Bachelor's Thesis

PlaneFormers: From Sparse View Planes to 3D Reconstruction

submitted by

Samir Agarwala

Degree program	B.S.E. Computer Science
Coauthors	Linyi Jin, Chris Rockwell
Supervised by	Dr. David Fouhey
Submitted on	April 26, 2022

Acknowledgements

I am grateful to all those I have had the pleasure of working on this project over this past year. This project would not have been possible without Linyi, Chris and Prof. Fouhey's support and I am thankful to them for all their guidance and support during this project. I also want to thank Prof. Fouhey for being an understanding and supportive mentor and research advisor. I will always be grateful to him for this experience and all the support he has provided me during this time.

I would like to thank Richard Higgins and members of the Fouhey AI Lab for helpful discussions and feedback on the project. I would also like to thank Prof. Amy Cohn, Prof. Albert Berahas, Prof. Stephen Parker and several other faculty members and students at the University of Michigan for guiding me over the past few years and always being there to support me.

I would like to acknowledge the Summer Undergraduate Research in Engineering (SURE) program at the University of Michigan for their research funding for work done in Summer 2021.

Abstract

Indoor scenes such as rooms, kitchens and other areas of houses are pre-dominantly composed of planar surfaces. These surfaces may thus be used to generate high-quality scene reconstructions. Previous approaches in this domain are generally optimization-based and use an expensive bundle adjustment step. We propose the PlaneFormer, a transformer-based approach, that takes in 3D-aware plane tokens as inputs and can reconstruct scenes using planar surfaces. Our experiments show that our method outperforms baselines on most metrics, can be extended reasonably to multiple views and requires several 3D-specific choices to perform effectively on the task.

Keywords: Computer Vision, Scene Reconstruction, Sparse Views

Contents

1 Introduction

Given a set of views of a scene, humans can easily use information from images to reason about the scene's 3D layout. For instance, given the two views of a scene in Figure 1.1, humans are able to reason about how the cameras that took the two images are positioned relative to each other and find matching surfaces such as the door and cupboard to reason about the depicted scene's 3D layout. Existing computer vision systems find it challenging to do the same. In this paper, we propose a novel method, the PlaneFormer, that can reconstruct scenes using planar surfaces and thus contribute to progress in this task.

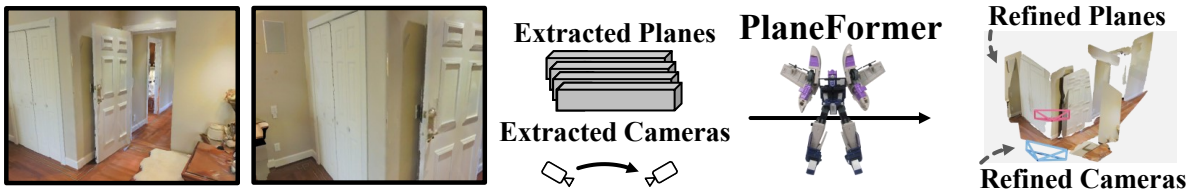

Figure 1.1: Given sparse views of a scene, our method matches extracted planes across views and improves the predicted camera pose to create a coherent scene reconstruction. Figure taken from submitted paper.

We consider the wide-baseline case in scene reconstruction. This refers to the challenging case where there are a few input views with limited overlap (i.e., there might only be a few common surfaces between the input views) and the cameras that took the images are relatively far apart. Existing computer vision systems generally focus on the task of single-view ([1]–[3]) and multiview scene reconstruction ([4]–[6]). In the single-view case, we see that learning-based models are used to create scene reconstructions but merging individual single-view reconstructions to create a scene reconstruction remains challenging. In the multiview case, we see that models often try to use triangulation in order to create scene reconstruction. However, triangulation is challenging in the wide-baseline case because of limited overlap between the input views.

In multiview work for scene reconstruction, methods often assume that they are given the relative camera pose between two images as an input which simplifies the problem. Furthermore, there is limited work in examining how such systems perform in the sparse view, wide-baseline case. Existing work in the sparse view, wide-baseline case includes [7], [8]. We see that [7] uses a complex RANSAC-like search to find

object correspondences and camera pose, while [8] uses a hand-designed optimization problem which includes an expensive bundle adjustment step where viewpoint invariant features [9] are used while optimizing the camera pose and plane parameter predictions.

We propose a simpler transformer-based approach, the PlaneFormer, which takes in 3D-aware plane tokens from planes across two-views as an input and outputs planar correspondences across the views and a correction to the initial camera pose estimate. The planar correspondences and the refined relative camera pose can then be used to generate a scene reconstruction. Our method can be extended to multiple views and remains competitive against baselines.

We train and evaluate the effectiveness of our model on the Matterport3D dataset [10] which contains RGB-D scans contains of indoor scenes. The sets of images used during training and evaluation have a wide-baseline (mean 53° rotation, 2.3m translation, 21% overlap). Across multiple metrics, our experiments show that our proposed method outperforms the state-of-art before its bundle adjustment step [8]. The proportion of predicted camera translation within 1m of the ground-truth increases from 56.5% to 66.8% before the expensive continuous optimization step in [8] and the image pairs where more than 90% of planes are associated correctly increases from 28.1% to 40.6% compared to [8]. Our method remains competitive against [8] even after its continuous optimization step.

2 Approach

In our approach, we take in two images with an unknown relationship as an input and output a full planar reconstruction of the underlying scene. In order to reconstruct a scene, we need to detect planes in each image, find correspondences between planes across images and estimate the relative camera pose transformation between images that helps us understand how the images are related to each other.

The main contribution of this paper is the PlaneFormer, a standard transformer-encoder, that can jointly reason about planes across a pair of input images to find planar correspondences across views and predict a correction to the relative camera pose. The PlaneFormer takes in a planar reconstruction of each view as an input along with their hypothesized relative camera transformation in a world coordinate system and predicts if the camera hypothesis is correct, a residual to improve the relative camera transformation and the correspondences between planes across the input pair of images. The complete approach thus consists of two main parts namely the backbone plane detector and the PlaneFormer.

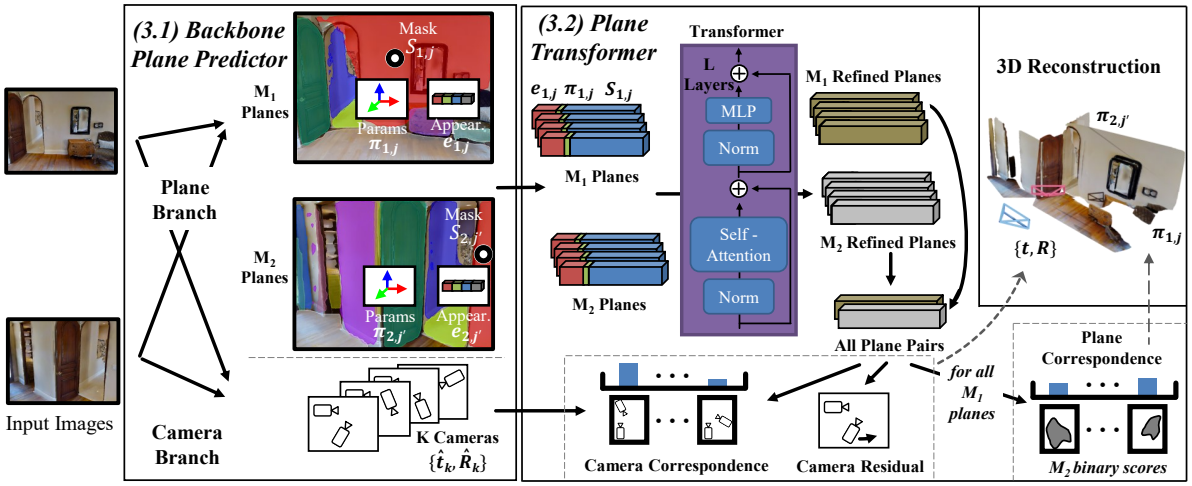

Figure 2.1: **Model Architecture.** The backbone plane detector detects planes and estimates the relative camera pose between a pair of input images. The detected planes are passed through the PlaneFormer which predicts which planes correspond across views, if the hypothesized relative camera pose is correct and how to make the relative camera pose hypothesis more accurate. Figure taken from submitted paper.

2.1 Backbone Plane Detector

The PlaneFormer is built on a backbone plane detector that detects planes independently for each input view and estimates the relative camera transformation between the pair of views. For fair comparison with the existing state-of-art and to show that improvements in performance are due to the use of our proposed approach, we use the same backbone as [8] which consists of a plane branch and a camera branch.

2.1.1 Plane Branch

The plane branch from [8] consists of a modified Plane R-CNN [1] which takes in a single image as an input and detects planes in the image. For each detected plane j in image i, the modified Plane R-CNN predicts the plane parameters $\boldsymbol{\pi}_{i,j} \in \mathbb{R}^4$, segmentation mask $\mathbf{S}_{i,j}$ and appearance feature $\mathbf{e}_{i,j} \in \mathbb{R}^{128}$. The plane parameters indicate the position and orientation of the plane in space, the segmentation masks tells us the shape and location of the plane in the input view and the appearance features summarises appearance information such as texture and color. The appearance features have been trained using the triplet loss in [8] which makes plane with similar appearances have appearance features that have a small Euclidean distance between them. For instance, if plane i' in image i and plane j' in image j have a similar appearance then the Euclidean distance between the appearance features $||\mathbf{e}_{i,i'} - \mathbf{e}j, j'||_2$ would be small.

2.1.2 Camera Branch

The camera branch from [8] takes in two images with an unknown relationship as an input and estimates the relative camera pose transformation between the two views. The relative camera pose transformation prediction (R, t) consists of two parts namely the relative rotation R and the relative translation t between the cameras that took the two views.

The relative camera pose prediction from the camera branch in [8] is in the form of independent multinomial distributions over 32 rotation and translation bins that were found by clustering on their validation set. The predicted relative camera pose is thus a product of the two distributions and gives a good estimate of the relative camera pose. It is important to note that there will likely be some error in this estimate since as noted before the camera branch tries to estimate the relative camera pose as a rotation and a translation cluster.

2.2 PlaneFormer

The main component of our approach is a plane transformer which we called the PlaneFormer. The PlaneFormer is built on top of the backbone plane detector from [8] and consists of a standard transformer encoder [11] that takes in single-view planar reconstructions from two views along with a hypothesized relative camera pose between the views. The outputs of the PlaneFormer are correspondences between planes across views, if the provided relative camera pose hypothesis is correct and an update to improve the camera pose hypothesis. The inference for our model comprises multiple forward passes of the PlaneFormer on a fixed number of camera hypotheses allowing us to select the most accurate hypothesis from our search space. A complete description of the model architecture can be found in Table 2.1.

2.2.1 PlaneFormer Inputs

The PlaneFormer operates on a set planes from a pair of input views. As seen in Figure 2.1, let us say we detect M_1 planes in image 1 and M_2 planes in image 2 for a total of $M = M_1 + M_2$ planes, and a relative camera pose estimate (R, t) between the image pairs by passing the images through the backbone plane detector. Each of the M planes is represented as a 899D token that comprises of the following features that were detected by the backbone plane detector:

1. Appearance feature ($\mathbb{R}^{128}$): The appearance feature for each plane captures how the plane looks and would give the PlaneFormer information that may allow it to discriminate between planes that are located in a similar location but have different appearances.

2. Plane parameter ($\mathbb{R}^3$): We represent the plane parameter as a 3D vector that is found by scaling the detected plane normal by the offset in the plane equation. The plane parameters of all planes input to the PlaneFormer are converted to the same world coordinate system using the relative camera pose detected from the camera branch seen in section 2.1.2. This feature tells the PlaneFormer about the location of planes in 3D and may allow the plane transformer to reason that planes with the same location and orientation in space that have similar appearance across views are likely to be the same plane (i.e. in correspondence with each other).

3. Plane Mask ($\mathbb{R}^{768}$): The plane mask is the segmentation mask of the detected plane which has been flattened into a vector. The plane masks of all planes are mapped to a common view using a standard homography transformation $H = R_i + (t_i^t n_{i,j})/o_{i,j}$ [12] so that the model can see the positions, size and shape of all planes from a common view and use that while reasoning about plane correspondence and the camera pose update. The plane segmentation mask

in the common reference frame is downsampled to a 24×32 image and then flattened into a $768D$ vector which is included in the plane token provided to the PlaneFormer.

2.2.2 PlaneFormer Outputs

The PlaneFormer operates on the M input planes across a pair of input images and predicts the following:

1. Plane Correspondence ($\mathbf{\Pi} \in \mathbb{R}^{M \times M}$): This matrix gives a score between each possible pair of planes across the input images with planes that are likely to be in correspondence having a higher pair-wise score. The plane correspondence head is trained using a binary cross-entropy loss.

2. Camera Correspondence ($\mathbf{C} \in \mathbb{R}$): This value tells us if the PlaneFormer believes that the hypothesized relative camera pose input to the model is correct or not. A high value indicates that the model is confident that the working hypothesis seems to be accurate while a low value would indicate that it is likely that the input relative camera pose hypothesis is not a good estimate. The camera correspondence head is trained using a binary cross-entropy loss.

3. Camera Residual ($\mathbf{\Delta} \in \mathbb{R}^7$): The camera residual comprises of an update to the relative rotation and translation. The rotation residual ($\mathbf{\Delta}_R \in \mathbb{R}^4$) and translation residual ($\mathbf{\Delta}_t \in \mathbb{R}^3$) are added to the relative camera pose hypothesis provided to the PlaneFormer in order to make them more accurate after considering the context provided by the plane tokens. The camera residual is trained using a L1 loss.

2.2.3 Model Architecture

The backbone plane detector gives us $M = M_1 + M_2$ planes from a pair of input views and a relative camera pose hypothesis. We build plane tokens for each of the detected planes as discussed in Section 2.2.1. The M input planes are passed through a 5-layer transformer encoder which has 1 head, dropout probability of 0.1 and a feedforward network dimension of 2048. Using the output of the transformer, we create a pair-wise feature tensor of dimension $M \times M \times 4D$ and pass this tensor through 4 separate multi-layer perceptron (MLP) heads that estimate plane correspondence, camera correspondence, rotation residual and translation residual. We then mask out entries in the MLP outputs such that only pairwise predictions between planes across views are considered during average pooling in the camera correspondence and camera residual heads. Finally, we apply a sigmoid function to the plane correspondence and camera correspondence, and extract planes correspondences across views. A detailed

Table 2.1: **Model Architecture**. We define the number of planes from view i to be M_i, $M = M_1 + M_2$ and dimension $D = 899$. The table describes the inputs, operations and outputs of different parts of our model (note: the shape after masking in the table represents non-zero entries). Table and caption taken from submitted paper.

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

description of the architecture with input and output shapes can be found in Table 2.1. Text taken from submitted paper.

2.2.4 Model Training

The backbone plane predictor is taken from [8] and thus does not need to be trained for our approach. The PlaneFormer model is trained on pairs of input views from the same dataset as [8]. We train on balanced batches that contain an equal number of correct and incorrect camera hypothesis. A camera hypothesis is considered correct if it consists of the closest rotation and translation cluster from the codebook of [8]'s

camera branch to the ground-truth relative camera pose. This helps ensure a balanced training for the camera correspondence head of our model. Losses are updated for the plane correspondence head and the camera residual head only when a training sample uses a correct camera hypothesis to ensure that training is not affected by spurious relationships arising from incorrect camera pose.

We train the model for 40,000 iterations using a batch size of 40, stochastic gradient descent with momentum of 0.9 and learning rate of 0.01 as the optimizer and a cosine annealing learning rate decay schedule. All losses are weighted equally except the translation residual loss which has a weight of 0.5. The weights of the losses were set such that all the individual losses are on a similar scale at the beginning of training. The model takes about 36 hours to train on 4 RTX 2080 Ti GPUs.

2.2.5 Model Inference

After the training procedure, the PlaneFormer model can be used to reconstruct scenes in two main cases namely the two-view case and the multiview case.

Two-View Case

Given two views of a scene, we detect planes in each image and estimate the relative camera pose using the backbone from [8]. We take the top $h = 9$ relative camera pose hypotheses from [8]'s camera branch and run the PlaneFormer on each of the h camera hypotheses. We choose the hypothesis with the highest camera correspondence score and use it for scene reconstruction. We update the camera pose from the selected hypothesis using the predicted camera residuals and get binary planar correspondences after applying the Hungarian algorithm with thresholding on the planar correspondence matrix. The binary planar correspondences are used to merge corresponding planes. Finally, the updated camera pose and the merged planes allow us to generate a planar scene reconstruction of the two input views.

Multiview Case

Given more than two views of a scene, we first make an acyclic view graph connecting all the views. For the multiview case, we apply the two-view approach on each edge in the graph and generate a scene reconstruction.

The view graph is greedily created using a connectivity score that we calculate between each pair of views. This connectivity score represents the number of planes that seem to correspond together across two views based on the appearance features. We compute the score $d_j = \min_{j'} ||\mathbf{e}_{i,j} - \mathbf{e}_{i',j'}||$ which represents the minimum distance from the appearance feature of plane j in image i to any plane in image i'. We then accumulate this score across all planes in image i to get $\sum_j d_j$ and repeat this process

from view i′ to i to get $\sum_{j'} d_{j'}$ and sum the two to get a symmetric score between views i and i′.

3 Experiments

We want to see how the PlaneFormer performs in the wide-baseline case for scene reconstruction. We thus ran experiments to evaluate how our approach performs in the wide-baseline two view case and the wide-baseline multiview case and compare it to existing methods. To gain insight into how the different parts of our plane tokens and network contribute to 3D reasoning by our model, we also performed a feature and network ablation study. We now introduce the metrics, datasets and baselines we use in our experiments.

Metrics To evaluate our model in these settings, it is important to define metrics that can measure how well we perform in the various tasks we perform using our model, namely plane correspondence, relative camera pose prediction and entire scene evaluation. We use the following metrics for evaluating our model:

- Plane Correspondence: We measure plane correspondence using IPAA-X from [13]. This metric measures the percentage of image pairs where the model gets at least X% of plane correspondences correct.
- Relative Camera Pose: We measure the relative camera pose prediction from our model using mean and median error, and measure the percentage of pairs where we have a translation error of $\leq 1m$ and rotation error of $\leq 30^\circ$ similar to [8].
- Overall scene reconstruction: We measure overall scene reconstruction using average precision (AP). Similar to [8], we consider true positive planes to be planes whose mask IoU is ≥ 0.5, surface normal distance is $\leq 30^\circ$ and offset distance $\leq 1m$. This metric thus measures if all detected planes in the scene have been predicted correctly and are close to the ground-truth planes, and accounts for factors including plane detection quality, relative camera pose prediction and plane correspondences.

Datasets For evaluation, in the two-view setting we use the same dataset as [8]. For evaluation in the 3- and 5-view cases, we generate a wide baseline dataset using the same procedure as [8]. The two-view dataset consists of 31392 training image pairs, 4707 validation image pairs and 7996 test image pairs. The three-view dataset contains 258 test image sets and the five-view dataset contains 76 test image sets. The images in these datasets have a wide-baseline with a mean overlap of 21% pixels, relative rotation of 53° and relative translation of 2.3m.

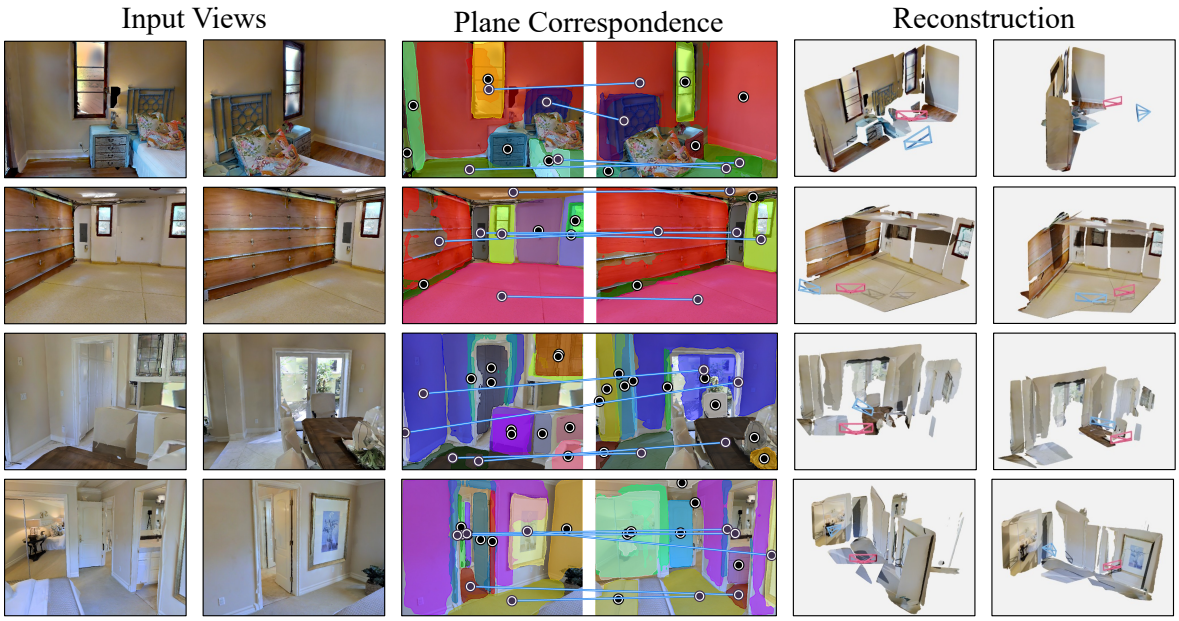

Figure 3.1: **Sample Outputs on the Test Set.** PlaneFormer produces jointly refined plane correspondences and cameras, from which it reconstructs the input scene. It can produce high-quality reconstructions in cases of moderate view change (top 2 rows), and coherent reconstructions in cases of large view change (bottom two rows). Figure and caption taken from submitted paper.

Baselines We also use several baselines to compare our model to and evaluate its performance. We compare against the full model from [8] and also compare against [8] without its bundle adjustment step which is referred to as *No Continuous* in all settings we consider. Although the full model from [8] is our strongest baseline, we consider [8] without its continuous optimization step as a more appropriate baseline since in the continuous optimization step, [8] extracts viewpoint invariant SIFT features [9] and optimizes the scene prediction which we do not do in our method. We also consider the following baselines:

- Plane Correspondence: In addition to the Sparse Planes baseline [8], we compare against an appearance feature baseline where we compute the pairwise distance between the appearance features of detected planes and run the thresholded Hungarian algorithm on the resultant cost matrix.

- Relative Camera Pose: Apart from [8], we compare against the Camera Branch from [8]. This baseline is the most important baseline for us since any performance gain over this baseline can be attributed to the PlaneFormer since the PlaneFormer is built on top of the backbone from [8] and uses it camera correspondence and residual heads to improve the camera pose estimate from the

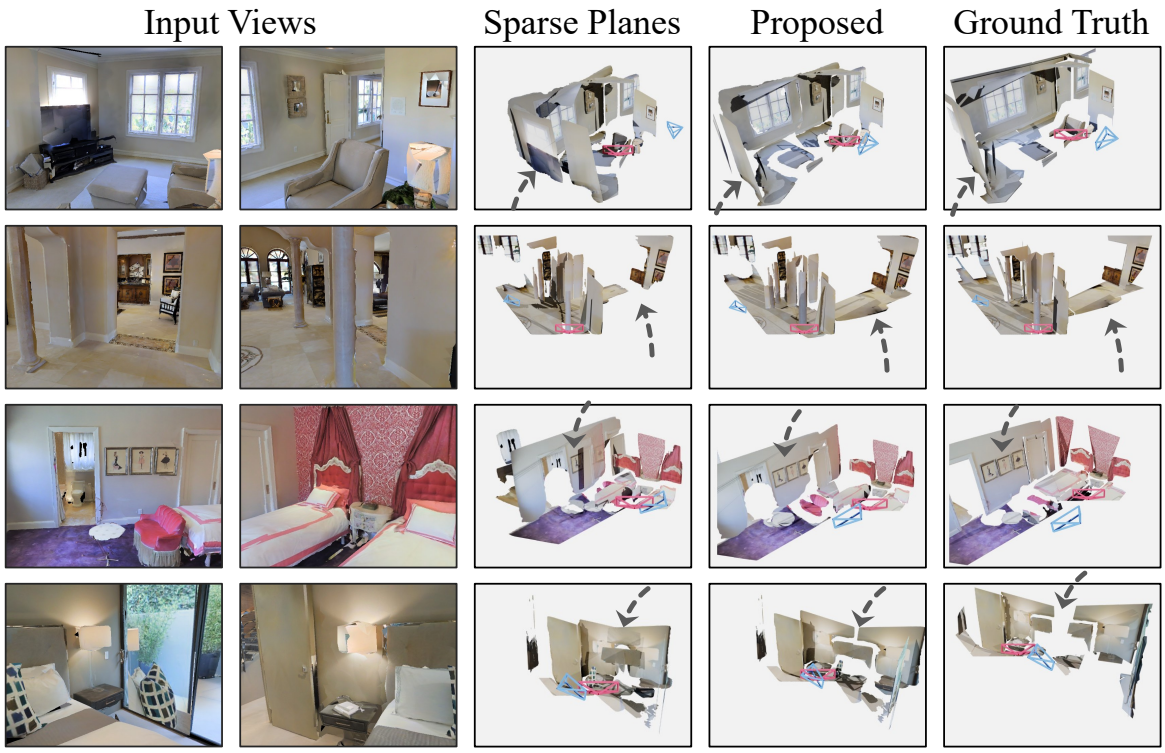

Figure 3.2: **Reconstruction Comparison.** Sparse Plane reconstructions are a good baseline, but PlaneFormer yields superior results. It produces both better stitched planes (top 2 rows), and more accurate camera (bottom two rows). Figure and caption taken from submitted paper.

camera branch. We also compare to other methods including odometry methods [14] with GT depth and with predicted depth from [15], and also to [16] which uses learned feature-matching to estimate the essential matrix. Since [16] estimates the essential matrix, it cannot predict translation scale [4].

- Full scene reconstruction: We compare against [8] and [8]'s top performing baselines which use the plane R-CNN outputs along with a relative camera pose estimation method to generate a scene reconstruction. For [16], we also provide the ground-truth translation scale to allow it to perform scene reconstruction.

3.1 Wide-Baseline Two-View Case

We show qualitative results of our approach in Figure 3.1 and also show a qualitative comparison of our approach with [8] in Figure 3.2. We compare against [8] and the top-performing baselines from [8] quantitatively. We compare each aspect of the

Table 3.1: **Two View Plane Correspondence.** IPAA-X [13] 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. Table and caption taken from submitted paper.

	IPAA-100	IPAA-90	IPAA-80
Appearance Only	6.8	23.5	55.7
Sparse Planes [8]	16.2	28.1	55.3
Proposed	**19.6**	**40.6**	**71.0**

Table 3.2: **Two View Relative Camera Pose**. We report median, mean error and % error $\leq$ 1m or 30° for translation and rotation. Table and caption taken from submitted paper.

	Translation			Rotation		
Method	Med.	Mean	($\leq$1m)	Med.	Mean	($\leq$30°)
Odometry [14] + GT Depth	3.20	3.87	16.0	50.43	55.10	40.9
Odometry [14] + [15]	3.34	4.00	8.3	50.98	57.92	29.9
Assoc. 3D [7]	2.17	2.50	14.8	42.09	52.97	38.1
Camera Branch [8]	0.90	1.40	55.5	7.65	24.57	81.9
Sparse Planes [8] (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 [8] (Full)	**0.63**	1.25	66.6	7.33	22.78	83.4
SuperGlue [16]	-	-	-	**3.88**	24.17	77.8

system including: plane correspondences, relative camera pose estimation and full scene reconstruction quality.

Plane Correspondences We report plane correspondence results in in Table 3.1. PlaneFormer substantially outperforms [8] and the appearance feature baseline across all reported IPAA-X metrics. We also evaluate plane correspondences qualitatively with [8] and show results in Figure 3.3.

Relative Camera Pose Estimation We now evaluate relative camera pose estimation from the PlaneFormer in Table 3.2. We see that our approach outperforms baselines that do not use bundle adjustment across all metrics. When methods use bundle adjustment such as [8] with its continuous optimization step, we still perform competitively. Our

Table 3.3: **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 $\leq 30^{\circ}$, and Offset error ≤ 1m. (**-Offset**) removes the offset condition; (**-Normal**) removes the normal condition. Table and caption taken from submitted paper.

Methods	All	-Offset	-Normal
Odometry [14] + PlaneRCNN [1]	21.33	27.08	24.99
SuperGlue-GT Scale [16] + PlaneRCNN [1]	30.06	33.24	33.52
Camera Branch [8] + PlaneRCNN [1]	29.44	35.25	31.67
Sparse Planes [8] (No Continuous)	35.87	42.13	38.8
Proposed	**37.62**	**43.19**	**40.36**
Sparse Planes [8] (Full)	36.02	42.01	39.04

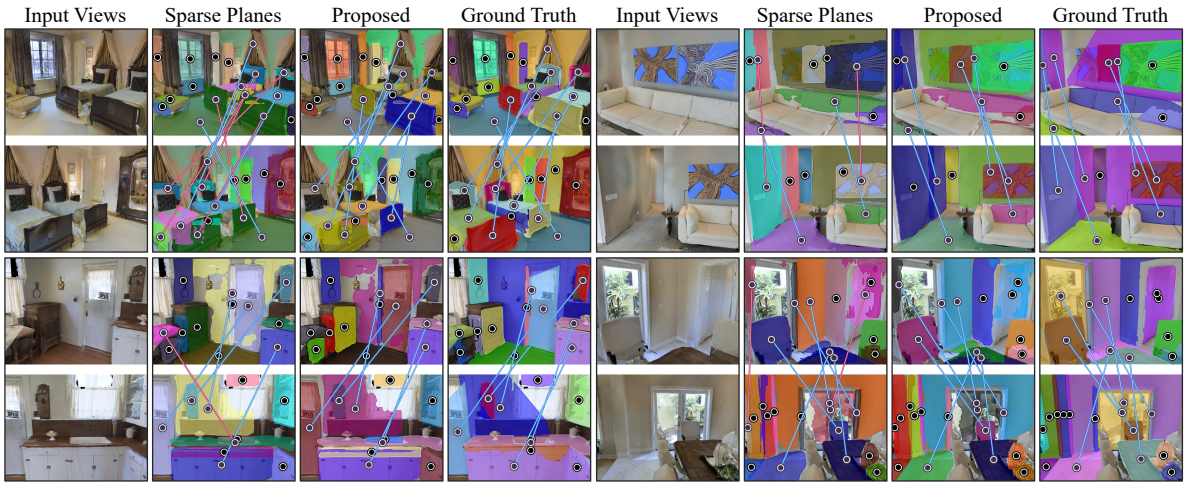

Figure 3.3: **Plane Comparison.** Matching surfaces across large view changes is challenging. Multiple surfaces may be similar in appearance, causing correspondence mixups like bed footboards (top left) or paintings (top right). By jointly refining planes across images via a transformer, the proposed method better associates across images. It can also reduce inconsistent outlier detections (bottom). Figure and caption taken from submitted paper.

method also performs competitvely against [16], which does not report a translation scale.

Full Scene Reconstruction We compare our method to the baselines reported in the relative camera pose case with respect to full scene reconstruction evaluation. We see that our method performs the best across all baselines for full scene reconstruction. This being said, we see that the relative gains in performance in AP is not as substantial

Table 3.4: **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). Table and caption taken from submitted paper.

	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 [8]	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 3.5: **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. Table and caption taken from submitted paper.

	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 [8]	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. [8]	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 [8]	0.84	**1.74**	54.91	8.83	**30.19**	75.58	1.13	**2.29**	**47.37**	11.35	44.16	64.21

as the improvement in plane correspondences as reported in Table 3.1 and this may be because AP measures all aspects of scene reconstruction and even plane detection quality from the backbone would impact the score.

3.2 Wide-Baseline Multiview Case

We evaluate our approach in the multiview case with 3 and 5 views. This task is substantially more challenging than the two-view case since the approach must be able to create coherent scene reconstruction leveraging information from all the views. We show qualitative results in the multiview case in Figure 3.4 and see that our method can generate high-quality scene reconstructions even in the multiview case.

For the multiview case, we quantiatively compare against the full method from [8] and [8] without continuous optimization. For the plane correspondence evaluation, we also report the appearance feature baseline and for relative camera pose evaluation, we report the camera branch from [8].

Similar to the two-view case, we see that our method substantially improves plane correspondences over [8] and the appearance feature baseline as seen in table 3.4. We also see that our method performs competitively with the full method from [8] and often surpasses it on a few metrics with regards to relative camera pose estimation

Figure 3.4: **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). Figure and caption taken from submitted paper.

as seen in Table 3.5. We also always improve upon the camera branch from [8] and generally perform better than [8] without its continuous optimization step.

3.3 Ablations

We performed ablation studies to see the importance of features we used and our network architecture. For the ablation studies, we report IPAA-90 as a measure of plane correspondences, and mean rotation and translation errors as a measure of the relative camera pose estimation as seen in Table 3.6. We train all ablations till validation metrics plateau for fair comparison.

Feature ablation We perform a feature ablation to test the importance of various features included in our plane tokens in Table 3.6 (left). For fair comparison, we project input features to the transformer to the same dimension using an MLP during this ablation study. We see that the appearance features seem to be the most important contributor to plane correspondences in our approach. Without appearance information a system might find it hard to match planes at similar locations that look very different from one another. We see that the biggest contributor to the relative camera pose estimates are the plane parameters, and thus the position and orientation of planes seems to be a very useful cue to the network in determining relative camera pose between images. Lastly, we see that the mask features contribute to plane correspondences but

Table 3.6: **Ablations.** We perform ablations of input features (left) and network design (right). We report IPAA-90 and relative camera pose translation and rotation error. Table and caption taken from submitted paper.

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

does not impact camera estimates as much. We see that the overall model that uses all the features performs the best in the ablation.

Network ablation We perform a network ablation to test the importance of the various components of our network in Table 3.6 (right). We see that allowing the planes to interact with each other through the transformer network contributes substantially to performance for both plane correspondence and relative camera pose estimates. We also see that the camera residual contributes to substantial improvements in the relative camera pose estimates. Thus, we see that the proposed network with all its components is necessary in effectively estimating planar correspondences and relative camera pose.

4 Conclusion

We propose the PlaneFormer, a transformer-based approach, that can reconstruct scenes given images separated by a wide-baseline (i.e. limited overlap). Our method performs better than the state-of-art [8] across multiple metrics while not using complex optimization steps or bundle adjustment. Our method remains competitive against the state-of-art [8] or often surpasses it even when it does bundle adjustment. We also extend our method to work on multiple views and evaluate it on 3- and 5-views, and see that our method continues to often generate high quality scene reconstructions while remaining competitive on plane correspondence and relative camera pose estimation metrics in the multiview setting.

Bibliography

[1] C. Liu, K. Kim, J. Gu, Y. Furukawa, and J. Kautz, "Planercnn: 3d plane detection and reconstruction from a single image," *CoRR*, vol. abs/1812.04072, 2018. arXiv: 1812.04072.

[2] D. Hoiem, A. Efros, and M. Hebert, "Geometric context from a single image," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 1, 2005, 654–661 Vol. 1.

[3] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," English (US), in *2015 International Conference on Computer Vision, ICCV 2015*, ser. Proceedings of the IEEE International Conference on Computer Vision, Publisher Copyright: © 2015 IEEE. Copyright: Copyright 2017 Elsevier B.V., All rights reserved.; 15th IEEE International Conference on Computer Vision, ICCV 2015 ; Conference date: 11-12-2015 Through 18-12-2015, Institute of Electrical and Electronics Engineers Inc., Feb. 2015, pp. 2650–2658.

[4] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Second. Cambridge University Press, ISBN: 0521540518, 2004.

[5] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4104–4113.

[6] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Cham: Springer International Publishing, 2016, pp. 501–518.

[7] S. Qian, L. Jin, and D. F. Fouhey, "Associative3d: Volumetric reconstruction from sparse views," *CoRR*, vol. abs/2007.13727, 2020. arXiv: 2007.13727.

[8] L. Jin, S. Qian, A. Owens, and D. F. Fouhey, "Planar surface reconstruction from sparse views," *CoRR*, vol. abs/2103.14644, 2021. arXiv: 2103.14644.

[9] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, Ieee, vol. 2, 1999, pp. 1150–1157.

[10] A. X. Chang, A. Dai, T. A. Funkhouser, M. Halber, M. Nießner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from RGB-D data in indoor environments," *CoRR*, vol. abs/1709.06158, 2017. arXiv: 1709.06158.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. arXiv: 1706.03762.

[12] Y. Ma, S. Soatto, J. Košecká, and S. Sastry, *An invitation to 3-d vision: from images to geometric models*. Springer, 2004, vol. 26.

[13] Z. Cai, J. Zhang, D. Ren, C. Yu, H. Zhao, S. Yi, C. K. Yeo, and C. C. Loy, "Messytable: Instance association in multiple camera views," *CoRR*, vol. abs/2007.14878, 2020. arXiv: 2007.14878.

[14] C. Raposo, M. Lourenço, M. Antunes, and J. P. Barreto, "Plane-based odometry using an rgb-d camera.," in *BMVC*, 2013.

[15] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *TPAMI*, 2020.

[16] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *CVPR*, 2020.

List of Figures

List of Tables