remote sensing

MDPI

Article

Planar Reconstruction of Indoor Scenes from Sparse Views and Relative Camera Poses

Fangli Guan [1], Jiakang Liu [1], Jianhui Zhang [1,*], Liqi Yan [1] and Ling Jiang [2,3,4]

[1] School of Computer Science, Hangzhou Dianzi University, Hangzhou 310018, China; fangli.guan@hdu.edu.cn (F.G.); liujiakang@hdu.edu.cn (J.L.); lqyan18@fudan.edu.cn (L.Y.)
[2] Anhui Province Key Laboratory of Physical Geographic Environment, Chuzhou University, Chuzhou 239000, China; ling.jiang@chzu.edu.cn
[3] Anhui Engineering Laboratory of Geo-Information Smart Sensing and Services, Chuzhou 239000, China
[4] Anhui Center for Collaborative Innovation in Geographical Information Integration and Application, Chuzhou 239000, China
* Correspondence: jh_zhang@hdu.edu.cn

Abstract: Planar reconstruction detects planar segments and deduces their 3D planar parameters (normals and offsets) from the input image; this has significant potential in the fields of digital preservation of cultural heritage, architectural design, robot navigation, intelligent transportation, and security monitoring. Existing methods mainly employ multiple-view images with limited overlap for reconstruction but lack the utilization of the relative position and rotation information between the images. To fill this gap, this paper uses two views and their relative camera pose to reconstruct indoor scene planar surfaces. Firstly, we detect plane segments with their 3D planar parameters and appearance embedding features using PlaneRCNN. Then, we transform the plane segments into a global coordinate frame using the relative camera transformation and find matched planes using the assignment algorithm. Finally, matched planes are merged by tackling a nonlinear optimization problem with a trust-region reflective minimizer. An experiment on the Matterport3D dataset demonstrates that the proposed method achieves 40.67% average precision of plane reconstruction, which is an improvement of roughly 3% over *Sparse Planes*, and it improves the IPAA-80 metric by 10% to 65.7%. This study can provide methodological support for 3D sensing and scene reconstruction in sparse view contexts.

Keywords: 3D reconstruction; 3D sensing; sparse view; planar surface

Citation: Guan, F.; Liu, J.; Zhang, J.; Yan, L.; Jiang, L. Planar Reconstruction of Indoor Scenes from Sparse Views and Relative Camera Poses. *Remote Sens.* **2024**, *16*, 1616. https://doi.org/10.3390/rs16091616

Academic Editors: Hao Li, Mingyang Zhang and Gonzalo Pajares Martinsanz

Received: 25 March 2024
Revised: 24 April 2024
Accepted: 29 April 2024
Published: 30 April 2024

1. Introduction

Planar surfaces play a crucial role in three dimension (3D) sensing technology, which is used in a variety of applications such as robot navigation [1,2], 3D reconstruction [3,4], and scene understanding [5,6]. More specifically, Dong et al. [7] locate autonomous vehicles in underground mines by combining image recognition technology and data from other sensors. Xie et al. [8] extract planar primitives from point clouds and reconstruct 3D building models based on topological relationships. Li et al. [9] decomposite 3D space by partitioning different buildings based on planar primitives. As planar surfaces provide a more compact representation than mesh-based non-planar models, they are more computationally efficient [10]. Thus, planar reconstruction has been a focus of computer vision research for years: for example, piece-wise planar reconstruction from point clouds [3] and Manhattan-world reconstruction from multiple images [11].

While traditional methods can produce multi-view geometry [12,13], they rely on correspondence for reconstruction and are limited to the part of the scene that has direct overlaps across images. In recent years, learning-based single-view 3D tools have improved a lot. They can reconstruct 3D planar surfaces from a single input image [14,15], which has inspired more scene reconstruction methods under sparse view settings [16]. However,

neither of those methods takes advantage of the relative camera pose when performing reconstruction.

To address this problem, this paper proposes a new method to produce a unified scene reconstruction from two images and their relative camera pose. With the additional input of the relative camera pose, the proposed method can overcome some of those limitations. Following existing work in this area [16,17], this study produces planar reconstruction using a three-step approach: detecting planes in each image, matching planes across images, and merging planes. Firstly, it detects planes in both images and predicts the plane mask, plane parameters (normal and offset), and appearance embedding features for each plane by a PlaneRCNN-based architecture [15]. Secondly, it matches planes between images by considering appearance and geometric features, including plane parameters projected by the relative camera pose, to find planes that appear in both images. Finally, matched planes are merged by optimizing their plane parameters to produce a unified scene reconstruction.

The major contributions of this paper are as follows:

1. We present a novel framework for 3D planar reconstruction from two sparse views and their relative camera pose based on a merging framework that combines PlaneRCNN for plane detection and a trust region reflective for optimization.
2. We present a method for matching cross-view plane segments that utilizes appearance embedding and geometric features.
3. We develop an efficient solution for reconstructing 3D planar representations of indoor scenes from sparse views and relative camera poses that achieves 40.67% average precision for reconstructing plane segments and 65.7% IPAA-80 for cross-view plane matching accuracy.

In the subsequent sections, we review previous studies (Section 2), introduce our method in detail (Section 3), and present the experimental details along with the results of comparisons (Section 4). Finally, we give our conclusions in Section 5.

2. Related Work

The proposed method for planar reconstruction in 3D space draws upon some well-suited tasks from computer vision topics, ranging from single-view or multi-view 3D reconstruction of RGB images to correspondence estimation in multi-view scenes.

Reconstruction from a single image: This is a well-known computer vision task with established methods that aims to infer 3D structures from 2D images and includes depth maps [18,19], normal maps [20,21], point clouds [22,23], meshes [24,25], voxels [26,27], and planes [28,29]. The proposed method achieves 3D reconstruction by planes, as they are often good approximations of indoor scenes [30]. PlaneNet, proposed by Liu et al. [28], solves piece-wise planar depth map reconstruction problems from images of indoor scenes with an end-to-end ML (machine learning)-based framework. While it is capable of inferring plane parameters, there is a lot of room for improvement in terms of planar segmentation. PlaneRecover, proposed by Yang et al. [29], uses an unsupervised learning approach for outdoor scenes. However, both of these methods can only cover segmentation problems with a fixed number of planar regions (10 in PlaneNet and 5 in PlaneRecover), limiting their reconstructions' expressiveness. Yu et al. [31] segment plane instances and recover 3D plane parameters from an image via a two-stage method based on associative embedding that maps pixels to an embedding space and groups them in planar regions, and their method can detect an arbitrary number of planes, but it may fail when subjected to adjacent planes with similar textures because of a lack of semantic information. PlaneRCNN, proposed by Liu et al. [15], employs a variant of Mask R-CNN [32] to detect planes with their plane parameters in a single step; then, it jointly refines all the segmentation masks using a refinement network. Even though it is trained on indoor scenes, PlaneRCNN is able to reconstruct most planes from unseen datasets regardless of their shapes and textures. The detection backbone in the proposed method is based on the PlaneRCNN architecture. In contrast to PlaneRCNN, the proposed method considers appearance and position across multiple views and can produce a unified scene reconstruction.

Multiview 3D reconstruction: While traditional methods can produce multi-view geometry [12,13], they rely on correspondence for reconstruction and are limited to the part of the scene that has direct overlaps across images. In particular, they rely on homographies or triangulation as a cue and usually fail when input images have little overlap. Other methods [33,34] utilize a plane-sweep algorithm to establish matching confidence and subsequently refine the disparity of the reference frustum based on this confidence. While some methods have proven successful under optimal Lambertian conditions [35], they tend to suffer from poor generalizability to other settings. This often leads to incomplete or incorrect reconstructions due to low surface textures, occlusion of objects, or transparency. ML-based single-view 3D tools [14,15] can produce reconstructions from single input images. Still, it is challenging to merge reconstruction between views to reconstruct one unified scene from multiple images: figuring out the position of an extracted 3D object requires joint reasoning appearance features, geometry relationship, and translation between camera poses. Xi et al. [36] enhance the capability of piece-wise planar reconstruction by multi-view regularization during the training phase, improving the consistency among multiple views by making the feature embedding more robust. PlaneMVS, proposed by Liu et al. [37], decouples plane reconstruction into a semantic plane detection branch and a plane MVS (multi-view stereo) branch and takes advantage of multi-view geometry. Its semantic plane detection branch is also based on PlaneRCNN [15].

Correspondence estimation: Estimating correspondences across images is vital for producing 3D reconstructions from multiple images [38–40]. The wide baseline stereo (WBS) matching problem formulated by Pritchett et al. [12] has received significant attention for many years [41,42]. Existing algorithms [43] and frameworks [44] have successfully matched views of planar objects with orientation differences of up to 160°. However, viewpoint orientation is not the only factor that influences the complexity of establishing geometric correspondence between images. The WxBS (wide "properties" baseline stereo) problem introduced by Mishkin et al. [45] considers more properties like illumination, geometry, appearance, and sensor type and solves the problem by repeating the matching steps on multiple local feature descriptors. There are also SLAM-based (simultaneous localization and mapping) methods like [46,47], which, unlike stereo-based ones, use bundle adjustment to estimate 3D correspondence and often rely on a sequence of RGB-D images as input. Yi et al. trained an end-to-end network by embedding global information in each data point to filter out putative sparse matches between two distinct views. Choy et al. [40] proposed a framework for pairwise 3D model registration by predicting correspondence using a convolutional network and optimizing it later. This paper takes inspiration from these methods to produce cross-image correspondence between plane segments.

Reconstruction from sparse views: Under sparse view settings, many existing methods can estimate the camera pose but do not produce 3D scene reconstruction [48,49]; other methods that do have this capability come with limitations as well. DSNeRF, proposed by Deng et al. [50], overcomes the problem of having an insufficient number of input views by training the radiance fields with readily available depth supervision. SparseNeRF, proposed by Wang et al. [51], complements the lack of 3D information in sparse views by exploiting depth priors from inaccurate depth observations produced by pre-trained depth models. However, NeRF (neural radiance field)-based methods cannot produce full scene reconstruction and only generate novel views from certain angles. In contrast, Qian et al. [16] combined multiple networks to produce a volumetric reconstruction of an indoor scene, but this requires the dataset to have watertight ground-truth meshes and only works under synthetic environments wherein only a few objects are in the scene. Jin et al. [17] predicted planes from two views using an extended PlaneRCNN [15] architecture and jointly estimated plane correspondences with a hand-designed optimization. Agarwala et al. [52] also employed the same plane prediction architecture but used a transformer to predict the plane correspondence. However, neither of these takes advantage of

the relative camera pose between images. The proposed method augments plane features with the relative camera pose to improve the accuracy of the reconstructed scene further.

3. Methodology

The proposed method, shown in Figure 1, aims to jointly process two images with their relative camera pose and produce a unified planar reconstruction of the scene where those images were taken. This process requires two pieces of key information: the mask of the planes in the input images to fill in their textures and the correspondence among planes in each view to merge the same plane that appears in both images. This can be done with a network that predicts plane parameters and appearance embeddings, followed by a matching module that considers both the relative camera pose and appearance, and finally, the merging of matched planes via joint optimization to produce a unified scene reconstruction. The system is built upon Detectron2 [53] and PyTorch 1.17.

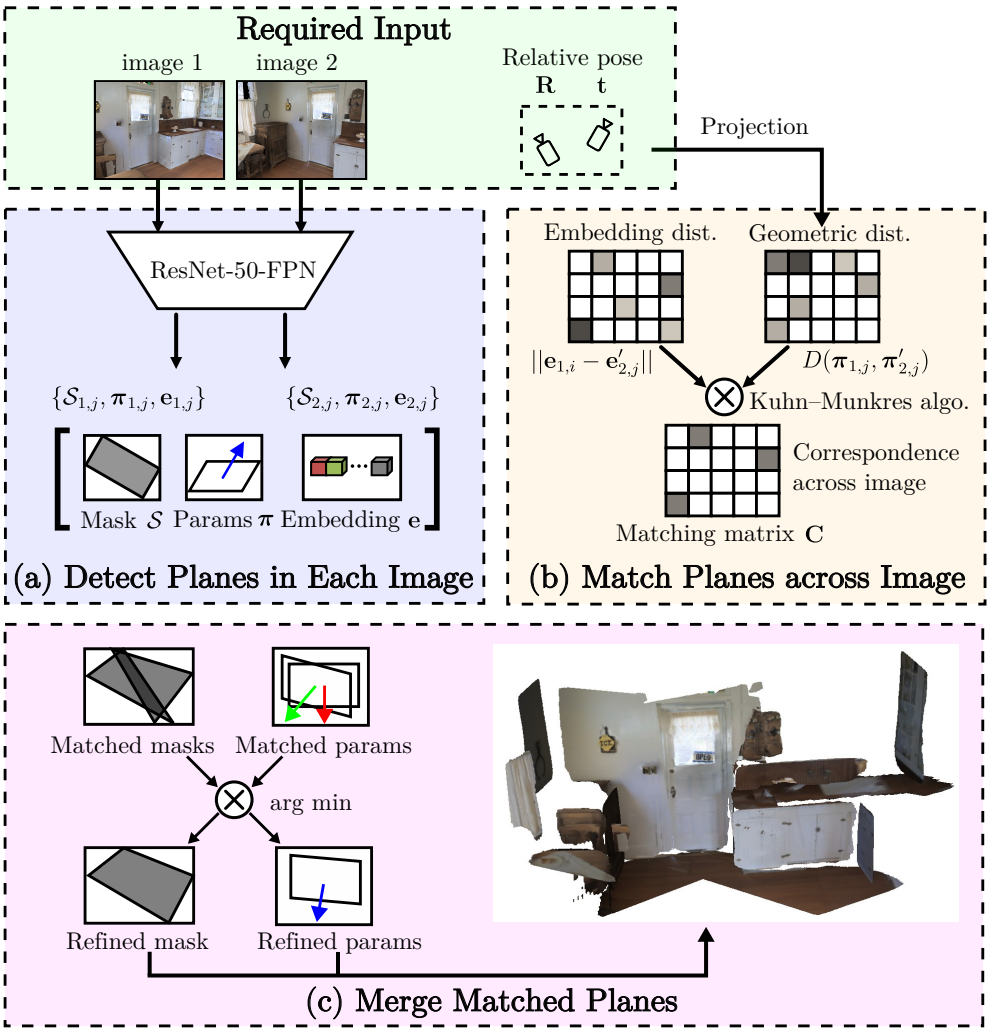


Figure 1. Pipeline of the proposed method. Given a pair of images and their relative camera pose (translation and rotation), our method uses ResNet-50-FPN to detect planes. Then, it finds planes that appear on both images and merges them further by optimizing their parameters. (**a**) The network predicts a mask, plane parameters (normal and offset), and appearance embedding feature for each plane. (**b**) It detects matching planes across images based on embedding and geometric distances transformed with the relative pose. (**c**) Finally, it performs continuous optimization to update and merge plane parameters.

3.1. Detecting Planes in Each Image

The plane detection module is built on top of a single-view plane prediction backbone from [17], which is an improved version of PlaneRCNN [15]. This module extracts planes for each image and predicts parameters for each plane. As shown in Figure 2, given the ith image, the detected plane 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 $\boldsymbol{\pi}_{i,j}$ consists of a unit vector normal $\mathbf{n}_{i,j}$ and an offset $o_{i,j}$ that defines a plane in 3D space by equation $\mathbf{n}_{i,j}^T[x, y, z] - o_{i,j} = 0$.

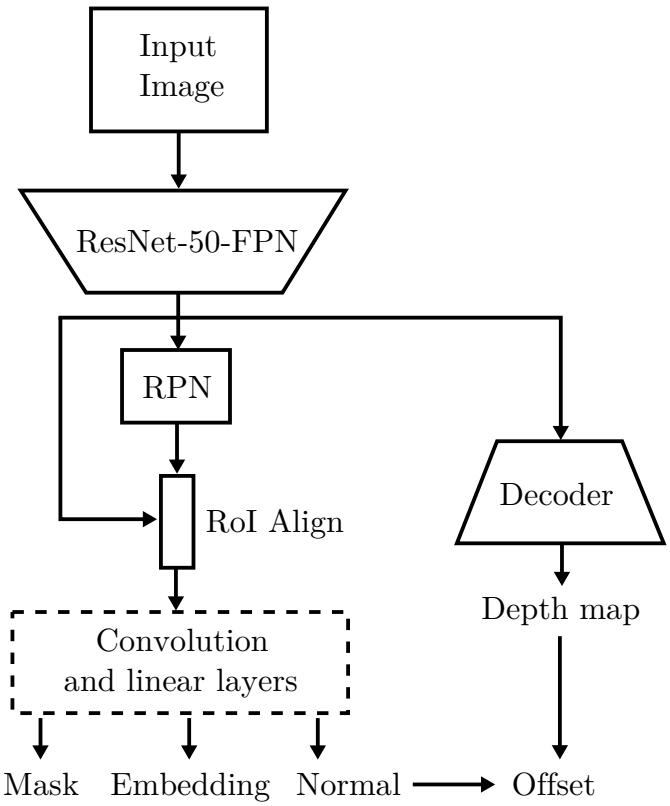


Figure 2. Overall workflow of the plane detection module.

Feature extraction: The module uses ResNet-50-FPN [54] pre-trained on COCO [55] as the backbone feature extraction layer. Given an input image, this layer produces feature maps with resolutions of $2 \times 2, 4 \times 4, 8 \times 8, 16 \times 16$, and 32×32.

Plane detection: Following PlaneRCNN, the plane extraction module is built on Mask R-CNN [32]. This module uses a region proposal network (RPN) [56] to generate a set of candidate RoIs (regions of interest) and filter out invalid candidates that cannot fit any planes in their bounding boxes using RoI align. The mask branch uses a fully convolutional network (FCN) [57] to produce semantic segmentation mask $\mathcal{S}$ for each plane instance.

Instead of predicting object categories, the normal branch predicts one of seven anchor normals from K-means clustering in 10,000 randomly sampled training images [15]. Then, it regresses the residual 3D vector to refine the normal prediction.

Additionally, the backbone features are also piped into a decoder to estimate the depth map of the entire input image with the same resolution. Starting with the smallest 2×2 feature map, the decoder upsamples it and concatenates the upsampled feature map with the next-larger feature map. After all the feature maps have been processed, a 3×3 convolution regresses the depth map and upsamples it to the image solution with bilinear interpolation.

Once the normal $\mathbf{n}$ and depth map have been determined, the plane offset o can be estimated by summing over all the pixels in the image:

$$o = \frac{\sum_i b_i(\mathbf{n}^\top(d_i K^{-1}\mathbf{x}_i))}{\sum_i b_i} \tag{1}$$

where K is the 3×3 camera-intrinsic parameter matrix, $\mathbf{x}_i$ is the ith pixel coordinate in a homogeneous representation, d_i is the predicted depth of the pixel, and indicator variable b_i becomes 1 if the pixel belongs to the plane.

Appearance embedding: The appearance embedding is produced by a network trained on pairs of images with triplet loss. Following [17], cross-view plane correspondence triplets are defined as $\mathbf{e}_a, \mathbf{e}'_p, \mathbf{e}'_n$, where anchor $\mathbf{e}_a$ corresponds to the positive match $\mathbf{e}'_p$ but not the negative match $\mathbf{e}'_n$. The standard triplet loss [58] is defined as:

$$\max(||\mathbf{e}_a - \mathbf{e}'_n||_2 - ||\mathbf{e}_a - \mathbf{e}'_p||_2 + \alpha, 0) \tag{2}$$

The anchor and positive must be closer than the anchor and negative by a margin of $\alpha = 0.2$; otherwise, it will give a loss. When training the network, online triplet mining is used to generate triplets of data points dynamically. As a result, the appearance embedding vector $\mathbf{e}_i$ can be used for cross-view matching: given images i and i' of the same scene, whenever plane j' matches plane j, the distance between their embedding vectors $||\mathbf{e}_{i,j} - \mathbf{e}_{i',j'}||$ should be small.

3.2. Matching Planes by Appearance and Relative Camera Pose

The plane matching module is designed to find every plane that appears in each input image by predicted plane parameters and appearance embedding. As input, there is a set of m planes for Image 1 $\{\mathcal{S}_{1,j}, \boldsymbol{\pi}_{1,j}, \mathbf{e}_{1,j}\}$ and a set of n planes for Image 2 $\{\mathcal{S}_{2,j}, \boldsymbol{\pi}_{2,j}, \mathbf{e}_{2,j}\}$ as well as the relative camera pose represented by a camera-to-camera transform: the 3D translation vector $\mathbf{t}$ and translation matrix $\mathbf{R}$ produced by the rotation. The translation vector $\mathbf{t} = (d_x, d_y, d_z)$ contains the translation distances of the cameras on the X, Y, Z axes, respectively. The rotation between cameras can be expressed as quaternion $q = a + bi + cj + dk$, where $i^2 = j^2 = k^2 = ijk = -1$. The translation matrix $\mathbf{R}$ an be calculated from the quaternion q to improve the efficiency of subsequent computations:

$$\mathbf{R} = \begin{bmatrix} 1 - 2c^2 - 2d^2 & 2bc - 2ad & 2ac + 2bd \\ 2bc + 2ad & 1 - 2b^2 - 2d^2 & 2cd - 2ab \\ 2bd - 2ac & 2ab + 2cd & 1 - 2b^2 - 2c^2 \end{bmatrix} \tag{3}$$

The first step is to transform the normal $\mathbf{n}_{2,j}$ and the offset $o_{2,j}$ of the plane parameters $\boldsymbol{\pi}_{2,j}$ from the second image by $\mathbf{R}$ and $\mathbf{t}$ to fit them in the same coordinates of the first image, resulting in a new set of plane segments $\boldsymbol{\pi}'_{2,j} = [\mathbf{n}'_{2,j}, o'_{2,j}]$:

$$\begin{aligned} \mathbf{P} &= \left(1 + \frac{\mathbf{t} \cdot \mathbf{R}(o_{2,j} \cdot \mathbf{n_{2,j}})}{||\mathbf{R}(o_{2,j} \cdot \mathbf{n_{2,j}})||^2}\right)\mathbf{R}(o_{2,j} \cdot \mathbf{n_{2,j}}) \\ \mathbf{n}'_{2,j} &= \mathbf{P}/||\mathbf{P}|| \\ o'_{2,j} &= ||\mathbf{P}|| \end{aligned} \tag{4}$$

The next step is to define a cross-view plane correspondence matrix $\mathbf{C} \in \{0,1\}^{m \times n}$, for which $\mathbf{C}_{i,j}$ becomes 1 only when the ith plane in Image 1 corresponds to the jth plane in Image 2. To find the closet matches, the network should maintain an $m \times n$ cost matrix $\mathbf{K}$ that encodes the quality of plane matching $\mathbf{C}$:

$$\mathbf{K}_{i,j} = \mathbf{K}_{i,j}^{\text{normal}} + \mathbf{K}_{i,j}^{\text{offset}} + \mathbf{K}_{i,j}^{\text{embedding}} \tag{5}$$

where $\mathbf{K}_{i,j}^{\text{normal}}, \mathbf{K}_{i,j}^{\text{offset}}, \mathbf{K}_{i,j}^{\text{offset}}$ encodes the costs between the plane normal vector, plane offset, and appearance embedding, respectively.

The similarity of plane normals can be encoded using cosine similarity: given two vectors, in this case, plane normal $\mathbf{n}_{1,i}$ and $\mathbf{n}_{2,j}$, the cosine similarity can be calculated as

$$\text{similarity} = \cos\langle \mathbf{n}_{1,i}, \mathbf{n}_{2,j} \rangle = \frac{\mathbf{n}_{1,i} \cdot \mathbf{n}_{2,j}}{||\mathbf{n}_{1,i}||||\mathbf{n}_{2,j}||} \tag{6}$$

Since the plane normals have already been normalized, the modulus of these vectors is always 1. The cosine similarity value ranges from -1, meaning exactly opposite, to 1, meaning exactly the same. The cost between two identical normals should be 0, so the actual cost can be calculated by subtracting the similarity from 1:

$$\mathbf{K}_{i,j}^{\text{normal}} = 1 - \mathbf{n}_{1,i} \cdot \mathbf{n}_{2,j} \tag{7}$$

Because the plane offset value is just a 1D number, the offset cost can be encoded as the difference between those two values. And the embedding cost can also be encoded as the vector distance in the embedding space:

$$\mathbf{K}_{i,j}^{\text{offset}} = |o_{1,i} - o'_{2,j}| \tag{8}$$

$$\mathbf{K}_{i,j}^{\text{embedding}} = ||\mathbf{e}_{1,i} - \mathbf{e}'_{2,j}|| \tag{9}$$

Finally, the cost $\mathbf{K}$ of cross-view plane correspondence can be encoded by combining all of those factors with a certain weight:

$$\mathbf{K}_{i,j} = k_n(1 - \mathbf{n}_{1,i} \cdot \mathbf{n}_{2,j}) + k_o|o_{1,i} - o'_{2,j}| + k_e||\mathbf{e}_{1,i} - \mathbf{e}'_{2,j}|| \tag{10}$$

where k_n, k_o, k_e are weight parameters for the normal, offset, and embedding, respectively, which are fit using a randomized search on the validation set.

According to the cost matrix, a cost value of 0 indicates a perfect match between planes, with the same normal and offset and an exact match in the embedding space. With the cost matrix determined, this matching problem can be efficiently solved using the Hungarian algorithm [59]. However, in real-world scene scenarios, not every plane in one image would match a plane in another image, so there needs to be a threshold to reject matches with large distance errors.

3.3. Merging Matched Planes

Now that the network has detected planes in both images and found matched planes across images, the final step is to align and merge matched segments into one coherent plane by refining predictions from the deep networks.

To align plane segments from input images with different view perspectives, the bounding box $b_i (i \in \{1, 2, 3, 4\})$ of the segment is first projected to 3D space, then to the normalized global frame where its normal is parallel to the viewing perspective, as demonstrated in Figure 3.

Given a plane with parameters normal $\mathbf{n}$ and offset o, point (x, y) on the plane can be projected into 3D space as point (x', y', z'):

$$d = \frac{o}{\mathbf{n} \cdot (K^{-1}[x, y, 1]^\intercal)} \tag{11}$$

$$[x', y', z']^\intercal = d \cdot (K^{-1}[x, y, 1]^\intercal) \tag{12}$$

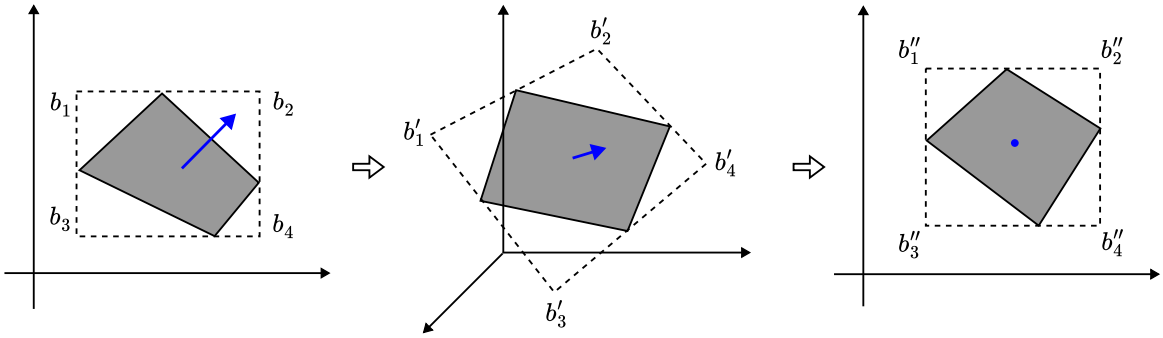
where d is the estimated depth of the point based on the plane parameters, and K is the 3×3 camera intrinsic matrix. Then, the bounding box of the segment in the global frame can be calculated with:

$$\begin{aligned} \hat{x} &= \frac{b'_2 - b'_1}{||b'_2 - b'_1||} \\ \hat{y} &= \mathbf{n}\hat{x} \\ b''_i &= [(b'_i - b'_1)\hat{x}, (b'_i - b'_1)\hat{y}], \quad i \in \{1,2,3,4\} \end{aligned} \tag{13}$$

Afterward, a 3×3 matrix $\mathbf{P}$ of the perspective transform that projects the pixels of the plane in the image to the global frame can be found:

$$\begin{bmatrix} t_i b''^{\mathsf{T}}_i \\ t_i \end{bmatrix} = \mathbf{P} \cdot \begin{bmatrix} b_1^{\mathsf{T}} \\ 1 \end{bmatrix}, \quad i \in \{1,2,3,4\} \tag{14}$$

All matched plane segments are projected to the global frame, and key points can be extracted from those projected pixels using the SIFT (scale-invariant feature transform) [60] feature. To filter out incorrect key points, a random affine transformation is applied to each pair of projected planes: collecting another set of points and transforming those points to the global frame. This process must be repeated several times, followed by a RANSAC (random sample consensus) to select the most likely key points, assuming that those features are not affected by affine transformation [60].



(a) Plane segment in image (b) 3D space (c) Normalized frame

Figure 3. Transforming the bounding box and normal of the plane fragment to the normalized global frame. Dashed box indidates bounding box, blue arrow indicates plane normal and blue dot indicates that the normal is parallel to the viewing perspective.

Since plane parameters produced by deep networks are likely to have errors between ground-truth values, and parameters of matched planes between images can also be misaligned due to said errors, it is necessary to minimize these errors before the network can produce a unified scene reconstruction.

The proposed method accomplishes this by optimizing over plane parameters $\hat{\boldsymbol{\pi}}_{1,i}, \hat{\boldsymbol{\pi}}'_{2,j}$ to minimize the geometric distance between the matched planes and the pixel offset error:

$$\underset{\hat{\boldsymbol{\pi}}_{1,i}, \hat{\boldsymbol{\pi}}'_{2,j}}{\arg\min} \sum_{i,j} \mathbf{C}_{i,j}(||\hat{\boldsymbol{\pi}}_{1,i} - \hat{\boldsymbol{\pi}}'_{2,j}|| + D_{\mathrm{k}}(\hat{\boldsymbol{\pi}}_{1,i}, \hat{\boldsymbol{\pi}}'_{2,j})) \tag{15}$$

where D_{k} measures the Euclidean distance of key points between matched planes. The pixels of those key points are back-projected to 3D to calculate point-wise Euclidean distances. The optimization uses $\boldsymbol{\pi}_{1,i}, \boldsymbol{\pi}'_{2,j}$ as initial variables and solves this non-linear problem with the trust region reflective algorithm. This algorithm is particularly suitable for large, sparse problems with bounds. The loss uses a standard least-squares function, and the solver for trust region subproblems is chosen based on the type of Jacobian returned in

the first iteration. The tolerance for termination by the change in the cost function, variables, and gradient norm are all set to 10^{-8}.

Once plane parameters are determined, it is finally time to merge matched planes across images. Masks of plane segments can be merged by aligning key points in the global frame, offsets are merged by averaging, and normals are merged by maximizing $\sum_j (\hat{\mathbf{n}}_j^\top \mathbf{n}_{i,j})^2$.

4. Experiments

The proposed method is evaluated by both qualitative and quantitative results for full-scene reconstruction and plane-matching accuracy. The evaluation is processed on a Dell PowerEdge T640 tower server with an Intel Xeon 5218R CPU and Nvidia GeForce RTX 3080Ti GPU and running Ubuntu 18.04.6 LTS. We implemented the proposed method with Python 3.7.16 and PyTorch 1.13.1, which runs on CUDA 11.7. The non-linear optimization algorithm is implemented with the `least_squares` optimization method from SciPy 1.7.3.

4.1. Dataset Description

For a fair comparison of reconstruction quality, the exact dataset from [17] is used. The dataset contains rendered views of real-world scenes from the Matterport3D [61] dataset using AI Habitat [62]. AI Habitat is a simulation platform that not only enables the rendering of realistic images but also provides ground truth depth information to be used in the evaluation process. There are 31,392 image pairs in the training set, 4707 image pairs in the validation set, and 7996 image pairs in the test set. The generated views are widely separated: about 21% of the views overlap with 53° relative rotation and 2.3 m relative translation. Figure 4 shows some random examples of image pairs in the dataset and their overlapping regions.

Figure 4. Overlapping regions from random examples of the dataset.

4.2. Scene Reconstruction Quality

The proposed method is capable of producing reconstruction of real-world scenes represented by 3D plane segments in the global coordinate frame. To begin with, the experiment compares the quality of the reconstructions with various baselines from [17].

Metrics: The scene reconstruction problem can be treated like a detection problem following other approaches that produce individual components for objects [16,63–66], and we evaluate the quality of reconstruction using the average precision (AP). Given a pair of input images with a relative pose and a set of ground-truth plane segments, the experiment evaluates how well the planes are detected and reconstructed. The metric for full scene reconstruction comes from [17] and counts each reconstructed plane as true positive whenever: (i) mask IoU (intersection-over-union) ≥ 0.5, (ii) plane normal angle $\leq 30°$, and (iii) offset distance ≤ 1 m.

Baselines: The experiment compares the proposed method with full systems like *Sparse Planes* [17] and *PlaneFormers* [52]. They both can produce scene reconstruction in the form of plane segments from image pairs but do not require the relative camera pose as input. Some results of the combination of existing systems from [17,52] are also included.

Qualitative Results: Figures 5–8 show several examples of full scene reconstruction results from two images in the Matterport3D test set. Detected plane segments are annotated with circles and colored differently, with lines indicating matching results. Prediction and ground-truth models are shown in two distinct views to contain all reconstructed plane segments in the same scene. The proposed method detects planes from two views, merges corresponding ones, and produces a unified scene reconstruction.

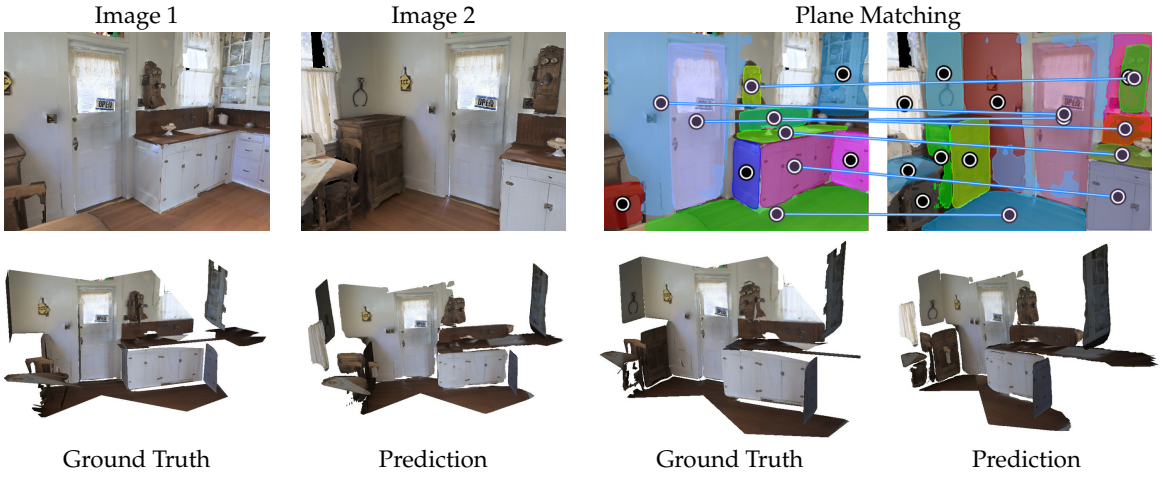


Figure 5. Qualitative result of scene reconstruction using the proposed method (Matterport 3D dataset, house id `2t7WUuJeko7`).

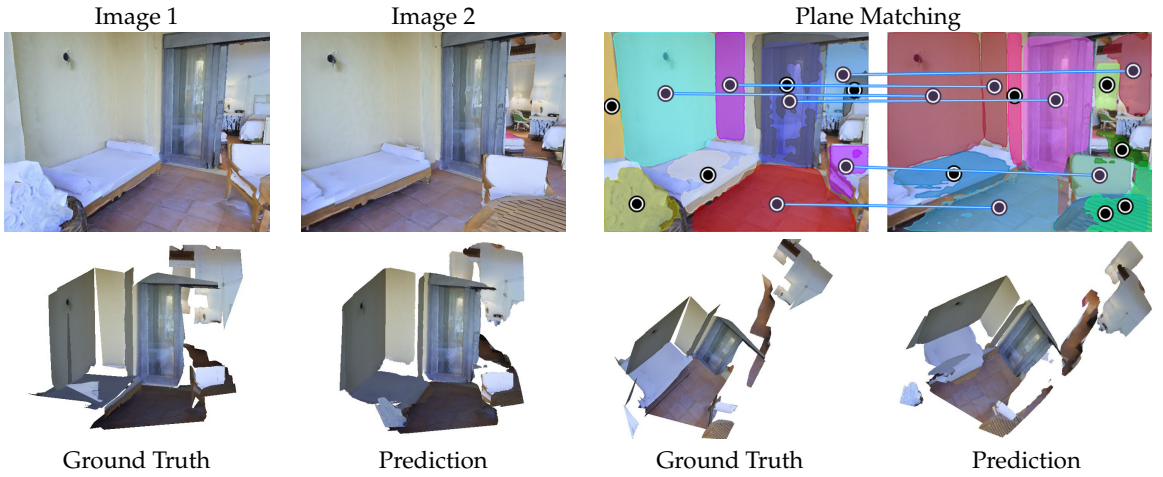


Figure 6. Qualitative result and scene reconstruction using the proposed method (Matterport 3D dataset, house id `q9vSo1VnCiC`).

Comparison between the proposed method and *Sparse Planes* [17] is presented in Figures 9–16. As shown in Figure 9, *Sparse Planes* fails to predict the continuity of the walls and leaves larger holes when reconstructing the wall, while the proposed method manages to construct a coherent indoor structure. Figure 10 demonstrates that *Sparse Planes* sometimes fails to predict the relative positions of planes, resulting in poor reconstruction quality. The reason may be that these images have a rather large relative camera rotation angle. In Figure 11, *Sparse Planes* misses some smaller planar segments among larger ones and incorrectly splits the chair next to the wall. Under larger view angles like in Figure 12, the proposed method reconstructs the wall and the floor as a connected corner, but *Sparse Planes* misplaces the floor far from the wall. For Figure 13, *Sparse Planes* produces a cluster of disconnected small pieces from a narrow segment of a far-away wall, while the proposed method constructs the wall successfully. Figure 14 shows a situation where, from different viewpoints, different objects can be seen through the door. The proposed method correctly

reconstructs the planes behind the door, while *Sparse Planes* wrongly covers the door with another plane segment next to it. Further, Figure 15 shows a rather complicated scene in which the two images have little overlap and many pieces of furniture are placed next to the wall. *Sparse Planes* does reconstruct the larger wall with paint on it but fails with the floor. The proposed method manages to connect the floor between those two images and produces a more complete scene. Last but not least, Figure 16 shows that the proposed method can predict the relative positions of smaller plane segments more precisely, while *Sparse Planes* sometimes leaves gaps between connected planes.

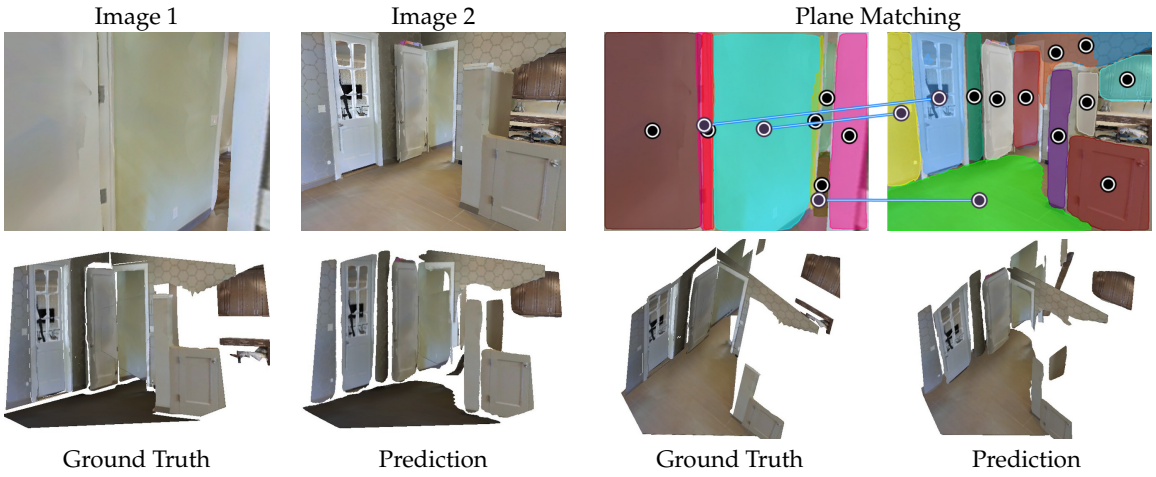


Figure 7. Qualitative result and scene reconstruction using the proposed method (Matterport 3D dataset, house id `jtcxE69GiFV`).

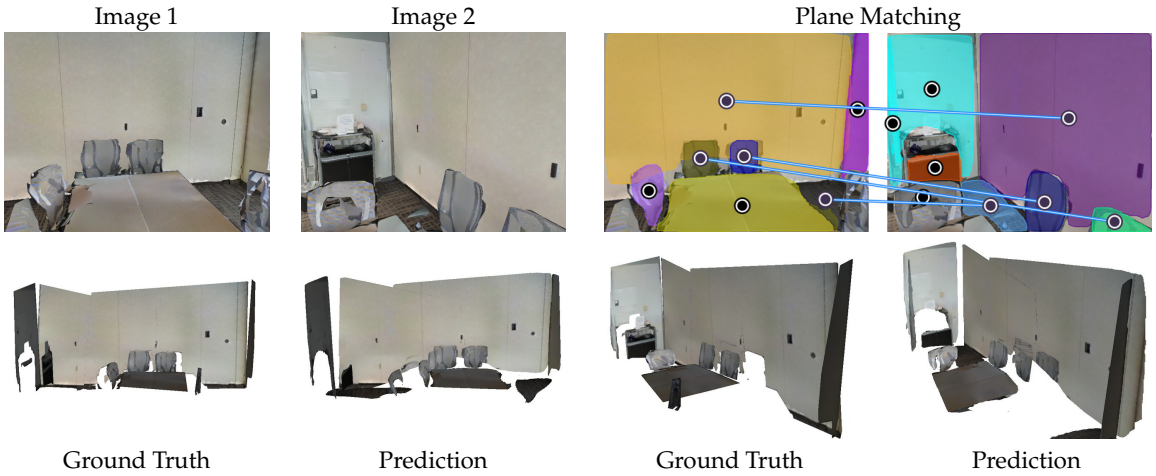


Figure 8. Qualitative result and scene reconstruction using the proposed method (Matterport 3D dataset, house id `YVUC4YcDtcY`).

Quantitative Results As shown in Table 1, the proposed method outperforms previous approaches with the help of relative camera pose input, even without optimizing plane parameters. Although *Sparse Planes* [17] does not require the relative camera pose as an input, it needs to select a camera pose from a set of 32 predefined ones, which, in turn, comes from *k*-means clustering of the training set; therefore, it can be inaccurate when the real-world camera pose does not fit in said clusters. *PlaneFormers* [52] can also produce a probability distribution over a predefined set of camera transformations then apply a transformer to the plane parameters to perform 3D reasoning, which yields better results but still fails for image pairs with uncommon camera poses. Unlike previous methods, the proposed method takes advantage of the relative camera pose to perform plane matching

across images, which makes the reconstruction of relative positions between planes more robust and, therefore, yields higher quality 3D scene reconstruction results.

Figure 9. Comparison of scene reconstruction results generated by the proposed method and Sparse Planes [17] (Matterport 3D dataset, house id `2t7WUuJeko7`).

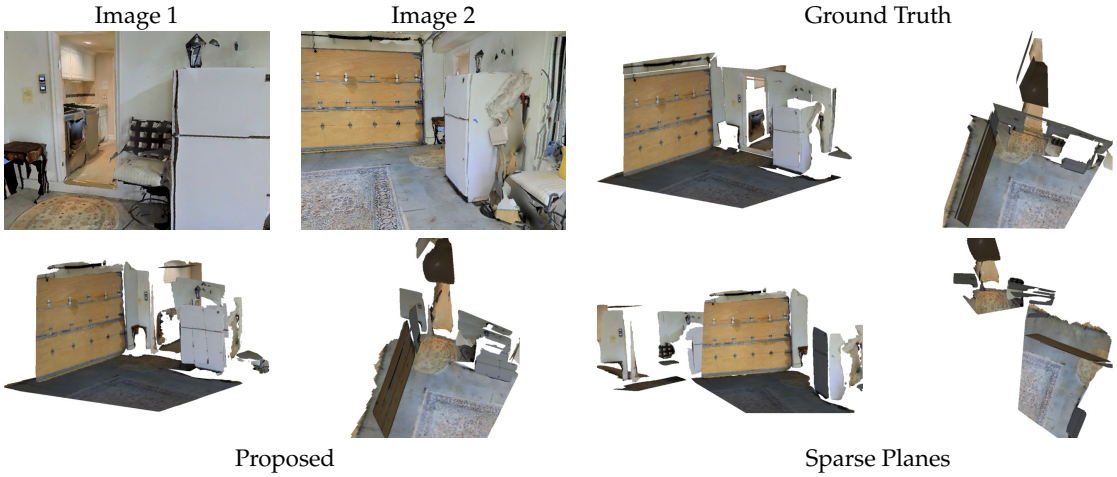


Figure 10. Comparison of scene reconstruction results generated by the proposed method and Sparse Planes [17] (Matterport 3D dataset, house id `5ZKStnWn8Zo`).

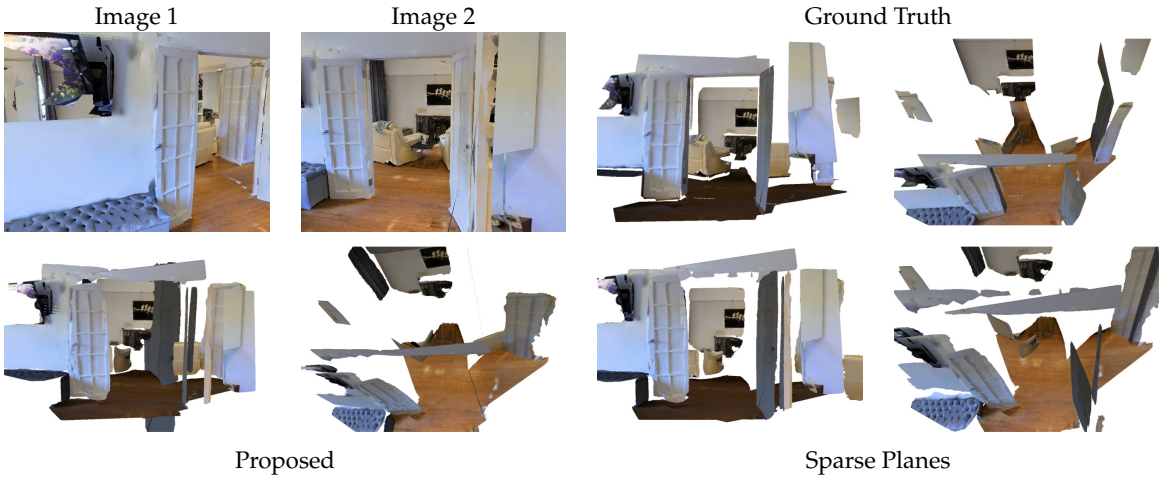


Figure 11. Comparison of scene reconstruction results generated by the proposed method and Sparse Planes [17] (Matterport 3D dataset, house id `ARNzJeq3xxb`).

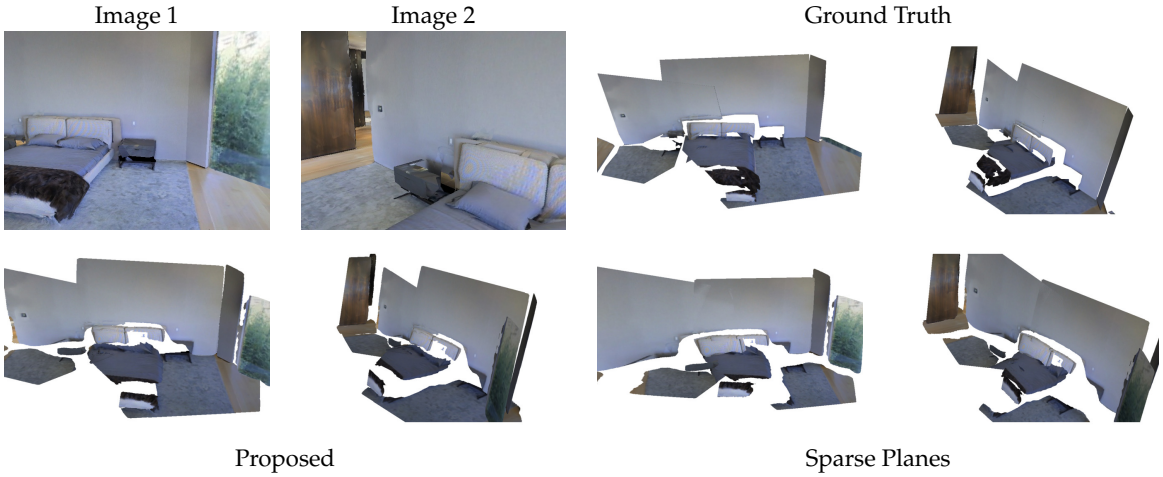


Figure 12. Comparison of scene reconstruction results generated by the proposed method and Sparse Planes [17] (Matterport 3D dataset, house id `fzynW3qQPVF`).

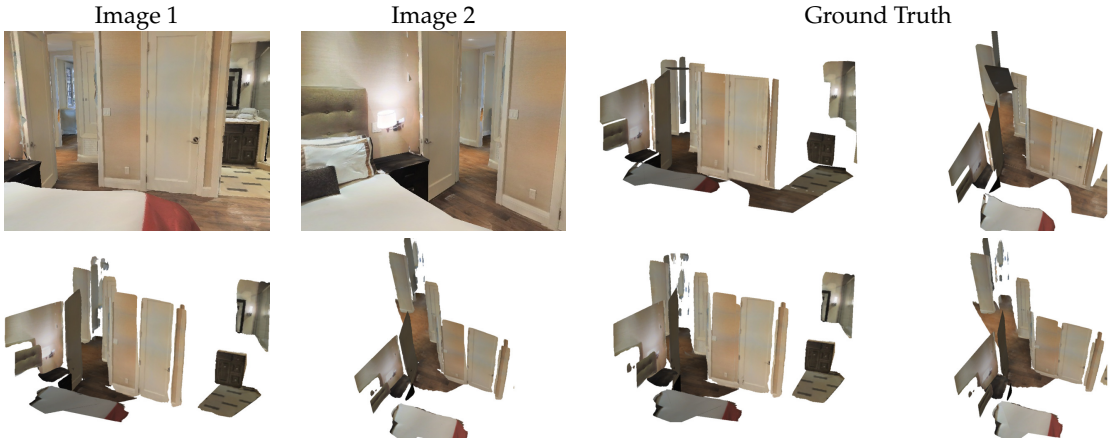


Figure 13. Comparison of scene reconstruction results generated by the proposed method and Sparse Planes [17] (Matterport 3D dataset, house id `jtcxE69GiFV`).

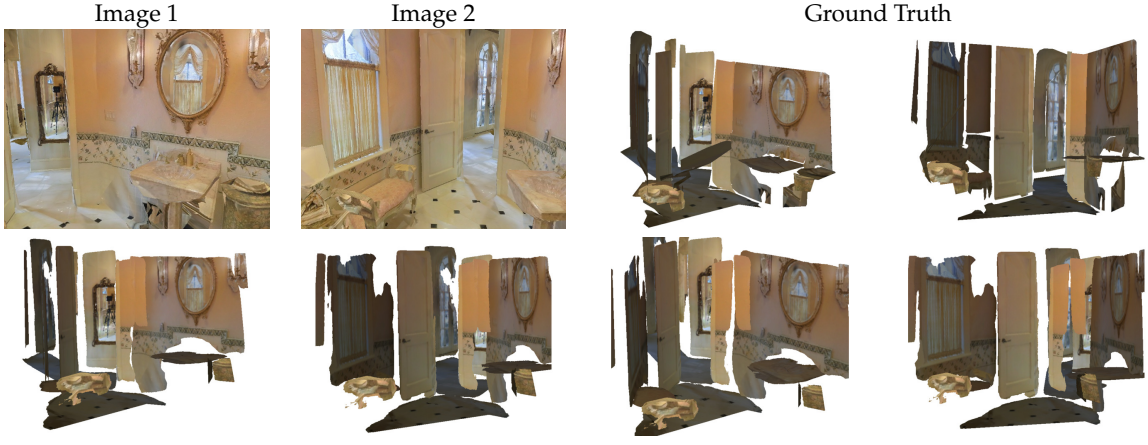


Figure 14. Comparison of scene reconstruction results generated by the proposed method and Sparse Planes [17] (Matterport 3D dataset, house id `pa4otMbVnkk`).

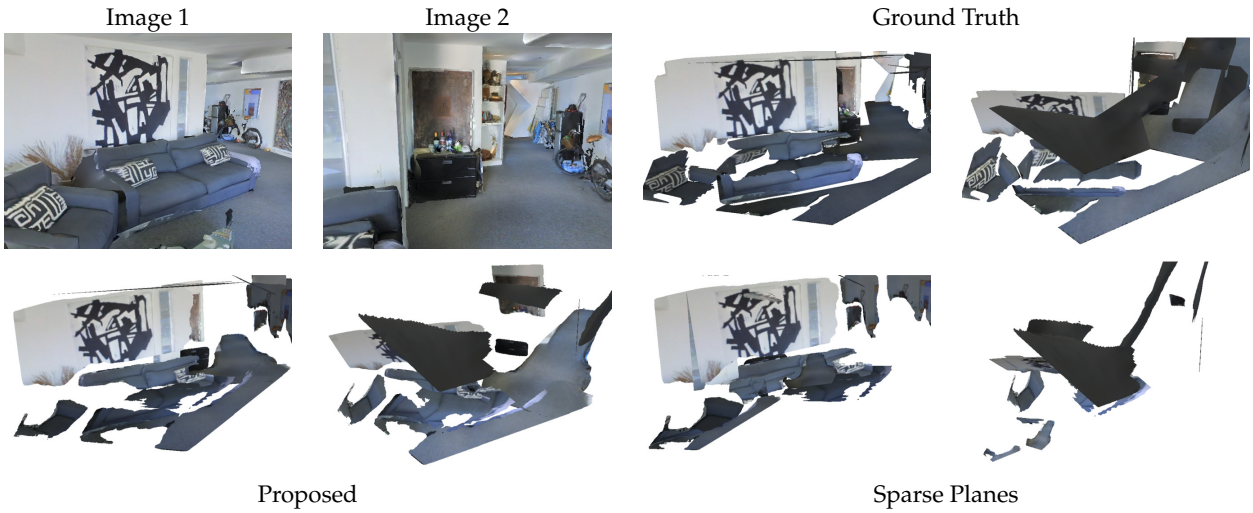


Figure 15. Comparison of scene reconstruction results generated by the proposed method and Sparse Planes [17] (Matterport 3D dataset, house id `rqfALeAoiTq`).

Additionally, the average time consumption to process an input image pair is measured in Table 2. For a fair comparison of processing speed, only established methods are included. The test split in the Matterport3D dataset contains seven planes per image and four matching planes across images on average. As previously noted above, *Sparse Planes* cannot take advantage of the relative camera pose information and needs to select a camera pose from a predefined set; it takes the longest time to process the image and produce a 3D reconstruction. Although *PlaneFormers* manages to consume less time to generate higher quality reconstruction than *Sparse Planes*, the proposed method can produce even higher quality results while only increasing the average time consumption by about 1 s. Moreover, *PlaneFormers* needs two separate models for plane detection and 3D reasoning: it requires more memory to be able to run the algorithm at all.

Figure 16. Comparison of scene reconstruction results generated by the proposed method and Sparse Planes [17] (Matterport 3D dataset, house id `UwV83HsGsw3`).

Table 1. AP (average precision), which treats plane reconstruction as plane detection in 3D space with a different definition of true positive. *All:* mask IoU ≥ 0.5; plane normal angle $\leq 30°$; offset distance ≤ 1 m. *-Offset* and *-Normal*: ignores the offset and normal condition.

Methods	All	-Offset	-Normal
Appearance Embedding Only	33.04	39.78	36.85
Associative3D [16] Optimization	33.01	39.43	35.76
Sparse Planes [17]	36.02	42.01	39.04
PlaneFormers [52]	37.62	43.19	40.36
No Optimizing Plane Parameters	39.78	45.32	41.23
Proposed	40.67	45.69	41.85

Table 2. Average time consumption (in seconds) to process an image pair in the test dataset with an average of 7 planes per image and 4 matching planes across images.

Method	Sparse Planes [17]	PlaneFormers [52]	Proposed
Time Consumption (Seconds)	8.10	5.10	6.37

4.3. Plane Matching Accuracy

One challenging point for producing a single unified scene reconstruction from two images is ensuring that every plane that appears in both images is reconstructed exactly once. So the experiment needs to evaluate how well a method can match planes across different views.

Metrics: To measure how well these methods associate planes from two images, the experiment uses the performance metric IPAA (image pair association accuracy) by Cai et al. [67] that evaluates the image-pair level matching results. Written as IPAA-X, the value represents the fraction of image pairs with not less than X% of the planes associated correctly.

Baselines: The experiment compares the proposed method with modules from other systems as baselines. These results are measured during evaluation based on ground-truth bounding boxes to eliminate the interference of plane detection. *Appearance Only* runs the optimization only with the appearance embedding feature $\mathbf{e}_{1,j}$ predicted for each plane, which matches planes based on learning-based embeddings. *Sparse Planes* [17] runs its optimization without the relative camera pose as input.

Quantitative Results: Figure 17 shows that the proposed method outperforms baselines and other systems on IPAA-X metrics. It improves IPAA-80 by nearly 10% compared to *Sparse Planes* [17]. The *Appearance Only* baseline method only considers the texture of planes when matching; it can hardly reach 100% accuracy because it is very common for different objects to have similar textures in real-world scenarios. *Sparse Planes* occasionally reaches 100% accuracy but soon falls behind when a small portion of mismatches can be tolerated. The proposed method increases the successful matching rate for all accuracy levels; these improvements should come from more planes being matched due to the proposed method being able to leverage the relative camera pose. Detailed statistics are presented in Table 3.

Qualitative Results: Comparison results are shown in Figure 18. *Appearance Only* does not take advantage of geometric information and struggles to distinguish between planes with similar textures. Without the help of the relative camera pose, *Sparse Planes* [17] sometimes fails to match plane segments with smaller surface areas. The proposed method distinguishes between planes with similar textures and matches plane segments of any size.

Table 3. Plane matching using ground-truth bounding boxes. IPAA-X [67] represents the fraction of image pairs with not less than X% of the planes associated successfully.

	IPAA-100	IPAA-90	IPAA-80
Appearance Only	6.8	23.5	55.7
Sparse Planes	16.2	28.1	55.3
Proposed	16.3	32.4	65.7

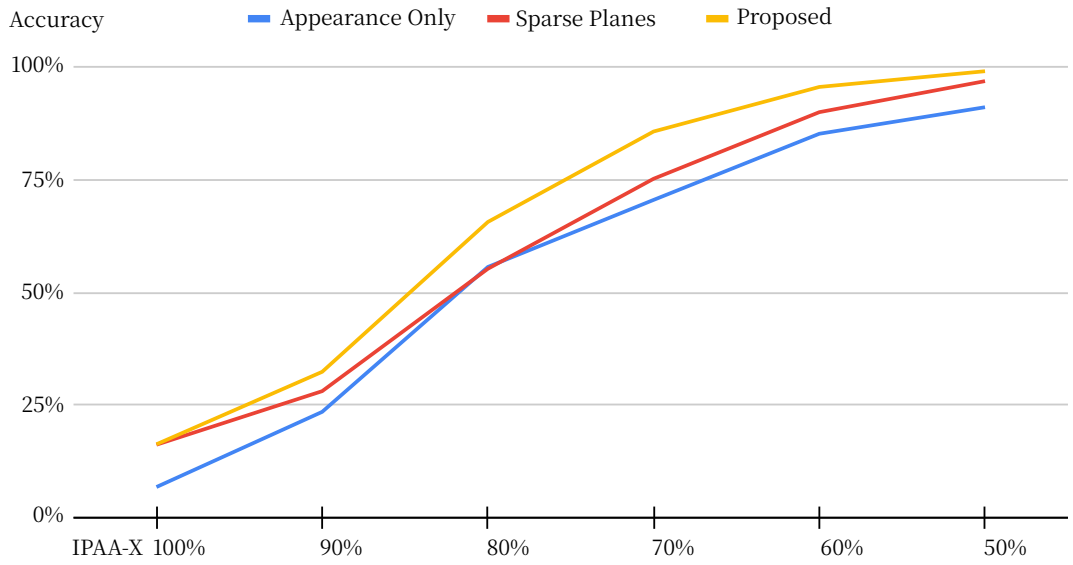


Figure 17. Comparison of IPAA-X [67] metric for plane matching accuracies of 100% to 50%.

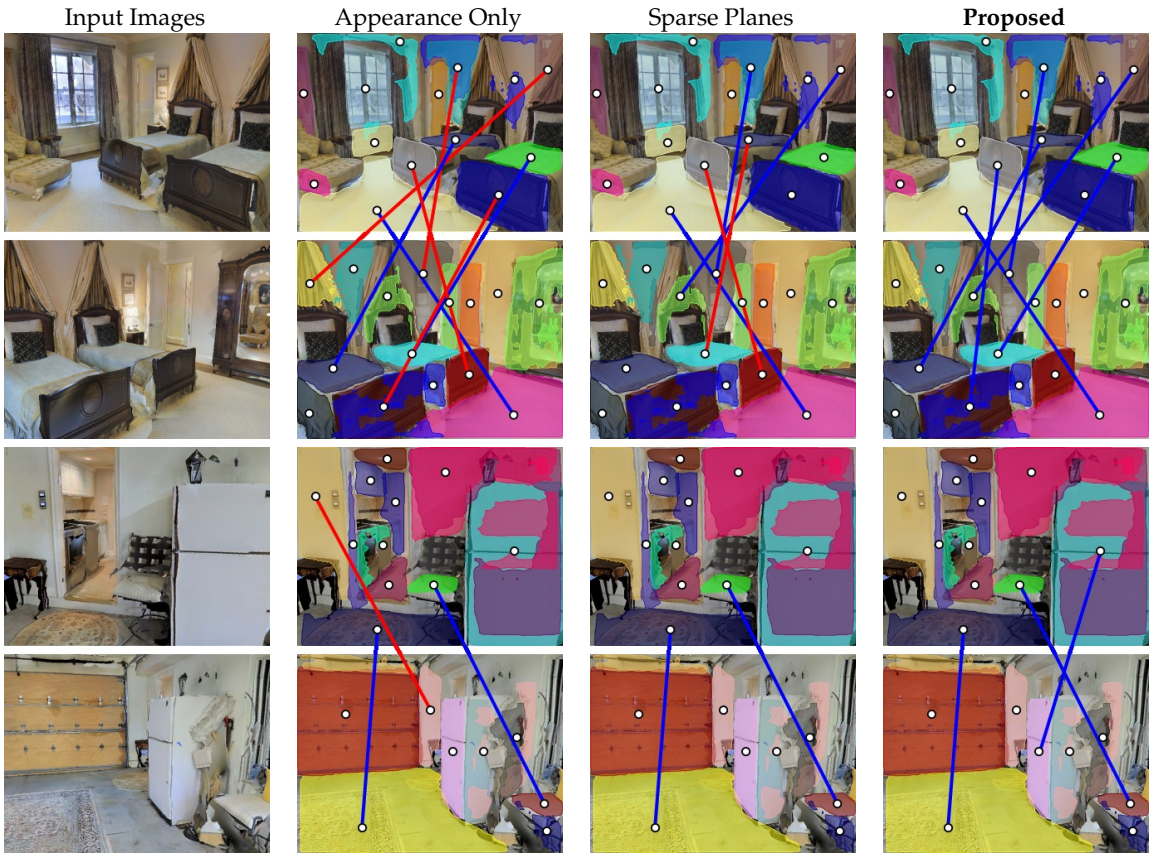


Figure 18. *Cont.*

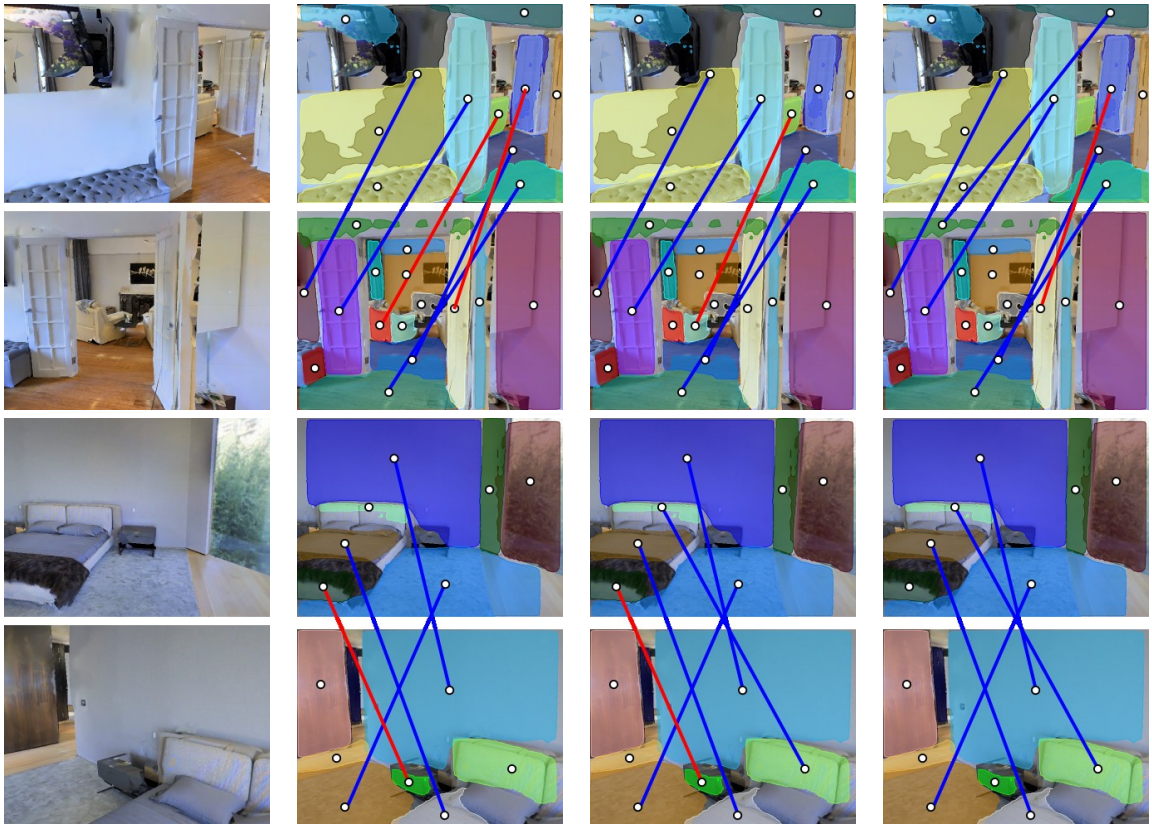


Figure 18. Plane matching comparison. Each detected plane is annotated with a circle, with blue lines indicating successful matches and red lines showing false positive cases.

5. Conclusions

This paper proposed a novel method for 3D planar reconstruction from a pair of images using the relative camera pose. Compared to existing methods, the proposed method can take advantage of the relative camera pose to optimize full-scene reconstruction results further and to improve reconstruction quality when the camera angle between input images is less common. The method also improved plane matching accuracy with the help of the relative camera pose and increased the IPAA-80 by nearly 10% over the existing system.

The proposed method can provide planar surface reconstruction for 3D sensing or scene understanding for sparse view scenarios, with fewer sensors and input data required. To improve the scene reconstruction quality, future work can embrace a more robust plane prediction backbone to provide more accurate plane segmentation masks and parameters for the plane matching and merging modules. To expand the application's scope, future work can focus on generalization and can evaluate the method on newly collected real-world datasets or more realistically rendered datasets. Last but not least, future work may also extend this architecture to a multi-view basis to reconstruct larger scenes in a progressive way.

Author Contributions: Conceptualization, F.G. and J.L.; methodology, J.L., F.G., J.Z. and L.Y.; software, J.L. and L.Y.; writing—original draft preparation, F.G. and J.L.; writing—review and editing, F.G., J.Z. and L.J.; visualization, J.L. and F.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Key R&D Program of China under grant No. 2021YFC3320301 and is supported by the Open Research Fund of the Anhui Province Key Laboratory of Physical Geographic Environment, Chuzhou University (grant No. 2023PGE001). This work is also supported by the Excellent Scientific Research and Innovation Team of Universities in Anhui Province (No. 2023AH010071), the Major Project on Natural Science Foundation of Universities in Anhui Province (No. 2022AH040156), Excellent Young Scientists Project of Universities in Anhui

Province (No. 2022AH030112), and the Academic Foundation for Top Talents in Disciplines of Anhui Universities (No. gxbjZD2022069).

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: We sincerely thank the anonymous reviewers for their critical comments and suggestions for improving the manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kaess, M. Simultaneous localization and mapping with infinite planes. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015 ; IEEE: Piscataway, NJ, USA, 2015; pp. 4605–4611.
2. Dai, A.; Lund, G.; Gao, G. PlaneSLAM: Plane-based LiDAR SLAM for Motion Planning in Structured 3D Environments. *arXiv* **2022**, arXiv:2209.08248.
3. Chauve, A.L.; Labatut, P.; Pons, J.P. Robust piecewise-planar 3D reconstruction and completion from large-scale unstructured point data. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 1261–1268.
4. Zhang, L.; Dong, H.; Saddik, A.E. From 3D sensing to printing: A survey. *ACM Trans. Multimed. Comput. Commun. Appl. (TOMM)* **2015**, *12*, 1–23. [CrossRef]
5. Tsai, G.; Xu, C.; Liu, J.; Kuipers, B. Real-time indoor scene understanding using bayesian filtering with motion cues. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 121–128.
6. Cakiroglu, O.; Wieser, V.; Zellinger, W.; Souza Ribeiro, A.; Kloihofer, W.; Kromp, F. Detection of the 3D Ground Plane from 2D Images for Distance Measurement to the Ground. In Proceedings of the International Conference on Database and Expert Systems Applications, Vienna, Austria, 22–24 August 2022; pp. 44–54.
7. Dong, L.; Sun, D.; Han, G.; Li, X.; Hu, Q.; Shu, L. Velocity-free localization of autonomous driverless vehicles in underground intelligent mines. *IEEE Trans. Veh. Technol.* **2020**, *69*, 9292–9303. [CrossRef]
8. Xie, L.; Hu, H.; Zhu, Q.; Li, X.; Tang, S.; Li, Y.; Guo, R.; Zhang, Y.; Wang, W. Combined rule-based and hypothesis-based method for building model reconstruction from photogrammetric point clouds. *Remote. Sens.* **2021**, *13*, 1107. [CrossRef]
9. Li, Y.; Wu, B. Relation-constrained 3D reconstruction of buildings in metropolitan areas from photogrammetric point clouds. *Remote. Sens.* **2021**, *13*, 129. [CrossRef]
10. Xie, Y.; Gadelha, M.; Yang, F.; Zhou, X.; Jiang, H. Planarrecon: Real-time 3d plane detection and reconstruction from posed monocular videos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 6219–6228.
11. Furukawa, Y.; Curless, B.; Seitz, S.M.; Szeliski, R. Manhattan-world stereo. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 1422–1429.
12. Pritchett, P.; Zisserman, A. Wide baseline stereo matching. In Proceedings of the Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271), Bombay, India, 4–7 January 1998; IEEE: Piscataway, NJ, USA, 1998; pp. 754–760.
13. Bao, S.Y.; Bagra, M.; Chao, Y.W.; Savarese, S. Semantic structure from motion with points, regions, and objects. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 2703–2710.
14. Eigen, D.; Fergus, R. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 2650–2658.
15. Liu, C.; Kim, K.; Gu, J.; Furukawa, Y.; Kautz, J. Planercnn: 3d plane detection and reconstruction from a single image. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4450–4459.
16. Qian, S.; Jin, L.; Fouhey, D.F. Associative3d: Volumetric reconstruction from sparse views. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Part XV 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 140–157.
17. Jin, L.; Qian, S.; Owens, A.; Fouhey, D.F. Planar surface reconstruction from sparse views. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11-17 October 2021; pp. 12991–13000.
18. Ranftl, R.; Lasinger, K.; Hafner, D.; Schindler, K.; Koltun, V. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *44*, 1623–1637. [CrossRef] [PubMed]
19. Li, Z.; Snavely, N. Megadepth: Learning single-view depth prediction from internet photos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2041–2050.
20. Wang, X.; Fouhey, D.; Gupta, A. Designing deep networks for surface normal estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 539–547.

21. 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 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, Seattle, WA, USA, 13–19 June 2020; pp. 679–688.
22. Fan, H.; Su, H.; Guibas, L.J. A point set generation network for 3d object reconstruction from a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 605–613.
23. Wiles, O.; Gkioxari, G.; Szeliski, R.; Johnson, J. Synsin: End-to-end view synthesis from a single image. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA 13–19 June 2020; pp. 7467–7477.
24. Gkioxari, G.; Malik, J.; Johnson, J. Mesh r-cnn. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27–28 October 2019; pp. 9785–9795.
25. Wang, N.; Zhang, Y.; Li, Z.; Fu, Y.; Liu, W.; Jiang, Y.G. Pixel2mesh: Generating 3d mesh models from single rgb images. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 52–67.
26. 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 Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part VIII 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 628–644.
27. Song, S.; Yu, F.; Zeng, A.; Chang, A.X.; Savva, M.; Funkhouser, T. Semantic scene completion from a single depth image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1746–1754.
28. Liu, C.; Yang, J.; Ceylan, D.; Yumer, E.; Furukawa, Y. Planenet: Piece-wise planar reconstruction from a single rgb image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2579–2588.
29. Yang, F.; Zhou, Z. Recovering 3d planes from a single image via convolutional neural networks. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 85–100.
30. Ataer-Cansizoglu, E.; Taguchi, Y.; Ramalingam, S.; Garaas, T. Tracking an RGB-D camera using points and planes. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Sydney, Australia, 1–8 December 2013; pp. 51–58.
31. Yu, Z.; Zheng, J.; Lian, D.; Zhou, Z.; Gao, S. Single-image piece-wise planar 3d reconstruction via associative embedding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1029–1037.
32. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
33. Gallup, D.; Frahm, J.M.; Mordohai, P.; Yang, Q.; Pollefeys, M. Real-time plane-sweeping stereo with multiple sweeping directions. In Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 1–8.
34. Wu, C.; Clipp, B.; Li, X.; Frahm, J.M.; Pollefeys, M. 3D model matching with viewpoint-invariant patches (VIP). In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 1–8.
35. Yao, Y.; Luo, Z.; Li, S.; Fang, T.; Quan, L. Mvsnet: Depth inference for unstructured multi-view stereo. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany,8–14 September 2018; pp. 767–783.
36. Xi, W.; Chen, X. Reconstructing piecewise planar scenes with multi-view regularization. *Comput. Vis. Media* **2019**, *5*, 337–345. [CrossRef]
37. Liu, J.; Ji, P.; Bansal, N.; Cai, C.; Yan, Q.; Huang, X.; Xu, Y. Planemvs: 3d plane reconstruction from multi-view stereo. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 8665–8675.
38. Yi, K.M.; Trulls, E.; Ono, Y.; Lepetit, V.; Salzmann, M.; Fua, P. Learning to find good correspondences. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2666–2674.
39. Sarlin, P.E.; DeTone, D.; Malisiewicz, T.; Rabinovich, A. Superglue: Learning feature matching with graph neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 18–19 June 2020; pp. 4938–4947.
40. Choy, C.; Dong, W.; Koltun, V. Deep global registration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 18–19 June 2020; pp. 2514–2523.
41. Mikolajczyk, K.; Tuytelaars, T.; Schmid, C.; Zisserman, A.; Matas, J.; Schaffalitzky, F.; Kadir, T.; Gool, L.V. A comparison of affine region detectors. *Int. J. Comput. Vis.* **2005**, *65*, 43–72. [CrossRef]
42. Tuytelaars, T.; Mikolajczyk, K. Local invariant feature detectors: A survey. *Found. Trends® Comput. Graph. Vis.* **2008**, *3*, 177–280. [CrossRef]
43. Mishkin, D.; Matas, J.; Perdoch, M. MODS: Fast and robust method for two-view matching. *Comput. Vis. Image Underst.* **2015**, *141*, 81–93. [CrossRef]
44. Morel, J.M.; Yu, G. ASIFT: A new framework for fully affine invariant image comparison. *SIAM J. Imaging Sci.* **2009**, *2*, 438–469. [CrossRef]
45. Mishkin, D.; Matas, J.; Perdoch, M.; Lenc, K. Wxbs: Wide baseline stereo generalizations. *arXiv* **2015**, arXiv:1504.06603.

46. Whelan, T.; Johannsson, H.; Kaess, M.; Leonard, J.J.; McDonald, J. Robust real-time visual odometry for dense RGB-D mapping. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 5724–5731.
47. Schops, T.; Sattler, T.; Pollefeys, M. Bad slam: Bundle adjusted direct rgb-d slam. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 134–144.
48. Cai, R.; Hariharan, B.; Snavely, N.; Averbuch-Elor, H. Extreme rotation estimation using dense correlation volumes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 14566–14575.
49. Chen, K.; Snavely, N.; Makadia, A. Wide-baseline relative camera pose estimation with directional learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 3258–3268.
50. Deng, K.; Liu, A.; Zhu, J.Y.; Ramanan, D. Depth-supervised nerf: Fewer views and faster training for free. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 12882–12891.
51. Wang, G.; Chen, Z.; Loy, C.C.; Liu, Z. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023; pp. 9065–9076.
52. Agarwala, S.; Jin, L.; Rockwell, C.; Fouhey, D.F. Planeformers: From sparse view planes to 3d reconstruction. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 192–209.
53. Wu, Y.; Kirillov, A.; Massa, F.; Lo, W.Y.; Girshick, R. Detectron2. 2019. Available online: https://github.com/facebookresearch/detectron2 (accessed on 21 August 2023).
54. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
55. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part V 13; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
56. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv* **2015**, arXiv:1506.01497.
57. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
58. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 815–823.
59. Kuhn, H.W. The Hungarian method for the assignment problem. *Nav. Res. Logist. (NRL)* **2005**, *52*, 7–21. [CrossRef]
60. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
61. 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 Proceedings of the International Conference on 3D Vision (3DV), Qingdao, China, 10–12 October 2017.
62. Savva, M.; Kadian, A.; Maksymets, O.; Zhao, Y.; Wijmans, E.; Jain, B.; Straub, J.; Liu, J.; Koltun, V.; Malik, J.; et al. Habitat: A platform for embodied ai research. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27–28 October 2019; pp. 9339–9347.
63. Tulsiani, S.; Gupta, S.; Fouhey, D.F.; Efros, A.A.; Malik, J. Factoring shape, pose, and layout from the 2d image of a 3d scene. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 302–310.
64. Kulkarni, N.; Misra, I.; Tulsiani, S.; Gupta, A. 3d-relnet: Joint object and relational network for 3d prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27–28 October 2019; pp. 2212–2221.
65. Li, L.; Khan, S.; Barnes, N. Silhouette-assisted 3d object instance reconstruction from a cluttered scene. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 2080-2088 .
66. Nie, Y.; Han, X.; Guo, S.; Zheng, Y.; Chang, J.; Zhang, J.J. Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 55–64.
67. Cai, Z.; Zhang, J.; Ren, D.; Yu, C.; Zhao, H.; Yi, S.; Yeo, C.K.; Change Loy, C. Messytable: Instance association in multiple camera views. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Part XI 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 1–16.