arXiv:2208.04307v1 [cs.CV] 8 Aug 2022

PlaneFormers: From Sparse View Planes to 3D Reconstruction

Samir Agarwala, Linyi Jin, Chris Rockwell, and David F. Fouhey

University of Michigan, Ann Arbor
{samirag,jinlinyi,cnris,fouhey}@umich.edu

Abstract. We present an approach for the planar surface reconstruction of a scene from images with limited overlap. This reconstruction task is challenging since it requires jointly reasoning about single image 3D reconstruction, correspondence between images, and the relative camera pose between images. Past work has proposed optimization-based approaches. We introduce a simpler approach, the PlaneFormer, that uses a transformer applied to 3D-aware plane tokens to perform 3D reasoning. Our experiments show that our approach is substantially more effective than prior work, and that several 3D-specific design decisions are crucial for its success. Project page: https://samiragarwala.github.io/PlaneFormers.

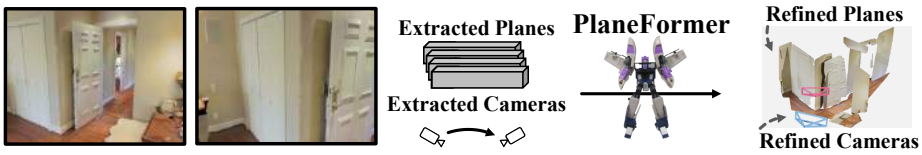

Fig. 1. Given a sparse set of images, our method detects planes and cameras, and produces plane correspondences and refined cameras using a Plane Transformer (PlaneFormer [60]), from which it can reconstruct the scene in 3D.

1 Introduction

Consider the two images shown in Figure 1. Even though you are not provided with the relative pose between the cameras that took the pictures you see, and even though you have never been to this particular location, you can form a single coherent explanation of the scene. You may notice, for instance, the doors and closet that are visible in both pictures. From here, you can deduce the relative positioning of the cameras and join your 3D perception of each image. The goal of this paper is to further the ability of computers to solve this problem.

The *sparse view* (wide and unknown baseline, few image) setting is challenging for existing systems because it falls between two main strands of 3D

reconstruction in contemporary computer vision: multiview 3D reconstruction (usually by correspondence) and learned single view 3D reconstruction (usually by statistical models). In particular, traditional multiview tools [17,46,47,41,25] depend heavily on triangulation as a cue. Thus, in addition to struggling when view overlap is small, their cues usually entirely fail with no overlap. While single-view tools [19,12,34] can reconstruct single views via learning, merging the overlap between the views to produce one coherent reconstruction is challenging: identifying whether one extracted wall goes with another requires understanding appearance, the local geometry, as well as the relationship between the cameras.

Existing approaches in this multiview area have key limitations in either input requirements or approach. Many approaches assume known camera poses [26,39], which fundamentally changes the problem by restricting a search for correspondence for a pixel in one image to a single line in another [17]. While some works relax the assumption [30] or avoid it via many images [21], these have not been demonstrated in the few-image, wide baseline case. Most work in the sparse view setting (e.g., [4,8]) does pose estimation but not reconstruction and works that produce reconstructions from sparse views [42,24] come with substantial limitations. Qian et. al [42] require multiple networks, watertight synthetic ground-truth, and use a heuristic RANSAC-like search. Jin et. al [24] apply a complex hand-designed discrete/continuous optimization applied to plane segments found by an extended PlaneRCNN [34] output. This optimization includes bundle-adjustment on SIFT [36] on viewpoint-normalized texture like VIP [61].

We propose an approach (§3), named the PlaneFormer, that overcomes these limitations. Following existing work in this area [42,24], we construct a scene reconstruction by merging scene elements that are visible in multiple views and estimating relative camera transformations. We build on [24] and construct a piecewise planar reconstruction from the images. However, rather than perform an optimization, we directly train a transformer that ingests the scene components as tokens. These tokens integrate 3D knowledge and a working hypothesis about the relative pose between input views. As output, this transformer estimates plane correspondence, predicts the accuracy of the working hypothesis for the relative poses, as well as a correction to the poses. By casting the problem via transformers, we eliminate manual design and tuning of an optimization. Moreover, once planes are predicted, our reconstruction operations are performed via transformer forward passes that test out hypothesized relative camera poses.

Our experiments (§4) on Matterport3D [6] demonstrate the effectiveness of our approach compared to other approaches. We evaluate with set of image pairs with limited overlap (mean rotation: 53°; translation: 2.3m; overlap: 21%). Our approach substantially surpasses the state of the art [24] before its post-processing bundle adjustment step: the number of pairs registered within 1m increases from 56.5% to 66.8%, and pairs with 90% correspondences correct increases from 28.1% to 40.6%. Even when [24] uses the additional bundle adjustment step, the our approach matches or exceeds the method. We next show that our approach can be used on multiple views, and that several 3D design decisions in the construction of the PlaneFormer are critical to its success.

2 Related Work

Our approach to 3D reconstruction from sparse views draws upon the well-studied tasks of correspondence estimation, i.e., 3D from many images; and learning strong 3D priors, i.e., 3D from a single image.

Correspondence and camera pose estimation. The tasks of estimating correspondences and relative camera pose [13,45,10,67,63,66] across images are central to predicting 3D structure from multiple images [3,46,20,50]. Some methods jointly refine camera and depth across many images [33,51,40,30,68] in a process classically approached via Bundle Adjustment [52,1]. We also refine both camera and reconstruction; however, we do not have the requirement of many views. Additionally, we use self-attention, a powerful concept that has been successfully used in several vision tasks [45,49,31,3,69]. Our approach of using self-attention through transformers [54] is similar to SuperGlue [45] and LoFTR [49] in that it permits joint reasoning over the set of potential correspondences. We apply it to the task of planes, and also show that the learned networks can also predict relative camera pose directly (via residuals to a working hypothesis).

3D from a single image. Learned methods have enabled 3D inference given only a single viewpoint. These methods cannot rely on correspondences, and therefore use image cues along with learned priors and a variety of representations. Their 3D structure representations include voxels [11,48], meshes [16,55], point clouds [14,59], implicit functions [38,23], depth [43,29], surface normals [58,9], and planes [35,62,65]. We use planes to reconstruct 3D as they are often good approximations [15] and have strong baselines for detection such as PlaneRCNN [34]; we build off this architecture. In contrast with PlaneRCNN and single-image methods, we incorporate information across multiple views and therefore can also use correspondences.

3D from sparse views. Recent approaches enable learned reasoning with multiple views. Several works perform novel view synthesis using radiance fields [64,22,56,7]. Learned methods also estimate pose [4,57] and depth [53,27] given few views, but do not create a unified scene reconstruction. Our focus is also on wide-baseline views [41], further separating us from monocular stereo methods [53,27]. Two recent works approach this task. Qian *et al.* [42] reconstruct objects from two views but use heuristic stitching across views, and struggle on realistic data [24]. Jin *et al.* [24] jointly optimize plane correspondences and camera pose from two views with a hand-designed optimization. In contrast, we use a transformer to directly predict plane correspondence and camera pose. Our experiments (§4) show our approach outperforms these methods.

3 Approach

Our approach aims to jointly reason about a pair of images with an unknown relationship and reconstruct a single, coherent, global planar reconstruction of the scene depicted by the images. This process entails extracting three key related pieces of information: the position of the planes that constitute the scene; the

correspondence between the planes in each view so that each real piece of the scene is reconstructed once and only once; and finally, the previously unknown relationship between the cameras that took each image.

At the heart of our approach is a plane transformer that accepts an initial independent reconstruction of each view and hypothesized global coordinate frame for the cameras. In a single forward pass, the plane transformer identifies which planes correspond with each other, predicts whether the cameras have correct relative pose, and estimates an updated relative camera pose as a residual. Inference for the scene consists of running one forward pass of the PlaneFormer network per camera hypothesis.

3.1 Backbone Plane Predictor

The PlaneFormer is built on top of a single-view plane estimation backbone from [24], which is an extended version of PlaneRCNN [34]. We refer the reader to [24] for training details, but summarize the key properties here. This plane backbone produces two outputs: per-image planes and a probability distribution over relative camera poses.

Plane Branch. The per-image planes are extracted from an image I_i via a Mask-RCNN [18]-like architecture. This architecture detects a set of plane segments, yielding M_i detections. Each detected segment in the ith image is indexed by j and has a mask segment $\mathcal{S}_{i,j}$, plane parameters $\boldsymbol{\pi}_{i,j} \in \mathbb{R}^4$, and appearance embedding $\mathbf{e}_{i,j} \in \mathbb{R}^{128}$. The plane parameter $\boldsymbol{\pi}_{i,j}$ can further be factored into a normal $\mathbf{n}_{i,j}$ and offset $o_{i,j}$ (defining a plane equation $\mathbf{n}_{i,j}^T[x, y, z] - o_{i,j} = 0$). The appearance embedding can be used to match between images i and i': the distance in embedding space $||\mathbf{e}_{i,j} - \mathbf{e}_{i',j'}||$ ought to be small whenever plane j' corresponds to the plane j.

Camera Branch. The backbone also produces a probability distribution over a predefined codebook of relative camera transformations $\{\hat{\mathbf{R}}_k, \hat{\mathbf{t}}_k\}$. To predict this distribution, it uses a CNN applied to cross-attention features between early layers of the network backbone (specifically, the `P3` layers of the ResNet-50-FPN [32]). This camera branch combines cross-attention features [24,4] and pose via regression-by-classification [42,24,8,4], both of which have been shown to lead to strong performance. While a strong baseline, past work [42,24] has shown that the predictions of such networks need to be coupled with reasoning. For instance, [24] uses the probability for each camera pose as a term in its optimization problem. In our case, we use it to generate a set of initial hypotheses about the relative camera poses.

3.2 The PlaneFormer

The core of our method is a transformer [54,2] that jointly processes the planes detected in the 2 images given a hypothesized global coordinate system for the images. Since transformers operate on sets of inputs using a self-attention mechanism, they are able to consider context from all inputs while making predictions. This makes them effective in tasks such as ours where we want to collectively

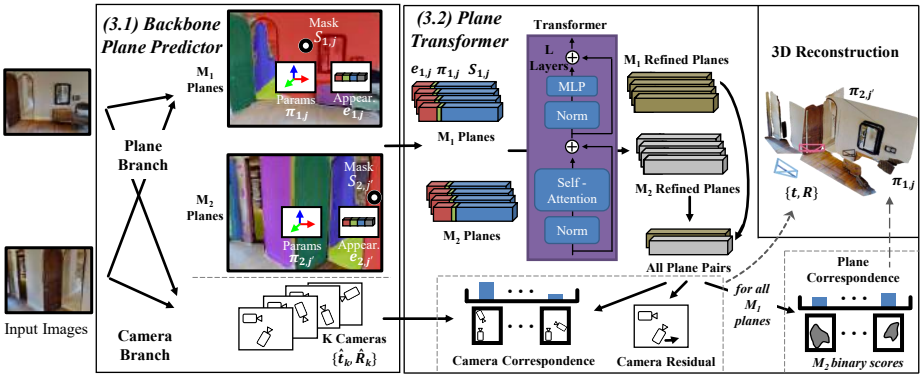

Fig. 2. Approach. Given two input images, the backbone network detects planes and predicts camera pose across images. The plane transformer refines these planes by predicting correspondence and refined camera pose, producing a final 3D reconstruction.

reason about multiple planes across the images to generate a coherent reconstruction. In our case, the transformer takes in a set of feature vectors representing plane detections as the input and maps them to an equal number of outputs which are then further processed and passed through MLP heads to predict our outputs. This function can take in a variable number of feature vectors as an input and is learned end-to-end. The plane transformer aims to identify: correspondence between the planes (i.e., whether they represent the same plane in 3D and can be merged), whether the hypothesized relative camera pose is correct, and how to improve the relative camera pose.

As input, the plane transformer takes $M = M_1 + M_2$ tokens. These tokens contain features that integrate the hypothesized coordinate frame to help the transformer. The hypothesized coordinate frame consists of rotations and translations $\{\mathbf{R}_i, \mathbf{t}_i\}$ that are hypothesized to bring the two camera views into a common coordinate frame. For convenience, we assume that the common coordinate frame is centered at image 2 (i.e., $\mathbf{R}_2 = \mathbf{I}_{3\times3}$, $\mathbf{t}_2 = \mathbf{0}$), but note that the transformer never has explicit access to these rotation and translations.

Input Features. We concatenate three inputs to the network from the plane backbone to represent each of the M plane tokens. Each token is 899D.

Appearance Features (128D). The first token part is the appearance feature from [24]'s extended PlaneRCNN.

Plane Features (3D). The second part of the token is the plane equation $\boldsymbol{\pi}_{i,j}$ comprising a normal that is scaled by the plane offset. This equation is transformed to the hypothesized canonical space by $\mathbf{R}_i, \mathbf{t}_i$. This feature functions like a positional encoding, and enables logic such as: if two planes have a similar appearance and plane parameters, then they are likely the same plane.

Mask Features (768D). We directly provide information about the plane segments via mask features. These features complement the plane features since they represent a plane segment rather than an infinite plane. We use the hy-

pothesized relative camera pose and 3D to produce mask features. The mapping from image i to the common coordinate frame's view for plane j is given by a homography $\mathbf{H} = \mathbf{R}_i + (\mathbf{t}_i^T \mathbf{n}_{i,j})/o_{i,j}$ [37]. This lets us warp each mask $\mathcal{S}_{i,j}$ to a common reference frame. Once the mask is warped to the common reference frame, we downsample it to a 24×32 image. We hypothesize that the explicit representation facilitates reasoning such as: these two planes look the same, but they are on opposite sides so the provided transformation may be wrong; or these planes are the same and roughly in the right location but one is bigger, so the translation ought to be adjusted.

Outputs. As output, we produce a set of tensors that represent plane correspondence, whether cameras have the correct relative transformation, and updated camera transformations. Specifically, the outputs are:

Plane Correspondence. $\boldsymbol{\Pi} \in \mathbb{R}^{M_1 \times M_2}$ that gives the correspondence score between two planes across the input images. If $\boldsymbol{\Pi}_{j,j'}$ is large, then the planes j and j' likely are the same plane in a different view. We minimize a binary cross-entropy loss between the predicted $\boldsymbol{\Pi}$ and ground-truth..

Camera Correspondence. $C \in \mathbb{R}$ that indicates whether the two cameras have the correct relative transformation. If C has a high score, then it is likely that the hypothesized relative relative pose between the input cameras is correct. We minimize a binary cross-entropy loss between the predicted C and ground-truth.

Camera Residual. $\boldsymbol{\Delta} \in \mathbb{R}^7$ giving a residual for the hypothesized relative pose. This residual is expressed as the concatenation of a 4D quaternion for rotation and 3D translation vector for translation. Updating the relative transformation between the cameras is likely to improve the transformation. We minimize an L_1 loss between the predicted camera rotation and translation residual and the ground-truth camera rotation and translation residual with relative weight λ_t to translation. During training, hypothesized camera poses come from the codebook from the Camera Branch; thus there is a residual that needs to be corrected.

PlaneFormer Model. A full description of the method appears in the supplement, but our PlaneFormer consists of a standard transformer followed by the construction of pairwise features between planes. The transformer maps the M plane input tokens to M output tokens using a standard Transformer [54] with 5 layers and a feature size equal to plane tokens of 899. We use only a single head to facilitate joint modeling of all plane features.

After the transformer produces M per-plane output tokens, the M outputs are expanded to $M \times M$ pairwise features in an outer-product-like fashion. To assist in prediction, we also produce a per-image token: given M output tokens, where $\mathbf{o}_{i,j}$ denotes the output token for the jth plane in image i, we compute the per-image average token $\boldsymbol{\mu}_i = (1/M_i) \sum_{j=1}^{M_i} \mathbf{o}_{i,j}$. The pairwise feature for planes (i, j) and (i', j') is the concatenation of the plane output tokens $\mathbf{o}_{i,j}$, $\mathbf{o}_{i',j'}$, and their per-image tokens $\boldsymbol{\mu}_i$, and $\boldsymbol{\mu}_{i'}$. This 3596 (4×899) feature is passed into separate 4-hidden-layer MLP heads that estimate $\boldsymbol{\Pi}$, C, and $\boldsymbol{\Delta}$ per-pair of planes. At each hidden layer, we halve the input feature dimension. We average pool the MLP output over plane pairs across the images to produce the

final estimate for C and $\boldsymbol{\Delta}$. $\boldsymbol{\Pi}$ can be used after masking to $M_1 \times M_2$. Finally, we apply a sigmoid function to C and $\boldsymbol{\Pi}$ to generate the model output.

3.3 Inference

Once the PlaneFormer has been trained, we can apply it to solve reconstruction tasks. Given a set of planes and hypothesized poses of cameras, the Planeformer can estimate correspondence, identify whether the hypothesized poses are correct, and estimate a correction to the camera poses.

Two View Inference. Given two images, one takes the top h hypotheses for the relative camera pose from the Camera Branch (§3.1) and evaluates them with the PlaneFormer. The pose hypothesis with highest camera correspondence score is selected, and the predicted residual is added. We note that these camera pose hypotheses can be explored in parallel, since they only change the token features. After the plane correspondences have been predicted, we match using the Hungarian Algorithm [28] with thresholding. Sample outputs of PlaneFormer appear in Fig. 3 and throughout the paper.

Multiview Inference. In order to extend the method to multiple views, we can apply it pairwise to the images. We apply the above approach pairwise to edges in an acyclic view graph that connects the images. The graph is generated greedily on a visibility score for a pair of images (i, i') that represents the number of planes with close matches. For the appearance embedding $\mathbf{e}_{i,j}$ of plane j in image i, we compute the minimum distance to the appearance embeddings of the planes in image i', or $d_j = \min_{j'} ||\mathbf{e}_{i,j} - \mathbf{e}_{i',j'}||$. Rather than threshold the distance, we softly count the numbers of close correspondence via a score $\sum_j \exp(-d_j^2/\sigma^2)$. We repeat the process from i' to i and then sum for symmetry.

3.4 Training and Implementation Details

Training procedure. During training, each sample must assume a set of camera transformations (i.e., $\{\mathbf{R}_i, \mathbf{t}_i\}$). We train on a mix of correct camera transformations (using the nearest rotation and translation in the sparse codebook) and incorrect camera transformations (using a randomly selected non-nearest rotation and translation in the sparse codebook). Given a correct camera hypothesis, we backpropagate losses on all outputs; given an incorrect hypothesis, we backpropagate losses only on the camera correspondence C. Thus, the camera correspondence output is trained to discriminate between correct and incorrect cameras, and the other outputs do not have their training contaminated (e.g., by having to predict residuals even if the camera hypothesis is completely incorrect). The correct and incorrect cameras are sampled equally during training.

Implementation Details. We train for 40k iterations using a batch size of 40 and the same Matterport3D [6] setup as Jin *et al.* [24] We use SGD with momentum of 0.9 and a learning rate of 1e-2, and follow a one-cycle cosine annealing schedule. We weight all losses equally, with the exception of $\lambda_t = 0.5$ for the residual translation loss. Training takes about 36 hours using 4 RTX 2080 Tis. At inference, we select from $h = 9$ camera hypotheses.

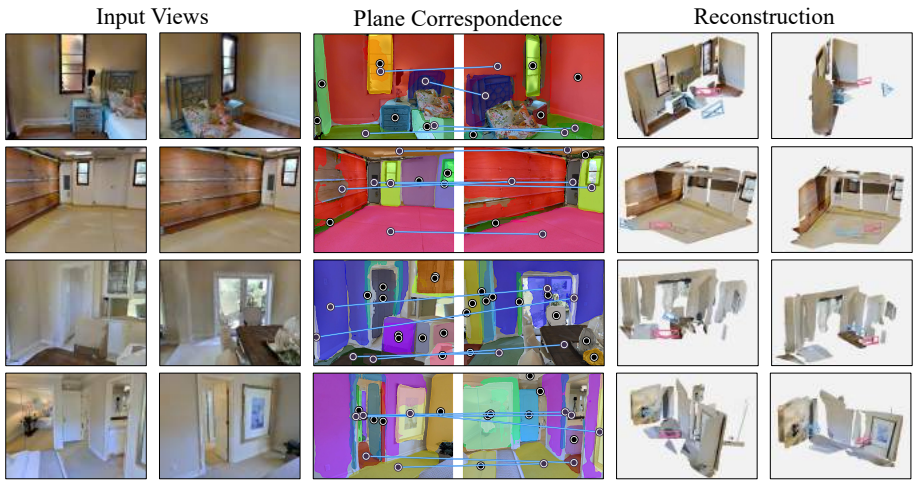

Fig. 3. Sample Outputs on the Test Set. PlaneFormer produces jointly refined plane correspondences and cameras, from which it reconstructs the input scene. It can produce high-quality reconstructions in cases of moderate view change (top two rows), and coherent reconstructions in cases of large view change (bottom two rows).

4 Experiments

We now evaluate the proposed approach in multiple settings. We first introduce our experimental setup, including metrics and datasets. We then introduce experiments for the wide baseline two view case in §4.1. The two view setting has an abundance of baselines that we compare with. Next, we introduce experiments for more views, specifically 3 and 5 views, in §4.2. Finally, we analyze which parts of our method are most important in §4.3.

Metrics. The sparse view reconstruction problem integrates several challenging, complex problems: detecting 3D planes from a 2D image, establishing correspondence across images, and estimating relative camera pose. We therefore evaluate the problem in multiple parts.

Plane Correspondence. We evaluate correspondence separately. We follow Cai et al. [5] and use IPAA-X, or the fraction of image pairs with no less than X% of planes associated correctly. We use ground-truth plane boxes in this setting since otherwise this metric measures both plane detection and plane correspondence.

Relative Camera Pose. We next evaluate camera relative pose estimation. We follow [42,24] and report the mean error, the median error, and the fraction of image pairs with error below a threshold of 30° and 1m following [42].

Full Scene Results. Finally, we report results using the full scene metric from [24]. This metric counts detected planes as true positives if their mask IoU is $\geq$ 0.5, surface normal distance is $\leq 30°$; and offset distance is less than 1m. This metric integrates all three components: to get the planes correct, one needs to reconstruct them in 3D, estimate the relative camera configuration to map

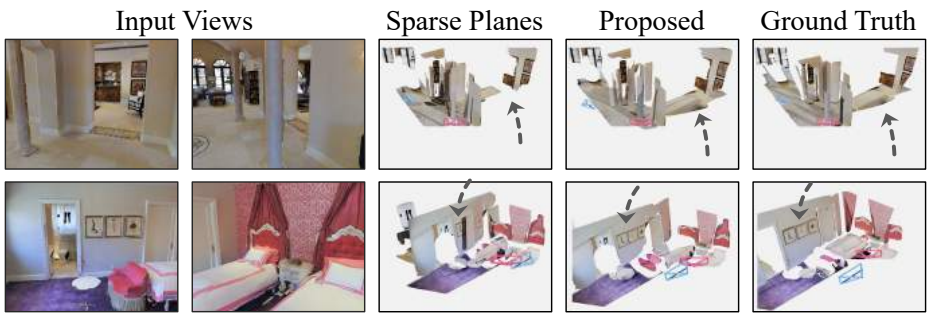

Fig. 4. Reconstruction Comparison. Sparse Plane reconstructions are a good, but PlaneFormer's are better in terms of stitched planes (top), and camera poses (bottom).

the second view's planes into the first view, and identify duplicated planes to suppress false positives. While this metric is important, any component can limit performance, including components we do not alter, like plane detection.

Datasets. We evaluate on three datasets: two-, three-, and five-view. *Two view dataset:* We use the exact dataset used in [24] for fair comparison. This consists of 31392 training image pairs, 4707 validation image pairs, and 7996 test image pairs. These views are widely spearated. On average: view overlap is 21% of pixels; relative rotation is 53°; and relative translation is 2.3m. *Multiview datasets:* We generate a set of 3- and 5- view pairs using the same procedure as [24]. We evaluate on a total of 258 3-view and 76 5-view samples.

Baselines. The full problem of reconstructing the scene from a set of sparse views requires solving the three separate problems of correspondence, relative camera pose estimation, and 3D reconstruction. We compare with full systems as well as approaches that solve each problem independently.

All Settings. In all cases, we compare with *Sparse Planes* [24]. For fair comparison, we use an identical backbone to [24] so that any performance gain stems from the PlaneFormer rather than improved systems tuning. The full version is our strongest baseline. It uses the same plane information and follows it with a discrete-continuous optimization. The continuous optimization requires extracting view-normalized texture maps, performing SIFT matching, and then bundle adjustment and is expensive and a complementary contribution. Since our method does not do an additional step of extracting feature correspondences and optimizing, a more comparable system to the contribution of our paper is the discrete-optimization only version, or *Sparse Planes* [24] *(No Continuous)*, which performs all the steps except bundle adjustment on point correspondences.

Plane Correspondence. We additionally compare with (*Appearance Only*), or the Hungarian algorithm with thresholding on the appearance embedding distances. This approach outperformed other methods like [5] and [42] in [24].

Relative Camera Pose Estimation. We also compare with a number of other methods. The most important is the (*Sparse Planes* [24] Camera Branch) which is the top prediction from the Camera Branch network that our system uses for

Table 1. Two View Plane Correspondence. IPAA-X [5] measures the fraction of pairs with no less than X% of planes associated correctly. Ground truth bounding boxes are used. Since the Sparse Planes continuous optimization does not update correspondence, there is not a separate entry for Sparse Planes without continuous optimization.

	IPAA-100	IPAA-90	IPAA-80
Appearance Only	6.8	23.5	55.7
Sparse Planes [24]	16.2	28.1	55.3
Proposed	**19.6**	**40.6**	**71.0**

Table 2. Two View Relative Camera Pose. We report median, mean error and % error $\leq$ 1m or 30° for translation and rotation.

	Translation			Rotation		
Method	Med.	Mean	($\leq$1m)	Med.	Mean	($\leq$30°)
Odometry [44] + GT Depth	3.20	3.87	16.0	50.43	55.10	40.9
Odometry [44] + [43]	3.34	4.00	8.3	50.98	57.92	29.9
Assoc. 3D [42]	2.17	2.50	14.8	42.09	52.97	38.1
Dense Correlation Volumes [4]	-	-	-	28.01	41.56	52.45
Camera Branch [24]	0.90	1.40	55.5	7.65	24.57	81.9
Sparse Planes [24] (No Continuous)	0.88	1.36	56.5	7.58	22.84	83.7
Proposed	0.66	**1.19**	**66.8**	5.96	22.20	83.8
Sparse Planes [24] (Full)	**0.63**	1.25	66.6	7.33	22.78	83.4
SuperGlue [45]	-	-	-	3.88	24.17	77.8
LoFTR-DS [49]	-	-	-	**0.71**	**11.11**	**90.47**

hypotheses. Gain over this is attributable to the PlaneFormer camera correspondence and residual branch, since these produce different relative camera poses. Other methods include: (Odometry [44] + GT/[43]), which combines a RGBD odometry with ground-truth or estimated depth; (*Assoc. 3D* [42]), a previous approach for camera pose estimation; (*Dense Correlation Volumes* [4]) which uses correlation volumes to predict rotation; *SuperGlue* [45] and *LoFTR* [49], which are learned feature matching system. Since [45] and [49] solve for an essential matrix, their estimate of translation is intrinsically scale-free [17].

Full Scene Reconstruction. For full-evaluation, we report some of the top performing baselines from [24] along with [49]. These are constructed by joining the outputs of [24]'s extended PlaneRCNN [34] with a relative camera pose estimation method that gives a joint coordinate frame. These are as described in the relative camera pose estimation, except *SuperGlue GT Scale* and *LoFTR GT Scale* are also given the ground-truth translation scale. This extra information is needed since the method intrinsically cannot provide a translation scale.

Table 3. Two View Evaluation. Average Precision, treating reconstruction as a 3D plane detection problem. We use three definitions of true positive. (*All*) requires Mask IoU ≥ 0.5, Normal error $\leq 30°$, and Offset error $\leq$ 1m. (*-Offset*) removes the offset condition; (*-Normal*) removes the normal condition.

Methods	All	-Offset	-Normal
Odometry [44] + PlaneRCNN [34]	21.33	27.08	24.99
SuperGlue-GT Scale [45] + PlaneRCNN [34]	30.06	33.24	33.52
LoFTR-DS-GT Scale [49]	33.31	36.17	35.72
Camera Branch [24] + PlaneRCNN [34]	29.44	35.25	31.67
Sparse Planes [24] (No Continuous)	35.87	42.13	38.8
Proposed	**37.62**	**43.19**	**40.36**
Sparse Planes [24] (Full)	36.02	42.01	39.04

4.1 Wide Baseline Two-View Case

Our primary point of comparison is the wide baseline two view case. This two-view case has been extensively studied and benchmarked in [24]. We have shown qualitative results of the full system in Fig. 3 by itself and show a comparison with Sparse Planes in Fig. 4. We now discuss each aspect of performance.

Plane Correspondence Results. As reported in Table 1, the PlaneFormer substantially increases IPAA across multiple metrics compared to Sparse Planes [24]. We show qualitative results on Fig. 5, including one of the images that Jin *et al.* [24] reported as a representative failure mode of their system. This particular case is challenging for geometry-based optimization since the bed footboards are co-planar and similar in appearance. These have similar appearance and plane parameter features; our mask token, however, can separate them out.

Relative Camera Pose Results. We next evaluate relative camera pose. Our results in Table 2 show that the PlaneFormer outperforms most other approaches that do not do bundle adjustment on feature correspondence: by integrating plane information, the proposed system improves over the camera branch by over 10% in translation accuracy and reduces rotation error by 22% (relative). The approach outperforms the Sparse Planes system before continuous optimization. Even when Sparse Planes performs this step, our approach outperforms it in all but one metric. Our approach is competitive with SuperGlue [45] while LoFTR [49] outperforms competing systems in rotation estimation. Since these point-feature based approaches do not provide translation scale, we see them as complementary. Future systems might benefit from both points and planes.

Full Scene Evaluation Results. We finally report full scene evaluation results (AP) in Table 3. Our approach outperforms alternate methods, including the full version of Sparse Planes. The relative performance gains of our method are smaller than compared to those for plane correspondence. However, it is important to note that the full scene evaluations is limited by every component. We hypothesize that one of the current key limiting factors is the accuracy of single-view reconstruction, or the initial PlaneRCNN [34].

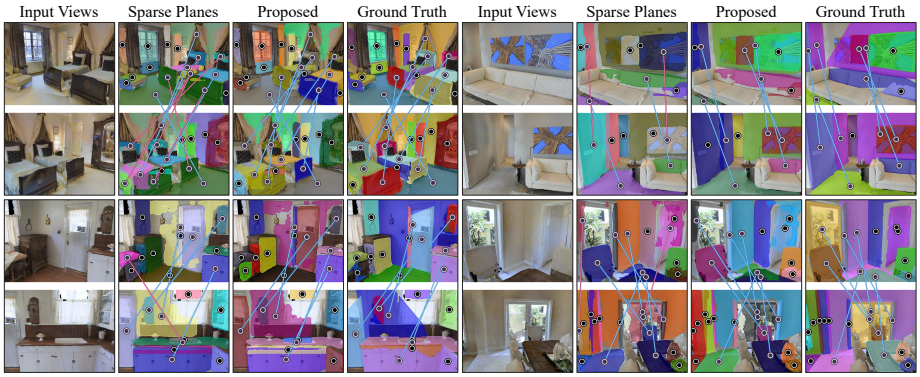

Fig. 5. Plane Comparison. Matching surfaces across large view changes is challenging. Multiple surfaces may be similar in appearance, causing correspondence mixups like bed footboards (top left) or paintings (top right). By jointly refining planes across images via a transformer, the proposed method better associates across images. It can also reduce inconsistent outlier detections (bottom).

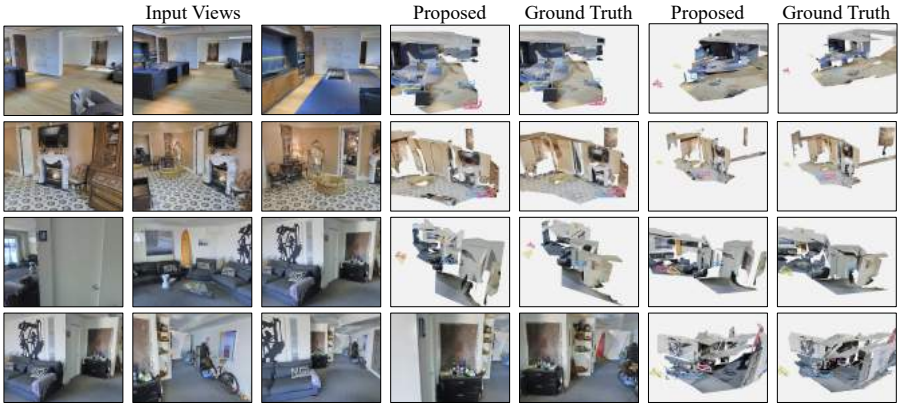

Fig. 6. Multiview Test Results. With 3 views, our approach model can often construct extensive reconstruction of rooms (top 3 rows). With 5 views, the model continues to stitch larger sets of planes together effectively (bottom row).

4.2 Wide Baseline Multiview Case

We next report the multiview case. The multiview case is substantially more challenging than performing pairwise reconstruction since the output must be a single coherent reconstruction. For instance, in relative camera pose, the composition of the rotation from image 1 to image 2 and the rotation from image 2 to image 3 must be the rotation from image 1 to image 3.

Baselines. We extend the top baselines from the two view case to the multiview case. For fair comparison, we apply each baseline to the same view graph that

Table 4. Multiview Evaluation: Plane Correspondence We report IPAA-X for 3- and 5-view datasets. Our approach continues to substantially outperform baseline methods (but overall performance drops due to the increasing difficulty of the task).

	3-view IPAA-X			5-view IPAA-X		
	IPAA-100	IPAA-90	IPAA-80	IPAA-100	IPAA-90	IPAA-80
Appearance	5.94	20.28	52.97	1.45	13.68	52.37
SparsePlanes [24]	9.95	23.77	51.16	4.87	16.58	41.45
Proposed	**14.60**	**32.69**	**66.15**	**5.92**	**20.66**	**55.92**

Table 5. Multiview Evaluation: Relative Camera Pose Estimation We report the same metrics as the two view case, while running on the 3- and 5-view dataset.

	3-view						5-view					
	Transl. Error (m)			Rot. Error (deg)			Transl. Error (m)			Rot. Error (deg)		
	Med.	Mean	$\leq$ 1m	Med.	Mean	$\leq 30^\circ$	Med.	Mean.	$\leq$ 1m	Med.	Mean	$\leq 30^\circ$
Camera [24]	1.25	2.21	41.47	9.40	37.08	71.71	1.69	2.80	29.61	13.72	48.07	63.55
No Cont. [24]	1.15	2.02	43.67	8.97	30.89	**75.97**	1.62	2.73	31.58	12.08	44.99	64.08
Proposed	**0.83**	1.81	**56.69**	**7.88**	32.22	74.94	**1.10**	2.33	47.24	**9.52**	**43.22**	**67.5**
Full [24]	0.84	**1.74**	54.91	8.83	**30.19**	75.58	1.13	**2.29**	**47.37**	11.35	44.16	64.21

is used for our method (defined in §3.3). We report both the full version of [24] and the version without the continuous optimization (*No Cont.* [24]). For correspondence, we additionally report the appearance feature only baseline, which outperforms [42,5]. For camera pose estimation, we report the Camera Branch (Camera) of [24], which outperforms multiple other baselines such as [44,42].

Quantitative Results. We report correspondence results in Table 4. As was the case in the 2-view setting, our approach substantially outperforms the baselines. Overall performance reduces as the number of views increases; this is because the space covered often spreads out as the number of images increase, raising the difficulty of reconstruction. We next report relative camera pose estimation results in Table 5. Trends are similar to the 2-view case: our method is competitive with the full pipeline of Jin et al. [24] in camera estimation, and often surpasses it. Our approach also substantially outperforms the top prediction from the Camera Branch and the discrete optimization version of [24].

Qualitative Results. We show qualitative results on 3- and 5-view inputs in Figure 6. Our method can often generate high quality scenes.

4.3 Ablations

We finally analyze ablations of the method in Table 6. We report the IPAA-90 and average camera pose translation and rotation errors. In all cases, ablations follow the same training procedure and are trained until validation accuracy plateaus. Full details and comparisons appear in supplement.

Table 6. Ablations. We perform ablations of input features (left) and network design (right). We report IPAA-90 and relative camera pose translation and rotation error.

Feature Ablation	Plane IPAA-90 ↑	Trans. Mean ↓	Rot. Mean ↓
Proposed	**40.6**	**1.19**	22.20
- Appearance	26.9	1.23	22.78
- Plane	35.2	1.32	25.92
- Mask	34.5	1.26	**21.21**

Network Ablation	Plane IPAA-90 ↑	Trans. Mean ↓	Rot. Mean ↓
Proposed	**40.6**	**1.19**	**22.20**
- Transformer	32.7	1.48	26.43
- Residual	40.6	1.34	22.38

Feature ablations. To test feature importance, we report results when each feature has been removed from the token. For fair comparison, we keep transformer feature size equal to the full model by mapping inputs through an MLP. Table 6 (left) shows all three sets of features are important for performance. Removing appearance features causes the largest drop in plane correspondence, likely due to the importance of appearance when matching many planes across images. In contrast, removing plane parameters is most damaging to camera accuracy. As plane parameters represent position and orientation, this comparison indicates the position and orientation of planes are a powerful signal for inferring relative camera pose across images. Mask features have little impact on camera performance, but are still important for plane correspondence.

Network ablations. We next test the importance of the transformer and camera residual (Table 6, right). Our no transformer model simply applies the MLP heads for plane correspondence, camera correctness, and residuals. The plane features do not interact in this model, but it outperforms all prior baselines for plane correspondence and is competitive in camera pose, which illustrates the value of discriminatively learned correspondence rather than optimization. However, adding a transformer to enable inter-plane interaction substantially improve performance, with IPAA-90 increasing by nearly 8%. The camera residual is also an important design decision, enabling refinement on predicted cameras. Note even without the residual, camera performance is similar to or better than all baselines. The camera residual does not impact plane predictions.

5 Conclusion

We have introduced a new model for performing reconstructions between images separated by wide baselines. Our approach replaces hand-designed optimization with a discriminatively learned transformer and shows substantial improvements over the state of the art across multiple metrics and settings.

Acknowledgements. This work was supported by the DARPA Machine Common Sense Program. We would like to thank Richard Higgins and members of the Fouhey lab for helpful discussions and feedback.

References

1. Agarwal, S., Snavely, N., Seitz, S.M., Szeliski, R.: Bundle adjustment in the large. In: ECCV (2010) 3
2. Bloem, P.: Transformers from scratch (2019), http://peterbloem.nl/blog/transformers 4
3. Bozic, A., Palafox, P., Thies, J., Dai, A., Nießner, M.: Transformerfusion: Monocular rgb scene reconstruction using transformers. NeurIPS (2021) 3
4. Cai, R., Hariharan, B., Snavely, N., Averbuch-Elor, H.: Extreme rotation estimation using dense correlation volumes. In: CVPR (2021) 2, 3, 4, 10
5. Cai, Z., Zhang, J., Ren, D., Yu, C., Zhao, H., Yi, S., Yeo, C.K., Loy, C.C.: Messytable: Instance association in multiple camera views. In: ECCV (2020) 8, 9, 10, 13
6. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3d: Learning from rgb-d data in indoor environments. In: 3DV (2017) 2, 7
7. Chen, A., Xu, Z., Zhao, F., Zhang, X., Xiang, F., Yu, J., Su, H.: Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In: ICCV (2021) 3
8. Chen, K., Snavely, N., Makadia, A.: Wide-baseline relative camera pose estimation with directional learning. In: CVPR (2021) 2, 4
9. Chen, W., Qian, S., Fan, D., Kojima, N., Hamilton, M., Deng, J.: Oasis: A large-scale dataset for single image 3d in the wild. In: CVPR (2020) 3
10. Choy, C., Dong, W., Koltun, V.: Deep global registration. In: CVPR (2020) 3
11. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: ECCV (2016) 3
12. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: ICCV (2015) 2
13. El Banani, M., Gao, L., Johnson, J.: Unsupervised r&r: Unsupervised point cloud registration via differentiable rendering. In: CVPR (2021) 3
14. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3d object reconstruction from a single image. In: CVPR (2017) 3
15. Furukawa, Y., Curless, B., Seitz, S.M., Szeliski, R.: Manhattan-world stereo. In: CVPR (2009) 3
16. Gkioxari, G., Malik, J., Johnson, J.: Mesh r-cnn. In: ICCV (2019) 3
17. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, ISBN: 0521540518 (2004) 2, 10
18. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV (2017) 4
19. Hoiem, D., Efros, A.A., Hebert, M.: Geometric context from a single image. In: ICCV (2005) 2
20. Huang, P.H., Matzen, K., Kopf, J., Ahuja, N., Huang, J.B.: Deepmvs: Learning multi-view stereopsis. In: CVPR (2018) 3
21. Huang, Z., Li, T., Chen, W., Zhao, Y., Xing, J., LeGendre, C., Luo, L., Ma, C., Li, H.: Deep volumetric video from very sparse multi-view performance capture. In: ECCV (2018) 2
22. Jain, A., Tancik, M., Abbeel, P.: Putting nerf on a diet: Semantically consistent few-shot view synthesis. In: ICCV (2021) 3
23. Jiang, C., Sud, A., Makadia, A., Huang, J., Nießner, M., Funkhouser, T., et al.: Local implicit grid representations for 3d scenes. In: CVPR (2020) 3

24. Jin, L., Qian, S., Owens, A., Fouhey, D.F.: Planar surface reconstruction from sparse views. In: ICCV (2021) 2, 3, 4, 5, 7, 8, 9, 10, 11, 13, 19
25. Jin, Y., Mishkin, D., Mishchuk, A., Matas, J., Fua, P., Yi, K.M., Trulls, E.: Image Matching across Wide Baselines: From Paper to Practice. IJCV (2020) 2
26. Kar, A., Häne, C., Malik, J.: Learning a multi-view stereo machine. In: NeurIPS (2017) 2
27. Kopf, J., Rong, X., Huang, J.B.: Robust consistent video depth estimation. In: CVPR (2021) 3
28. Kuhn, H.W.: The hungarian method for the assignment problem. Naval Research Logistics Quarterly (1955) 7
29. Li, Z., Snavely, N.: Megadepth: Learning single-view depth prediction from internet photos. In: CVPR (2018) 3
30. Lin, C.H., Ma, W.C., Torralba, A., Lucey, S.: Barf: Bundle-adjusting neural radiance fields. In: ICCV (2021) 2, 3
31. Lin, K., Wang, L., Liu, Z.: End-to-end human pose and mesh reconstruction with transformers. In: CVPR (2021) 3
32. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR (2017) 4
33. Lindenberger, P., Sarlin, P.E., Larsson, V., Pollefeys, M.: Pixel-perfect structure-from-motion with featuremetric refinement. In: ICCV (2021) 3
34. Liu, C., Kim, K., Gu, J., Furukawa, Y., Kautz, J.: Planercnn: 3d plane detection and reconstruction from a single image. In: CVPR (2019) 2, 3, 4, 10, 11
35. Liu, C., Yang, J., Ceylan, D., Yumer, E., Furukawa, Y.: Planenet: Piece-wise planar reconstruction from a single rgb image. In: CVPR (2018) 3
36. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV (2004) 2
37. Ma, Y., Soatto, S., Košecká, J., Sastry, S.: An invitation to 3-d vision: from images to geometric models. Springer (2004) 6
38. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: CVPR (2019) 3
39. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020) 2
40. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. TOG (2015) 3
41. Pritchett, P., Zisserman, A.: Wide baseline stereo matching. In: ICCV (1998) 2, 3
42. Qian, S., Jin, L., Fouhey, D.F.: Associative3d: Volumetric reconstruction from sparse views. In: ECCV (2020) 2, 3, 4, 8, 9, 10, 13
43. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. TPAMI (2020) 3, 10
44. Raposo, C., Lourenço, M., Antunes, M., Barreto, J.P.: Plane-based odometry using an rgb-d camera. In: BMVC (2013) 10, 11, 13
45. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks. In: CVPR (2020) 3, 10, 11
46. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR (2016) 2, 3
47. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise View Selection for Unstructured Multi-View Stereo. In: ECCV (2016) 2
48. Song, S., Yu, F., Zeng, A., Chang, A.X., Savva, M., Funkhouser, T.: Semantic scene completion from a single depth image. In: CVPR (2017) 3

49. Sun, J., Shen, Z., Wang, Y., Bao, H., Zhou, X.: LoFTR: Detector-free local feature matching with transformers. CVPR (2021) 3, 10, 11
50. Sun, J., Xie, Y., Chen, L., Zhou, X., Bao, H.: Neuralrecon: Real-time coherent 3d reconstruction from monocular video. In: CVPR (2021) 3
51. Teed, Z., Deng, J.: Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. NeurIPS (2021) 3
52. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment—a modern synthesis. In: International workshop on vision algorithms (1999) 3
53. Ummenhofer, B., Zhou, H., Uhrig, J., Mayer, N., Ilg, E., Dosovitskiy, A., Brox, T.: Demon: Depth and motion network for learning monocular stereo. In: CVPR (2017) 3
54. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017) 3, 4, 6
55. Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.G.: Pixel2mesh: Generating 3d mesh models from single rgb images. In: ECCV (2018) 3
56. Wang, Q., Wang, Z., Genova, K., Srinivasan, P., Zhou, H., Barron, J.T., Martin-Brualla, R., Snavely, N., Funkhouser, T.: Ibrnet: Learning multi-view image-based rendering. In: CVPR (2021) 3
57. Wang, W., Hu, Y., Scherer, S.: Tartanvo: A generalizable learning-based vo. In: CoRL (2020) 3
58. Wang, X., Fouhey, D.F., Gupta, A.: Designing deep networks for surface normal estimation. In: CVPR (2015) 3
59. Wiles, O., Gkioxari, G., Szeliski, R., Johnson, J.: Synsin: End-to-end view synthesis from a single image. In: CVPR (2020) 3
60. Wong, S.: Takaratomy transformers henkei octane, https://live.staticflickr.com/3166/2970928056_c3b59be5ca_b.jpg 1
61. Wu, C., Clipp, B., Li, X., Frahm, J.M., Pollefeys, M.: 3d model matching with viewpoint-invariant patches (vip). In: CVPR (2008) 2
62. Yang, F., Zhou, Z.: Recovering 3d planes from a single image via convolutional neural networks. In: ECCV (2018) 3
63. Yi, K.M., Trulls, E., Ono, Y., Lepetit, V., Salzmann, M., Fua, P.: Learning to find good correspondences. In: CVPR (2018) 3
64. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelnerf: Neural radiance fields from one or few images. In: CVPR (2021) 3
65. Yu, Z., Zheng, J., Lian, D., Zhou, Z., Gao, S.: Single-image piece-wise planar 3d reconstruction via associative embedding. In: CVPR (2019) 3
66. Zhang, J., Sun, D., Luo, Z., Yao, A., Zhou, L., Shen, T., Chen, Y., Quan, L., Liao, H.: Learning two-view correspondences and geometry using order-aware network. In: ICCV (2019) 3
67. Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces. IJCV (1994) 3
68. Zhang, Z., Cole, F., Tucker, R., Freeman, W.T., Dekel, T.: Consistent depth of moving objects in video. TOG (2021) 3
69. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: ICCV (2021) 3

Supplementary Material for PlaneFormers

Samir Agarwala, Linyi Jin, Chris Rockwell, and David F. Fouhey

University of Michigan, Ann Arbor
{samirag,jinlinyi,cnris,fouhey}@umich.edu

The supplemental material consists of both video results and this PDF. The PDF portion of the supplemental material shows: detailed descriptions of model architectures (Section 1), details about the experimental setup (Section 2), and additional results (Section 3). Video results visualize qualitative reconstructions from the main paper from a variety of viewpoints.

A Model Architecture

Table 1. Model Architecture. We define the number of planes from view i be M_i, $M = M_1 + M_2$, dimension $D = 899$. Embeddings are passed through a 5-layer transformer encoder which has 1 head, dropout probability of 0.1 and a feedforward network dimension of 2048. We create a pair-wise feature tensor of dimension $M \times M \times 4D$ and pass this tensor through 4 separate MLP heads to estimate the plane correspondence, camera correspondence, rotation residual and translation residual. We mask out entries in the MLP outputs such that only the pairwise predictions between planes across views are considered during average pooling in the camera correspondence and residual heads (note: the shape after masking in the table represents non-zero entries). Finally, we apply a sigmoid function to the plane correspondence and the camera correspondence scores, and extract plane correspondences across views.

Index	Inputs	Operation	Output Shape
(1)	Inputs	Input Embedding	$M \times D$
(2)	(1)	5-Layer Transformer Encoder	$M \times D$
(3)	(2)	Create Pair-wise Feature Tensor	$M \times M \times 4D$
(4)	(3)	Plane Correspondence: Linear($4D \to 2D$), Linear($2D \to D$), Linear($D \to D/2$), Linear($D/2 \to D/4$), Linear($D/4 \to 1$), Sigmoid($M \times M$), Extract Submatrix($M \times M \to M_1 \times M_2$)	$M_1 \times M_2$
(5)	(3)	Camera Correspondence: Linear($4D \to 2D$), Linear($2D \to D$), Linear($D \to D/2$), Linear($D/2 \to D/4$), Linear($D/4 \to 1$), Mask Matrix($M \times M \to M_1 \times M_2$), AveragePool($M_1 \times M_2 \to 1$), Sigmoid(1)	1
(6)	(3)	Rotation Residual: Linear($4D \to 2D$), Linear($2D \to D$), Linear($D \to D/2$), Linear($D/2 \to D/4$), Linear($D/4 \to 4$), Mask Matrix($M \times M \times 4 \to M_1 \times M_2 \times 4$), AveragePool($M_1 \times M_2 \times 4 \to 4$)	4
(7)	(3)	Translation Residual: Linear($4D \to 2D$), Linear($2D \to D$), Linear($D \to D/2$), Linear($D/2 \to D/4$), Linear($D/4 \to 3$), Mask Matrix($M \times M \times 3 \to M_1 \times M_2 \times 3$), AveragePool($M_1 \times M_2 \times 3 \to 3$)	3

B Experimental Details

B.1 Multiview Dataset Creation

The two view dataset is the same as [24]. For 3-view and 5-view datasets, we use the single images sampled by [24], then randomly sample combinations of images within each floor of the house. We select sets of images where each image in any pair has ≥ 3 matches and ≥ 3 unique planes. The maximum number of sets per floor is 10. We finally get a three-view test set of size 258 and a five-view test set of size 76. We do not need training set or validation set for 3-view and 5-view cases since our network is not trained on the multiview dataset.

B.2 Multiview Evaluation

We compared our proposed approach with baselines on the same view graph that was built as discussed in the approach section. For multi-view evaluation, we consider all combinations of input views in a sample and independently compute the pair-wise IPAA, rotation error and translation error for each combination. For instance, in the 3-view case, we consider the IPAA and camera error metrics independently between view 1 and view 2, view 1 and view 3, and view 2 and view 3.

The relative camera transformations and plane correspondences between any combination of views is computed by chaining together relative camera transformations and plane correspondences across the created view graph, and is then compared to the ground-truth. Finally, we compute IPAA-X and camera error statistics over all combinations of views across samples in the test set (i.e. we consider each combination of views in a test sample as an independent datapoint while computing our metrics).

B.3 Ablation Details

For fair comparison, we use our same training setup for ablations. Ablations are trained until validation accuracy plateaus, which in practice is 40k iterations; the same as for the full model.

Feature Ablations. Ablating features results in a smaller input feature space. For fair comparison, we therefore use a linear layer to project this smaller input feature to features the same size as the full model. Transformer layers then operate at the same size as in the case of the full model.

Model Ablations. In our full model, we classify camera pose into clusters from [24], and predict a corrective residual camera pose to the predicted cluster. In the *without residual* ablation, we remove this residual camera pose; this tests if the corrective residual improves predicted pose. Our *without transformer* ablation takes input features as direct input to final camera and plane MLP layers, testing if the transformer improves plane features for final prediction.

C Additional Results

C.1 Quantitative Results

Additional Plane Correspondence Results We report IPAA-100 and IPAA-80 for the feature and network ablations in addition to IPAA-90 that was provided in our experiments in Tables 2 and 3, respectively.

Table 2. IPAA-100, IPAA-90 and IPAA-80 for feature ablations.

Feature	IPAA-100 ↑	IPAA-90 ↑	IPAA-80 ↑
Proposed	**19.6**	**40.6**	**71.0**
- Appearance	11.3	26.9	57.1
- Plane	16.5	35.2	65.6
- Mask	15.1	34.5	67.2

Table 3. IPAA-100, IPAA-90 and IPAA-80 for network ablations. The IPAA-X results for the model without residual are the same as the proposed method since the camera residual affects the relative camera pose prediction but not the plane correspondences.

Network	IPAA-100 ↑	IPAA-90 ↑	IPAA-80 ↑
Proposed	**19.6**	**40.6**	**71.0**
- Transformer	13.8	32.7	64.3
- Residual	19.6	40.6	71.0

Additional Relative Camera Pose Estimation Results We report median error and % error $\leq$ 1m or 30° for the predicted relative translation and rotation for the feature and network ablations in addition to mean error that was provided in our experiments in Tables 4 and 5 respectively.

C.2 Qualitative Results

Video Results. Video results bring to life the reconstructions from the main paper. As stated, PlaneFormer planar reconstructions are often quite close to the ground truth even faced with large view changes and challenging coplanar settings. If 3 or 5 views are available, the model continues to produce coherent results.

Additional Examples. We include additional outputs for 2 input views (Figure 2), and 3 and 5 views (Figure 3). These results are consistent with those in the paper: plane correspondences tend to be accurate even in challenging cases, and

Table 4. Median error, mean error and % error $\leq$ 1m or 30° for translation and rotation for the feature ablations.

Method	Translation			Rotation		
	Med.	Mean	($\leq$1m)	Med.	Mean	($\leq$30°)
Proposed	**0.66**	**1.19**	**66.8**	5.96	22.20	83.8
- Appearance	0.69	1.23	65.2	6.01	22.78	83.3
- Plane	0.81	1.32	59.6	10.34	25.92	81.4
- Mask	0.75	1.26	61.9	**4.65**	**21.21**	**84.3**

Table 5. Median error, mean error and % error $\leq$ 1m or 30° for translation and rotation for the network ablations.

Method	Translation			Rotation		
	Med.	Mean	($\leq$1m)	Med.	Mean	($\leq$30°)
Proposed	**0.66**	**1.19**	**66.8**	**5.96**	**22.20**	**83.8**
- Transformer	1.02	1.48	49	10.54	26.43	80.8
- Residual	0.88	1.34	57.7	6.22	22.38	83.7

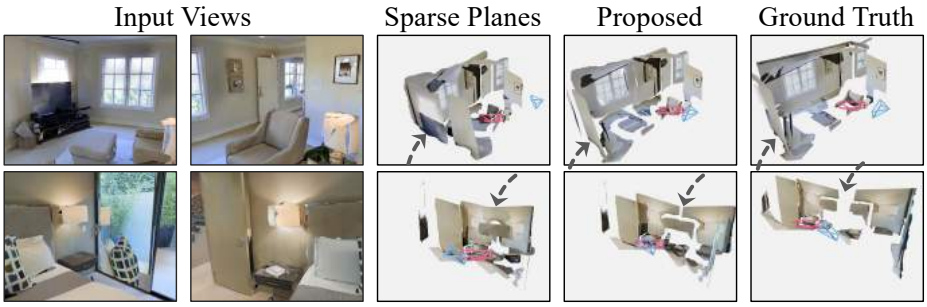

Fig. 1. Additional reconstruction comparison, extending Fig. 4 Sparse Plane reconstructions are a good baseline, but PlaneFormer yields superior results. It produces both better stitched planes (top), and more accurate camera (bottom).

reconstructions are reasonable in very large view change cases and accurate in smaller view change cases.

Additional reconstruction comparison, extending Fig. 4. See Figure 1.

Limitations and Failure Cases. We also include limitations and failure cases in Figure 4. One limitation of a plane representation is that planes struggle to model small details in scenes, which sometimes leads to incomplete reconstructions (top two examples). The model may also perform poorly in some circumstances. Plane correspondences struggle when many small, similar objects are visible across large view change (second two examples). Predicting camera can sometimes be difficult given large view change leading to significant difference in appearance (bottom two examples in two-view case). Sometimes both cam-

Fig. 2. Additional 2 View Results.

era and correspondence are poor (bottom example, two-view case). When more views are present, planes are not always fused cleanly, leading to intersections (final two examples, multi-view case).

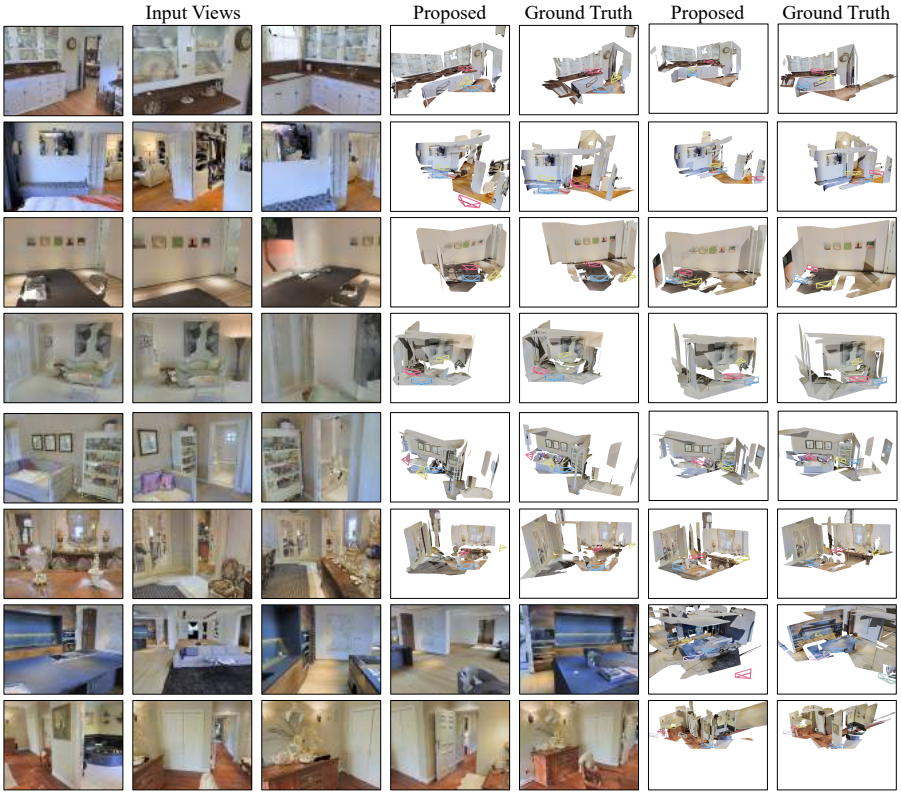

Fig. 3. Additional 3 and 5 View Results.

Fig. 4. Limitations and Failure Cases.