2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Deep Depth Estimation from Visual-Inertial SLAM

Kourosh Sartipi, Tien Do, Tong Ke, Khiem Vuong, and Stergios I. Roumeliotis†

2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-7281-6212-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/IROS45743.2020.9341448

***Abstract*— This paper addresses the problem of learning to complete a scene's depth from sparse depth points and images of indoor scenes. Specifically, we study the case in which the sparse depth is computed from a visual-inertial simultaneous localization and mapping (VI-SLAM) system. The resulting point cloud has low density, it is noisy, and has non-uniform spatial distribution, as compared to the input from active depth sensors, e.g., LiDAR or Kinect. Since the VI-SLAM produces point clouds only over textured areas, we compensate for the missing depth of the low-texture surfaces by leveraging their planar structures and their surface normals which is an important intermediate representation. The pre-trained surface normal network, however, suffers from large performance degradation when there is a significant difference in the viewing direction (especially the roll angle) of the test image as compared to the trained ones. To address this limitation, we use the available gravity estimate from the VI-SLAM to warp the input image to the orientation prevailing in the training dataset. This results in a significant performance gain for the surface normal estimate, and thus the dense depth estimates. Finally, we show that our method outperforms other state-of-the-art approaches both on training (ScanNet [1] and NYUv2 [2]) and testing (collected with Azure Kinect [3]) datasets.**

I. Introduction

Determining the dense depth of a scene has important applications in augmented reality, motion planning, and 3D mapping. This is often achieved by employing depth sensors such as Kinect and LiDAR. Besides the high-cost, size, and power requirements of depth sensors, their measurements are either sparse (LiDAR) or unreliable at glossy, reflective, and far-distance surfaces (Kinect). Recent research has shown that some of the limitations of depth sensors can be overcome by employing RGB images along with the strong contextual priors learned from large-scale datasets ($\simeq$ 200K images), using a deep convolutional neural network (CNN). Specifically, to produce dense depth estimates, recent approaches have employed CNNs with three types of inputs: (a) a single RGB image (e.g., [4], [5], [6], [7], [8]); (b) poses (position and orientation) and optical flow from multiple images (e.g., [9], [10], [11]); (c) a single RGB image and sparse depth [12], [13], [14], [15], [16], [17], [18], [19], [20], [21].

In our work, which falls under (c) and is inspired by [16], we seek to estimate dense depth by fusing RGB images, learned surface normals, and sparse depth information. In contrast to [16] that uses LiDAR, we obtain the sparse points from a real-time visual-inertial SLAM (VI-SLAM) system [22]. There are thee key differences between the point clouds directly measured by a Kinect or a LiDAR and those estimated by VI-SLAM: Density, accuracy, and spatial distribution. In particular, the VI-SLAM point cloud comprises sparse points ($\simeq$ 0.5% of an image's pixels) that are extracted and tracked across images and triangulated using the camera's estimated poses. For this reason, the accuracy of these points varies widely, depending on the local geometry and the camera's motion. Moreover, the VI-SLAM points are not uniformly distributed across an image. They are usually found on high-texture surfaces, while textureless areas such as walls, floors, and ceilings that are ubiquitous in man-made environments often contain few points only. Due to this large domain gap, networks trained using Kinect or LiDAR data suffer from a significant performance degradation when given 3D points triangulated by VI-SLAM as sparse depth information [18], [19]. Furthermore, in large-scale indoor datasets (ScanNet [1], NYUv2 [2], Matterport3D [23]), the images are usually aligned with gravity [24]. During inference time, this bias results in further performance degradation of the depth completion networks [14], as well as optical-flow to depth networks trained on these data [9].

A straightforward approach to address the domain gap and the lack of images from various vantage points is to collect and process more data and re-train the network. This, however, is both labor and time intensive. Alternatively, 3D mesh reconstructions from RGB-D sequences (e.g., BundleFusion [25], BAD-SLAM [26], etc.) can be employed to synthesize novel views of RGB images. The accuracy of the 3D mesh, however, is usually not sufficient and the quality of the resulting data depends on many factors, e.g., the overlap between frames, the sparse points tracking error, etc. For this reason, in our work, we employ the VI-SLAM point cloud when training the depth estimation network while incorporating the VI-SLAM estimate of gravity direction to reduce the effect of "unseen orientation." Specifically, to address the domain gap issue in the point cloud, we first train a network using sparse depth input from the point cloud generated by VI-SLAM, instead of randomly sampling from the ground-truth depth (e.g., [14]). Furthermore, to increase the density of the point cloud, we leverage the planar structures commonly found indoors. Secondly, to solve the domain gap in viewing directions, we align each input image taken with "unseen orientation" to a rectified orientation that the pre-trained network is more familiar with, using the gravity direction estimated from VI-SLAM. We show that this approach not only improves the performance on images taken with familiar orientations, but also achieves satisfactory

†Kourosh Sartipi, Tien Do, Tong Ke, Khiem Vuong, and Stergios I. Roumeliotis are with University of Minnesota, Minneapolis, MN 55455 sarti009@umn.edu, doxxx104@umn.edu, kexxx069@umn.edu, vuong067@umn.edu, stergios@umn.edu.

generalization on the unfamiliar ones. In summary, our main contributions are:

- We introduce an efficient approach to improve the generalization of the VI-SLAM depth completion that leverages (i) the planar geometry of the scene and (ii) the camera's orientation with respect to gravity.
- We implement a full pipeline from VI-SLAM to dense depth estimation for evaluation on Azure Kinect [3] and perform extensive experiments that demonstrate the advantages of our method over state-of-the-art approaches on dense-depth estimation [9], [14].

II. Related Work

Single-view depth estimation. Depth estimation from a single image has been studied by early works such as [27] which is based on handcrafted features. More recently, numerous deep learning-based approaches have appeared (e.g., [5], [4], [6], [7]) for estimating dense depth. Note, however, that given a single image, the pixels' depths cannot be determined only from local features; hence data-driven approaches have to rely on the global context of the image, which is learned from the training data. Therefore, despite the surprisingly good performance when trained and tested with images from the same dataset, they exhibit poor generalization on cross-dataset experiments [8].

Multi-view depth estimation. One way to overcome the single-view depth estimation challenges is to consider multiple camera poses and optical flow. Specifically, the scene depth can be recovered up to scale, given information from multiple views, or with metric scale if the poses are estimated with the aid of other sensors (e.g., IMU). In particular, [10] takes two images as input, estimates the optical flow as an intermediate result, and refines the depths as well as the poses iteratively. On the other hand, [11] computes the photometric errors by warping adjacent images to the current one, and inputs them along with the current image to neural network. Lastly, [9] estimates a probability distribution for the depth, instead of a single depth, and refines the initial depth estimates from a neural network by warping the depth distribution of adjacent images and fusing them in a Bayesian fashion. These methods implicitly estimate depth from poses and optical flow using exclusively neural networks, instead of computing the depths of at least some points directly from geometry. As shown in [20], employing these sparse or semi-dense depths as the input of neural network results in higher performance as compared to relying on the optical flow.

Depth completion from RGB and sparse depth. A key difference between the methods described hereafter and the previous two families of approaches is that the domains of their inputs are significantly different. Specifically, the RGB image has a well-defined range of values for all pixels, while the sparse depth image has only few valid values where the majority of the pixels' depths are unknown. To address this domain difference, many approaches such as [28], [29] propose normalized convolution and upsampling layers so that the missing pixels will not be processed in the convolution kernels. On the other hand, methods such as [30], [31], [15], [32], [33] do not treat the sparse normal input differently, which indicates with proper training applying standard convolution can achieve comparable results.

A different approach proposed by [16] showed that employing a network to compute the surface normals based on the RGB image and using the normals as an additional input to the later layers can improve the network's performance. Moreover, it was proposed to concatenate color/normal channels and sum the depth channels for the skip-connections between the encoder and decoder. Due to space considerations, we have included the experiments on generalizability of this network in [34].

In [14], Cheng et al. introduced the convolutional spatial propagation network (CSPN), where the initial depth image and an affinity matrix are first produced by a CNN and then the depth is iteratively refined through a diffusion process involving the affinity matrix and the current depth estimate. Furthermore, for the case of depth completion, CSPN employs validity masks to retain the depth of the sparse input, and hence does not allow the network to correct potentially noisy measurements. As we will show in our experimental results, this policy will lead to loss in accuracy when the input depth is noisy.

Closely related to our work are those of [18] and [35]. In [18], a system to compute dense depth from either LiDAR or SLAM point-cloud on-board MAVs is described using the confidence propagation network similar to [28]. Since, however, their focus is on real-time performance on devices with limited resources, their accuracy is lower than the state of the art. As in our work, [35] also employs a VI-SLAM system to generate the sparse input. Moreover, it uses a two-step process to first extend the depth data from the sparse 3D point cloud and then combine them with the RGB image as the input to a network that produces the final dense-depth image. In particular, [35] first employs Delaunay triangulation [36] to fit a triangular mesh on the sparse points of the image and then computes the depth of all the pixels falling within each triangle using its plane equation. Note that the accuracy of this approach depends heavily on the assumption that the triangles match with the planar surfaces of the image's scene, which, in general, is not true.[1]

III. Technical Approach

In this paper, we propose a method to accurately predict dense depths using only sensors available on most mobile devices, i.e., a camera and an IMU. Fig. 1 depicts an overview of our system, where the depth completion network (DCN) estimates the dense depth from the following inputs: (i) the RGB image, (ii) a sparse depth image based on the 3D points triangulated by the VI-SLAM, and (iii) the surface normal map predicted by another CNN. As shown

[1]Another contribution of [35] is creating visual-inertial datasets for depth estimation. However, it does not have ground-truth surface normal, which our networks relies on. Therefore, we instead employ ScanNet [1], with surface normal readily available from 3D mesh, to train our network. As for evaluating generalization capability, we collected our own dataset with sufficient roll and pitch variation (which is not the case for the dataset of [35]) to highlight the effect of incorporating gravity.

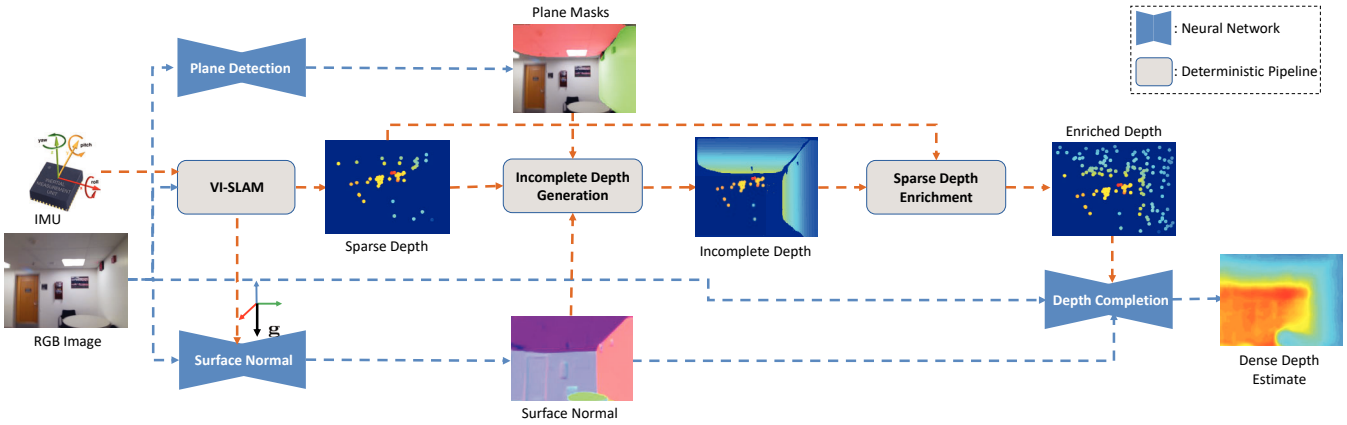


Fig. 1. Overview of the system. At each keyframe, the VI-SLAM (Sec. III-A) processes images and IMU measurements to compute (i) the gravity direction which is provided to the surface-normal network (Sec. III-B), and (ii) the sparse depth image. The sparse depth enrichment module (Sec. III-D) increases the density of the sparse-depth image using information about planes in the image. This is provided by the plane detection network (Sec. III-C) that classifies image pixels as belonging to a particular plane and the surface normal network that estimates the normal of each pixel. The depth completion network (DCN) (Sec. III-E) employs the RGB image, the pixel normals, and the enriched sparse-depth image to produce a dense depth image.

experimentally by [16] and evident in our ablation studies, (iii) improves the accuracy of depth estimation.

Specifically, we employ VI-SLAM [22] (see Sec. III-A), which takes as input images and IMU measurements, to compute the *sparse 3D point cloud* observed by the camera. Then, the *sparse depth image* is obtained by projecting the point cloud to the camera frame. Additionally, a CNN predicts the *surface normal* of every pixel in the RGB image (see Sec. III-B) while leveraging the gravity direction estimated by the VI-SLAM to improve its accuracy. Note that although the sparse depth image from the VI-SLAM can be used directly as input to the DCN, we seek to first increase its density by performing a sparse-depth enrichment step. To do so, we extract the *planar patches* of the image using a CNN (see Sec. III-C), and use the estimated normals along with any 3D points that fall inside a plane, to compute a denser depth representation of the scene (see Sec. III-D). Finally, the DCN (see Sec. III-E) computes the dense depth estimate based on the RGB image, the enriched sparse depth image, and the surface normals.

A. VI-SLAM: Sparse Depth Image Generation

In this work, we employ the inverse square-root sliding window filter (SR-ISWF) [22] to estimate in real-time camera poses and 3D feature positions. Specifically, at each time step the SR-ISWF extracts FAST [37] corners in the current image, and tracks them by matching their corresponding ORB descriptors [38] to the previous images. The SR-ISWF then fuses the IMU measurements and the 2D-to-2D feature tracks across the sliding window to estimate the camera's motion along with the features' positions.

Every time the SR-ISWF processes an image, we project all visible 3D features on the image and use their depth (i.e., their Z component) to create the sparse depth image. The SR-ISWF also computes the gravity direction which is passed to the surface normal network.

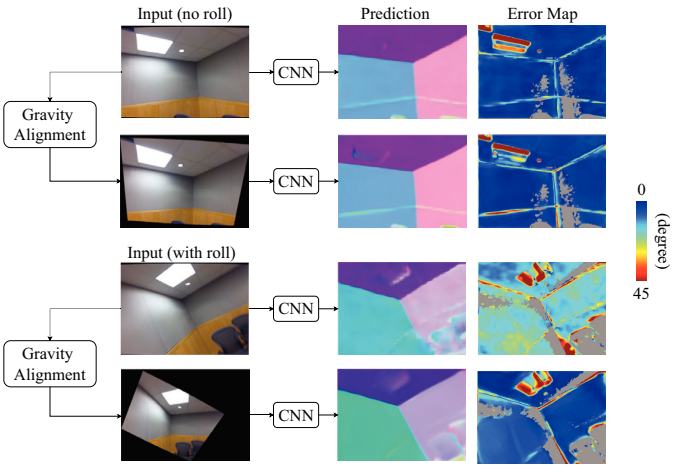
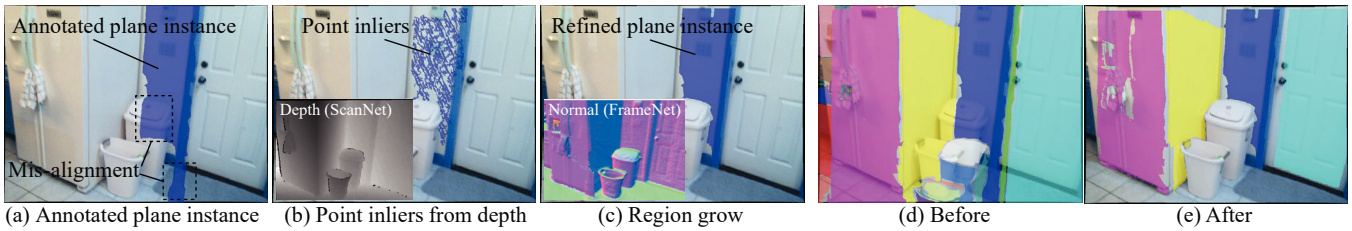


Fig. 2. Top row: Input image from Azure Kinect along with the gravity-aligned image; the performance of FrameNet is satisfactory. Bottom row: Input image with significant roll component, resulting in poor performance. Warping the input based on the gravity direction before passing it through the CNN improves the performance in both cases.

B. Surface Normal Network

As it will become evident from our experimental validation, the performance of the DCN depends on the accuracy of the surface normals prediction. In particular, training state-of-the-art surface-normal-estimation networks such as the FrameNet [39] on large-scale indoor datasets (e.g., ScanNet [1], NYUv2 [2], Matterport3D [23]) does not yield satisfactory results. This is due to the fact that the FrameNet's surface-normal estimator's performance degrades significantly when tested on images whose roll angles deviate substantially from the vertically-aligned images used during training (see Fig. 2).

To address this issue, we follow the approach of [40] to warp the input image so that the gravity, which is estimated from IMU, is aligned with the image's vertical axis. Note that, in this paper, we are interested in how the accuracy



(a) Annotated plane instance (b) Point inliers from depth (c) Region grow (d) Before (e) After

Fig. 3. We refine the imprecise plane annotation: (a) Given an imprecise plane instance, (b) we fit a plane using the depth from ScanNet with a strict threshold, which produces a set of sparse, yet accurate, points. (c) Using the point set, we incrementally grow the planar region based on the surface normal from FrameNet. (d-e) This refinement allows us to precisely label the plane instances.

of surface-normal prediction affects the depth-completion performance, rather than solely focusing on the surface-normal accuracy as in [40].

The idea of using gravity from VI-SLAM has been proposed in [41] as a regularization for a depth-prediction network during *training* time. In contrast, in our work we employ the online estimated gravity to improve the generalization during *inference* time. Recently, [24] proposed to remove the roll angle from an image by using the orientation estimated from monocular SLAM [42]. This method, though, lacks global information hence it relies on the assumption that the first camera's view is aligned with gravity. In contrast, and due to the observability of gravity in VI-SLAM system [43], our method makes no assumptions about the camera's motion.

C. Plane Detection

A key idea behind our approach is to take advantage of planar surfaces present in the scene to enrich the sparse depth image (see Sec. III-D). To do so, we predict plane masks on the image using a CNN. Specifically, we leverage the ground-truth plane masks from Plane-RCNN [44] which employs the 3D mesh reconstruction created from ScanNet [1] for the multi-plane instance proposals. Unfortunately, the resulting plane annotations are mis-aligned as shown in Fig. 3. Hence, to obtain reliable training data, Plane-RCNN proposed a heuristic method to detect this misalignment based on the discrepancies between the projected 3D mesh reconstruction on each image and their corresponding depth, and then drop any frames with large discrepancies during training. While effective, this method discards a large set of planes annotations during training, which can potentially degrade the plane detector's performance.

To address this problem, we employ RANSAC-based [45] plane fitting and region growing to refine the plane annotations. Specifically, for each annotated plane from Plane-RCNN, we employ 3pt RANSAC, using the corresponding depth from ScanNet and a strict inlier threshold (2 cm) to fit a plane. This yields only a small set of inliers due to the imprecise depth measurements. Given the initial inliers, we grow the coplanar region through neighboring pixels based on two criteria: (i) the distance of each 3D point to the plane is less than 20 cm and (ii) the corresponding surface normal from FrameNet [39] is close (less than 30°) to the plane's normal. If the plane computed through this process is substantially smaller than the annotation, we discard the annotation. By applying this method to all the annotated planes we are able to accurately compute the plane instances (see Fig. 3) and ensure that the planes are well aligned with the RGB images. Lastly, we employ the improved annotations as ground-truth to train a Mask R-CNN [46] network for plane detection.

D. Sparse Depth Enrichment

In this module, we enrich the sparse depth image by increasing the number of points it contains. We focus on the parts of the scene where 3D point features project on a detected plane (see Sect. III-C). This process comprises the following two steps for estimating each plane's parameters:

1) Plane Normal estimation: We randomly select a few pixels as plane-normal hypotheses and compare their directions to the rest of the plane's pixels to find the largest set of normal directions aligned within 10 degrees. We then assign as the plane's normal the average of the normal vectors of the largest set [47].
2) Plane Distance estimation: Given the plane's normal $\mathbf{n}$, each 3D point $\mathbf{p}_i$ expressed in the camera's coordinate frame is a candidate for computing the plane's distance hypothesis $d_i = -\mathbf{n}^T\mathbf{p}_i$. As before, the one with the largest set of inliers is accepted, and the plane's distance d is set to be the average distance of inliers.

Next, we employ $\mathbf{n}$ and d to compute the depth $z_i = -\frac{d}{\mathbf{n}^T\mathbf{b}_i}$ of each pixel, where $\mathbf{b}_i$ is its normalized homogeneous coordinate. Hereafter, we refer to the depth image resulting from this as the *"incomplete depth image"* (see Fig. 1).

Once the incomplete depth image is generated, we select only a subset of the points from it to form the enriched depth image provided to the DCN (see Fig. 1). The reason behind this choice is that often the 3D points triangulated by VI-SLAM may contain errors. In this case, using the entire incomplete depth image as the input to the DCN will bias its output by implicitly forcing it to trust the depth values of many densely distributed pixels with erroneous estimates. On the other hand, our experiments (see Sect. IV-B) and those of [18] show that the DCN performs significantly better when the sparse depth image is uniformly distributed over the RGB image. This requirement, however, is not typically satisfied by indoor areas containing large textureless surfaces. Instead, 3D points often cluster in few parts of an image where high texture is observed. By employing the proposed enrichment procedure, we are able to reduce the gap between the initial

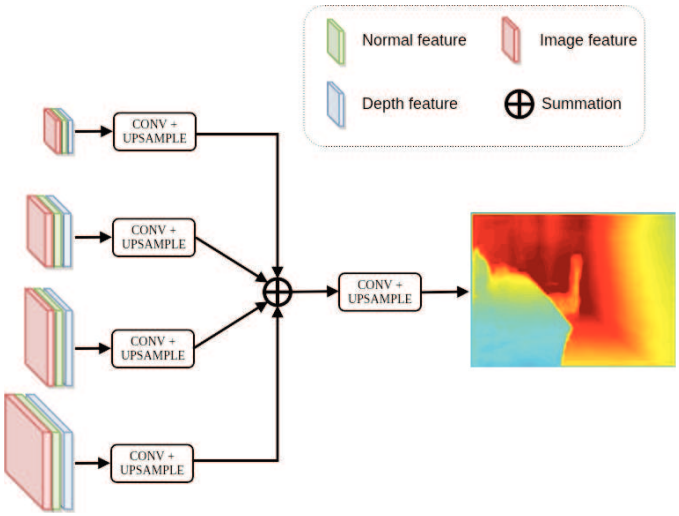


Fig. 4. Decoder structure. Features at different scales of the RGB, normal, and depth images are concatenated together, passed through convolution and upsampling, and then summed. A final convolution and upsampling is applied to compute the dense depth image.

distribution of sparse depth and a uniform distribution. As shown in Sect. IV, sampling about 100 points from the incomplete depth image produces the best results.

E. Depth Completion Network

Our DCN is inspired by the Panoptic FPN [48]. Specifically, each of the RGB, normal, and the sparse depth images are processed through separate encoders, which compute four feature tensors per input corresponding to different layers of the ResNet. Before being processed by the decoder, we concatenate the features of the RGB, normal, and sparse depth images at each scale and apply convolution and upsampling, resulting in four feature tensors of size $128 \times 60 \times 80$. These are then summed together and passed through a final convolution and upsampling layer resulting in the dense-depth image (see Fig. 4).

IV. Experimental Results

In what follows, we experimentally verify the performance of our method, and compare it to state-of-the-art approaches. Although our target application is for VI-SLAM systems, the lack of visual-inertial training data has led us to employ RGB-D datasets (i.e., ScanNet [1] and NYUv2 [2]) to train our networks. To assess the performance on visual-inertial data and verify the generalization capability of our approach, we have also collected data using Azure Kinect [3]. In terms of depth error metrics, we report root mean square error (RMSE) in meters, and $E(\hat{D}, \delta)$, which specifies the percentage of the estimated depths $\hat{D}$ for which $\max(\frac{\hat{D}}{D}, \frac{D}{\hat{D}}) < \delta$, where D is the ground-truth depth. For surface-normal error metrics, we report mean absolute of the error (MAD), median of absolute error (Median), and the percentage of pixels with angular error below a threshold ξ with $\xi = 11.25°, 22.5°, 30.0°$.

Experiment Setup: All the networks in this paper have been implemented in PyTorch [49], and the original authors' code and provided network weights have been used for comparisons against other methods. To train the DCN, we employ L_1 loss and the Adam optimizer [50] with a learning rate of 10^{-4}. We train the model for 20 epochs and report the best epoch on the corresponding dataset's validation set. The training was done on an NVIDIA Tesla V100 GPU with 32GB of memory with a batch size of 16. Since the aspect ratio of the Azure Kinect dataset images is significantly different than those of ScanNet and NYUv2, we first crop them and then resize to 320×240. For the ScanNet and NYUv2 we only apply resizing. Our neural network code is available at `https://github.com/MARSLab-UMN/vi_depth_completion` along with our datasets. Furthermore, due to limited space, we have provided our full experimental results in [34].

A. Comparison on ScanNet Datasets

We train and evaluate different configurations of our approach on the ScanNet indoor datasets and compare our performance with NeuralRGBD [9]. Although ScanNet does not contain inertial data, we leverage other available annotations to estimate (i) the gravity direction, and (ii) a 3D point cloud, required for our training and testing. Specifically, we obtain the gravity direction from the normal direction of pixels labeled as the ground (by FrameNet [39]). To compute the 3D point cloud, we first extract FAST corners [37], track them via KLT [51], and remove outliers by employing the 5pt-RANSAC [52]. The inlier tracks are then triangulated using the provided camera poses to generate the sparse 3D point cloud (on average, for each image we compute 58 sparse depth values from the point cloud which corresponds to 0.07% of pixels). Lastly, we generate plane masks for the entire dataset and compute the incomplete depth images through the process described in Sect. III-D.

In Table I, we evaluate the accuracy of the reconstructed sparse and incomplete depth inputs as well as the final depth output using the following configurations:

- *Triangulation*: The sparse depth resulting from projecting the 3D point cloud on the image.
- *Incomplete Depth*: The incomplete depth images generated through the process described in Sect. III-D.
- *NeuralRGBD* [9]: State-of-the-art depth estimation from a sequence of images.
- *Ours-SD*: DCN results trained and tested using the sparse depth as input.
- *Ours-ID*: DCN results trained and tested using the incomplete depth as input.
- *Ours-Enriched 100, 200*: Using *Ours-SD* trained network with selecting 100 or 200 additional points from the incomplete depth to generate the enriched depth (see Sect. III-D and Fig. 1).

As evident from Table I, the errors $E(\hat{D}, \delta)$ of the triangulated points and incomplete depths show that the inputs are relatively accurate. Furthermore, using the enriched depth with 100 points as input has slightly higher accuracy in the stricter metrics ($\delta = 1.05, 1.10$). On the other hand, a denser enriched depth image comprising 200 points decreases the

TABLE I

PERFORMANCE OF DEPTH COMPLETION ON SCANNET TEST SET

		$E(\hat{D},\delta)$				
	RMSE ↓	1.05 ↑	1.10↑	1.25↑	1.25^2 ↑	1.25^3 ↑
Triangulation	0.153	75.89	92.19	98.51	99.64	99.89
Incomplete Depth	0.201	66.39	85.30	96.64	99.05	99.60
NeuralRGBD [9]	0.294	46.35	72.99	93.00	98.30	99.43
Ours-SD	0.266	54.09	78.41	94.68	**98.91**	**99.71**
Ours-ID	**0.265**	54.65	**78.71**	**94.72**	98.85	99.67
Ours-Enriched 100	0.271	**54.85**	78.54	94.55	98.76	99.66
Ours-Enriched 200	0.271	54.37	78.11	94.46	98.73	99.63

accuracy for the reasons explained in Sect. III-D. Although the network trained and tested using incomplete depth performs slightly better than both sparse and enriched depths for the ScanNet dataset, its performance degrades drastically on cross-dataset (see Sect. IV-C). Finally, these results show that our approach outperforms [9] on all metrics, especially the stricter ones.

B. Comparison on NYUv2 Depth Dataset

To compare against the CSPN [14], we also test our approach on the NYUv2 datasets. In our evaluation, we used the official 249 training scenes and sub-sampled $25,000$ color-depth image pairs for the network to train, while tested on the standard 654-image test set. Since NYUv2 does not provide camera poses, to generate the sparse depth, we first extract 2D-to-2D correspondences (see Sect. IV-A), and then sample from the ground-truth depth instead of projecting a 3D point-cloud as input to the DCN.

Table II shows the performance of the proposed approach compared to CSPN using their pre-trained model. Note that the CSPN model has been trained on sparse depth generated by randomly sampling from the ground-truth depth. In our tests, we compare the computed depth when the sparse depth is from 2D-to-2D tracks (denoted as "Feature" in Table II), and by randomly sampling 200 points from the ground-truth ("Uniform"). These results show that our approach outperforms CSPN in all metrics. Additionally, the gap in performance when employing features is much larger than using random samples. This is due to the distribution of FAST corners on the images, where more points are located on the textured areas as compared to the textureless ones.

TABLE II

PERFORMANCE OF DEPTH COMPLETION ON NYUV2 TEST SET

		$E(\hat{D},\delta)$				
	RMSE ↓	1.05 ↑	1.10↑	1.25↑	1.25^2 ↑	1.25^3 ↑
CSPN[14] (Feature)	0.46	69.31	78.39	86.47	91.65	94.57
Ours-SD (Feature)	**0.20**	**80.38**	**91.27**	**97.53**	**99.50**	**99.89**
CSPN[14] (Uniform)	**0.18**	87.56	94.17	98.24	99.59	99.88
Ours-SD (Uniform)	**0.18**	**88.86**	**94.82**	**98.46**	**99.66**	**99.91**

C. Generalization Capability on Azure Kinect Datasets

In order to evaluate our DCN on datasets with visual-inertial data, and to better compare the generalization capability of our approach against the alternative ones, we collected 24 datasets in indoor areas using the Azure Kinect [3]. Each dataset comprises color and depth images at 30 Hz, as well as IMU data at 1.6 KHz. The depth images are employed as the ground-truth, while the color images and the IMU measurements are processed by the VI-SLAM to compute the camera's poses and the triangulated 3D feature positions. These datasets consist of ∼8K keyframes used for evaluation. Unless otherwise specified, all networks considered in this section are trained only on ScanNet.

TABLE III

PERFORMANCE OF SURFACE NORMAL ESTIMATION ON AZURE KINECT

Gravity-aligned frames	MAD↓	Median↓	RMSE↓	11.25°↑	22.5°↑	30.0°↑
Vanilla	9.15	4.89	15.91	80.83	90.83	93.41
Warping Augmentation	9.31	4.61	17.23	81.88	90.59	92.95
Gravity Alignment	**8.62**	**4.26**	**15.57**	**82.03**	**91.17**	**93.68**
Gravity-non-aligned frames	MAD↓	Median↓	RMSE↓	11.25°↑	22.5°↑	30.0°↑
Vanilla	14.40	6.86	23.32	65.55	81.56	85.77
Warping Augmentation	11.87	5.63	20.87	**74.91**	**86.48**	89.67
Gravity Alignment	**11.46**	**5.22**	**19.88**	74.24	86.29	**89.75**
All frames	MAD↓	Median↓	RMSE↓	11.25°↑	22.5°↑	30.0°↑
Vanilla	11.37	5.41	19.31	74.35	86.48	89.95
Warping Augmentation	10.62	5.09	19.19	78.31	88.48	91.27
Gravity Alignment	**10.03**	**4.65**	**17.84**	**78.33**	**88.73**	**91.74**

First, we present the performance and the effectiveness of our gravity alignment for surface-normal estimation (Sec. III-B) on Azure Kinect datasets. Table III compares the proposed gravity alignment (see Sect. III-B) against the following alternatives: (i) Vanilla - training with standard ground-truth surface normal and (ii) Warping augmentation - improving generalization by warping the input image with random rotation during training; on two categories of images (1) gravity aligned and (2) gravity non-aligned. The network used in all cases is DORN [5] and is trained with the truncated angular loss [40]. In these tests, we achieve satisfactory performance when evaluating DORN on the scenes with gravity-aligned images (Table III, Gravity-aligned frames). Its performance, however, reduces drastically when the images have large pitch/roll rotations (Table III, Gravity-non-aligned frames). We attribute this degradation to the lack of training images with large pitch/roll angles. To assess the accuracy of our approach, we also compare against a naive data augmentation scheme that randomly warps each input image to provide the network with more diverse viewing directions. Finally, Table III shows that by warping in a direction aligned with the gravity estimated by VI-SLAM, our method outperforms both the baseline DORN and training with data augmentation (i.e., randomly warping images) in terms of generalization performance without requiring data augmentation, thus resulting in shorter training. Table IV presents the depth completion performance of our method, as compared to the NeuralRGBD and CSPN, on the Azure Kinect dataset.

TABLE IV

PERFORMANCE OF DEPTH COMPLETION ON AZURE KINECT DATASET

		$E(\hat{D},\delta)$				
	RMSE ↓	1.05 ↑	1.10↑	1.25↑	1.25^2 ↑	1.25^3 ↑
CSPN-NOISY	1.461	5.10	9.03	17.02	29.52	43.25
CSPN-GT	1.465	8.10	11.20	17.93	29.49	42.69
Ours-SD†	0.913	11.70	22.55	47.96	74.85	88.03
Ours-Enriched 100†	**0.814**	**15.09**	**28.74**	**57.93**	**82.23**	**91.48**
NeuralRGBD	0.717	20.03	36.25	65.25	86.28	94.59
Ours-ID	0.534	18.64	35.54	71.00	93.84	98.20
Ours-SD	0.524	19.96	37.55	73.78	94.24	98.52
Ours-Enriched 100	0.496	24.21	**44.34**	79.07	95.32	98.71
Ours-Enriched 100+g	**0.490**	**24.26**	44.31	**79.23**	**95.65**	**98.95**

To properly compare with the pretrained NeuralRGBD and CSPN, we separately train our DCN network on ScanNet and NYUv2 (denoted with †). Since CSPN is trained using points sampled from the ground-truth depth, besides evaluating its performance on the triangulated point cloud (CSPN-NOISY), we also assess its performance when the sparse

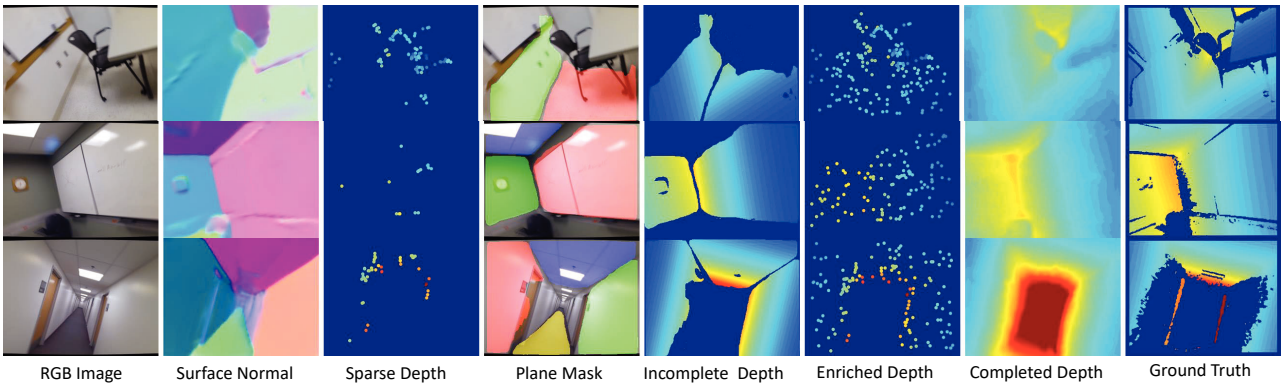


Fig. 5. Qualitative results on gravity-non-aligned scenes.

depth of the same locations is extracted from the ground-truth depth (CSPN-GT). In addition to the previous results, we also include our depth-enrichment method with the surface normal computed using our proposed gravity-aware network (Ours-Enriched 100+g). Table IV shows that our algorithms significantly outperform the alternative methods in all metrics for cross-dataset evaluation, thus confirming the generalization capability of the depth-enrichment approach for the network trained on either NYUv2 or ScanNet. In addition, it shows that the depth-enrichment method has a stronger impact than the accuracy of the surface-normal prediction. Fig. 5 qualitatively illustrates our method for gravity-non-aligned images.

Higher accuracy normals result in more precise depth. We examine the performance of our DCN network using surface-normal input with and without the gravity alignment, denoted as Ours-SD+g and Ours-SD in Table V, respectively. In order to highlight the impact of the surface-normal accuracy, we evaluate the networks on the gravity-non-aligned subset of the Azure Kinect dataset, which is shown to have a significant improvement on surface-normal accuracy with the gravity alignment (see Sect. IV-C). Table V illustrates that with gravity alignment, the depth accuracy increases more as compared to ones evaluated on the entire Azure Kinect dataset (see Sect. IV-C).

TABLE V

BETTER NORMAL RESULTS IN BETTER DEPTH

		$E(\hat{D}, \delta)$				
	RMSE ↓	1.05 ↑	1.10↑	1.25↑	1.25^2 ↑	1.25^3 ↑
Ours-SD	0.540	15.58	31.33	70.81	93.86	98.11
Ours-SD+g	**0.514**	**15.94**	**32.55**	**71.94**	**95.06**	**98.66**

V. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced two efficient methods for improving the generalization performance of sparse-to-dense depth completion: (i) Transform the data to a form that the network is familiar with to produce better surface normal estimates, and (ii) Generate an enriched sparse-depth image that significantly improves the performance both on similar scenes to the training ones and the generalization datasets collected using different devices. We thoroughly evaluate multiple configurations of our approach and show its superior performance and generalization ability comparing to other state-of-the-art depth-completion methods. As part of our future work, we plan to incorporate the uncertainty of depth inputs to the network. Specifically, we will investigate alternative methods for enriching the sparse depth based on uncertainty instead of random sampling.

REFERENCES

[1] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, July 21–26 2017, pp. 5828–5839.

[2] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. of the European Conference on Computer Vision*, Florence, Italy, Oct. 7–13 2012, pp. 746–760.

[3] Microsoft, "Azure Kinect DK," https://azure.microsoft.com/en-us/services/kinect-dk/.

[4] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Advances in Neural Information Processing Systems*, Montreal, Canada, Dec. 8–13 2014, pp. 2366–2374.

[5] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, June 18–22 2018, pp. 2002–2011.

[6] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *Proc. of the International Conference on 3D Vision*, Stanford, CA, Oct. 25–28 2016, pp. 239–248.

[7] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2024–2039, 2015.

[8] Z. Li and N. Snavely, "Megadepth: Learning single-view depth prediction from internet photos," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, June 18–22 2018, pp. 2041–2050.

[9] C. Liu, J. Gu, K. Kim, S. G. Narasimhan, and J. Kautz, "Neural RGBD sensing: Depth and uncertainty from a video camera," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, June 16–20 2019, pp. 10 986–10 995.

[10] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox, "Demon: Depth and motion network for learning monocular stereo," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, July 21–26 2017, pp. 5038–5047.

[11] H. Zhou, B. Ummenhofer, and T. Brox, "Deeptam: Deep tracking and mapping," in *Proc. of the European Conference on Computer Vision*, Munich, Germany, Sept. 8–14 2018, pp. 822–838.

[12] J. Tang, J. Folkesson, and P. Jensfelt, "Sparse2dense: From direct sparse odometry to dense 3-D reconstruction," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 530–537, 2019.

[13] Y. Liao, L. Huang, Y. Wang, S. Kodagoda, Y. Yu, and Y. Liu, "Parse geometry from a line: Monocular depth estimation with partial laser observation," in *Proc. of the IEEE International Conference on*

Robotics and Automation, Singapore, Singapore, May 29 – June 3 2017, pp. 5059–5066.

[14] X. Cheng, P. Wang, and R. Yang, "Depth estimation via affinity learned with convolutional spatial propagation network," in *Proc. of the European Conference on Computer Vision*, Munich, Germany, Sept. 8–14 2018, pp. 103–119.

[15] F. Mal and S. Karaman, "Sparse-to-dense: Depth prediction from sparse depth samples and a single image," in *Proc. of the IEEE International Conference on Robotics and Automation*, Brisbane, Australia, May 21–25 2018, pp. 1–8.

[16] J. Qiu, Z. Cui, Y. Zhang, X. Zhang, S. Liu, B. Zeng, and M. Pollefeys, "Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, June 16–20 2019, pp. 3313–3322.

[17] Y. Yang, A. Wong, and S. Soatto, "Dense depth posterior (DDP) from single image and sparse range," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, June 16–20 2019, pp. 3353–3362.

[18] L. Teixeira, M. R. Oswald, M. Pollefeys, and M. Chli, "Aerial single-view depth completion with image-guided uncertainty estimation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1055–1062, 2020.

[19] A. Wong, X. Fei, S. Tsuei, and S. Soatto, "Unsupervised depth completion from visual inertial odometry," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1899–1906, 2020.

[20] Z. Li, T. Dekel, F. Cole, R. Tucker, N. Snavely, C. Liu, and W. T. Freeman, "Learning the depths of moving people by watching frozen people," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, June 16–20 2019, pp. 4521–4530.

[21] Y. Zhang and T. Funkhouser, "Deep depth completion of a single RGB-D image," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, June 18–22 2018, pp. 175–185.

[22] K. J. Wu, A. M. Ahmed, G. A. Georgiou, and S. I. Roumeliotis, "A square root inverse filter for efficient vision-aided inertial navigation on mobile devices," in *Proc. of Robotics: Science and Systems*, Rome, Italy, July 12–16 2015.

[23] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3D: Learning from RGB-D data in indoor environments," *Proc. of the International Conference on 3D Vision*, Oct. 10–12 2017.

[24] Y. Saito, R. Hachiuma, M. Yamaguchi, and H. Saito, "In-plane rotation-aware monocular depth estimation using SLAM," in *International Workshop on Frontiers of Computer Vision (IW-FCV)*. Springer, Singapore, 2020, pp. 305–317.

[25] A. Dai, M. Nießner, M. Zollöfer, S. Izadi, and C. Theobalt, "BundleFusion: Real-time globally consistent 3D reconstruction using on-the-fly surface re-integration," *ACM Transactions on Graphics*, vol. 36, no. 4, p. 1, 2017.

[26] T. Schops, T. Sattler, and M. Pollefeys, "BAD SLAM: Bundle adjusted direct RGB-D SLAM," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, June 16–20 2019, pp. 134–144.

[27] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning depth from single monocular images," in *Advances in Neural Information Processing Systems*, Vancouver, Canada, Dec. 4–9 2006, pp. 1161–1168.

[28] A. Eldesokey, M. Felsberg, and F. S. Khan, "Confidence propagation through CNNs for guided sparse depth regression," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, July 2019.

[29] Z. Huang, J. Fan, S. Cheng, S. Yi, X. Wang, and H. Li, "HMS-Net: Hierarchical multi-scale sparsity-invariant network for sparse depth completion," *IEEE Transactions on Image Processing*, vol. 29, pp. 3429–3441, Dec. 2019.

[30] M. Jaritz, R. De Charette, E. Wirbel, X. Perrotton, and F. Nashashibi, "Sparse and dense data with CNNs: Depth completion and semantic segmentation," in *Proc. of the International Conference on 3D Vision*, Verona, Italy, Sept. 5–8 2018, pp. 52–60.

[31] W. Van Gansbeke, D. Neven, B. De Brabandere, and L. Van Gool, "Sparse and noisy LiDAR completion with RGB guidance and uncertainty," in *Proc. of the International Conference on Machine Vision Applications*, Tokyo, Japan, May 27–31 2019, pp. 1–6.

[32] F. Ma, G. V. Cavalheiro, and S. Karaman, "Self-supervised sparse-to-dense: Self-supervised depth completion from LiDAR and monocular camera," in *Proc. of the IEEE International Conference on Robotics and Automation*, Montreal, Canada, May 20–24 2019, pp. 3288–3295.

[33] B.-U. Lee, H.-G. Jeon, S. Im, and I. S. Kweon, "Depth completion with deep geometry and context guidance," in *Proc. of the IEEE International Conference on Robotics and Automation*, Montreal, Canada, May 20–24 2019, pp. 3281–3287.

[34] K. Sartipi, T. Do, T. Ke, K. Vuong, and S. I. Roumeliotis, "Deep depth estimation from visual-inertial slam," University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep., July 2020. [Online]. Available: http://mars.cs.umn.edu/papers/DDE_iros2020.pdf

[35] A. Wong, X. Fei, and S. Soatto, "VOICED: Depth completion from inertial odometry and vision," *IEEE Robotics and Automation Letters*, 2020.

[36] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software*, vol. 22, no. 4, pp. 469–483, 1996.

[37] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. of the European Conference on Computer Vision*. Graz, Austria: Springer, May 7 – 13 2006, pp. 430–443.

[38] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. of the IEEE International Conference on Computer Vision*, Barcelona, Spain, Nov. 6–13 2011, pp. 2564–2571.

[39] J. Huang, Y. Zhou, T. Funkhouser, and L. J. Guibas, "FrameNet: Learning local canonical frames of 3D surfaces from a single RGB image," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, June 16–20 2019, pp. 8638–8647.

[40] T. Do, K. Vuong, S. I. Roumeliotis, and H. S. Park, "Surface normal estimation of tilted images via spatial rectifier," in *Proc. of the European Conference on Computer Vision*, Virtual Conference, August 23–28 2020.

[41] X. Fei, A. Wong, and S. Soatto, "Geo-supervised visual depth prediction," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1661–1668, 2019.

[42] R. Mur-Artal, J. Montiel, and J. D. Tardós, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Trans. on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[43] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Consistency analysis and improvement of vision-aided inertial navigation," *IEEE Trans. on Robotics*, vol. 30, no. 1, pp. 158–176, Feb. 2014.

[44] C. Liu, K. Kim, J. Gu, Y. Furukawa, and J. Kautz, "PlaneRCNN: 3D plane detection and reconstruction from a single image," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, June 16–20 2019, pp. 4450–4459.

[45] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

[46] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, July 21–26 2017, pp. 2961–2969.

[47] K. V. Mardia and P. E. Jupp, *Directional statistics*. John Wiley & Sons, 1999.

[48] A. Kirillov, R. Girshick, K. He, and P. Dollár, "Panoptic feature pyramid networks," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, June 16–20 2019, pp. 6399–6408.

[49] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, Vancouver, Canada, Dec. 6–12 2019, pp. 8024–8035.

[50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of the International Conference on Learning Representations*, San Juan, Puerto Rico, May 2–4 2016.

[51] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. of the International Joint Conference on Artificial Intelligence*, Vancouver, British Columbia, Aug. 24–28 1981, pp. 674–679.

[52] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, June 2004.