PLANAR GEOMETRY ESTIMATION WITH DEEP LEARNING

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in Computer Engineering

by
Furkan Eren UZYILDIRIM

June 2022
İZMİR

ACKNOWLEDGMENTS

Life is composed of different obstacles and we somehow break through to most of them. However, the way you progress in life depends on the people around you. I would like to express my greatest gratitude to the people who have being with me during my long and exhaustive Ph.D. journey.

First, I would like to express my gratitude to my Ph.D. supervisor, Dr. Mustafa Özuysal, who advised and encouraged me all the way during my research life. I am grateful for his motivation, patience, continuous support, and sharing his immense knowledge with me. Furthermore, I would like to thank Asst. Prof. Dr. Nesli Erdoğmuş, Assoc. Prof. Dr. Şevket Gümüştekin, Assoc. Prof. Dr. Derya Birant and Assoc. Prof. Dr. Devrim Ünay for taking part in my Ph.D. examination and their valuable feedbacks.

I would also like to thank to my colleagues in Computer Engineering Department at IZTECH especially Dr. Ekincan Ufuktepe, Deniz Kavzak Ufuktepe, Ersin Çine, Dilek Öztürk, Samet Tenekeci, Altuğ Yiğit, and Hüseyin Ünlü for providing a motivated and entertaining working environment. Academic discussions and sharing experiences with them always encourage me to try new ideas and create solutions to problems that I encounter.

Finally, I would like to express my infinite gratitude to my mom Fatma Uzyıldırım and my father Ertuğrul Uzyıldırım for their unconditional love and endless support without any expectations. They always take care me more than myself and make so many sacrifices for me. I completely dedicated this thesis to them.

ABSTRACT

PLANAR GEOMETRY ESTIMATION WITH DEEP LEARNING

Understanding the geometric structure of any scene is one of the oldest problems in Computer Vision. Most scenes include planar regions that provide information about the geometric structure and their automatic detection and segmentation plays an important role in many computer vision applications. In recent years, convolutional neural network architectures have been introduced for piece-wise planar segmentation. They outperform the traditional approaches that generate plane candidates with 3D segmentation methods from the explicitly reconstructed 3D point cloud. However, most of the convolutional neural network architectures are not designed and trained for outdoor scenes, because they require manual annotation, which is a time-consuming task that results in a lack of training data. In this thesis, we propose and develop a deep learning based framework for piece-wise plane detection and segmentation of outdoor scenes without requiring manually annotated training data. We exploit a network trained on imagery with annotated targets and an automatically reconstructed point cloud from either Structure from Motion-Multi View Stereo pipeline or monocular depth estimation network to estimate the training ground truth on the outdoor images in an iterative energy minimization framework. We show that the resulting ground truth estimate of various sets of images in the outdoor domain is good enough to improve network weights of different architectures trained on ground truth annotated images. Moreover, we demonstrate that this transfer learning scheme can be repeated multiple times iteratively to further improve the accuracy of plane detection and segmentation on monocular images of outdoor scenes.

ÖZET

DERİN ÖĞRENME İLE DÜZLEMSEL GEOMETRİNİN TAHMİNLENMESİ

Sahnelerin geometrik yapılarının anlaşılması bilgisayarlı görünün en eski problemlerinden biridir. Çoğu sahne geometrik yapı hakkında bilgi sağlayan düzlemsel bölge-ler içerir ve bunların otomatik olarak bölütlenmesi birçok bilgisayarlı görü uygulamasında önemli rol oynar. Son yıllarda, parçalı düzlemsel bölütleme yapan evrişimsel sinir ağı mimarileri önerilmiştir. Bunlar, düzlem adaylarını 3B bölütleme yöntemleriyle 3B nokta bulutu geriçatımından üreten geleneksel yaklaşımlardan üstün olmuştur. Fakat, çoğu evrişimsel sinir ağı mimarisi dış sahneler için tasarlanmamış ve eğitilmemiştir çünkü onların elle etiketlenmesinin zaman alıcı bir iş olması eğitim verisi eksikliğine neden olmaktadır. Bu tez çalışmasında, dış sahnelerin hiçbir şekilde elle etiketlemeye ihtiyaç olmadan parçalı düzlem tespiti ve bölütlenmesi için derin öğrenme tabanlı sistem önerilmiş ve geliştirilmiştir. Dış imgelerin mutlak doğru eğitim verilerinin tahminlenmesi için etiketlenmiş verilerle eğitilmiş bir sinir ağı mimarisinden ve Hareket ile Nesne Oluşturma-İkili Çoklu Görüntü ardışık düzeni veya tekli imgelerden derinlik tahminlemesi yapan sinir ağları mimarisinden kullanılarak elde edilen 3B otomatik nokta bulutu geriçatımından yararlanılmıştır. Dış bölgelere ait çeşitli imge kümeleri için elde edilen eğitim verilerinin mutlak doğru olarak etiketlenmiş imgelerle eğitilen farklı sinir ağı mimarilerinin ağırlıklarını yeterince iyi geliştirdiği gösterilmiştir. Buna ek olarak, bu transfer öğrenme düzeninin çoklu olarak tekrarlanabildiği ve dış sahnelerin tekli imgelerinde düzlem tespitinin ve bölütlenmesinin doğruluğunu geliştirdiği gösterilmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

Figure **Page**

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Understanding the geometric structure of any scene is a long-standing goal of Computer Vision. Most scenes include planar regions that are informative for the geometric structure. Extracted features from planar regions and spatial relationships between them enable to model environments. Automatic detection and segmentation of planar regions from images belonging to the scene plays key role in scene reconstruction [30, 35], scene understanding [38, 83], augmented reality [91, 96], and robotics [41, 67]. This makes piece-wise plane reconstruction one of the common vision subproblems. Early solutions to deal with such a problem include reconstructing a 3D point cloud from images with overlapping views and generating plane candidates from the point cloud. Planes are generated by using plane-related cues [29, 86, 49, 94] or with 3D segmentation methods such as robust plane fitting via RANSAC [26]. Generating plane candidates from the 3D point cloud does not always give an accurate segmentation because the point cloud carries high uncertainty at plane boundaries. Furthermore, since the reconstruction quality of the 3D point cloud depends on matching accuracy between images of the scene, textured surfaces are required.

For nearly a decade, deep learning frameworks have been applied to almost all Computer Vision problems for which traditional solutions have weaknesses or could not encapsulate the whole corresponding problem space. The success of deep neural networks comes from the ability to learn from huge amounts of training data. For most of the vision tasks, convolutional neural networks(CNNs) are preferred as a deep module since they are able to extract various features from images while keeping the number of parameters relatively small. Early CNNs [47, 85, 88, 40] are proposed for image classification and/or object detection tasks on different massive datasets such as ImageNet [77] which is composed of over 15 million labeled images with approximately 22.000 categories. The robustness of early CNNs for image classification and object detection gives way to new deep architectures [102, 95, 42] for the same tasks, and allows CNNs to be used for several other vision problems such as object detection [73, 39, 72], semantic segmentation [11, 75, 17], and monocular depth estimation [19, 24, 34].

Human vision has an amazing ability for understanding high-level scene structures and can instantly parse a scene into dominant planes even from a single image. Due to their success in many vision tasks, CNN-based approaches [56, 97, 101, 55] have been

introduced for piece-wise planar segmentation from a single image. These approaches eliminate the need for explicit reconstruction of a 3D point cloud which requires overlapping textured images. Although these CNN architectures outperform traditional methods in terms of indoor reconstruction accuracy, they do not perform well for outdoor scenes. Besides the representative design of the deep neural architecture, the success on indoor scenes comes from the accessibility of large training sets thanks to easy depth sensing with the aid of active sensors. Since such sensors have a limited operating range, manual annotation of outdoor scenes and constructing a large training set for them is a time-consuming task. Transfer learning [84, 82] can be considered to compensate for such lack of data. However, this too requires manually annotating outdoor images individually to increase the transfer performance as much as possible. Consequently, there is a need for transferring learned features from existing piece-wise plane segmentation deep neural architectures to an appropriately collected set of images of outdoor scenes without requiring manual annotation.

In this thesis, we mainly focus on the deep piece-wise planar segmentation of outdoor scenes without requiring manual annotation. We considered existing early geometric and recent deep learning based techniques introduced for the plane reconstruction and proposed a new novel deep learning based framework that solves the deficiencies for piece-wise plane detection and segmentation of outdoor scenes.

In the rest of this chapter, we first present popular applications and products that use piece-wise plane detection and segmentation in Section 1.1. In Section 1.2, we briefly review the related literature. Section 1.3 provides a top-level overview of the novel approach developed in this thesis. Section 1.4 outlines the contents of the remaining chapters.

1.1 Applications

Piece-wise plane detection and segmentation have become essential computer vision tools, especially for augmented reality and robotics applications.

Augmented Reality(AR) refers to placing the virtual objects into a real scene captured by a video camera. It has become very popular in recent years and many mobile applications including social media platforms have been introduced with embedded AR software. Generally, the AR module starts with the reconstruction of the point cloud by guiding the user to move the camera with enough translation on top of the desired region that the virtual object will be placed in order to ensure a dense reconstruction. The accuracy of the plane detection and segmentation determines the quality of the

corresponding application. Missing planes makes the application useless and inaccurate plane boundaries cause poor quality. Samples from the augmented reality application of Google, Google AR, for different desired planar regions to place various virtual objects are shown in Figure 1.1. Not just for entertainment, AR has also been used for other domains such as military, architecture, sports, and fashion.

Figure 1.1.: Samples from Google AR™.

Robotics is a technology that combines computer and mechanical engineering. Computer vision has been incorporated into robotics, which gives rise to the concept of machine vision [45]. Different types of cameras (RGB, depth, and thermal) or sensors (Lidar) embedded into the robot enables it to get visual information from the environment, which can be then evaluated in image processing and computer vision modules of the robot. This opens up the possibility of using machine vision to be widely used in different domains such as military, space, industry, construction, architecture, medicine, warehousing. Furthermore, advances in deep learning solutions for computer vision problems provide self-learning capability for robots with machine vision so that the prospect of

robots "acting like a human" gets closer to becoming a real.

Piece-wise plane detection and segmentation play an important role in robot products with machine vision modules. Boston Dynamics, which is one of the prominent companies that provide industrial robotic solutions, introduced a product called "Spot" for construction that autonomously captures images and video indoors or on challenging exterior sites. Furthermore, it collects 3D data on construction progress with a laser scanner which includes plane information. With this knowledge, it compared the current construction progress with the planned design of the site and detects discrepancies early to minimize rework. Example images from "Spot" are illustrated in Figure 1.2.(a). Figure 1.2.(b) shows the "Limo," the product of AgileX Robotics company providing mobile robot chassis and customized unmanned driving solutions for industry. "Limo" is a mobile robot with artificial intelligence modules including computer vision which provides mapping and navigation, path planning, obstacle detection, and simultaneous localization and mapping(SLAM) tasks. Piece-wise plane detection and segmentation is an essential part of these specific core computer vision modules.

(a) Samples from the product of Boston Dynamics , "Spot", designed for construction industry [3].

(b) Samples from the ,"Limo", which is the customizable mobile robot product of AgileX [2].

Figure 1.2.: Examples from the mobile robots that have the ability of piece-wise plane detection and segmentation.

Consequently, piece-wise plane detection and segmentation are vital computer vision modules for AR applications and robotics. Since both areas have been trendy and constantly evolving, studies on plane detection and segmentation will continue on an ongoing basis especially with deep learning-based methods.

1.2 Related Work

In this section, related studies for piece-wise planar segmentation in the literature are reviewed. In Section 1.2.1, traditional methods are listed which propose different approaches for obtaining planar regions from a constructed 3D point cloud. Convolutional neural architectures designed for piece-wise planar segmentation from a single image are discussed in Section 1.2.2.

1.2.1 Piece-wise Plane Segmentation with Traditional Methods

Traditional piece-wise plane detection and segmentation methods for outdoor scenes require images of multiple views. Furukawa et al. [29] reconstructs 3D oriented points with the aid of a multi-view stereo approach and then generates plane candidates with heuristics and with Markov Random Fields (MRF) optimization. From 3D oriented points, they extract dominant axes and obtain plane hypotheses from the peaks along each dominant axis.

Sinha et al. [86] generate a 3D sparse point cloud with a Structure-from-Motion (SfM) approach. They extract planes from sparse 3D points and line segments. Then, piece-wise planar segmentation is obtained by graph-cut based energy minimization.

Gallup et al. [30] generate plane hypotheses with RANSAC from a set of depth maps. These candidates are refined with the MRF framework to obtain the final result. They define an energy function by considering multi-view photoconsistency besides the color and texture information and get plane assignments by energy minimization through the graph-cuts algorithm.

Oehler et al. [66] introduce a multi-resolution way for extracting planar segments from 3D point clouds. They use coarse to fine strategy in terms of 3D resolutions. They compute surface normals in order to describe surface elements at each resolution level. Surface elements from a coarse resolution that do not belong to any of the planes are clustered with the Hough transform [43] and then extracted connected components

belonging to those clusters are used for the best plane fitting through RANSAC. They measure their performance on indoor datasets composed of Kinect depth images, 3D laser scans, and range images without any generalization to the custom images.

The RANSAC algorithm mainly consists of hypothesis generation and verification steps. For the task of 3D point cloud plane segmentation, hypothesis generation first fit a plane with three randomly selected points by the least-squares method and then verification is done by checking how many of the remaining points in the point cloud can be approximated by the fitted plane. This two-step process is applied for a number of iterations, and the plane that gives the highest number of inliers is assigned as a valid plane on the 3D point cloud. The parameters of the plane are finally computed with all inliers by fitting all of them to it. Then, the same procedure repeats with the outliers based on the previously detected plane until the number of inliers drops down below a certain number or the number of segmented planes reaches the maximally allowed plane count. 3D point cloud plane segmentation procedure is shown in Figure 1.3..

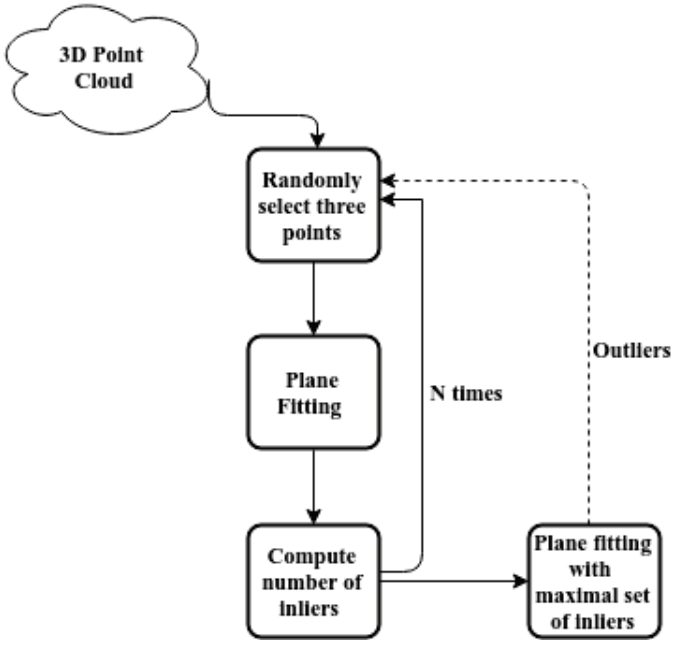
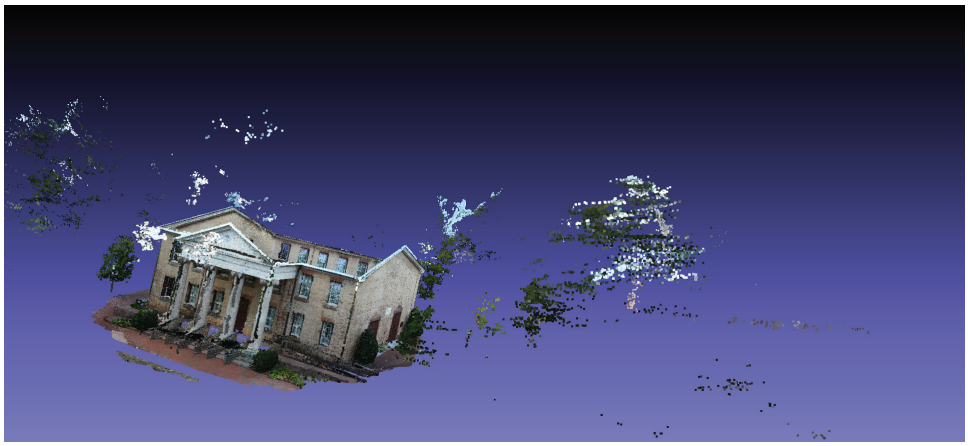


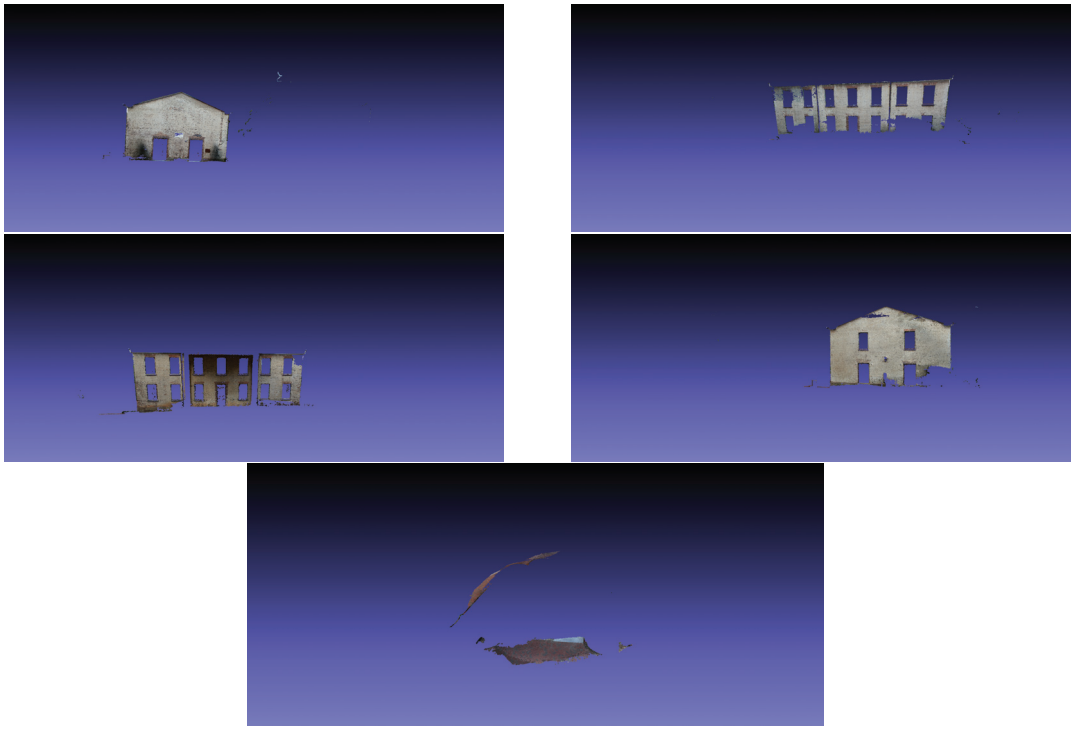
Figure 1.3.: Piece-wise plane segmentation procedure from 3D point cloud with RANSAC algorithm.

An example for piece-wise plane segmentation of a dense 3D point cloud through RANSAC is illustrated in Figure 1.4.. The point cloud is obtained from Gerrard Hall dataset [20] and segmented into planes that belong to different sides of the building and the ground.

Xiao et al. [94] propose individual plane segmentation methods for both structured



(a) 3D reconstruction of Gerrard Hall dataset with COLMAP.



(b) Plane segmentation results with RANSAC applied to the dense 3D point cloud shown in (a).

Figure 1.4.: Illustration of 3D reconstruction of Gerrard Hall dataset and corresponding segmented planes from the 3D point cloud. In (a), a dense 3D reconstruction of the Gerrard Hall dataset is shown. The dense 3D point cloud is segmented into planes by applying RANSAC, which are shown in (b).

and unstructured environments. Their work aims to have an accurate segmentation with a region-growing approach. For a structured environment, they use a sub-window as a growth unit instead of points. On the other hand, for an unstructured environment, both sub-window and single point are used as growth units which are called the hybrid region growing algorithm. Sub-windows are categorized as planar and non-planar according to their shape and only planar sub-windows are taken into account. The performance of the method for an unstructured environment(hybrid region growing algorithm) is evaluated with outdoor datasets. Although it is stated that results are promising, the accuracy of the algorithm depends on the preset sub-window size which makes it less flexible for different environments. Furthermore, region growing segmentation methods are highly sensitive to initialization and inaccurate estimations of the normals around planar region boundaries degrades the performance [37].

Li. et al. [49] introduce an improved RANSAC method based on normal distribution transformation cells(NDT) for 3D point cloud plane segmentation. Instead of straight application of the standard RANSAC algorithm, a 3D point cloud is represented with a set of NDT cells and is modeled with a normal distribution within each cell. Each NDT cell is categorized as either planar or non-planar cells and a planar NDT cell is selected as a minimal sample at each iteration of the algorithm. The performance of the method is evaluated with three indoor datasets and the success of the algorithm is highly dependent on selecting optimal cell size which makes it less flexible for different environments.

Bodis et.al [13] propose a piece-wise planar model and obtains a 3D sparse point cloud from a set of images and solves the reconstruction problem with graph cuts by assigning plane labels to superpixels under the guidance of the point cloud. However, they do not use machine learning to guide the estimation process in textureless regions and the initialization with planes fitted to superpixels is not robust due to noise.

1.2.2 Segmentation with CNN-based Approaches

Recently, deep neural architectures have been designed for piece-wise planar reconstruction from a single image. PlaneNet ([56]) is the first one that can segment a monocular image. It is designed for and trained with images of indoor scenes. It estimates parameters for each detected plane, depth map for the non-planar region, and probabilistic segmentation mask. An input RGB image is given to the Dilated Residual Network(DRN), which is a type of neural network that aims to provide a larger output feature map and gives better estimate of image classification and semantic segmentation. Output feature map is given to the pooling, and convolutional layers and planar region segmentation

and their corresponding parameters are obtained. The overall architecture of PlaneNet is illustrated in Figure 1.5.. The ground-truth data for network training and test are obtained from ScanNet [23] dataset, which consists of large-scale indoor RGB-D video frames. Although PlaneNet outperforms several traditional plane segmentation methods, it has two main limitations. First, the maximum number of planes should be given as prior information, which limits the flexibility of the system. Second, small surfaces are missed, which degrades scene understanding quality.

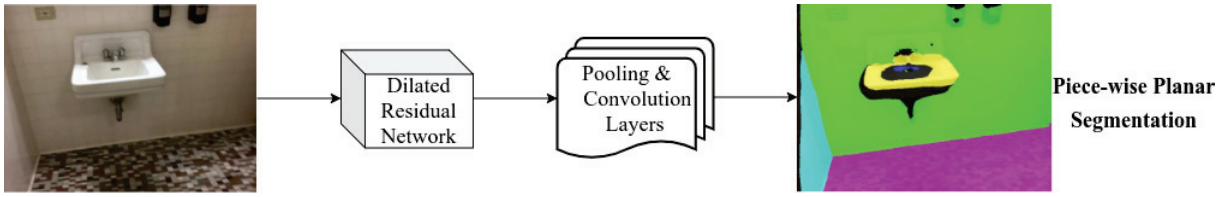


Figure 1.5.: An illustration of PlaneNet architecture. A single RGB image is given to the Dilated Residual Neural Network, which outputs high-resolution features given to the pooling and convolution layers to estimate piece-wise planar segmentation with corresponding plane parameters.

PlaneRecover [97] presents an unsupervised learning approach instead of having manual annotations for 3D plane parameters. The framework has a single convolutional neural network component that outputs plane segmentation besides the non-planar surface identification and plane normals. For plane segmentation, they use an encoder-decoder architecture with skip connections. The network is trained and tested with the SYNTHIA [76] dataset, which is composed of a large number of synthetic images of urban scenes with the corresponding depth maps and pixel-wise semantic annotations and the CityScapes [21] dataset, which includes real street-view video sequences. This deep framework allows at most five planes detected on the given image.

Both PlaneNet and PlaneRecover limit the maximum possible number of planar regions, which degrades applicability in general scenarios. To overcome this, Yu et al. [101] propose an approach based on associative embedding. Their deep framework consists of two stages. In the first phase, a convolutional neural network is trained to map each pixel to an embedding space with the aim of having similar embeddings for pixels belonging to the same planar region. After embeddings are obtained, plane instances are computed by clustering the embedding vectors in planar surfaces via the mean-shift algorithm. In the second stage, the system outputs parameters for each plane instance by taking pixel-level and instance-level consistencies into account. As PlaneNet, they used the ScanNet dataset for ground-truth data acquisition used for both training and test. An arbitrary number of planes can be detected, and small planar surfaces are more likely

detected than PlaneNet.

Recently, PlaneRCNN [55] improves upon PlaneNet. PlaneRCNN is composed of three components. The first component is a plane detection network that is built upon Mask RCNN [39] which is one of the state-of-the-art object detection deep frameworks in the literature. This branch estimates an instance mask for each planar region, plane normal, and depth values for each pixel. Unlike PlaneNet, an arbitrary number of planes can be dealt with. The second branch is designed for segmentation refinement. Here, piece-wise planar segmentation masks that are extracted from the first component are optimized jointly in order to improve them and make them more coherent to the scene. The third component is a warping loss module which aims at enforcing the consistency of reconstructions by improving the accuracy of parameters of plane and depth map during training by taking into account a neighbor view of the corresponding scene. As in PlaneNet, ground-truth data for training and test are obtained from ScanNet dataset. PlaneRCNN outperforms PlaneNet in terms of plane detection, plane segmentation, and depth map estimation accuracy. The architecture of PlaneRCNN is shown in Figure 1.6..

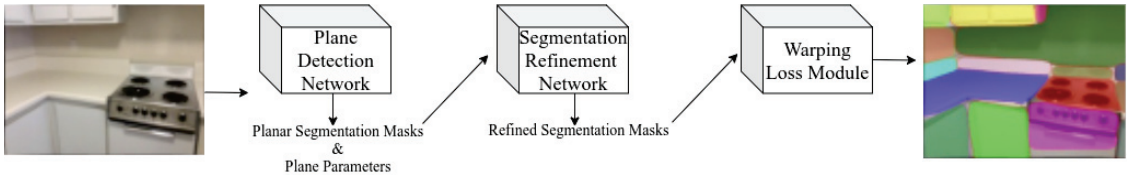


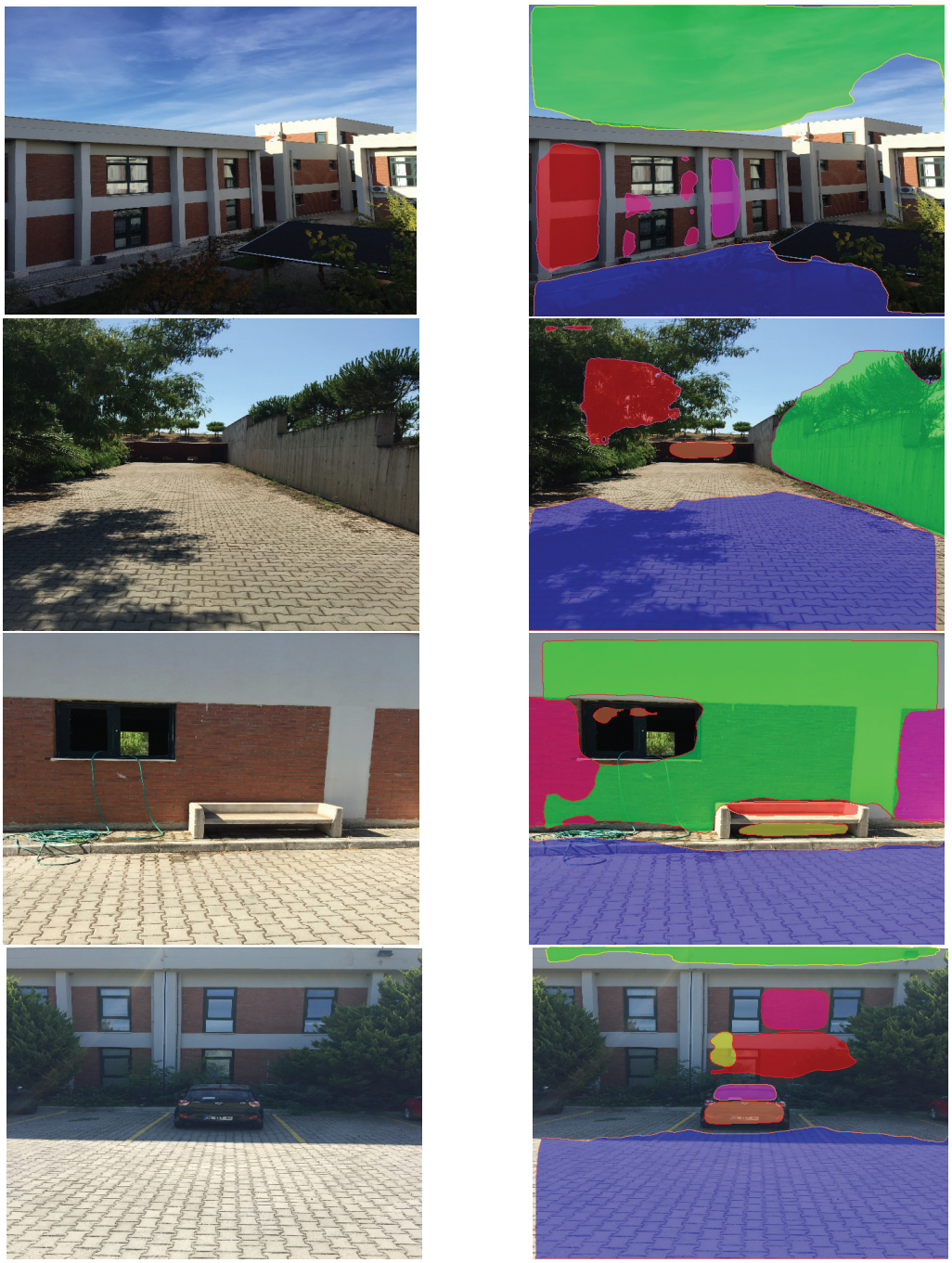
Figure 1.6.: An illustration of PlaneRCNN architecture. This deep learning framework for piecewise plane segmentation is composed of three components. A plane detection network takes a single RGB image and produces segmentation masks, plane parameters, and a depth map for an entire image. Segmentation refinement network takes estimated instance segmentation masks and refines them by joint optimization. Refined segmentation masks are given to the warping loss module, which aims at enforcing the consistency of reconstructed planes with a neighbor view of the corresponding scene.

Although PlaneRCNN is designed for and trained with images of indoor scenes like PlaneNet, piece-wise planar segmentation accuracy for outdoor images is slightly better. However, as we will demonstrate in Section 2 its performance is still constrained by the features learned on indoor images. Piece-wise plane reconstruction of PlaneRCNN for unseen indoor and outdoor scenes are shown in Figure 1.7. and Figure 1.8. respectively. Although piece-wise planar segmentation of PlaneRCNN looks accurate for images of different unseen indoor scenes, it misses most of the planar regions, and undersegments/oversegments found plane assignments.

Yang et al. introduce a CNN named StruMonoNet [99] for monocular 3D estimation of indoor imagery. Besides the piece-wise plane reconstruction, the network also

(a) Input indoor RGB image (b) PlaneRCNN segmentation

Figure 1.7.: Piece-wise plane segmentation of PlaneRCNN for unseen images of different indoor scenes. It finds most of the planes with highly accurate segmentation boundaries.



(a) Input outdoor RGB image

(b) PlaneRCNN segmentation

Figure 1.8.: Piece-wise plane segmentation of PlaneRCNN for unseen images of different outdoor scenes. Besides most of the planar regions are missed and it oversegments/undersegments found ones.

extracts the relations between estimated planes, such as adjacency, perpendicularity, and parallelism. The network is composed of three main components. In the first component, surface elements are predicted, including depth, normal, visual descriptor, and boundary pixels. Planes are detected in the second component. The third one is the geometric rectification module which takes the surface elements and detected planes as an input. The module refines the surface elements and gives the relations between detected planes.

Liu et al. propose PlaneMVS [58], a two-branch deep framework for 3D plane reconstruction from images of multiple views belonging to various indoor scenes. The plane MVS branch gives plane parameters while the plane detection branch outputs plane segmentation masks. Both outputs are evaluated in a single loss function to improve the prediction of each other.

1.3 Research Summary and Key Findings

The research conducted in this thesis mainly concentrates on planar geometry estimation with deep learning. Our primary purpose is to develop a deep learning based framework for the piece-wise plane detection and segmentation of outdoor scenes without requiring manual annotation. The challenging part of the problem comes from the need for automatically generated piece-wise plane annotation for images belonging to outdoor scenes. To overcome this, we propose two novel methods that combine the traditional and recent CNN-based solutions for the piece-wise plane detection and segmentation problem. The first proposed method exploits a deep neural network trained on manually annotated images(recent CNN-based approaches) and an automatically reconstructed point cloud(traditional approaches) to estimate the training ground truth labels on the outdoor images in an energy minimization framework. We demonstrate that a piece-wise plane reconstruction network trained on indoor images can be adapted to outdoor scenes for the task of piece-wise planar segmentation without requiring manual annotations. We also propose a second method that exploits deep monocular depth estimation to obtain a 3D dense point cloud instead of the standard Structure from Motion and Multi-View Stereo pipeline. This makes our approach customizable to any appropriate outdoor dataset. With this improvement, we also apply our proposed novel approach to the Unmanned Aerial Vehicle(UAV) outdoor imagery for ground plane estimation without requiring manual annotations. We employ our transfer learning scheme between different UAV outdoor image datasets collected from various altitudes in several environments by providing ground truth training targets just for a single set of images. The main contribution of the thesis can be summarized as follows:

- We combine the traditional and recent approaches for the task of piece-wise plane detection and segmentation.
- We develop a transfer learning scheme from a piece-wise plane reconstruction network trained on annotated indoor images to outdoor domain without requiring manual annotation.
- We formulate an energy function initialized with the segmentation estimation of a network trained on annotated images and minimized under the guidance of 3D dense point cloud at training time which generates ground-truth training labels automatically.
- We iterate the transfer learning scheme that alternates between improving network weights in the outdoor domain and generating approximate ground truth training labels.

- We demonstrate that a piece-wise plane reconstruction network trained on indoor images can be adapted to outdoor scenes for the task of piece-wise planar segmentation without requiring manual annotations.
- We integrate a deep monocular depth estimation network to our approach for generating 3D dense point cloud from the estimated depth.
- We employ our novel proposed approach to ground plane estimation on UAV outdoor imagery.
- We show that a semantic segmentation network trained on UAV outdoor images can be adapted to any other UAV outdoor imagery for the task of ground plane estimation without requiring manual labor.

Figure 1.9. shows schematically how the work done for this thesis contributes to solving the common computer vision problem of piece-wise plane detection and segmentation. In the literature, relevant studies can be grouped as traditional methods and CNN-based approaches. In this thesis, we propose and develop a novel deep learning based method by combining CNN-based and traditional approaches for piece-wise plane detection and segmentation for outdoor imagery without requiring manual annotation.

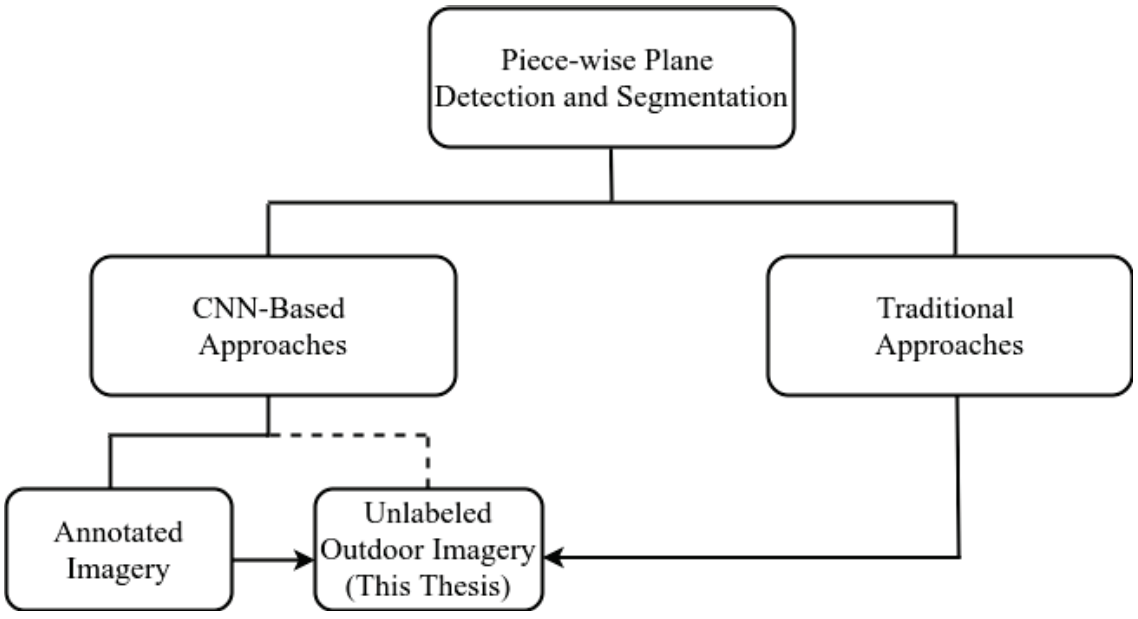


Figure 1.9.: The top view of our thesis work from piece-wise plane detection and segmentation perspective.

1.4 Outline of the Thesis

In Chapter 2, we provide the details of our primary approach including related work and corresponding background information. We also show the performance of our approach with a different set of experiments. Chapter 3 gives the proposed idea about modification to our main approach so that requirement of overlapping images in the input outdoor dataset is removed. We compare this modified approach with the former one and the baseline with different plane segmentation metrics. Chapter 4 includes the study of employing our proposed novel approach to ground plane estimation on UAV outdoor imagery. We demonstrate that the proposed method can be applied to other plane estimation problems for different outdoor imagery domains. Chapter 5 concludes this thesis.

CHAPTER 2

IMPROVING OUTDOOR PLANE ESTIMATION WITHOUT MANUAL SUPERVISION

2.1 Introduction

The automatic segmentation of planar regions, which are present in most scenes and offer details on the scene's geometric structure, has long been an objective of Computer Vision. Multiple views are required for this task in traditional methods ([29, 30, 86]). Generally, they first reconstruct 3D point clouds from the images and generate plane candidates with 3D segmentation methods such as robust plane fitting via RANSAC [26]. However, an accurate segmentation might not be obtained since the plane boundary in the 3D point cloud carries high uncertainty. In addition, textured planar surfaces are required for using the point clouds to apply stereo matching in reconstruction.

Recently, Convolutional Neural Network (CNN) based approaches ([56, 97, 101, 55]) have been introduced for piece-wise plane detection and segmentation from a single image without explicit reconstruction of a 3D point cloud. Although they outperform traditional approaches in terms of reconstruction quality, most of them are designed for and trained with indoor imagery and perform poorly for outdoor scenes.

The advantage of indoor scenes comes from the easy depth sensing provided by active sensors. However, such sensors have a limited operating range that makes manual annotation of images of outdoor scenes a time-consuming task, preventing replicating the success on indoor datasets. To compensate the lack of large training sets, transfer learning can be considered. However, this also requires manually annotated outdoor image datasets to transfer the features. So, transferring features from the existing networks to a set of outdoor images is required without manual annotation.

In this part of the thesis work, we propose and develop such an approach that requires a collected training set of outdoor images that can be processed by a structure from motion - multi view stereo pipeline to produce a dense 3D point cloud. Instead of directly generating plane candidates from the point cloud as in traditional methods, we exploit the dense 3D point cloud as a weak supervision signal to improve piece-wise plane segmentation quality. Our approach is developed on the idea of combining such a point

cloud and the output from a state-of-the-art plane segmentation network trained on indoor images such as PlaneRCNN [55], an approximate but high quality estimate of ground truth annotations on the outdoor images can be obtained. We use this estimate as training targets to improve the network weights, achieving transfer without manual labeling. In addition, we show that iteratively applying this process further improves piece-wise plane segmentation quality. Once the training and the feature transfer is completed, during test time, our approach can detect and segment planar regions on a given monocular outdoor image with a much greater accuracy than a network trained on indoor images. We formalize our approach in Section 2.3 and provide experimental results on several datasets collected outdoors.

Our main contributions can be summarized as follows:

- We combine the traditional and recent approaches for the task of piece-wise plane reconstruction by achieving feature transfer under the guidance of 3D dense point cloud at training time with the initialization provided by a network trained on indoor images.
- We formulate an approximate and iterative transfer scheme that alternates between estimating ground truth labels and improving network weights in the target domain.
- We demonstrate that PlaneRCNN can be adapted to outdoor scenes for the task of piece-wise planar reconstruction without requiring manual annotations.

2.2 Related Work

Traditional piece-wise planar reconstruction methods([29, 30, 86]) for outdoor scenes require images of multiple views. Furukawa et al. [29] reconstructs 3D oriented points with the aid of a multiview stereo approach and then generates plane candidates with heuristics and with Markov Random Fields (MRF) optimization. Gallup et.al [30] generates plane hypotheses with RANSAC from a set of depth maps. These candidates are refined with the MRF framework to obtain the final result. Sinha [86] generates a 3D sparse point cloud with a Structure-from-Motion (SfM) approach and extracts 3D line segments which are used in a graph cut formulation.

Bodis et al. [13] proposes a piece-wise planar model and obtains a 3D sparse point cloud from a set of images and solves the reconstruction problem with graph cuts by assigning plane labels to superpixels under the guidance of the point cloud. However, they do not use machine learning to regularize the estimation in textureless regions and initialization with planes fitted to superpixels is not robust due to noise.

Recently, deep neural architectures are trained for piece-wise planar reconstruction from a single image. PlaneNet ([56]) is the first one that can segment a monocular image. It is designed for and trained with the images of indoor scenes. PlaneRecover([97]) presents an unsupervised learning approach instead of having manual annotations for 3D plane parameters. It is trained with a synthetic outdoor dataset. Both PlaneNet and PlaneRecover limits the maximum possible number of planar regions which degrades applicability in general scenarios. To overcome this, [101] proposes an approach based on associative embedding. Recently, PlaneRCNN ([55]) improves upon PlaneNet. Although, PlaneRCNN is designed for and trained with the images of indoor scenes like PlaneNet, piece-wise planar segmentation accuracy for outdoor images is slightly better. However, as we will demonstrate its performance is still constrained by the features learned on indoor images.

We combine ideas from both traditional approaches and neural network architectures. An automatically reconstructured point cloud is exploited only during training to estimate ground truth segmentations on the outdoor datasets. As a result, once training is completed, we do not require multiple images and our approach is able to reconstruct planar regions even in less textured areas. Moreover, since it adapts to the image features on an outdoor dataset, its plane detection and segmentation performance surpasses the existing networks trained indoors.

Finally, Zeng et. al [103] showed that integrating geometric cues such as vanishing points and lines to constrain the plane segmentation results improves both segmentation quality and estimated plane parameters. Our approach relies on similar reasoning to exploit geometry of the scene to ease domain transfer from indoor to outdoor imagery. It might be possible to fuse our approach with that of [103], either to improve the output of our method using perspective cues or to provide stronger training for the approach of [103] by integrating larger amounts of unlabeled outdoor training data.

2.3 Transfer Learning without Manual Supervision

Training a neural network for piece-wise plane estimation on outdoor images requires a set of training outdoor images $\mathtt{D}^{\mathtt{out}}$ with annotated ground truth training targets $\mathtt{T}^{\mathtt{out}}$. The training targets are composed of segmentation masks $\mathtt{S}^{\mathtt{out}}$ and plane equations $\pi^{\mathtt{out}}$, $\mathtt{T}^{\mathtt{out}} = (\mathtt{S}^{\mathtt{out}}, \pi^{\mathtt{out}})$. The neural network is trained by searching for a set of weights $\mathbf{w}$ that minimizes a loss function $\mathcal{L}(\mathbf{w})$ that takes the network output and the ground-truth training targets as parameters

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \mathcal{L}\left(f(\mathtt{D}^{\mathtt{out}} \mid \mathbf{w}),\, \mathtt{T}^{\mathtt{out}}\right),$$

where $\mathcal{L}(\mathbf{w})$ measures the disagreement between $\mathtt{T}^{\mathtt{out}}$ and the network output $f(\mathtt{D}^{\mathtt{out}} \mid \mathbf{w})$ on the training images.

Our assumption is that the ground truth training targets $\mathtt{T}^{\mathtt{out}}$ are not available, but we have network weights $\mathbf{w}_{\mathtt{PlaneRCNN}}$ trained on indoor images and we can obtain a point cloud $\mathtt{P}^{\mathtt{MVS}}$ by using a state-of-the-art Structure from Motion (SfM) and Multiview Stereo (MVS) system such as COLMAP ([78, 79]) on the collected outdoor image set.

Our approach is developed based on the idea of solving the minimization problem given in the above approximately by first estimating the training targets based on $\mathbf{w}_{\mathtt{PlaneRCNN}}$ and $\mathtt{P}^{\mathtt{MVS}}$, and then improving the weights $\mathbf{w}$ by using this estimated $\tilde{\mathtt{T}}^{\mathtt{out}}$ which provides weak-supervision:

$$\tilde{\mathtt{T}}^{\mathtt{out}} = \arg\min_{\mathtt{T}^{\mathtt{out}}} E\left(\mathtt{T}^{\mathtt{out}} \mid \mathtt{P}^{\mathtt{MVS}},\, f(\mathtt{D}^{\mathtt{out}} \mid \mathbf{w}_{\mathtt{PlaneRCNN}})\right)$$
$$\tilde{\mathbf{w}}^{*} = \arg\min_{\mathbf{w}} \mathcal{L}\left(f(\mathtt{D}^{\mathtt{out}} \mid \mathbf{w}),\, \tilde{\mathtt{T}}^{\mathtt{out}}\right),$$

where $E\left(\mathtt{T}^{\mathtt{out}}\right)$ measures the mismatch between the estimated training targets and the information provided by the 3D point cloud $\mathtt{P}^{\mathtt{MVS}}$ and the network output using pretrained weights $f(\mathtt{D}^{\mathtt{out}} \mid \mathbf{w}_{\mathtt{PlaneRCNN}})$. It also ensures that the segmentation masks are smooth. The exact formulation of the energy function is given in more detail in Section 2.3.3.

Better piece-wise plane reconstruction quality is expected than what is initially possible with $\mathbf{w}_{\mathtt{PlaneRCNN}}$ by the new set of weights computed after solving the two minimization problems above. The training targets can be reestimated which implicitly provides much representative and informative set of weights than the previous one. This leads to apply the idea iteratively that alternates between the estimation of training targets and the optimization of the network weights based on the last estimate:

$$\tilde{\mathbf{w}}_0 = \mathbf{w}_{\mathtt{PlaneRCNN}}$$
$$\tilde{\mathtt{T}}^{\mathtt{out}}_{i+1} = \arg\min_{\mathtt{T}^{\mathtt{out}}} E\left(\mathtt{T}^{\mathtt{out}} \mid \mathtt{P}^{\mathtt{MVS}},\, f(\mathtt{D}^{\mathtt{out}} \mid \tilde{\mathbf{w}}_i)\right)$$
$$\tilde{\mathbf{w}}_{i+1} = \arg\min_{\mathbf{w}} \mathcal{L}\left(f(\mathtt{D}^{\mathtt{out}} \mid \mathbf{w}),\, \tilde{\mathtt{T}}^{\mathtt{out}}_{i+1}\right),$$

As it is stated in the above, training targets are composed of plane segmentation masks and plane parameters. Plane parameters can easily estimated based on the information provided by the 3D point cloud and the segmentation masks. We just assign each 3D point to one of the planes based on its projection into the segmentation masks with the camera rotation matrices and translations estimated during the SfM stage. After corresponding 3D points are determined for each plane, we apply robust least square fitting to estimate a better set of parameters. This makes energy minimization stage in our iterative transfer learning scheme is estimating the segmentation masks. Once piece-wise plane

segmentation masks are estimated, plane equations are computed as it is explained above:

$$
\begin{aligned}
\tilde{\mathbf{w}}_0 &= \mathbf{w}_{\texttt{PlaneRCNN}} \\
\tilde{\mathtt{S}}_{i+1}^{\mathtt{out}} &= \underset{\mathtt{S}^{\mathtt{out}}}{\arg\min}\ E\left(\mathtt{S}^{\mathtt{out}} \mid \mathtt{P}^{\mathtt{MVS}},\ f(\mathtt{D}^{\mathtt{out}} \mid \tilde{\mathbf{w}}_i)\right) \\
\tilde{\pi}_{i+1}^{\mathtt{out}} &= \mathtt{LSQ}\left(\mathtt{P}^{\mathtt{MVS}},\ \tilde{\mathtt{S}}_{i+1}^{\mathtt{out}} \mid \mathtt{R}^{\mathtt{SFM}}, \mathtt{t}^{\mathtt{SFM}}\right) \\
\tilde{\mathtt{T}}_{i+1}^{\mathtt{out}} &= \left(\tilde{\mathtt{S}}_{i+1}^{\mathtt{out}}, \tilde{\pi}_{i+1}^{\mathtt{out}}\right) \\
\tilde{\mathbf{w}}_{i+1} &= \underset{\mathbf{w}}{\arg\min}\ \mathcal{L}\left(f(\mathtt{D}^{\mathtt{out}} \mid \mathbf{w}),\ \tilde{\mathtt{T}}_{i+1}^{\mathtt{out}}\right),
\end{aligned}
$$

As Figure 2.1. illustrates, we propose and develop an iterative transfer learning scheme in which a neural network is trained for outdoor plane estimation and segmentation without requiring annotated ground truth information. We exploit a network and its pretrained weights computed for the same task but obtained from a training set of indoor images. Our proposed iterative transfer learning scheme depends on weak supervision provided by the point cloud $\mathtt{P}^{\mathtt{MVS}}$ that can be computed automatically under some mild assumptions about the training set, such as overlap of viewpoints and presence of textured regions. In the experiments section, we show that this approximate training scheme successfully improves the quality of estimated planes and segmentation masks. In the following, we present the details of each stage in the proposed approach.

2.3.1 3D Point Cloud Acquisition from a Set of Images

Reconstruction of a 3D model from a set of unordered images belonging to real scenes has been widely used in many computer vision applications such as augmented reality [10, 9], motion capture [63], and SLAM [25, 65]. A dense 3D point cloud from a collection of images can be obtained with a Structure From Motion(SfM) [46] - Multi-View Stereo(MVS) [80] pipeline.

2.3.1.1 Structure From Motion(SfM)

SfM is the process of obtaining 3D reconstruction(structure) from a collection of images. A set of overlapping images from different viewpoints of the same scene are given as input. The output is camera parameters for each input image and a sparse A 3D point cloud is composed of a set of points visible in a subset of input image collection. SfM is a sequential process that is mainly composed of three steps:

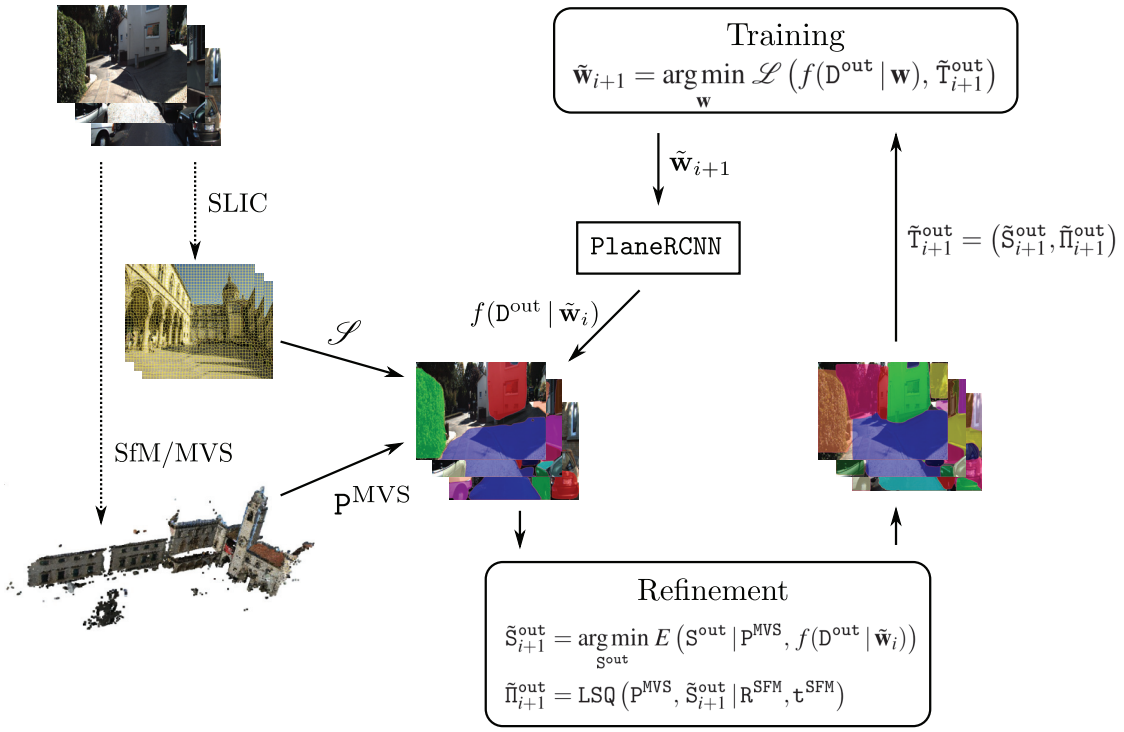


Figure 2.1.: Proposed iterative transfer learning approach. We preprocess the outdoor images to extract a point cloud $\mathtt{P}^{\mathtt{MVS}}$ and a set of SLIC superpixels $\mathcal{S}$. Using these data, we initialize segmentation maps based on the current network output with weights $\tilde{\mathbf{w}}_i$. An energy based minimization problem is solved to refine this crude initialization into an estimated set of training targets $\tilde{\mathtt{T}}^{\mathtt{out}}_{i+1}$. The network is then trained to minimize a loss function on these training targets, yielding improved network weights $\tilde{\mathbf{w}}_{i+1}$. These new weights provide a better initialization, so we can repeat the process multiple times. Note that the whole process is automated and the point cloud is only used in the training phase.

1. **Feature Extraction and Matching:** For each input image, a set of local features are detected. Then, descriptors are computed for the features, and a set of potentially overlapping image pairs are found by finding feature correspondences based on their descriptors.

2. **Final Image Pairs Selection:** In Step 1, image pairs are decided according to their appearance. So, in order to guarantee that matches belong to the same scene point, a geometric verification is done. Potential matches are verified by estimating the geometric transformation between images through features by using projective geometry. Geometrically verified matches are used for the remaining iterative process.

3. **Reconstruction:** The output of Step 2, which is the graph of the verified image pairs are given as the input for the reconstruction.

   3.1 **Model Initialization:** The 3D model is initialized by selecting two views from the graph and obtaining reconstruction from them. At the end of this step, camera poses and triangulated 3D points from feature correspondences are obtained.

3.2 **Image Registration:** A new image from the input set can be registered to the model by solving the PnP [27] problem with the input of 2D-3D correspondences obtained from the already reconstructed model. The solution of the PnP problem gives the camera pose of the newly registered image.

3.3 **Triangulation:** After the camera pose is computed for the already registered image, new feature correspondences can be used for triangulation and new scene points are added to the 3D model.

3.4 **Bundle Adjustment:** The Computation of both cameras poses and scene points can be erroneous. In order to refine the estimation of model parameters, bundle adjustment is applied in order to minimize the reprojection error for both all camera poses and scene points.

An overall procedure of SfM is illustrated in Figure 2.2..

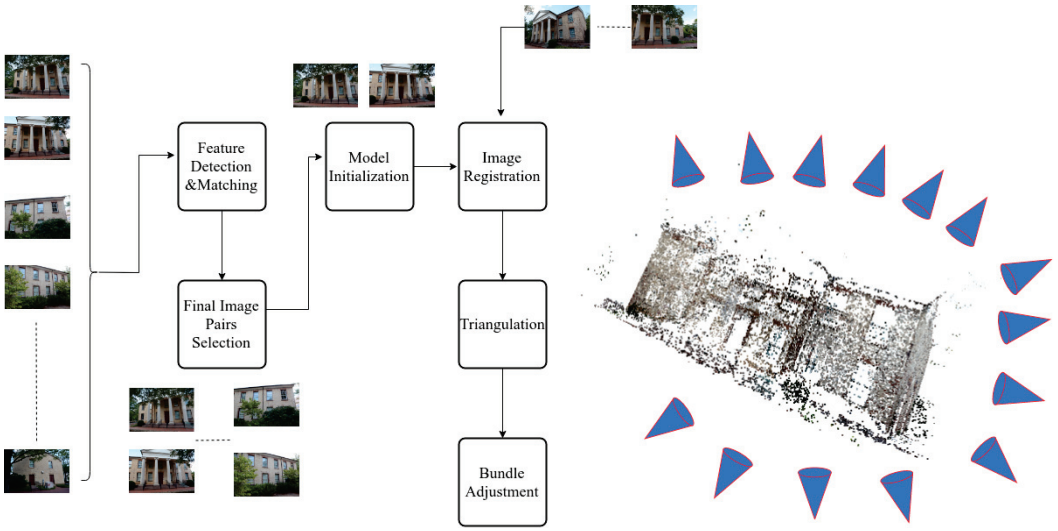


Figure 2.2.: An overall procedure of SfM. Local features are detected on each of the input images and feature correspondences between image pairs are found. From the set of potentially overlapping images, geometrically verified ones are selected and are given to the reconstruction process as a graph. Reconstruction starts with a 3D model initialization from the image pair selected from the graph. At the end of the SfM, the output is the camera pose for each of the input images and the sparse 3D point cloud.

2.3.1.2 Multi-view Stereo(MVS)

MVS is the process of obtaining a 3D dense point cloud by using the input images. Camera poses information per each image and sparse point clouds are given by the SfM algorithm. A general view of the MVS is illustrated in Figure 2.3.. Since camera parameters are known, solving the 3D geometry of the scene can be evaluated as finding feature matches between input images. Furthermore, that finding correspondence problem can be reduced to one-dimensional search by computing the epipolar geometry. After finding correspondences, depth and normal maps for every image are computed and then a 3D dense point cloud is constructed by fusing them.

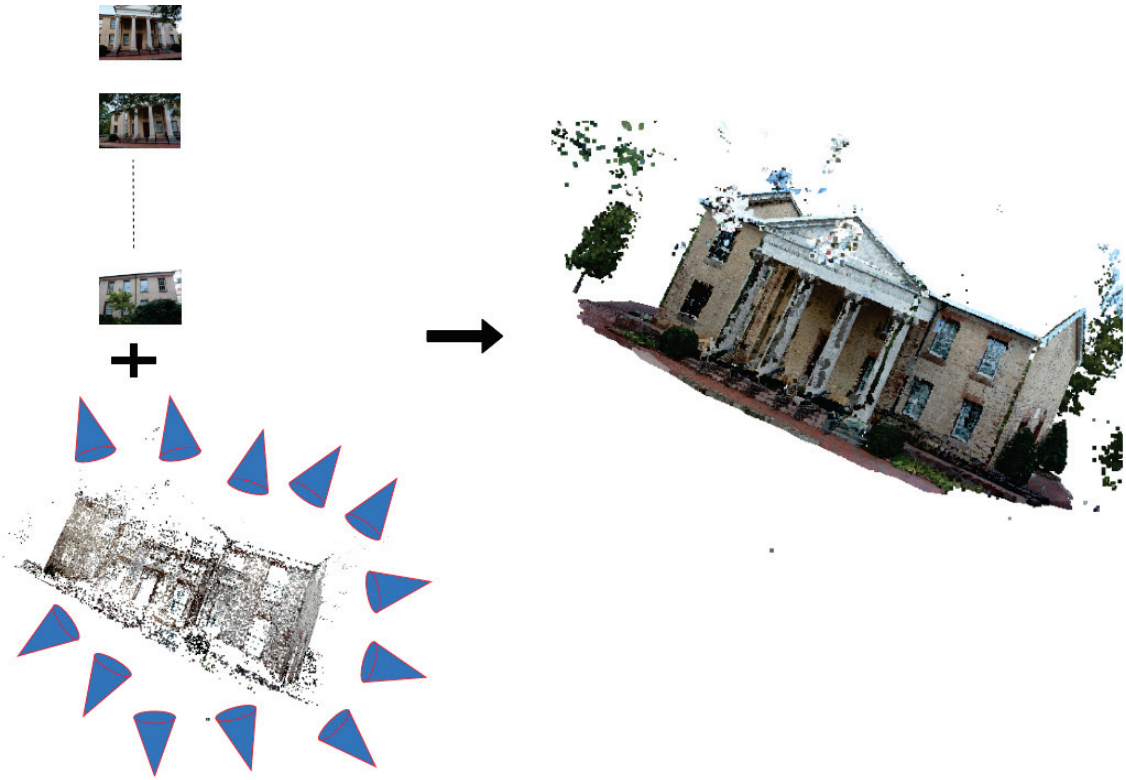
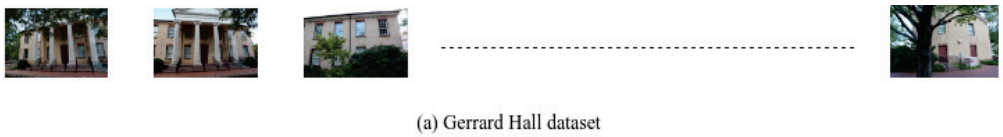


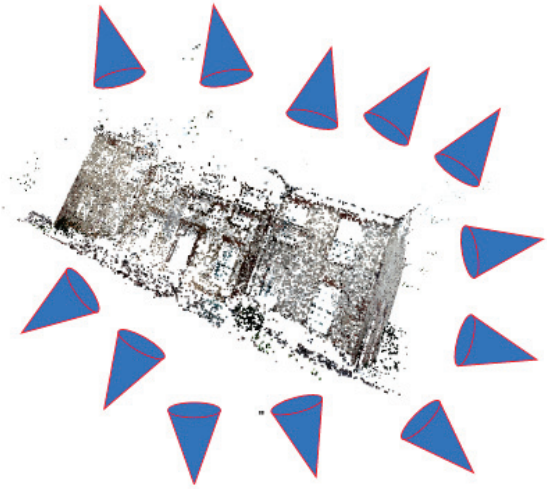
Figure 2.3.: A general view for the MVS. It takes the output of the SfM which are camera poses for each image and 3D sparse point cloud besides the input image collection. After finding correspondences between images with an one-dimensional search via epipolar geometry computation, depth and normal maps for every image are computed and then 3D dense point cloud is constructed by fusing them.

2.3.1.3 SfM-MVS pipeline through COLMAP

COLMAP is a software with a graphical and command-line interface that provides SfM-MVS pipeline. It enables different reconstruction settings for ordered and unordered image collections. In this thesis work, in order to obtain 3D dense point cloud, COLMAP is used for SfM-MVS pipeline for which example outputs are shown for Gerrard Hall dataset in Figure 2.4..



(a) Gerrard Hall dataset



(b) SfM output of Gerrard Hall dataset with COLMAP

(c) MVS output of Gerrard Hall dataset with COLMAP

Figure 2.4.: SfM-MVS pipeline for Gerrard Hall dataset with COLMAP.

2.3.2 Estimation of the Initial Segmentation Masks

To refine the network weights, we exploit a point cloud $\mathrm{P}^{\mathrm{MVS}}$ which provides weak supervision and is obtained from the training images via SfM-MVS pipeline. From the $\mathrm{P}^{\mathrm{MVS}}$ it is possible to estimate the planar regions in the scene by robust geometric fitting using standard least squares estimation and RANSAC.

Although the reconstructed point cloud $\mathtt{P}^{\mathtt{MVS}}$ carries information about planar regions on the scene, there are two drawbacks to computing segmentation masks and plane equations directly from it by naive plane fitting to the 3D points. As a first, the estimated planes infinite extent and directly computation of segmentation boundaries with plane fitting is a challenging problem. Secondly, SfM-MVS pipeline requires textured surfaces so the resulting point cloud might be partial and some planar surfaces can not be reconstructed. Due to these drawbacks, we formulate estimation of the segmentation masks as an energy minimization problem. After the good segmentation masks are estimated, plane equations are refined easily by using the 3D points.

We formulate the estimation of segmentation masks as a min-cut problem which is solved by graph cuts ([15]). In order to make the problem feasible, we extract SLIC superpixels ([4, 5, 57]) and solve the problem for the superpixel labels instead of each pixel. As a result, the segmentation mask $\mathtt{S}^{\mathtt{out}}$ is given as a set of label assignments $\{l_s : \forall s \in \mathcal{S}\}$, where $\mathcal{S}$ is the set of extracted superpixels and the labels $l_s \in \{-1, 0, 1, \ldots, K-1\}$ are selected from a set of $K+1$ possibilities. The labels with non-negative indices $j, j = 0, \ldots, K-1$ represent assignment to one of the possible planes π^j in the network output and the label -1 represents non-planar regions that we denote as π^{-1} for notational convenience. The formulation using superpixels allows us to adjust the granularity of the estimated ground truth to the density of the reconstructed point cloud and the resolution of the training images.

We calculate an initial set of label assignments before we estimate the segmentation labels from energy minimization framework. First, we project each 3D point $p \in \mathtt{P}^{\mathtt{MVS}}$ into the image and assign it to the superpixel it falls into based on superpixel boundaries. Similarly, we assign each projected point to a plane using the per-pixel segmentation masks estimated by the current network weights $f(\mathtt{D}^{\mathtt{out}} \mid \tilde{\mathbf{w}}_i)$. Within each superpixel s, each projected point votes for its assigned plane. Each superpixel is assigned an initial label $\hat{l}_s$ corresponding to the plane $\pi^{\hat{l}_s}$ that received the majority of the votes. The energy formulation described below ensures that the initial assignments $\{\hat{l}_s\}$ are also taken into account.

2.3.3 Updating the Segmentation Masks by Energy Minimization

Given the point cloud $\mathtt{P}^{\mathtt{MVS}}$ and the network weights $\tilde{\mathbf{w}}_i$ obtained in the last iteration, the energy of a superpixel segmentation mask $\mathtt{S}^{\mathtt{out}} = \{l_s\}$ is defined as follows

$$E\left(\mathtt{S}^{\mathtt{out}}\right) = \sum_{s \in \mathcal{S}} E_d\left(l_s \mid \mathtt{P}^{\mathtt{MVS}},\ f(\mathtt{D}^{\mathtt{out}} \mid \tilde{\mathbf{w}}_i)\right) + \lambda_s \sum_{(s,t) \in \mathcal{N}_{\mathcal{S}}} E_s(l_s, l_t \mid \mathtt{P}^{\mathtt{MVS}},\ f(\mathtt{D}^{\mathtt{out}} \mid \tilde{\mathbf{w}}_i)),$$

where $\mathcal{N}_{\mathcal{S}}$ is the set of neighboring superpixels.

The energy data term $E_d\left(l_s \mid \mathtt{P}^{\mathtt{MVS}},\ f(\mathtt{D}^{\mathtt{out}} \mid \tilde{\mathbf{w}}_i)\right)$ measures the discrepancy between a given superpixel label l_s and the point cloud $\mathtt{P}^{\mathtt{MVS}}$. It depends on two components $E_{\text{support}}(l_s \mid \mathtt{P}^{\mathtt{MVS}})$ and $E_{\text{distance}}(l_s \mid \mathtt{P}^{\mathtt{MVS}})$. These components are combined in a weighted fashion as follows:

$$E_d(l_s) = \left(\alpha_1 + \delta(l_s - \hat{l}_s)\right) E_{\text{support}}(l_s \mid \mathtt{P}^{\mathtt{MVS}}) + \left(\alpha_2 + \delta(l_s - \hat{l}_s))\right) E_{\text{distance}}(l_s \mid \mathtt{P}^{\mathtt{MVS}}),$$

where α_1 and α_2 are scalar constants. $\delta(l_s - \hat{l}_s)$ is equal to zero whenever the new label is the same as the initial assignment, it is equal to one otherwise. The delta terms increase the cost of assignments that change the initial labels that were calculated based on the current network output. This ensures that after the energy minimization, the estimated labels will change only when the 3D points of $\mathtt{P}^{\mathtt{MVS}}$ consistently get assigned to a plane other than the one indicated by the initial label.

$E_{\text{support}}(l_s \mid \mathtt{P}^{\mathtt{MVS}})$ measures the ratio of projected 3D points assigned to the same plane corresponding to the label l_s. It is computed as

$$E_{\text{support}}(l_s \mid \mathtt{P}^{\mathtt{MVS}}) = \frac{n_t - n_s}{n_t},$$

where n_s is the number of projected points in the superpixel assigned to plane π^{l_s} and n_t is the total number of projected points in the superpixel.

$E_{\text{distance}}(l_s \mid \mathtt{P}^{\mathtt{MVS}})$ measures the average distance of projected 3D points to the plane corresponding to the label l_s. It is computed as

$$E_{\text{distance}}(l_s \mid \mathtt{P}^{\mathtt{MVS}}) = \frac{1}{n_t} \sum_{p \in s} d(\pi^{l_s}, p),$$

where $d(\pi^{l_s}, p)$ is the 3D Euclidean distance between the 3D point p and plane π^{l_s}.

The smoothness data term $E_s(l_s, l_t \mid \mathtt{P}^{\mathtt{MVS}},\ f(\mathtt{D}^{\mathtt{out}} \mid \tilde{\mathbf{w}}_i))$ ensures that the estimated labels respect image color and depth information which regularizes the problem by constraining the labels of superpixels with a small number of projected 3D points. It is

calculated as

$$E_s(l_s, l_t) = E_{\text{color}}\left(l_s, l_t \mid \mathtt{P}^{\mathtt{MVS}},\ f(\mathtt{D}^{\mathtt{out}} \mid \tilde{\mathbf{w}}_i)\right) + \alpha_3 E_{\text{depth}}\left(l_s, l_t \mid \mathtt{P}^{\mathtt{MVS}},\ f(\mathtt{D}^{\mathtt{out}} \mid \tilde{\mathbf{w}}_i)\right),$$

where $E_{\text{color}}(l_s, l_t)$ penalizes label changes over smooth intensity regions and $E_{\text{depth}}(l_s, l_t)$ penalizes label changes over regions of similar depth. They are calculated as

$$E_{\text{color}}(l_s, l_t) = \{\exp(-\Delta_c), if\ l_s \neq l_t\}$$

and

$$E_{\text{depth}}(l_s, l_t) = \{\exp(-\Delta_d), if\ l_s \neq l_t\}$$

where Δ_c is the difference between mean intensity values (average of color channels) over superpixels s and t, and Δ_d is the difference between mean depth values of 3D points projected into superpixels s and t.

2.3.3.1 Energy Minimization through Graph-Cuts with Alpha-Expansion Algorithm

The energy function E which is explained in detail from beginning of Section 2.3 is minimized trough graph-cuts with the alpha-expansion algorithm [15]. The algorithm approximately minimizes the energy function for an arbitrary set of labels. It is designed based on α-expansion moves in which any set of image elements is allowed to change its label to α. By these expansion moves, the algorithm finds a local minimum.

The steps of the algorithm are given in the below:

1. Initialize with an arbitrary labeling f.
2. For each label

   - Use single graph-cuts computation to find $f'' = argminE(f')$ where f' is the labeling after one α-expansion of f.

   - If E(f") < E(f) update labeling f as f" and iterate with next label.
3. Return f

The initialization of labels is explained in Section 2.3.2.

By finding a set of superpixel labels $\mathtt{S}^{\mathtt{out}} = \{l_s : \forall s \in \mathcal{S}\}$ that minimize the combined energy terms, we recover a new segmentation mask $\mathtt{S}^{\mathtt{out}}_{i+1}$ for each planar region

and the non-planar areas. For each planar region, a set of updated parameters $\tilde{\pi}_{i+1}^{\texttt{out}}$ are calculated by robust plane fitting to each 3D point projecting onto the corresponding segment of the image. The combined set of estimated segmentation mask and plane parameters $\tilde{\texttt{T}}_{i+1}^{\texttt{out}}$ can now be used in training. In the next section, we show that by repeating this process, we can improve the outdoor plane estimation performance of the network trained on indoor images to a large extent. We also provide a detailed analysis of the contribution from each energy term described in this section to the aforementioned performance increase.

2.4 Experiments

The effectiveness of our approximate training strategy in enhancing the outdoor plane estimation performance of a state-of-the-art network trained on indoor data has been tested through a set of experiments. We have used well-known benchmark datasets for simultaneous localization and mapping (SLAM) and structure from motion (SfM) in our experiments because our method required an outdoor dataset that is adequate for geometric estimation. Both SfM and SLAM benchmarks provide suitable imagery of structured urban scenes that contain the necessary textured surfaces and viewpoint overlap that our approach relies upon. They also contain many planar surfaces and typical scenes for which outdoor plane estimation applications are likely to operate on.

We perform quantitative experiments that demonstrate a single iteration of our approach improves overall plane estimation quality over the baseline. Furthermore, we show that as proposed in Section 2.3, repeated iterations further improves performance. We also present qualitative results that demonstrate the improved segmentation performance as the transfer learning iterations progress. Finally, we present results of an ablation study that measures the contribution from each of the energy terms detailed in Section 2.3.3.

2.4.1 Experiments on a Structure-from-Motion Dataset

We have used the `Dubrovnik` dataset([52]) as a primary set of images to test our approach. It is the collection of 6844 city images taken by different cameras and from varying viewpoints. In order to have different training, validation and test splits, the Dubrovnik dataset is grouped into three parts each consisting of images that depict different city regions. This split ensures that our approach does not overfit to the textures

of particular buildings in the same region. Example images corresponding to each part are shown in Figure 2.5..

We manually annotate the test and validation images for each part of the `Dubrovnik` dataset since it does not contain ground-truth piece-wise plane segmentation data. Both test and validation splits includes 50 images. The training set size is 150 for each part and no ground truth data is required for this set.

Each part is separately processed by COLMAP to obtain the 3D point clouds which we exploit them as a weak-supervision training signal in our proposed framework. We also compute SLIC([4, 5]) superpixels as shown in Figure 2.6.. We extract 1500 superpixels from each training image. We determine the number of extracted superpixels by performing preliminary experiments on `Dubrovnik` dataset. We extract 500 to 2500 superpixels increasing by 500 from the training images of Part I and measure the piece-wise plane segmentation performance on test images of Part II.

2.4.1.1 Image Representation with SLIC Superpixels

Superpixels are a group of pixels that have common individual properties. Images are represented with their superpixels in many computer vision problems such as object segmentation [36, 53] and localization [28] considering efficiency issues. Superpixels are preferable to pixels in such kind of problems since they carry more information than pixels, provide perceptual meaning for the pixels of it and enable compact representation of images which is very useful for computationally demanding problems.

SLIC superpixels is one of the state-of-the-art superpixel algorithms and is faster and more memory efficient than the others [5]. SLIC algorithm is an adaptation of k-means for superpixel generation. It starts sampling a given number of cluster centers. Here, each cluster represents a superpixel. Then, each pixel is assigned to one of the clusters based on a distance metric iteratively until consecutive cluster(superpixel) centers are the same. Finally, to enforce connectivity, disjoint pixels are reassigned to nearby superpixels. In our approach, images are represented with SLIC superpixels. To make it feasible for an optimization algorithm through graph-cuts which is explained in Section 2.3.3.1, a graph structure is constructed where nodes are superpixels and edges are neighbors.

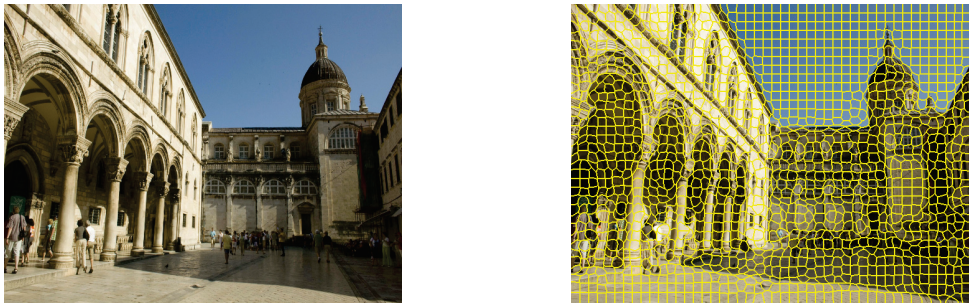
(a) Example images from Part I

(b) Example images from Part II

(c) Example images from Part III

(d) Point cloud computed for Part I

Figure 2.5.: Dubrovnik dataset splits. **(a)-(c)** Three parts of the dataset where each consists of images that belong to a different part of the city. We form the training, validation, and the test sets with different parts to ensure spatial separation of the images in the splits. **(d)** The COLMAP output point cloud extracted from images in Part I.



(a) Input Image (b) SLIC superpixels from (a)

Figure 2.6.: Our training approach estimates a plane segmentation map for each image based on the reconstructed point cloud and the current network output. Since the point cloud is not dense enough to cover each pixel, we compute SLIC superpixels and estimate segmentation labels per superpixel. This ensures that the energy data terms for most of the superpixels depend on several projected 3D points that fall into the corresponding superpixel boundary.

Table 2.1.: Experiment setup for `Dubrovnik` dataset. We perform six different experimental runs for which each split is used for training set, validation set, and test set twice. As iterations progress this improvement slows down, so we stop the iterations at iteration four.

Training	Validation	Test	Number of Iterations
Part I	Part II	Part III	4
Part I	Part III	Part II	4
Part II	Part I	Part III	4
Part II	Part III	Part I	4
Part III	Part I	Part II	4
Part III	Part II	Part I	4

2.4.1.2 Experiment Setup and Results

We conduct six different experiments since we have three splits and each part of the dataset is used for training, validation, and test twice. The overall experiment setup that we performed on `Dubrovnik` dataset is shown in Table 2.1..

For each experimental run, we initialize the PlaneRCNN network with pretrained weights obtained by using the indoor dataset. We evaluate its initial piece-wise plane segmentation quality on the test set. We estimate a refined set of segmentation masks and plane parameters as described in Section 2.3 to act as the training targets on the outdoor dataset. We then retrain the PlaneRCNN layers that belong to the mask head, box head, classifier head and the depthmap decoder of the plane detection network for 60 epochs using the estimated training targets. The cross-entropy loss is computed between the segmentation output of the network and estimated approximate ground-truth targets. An example loss behavior under the training of Part II as the number of iterations increases is given in Figure 2.7..

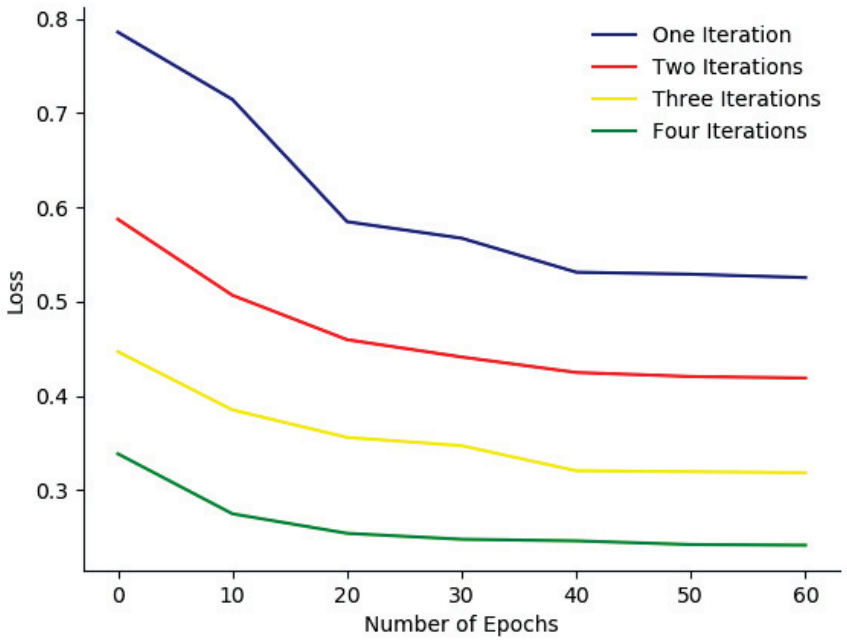


Figure 2.7.: Loss behavior under the training of Part II as the number of iterations increases.

The training takes approximately 3.5 days for each part of the `Dubrovnik` dataset. We determine the training result by taking the set of weights gives the best best piece-wise

segmentation performance for the validation set of images. Similarly, the scalar weights in the energy function, $\lambda_s, \alpha_1 - \alpha_3$ are set by a grid search that maximize the validation set performance.

In order to evaluate piece-wise plane segmentation accuracy, we measure the plane detection metric, *Plane Recall*, which is defined as the ratio of the number of estimated planes that have at least 0.5 Intersection over Union (IOU) score with one of the ground-truth planes to the number of ground-truth planes. The IOU score is measured with a varying depth error threshold from 0 to 1 meters with an increment of 0.05m for indoor images of PlaneRCNN. We set the depth error threshold to vary from 0 to 10 meters with an increment of 0.5m for outdoor images.

We perform four iterations in each experiment since improvement slows down after iteration four. To determine this, we perform a preliminary experiment in which we measure the plane recall for Part I under the training of part II for seven iterations. As the Figure 2.8. shows, the improvement of plane recall stops after the fourth iteration.

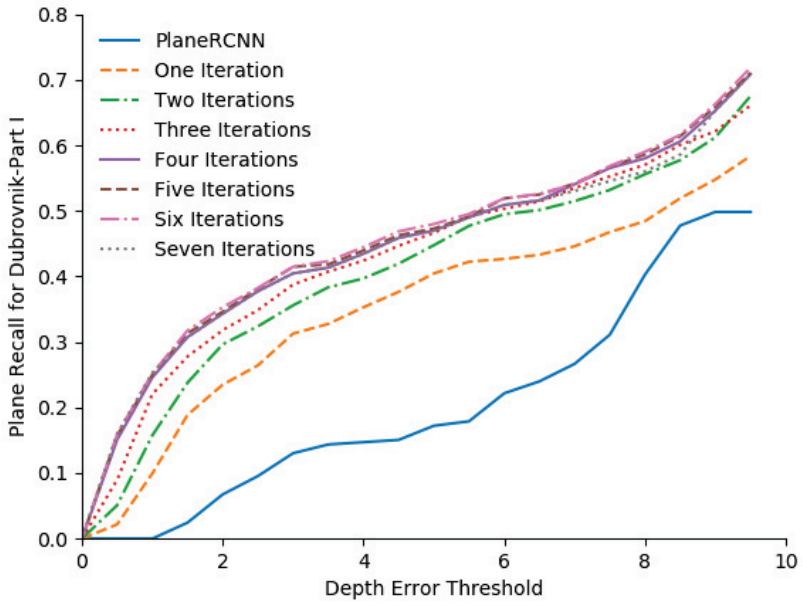


Figure 2.8.: Preliminary experiments on Part I of `Dubrovnik` dataset under the training of Part II to determine the maximum number of iterations. After the four iterations, improvement on plane recall slows down.

Figure 2.9. shows plane recall results of our approach for Part I under the training of part II and part III. Plane recall performance under Part II training is better than PlaneRCNN, even with a single iteration. Since the texture of the scene images belonging to part III is considerably different from part I, plane recall of PlaneRCNN is slightly better

than our approach with a single iteration for small depth error thresholds. The performance of our approach is better than PlaneRCNN for each of the depth error thresholds from the second iteration. For average plane recall for Part I, our approach performs better than the PlaneRCNN for each depth error threshold, even with a single iteration. Plane detection and segmentation performance improved for each depth error interval as the number of iterations increased.

Figure 2.10. illustrates the performance of our approach for Part II under the training of part I and part III. Plane recall performance for Part II under Part I or Part III training is better than PlaneRCNN even with a single iteration. For average plane recall for Part II, our performance is better than the PlaneRCNN for each depth error threshold, even with a single iteration. Plane detection and segmentation performance improved for each depth error interval as the number of iterations increased.

As Figure 2.11. shows, our performance for Part III under the training of part I and part II is not better than PlaneRCNN for all depth error thresholds and iterations for Part I and small depth error thresholds for part II. Since the texture of the images of the scene belonging to part III is considerably different from part I, network weights computed on training with Part I can not be informative for part III with fewer iterations. For average plane recall for Part III, our approach performs better than the PlaneRCNN for significant depth error thresholds after four iterations. Our plane recall performance shows that our approach becomes better than PlaneRCNN as the number of iterations increases.

Figure 2.12. shows the performance of our approach for the `Dubrovnik` dataset experiments averaged over the six experimental runs described above. As the figure shows, our approach performs better than PlaneRCNN even with a single training iteration especially at larger depth error thresholds. As number of iterations increases, its performance becomes significantly better than the PlaneRCNN trained on indoor images for all depth thresholds. Moreover, multiple iterations of our approach is able to improve the network performance on outdoor images.

We also evaluate our piece-wise plane detection and segmentation performance with segmentation quality metrics for multiple planes. We measure three different plane segmentation metrics:

- **Segmentation Covering(SC) [7]**:Segmentation covering is the metric that measures the overlap between overall estimated segmentation and the overall ground-truth segmentation by covering estimated segmentation masks with each of the ground-truth individual segmentation masks and averaging over the ground-truth segmentation masks. It is computed as follows:

$$SC(S_e, S_{gt}) = \frac{1}{N} \sum_{R_{gt} \epsilon S_{gt}} |R_{gt}| \cdot \max_{R_e \epsilon S_e} O(R_{gt}, R_e) \tag{2.1}$$

  where *S_e* and *S_gt* are estimated and ground-truth segmentations correspondingly.

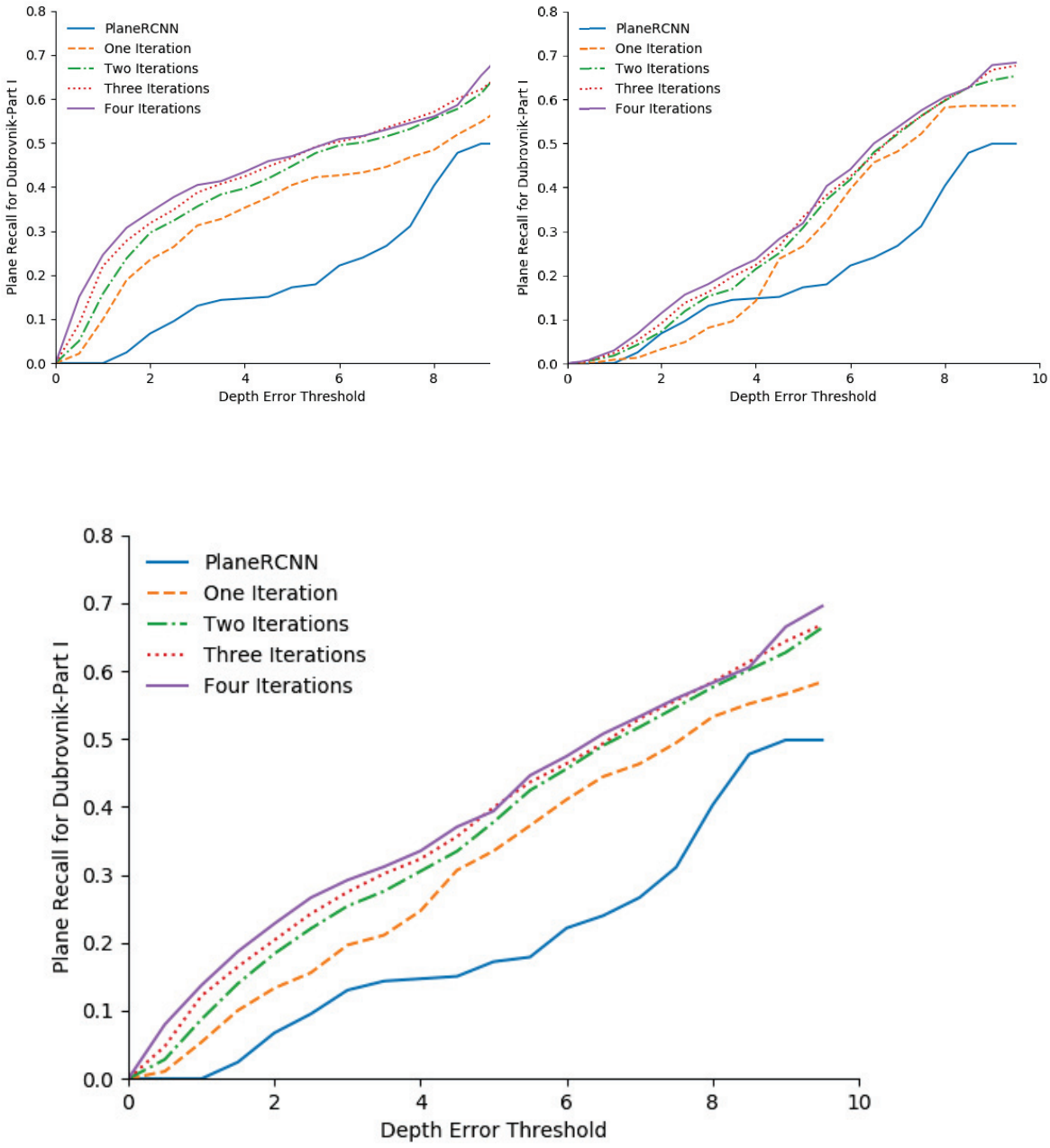


Figure 2.9.: Plane recall for the Part I of `Dubrovnik` dataset. In the upper row, plane recall for the Part I is illustrated under training of Part II and part III respectively from left to right. Average plane recall for the Part I is shown in the lower row as the number of training iterations is increased.

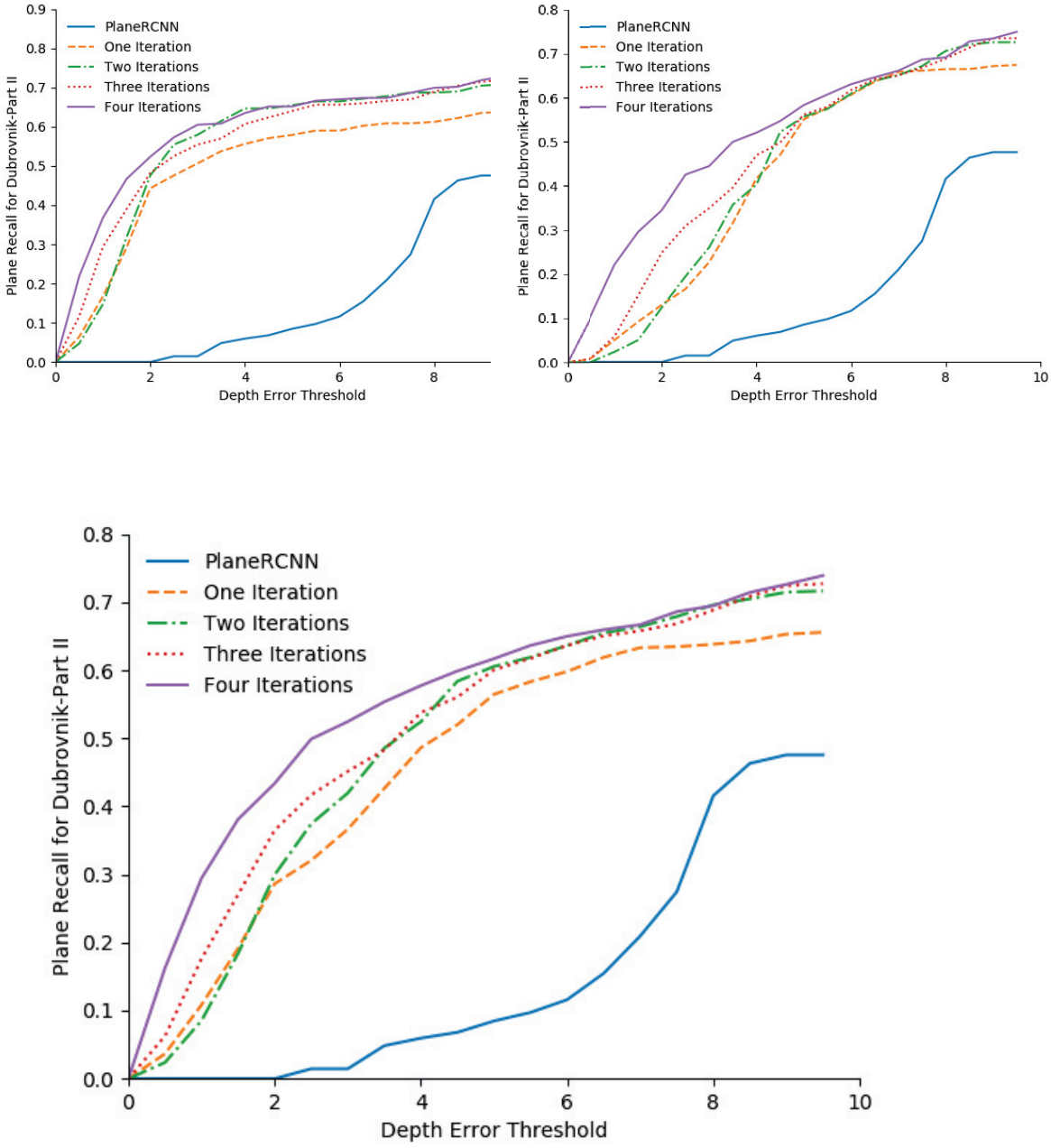


Figure 2.10.: Plane recall for the Part II of `Dubrovnik` dataset. In the upper row, plane recall for the Part II is illustrated under training of Part I and part III respectively from left to right. Average plane recall for the Part II is shown in the lower row as the number of training iterations is increased.

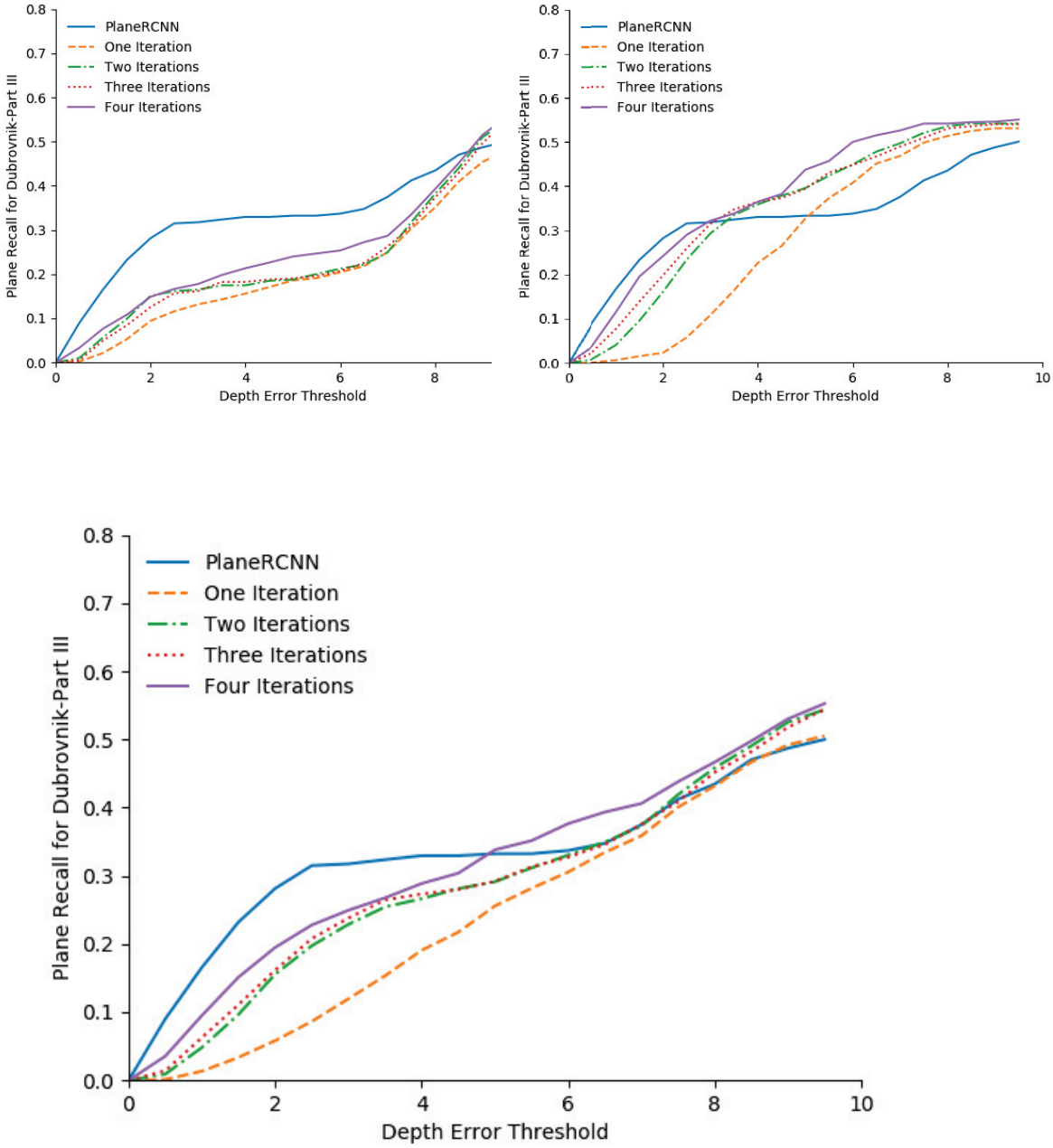


Figure 2.11.: Plane recall for the Part III of `Dubrovnik` dataset. In the upper row, plane recall for the Part III is illustrated under training of Part II and part I respectively from left to right. Average plane recall for the Part III is shown in the lower row as the number of training iterations is increased.

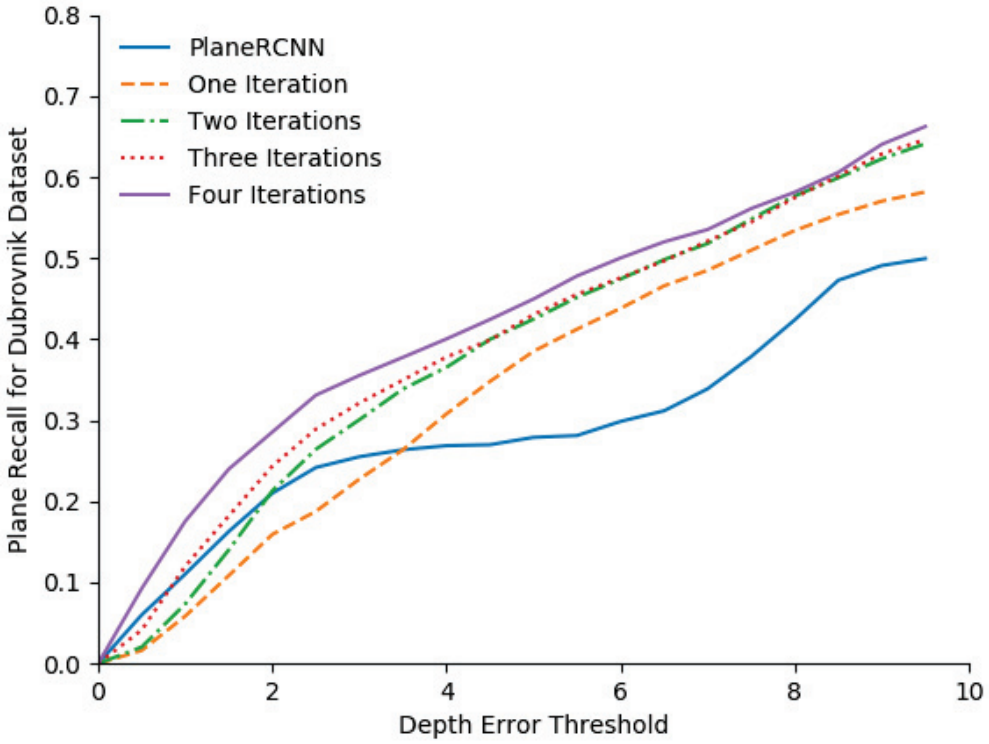


Figure 2.12.: Average plane recall for the `Dubrovnik` dataset as the number of training iterations is increased.

N is the total number of pixels in the image. $O(R_{gt}, R_e)$ is the overlap between estimated and ground-truth segmentation mask and computed as follows:

$$O(R_{gt}, R_e) = \frac{|R_{gt} \cap R_e|}{|R_{gt} \cup R_e|} \tag{2.2}$$

- **Variation of Information(VOI) [61]**:The Variation of Information metric was originally introduced for comparing two clusterings. Since image segmentation for any purpose is a kind of clustering, this metric can be considered as plane segmentation evaluation criteria.

  Variation of Information(VOI) measures the distance between estimated and ground-truth segmentations in terms of their average conditional entropy and can be formulated as following:

$$VOI(S_e, S_{gt}) = E(S_e) + E(S_{gt}) - 2M(S_e, S_{gt}) \tag{2.3}$$

  where E represents entropy and M represents mutual information between estimated and ground-truth segmentations.

- **Rand Index(RI) [70]**: Rand Index was originally introduced for clustering assessment. In the case of plane segmentation, Rand Index(RI) is computed by the sum of the number of pairs of pixels that have the same segmentation label in both estimated and ground-truth segmentations and those that have different segmentation labels in both segmentations divided by the total number of pairs of pixels.

Table 2.4.1.2 shows the results for `Dubrovnik` dataset belong to multiple plane segmentation quality metrics as the number of iterations increases. Even with a single iteration the performance of our approach is better than PlaneRCNN. Applying our approach iteratively improves the segmentation quality based on all measured metrics.

2.4.2 Experiments on a SLAM Dataset

We conduct another set of tests on a SLAM dataset to further evaluate our method and demonstrate its capacity to increase outdoor plane segmentation accuracy. For this purpose, we use the images from the `KITTI` dataset ([32]) that depict urban scenes captured from a car travelling around city blocks for test purposes. We use the images in the `Dubrovnik` dataset to form the training and validation sets. Our experimental

Table 2.2.: Dubrovnik dataset performance of our approach for multiple planes segmentation quality metrics.

	Plane Segmentation Metric		
	SC	VOI ↓	RI
Dubrovnik-PlaneRCNN	0.478	2.236	0.533
Dubrovnik-One Iteration	0.498	1.854	0.541
Dubrovnik-Two Iterations	0.509	1.741	0.546
Dubrovnik-Three Iterations	0.519	1.644	0.549
Dubrovnik-Four Iterations	**0.525**	**1.605**	**0.551**

Table 2.3.: Experiment setup for KITTI dataset. For these experimental runs, we construct training and validation splits from Dubrovnik dataset by using all determined parts. As iterations progress this improvement slows down, so we stop the iterations at iteration four.

Training	Validation	Test	Number of Iterations
Dubrovnik Part I-II-III	Dubrovnik Part I-II-III	KITTI Sequence 11	4
Dubrovnik Part I-II-III	Dubrovnik Part I-II-III	KITTI Sequence 13	4
Dubrovnik Part I-II-III	Dubrovnik Part I-II-III	KITTI Sequence 15	4
Dubrovnik Part I-II-III	Dubrovnik Part I-II-III	KITTI Sequence 16	4
Dubrovnik Part I-II-III	Dubrovnik Part I-II-III	KITTI Sequence 18	4
Dubrovnik Part I-II-III	Dubrovnik Part I-II-III	KITTI Sequence 20	4

setup is shown in Table 2.3.. This is a more challenging test design than the prior set of experiments on the `Dubrovnik` dataset since the building styles and viewpoint distribution are considerably different between the datasets.

(a) Example Image from Sequence 11

(b) Example Image from Sequence 13

(c) Example Image from Sequence 15

(d) Example Image from Sequence 16

(e) Example Image from Sequence 18

(f) Example Image from Sequence 20

Figure 2.13.: Example images from each sequence of `KITTI` dataset that we used in our experiments.

Table 2.4.: KITTI dataset performance of our approach for multiple planes segmentation quality metrics.

	Plane Segmentation Metric		
	SC	VOI ↓	RI
KITTI-PlaneRCNN	0.463	2.113	0.519
KITTI-One Iteration	0.482	1.742	0.53
KITTI-Two Iterations	0.495	1.705	0.535
KITTI-Three Iterations	0.514	1.662	0.539
KITTI-Four Iterations	**0.518**	**1.648**	**0.542**

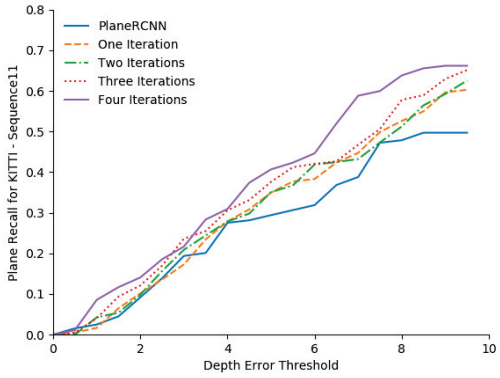
KITTI dataset contains eleven test sequences numbered from 11 to 21. We randomly select 50 test images from test sequences numbered 11, 13, 15, 16, 18, 19, and 20 and exclude others that do not have at least two dominant planes. An example image from each selected sequence from the KITTI dataset is shown in Figure 2.13.. These test images are manually annotated, the same experimental process is used, and the plane recall is evaluated.

In Figure 2.14., plane recall of our approach and PlaneRCNN is shown for each of the selected sequences from KITTI dataset. Our approach performs better than PlaneRCNN for each of the sequences, even with a single iteration for most of them. As the number of iterations increases, each sequence's plane detection and segmentation performance improved for each of the depth error intervals.

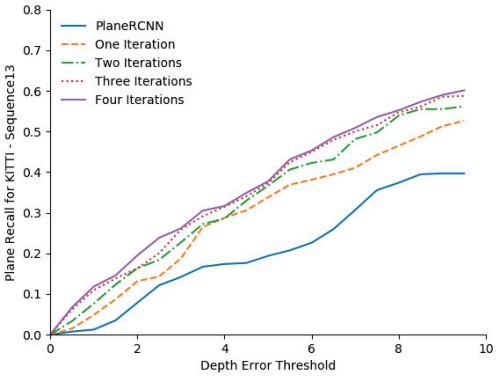
Figure 2.15. shows the overall results averaged over all test sequences of the KITTI dataset. Despite the large visual differences between the images from two datasets, our approach trained on the Dubrovnik dataset significantly improves the outdoor plane estimation performance on the images of the KITTI dataset. As iterations progress this improvement slows down, so we stop the iterations at iteration four.

We measure the performance of our approach for KITTI dataset based on multiple planes segmentation quality metrics. Table 2.4 shows the results for multiple plane segmentation quality metrics as the number of iterations increases. Even with a single iteration the performance of our approach is better than PlaneRCNN. Applying our approach iteratively improves the segmentation quality based on all measured metrics.

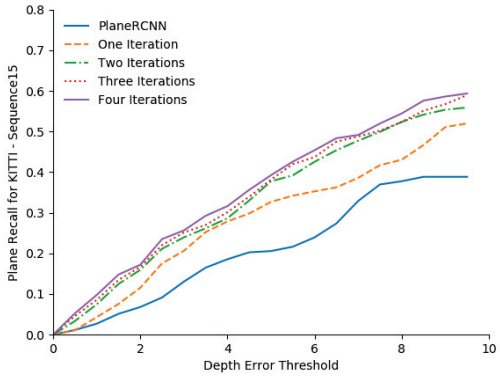
Figure 2.16. shows piece-wise plane segmentation estimations for different test images from both the Dubrovnik and KITTI datasets for a qualitative comparison. PlaneRCNN trained for indoor images misses most of the planar regions and undersegments the detected ones. The same architecture retrained on outdoor images by the proposed approach is able to detect most of the planes in the scene with more accurate boundaries. This improvement is achieved without providing detailed segmentation maps on the outdoor images of the training set.



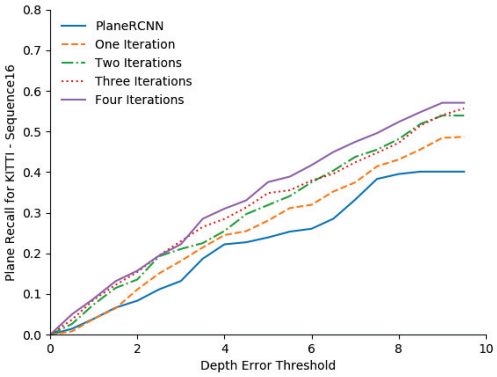
(a) Plane Recall for Sequence 11



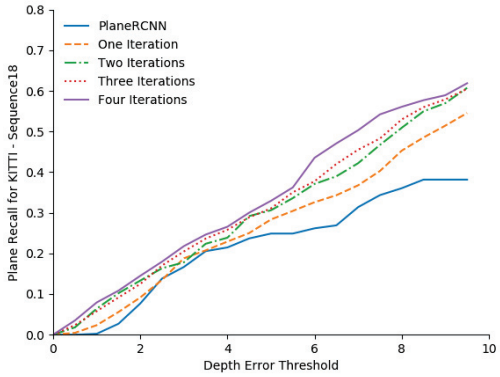
(b) Plane Recall for Sequence 13



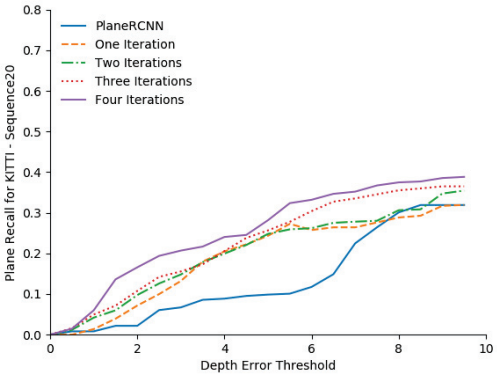
(c) Plane Recall for Sequence 15



(d) Plane Recall for Sequence 16



(e) Plane Recall for Sequence 18



(f) Plane Recall for Sequence 20

Figure 2.14.: Plane recall for each test sequence used from KITTI dataset. Although our approach is trained on KITTIdataset, it performs better than PlaneRCNN for all of the sequences. For each sequence, as the number of iterations increases, plane detection and segmentation performance improved for each of the depth error intervals.

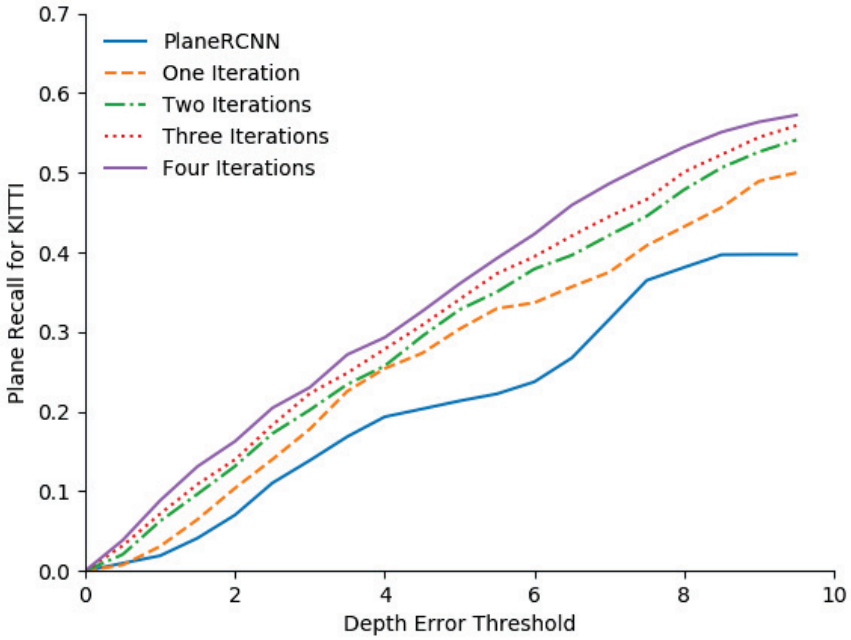


Figure 2.15.: Average plane recall for the `KITTI` dataset test sequences. Even with a single iteration, our approach performs better than the PlaneRCNN trained indoors despite being trained on images from the `Dubrovnik` dataset. Plane detection and segmentation performance improved for each of the depth error intervals as the number of iterations increase.

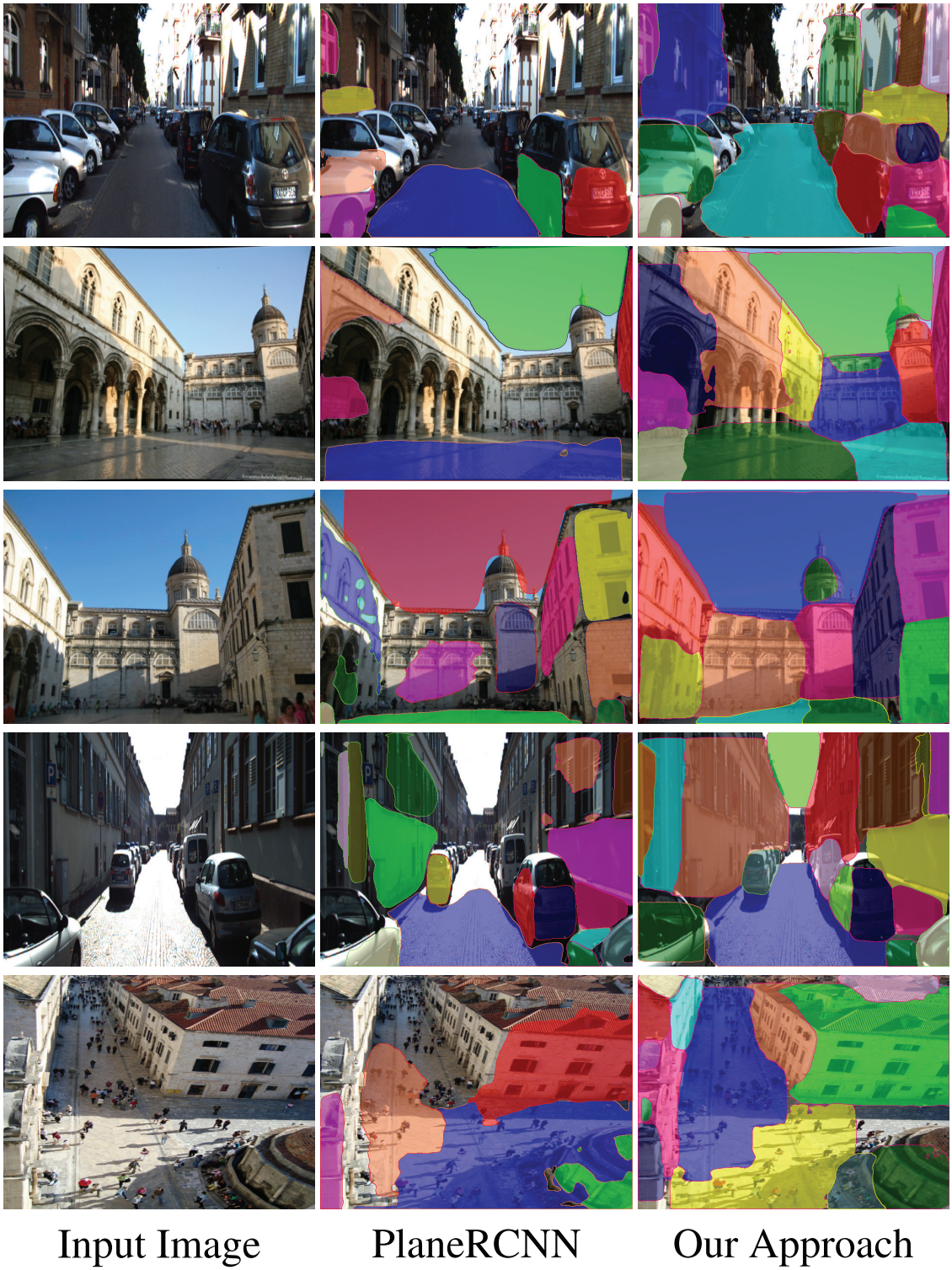


Figure 2.16.: Comparison of piece-wise planar segmentation maps for different test images from both `Dubrovnik` and `KITTI` datasets. Results of our approach belong to estimations obtained after fourth training iteration. PlaneRCNN trained indoors misses most of the planar regions and undersegments the detected ones. Despite this, the ground truth estimates automatically obtained with our energy minimization formulation are accurate enough to improve both plane detection rates and segmentation accuracy.

2.4.3 Ablation Study

We perform two sets of ablation studies. As a first, we compare the plane recall performance when we retrain the PlaneRCNN with the ground-truth data and our estimated approximate training targets. Since there is no ground-truth annotated data for training images of `Dubrovnik` dataset, we use test images of Part II for training which are manually labeled for evaluation and measure plane recall of test images belong to Part I. As it is shown in Figure 2.17., our plane recall performance is not too much behind from the retraining with ground-truth targets even with a single iteration. After the four iterations, retraining with our approximate training targets gives close piece-wise plane segmentation estimation performance with the network retrained with ground-truth targets.

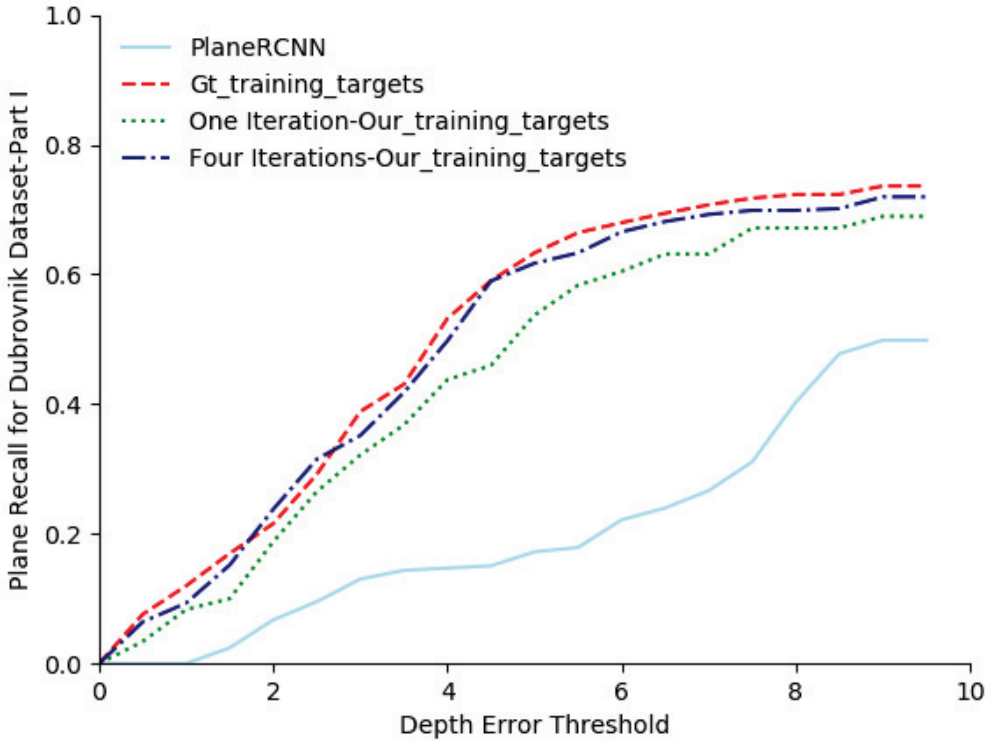


Figure 2.17.: Plane recall comparison between retraining the PlaneRCNN under ground-truth and our estimated targets.

Furthermore, we perform an ablation study to show the contribution of each individual data term of the energy function described in Section 2.3.3. From `Dubrovnik` dataset, we use Part I as training and measure the plane recall performance on Part II.

Table 2.5.: Plane recall values as the data term $E_d(l_s)$ varies. To better understand the effect of different parts of the segmentation energy data cost, we gradually add more complex terms and measure the plane recall for each variation. Adding terms for both E_{support} and E_{distance} improves results over using either term. Using the additive $\delta(l_s - \hat{l}_s)$ factor also boost results by increasing the label cost changes when there is less evidence from the point cloud.

	Depth Error Threshold			
$E_d(l_s)$	0.0-2.5	2.5-5.0	5.0-7.5	7.5-10
$\alpha_1 E_{\text{support}}$	0.101	0.226	0.261	0.282
$\alpha_2 E_{\text{distance}}$	0.063	0.141	0.172	0.207
$\alpha_1 E_{\text{support}} + \alpha_2 E_{\text{distance}}$	0.124	0.276	0.313	0.357
$\left(\alpha_1 + \delta(l_s - \hat{l}_s)\right) E_{\text{support}}$	0.171	0.495	0.578	0.621
$\left(\alpha_2 + \delta(l_s - \hat{l}_s)\right) E_{\text{distance}}$	0.227	0.499	0.551	0.593
$\left(\alpha_1 + \delta(l_s - \hat{l}_s)\right) E_{\text{support}}$ $+ \left(\alpha_2 + \delta(l_s - \hat{l}_s)\right) E_{\text{distance}}$	**0.316**	**0.614**	**0.667**	**0.701**

As before, PlaneRCNN is retrained in four iterations for each term of data cost function $E_d(l_s)$. The results of Table 2.5. show that the term measuring support from projected 3D points and the one measuring 3D plane distance both contribute to plane segmentation performance. Moreover, the $\delta(l_s - \hat{l}_s)$ terms that increase the cost of label changes positively affect the network training. Best results are obtained when the data term matches the final form described in Section 2.3.3.

2.5 Conclusion

We have developed a method for improving piece-wise plane detection and segmentation accuracy of outdoor scenes without requiring manual annotation. The only prerequisite for our method's success is that the training dataset can be handled by a SfM-MVS pipeline and the network trained on indoors. We have demonstrated that the initial network's and the point cloud's weak supervision is sufficient to accurately estimate the ground truth labels on the outdoor imagery. This allows for the weights to be improved, which in turn enhances the estimation of the outdoor ground truth. Applying this idea iteratively increases the detection and segmentation accuracy on several images of outdoor scenes. As a result, we have demonstrated that a network that was trained on indoor images

can be adapted to deal with outdoor imagery without the need for a manually annotated training set. This proposed and developed approach might be applicable in construction environment to monitor the overall process.

CHAPTER 3

USING DEPTH CNN FOR 3D POINT CLOUD ACQUISITION

3.1 Introduction

Depth estimation from a single image is a challenging computer vision task that gives the scene structure as an output. Monocular depth estimation is commonly used in object detection [87, 6], Simultaneous Localization and Mapping(SLAM) [53] and semantic segmentation [22].

Leveraging geometric constraints and a set of overlapping monocular images to obtain the 3D structure is the traditional approach for monocular depth estimation. Structure from Motion (SfM) [46] enables extraction of 3D structure so that a sparse depth map for each image in a given input sequence can be obtained. As it is stated, an overlapping input image sequence is required for SfM. The quality of the final scene structure highly depends on feature extraction and matching. Also, the scale uncertainty exists unless the absolute distance between cameras is known. Monocular depth information can also be obtained from tools embedded with depth sensors, RGB-D cameras or LIDAR. However, depth sensors have a limited operating range making it significantly harder to be practical for the outdoor environment.

With the advances in deep learning approaches, several methods have been introduced for monocular depth estimation. Some of them [19, 24, 34] are trained to learn geometric priors related to depth. Besides the geometric priors, semantic constraints are also taken into account in some deep-learning based methods [104, 92, 18]. Newly introduced loss functions considering depth-related constraints are proposed in some approaches [54, 71, 48].

In this part of the thesis work, we exploit a deep neural network designed for monocular depth estimation [62] in order to obtain 3D dense point cloud that we used as a weak supervision signal in our iterative transfer learning approach which is explained in Section 2.3(See Figure 2.1.). As Figure 3.1. shows, we modified our primary framework so that instead of obtaining dense point cloud from the SfM-MVS pipeline which requires overlapping set of images, we obtain it for each input training image from the estimated

depthmap given by the monocular depth estimation network. We observe that exploiting 3D dense point cloud from deep neural network designed for monocular depth estimation gives better segmentation performance than the one where the dense point cloud is obtained from SfM-MVS pipeline.

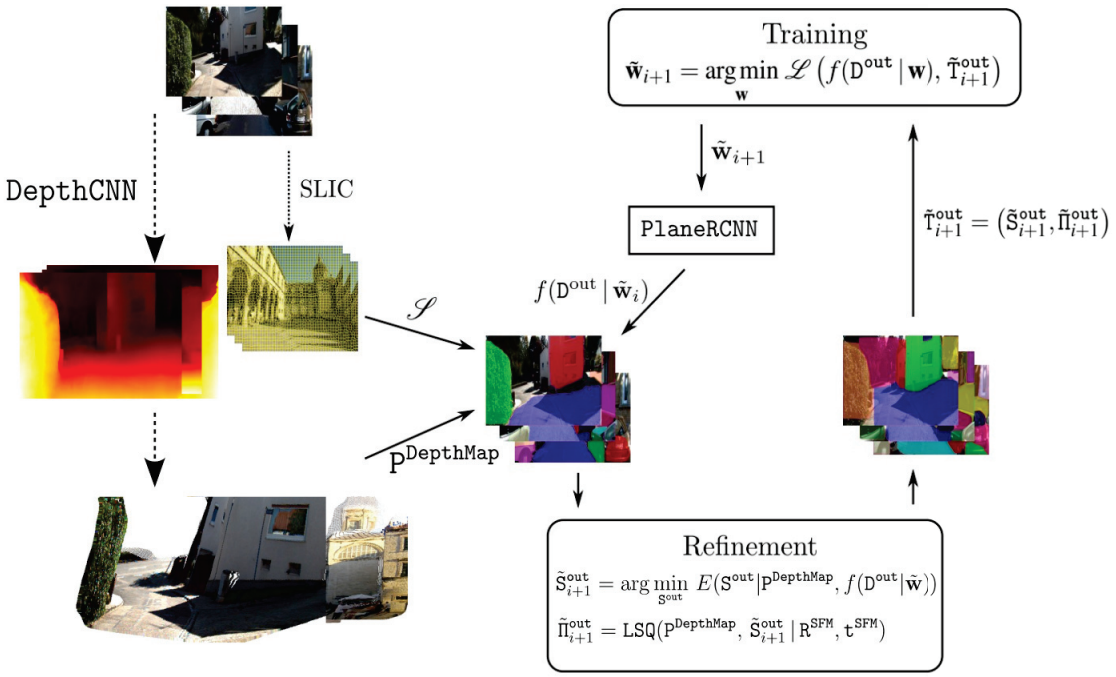


Figure 3.1.: Modified framework in terms of point cloud acquisition. In our primary approach, we obtain dense point cloud from SfM-MVS pipeline. This requires a set of overlapping images which prevents to apply the developed idea to custom outdoor imagery. To remove this limitation, we obtain dense point cloud from the estimated depthmap given by deep monocular depth estimation network for each training image.The other components of the framework remains the same.

In this chapter, our studies can be summarized as follows:

- We exploit a deep neural network designed for monocular depth estimation in order to obtain 3D dense point cloud.
- We demonstrate that obtaining 3D dense point cloud from the deep neural network gives better plane detection performance than the one where 3D dense point cloud is obtained from SfM-MVS pipeline.
- We perform experiments on plane segmentation metrics to compare our main proposed approach with PlaneRCNN based on different plane-related issues. We also show that obtaining 3D dense point cloud from the deep neural network gives bet-

ter plane segmentation performance than the one where 3D dense point cloud is obtained from SfM-MVS pipeline.

3.2 Studies in Deep Monocular Depth Estimation

Eigen et al. [24] proposed a deep neural architecture that is composed of two components. The coarse-scale network estimates the entire depth map structure by collecting global view features through the layers. The fine-scale network refines the coarse depth map by considering the local details of the given scene.

Chen et al. [19] introduced an approach that estimates pixel-wise depth from a single image from an unconstrained environment by taking annotations of relative depth as training input. They use a ranking-based loss function related to input relative depth annotations as an input for the pixel-wise prediction for depth.

Godard et.al. [34] proposed an unsupervised monocular depth estimation exploiting epipolar geometry. Instead of using ground-truth depth data, they extract epipolar constraints and obtain disparity images. They train the network with a loss function to deal with the disparity consistency in both left and right epipolar images.

Lee et al. [48] presented an approach that combines multiple loss functions for monocular depth estimation. They used a standard encoder-decoder network for depth estimation. As a novelty, they developed a loss re-balancing algorithm so that the weight for each loss function is rebalanced dynamically during training.

Ranftl et al. [71] developed an approach that mixes different datasets during training for the task of monocular depth estimation. They proposed a loss function that is unvarying to changes in depth extent based on any individual dataset. They do training with the fully convolutional encoder-decoder depth estimation networks. For training, they resize the biggest dimension of the input image to the training resolution while keeping the aspect ratio. They claimed that their performance is better than the previous monocular depth estimation approaches in most cases.

Miangoleh et al. [62] proposed a method for monocular depth estimation. They used existing deep monocular depth estimation frameworks ([71], [93],[100]) to generate highly detailed estimations without any retraining. They get several estimations at different resolutions and then merge those into a structurally consistent high-resolution depth map. Different resolutions are taken into account since different depth qualities arise to various resolutions. Most of the details in the scene are missing at lower resolutions. The problem of the inconsistent overall structure occurs at high resolutions even the network is able to generate high-frequency details. They set the highest resolution that will

provide a consistent structure by ensuring that every pixel is contextually informative for any given image. For this purpose, they obtain the distribution of contextual cues by approximating them with an edge map. They train the image-to-image translation network to merge the low-resolution depth with the high-resolution details. This removes structural inconsistencies of the high-resolution input. With this proposed CNN, they get state-of-the-art monocular depth estimation results for several datasets.

In this part of the study, we exploit deep monocular depth estimation network [62] to obtain 3D dense point cloud instead of acquisition with SfM-MVS pipeline in our main approach.

3.3 Refined Energy Formulation

In this proposed method, we obtain dense point cloud $\mathtt{P}^{\mathtt{DepthMap}}$ from the depth estimated by the monocular depth estimation network instead of exploiting $\mathtt{P}^{\mathtt{MVS}}$ from SfM-MVS pipeline as it is explained in Section 2.3. The superpixel based energy minimization formulation for estimating the training targets given in Section 2.3.3 is updated as follows:

$$E\left(\mathtt{S}^{\mathtt{out}}\right) = \sum_{s\in\mathcal{S}} E_d\left(l_s \mid \mathtt{P}^{\mathtt{DepthMap}},\ f(\mathtt{D}^{\mathtt{out}} \mid \tilde{\mathbf{w}}_i)\right) + \lambda_s \sum_{(s,t)\in\mathcal{N}_S} E_s(l_s, l_t \mid \mathtt{P}^{\mathtt{DepthMap}},\ f(\mathtt{D}^{\mathtt{out}} \mid \tilde{\mathbf{w}}_i)),$$

where the both data and smoothness terms and their corresponding components with scalar weights are the same with the first proposed method stated in Section 2.3.3 in which 3D dense point cloud is obtained from SfM-MVS pipeline.

3.4 Experiments

In this section, we perform two sets of experiments. First, we compare plane recall performance on Dubrovnik dataset when the 3D dense point cloud is obtained from the depth estimated by deep monocular depth estimation network $\mathtt{P}^{\mathtt{DepthMap}}$ and SfM-MVS pipeline $\mathtt{P}^{\mathtt{MVS}}$. Since plane recall is a plane detection metric, we also measure the performance of PlaneRCNN and our approach with both 3D point cloud obtained from the estimated depthmap and SfM-MVS pipeline based on plane segmentation metrics in the second part of the experiments.

3.4.1 Evaluation with Plane Recall

Figure 3.2. shows the comparative plane recall results at the end of the fourth iteration for the `Dubrovnik` dataset for which 3D dense point cloud obtained from SfM-MVS pipeline and depthmap estimated by the deep monocular depth estimation network. As a ground-truth depth data, the depth obtained after SfM-MVS pipeline is usedfor each of the experiments. We observe that the 3D dense point cloud from the depth estimated from deep monocular depth estimation network $\mathrm{P}^{\mathtt{DepthMap}}$ gives better plane recall results than the point cloud obtained with SfM-MVS pipeline $\mathrm{P}^{\mathtt{MVS}}$ in our approach. When 3D dense point cloud is obtained from deep monocular depth estimation network, plane recall is higher than the SfM-MVS pipeline especially at larger depth error thresholds.

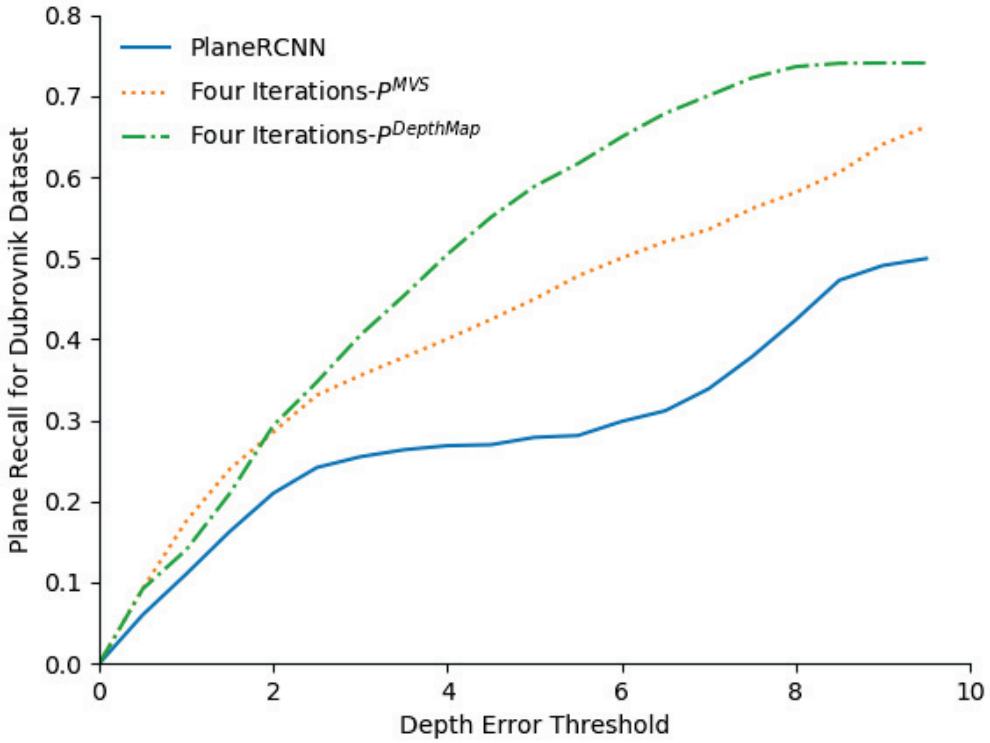


Figure 3.2.: Comparative plane recall results at the end of the fourth iteration for the Dubrovnik dataset for which 3D point cloud obtained from SfM-MVS pipeline $\mathrm{P}^{\mathtt{MVS}}$ and depthmap estimated by deep monocular depth estimation network $\mathrm{P}^{\mathtt{DepthMap}}$.

Figure 3.3. shows the comparative plane recall comparison between exploiting $\mathrm{P}^{\mathtt{MVS}}$ and $\mathrm{P}^{\mathtt{DepthMap}}$ for `KITTI` dataset. Point cloud from estimated depth map by the deep neural

network gives better plane recall results than the exploiting point cloud from SfM-MVS pipeline for KITTI dataset at each depth error threshold interval.

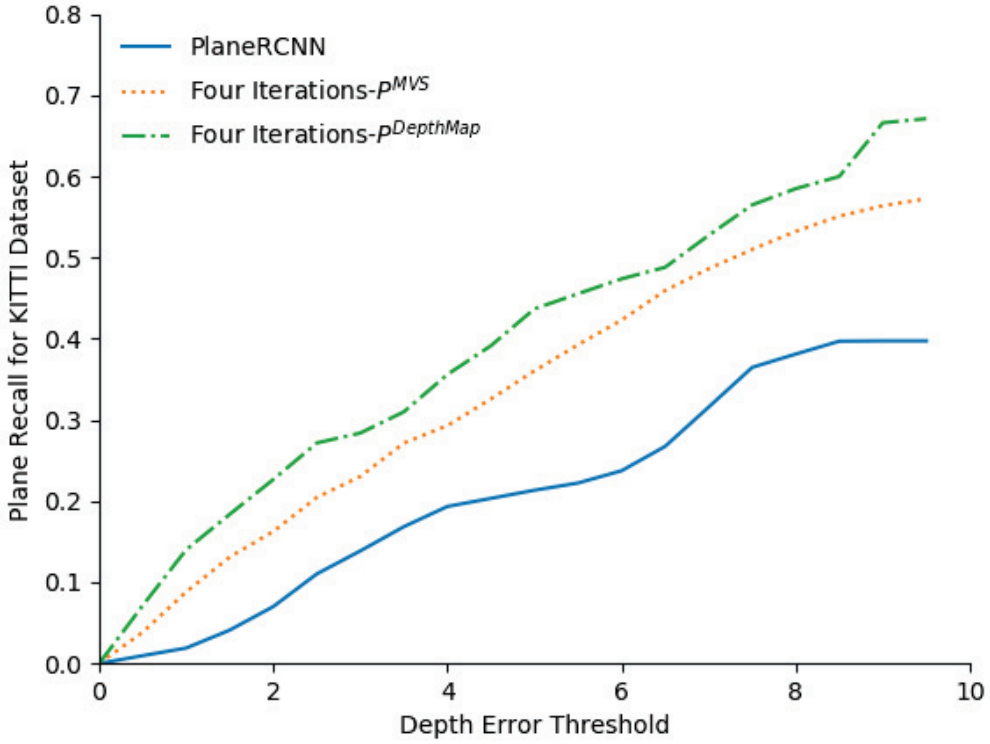


Figure 3.3.: Comparative plane recall results at the end of the fourth iteration for the KITTI dataset for which 3D point cloud obtained from SfM-MVS pipeline $\mathrm{P}^{\mathtt{MVS}}$ and depthmap estimated by deep monocular depth estimation network $\mathrm{P}^{\mathtt{DepthMap}}$.

3.4.2 Evaluation with Plane Segmentation Metrics

As it is explained in Section 2.4.1.2, we also measure the performance our approach with multiple planes segmentation quality metrics which are Segmentation Covering(SC) [7], Variation of Information(VOI) [61], and Rand Index(RI) [70]. So, we compare the plane segmentation accuracy where 3D dense point cloud is obtained from both estimated depthmap $\mathrm{P}^{\mathtt{DepthMap}}$ and SfM-MVS pipeline $\mathrm{P}^{\mathtt{MVS}}$. Table 3.1 shows the results for the plane segmentation metrics for both our approach and PlaneRCNN for the `Dubrovnik` dataset. For each part of the Dubrovnik dataset, we have better piece-wise

Table 3.1.: Comparative evaluation of plane segmentation metrics between our approach and PlaneRCNN. For each part of the Dubrovnik dataset, our piece-wise plane segmentation is more accurate than PlaneRCNN. Exploiting 3D dense point cloud from deep monocular depth estimation(Ours-$\mathtt{P}^{\mathtt{DepthMap}}$) gives better piece-wise plane segmentation than the one when we obtain point cloud from SfM-MVS pipeline(Ours-$\mathtt{P}^{\mathtt{MVS}}$).

	Plane Segmentation Metric		
	SC	VOI ↓	RI
Part1-PlaneRCNN	0.501	2.273	0.534
Part1-Ours-$\mathtt{P}^{\mathtt{MVS}}$	0.542	1.642	0.546
Part1-Ours- $\mathtt{P}^{\mathtt{DepthMap}}$	0.565	1.583	0.553
Part2-PlaneRCNN	0.397	2.174	0.533
Part2-Ours-$\mathtt{P}^{\mathtt{MVS}}$	0.463	1.605	0.534
Part2-Ours-$\mathtt{P}^{\mathtt{DepthMap}}$	0.511	1.567	0.538
Part3-PlaneRCNN	0.536	2.262	0.532
Part3-Ours-$\mathtt{P}^{\mathtt{MVS}}$	0.571	1.568	0.574
Part3-Ours-$\mathtt{P}^{\mathtt{DepthMap}}$	0.576	1.517	0.593
Avg.-PlaneRCNN	0.478	2.236	0.533
Avg.-Ours-$\mathtt{P}^{\mathtt{MVS}}$	0.525	1.605	0.551
Avg.-Ours-$\mathtt{P}^{\mathtt{DepthMap}}$	**0.551**	**1.558**	**0.561**

plane segmentation performance than PlaneRCNN for each of the metrics. Exploiting 3D dense point cloud from the estimated depth $\mathtt{P}^{\mathtt{DepthMap}}$ gives more accurate piece-wise plane segmentation than the one when we obtain point cloud from SfM-MVS pipeline $\mathtt{P}^{\mathtt{MVS}}$.

We also compare the piece-wise plane detection and segmentation performance between using 3D dense point cloud obtained from both estimated depthmap $\mathtt{P}^{\mathtt{DepthMap}}$ and SfM-MVS pipeline $\mathtt{P}^{\mathtt{MVS}}$ for `KITTI` dataset. As it is shown in Table 3.2, obtaining point cloud from estimated depthmap $\mathtt{P}^{\mathtt{DepthMap}}$ provides better piece-wise plane segmentation quality than the one we obtain dense point cloud from $\mathtt{P}^{\mathtt{MVS}}$.

3.5 Conclusion

Obtaining 3D dense point cloud from SfM-MVS pipeline requires overlapping set of images. This restricts our approach which is explained in Chapter 2 to generalize

Table 3.2.: Comparative evaluation of plane segmentation metrics between our approach and PlaneRCNN for KITTI dataset. For both experiment setups, our piece-wise plane segmentation is more accurate than PlaneRCNN. Exploiting 3D dense point cloud from deep monocular depth estimation(Ours-$P^{DepthMap}$) gives better piece-wise plane segmentation than the one when we obtain point cloud from SfM-MVS pipeline(Ours-P^{MVS}).

	Plane Segmentation Metric		
	SC	VOI ↓	RI
KITTI-PlaneRCNN	0.463	2.113	0.519
KITTI-Ours-P^{MVS}	0.518	1.642	0.542
KITTI-Ours-$P^{DepthMap}$	**0.537**	**1.598**	**0.569**

it any dataset composed of images belong to any outdoor scene. Furthermore, image regions with textureless regions might be missing in 3D reconstruction. In this chapter, we exploit from state-of-the-art deep monocular depth estimation network for 3D dense point cloud acquisition. We perform set of experiments to measure plane detection and segmentation accuracy of our approach when point cloud is obtained from deep monocular depth estimation network. We observe that our approach with the point cloud from deep neural network gives better plane recall as a plane detection metric than PlaneRCNN and our method with the point cloud from SfM-MVS pipeline. We also perform experiments on plane segmentation metrics which are segmentation covering, variation of information and rand index. Like plane detection accuracy, our approach with the point cloud from estimated depthmap by deep monocular depth estimation network gives better plane segmentation performance than PlaneRCNN and our method with the point cloud from SfM-MVS pipeline.

CHAPTER 4

GROUND PLANE ESTIMATION ON UAV OUTDOOR IMAGERY WITHOUT MANUAL SUPERVISION

4.1 Introduction

With the recent advances in unmanned aerial vehicles(UAVs), image data gathered from the embedded camera on the vehicle has been easily collected. UAV imagery has an important role especially in the fields of agriculture [12], military [16], aerial photography [51], and surveillance [8] that expands the application areas of computer vision.

In recent years, Convolutional Neural Network(CNN) approaches have been proposed for different computer vision tasks on UAV imagery such as object detection [89], semantic segmentation [33], instance segmentation [64], and object recognition [69]. However, deep learning based plane detection and segmentation methods for UAV outdoor images are rarely introduced due to the lack of ground truth data as has been also explained in Chapter 2 for custom outdoor imagery.

In this part of the thesis work, we mainly focus on deep learning based ground plane detection and segmentation of UAV imagery, for which an accurate estimation is vital for the determination of the safe landing zone. Autonomous landing in motion is a critical issue for UAVs and safe landing region determination prevents possible crashes. Moreover, an unexpected emergency situation such as engine failure, connection link lost, or though weather conditions might occur during flight. In such cases, the automatic safe landing zone detection becomes essential and highly critical for which ground plane is the prominent landing region [81].

Based on these motivations, we adapt our primary proposed iterative transfer learning approach(Chapter 2) to ground plane estimation on outdoor UAV imagery without requiring manual annotation. The adapted framework for ground plane estimation on UAV outdoor imagery is illustrated in Figure 4.1.. Like modified approach explained in Chapter 3, we obtain dense point cloud from the estimated depthmap given by the monucular depth estimation network for each training image. Instead of transferring features learned from the indoors to outdoor scenes, we learn and transfer features between

outdoor images gathered by the various UAVs from several environments at different altitudes. We exploit the state-of-the-art semantic segmentation network, SeMask [44], trained on UAV outdoor images with ground truth labeling for ground plane estimation and a dense point cloud obtained from the depthmap estimated by the deep monocular depth estimation architecture as a weak supervision input to obtain estimated ground plane segmentation masks for retraining the network with target UAV outdoor imagery without requiring any manual annotation.

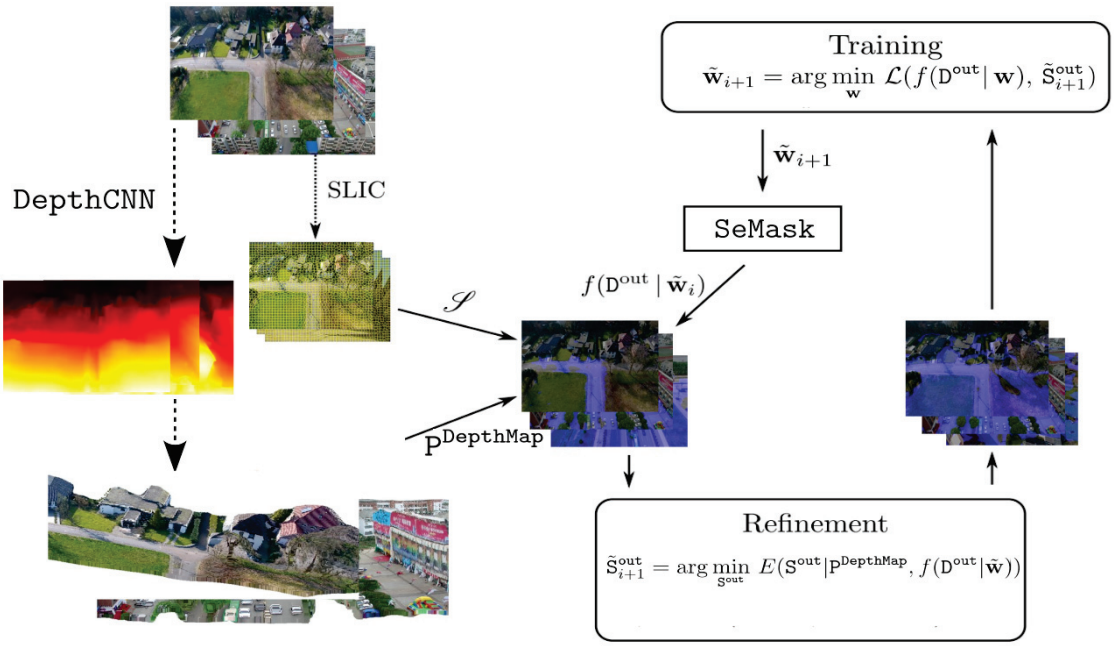


Figure 4.1.: Adapted framework for ground plane estimation on UAV outdoor imagery. A dense point cloud is obtained for each image from the estimated depthmap given by the monucular depth estimation network. We transfer the features between different UAV outdoor image datasets. We exploit the state-of-the-art semantic segmentation network, SeMask, trained on UAV outdoor images with ground truth labeling for ground plane estimation. We estimate training targets with energy minimization framework and retrain the network.

Our main contributions can be summarized as follows:

- We apply our proposed iterative transfer learning approach to UAV outdoor imagery for ground plane estimation without requiring manually annotated data.
- We show that the state-of-the-art semantic segmentation network, SeMask, trained with a manually labeled UAV outdoor image dataset for ground plane estimation can be adapted to any other UAV outdoor imagery collected from different altitudes in several environments without manual labor.

- We formulate the ground plane estimation as superpixel-based energy minimization with binary ground and non-ground labeling.

4.2 Related Work

In the literature, static landing zones can be classified as known and unknown for detection [81]. Static known landing zones are locations that known by the UAV system with coordinates and orientation. Moreover, they can be highlighted visually by marking them with different shapes or colors. However, such landing zones are appropriate only for controlled environment and are not practical and not feasible for real-time systems.

Ground plane segmentation is the most popular solution for the unknown static landing zone determination. Traditional methods use different computer vision techniques in order to detect a safe unknown static landing zone. Garg et al. [31] proposed an approach that exploits stereo vision with the aid of a monocular image-based approach. They compute surface depth information from stereo images and analyze the surface below the UAV with the roughness and slope metrics. Then, they apply a monocular image-based method for detecting non-rigid surfaces for which stereo vision could not accurately deal with it. Bosch et al. [14] introduced a method by using homography estimation and adaptive control. They detect ground planes by updating the stochastic grid. Popescu et al. [68] proposed a feature selection approach for segmentation of planar regions in aerial imagery by considering color and texture information in the scene. Yang et al. [98] introduced a monocular vision based Simultaneous Localization and Mapping(SLAM) approach for detecting a safe landing zone. Although traditional methods propose reasonable approaches for safe landing zone detection and segmentation, they do not provide a general solution to the problem and have many specific limitations.

Although there have been many deep learning based methods for various computer vision tasks, only a few approaches have been proposed for ground plane detection and segmentation including safe landing zone on UAV imagery. Perez et. al [74] introduced an approach for automatic detection of a landing zone. At first, they estimate the depth by using a CNN architecture. Then, they detect possible landing zones from a given depth map from an inception CNN called as LandNet. They trained the depth estimation deep neural architecture with the synthetic images and do not deal with real outdoor scenes. Moreover, they manually annotate training data in both networks. Li et. al [50] proposed a two-hierarchy architecture of cascaded deep neural networks for vision based autolanding. The first network called BboxLocateNet gives a coarse detection prediction by feeding it with an autolanding image dataset with annotations. This former coarse estimation is

taken by a network called PointRefine-Net which gives final coarse-to-fine landing zone prediction.

We apply the proposed idea from Uzyıldırım et al. [90] for deep learning based ground plane detection on UAV outdoor imagery without manual annotation. We exploit from the dense point cloud obtained from the monocular depth estimated by a deep neural network at training time. Combining the dense point cloud with the initial estimation of semantic segmentation network gives approximate but highly accurate ground truth annotations. We apply our iterative transfer learning approach by transferring features between different UAV outdoor images collected from various altitudes and formulate the ground plane estimation problem as a superpixel based energy minimization with binary labeling.

4.3 Adapting Proposed Iterative Transfer Learning Scheme for Ground Plane Estimation on UAV Outdoor Imagery

As it is explained in Section 2.3, we apply our iterative transfer learning scheme to piece-wise plane detection and segmentation of images belonging to outdoor scenes by exploiting the piece-wise plane reconstruction network trained on indoor images and the automatically reconstructed point cloud $\mathtt{P}^{\mathtt{MVS}}$ from SfM-MVS pipeline. We transfer features learned from the indoors to outdoor scenes by estimating training targets with superpixel based energy minimization framework based on the network weights $\mathbf{w}_{\mathtt{PlaneRCNN}}$ and $\mathtt{P}^{\mathtt{MVS}}$. Since acquisition of $\mathtt{P}^{\mathtt{MVS}}$ requires the set of overlapping images, a new method is proposed in which we exploit the deep monocular depth estimation network to obtain the 3D dense point cloud $\mathtt{P}^{\mathtt{DepthMap}}$ from the estimated depth as it is stated in Chapter 3. We replace $\mathtt{P}^{\mathtt{MVS}}$ with $\mathtt{P}^{\mathtt{DepthMap}}$ in our proposed framework and observe that overall piece-wise plane detection and segmentation quality for images of outdoor scenes improves.

For ground plane estimation on UAV outdoor imagery without requiring any manual annotation, we exploit the state-of-the-art semantic segmentation network, SeMask, trained for ground plane estimation on ground truth labeled UAV outdoor imagery collected from several environments at different altitudes than the target set of UAV outdoor images.

For a set of UAV outdoor imagery $\mathtt{D}^{\mathtt{out}}$, training targets for ground plane estimation are segmentation masks $\mathtt{S}^{\mathtt{out}}$. We initialize the network weights with $\mathbf{w}_{\mathtt{SeMask}}$ and estimate the training targets $\tilde{\mathtt{S}}^{\mathtt{out}}$ based on the network output $f(\mathtt{D}^{\mathtt{out}} \mid \mathbf{w}_{\mathtt{SeMask}})$ and a dense point cloud $\mathtt{P}^{\mathtt{DepthMap}}$ with an energy minimization framework. With the newly estimated training targets $\tilde{\mathtt{S}}^{\mathtt{out}}$, a set of weights $\mathbf{w}$ can be learned by training the network to minimize a loss

function $\mathcal{L}(\mathbf{w})$:

$$\tilde{\mathtt{S}}^{\mathtt{out}} = \underset{\mathtt{S}^{\mathtt{out}}}{\arg\min}\ E\left(\mathtt{S}^{\mathtt{out}} \mid \mathtt{P}^{\mathtt{DepthMap}},\ f(\mathtt{D}^{\mathtt{out}} \mid \mathbf{w}_{\mathtt{SeMask}})\right)$$
$$\tilde{\mathbf{w}}^{*} = \underset{\mathbf{w}}{\arg\min}\ \mathcal{L}\left(f(\mathtt{D}^{\mathtt{out}} \mid \mathbf{w}),\ \tilde{\mathtt{S}}^{\mathtt{out}}\right),$$

A better set of ground plane segmentation masks on UAV outdoor images than the ones that $\mathbf{w}_{\mathtt{SeMask}}$ gives initially can be obtained by solving the both minimization problems given in the above and obtaining more representative and informative network weights than the initial ones for the ground plane estimation on UAV outdoor imagery. To further improve the network weights and implicitly ground plane segmentation quality, our iterative approach can be considered:

$$\tilde{\mathbf{w}}_0 = \mathbf{w}_{\mathtt{SeMask}}$$
$$\tilde{\mathtt{S}}^{\mathtt{out}}_{i+1} = \underset{\mathtt{S}^{\mathtt{out}}}{\arg\min}\ E\left(\mathtt{S}^{\mathtt{out}} \mid \mathtt{P}^{\mathtt{DepthMap}},\ f(\mathtt{D}^{\mathtt{out}} \mid \tilde{\mathbf{w}}_i)\right)$$
$$\tilde{\mathbf{w}}_{i+1} = \underset{\mathbf{w}}{\arg\min}\ \mathcal{L}\left(f(\mathtt{D}^{\mathtt{out}} \mid \mathbf{w}),\ \tilde{\mathtt{S}}^{\mathtt{out}}_{i+1}\right),$$

We adapt our iterative transfer learning scheme to ground plane estimation on UAV outdoor imagery which we previously apply to piece-wise plane detection and segmentation of images belonging to outdoor scenes. We formulate the problem as a binary labeling in which each pixel of a image is assigned either ground or non-ground after our energy minimization framework. In the following, we give details of estimating initial ground plane segmentation masks and updating them by energy minimization formulation.

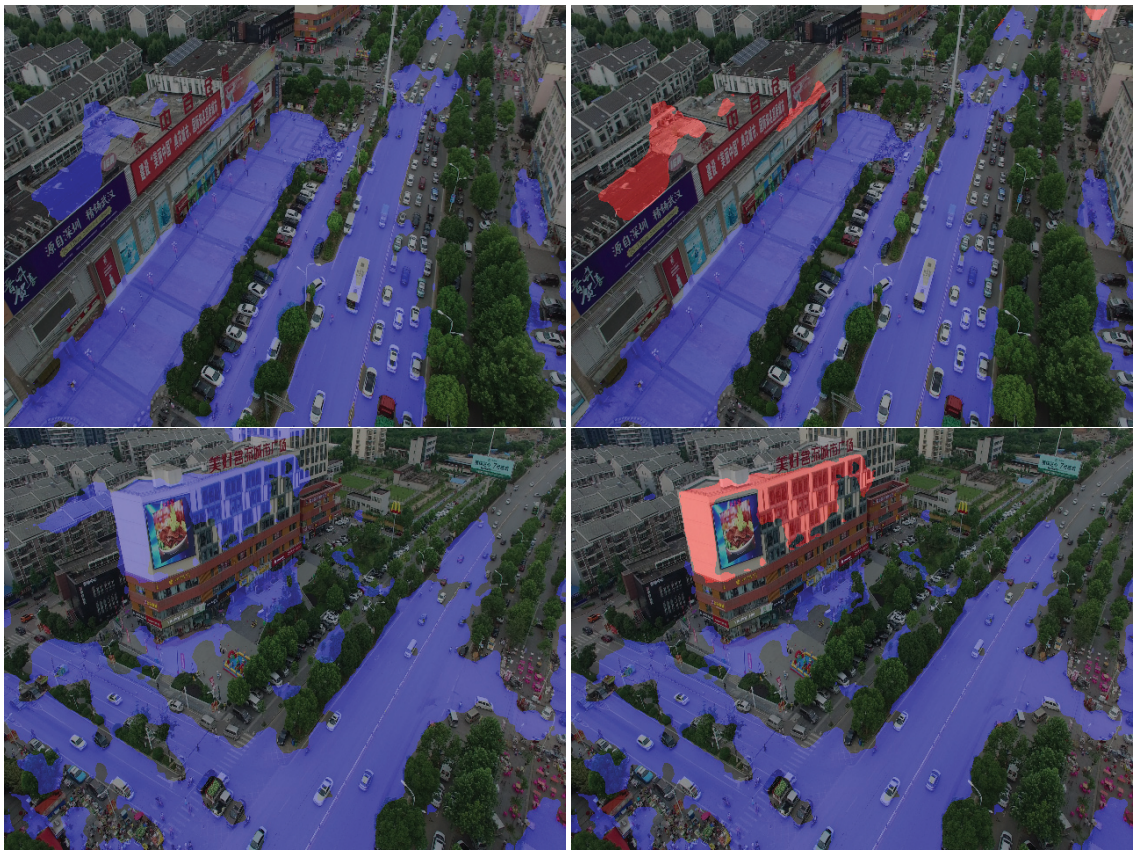
4.3.1 Estimation of Initial Ground Plane Segmentation Masks

As before, we formulate the estimation of segmentation masks as min-cut problem to be solved by graph-cuts for the superpixel labels. We formulate the ground plane estimation with binary labeling so that the segmentation masks $\mathtt{S}^{\mathtt{out}}$ is given as a set of label assignments $\{l_s : \forall s \in \mathcal{S}\}$ where $\mathcal{S}$ is the set of extracted superpixels and the labels $l_s \in \{0, 1\}$ where zero represents non-ground and one represents ground labeling.

We calculate the initial set of label assignments before ground plane segmentation masks are estimated by energy minimization formulation. We first fit a plane using RANSAC to corresponding 3D points from $\mathtt{P}^{\mathtt{DepthMap}}$ of the estimated ground plane segmentation mask by the network weights from the last iteration $f(\mathtt{D}^{\mathtt{out}} \mid \tilde{\mathbf{w}}_i)$ and assign projected inliers as the ground plane and filter out the outliers as non-ground. Each

superpixel is assigned an initial label that received the majority of the votes. We also compute the plane parameters π^{l_s} for both ground and non-ground assignments to provide geometric information for data terms in our energy minimization formulation. In order to compute the parameters of the non-ground plane π^0, we fit a plane using RANSAC to the corresponding 3D points of the initial outliers that are filtered out as non-ground.

Examples of initial ground plane segmentation mask and the points for the non-ground plane parameters π^0 computation upon the current network estimate $f(\mathtt{D}^{\mathtt{out}} \mid \tilde{\mathbf{w}}_i)$ are illustrated in Figure 4.2..



(a) Estimated ground segmentation masks by the current network weights $f(\mathtt{D}^{\mathtt{out}} \mid \tilde{\mathbf{w}}_i)$.

(b) Initial ground plane segmentation masks and the points represent the non-ground plane π^0.

Figure 4.2.: Examples for initial ground plane segmentation mask assignments. We fit a plane by using RANSAC to the corresponding 3D points of the estimated ground segmentation mask by the current network weights $f(\mathtt{D}^{\mathtt{out}} \mid \tilde{\mathbf{w}}_i)$**(a)**. We assign ground labeling to the projected points of the set of inliers(marked with blue) and filter out the outliers as non-ground. In order to compute the plane parameters for the non-ground plane π^0, we fit a plane using RANSAC to the initial outliers and resulting inliers(marked with red) represent the non-ground plane **(b)** .

4.3.2 Updating the Segmentation Masks by Energy Minimization

We define the energy of a superpixel segmentation mask $\mathtt{S}^{\mathtt{out}} = \{l_s\}$ based on the point cloud $\mathtt{P}^{\mathtt{DepthMap}}$ and the network weights $\tilde{\mathbf{w}}_i$ obtained in the last iteration as follows:

$$E\left(\mathtt{S}^{\mathtt{out}}\right) = \sum_{s \in \mathcal{S}} E_d\left(l_s \mid \mathtt{P}^{\mathtt{DepthMap}},\ f(\mathtt{D}^{\mathtt{out}} \mid \tilde{\mathbf{w}}_i)\right) + \lambda_s \sum_{(s,t) \in \mathcal{N}_{\mathcal{S}}} E_s(l_s, l_t \mid \mathtt{P}^{\mathtt{DepthMap}},\ f(\mathtt{D}^{\mathtt{out}} \mid \tilde{\mathbf{w}}_i)),$$

where $\mathcal{N}_{\mathcal{S}}$ is the set of neighboring superpixels.

$E_d\left(l_s \mid \mathtt{P}^{\mathtt{DepthMap}},\ f(\mathtt{D}^{\mathtt{out}} \mid \tilde{\mathbf{w}}_i)\right)$ is the energy data term that measures the disagreement between a given superpixel label l_s and the point cloud $\mathtt{P}^{\mathtt{DepthMap}}$. It is composed of two individual terms $E_{\text{support}}(l_s \mid \mathtt{P}^{\mathtt{DepthMap}})$ and $E_{\text{distance}}(l_s \mid \mathtt{P}^{\mathtt{DepthMap}})$ and these components are combined with corresponding weights as follows:

$$E_d(l_s) = \alpha_1 E_{\text{support}}(l_s \mid \mathtt{P}^{\mathtt{DepthMap}}) + \alpha_2 E_{\text{distance}}(l_s \mid \mathtt{P}^{\mathtt{DepthMap}}),$$

where α_1 and α_2 are scalar constants.

$E_{\text{support}}(l_s \mid \mathtt{P}^{\mathtt{DepthMap}})$ measures the ratio of projected 3D points assigned to the same plane corresponding to the label l_s. It is computed as

$$E_{\text{support}}(l_s \mid \mathtt{P}^{\mathtt{DepthMap}}) = \frac{n_t - n_s}{n_t},$$

where n_s is the number of projected points in the superpixel assigned to the corresponding plane and n_t is the total number of projected points in the superpixel.

$E_{\text{distance}}(l_s \mid \mathtt{P}^{\mathtt{DepthMap}})$ measures the median distance of projected 3D points to the plane corresponding to the label l_s. It is computed as

$$E_{\text{distance}}(l_s \mid \mathtt{P}^{\mathtt{DepthMap}}) = med_{p \in s} d(\pi^{l_s}, p),$$

where $d(\pi^{l_s}, p)$ is the 3D Euclidean distance between the 3D point p and plane π^{l_s}. Instead of taking average point to plane distance as before, we are taking the median distance since inconsistent depth values might be estimated from the monocular depth estimation network for some points which directly effects the average value. An ablation is study is performed for this modification and corresponding results are given in Section 4.4.2.3.

$E_s(l_s, l_t \mid \mathtt{P}^{\mathtt{DepthMap}},\ f(\mathtt{D}^{\mathtt{out}} \mid \tilde{\mathbf{w}}_i))$ is the smoothness data term and calculated as

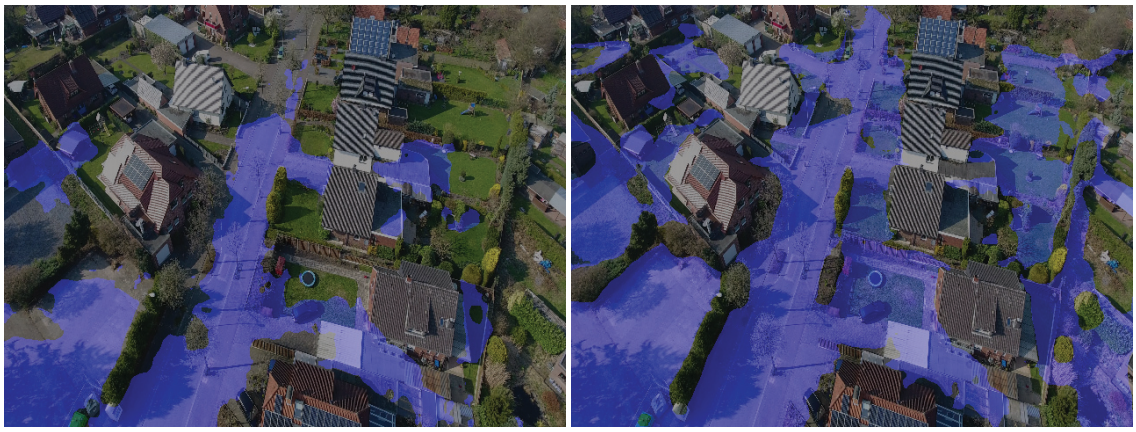
$$E_s(l_s, l_t) = E_{\text{color}}\left(l_s, l_t \mid \mathtt{P}^{\mathtt{DepthMap}},\ f(\mathtt{D}^{\mathtt{out}} \mid \tilde{\mathbf{w}}_i)\right)$$

where $E_{\text{color}}(l_s, l_t)$ penalizes different label assignments in smooth intensity regions and it is calculated as

$$E_{\text{color}}(l_s, l_t) = \exp(-\Delta_c)$$

where Δ_c is the difference between mean intensity values (average of color channels) over superpixels s and t. For smoothness term, we do not penalize label changes for similar depth regions since our aim is correctly separating of a ground plane from an object on it.

An example ground plane segmentation after the energy minimization is illustrated in Figure 4.3..



(a) Ground plane segmentation from the current network weights $f(\mathtt{D}^{\mathtt{out}} \mid \tilde{\mathbf{w}}_i)$.

(b) Ground plane segmentation after energy minimization.

Figure 4.3.: Ground plane segmentation after energy minimization upon the estimation from the current network weights $f(\mathtt{D}^{\mathtt{out}} \mid \tilde{\mathbf{w}}_i)$.

4.4 Experiments

We have performed a set of experiments to show our proposed deep learning based framework improves the outdoor UAV ground plane estimation performance of a state-of-the-art semantic segmentation network trained on UAV outdoor image collection gathered from a several environments where images are taken from the different altitudes than the target imagery. We have used low-altitude and high-altitude outdoor UAV image datasets in our experiments. Both benchmarks include appropriate imagery that contains the necessary ground plane.

We show that our approach improves ground plane segmentation estimation quality even with a single iteration by performing quantitative experiments on both benchmarks. We also demonstrate that applying transfer learning scheme iteratively enhances the accuracy of ground plane segmentation. Qualitative results are given in order to show the performance of our approach in terms of ground plane estimation quality advances as the

number of iterations increases. Furthermore, we perform an ablation study to observe the contribution of our superpixel based energy minimization framework which is detailed in Section 4.3.2 to estimate training targets for retraining upon the initial segmentation masks computed by the naive plane fitting explained in Section 4.3.1.

4.4.1 Benchmarks

We have used low-altitude and high-altitude outdoor UAV image datasets in our experiments. In the following, the details of the both benchmarks are given.

4.4.1.1 Low-altitude UAV outdoor image dataset

We have used `Semantic Drone` dataset [1] as a low-altitude UAV outdoor image dataset which is composed for the purpose of semantic understanding on various urban scenes. There are 390 images in the dataset acquired at an altitude of 5 to 30 meters above ground. Example images from the dataset are shown in Figure 4.4.. We have used 240 images for training and 75 images for both validation and test splits.

The semantic classes of the `Semantic Drone` dataset are tree, grass, other vegetation, dirt, gravel, rocks, water, paved area, pool, person, dog, car, bicycle, roof, wall, fence, window, door, and obstacle. Since we are interested in ground plane, we just get the paved area and grass labeling from the annotation and manually annotate remaining target ground planes in the scene. Example annotations are illustrated in Figure 4.5..

4.4.1.2 High-altitude UAV outdoor image dataset

We have used `UAVid` video dataset [60, 59] as a high-altitude UAV outdoor image dataset composed for semantic segmentation especially for urban scenes. The videos are gathered approximately at an altitude of 60 meters. There are training, validation and test splits in the dataset for which 20, 7, and 15 sequences exist respectively. Each sequence is composed of 10 frames. Example images from the dataset are illustrated in Figure 4.6.. We have used sequences that include at least one dominant ground plane based on the corresponding scene and exclude the others that results in 170 images for training, 50

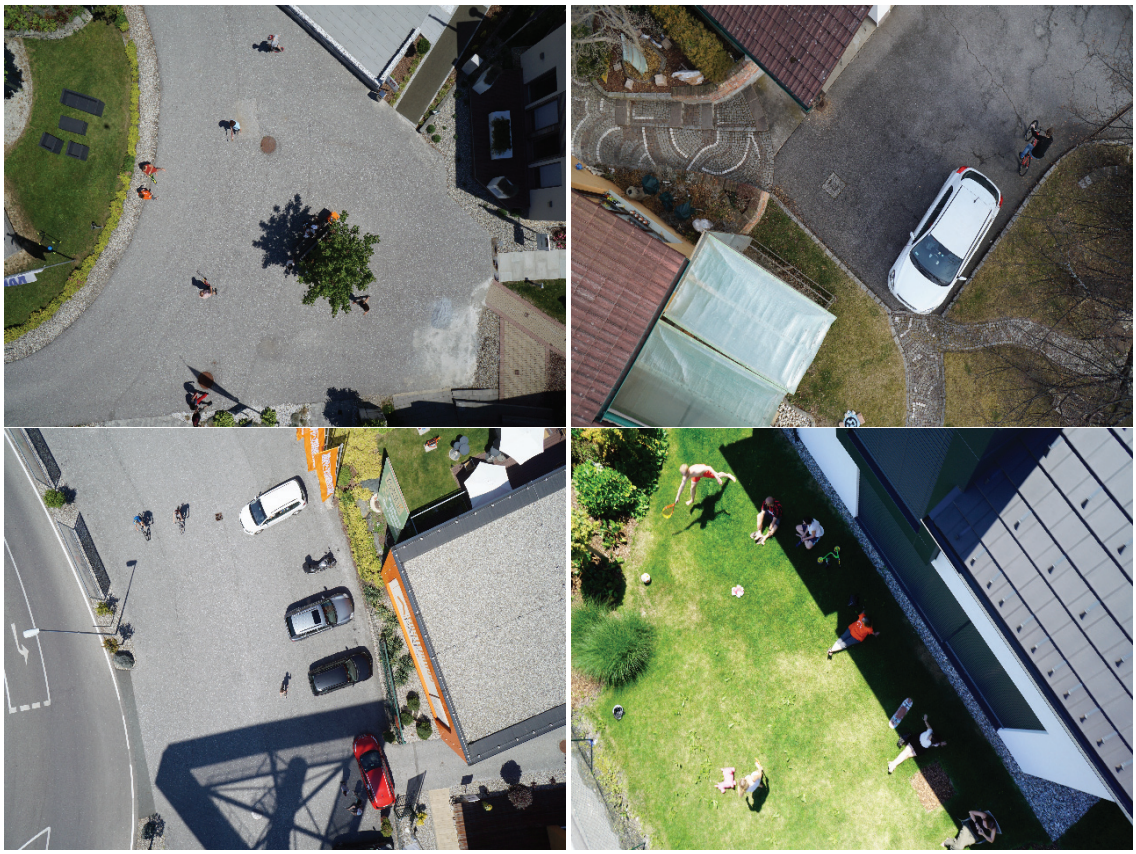


Figure 4.4.: Example images from the `Semantic Drone` dataset.

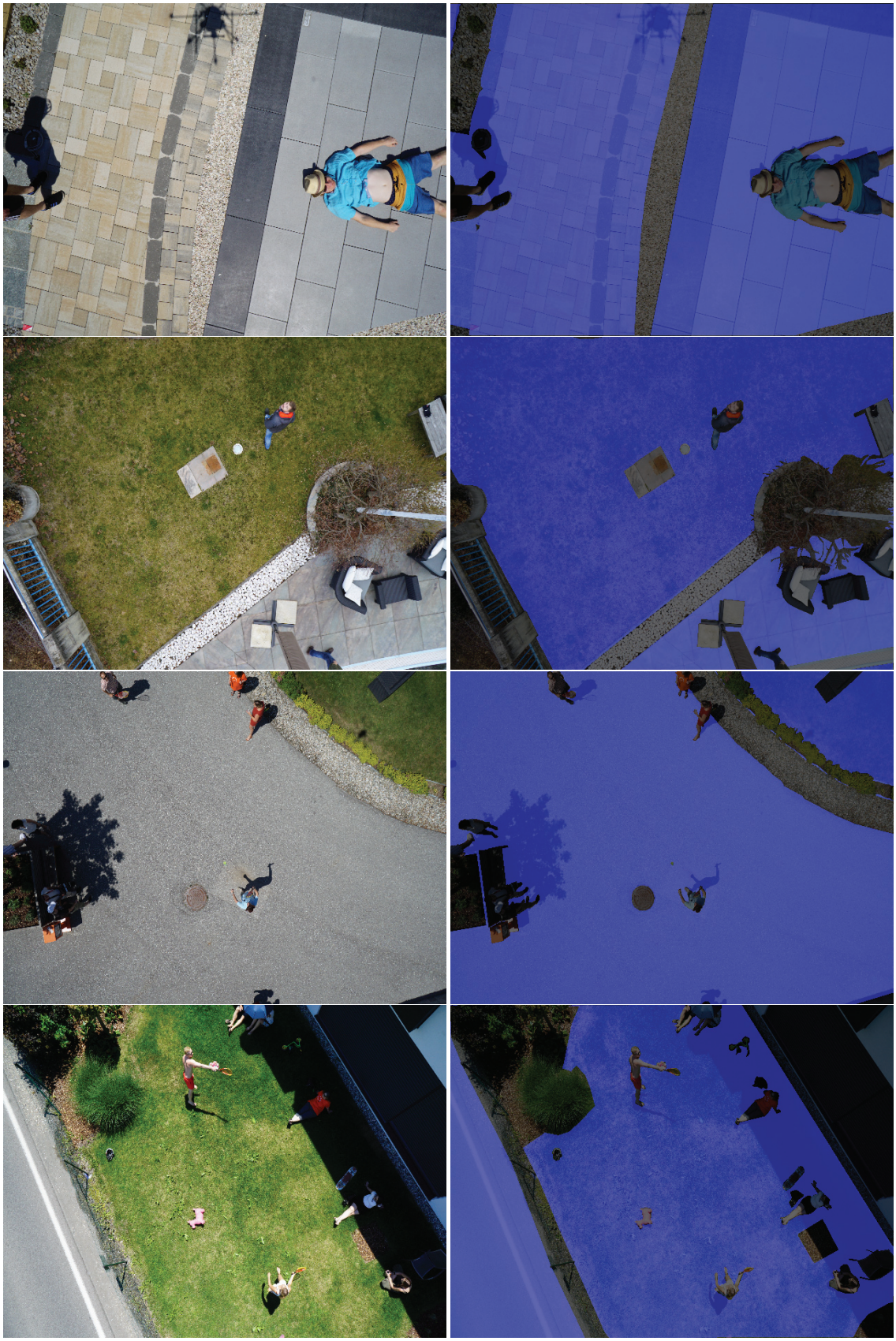
samples for validation, and 80 images for test.

The semantic categories of the dataset are building, road, static car, tree, low vegetation, human, moving car, and background clutter. We only get the road labeling from the annotation and manually annotate the remaining target ground planes in the scene. Example annotations are shown in Figure 4.7..

4.4.2 Experiment Setup and Results

We compute SLIC [4, 5] superpixels as shown in Figure 4.8.. We extract 1500 superpixels from each training image from both datasets.

We have used two different datasets so we have two main sets of experiments. For both experiment setups, the set of weights of the SeMask network is previously computed for semantic segmentation on Cityscapes dataset [21]. In order to measure the ground plane estimation performance on `Semantic Drone` dataset, we initialize the SeMask network with pretrained weights obtained by using the `UAVid` dataset for the task of ground plane estimation with ground truth annotations and measure its ground plane segmentation



(a) Input Image (b) Ground truth ground plane annotation

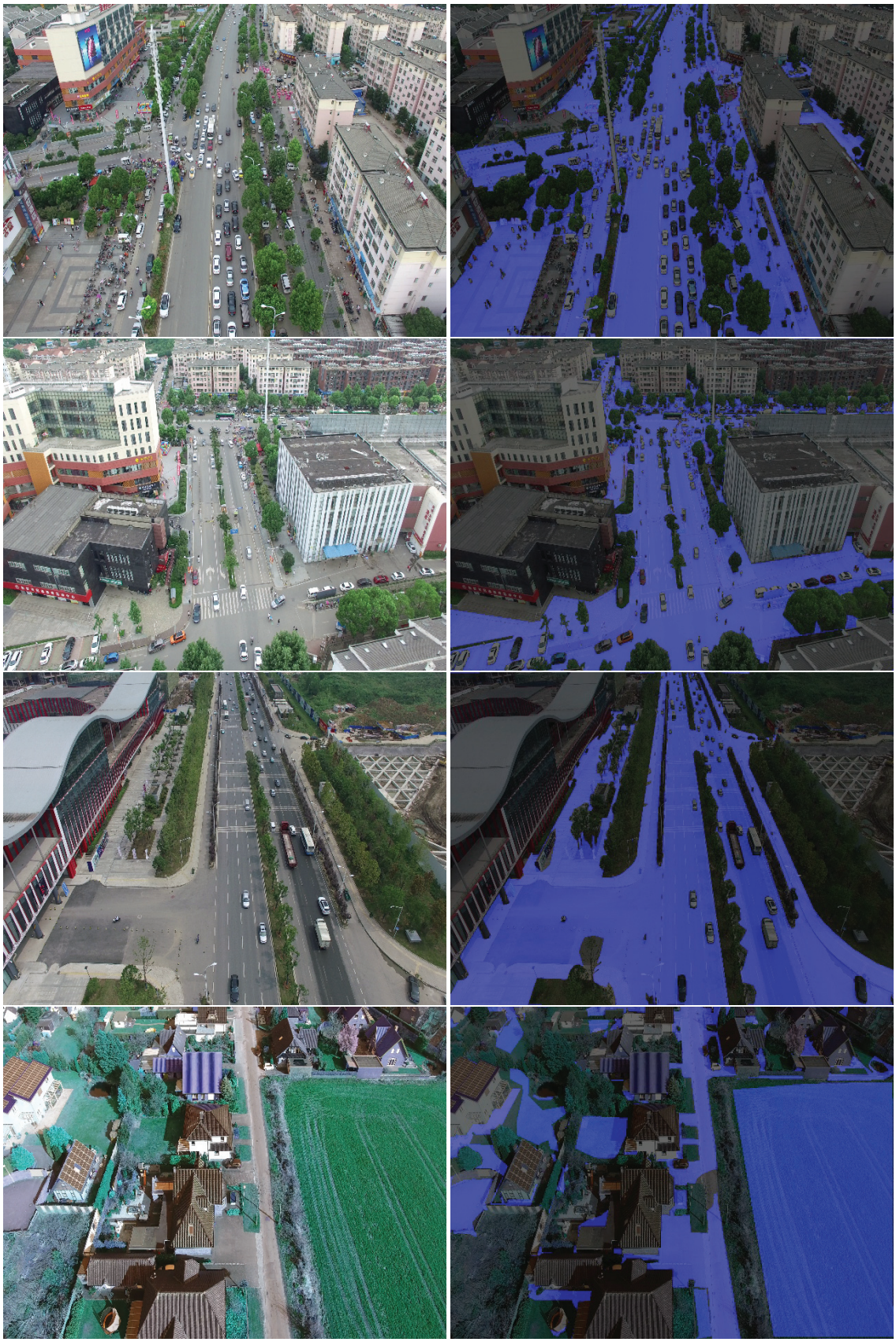
Figure 4.5.: Example ground truth ground plane annotations from `Semantic Drone` dataset.

Figure 4.6.: Example images from the UAVid dataset.

performance on the test set of Semantic Drone dataset. As it is stated in Section 4.3, ground plane segmentation masks as training targets on the Semantic Drone dataset for retraining are estimated by superpixel based energy minimization framework. Then, all layers of the SeMask are retrained using the estimated training targets for 500 epochs. We keep the set of weights that gives the highest ground plane segmentation estimation accuracy for the validation set as a training result. For measuring the ground plane estimation performance on UAVid dataset, we apply the same process for the Semantic Drone dataset by replacing the datasets. The training takes approximately 6 hours for each dataset. Furthermore, we set the scalar weights of the energy function detailed in Section 4.3.2 by a grid search that results in the best ground plane estimation performance for a validation set.

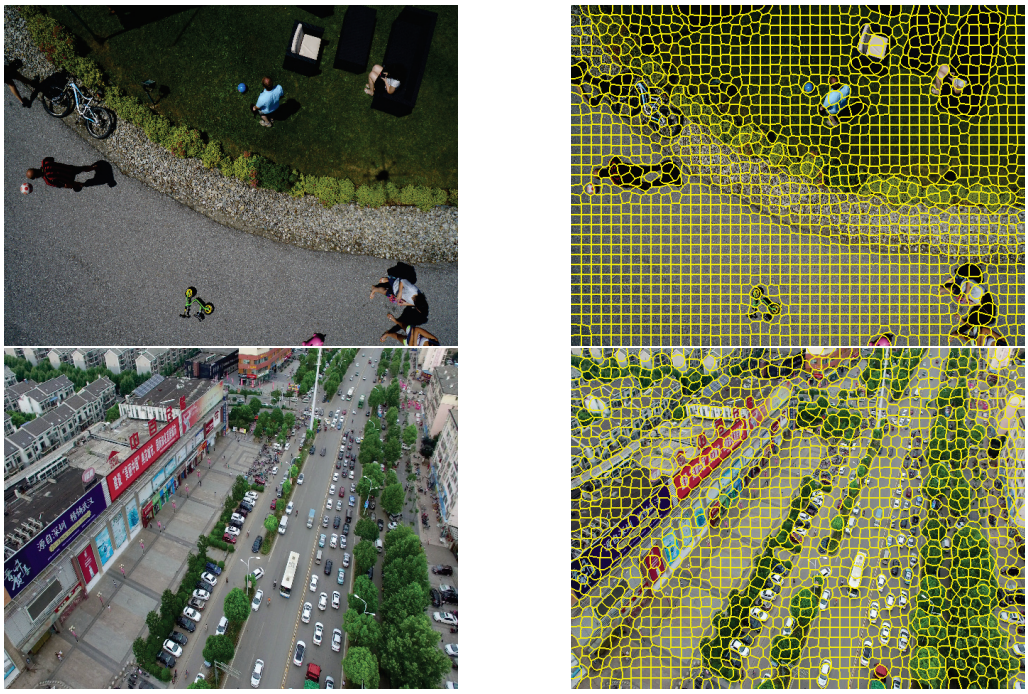
We measure two metrics to evaluate ground plane segmentation quality on UAV outdoor imagery. *Mean Intersection over Union(mIoU)* is defined as the amount of overlap between the estimated and ground truth plane segmentation masks over all test images. We also measure *Mean Non-ground Accuracy(mNgAcc)* which is the ratio of the number of points that are correctly excluded as non-ground to the number of non-ground points in the ground truth ground plane segmentation mask over all test images.

In order to obtain the dense point cloud, we get the depthmap estimation from the state-of-the art monocular depth estimation neural network as it is stated in Chapter 3. An example for an estimated depthmap and corresponding 3D dense point cloud is illustrated in Figure 4.9..



(a) Input Image (b) Ground truth ground plane annotation

Figure 4.7.: Example ground truth ground plane annotations from `UAVid` dataset.



(a) Input Images (b) SLIC superpixels from (a)

Figure 4.8.: Extracted SLIC superpixels from images of both datasets.

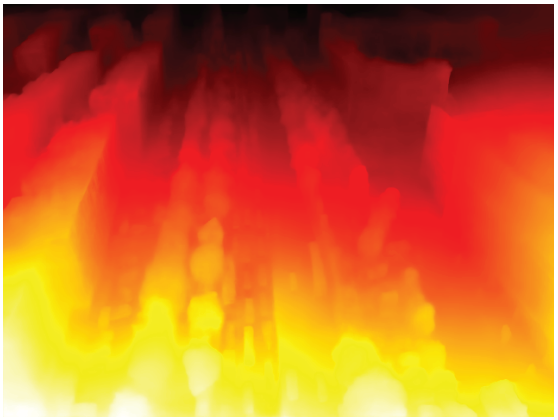
4.4.2.1 Ground Plane Estimation on Semantic Drone Dataset

In order to measure the performance of our proposed iterative transfer learning approach for ground plane estimation on `Semantic Drone` dataset, we train the SeMask network from scratch with the ground truth training data of `UAVid` dataset. We initialize the SeMask network with the pretrained weights from the `UAVid` dataset and retrain with the estimated training targets of `Semantic Drone` dataset obtained from our superpixel based energy minimization framework. We perform four iterations in each experiment since improvement slows down after iteration four.

Table 4.1. shows the performance of our approach for the `Semantic Drone` dataset experiments. As the table shows, our approach performs better than SeMask even with a single training iteration for both $mIoU$ and $mNgAcc$ metrics. Our performance becomes significantly better than the SeMask as the number of iterations increases.

Figure 4.10. shows ground plane segmentation results for a test image from `Semantic Drone` dataset for a qualitative comparison between the SeMask output and our ground plane segmentation as the number of iterations increases. SeMask trained with `UAVid` dataset misses most of the regions of the ground plane. The same architecture retrained on images of `Semantic Drone` dataset by the proposed approach is able to detect most of the parts of the ground plane even with a single iteration. As the number of iterations increases, the network is able to find ground plane regions of the scene missed

(a) Input training image.



(b) Depthmap estimation.

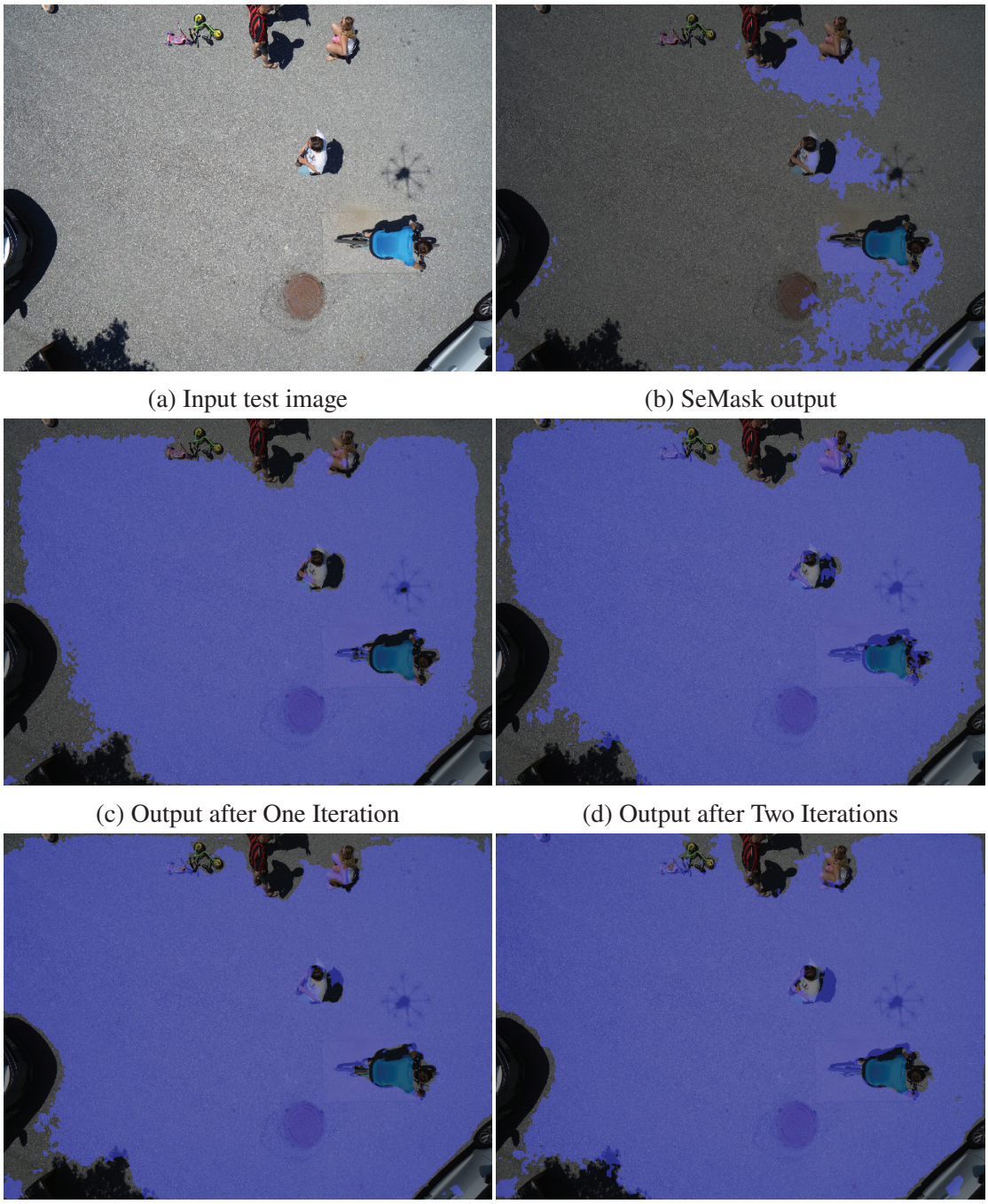
(c) 3D dense point cloud.

Figure 4.9.: Illustration of 3D dense point cloud**(c)** obtained from the depthmap given by deep monocular estimation network**(b)**.

Table 4.1.: *mIou* and *mNgAcc* results for the `Semantic Drone` dataset as the number of training iterations is increased.

	mIoU	***mNgAcc***
SeMask	0.308	0.403
One Iteration	0.461	0.512
Two Iterations	0.524	0.565
Three Iterations	0.561	0.588
Four Iterations	**0.579**	**0.601**

in early iterations.



(a) Input test image (b) SeMask output

(c) Output after One Iteration (d) Output after Two Iterations

(e) Output after Three Iterations (f) Output after Four Iterations

Figure 4.10.: Comparison of ground plane segmentation maps for a test image from `Semantic Drone` dataset**(a)**. SeMask trained on other UAV image dataset misses most of the ground plane**(b)**. Even after one iteration, the retrained network with our proposed approach is able to find most of the missed ground plane regions by SeMask**(c)** and the ground plane estimation improves as the number of iterations increases**(d-f)**.

4.4.2.2 Ground Plane Estimation on UAVid Dataset

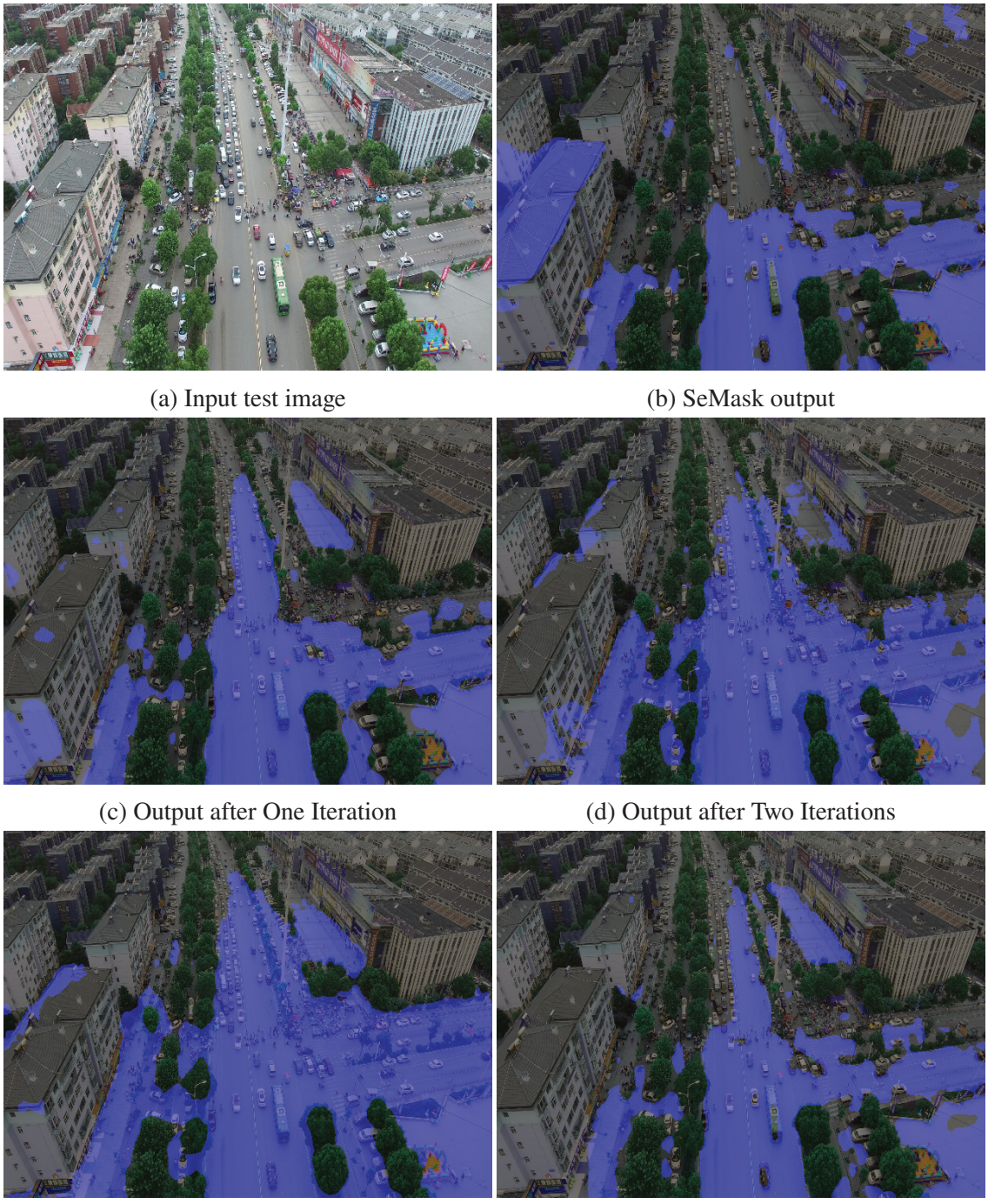
We measure the ground plane segmentation performance of our proposed iterative transfer learning approach on `UAVid` dataset by training the SeMask network from scratch with the ground truth training data of `Semantic Drone` dataset and initialize the network with this set of pretrained weights. We retrain the network with the estimated training targets of `UAVid` dataset obtained from our energy minimization framework. As in the experiments on `Semantic Drone` dataset, we stop the iterations at four since ground plane estimation improvement slows down after the fourth iteration.

Our ground plane segmentation performance on `UAVid` dataset with compared to the SeMask network is shown in Table 4.2.. We obtain better $mIoU$ and $mNgAcc$ than the SeMask even with a single iteration. As the number of iterations increases, our estimation becomes much better than the SeMask.

Table 4.2.: $mIou$ and $mNgAcc$ results for the `UAVid` dataset as the number of training iterations is increased.

	mIoU	***mNgAcc***
SeMask	0.349	0.729
One Iteration	0.474	0.782
Two Iterations	0.493	0.842
Three Iterations	0.528	0.867
Four Iterations	**0.541**	**0.873**

Ground plane segmentation results for a test image from `UAVid` dataset for a qualitative comparison between the SeMask output and our estimations are given in Figure 4.11.. Most of the parts of the ground plane are missed by the initial SeMask network and also assign some non-ground regions as ground. The retrained network obtained with the `Semantic Drone` dataset by our proposed approach is able to find missing ground regions and decreases the amount of non-ground points that are incorrectly assigned as ground plane even with a single iteration. As the number of iterations increases, the ground plane estimation quality on the scene improves.



(a) Input test image (b) SeMask output

(c) Output after One Iteration (d) Output after Two Iterations

(e) Output after Three Iterations (f) Output after Four Iterations

Figure 4.11.: Ground plane segmentation estimation for a test image from `UAVid` dataset**(a)**. SeMask trained on other UAV image dataset misses most of the ground plane and assign many non-ground points as ground**(b)**. Even after one iteration, the retrained network with our proposed approach is able to find most of the missed ground plane regions by SeMask and reduces the incorrectly assigned regions as ground**(c)**. As the number of iterations increases, the ground plane estimation and the removal of non-ground points assigned as ground improves. **(d-f)**.

4.4.2.3 Ablation Study

We perform two sets of ablation studies. First, we observe the effect of update in computing E_{distance} from taking average to median while estimating training targets. We also measure the contribution of superpixel-based energy minimization on estimation of training targets($\mathtt{S_f}^{\mathtt{out}}$) upon the initialization($\mathtt{S_i}^{\mathtt{out}}$) with naive plane fitting.

As it is explained in Section 2.3.3, we take the average point to plane distances for the data term. However, some depth values computed from deep monocular estimation network might be inconsistent which results in irrelevant point to plane distance. In order to deal with it, we take the median point to plane distance which is explained in Section 4.3.2. To show the contribution of this modification, we measure $mIou$ and $mNgAcc$ for the `Semantic Drone` dataset while estimating training targets on `UAVid` dataset by computing E_{distance} with average $E_{\text{distance}-avg}$ and with median $E_{\text{distance}-med}$. Table 4.3 shows that taking median to compute E_{distance} gives much better test results for both metrics than taking the average.

Table 4.3.: $mIou$ and $mNgAcc$ results for `Semantic Drone` dataset when taking the average($E_{\text{distance}-avg}$) and the median($E_{\text{distance}-med}$) to compute E_{distance} for estimating training targets of `UAVid` dataset.

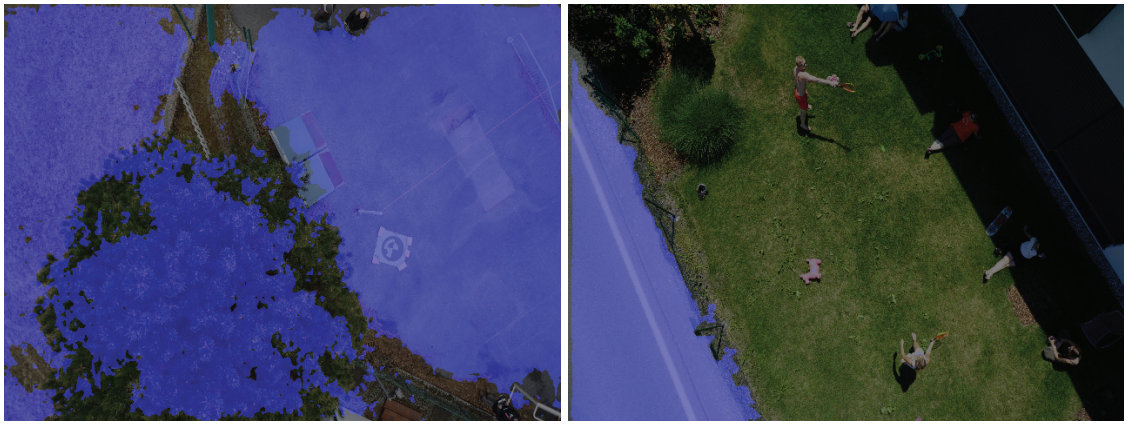
	mIoU	***mNgAcc***
`Semantic Drone` - SeMask	0.308	0.403
`Semantic Drone` - $E_{\text{distance}-avg}$	0.382	0.447
`Semantic Drone` - $E_{\text{distance}-med}$	**0.579**	**0.601**

Figure 4.12. shows qualitative comparison when taking average($E_{\text{distance}-avg}$) and median($E_{\text{distance}-avg}$) to compute E_{distance}. Results show that using $E_{\text{distance}-avg}$ as a data term for estimating training targets leads to have under-segmented or over-segmented ground plane estimations.

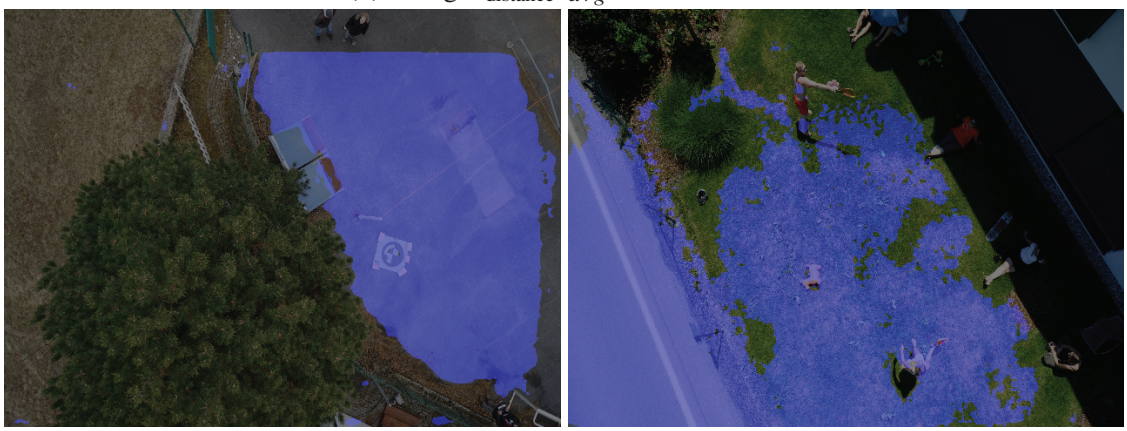
As it is stated in Section 4.3.1, we fit a plane using RANSAC to corresponding 3D points of the estimated ground plane segmentation mask by the network weights and assign projected inliers as the ground plane. Based on such initial label assignments, we apply superpixel based energy minimization framework which is explained in Section 4.3.2 and retrain the network with the estimated training targets.

We show the contribution of the superpixel based energy formulation for the ground plane estimation on both `Semantic Drone` and `UAVid` datasets. We compare the ground plane estimation quality on test images by using the training targets from

(a) Images from `Semantic Drone` dataset.



(b) Using $E_{\text{distance}-avg}$ as a data term.



(c) Using $E_{\text{distance}-med}$ as a data term.

Figure 4.12.: Comparison of ground plane estimation for test images from `Semantic Drone` and `UAVid` datasets by using the data term $E_{\text{distance}-avg}$ and $E_{\text{distance}-med}$ for estimating training targets. Due to some inconsistent depth estimations from the monocular depth estimation network, taking the average point to plane distance for estimating training targets results in over-segmented or under-segmented ground plane masks which leads to have similar test results

initial labeling(S_i^{out}) and final energy-based estimation(S_f^{out}) to retrain the network. The network is retrained four times and the *mIou* and *mNgAcc* values after fourth iteration are shown in Table 4.4.. The results show that estimating training targets from the energy minimization framework contributes ground plane segmentation performance of our proposed iterative transfer learning approach. Although training targets obtained based on naive plane fitting for initial estimation also provide the network to learn informative features for the target domain, the improvement on the ground plane estimation quality is not as much good as obtaining training targets from the energy minimization framework.

Table 4.4.: *mIou* and *mNgAcc* results after four iterations for both `Semantic Drone` and `UAVid` datasets using S_i^{out} and S_f^{out} as training targets.

	mIoU	***mNgAcc***
`Semantic Drone` - SeMask	0.308	0.403
`Semantic Drone` - S_i^{out}	0.434	0.404
`Semantic Drone` - S_f^{out}	**0.579**	**0.601**
`UAVid` - SeMask	0.349	0.729
`UAVid` - S_i^{out}	0.521	0.845
`UAVid` - S_f^{out}	**0.541**	**0.873**

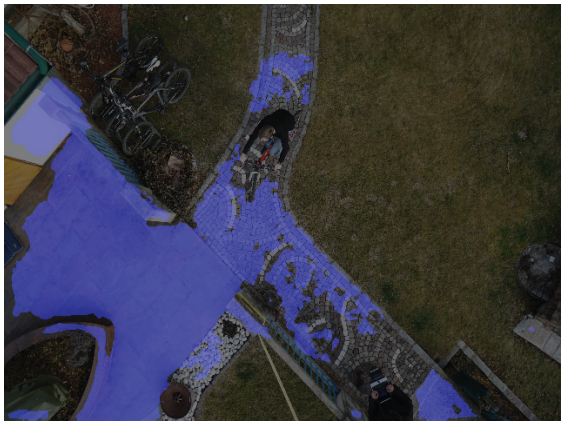
Figure 4.13. shows the ground plane segmentation estimation for test images from `Semantic Drone` dataset and `UAVid` dataset by using the both S_i^{out} and S_f^{out} as training targets. The results correspond to network output after four iterations. Using S_i^{out} as training targets yields to undersegmentation of ground plane especially for the test image of `Semantic Drone` dataset. For the test image of `UAVid` dataset, S_i^{out} as estimated training targets results in assigning non-ground points on the scene for instance vehicles on the road as ground regions. S_f^{out} , as training targets, finds most of the ground plane parts of the scene and decreases the amount of non-ground points incorrectly assigned as ground for the test images from both datasets.

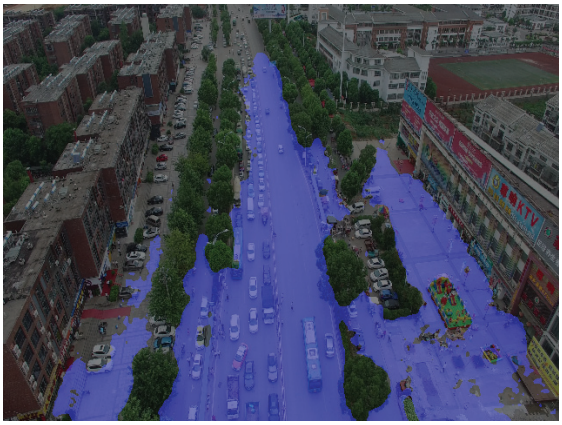
4.5 Conclusion

In this part of the thesis work, we apply our proposed iterative transfer learning approach to UAV outdoor imagery for ground plane segmentation without requiring manual annotations. We initialize the state-the-of-art semantic segmentation neural network, SeMask, with the pretrained weights obtained after training with UAV outdoor image dataset

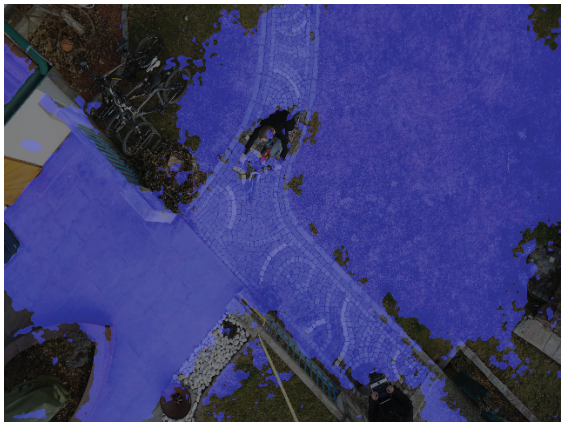
(a) Image from Semantic Drone dataset.

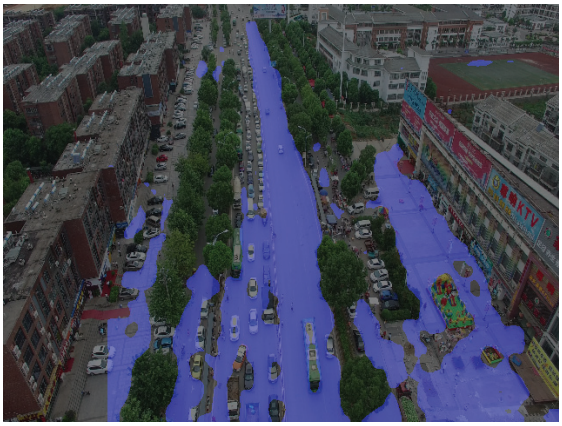
(b) Image from UAVid dataset.





(c) Using S_i^{out} as training targets.





(d) Using S_f^{out} as training targets.

Figure 4.13.: Comparison of ground plane estimation for test images from Semantic Drone and UAVid datasets by using the training targets S_i^{out} and S_f^{out}. Results belong to estimations obtained after fourth training iteration. Using S_i^{out} as training targets undersegments the ground plane regions and incorrectly assign non-ground points as ground plane. Retraining the network with S_f^{out} as training targets finds most of the ground plane and reduces the amount of non-ground points incorrectly assigned as ground.

with ground truth ground plane segmentation masks. The network is then retrained with another UAV outdoor image dataset collected from different altitudes than the ground truth UAV outdoor image dataset. Targets for retraining are obtained from superpixel based energy minimization framework without requiring any manual annotation.

We perform experiments on both low-altitude and high-altitude UAV image datasets to apply our iterative transfer learning approach for both benchmarks. Training targets obtained from our superpixel based energy minimization framework improves the network weights and consequently the estimate of the ground truth. Both quantitative and qualitative evaluations show that our approach improves the ground plane estimation quality on various UAV outdoor imagery.

CHAPTER 5

CONCLUSION

In this thesis, developing a deep learning based framework for piece-wise plane detection and segmentation of outdoor scenes without requiring any manual annotation is the main purpose. We combine the traditional and CNN-based piece-wise plane detection and segmentation approaches to achieve this. An iterative transfer learning scheme is proposed for which features are transferred from a network trained on ground truth targets to outdoor imagery without requiring any manual labor. We exploit automatically reconstructed dense point cloud and a network trained on images with ground truth labeling to obtain approximate but highly accurate training targets with a superpixel based energy minimization formulation. Retraining the network with these estimated targets makes the network weights more informative and representative than the weights of the initial network for the piece-wise plane detection and segmentation of images of outdoor scenes. Furthermore, applying this transfer scheme iteratively further improves the overall piece-wise plane detection and segmentation quality.

Since indoor scenes have the advantage of accessibility of large training sets thanks to easy depth sensing with the aid of active sensors, most of the CNN-based piece-wise plane reconstruction methods are designed and trained with indoor imagery. At first, we apply our proposed framework to transfer features from a piece-wise plane reconstruction network trained on indoor images to outdoor imagery. We initialize the weights of the network with the state-of-the-art deep piece-wise plane reconstruction architecture trained on indoor images, PlaneRCNN. Based on the initial estimate of PlaneRCNN for the set of outdoor images, we estimate the approximate segmentation targets for retraining from our superpixel based energy minimization framework under the guidance of 3D dense point cloud obtained from SfM-MVS pipeline. We then apply proposed iterative transfer learning scheme to make the network weights more informative and representative than the baseline for piece-wise plane detection and segmentation of outdoor imagery. To demonstrate the performance of our proposed approach, we perform experiments on SfM and SLAM benchmarks that provide suitable imagery of structured urban scenes. Results show that our proposed deep learning based framework improves piece-wise plane detection and segmentation quality of outdoor images without requiring manual labor by applying iterative transfer learning scheme from the state-of-the-art piece-wise plane reconstruction network trained on indoor images to target outdoor domain weakly

supervised by 3D dense point cloud.

We exploit 3D dense point cloud as a weak supervision signal in our proposed deep learning based framework for piece-wise plane detection and segmentation of outdoor images without manual annotation. Obtaining the point cloud from SfM-MVS pipeline requires overlapping set of images and textured surfaces which limits the application of our approach to apply custom outdoor imagery. In order to remove the requirement of overlapping set of outdoor images with textured surfaces, we exploit a state-of-the-art monocular depth estimation network. We compute a dense point cloud from the estimated depth given by the network. To observe the effect of this small but effective modification to our proposed framework, we perform the same set of experiments as before and compare the results obtained by computing the dense point cloud from SfM-MVS pipeline and monocular depth estimation network. We also measure plane segmentation metrics to observe the effect of the modification to overall piece-wise plane detection and segmentation quality. Experiments show that obtaining 3D dense point cloud from the depth estimated from monocular depth estimation network instead of SfM-MVS pipeline improves the piece-wise plane segmentation quality for outdoor images.

In order to test the applicability of our deep learning based proposed framework for different outdoor domains, we employ it to assess the ground plane estimation problem which can be considered as a subset of an overall piece-wise plane detection and segmentation problem on UAV outdoor imagery. Generally, UAV outdoor imagery is classified based on the altitude that the images are gathered from. So, we apply our iterative transfer scheme to different UAV outdoor image benchmarks collected from various altitudes in several environments. We initialize the state-of-the-art semantic segmentation network, SeMask, with the annotated set of UAV outdoor images. Estimated targets of another UAV outdoor image dataset collected from different altitudes than the annotated imagery for retraining are obtained from our superpixel based energy minimization framework under the guidance of dense point cloud obtained from the depth given by the monocular depth estimation network. We perform experiments on both low-altitude and high-altitude UAV outdoor image datasets. Results show that our proposed framework improves the ground plane estimation quality by applying a transfer learning scheme from a semantic segmentation network trained on manually annotated UAV outdoor images to different UAV outdoor imagery without requiring any manual labor.

To conclude, we have proposed and developed a deep learning based iterative transfer learning framework for piece-wise plane detection and segmentation of outdoor scenes without requiring manual annotation. The lack of ground truth annotations for the outdoor imagery can be compensated by approximate training targets obtained from superpixel-based energy minimization formulation. This formulation is based on neural network trained on data with ground-truth labeling and automatically reconstructed 3D dense point cloud. We apply the proposed method for the task of piece-wise plane detection

and segmentation for the images of outdoor scenes and ground plane estimation on UAV outdoor imagery and show that our framework achieves improved plane estimation on different outdoor image domains without requiring any manual annotation.

5.1 Discussion and Future Work

The success of deep learning on most of the traditional computer vision tasks comes from the opportunity to access huge amounts of annotated training data. This enables to make the performance of CNNs better than conventional methods for various vision problems while keeping the number of parameters relatively small. With the advances in hardware like GPUs and storage for a huge amount of data especially owned by global software companies, some deep neural architectures provide a hundred percent test accuracy even for real-time applications. However, the lack of supervised data opens up another problem space. Transfer learning or semi-supervised learning approaches can be applied to deal with such situation, but these methods also require ground truth labeled data. So, there is a need for approaches to compensate for the lack of ground truth training data in deep learning-based solutions.

Most of the CNNs for piece-wise plane detection and segmentation are not designed and trained for outdoor scenes due to lack of manually annotated data. In this thesis, a novel deep learning-based framework is introduced for the piece-wise plane detection and segmentation of outdoor images without requiring manual labor. We propose an iterative transfer learning scheme in which we exploit a network trained on ground truth labeled imagery and an automatically reconstructed point cloud as a weak supervision signal to estimate approximate but highly accurate training targets. We show that our proposed and developed approach improves the piece-wise plane detection and segmentation quality of various outdoor image domains.

As it is stated in the beginning, the research focus on deep learning has been shifted to the problem of dealing with the lack of ground truth training data. A self-supervised based approach could be applied for piece-wise plane detection and segmentation of outdoor imagery for future research in which training labels are obtained from the data itself and have been applied for different computer vision problems from a couple of years. Furthermore, the developed approach could be adapted to any other computer vision tasks by exploiting a different source like an automatically reconstructed point cloud as a weak supervision signal.

REFERENCES

[1] Semantic drone dataset. http://dronedataset.icg.tugraz.at, Accessed: February 15, 2022.

[2] Agilex - limo. https://global.agilex.ai/products/limo, Accessed: March 26, 2022.

[3] Boston dynamics - spot. https://www.bostondynamics.com/solutions/construction, Accessed: March 26, 2022.

[4] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels. page 15, 2010.

[5] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, Nov 2012.

[6] W. Ali, S. Abdelkarim, M. Zidan, M. Zahran, and A. El Sallab. Yolo3d: End-to-end real-time 3d oriented object bounding box detection from lidar point cloud. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[7] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2294–2301, 2009.

[8] D. Avola, G. L. Foresti, N. Martinel, C. Micheloni, D. Pannone, and C. Piciarelli. Aerial video surveillance system for small-scale UAV environment monitoring. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, 2017.

[9] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21(6):34–47, Nov 2001.

[10] R. T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, 1997.

[11] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[12] M. D. Bah, A. Hafiane, and R. Canals. Deep learning with unsupervised data labeling for weed detection in line crops in UAV images. *Remote sensing*, 10(11):1690, 2018.

[13] A. Bódis-Szomorú, H. Riemenschneider, and L. Van Gool. Fast, approximate piecewise-planar modeling based on sparse structure-from-motion and superpixels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 469–476, 2014.

[14] S. Bosch, S. Lacroix, and F. Caballero. Autonomous detection of safe landing areas for an UAV from monocular images. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5522–5527. IEEE, 2006.

[15] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239, 2001.

[16] J. Y. Chen. UAV-guided navigation for ground robot tele-operation in a military reconnaissance environment. *Ergonomics*, 53(8):940–950, 2010. PMID: 20658388.

[17] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[18] P.-Y. Chen, A. H. Liu, Y.-C. Liu, and Y.-C. F. Wang. Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2624–2632, 2019.

[19] W. Chen, Z. Fu, D. Yang, and J. Deng. Single-image depth perception in the wild. *Advances in neural information processing systems*, 29, 2016.

[20] A. Cohen, J. L. Schönberger, P. Speciale, T. Sattler, J.-M. Frahm, and M. Pollefeys. Indoor-outdoor 3d reconstruction alignment. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, pages 285–300, Cham, 2016. Springer International Publishing.

[21] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[22] C. Couprie, C. Farabet, L. Najman, and Y. LeCun. Indoor semantic segmentation using depth information. *arXiv preprint arXiv:1301.3572*, 2013.

[23] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[24] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[25] J. Engel, T. Schöps, and D. Cremers. Lsd-slam: Large-scale direct monocular slam. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 834–849, Cham, 2014. Springer International Publishing.

[26] 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*, 24(6):381–395, June 1981.

[27] M. A. Fischler and R. C. Bolles. Readings in computer vision: Issues, problems, principles, and paradigms. chapter Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, pages 726–740. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.

[28] B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *2009 IEEE 12th international conference on computer vision*, pages 670–677. IEEE, 2009.

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

[30] D. Gallup, J.-M. Frahm, and M. Pollefeys. Piecewise planar and non-planar stereo for urban scene reconstruction. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1418–1425. IEEE, 2010.

[31] R. Garg, S. Yang, and S. Scherer. Monocular and stereo cues for landing zone evaluation for micro UAVs. *arXiv e-prints*, pages arXiv–1812, 2018.

[32] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[33] T. L. Giang, K. B. Dang, Q. T. Le, V. G. Nguyen, S. S. Tong, and V.-M. Pham. U-net convolutional networks for mining land cover classification based on high-resolution UAV imagery. *Ieee Access*, 8:186257–186273, 2020.

[34] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 270–279, 2017.

[35] B. Gong, Z. Zhu, C. Yan, Z. Shi, and F. Xu. Planefusion: Real-time indoor scene reconstruction with planar prior. *IEEE Transactions on Visualization and Computer Graphics*, 2021.

[36] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller. Multi-class segmentation with relative location prior. *International Journal of Computer Vision*, 80(3):300–316, 2008.

[37] E. Grilli, F. Menna, and F. Remondino. A review of point clouds segmentation and classification algorithms. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42:339, 2017.

[38] S. Gupta, P. Arbeláez, R. Girshick, and J. Malik. Indoor scene understanding with rgb-d images: Bottom-up segmentation, object detection and semantic segmentation. *International Journal of Computer Vision*, 112(2):133–149, 2015.

[39] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[40] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[41] D. Holz, S. Holzer, R. B. Rusu, and S. Behnke. Real-time plane segmentation using rgb-d cameras. In T. Röfer, N. M. Mayer, J. Savage, and U. Saranlı, editors, *RoboCup 2011: Robot Soccer World Cup XV*, pages 306–317, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[42] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[43] J. Illingworth and J. Kittler. A survey of the hough transform. *Computer Vision, Graphics, and Image Processing*, 44(1):87 – 116, 1988.

[44] J. Jain, A. Singh, N. Orlov, Z. Huang, J. Li, S. Walton, and H. Shi. Semask: Semantically masked transformers for semantic segmentation. *arXiv e-prints*, pages arXiv–2112, 2021.

[45] R. Jain, R. Kasturi, B. G. Schunck, et al. *Machine vision*, volume 5. McGraw-hill New York, 1995.

[46] J. J. Koenderink and A. J. van Doorn. Affine structure from motion. *J. Opt. Soc. Am. A*, 8(2):377–385, Feb 1991.

[47] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[48] J.-H. Lee and C.-S. Kim. Multi-loss rebalancing algorithm for monocular depth estimation. In *European Conference on Computer Vision*, pages 785–801. Springer, 2020.

[49] L. Li, F. Yang, H. Zhu, D. Li, Y. Li, and L. Tang. An improved ransac for 3d point cloud plane segmentation based on normal distribution transformation cells. *Remote Sensing*, 9(5):433, May 2017.

[50] M. Li and T. Hu. Deep learning enabled localization for UAV autolanding. *Chinese Journal of Aeronautics*, 34(5):585–600, 2021.

[51] X. Li and L. Yang. Design and implementation of uav intelligent aerial photography system. In *2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics*, volume 2, pages 200–203, 2012.

[52] Y. Li, N. Snavely, and D. P. Huttenlocher. Location recognition using prioritized feature matching. In *European conference on computer vision*, pages 791–804. Springer, 2010.

[53] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. *ACM Transactions on Graphics (ToG)*, 23(3):303–308, 2004.

[54] Z. Li and N. Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2041–2050, 2018.

[55] C. Liu, K. Kim, J. Gu, Y. Furukawa, and J. Kautz. Planercnn: 3d plane detection and reconstruction from a single image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[56] C. Liu, J. Yang, D. Ceylan, E. Yumer, and Y. Furukawa. Planenet: Piece-wise planar reconstruction from a single rgb image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[57] G. Liu and J. Duan. Rgb-d image segmentation using superpixel and multi-feature fusion graph theory. *Signal, Image and Video Processing*, pages 1–9, 2020.

[58] J. Liu, P. Ji, N. Bansal, C. Cai, Q. Yan, X. Huang, and Y. Xu. Planemvs: 3d plane reconstruction from multi-view stereo. *arXiv preprint arXiv:2203.12082*, 2022.

[59] Y. Lyu, G. Vosselman, G. Xia, A. Yilmaz, and M. Y. Yang. The uavid dataset for video semantic segmentation. *arXiv preprint arXiv:1810.10438*, 1, 2018.

[60] Y. Lyu, G. Vosselman, G.-S. Xia, A. Yilmaz, and M. Y. Yang. Uavid: A semantic segmentation dataset for UAV imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 165:108 – 119, 2020.

[61] M. Meilundefined. Comparing clusterings: An axiomatic view. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 577–584, New York, NY, USA, 2005. Association for Computing Machinery.

[62] S. M. H. Miangoleh, S. Dille, L. Mai, S. Paris, and Y. Aksoy. Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9685–9694, 2021.

[63] T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3):231 – 268, 2001.

[64] L. Mou and X. X. Zhu. Vehicle instance segmentation from aerial image and video using a multitask learning residual fully convolutional network. *IEEE Transactions on Geoscience and Remote Sensing*, 56(11):6699–6711, 2018.

[65] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, Oct 2015.

[66] B. Oehler, J. Stueckler, J. Welle, D. Schulz, and S. Behnke. Efficient multi-resolution plane segmentation of 3d point clouds. In S. Jeschke, H. Liu, and D. Schilberg, editors, *Intelligent Robotics and Applications*, pages 145–156, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[67] N. Pears and B. Liang. Ground plane segmentation for mobile robot visual navigation. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, volume 3, pages 1513–1518 vol.3, 2001.

[68] D. Popescu and L. Ichim. Aerial image segmentation by use of textural features. In *2016 20th international conference on system theory, control and computing (ICSTCC)*, pages 721–726. IEEE, 2016.

[69] M. Radovic, O. Adarkwa, and Q. Wang. Object recognition in aerial images using convolutional neural networks. *Journal of Imaging*, 3(2):21, 2017.

[70] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

[71] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[72] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.

[73] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[74] L. O. Rojas-Perez, R. Munguia-Silva, and J. Martinez-Carranza. Real-time landing zone detection for UAVs using single aerial images. In *10th International Micro Air Vehicle Competition and Conference, Melbourne, Australia*, pages 243–248, 2018.

[75] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[76] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3234–3243, 2016.

[77] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet

Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[78] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[79] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[80] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 519–528, June 2006.

[81] M. Shah Alam and J. Oluoch. A survey of safe landing zone detection techniques for autonomous unmanned aerial vehicles (uavs). *Expert Systems with Applications*, 179:115091, 2021.

[82] M. Shaha and M. Pawar. Transfer learning for image classification. In *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 656–660, 2018.

[83] Y. Shi, P. Long, K. Xu, H. Huang, and Y. Xiong. Data-driven contextual modeling for 3d scene understanding. *ComputersGraphics*, 55:55–67, 2016.

[84] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE Transactions on Medical Imaging*, 35(5):1285–1298, 2016.

[85] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[86] S. N. Sinha, D. Steedly, and R. Szeliski. Piecewise planar stereo for image-based rendering. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1881–1888, 2009.

[87] S. Song and J. Xiao. Sliding shapes for 3d object detection in depth images. In *European conference on computer vision*, pages 634–651. Springer, 2014.

[88] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[89] G. Tian, J. Liu, and W. Yang. A dual neural network for object detection in UAV images. *Neurocomputing*, 443:292–301, 2021.

[90] F. E. Uzyıldırım and M. Özuysal. Improving outdoor plane estimation without manual supervision. *Signal, Image and Video Processing*, 16(2):1–9, 2022.

[91] J. Ventura and T. Hollerer. Online environment model estimation for augmented reality. In *2009 8th IEEE International Symposium on Mixed and Augmented Reality*, pages 103–106, 2009.

[92] L. Wang, J. Zhang, O. Wang, Z. Lin, and H. Lu. Sdc-depth: Semantic divide-and-conquer network for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 541–550, 2020.

[93] K. Xian, J. Zhang, O. Wang, L. Mai, Z. Lin, and Z. Cao. Structure-guided ranking loss for single image depth prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 611–620, 2020.

[94] J. Xiao, J. Zhang, B. Adler, H. Zhang, and J. Zhang. Three-dimensional point cloud plane segmentation in both structured and unstructured environments. *Robotics and Autonomous Systems*, 61(12):1641 – 1652, 2013.

[95] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[96] T. Xue, H. Luo, D. Cheng, Z. Yuan, and X. Yang. Real-time monocular dense mapping for augmented reality. In *Proceedings of the 25th ACM International Conference on Multimedia*, MM '17, page 510–518, New York, NY, USA, 2017. Association for Computing Machinery.

[97] F. Yang and Z. Zhou. Recovering 3d planes from a single image via convolutional neural networks. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, editors, *Computer Vision – ECCV 2018*, pages 87–103, Cham, 2018. Springer International Publishing.

[98] T. Yang, P. Li, H. Zhang, J. Li, and Z. Li. Monocular vision slam-based UAV autonomous landing in emergencies and unknown environments. *Electronics*, 7(5):73, 2018.

[99] Z. Yang, L. E. Li, and Q. Huang. Strumononet: Structure-aware monocular 3d prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7413–7422, 2021.

[100] W. Yin, J. Zhang, O. Wang, S. Niklaus, L. Mai, S. Chen, and C. Shen. Learning to recover 3d scene shape from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 204–213, 2021.

[101] Z. Yu, J. Zheng, D. Lian, Z. Zhou, and S. Gao. Single-image piece-wise planar 3d reconstruction via associative embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1029–1037, 2019.

[102] S. Zagoruyko and N. Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016.

[103] Z. Zeng, M. Wu, W. Zeng, and C.-W. Fu. Deep recognition of vanishing-point-constrained building planes in urban street views. *IEEE Transactions on Image Processing*, 29:5912–5923, 2020.

[104] Z. Zhang, Z. Cui, C. Xu, Y. Yan, N. Sebe, and J. Yang. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4106–4115, 2019.

VITA

Furkan Eren Uzyıldırım

Academic Experience

2015–2019 **Research/Teaching Assistant**
Department of Computer Engineering, Izmir Institute of Technology

2019–2022 **Head Research/Teaching Assistant**
Department of Computer Engineering, Izmir Institute of Technology

Education

2014–2016 **MSc in Computer Engineering**, Izmir Institute of Technology, Turkey.
Title: Keypoint Matching Based on Descriptor Statistics
Advisor: Assoc. Prof. Dr. Mustafa Özuysal

2009–2014 **BSc in Computer Engineering**, Izmir Institute of Technology, Turkey.

Publications

2022 Furkan Eren Uzyıldırım and Mustafa Özuysal. Improving outdoor plane estimation without manual supervision. *Signal, Image and Video Processing*, 16(2):1–9, 2022

2017 Ali Köksal, Furkan Eren Uzyıldırım, and Mustafa Özuysal. Three dimensional scenes and object recognition. In *2017 25th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4. IEEE, 2017

2017 Ali Köksal, Furkan Eren Uzyıldırım, and Mustafa Özuysal. Üç boyutlu sahneler ve nesne tanıma için gürbüz anahtar nokta eşleştirilmesi. In *25th Signal Processing and Communications Applications Conference, SIU 2017*. Institute of Electrical and Electronics Engineers Inc., 2017

2016 Furkan Eren Uzyıldırım and Mustafa Özuysal. Instance detection by keypoint matching beyond the nearest neighbor. *Signal, Image and Video Processing*, 10(8):1527–1534, 2016

2016 Furkan Eren Uzyıldırım. Keypoint matching based on descriptor statistics. Master's thesis, Izmir Institute of Technology, 2016

2015 Furkan Eren Uzyıldırım, Ali Köksal, and Mustafa Özuysal. A detailed analysis of mser and fast repeatibility. In *2015 23nd Signal Processing and Communications Applications Conference (SIU)*, pages 2098–2101. IEEE, 2015