as each region can still be viewed more than two times. The method proposed in [18] performs planar surface reconstruction from two views with unknown camera poses, but does not undertake the problem of single-view planar surface reconstruction.

All the plane-informed MVS methods exploit the pla-

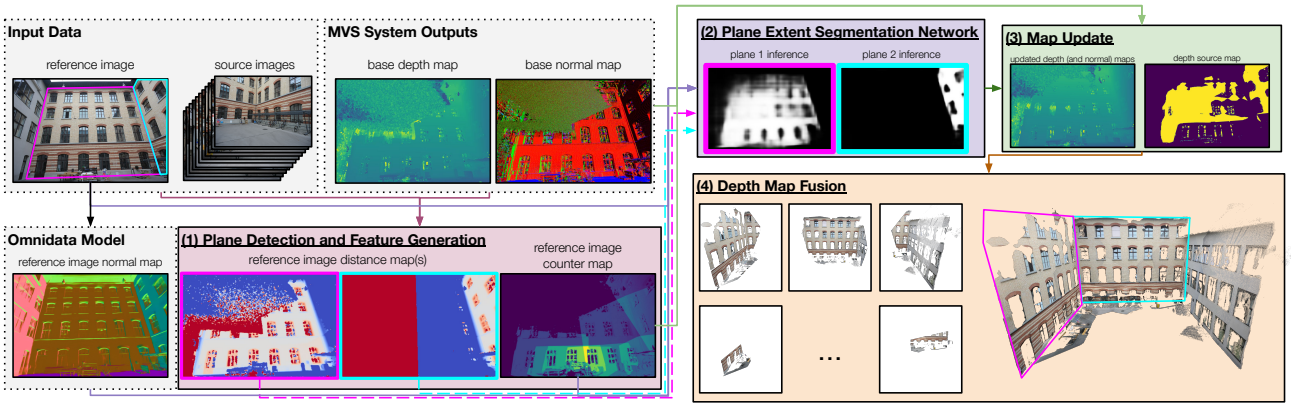

Figure 2. Our system consists of the *four* highlighted components. The inputs to our system are images, camera intrinsics and extrinsics, depth and normal maps from an existing MVS system, and complete normal maps using single-view prediction from the Omnidata model [13]. The (1) **plane detection and feature generation** component detects planar surfaces in the depth maps and generates corresponding counter maps and distance maps. The (2) **plane extent segmentation network** component infers plane extents for each detected plane. The (3) **map update** component uses the plane extent inferences to update the depth and normal maps. The (4) **depth map fusion** component fuses the updated depth maps.

narity of surfaces to improve the completeness of scenes, but their benefits are restricted to textureless (or low-texture) surfaces. These approaches were not designed for the low-shot case and thus cannot reconstruct regions with just one view due their reliance on the minimum number of views requirement. The monocular depth estimation methods do address the low-shot case, but cannot take geometry of the scene into consideration and result in models with significantly lower accuracy. In contrast, our method improves completeness by addressing the problem of insufficiently viewed surfaces and yield highly accurate reconstructions. In our experiments, we show significant improvements over ACMP [39], which is a state-of-the-art method for challenging benchmarks.

3. Method

In our system, we detect planar surfaces in depth maps estimated from any MVS system, and generate features such as normal, counter, and distance maps, which are input into our plane extent segmentation network (along with the images) to determine extents of the detected planes. Then, we update base depth (and normal) maps (from the MVS system) using the extent inferences, and perform depth map fusion. An overview of our system is illustrated in Figure 2.

3.1. Plane Detection

To detect planar regions in depth maps, we employ Efficient RANSAC [30] due to its efficacy and simplicity. This approach uses a localized sampling strategy and begins by drawing the first sample, p_1, from all points. The remaining samples, p_2 to p_k, are drawn from the points that are within a given radius around p_1 due to the higher likelihood of them belonging to the same shape. Once sufficient points are sampled, the model parameters are estimated and its score is computed. The score of the model consists of inlier points that are within an ϵ-band around the shape and whose normals are sufficiently close to the normal of the shape. To ensure that all inliers belong to the same shape, the largest connected component of inliers is used. Lastly, a refitting step is executed with relaxed constraints to remove unnecessary clutter from the point cloud.

3.2. Feature Generation

Once planar surfaces are detected, we generate **normal** and **counter maps** for each image, and **distance maps** for each detected plane in the image. The goal of normal maps is to help disambiguate surfaces with similar appearances. We generate the normal maps using the Omnidata [13] surface normal estimator which is trained with large datasets and informs our network that is trained on less data.

Counter maps indicate the number of total images that are depth consistent with a given pixel in the depth map. As indicated in [20], counter maps are indicative of confidence of the estimated depth (because higher values gives you more confidence). Since normal and counter maps are independent of planar surfaces, they can be generated in parallel to plane detection. An example of a counter map is shown in the first highlighted component in Figure 2. The total number of views for a region increase as the color changes from the dark blue (one view) to yellow (5+ views).

Distance maps are dependent on the planar surfaces detected in the depth map, and one is generated for each detected plane. A distance map stores the depth differences between the depth map and the depth of the detected plane at each pixel. Distance maps facilitate our plane extent segmentation network to learn the appearance of the detected plane because small depth differences at pixels signify the presence of the detected plane. An example of a distance map is shown in the first highlighted component in Figure 2. Blue colored regions appear in front of the detected plane, red colored regions appear behind the detected plane, and white colored regions appear very close to the plane.

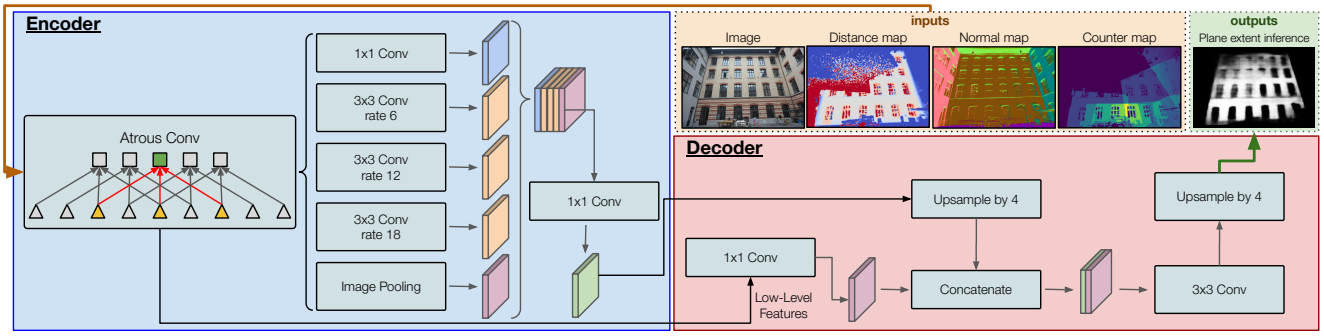

Figure 3. Our plane extent segmentation architecture is based on DeepLabv3 [7], which uses atrous convolutions thus enabling larger field of views for larger context. The inputs to our network are an image, its counter and normal map, and a distance map for a detected plane. The output is the plane extent inference which represents whether or not a pixel lies on the detected plane.

3.3. Plane Extent Segmentation Network

Our plane extent segmentation network is based on the DeepLabv3 [7] architecture which employs atrous convolutions to capture long range context. We modify the first layer to accommodate our 8-channel input which is the concatenation of an image, its normal and counter map, and a distance map corresponding to a single detected plane. We also modify the last layer for a 1-channel output which signifies the probability a pixel is on the detected plane. Semantic segmentation is commonly applied to predefined classes, but in our case, we do not treat planes as different classes (i.e. no plane specific parameters are learned). Instead, our network deduces the appearance of the plane of interest from pixels that are close to that plane, according to distance maps. Figure 3 shows the architecture of our network.

3.3.1 Training

To train our model, we employ the ScanNet [10] RGB-D video dataset that consists of over 1500 indoor scenes annotated with 3D camera poses and noise-free depth images captured via a commodity RGB-D sensor. To generate a dataset suitable for training our network, we generate examples using approximately 250 scenes from the training set and sample every 20th image in each scene to limit the redundancy between examples.

For each training image, we compute depth and counter maps using ACMP. In training, plane fitting is performed on ground truth depth maps. Plane segmentation labels are defined based on the distance of ground truth depth to each plane. We label a pixel as inlier if the depth difference is less than *5cm* and an outlier if the difference is greater than *10cm*. The remaining pixels are ignored in the loss calculation.

Since our plane extent segmentation network is a binary classifier (it infers whether a point lies on the plane or not), we use the binary cross-entropy loss function for training:

$$L_n = -m_n \left(\frac{\Sigma_{OB}}{\Sigma_{IB}+\Sigma_{OB}} y_n \log(\hat{y_n}) + \frac{\Sigma_{IB}}{\Sigma_{IB}+\Sigma_{OB}} (1-y_n) \log(1-\hat{y_n}) \right). \quad (1)$$

L_n is the loss for a single pixel, $\hat{y_n}$ is the inference prediction, and y_n is the ground-truth label (one for inliers and zero for outliers). m_n is zero for ignored pixels, and one otherwise. Σ_{IB} and Σ_{OB} are the number of inliers and outliers in the batch, respectively, and are used to account for class imbalance.

3.4. Plane Extent Inference Integration

For each pixel, the original depth value is replaced by the most likely predicted plane if the confidence is greater than a threshold. The threshold depends on how many images view the point (as computed in the counter map), since MVS depth is more reliable and precise with more views. In tests on ETH3D, the threshold is 0.5 for 1 view (reference view only), 0.9 for two views, and 0.99 for three views. The original MVS depth value is always retained if four or more views agree. See our supplemental material for details. We use a standard fusion algorithm, except that all 3D points produced by plane assignments in the updated depth map are retained, enabling reconstruction of portions of planar surfaces observed by only one image.

4. Experiments

In Section 4.1, we evaluate the efficacy of our approach for depth estimation of low-shot (viewed by less than *three* images) and textureless (viewed by *three* or more images) regions. We evaluate our method for the low-shot MVS problem in Section 4.2 while showing improvements on two different MVS systems. Furthermore, we evaluate our method on the test sets of the ETH3D and Tanks and Temples datasets in Section 4.3 to show that the benefit of our approach generalizes to unseen scenes. Lastly, we perform an ablation study in Section 4.4 to show the impact of each feature used by our plane extent segmentation network.

Datasets: We perform evaluations on two datasets: the ETH3D dataset [32] and the Tanks and Temples dataset [19]. The ETH3D dataset contains a total of 13 indoor and outdoor scenes mainly featuring urban buildings composed of planar surfaces while the Tanks and Temples dataset contains 6 indoor and outdoor scenes composed of more natural and curved architectural surfaces. We also use approximately 250 scenes from the ScanNet dataset [10] to

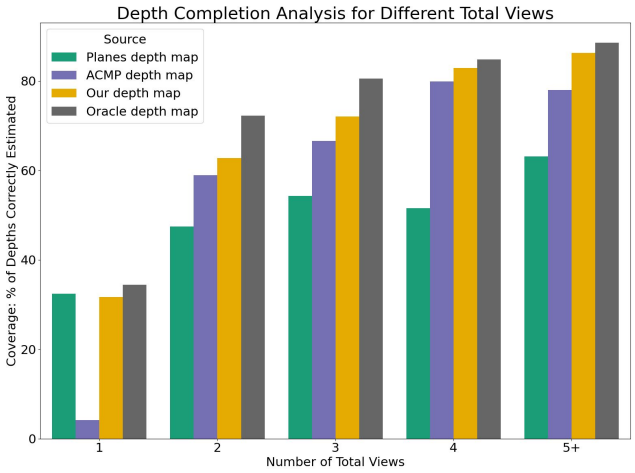

Figure 4. This figure depicts the completeness coverage, which signifies the percentage of depths correctly estimated within a distance of *5cm*, for different depth map sources organized by the number of total image views. In each group, the **first** bar is for the ***planes depth map***, which consists solely of depths estimated from inferred plane extents; the **second** bar is for ***ACMP depth map***, which is the output of the MVS system; the **third** bar is for ***our depth map***, in which we updated the ACMP depth map with plane inferred depths; and the **fourth** bar is for the ***oracle depth map***, in which we select at each pixel whichever of Planes or ACMP depth is closer to the ground truth. Our depth map yields considerably higher coverage for regions viewed only once, while still outperforming ACMP for two or more total views.

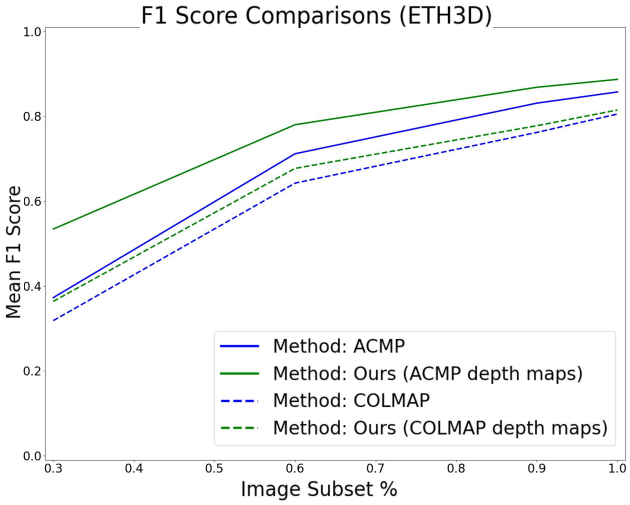

Figure 5. Mean F1 scores (y-axis) for the ETH3D dataset as we vary the percentage of images we use from the original scene (x-axis). The **blue** lines show the results for the base systems; a *solid* line for ACMP and a *dashed* line for COLMAP. The **green** lines show the results for our system and the line style indicates the source of the depth map we use for plane detection (and as base maps); a *solid* line indicates that we use ACMP depth maps and a *dashed* indicates that we use COLMAP depth maps. Our approach results in higher F1 scores and has the largest impact when the image overlap is low (low-shot case).

generate our training dataset for our plane extent segmentation network. This dataset is composed of indoor rooms with varying degrees of clutter and textureless surfaces.

Measures: MVS systems are evaluated using three metrics: (1) Accuracy (precision); (2) Completeness (recall); and (3) F1 score. These metrics are evaluated over a series of pre-specified distance thresholds, typically ranging from 0.01 to 0.05. Accuracy measures the percentage of estimated 3D points that are within a pre-specified distance of any point in the ground-truth point cloud. Conversely, completeness measures the percentage of ground-truth 3D points that are within a pre-specified distance of any point in the estimated point cloud. The F1 score is the harmonic mean of accuracy and completeness and yields a single value that is often used to compare MVS systems.

Base MVS Systems: Our method can operate on depth and normal maps generated from any MVS system. We choose the COLMAP (v3.7) [31] MVS system because it is commonly used and the ACMP [39] MVS system because it provides highly accurate depth and normal maps and is considered a state-of-the-art system for these challenging benchmarks. Except where otherwise noted, we use default parameters. We perform our experiments and ablation study using depth and normal maps from ACMP, and use COLMAP to demonstrate general applicability.

Implementation details: We use existing off-the-shelf implementations for the COLMAP and ACMP MVS systems. We also use off-the-shelf implementations (and parameters) for algorithms and models we employ in our system, such as Efficient RANSAC [30] for planar surface detection and the Omnidata [13] model for surface normal estimation. We use the same parameters for all datasets, except the confidence thresholds C_O used to choose between plane or MVS depth values, which is set differently for ETH3D [32] and Tanks and Temples [19], but not tuned for test sets. The base confidence parameter, C_O, represents the confidence in the base system map (depth or normal) as a function of total image views. This parameter dictates how the base depth (and normal) and planes depth (and normal) maps are merged. For the ETH3D [32] dataset, we set C_O values to *0.50, 0.90, 0.99, and 1.00* for total image views of *1, 2, 3, and 4+*, and for the Tanks and Temples [19] dataset, we set C_O values to *0.75, 0.95, and 1.00* for total image views of *1, 2, and 3+*. The chosen C_O values for Tanks and Temples [19] are stricter due to the stricter threshold of *1cm* used in Tanks and Temples [19] evaluation. See supplemental for details on training our plane extent segmentation network.

4.1. Completeness Coverage Analysis

Figure 4 shows our evaluation of how well our approach completes depth maps for regions with different numbers of image views. To measure completeness at a depth-map level, we require dense ground-truth depth maps and their corresponding counter maps. To obtain this data, we perform surface reconstruction on the laser-scanned point cloud and generate dense ground-truth depth maps using ray-mesh intersections. Lastly, we use the dense ground-

	*Threshold = **2 cm***						*Threshold = **5 cm***					
	ACMP [39]			OURS			ACMP [39]			OURS		
	A	C	F1	A	C	F1	A	C	F1	A	C	F1
Botanical garden	94.91	74.06	83.20	94.28	78.45	**85.64**	98.40	84.98	91.20	98.00	90.42	**94.06**
Boulders	84.50	52.37	64.67	86.11	57.59	**69.02**	93.04	67.14	78.00	94.23	71.73	**81.46**
Bridge	89.49	78.43	**83.60**	85.91	80.33	83.03	94.54	89.03	**91.70**	91.48	91.06	91.27
Door	93.98	85.88	89.75	93.60	87.39	**90.39**	98.05	93.18	95.55	98.17	94.07	**96.07**
Exhibition hall	79.68	62.03	69.75	80.16	63.86	**71.09**	90.45	80.26	85.05	90.72	82.00	**86.14**
Lecture room	92.09	58.91	71.85	89.13	68.76	**77.63**	96.31	72.87	82.97	95.30	83.29	**88.89**
Living room	93.89	81.21	87.09	93.44	89.62	**91.49**	96.76	88.15	92.25	96.47	95.08	**95.77**
Lounge	83.80	20.38	32.78	83.82	40.50	**54.62**	92.91	38.55	54.49	87.94	56.42	**68.73**
Observatory	93.10	91.15	**92.11**	90.98	91.72	91.35	98.72	96.26	**97.48**	96.98	97.55	97.26
Old computer	82.67	60.69	69.99	83.86	69.00	**75.71**	91.19	76.05	82.93	91.41	83.70	**87.39**
Statue	98.04	68.18	80.43	95.30	74.25	**83.47**	99.42	78.05	87.45	97.61	83.42	**89.96**
Terrace 2	93.64	83.16	88.09	93.60	89.58	**91.54**	98.32	88.02	92.89	98.30	93.49	**95.83**
Mean	89.98	68.04	73.11	89.18	74.25	**80.41**	95.68	79.38	86.00	94.72	85.18	**89.40**

Table 1. Results for the ETH3D [32] high-resolution multi-view stereo test set. The heading *"A"* represents accuracy; *"C"* represents completeness; and *"F"* represents F1 scores. **Bolded** results indicate the highest F1 score for a given scene (and threshold).

	*Sampling Rate = **1 image every 10 secs***						*Sampling Rate = **1 image every sec***					
	ACMP [39]			OURS			ACMP [39]			OURS		
	A	C	F1	A	C	F1	A	C	F1	A	C	F1
Auditorium	48.63	2.70	5.11	54.77	7.73	**7.89**	38.22	22.92	28.66	34.58	26.01	**29.69**
Ballroom	51.42	10.66	17.66	31.69	17.17	**22.28**	37.76	60.36	46.46	38.03	67.41	**48.62**
Courtroom	54.40	4.12	7.66	21.34	6.32	**9.75**	43.49	35.88	**39.32**	37.27	41.20	39.14
Museum	66.64	7.06	12.77	34.13	12.39	**18.18**	47.16	60.45	**52.98**	41.45	66.13	50.96
Palace	37.90	3.19	**5.88**	15.32	3.17	5.25	30.84	24.48	**27.30**	22.74	26.76	24.59
Temple	42.87	3.38	**6.26**	12.74	3.10	4.98	41.15	37.98	**39.50**	32.60	42.20	36.78
Mean	50.31	5.18	9.22	28.33	7.73	**11.39**	39.77	40.35	**39.04**	34.44	44.95	38.30

Table 2. Results for the Tanks and Temples [19] advanced set. The heading *"A"* represents accuracy; *"C"* represents completeness; and *"F"* represents F1 scores. **Bolded** results indicate the highest F1 score for a given scene (and interval).

truth depth maps to generate ***ground-truth counter maps*** using the procedure outlined in Section 3.2. Given the ground-truth data, we calculate distances between the depth maps from each source and ground-truth depth maps. Then, we threshold (*5cm*) the distances and calculate the completeness coverage for each depth source.

We compare four depth sources: (1) The **planes depth map** (green bar); (2) The **ACMP depth map** (purple bar); (3) **Our depth map** (gold bar); and (4) The **oracle depth map** (gray bar). The planes depth map consists solely of depth estimates from inferred plane extents and shows how much of the scene our inferences alone can complete. The ACMP depth map is the output of the MVS system and is our baseline. Our depth map is the result of updating the ACMP depth map with plane inferred depths, as explained in Section 3.4. The oracle depth map is similar to our depth map, except the depth (for each pixel) is selected from the source closest to the ground-truth. The oracle depth map is the best-case scenario for completeness coverage and shows how effective we are at combining inferred depths with the ACMP depth map. For this evaluation, we use 30% of images from three scenes (*courtyard*, *delivery area*, and *electro*) of the ETH3D training set due to the availability of laser-scanned point clouds that we treat as ground-truth.

Plane inferred depths significantly improve coverage for regions viewed just once. Figure 4 shows that our depth maps (gold bar) result in significantly higher completeness coverage than ACMP depth maps (purple bar) for regions viewed just once. As expected, ACMP depth maps are erroneous for these regions because PatchMatch based methods [31, 39] require at least two image views for depth estimation. Our depth maps better estimate depths for these regions because our plane extent segmentation network learns the appearance of a planar surface where total number of image views is greater than one, and subsequently infers a more complete extent for regions viewed just once.

Plane inferred depths contribute to coverage even for regions viewed more than once. Figure 4 shows that our depth maps (gold bar), which contain plane inferred depths, result in higher completeness coverage for regions that have two or more views. Despite having a sufficient number of views for a region, depth estimation can still be erroneous (or missing) primarily due to lack of texture.

4.2. Low-shot MVS Results

We generate smaller subsets of the ETH3D training set in order to reduce the overlap between images, thereby simulating the low-shot case (see supplemental material for details on subset generation). We assume images have suffi-

cient overlap to be correctly registered. Figure 5 shows how our approach compares to COLMAP [31] and ACMP [39] MVS systems as we vary the percentage of images used in each scene on the x-axis and plot the resulting mean F1 scores on the y-axis.

Our approach results in higher F1 scores across two MVS systems: As shown in Figure 5, our approach, which uses depth (and normal) maps of a base MVS system, yields higher F1 scores on the ETH3D [32] dataset over the two respective base systems [31, 39]. This demonstrates that our approach is agnostic to the base MVS system.

Our approach has the largest impact for the low-shot case: Our results demonstrate that for the low-shot case, which is analogous to low image overlap resulting from low image subset % (x-axis in Figure 5), the difference between F1 scores from our approach and the corresponding base system increases (results are more prominent for ACMP, although we see a similar trend for COLMAP). The increasing difference can be attributed by the completion of larger missing planar regions by our system as long as sufficient points exist for detection of the planar region. Our approach also outperforms both ACMP and COLMAP when all images from the scenes are used.

4.3. ETH3D and Tanks and Temples Test Results

Our approach outperforms ACMP on the complete image test set of the ETH3D dataset. Table 1 shows the test set results for the ETH3D [32] dataset for thresholds of *2cm* and *5cm*. Our approach yields a higher F1 score compared to ACMP [39] for 10/12 scenes (for both thresholds) and results in an overall mean improvements of **10.0%** and **4.0%** corresponding to the thresholds of *2cm* and *5cm*, respectively. For the two scenes where ACMP [39] yields a higher F1 score, we generate competitive results as our F1 scores are within 1% of ACMP [39] F1 scores for both thresholds.

Our approach outperforms ACMP on the Tanks and Temples dataset for the low-shot case, while being competitive when all images are used. Table 2 shows the advanced test set results for the Tanks and Temples [19] dataset. In addition to evaluating our approach on the provided image set, which is generated by extracting an image every second from the video, we evaluate our approach for the low-shot case by extracting an image every 10 seconds thereby reducing the image set and image overlap.

For the low-shot case, our approach yields higher F1 scores for 4/6 scenes when compared to ACMP, resulting in an average F1 score improvement of **23.5%**. For the provided image set (full-shot case), our method slightly underperforms ACMP, resulting in an average F1 score decrease of **1.9%**, mainly due to the strict *1cm* threshold used in the dataset evaluation and to architectural features being frequently curved. The curved surfaces, which often appear as part of large dome-like structures, result in multiple planes being detected, and the inferred planar extents yield insufficiently accurate depth values, decreasing precision.

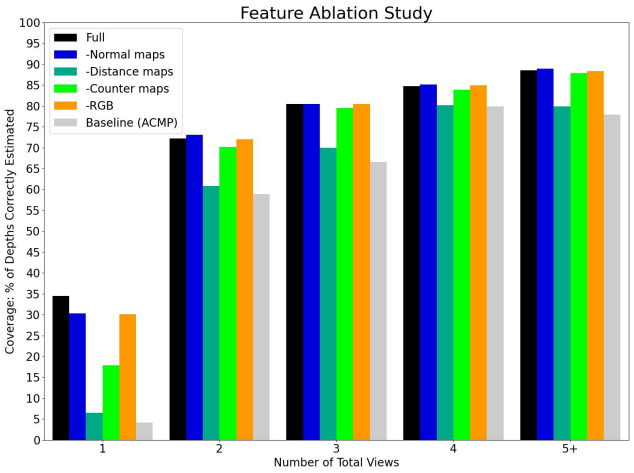

Figure 6. This figure depicts the impact of each feature on completeness coverage (percentage of depths correctly estimated within a distance of *5cm*, grouped by the number of total image views). In each group, we plot the completion coverage for *oracle depth maps* that uses inferred plane extents from networks trained on (1) All features ("Full"); (2) All features except normal maps ("-Normal maps"); (3) All features except distance maps ("-Distance maps"); (4) All features except counter maps ("-Counter maps"); and (5) All features except RGB images ("-RGB"). We also plot the completion coverage of ACMP [39] depth maps to demonstrate the relative impact of each feature. Distance maps have the biggest impact across all views while counter maps, normal maps, and RGB images are primarily helpful for regions viewed only once.

4.4. Ablation Study

In Figure 6, we display results of evaluating the impact of each feature on completeness coverage, which is the percentage of depths correctly estimated within a distance of *5 cm*. For this evaluation, we use the data from Section 4.1 and plot completeness coverage of oracle depth maps (in which we select depths from the source closest to the ground-truth). In using oracle depth maps, we're able to isolate the impact of the network features from the map update step, which merges depth maps (ACMP and planes depth maps) based on confidences.

We compare our final model (*"Full"*), which is trained using all features, to models trained (a) without normal maps (*"-Normal maps"*); (b) without distance maps (*"-Distance maps"*); (c) without counter maps (*"-Counter maps"*); and (d) without RGB images (*"-RGB"*). We also depict the completeness coverage of ACMP [39] depth maps to demonstrate the relative impact of each feature.

Distance maps have the biggest impact. Distance maps store the depth differences between the depth map and the depths of the detected plane and signify the region of interest for our segmentation network. Distance maps are the only indication of which pixels in the image are near the plane according to MVS estimates, without which the appearance of the plane is unspecified. As shown in Figure 6, the completeness coverage of the model without distance maps is similar to that of the depth maps attained from

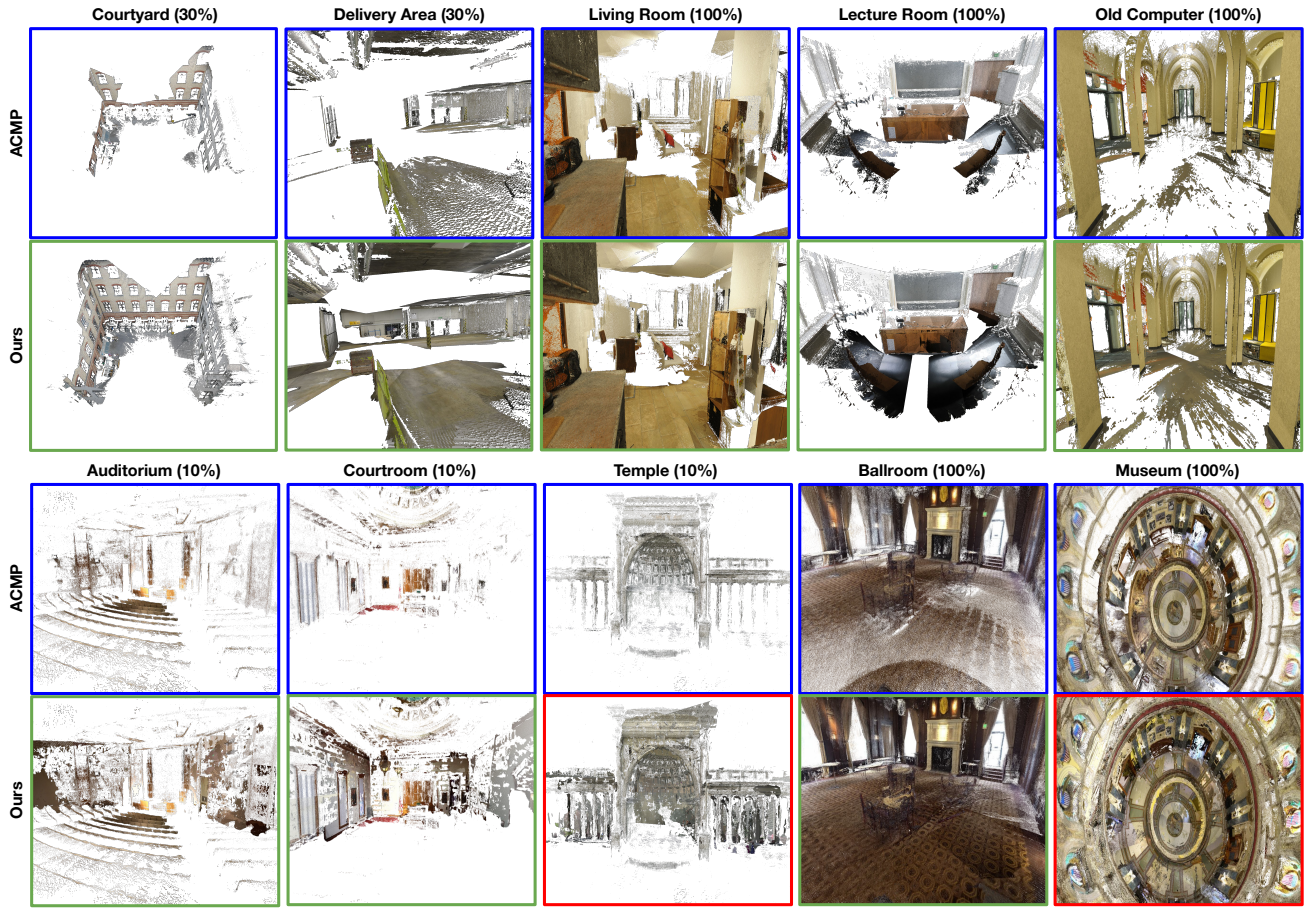

Figure 7. Qualitative results for the *ETH3D* [32] (top two rows) and *Tanks and Temples* [19] (bottom two rows) datasets. We state the scene names and the percentage of images (from the original image set) used for reconstruction as column headings. The ACMP [39] baseline results are outlined in **blue**, and our results are outlined in **green** or **red**, indicating an increase or drop in performance, respectively, compared to the baseline.

ACMP [39].

Counter maps are very beneficial for regions viewed only once. When no source images exist, counter maps yield more complete depth maps since depth estimates from the base system (ACMP) are unreliable (for these regions). As shown in Figure 6, the performance increase for our final model is close to double (from approximately 17% to 34%) in the *one* total image views group (when compared to the *"-Counter maps"* configuration).

Normal maps and RGB images have a small positive impact for regions viewed only once. For regions viewed only once, no geometric cues exist and our network solely relies on the learned appearance model for inference. Normal maps help in disambiguation of surfaces that appear similar in appearance and we surmise RGB images help for cases where the inferred normal maps are noisy. As shown in Figure 6, our final model yields an increase in coverage of 4% over the models that are not trained with normal maps and RGB images (*"-Normal maps"* and *"-RGB"* configurations).

5. Conclusion

Our method addresses the low-shot MVS problem, improving completeness of insufficiently viewed portions of the scene. We accomplish this by detecting planar surfaces in depth maps, and generate features such as normal, counter, and distance maps, which are input into our plane extent segmentation network to determine plane extents. Then, we update base depth and normal maps from the MVS system using the inferred extents, and perform depth map fusion. Results show that our method can use depth maps from any MVS system and yields improvements on F1 scores for the low-shot case (when surfaces are only viewed one or two times) on two different datasets.

6. Acknowledgments

This work was funded in part by NSF award 2020227 and gifts from Microsoft, Amazon, and Intel.

References

[1] Drone deploy. https://www.dronedeploy.com/. 1

[2] Pix4d. https://pix4d.com/.

[3] Reconstruct. https://www.reconstructinc.com/. 1

[4] Henrik Aanæs, Rasmus Jensen, George Vogiatzis, Engin Tola, and Anders Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, 120, 2016. 2

[5] Michel Antunes, Joao P Barreto, and Urbano Nunes. Piecewise-planar reconstruction using two views. *Image and Vision Computing*, 46, 2016. 2

[6] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo - stereo matching with slanted support windows. In *BMVC*, 2011. 2

[7] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation, 2017. 2, 4

[8] Ziang Cheng, Hongdong Li, Yuta Asano, Yinqiang Zheng, and Imari Sato. Multi-view 3d reconstruction of a texture-less smooth surface of unknown generic reflectance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16226–16235, 2021. 1

[9] Alejo Concha and Javier Civera. Dense Piecewise Planar Tracking and Mapping from a Monocular Sequence. In *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2015. 2

[10] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 4

[11] Andrey Dimitrov and Mani Golparvar-Fard. Segmentation of building point cloud models including detailed architectural/structural features and mep systems. *Automation in Construction*. 2

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020. 2

[13] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10786–10796, 2021. 3, 5

[14] 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*, 1981. 2

[15] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Manhattan-world stereo. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1422–1429, 2009. 2

[16] David Gallup, Jan-Michael Frahm, and Marc Pollefeys. Piecewise planar and non-planar stereo for urban scene reconstruction. pages 1418–1425, 2010. 2

[17] J. Illingworth and J. Kittler. The adaptive hough transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):690–698, 1987. 2

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

[19] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017. 1, 4, 5, 6, 7, 8

[20] Andreas Kuhn, Christian Sormann, Mattia Rossi, Oliver Erdler, and Friedrich Fraundorfer. Deepc-mvs: Deep confidence prediction for multi-view stereo reconstruction, 2019. 3

[21] P. Kultanen, Lei Xu, and Erkki Oja. Randomized hough transform (rht). pages 631 – 635 vol.1, 1990. 2

[22] Boying Li, Yuan Huang, Zeyu Liu, Danping Zou, and Wenxian Yu. Structdepth: Leveraging the structural regularities for self-supervised indoor depth estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021. 2

[23] Chen Liu, Jimei Yang, Duygu Ceylan, Ersin Yumer, and Yasutaka Furukawa. Planenet: Piece-wise planar reconstruction from a single rgb image. pages 2579–2588, 2018.

[24] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. Planercnn: 3d plane detection and reconstruction from a single image. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4445–4454, 2019. 2

[25] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021. 2

[26] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction, 2021. 2

[27] Carolina Raposo and Joao P Barreto. πmatch: Monocular vslam and piecewise planar reconstruction using fast plane correspondences. pages 380–395, 2016. 2

[28] Carolina Raposo, Michel Antunes, and João P. Barreto. Piecewise-planar stereoscan: Sequential structure and motion using plane primitives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(8):1918–1931, 2018. 2

[29] Andrea Romanoni and Matteo Matteucci. TAPA-MVS: textureless-aware patchmatch multi-view stereo. *CoRR*, abs/1903.10929, 2019. 1, 2

[30] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient ransac for point-cloud shape detection. *Computer Graphics Forum*, 26(2):214–226, 2007. 2, 3, 5

[31] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 2, 5, 6, 7

[32] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 4, 5, 6, 7, 8

[33] Sudipta Sinha, Drew Steedly, and Rick Szeliski. Piecewise planar stereo for image-based rendering. In *2009 International Conference on Computer Vision*, 2009. 2

[34] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions, 2021. 2

[35] O. J. Woodford, P. H. S. Torr, I. D. Reid, and A. W. Fitzgibbon. Global stereo reconstruction under second order smoothness priors. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 2

[36] 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

[37] Qingshan Xu and Wenbing Tao. Multi-view stereo with asymmetric checkerboard propagation and multi-hypothesis joint view selection. *CoRR*, abs/1805.07920, 2018. 1

[38] Qingshan Xu and Wenbing Tao. Multi-scale geometric consistency guided multi-view stereo. *Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[39] Qingshan Xu and Wenbing Tao. Planar prior assisted patchmatch multi-view stereo. *CoRR*, abs/1912.11744, 2019. 1, 2, 3, 5, 6, 7, 8

[40] Kohei Yamashita, Yuto Enyo, Shohei Nobuhara, and Ko Nishino. nlmvs-net: Deep non-lambertian multi-view stereo, 2022. 1

[41] Haiyang Ying, Zhang Jinzhi, Yuzhe Chen, Zheng Cao, Jing Xiao, Ruqi Huang, and Lu Fang. Parsemvs: Learning primitive-aware surface representations for sparse multi-view stereopsis. pages 6113–6124, 2022. 2

[42] Enliang Zheng, Enrique Dunn, Vladimir Jojic, and Jan-Michael Frahm. Patchmatch based joint view selection and depthmap estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[43] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H.S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6877–6886, 2021. 2