ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DOTTORATO DI RICERCA IN

COMPUTER SCIENCE AND ENGINEERING

Ciclo 37

Settore Concorsuale: 09/H1 - SISTEMI DI ELABORAZIONE DELLE INFORMAZIONI

Settore Scientifico Disciplinare: ING-INF/05 - SISTEMI DI ELABORAZIONE DELLE INFORMAZIONI

MONOCULAR DEPTH ESTIMATION BASED ON GROUND GEOMETRY

Presentata da: Huan Li

Coordinatore Dottorato

Ilaria Bartolini

Supervisore

Stefano Mattoccia

Co-supervisore

Matteo Poggi

Esame finale anno 2025

Abstract

Monocular depth estimation is typically regarded as an ill-posed problem due to the challenge of scale ambiguity. Unlike other depth estimation approaches, such as stereo depth estimation and LiDAR measurement, monocular depth estimation relies solely on the information presented in a single image, leading to lack of multi-views consistency cues. Nevertheless, monocular depth estimation does not depend on costly sensor equipment or complex calibration processes, making it easily deployable across a variety of scenarios, including autonomous driving, robotics, augmented reality, and scene understanding.

In recent years, significant advancements in deep learning have led researchers to explore end-to-end training methods for depth prediction. These methods typically utilize extensive depth annotations in conjunction with state-of-the-art neural networks, demonstrating strong generalization capabilities across diverse scenes. However, high-quality labeled datasets are time-consuming and expensive to create, resulting in increased training complexity. To alleviate this issue, some researchers have proposed self-supervised monocular depth estimation that only leverages natural video sequences, thereby reducing the reliance on large annotated depth labels. Despite these advancements, self-supervised depth estimation methods typically assume that all objects in the training scenes remain static. This assumption can result in numerous failure cases when estimating the depth of dynamic objects.

Aimed to above limitations, we propose the integration of ground geometry information into depth estimation processes. In static scenes, the ground normal vector predicted by people probes offers accurate scale information about the 3D scenes, which not only allows to accurately align predicted 3D scene with the real-world environment and also provide a reliable scale factor to convert all relative depths to absolute distances. Our proposed method enables to achieve metric depth estimation for any static scenes. Furthermore, in dynamic scenes, we assume that the depth of all moving objects is consistent with the depth of their ground contact points. Based on this geometric prior, we develop a ground propagation module for self-supervised depth estimation which iteratively propagates ground features to dynamic objects in the latent space of decoder, thereby facilitating depth calibration of dynamic objects. The final experimental results demonstrate that our method can effectively improve the depth estimation of moving targets and achieve superior generalization performance.

In summary, incorporating ground geometry information can significantly enhance the accuracy of monocular depth estimation in both static and dynamic environments, making monocular depth estimation a more dependable solution for future various applications.

Keywords: Monocular depth estimation; Ground geometry; 3D vision

Acknowledgments

It’s a great honor to take a Ph.D. position in Bologna, Italy. Looking back over the past three years, I would like to express my deepest gratitude to my supervisors, Prof. Stefano Mattoccia and Matteo Poggi, whose guidance and unwavering support have been instrumental in my doctoral journey. Their combined expertise and constructive feedback have been invaluable, and I am incredibly fortunate to have had the opportunity to work under their guidance. I would also like to extend my sincere thanks to Fabio Tosi, Zhang Youmin, Gong Ziren and all members of CVLAB, It’s a great time to spend with all of you together. Finally, I owe my deepest thanks to my parents, Yuan tang and Jian hua, for their unconditional love, encouragement, and supports. Their belief in me has been my greatest source of strength and motivation. This achievement would not have been possible without them.

Contents

Chapter 1

Introduction

Monocular depth estimation has become an active research area in computer vision, with significant advancements achieved in recent years. This task focuses on predicting the distances of objects within a 3D scene using only a single 2D image as input. Fig. 1.1 displays depth maps across various scenes, where darker hues represent farther distances while lighter hues indicate closer ones. As shown, depth maps reflect the relative distances of objects to the camera and spatial structures of scenes, thereby can be applied in a wide range of real-world applications such as autonomous driving, social distance measurement, and augmented reality.

Additionally, as the inverse of camera projection, depth estimation enables to lift 2D images to 3D scenes, making it a foundational component of many computer vision tasks, including 3D object detection, scene flow prediction and 3D segmentation. Compared to other depth estimation methods, such as LiDAR [134] or stereo depth estimation [131], monocular depth estimation does not need costly sensors or complex calibration procedures, making it adoptable to be deployed in any settings, from indoor environments to dynamic outdoor scenes.

To provide a comprehensive exploration of monocular depth estimation, the following sections are organized to discuss core geometric concepts, widely used datasets, evaluation metrics, existing methodologies, and current challenges, presenting a structured overview and analysis of this field. Building on these insights, we propose our improved approaches to monocular depth estimation, which are introduced in Chapters 3 and 4 in detail.

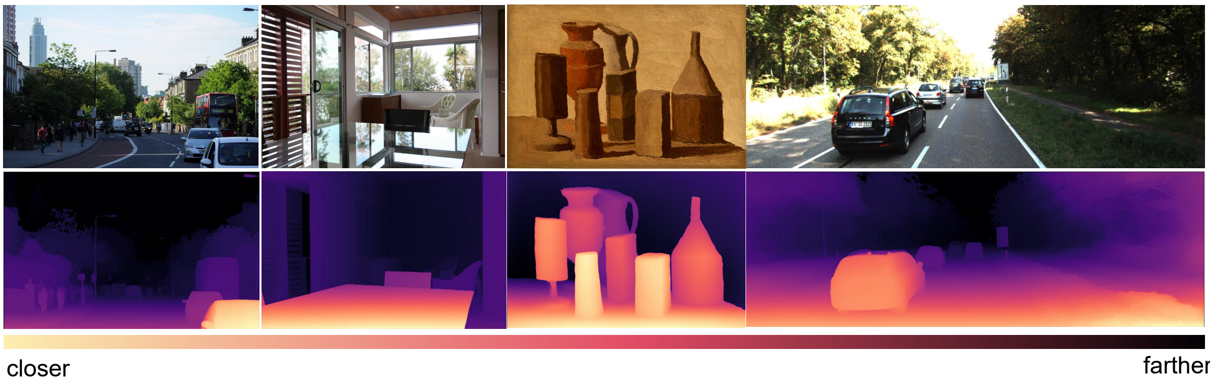


Figure 1.1: The visualization of depth maps across different scenes. The darker hues indicate farther distances while lighter hues means closer distances.

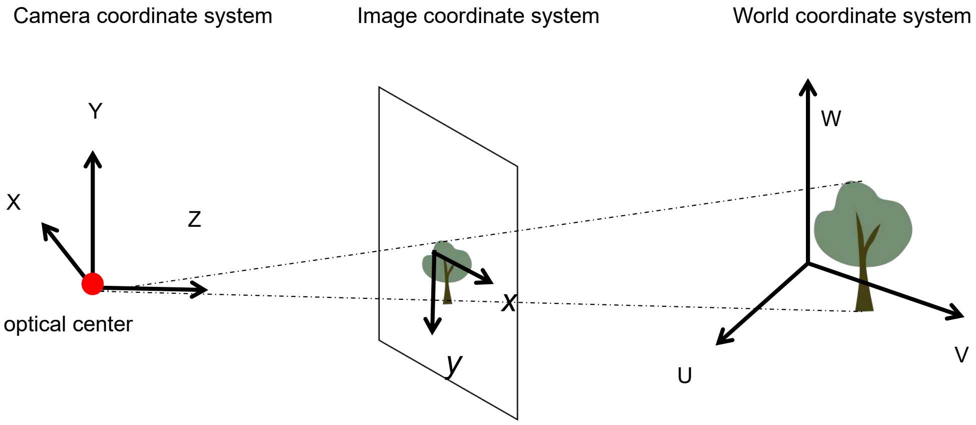


Figure 1.2: The illustration of camera coordinate, image coordinate and world coordinate systems.

1.1 Geometric Concepts

1.1.1 Coordinate Systems

In the concept of imaging geometry, there are three different coordinate systems.

World coordinate system is used to describe the absolute position and orientation of objects in 3D space, with three orthogonal axes labeled as (U, V, W). The origin and orientation of these axes can be set arbitrarily to best suit the scene.

Camera coordinate system is centered at the optical center of the camera, with the camera's lens pointing along the the optical axis Z-axis. The X- and Y-axis define the horizontal and vertical planes of the camera's field of view. The objects in this system are defined in terms of their distance and direction relative to the camera's position.

Image coordinate system represents points in a 2D image plane. Typically, the origin point is defined at the top-left corner of the image. The x-axis increases to the right, and the y-axis increases downward. The differences and relations between these three coordinates are shown in Fig. 1.2.

1.1.2 Camera intrinsics and extrinsics

The camera intrinsic and extrinsic parameters determine how a camera captures a 3D scene and projects it onto a 2D image.

Camera intrinsics includes parameters related to the camera's internal characteristics, such as focal length, distortion, principal point and scales, which are independent of the camera's position and orientation. The focal length f defines the distance between the camera's lens and the optical center, affecting magnification and field of view of the optical system. The camera distortion refers to the optical imperfections introduced by the camera lens, which cause the captured image to deviate from the real-world geometry. There are two main types of camera distortion:

- Radial distortion: it occurs when light rays bend as they pass through the lens, causing straight lines to appear curved in the image. The areas further away from the optical center would suffer more severe distortion. Given the radial distortion coefficients k_1, k_2, k_3, the distorted image coordinates (x', y') and the correct coordinates (x, y) have the following mathematical relationship, where $r = x'^2 + y'^2$:

$$\begin{pmatrix} x \\ y \end{pmatrix} = (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \begin{pmatrix} x' \\ y' \end{pmatrix} \tag{1.1}$$

- Tangential distortion: it occurs when the lens is not perfectly aligned with the image sensor. This causes the image to appear tilted or shifted, leading to slight inaccuracies in how objects are positioned within the frame. Similarly, Eq.1.2 gives the conversion formula between distorted coordinates and correct coordinates when tangential distortion coefficients p_1, p_2 are known.

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 2p_1 x'y' + p_2(r^2 + 2x'^2) \\ 2p_2 x'y' + p_1(r^2 + 2y'^2) \end{pmatrix} \tag{1.2}$$

The principal point is the offset of the image center from the up-left corner, typically represented by pixel coordinates (c_x,c_y), as depicted in Fig.1.3

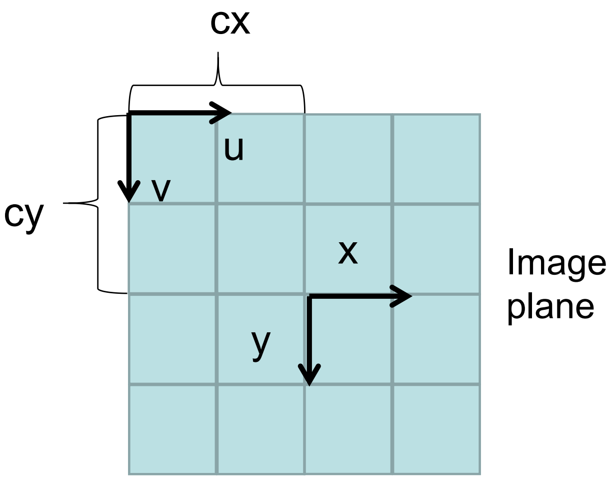


Figure 1.3: The principal point (c_x, c_y)

The camera scales determine scale factor to convert distances in the image plane from pixels to real-world units. The scale factors in the x-axis and y-axis are denoted as s_x and s_y. Typically, the camera intrinsics can be expressed as a matrix K as follows, where f_x and f_y are equal to f/s_x, f/s_y respectively.

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{1.3}$$

Camera extrinsics are parameters to depict position and pose of the camera relative to the world coordinate. It involves rotation matrix R and translation vector T which will be changed when the camera moves or rotate in the 3D space. Given the camera extrinsic parameters, the transformation between the world and camera coordinate systems can be achieved through Eq. 1.4, where P_c is the point in the camera coordinate system and P_w is the corresponding location in the world coordinate system.

$$P_c = RP_w + T \tag{1.4}$$

1.1.3 Coordinate transformations

The coordinate transformations is aimed to build a mathematical model to describe how the points in the 3D world get projected into 2D pixel coordinates. As formulated in Eq. 1.4,

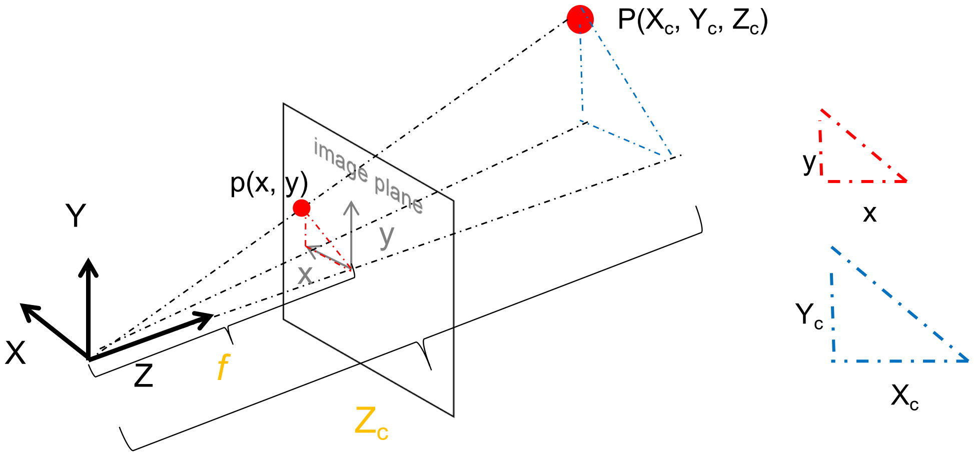


Figure 1.4: The perspective projection. The blue triangle is constructed by the projections of P on the X and Y axis under the camera coordinate system, while the projections of p on the x-axis and y-axis form red triangle. The two triangles are similar with proportion of Z_c/f, where Z_c is the depth of point P from the camera and f is the camera focus.

the location of the objects in the 3D scenes can be transformed into camera coordinate system through camera extrinsics. The following explains how to convert camera coordinates into pixel coordinates.

Perspective projection is a process to project points in the camera coordinate system into 2D image planes. As illustrated in Fig. 1.4, a point P with coordinates $P(X_c, Y_c, Z_c)$ is projected onto p on the image plane. The projections of P on the X-axis and Y-axis form a right triangle, where the two bases represent the coordinates X_c and Y_c respectively. Similarly, the projections of point p on the image plane along the x-axis and y-axis form another right triangle, as depicted by the red triangle in Fig. 1.4. Since these two triangles are similar, a proportional relationship can be established through the Eq. 1.5, where f is the camera focus length.

$$\frac{x}{X_c} = \frac{y}{Y_c} = \frac{f}{Z_c} \tag{1.5}$$

Pixel coordinate transformation works for translating image coordinates into pixel locations on the image. The Fig. 1.3 has demonstrated that there is an offset (c_x, c_y) between image coordinates and pixel coordinates, by incorporating Eq. 1.5, the pixel coordinates (u, v) can be determined using Eq. 1.6. In addition, the Eq. 1.6 can be reformulated into matrix multiplication, as shown in Eq. 1.7, where P denotes the camera extrinsics matrix.

$$u = \frac{X_c \bullet f}{Z_c} + c_x, v = \frac{Y_c \bullet f}{Z_c} + c_y \tag{1.6}$$

$$Z_c \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} = KPP_w \tag{1.7}$$

1.1.4 Camera calibration

The camera calibration is targeted to estimate all intrinsic and extrinsic parameters introduced in the Sec. 1.1.2. The Zhang's Camera Calibration [223], proposed in 1998, is a widely adopted calibration method, providing a practical solution for this task. The Zhang's camera calibration employs a checkerboard board, as shown in Fig. 1.5, and captures the board from different views. By fixing the world coordinate system on the checkerboard, the world coordinates of each corner can be determined, as the distances between the black-white squares are known and consistent. It allows the extraction of multiple pairs of image coordinates and corresponding world coordinates, which are then used to solve Eq. 1.7 for camera calibration.

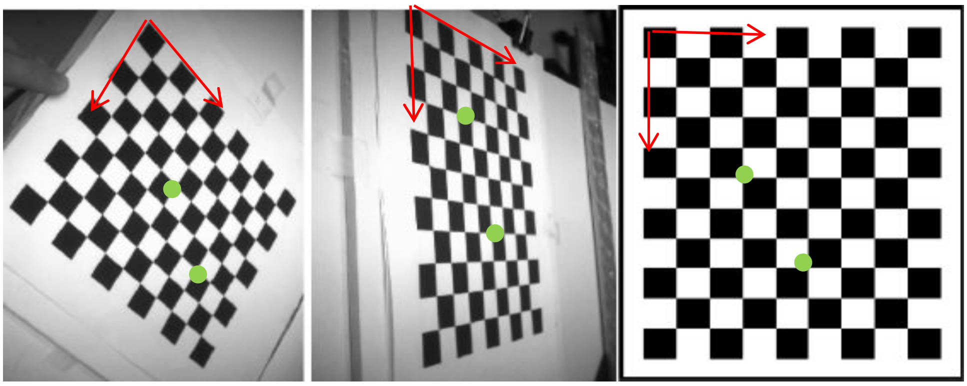


Figure 1.5: The checkerboard for camera calibration. The white and black squares are the same size. When fixing the world coordinate system on the checkerboard, the spatial coordinates of all corner points will be determined.

1.2 Datasets

The depth datasets are crucial for training and evaluating models of monocular depth estimation. These datasets typically consist of RGB images and corresponding ground truth depth maps which are generated using depth sensors like LiDAR, stereo cameras or simulation software. The following lists introduce these datasets in detail.

KITTI Dataset [51]: The KITTI dataset is a widely used dataset in computer vision and autonomous driving, developed by the Karlsruhe Institute of Technology (KIT) in collaboration with Audi. It serves as a benchmark for evaluating perception algorithms used in self-driving cars and covers a wide range of tasks, e.g. 3D Object detection, optical flow, semantic segmentation, SLAM and depth estimation. The KITTI dataset contains approximately 6 hours of driving data, with the following documents:

- The raw and synchronized/rectified binocular grayscale image sequences with the size of 1242×375.

- The raw and synchronized/rectified binocular RGB image sequences with the size of 1240×370

- The 3D Velodyne point clouds collected from LiDAR sensors, with 100k points per frame, which provide accurate depth information.
- The 3D GPS/IMU data, used to track the vehicle's position and orientation in real-time.
- The calibration documents, recording intrinsic and extrinsic parameters of camera.

NYU Depth v2 [127]: The NYU-Depth V2 dataset was released from New York University in 2012. It consists of video sequences of various indoor scenes recorded by the RGB and depth cameras of Microsoft Kinect. It features:

- 1449 densely labeled pairs of aligned RGB and depth images, with size of 640×480.
- 464 new scenes taken from 3 cities.
- For the indoor environments, it captures a wide variety of objects, furniture, and cluttered scenes and provides an excellent testbed for algorithms dealing with occlusions, lighting variations, and complex spatial layouts.

Make3D [145, 146]: The Make3D is another well-known 3D vision dataset. It was developed by researchers at Stanford University and provides ground truth depth maps along with corresponding RGB images.

- It consists of 400 training images and 134 test images with the size of 2272×1704, each paired with depth maps.
- The depth data in the Make3D dataset is generated using the Lidar scanner with the size of 55×305.

Cityscapes [34]: The Cityscapes dataset is a large-scale dataset primarily designed for urban scene understanding, with a diverse set of stereo video sequences recorded in street scenes from 50 different cities.The details of this dataset are as follows:

- The Cityscapes provides pixel-wise semantic labels and instance-level annotations. It contains 5,000 finely annotated images and 20,000 coarsely annotated images, with the size of 2048×1024.
- The dataset provides disparity maps(i.e the reverse of depth) precomputed through SGM algorithm [63].
- It also gives intrinsic and extrinsic camera parameters for train, validation, and test sets.

Virtual KITTI [22]: In 2015, researchers at the Naver Labs Europe used the Unity game engine [153] to simulate real-world videos from the raw KITTI autonomous driving benchmark suite, leading to the creation of the Virtual KITTI dataset. This synthetic dataset was one of the pioneering efforts in generating artificial training data specifically for autonomous driving applications. Virtual KITTI reproduces five driving scenes under various weather and lighting conditions, including fog, rain, and sunset. It provides comprehensive training data for several key computer vision tasks, such as semantic segmentation, optical flow, depth estimation, and scene flow analysis. In detail,

- The Virtual KITTI dataset contains 21,000 RGB images with the size of 1242×375.

- The depth images are dense and aligned with all the RGB images. They are encoded as grayscale 16bit PNG files. To normalize depth maps into 0-1, the loaded depth map will be divided by 65535.

MegaDepth [96]: The MegaDepth comprises 1 million images sourced from the internet, making it one of the largest datasets available for depth estimation, which cover a variety of scenes, from urban landscapes to natural environments and indoor settings. The key features are as follows:

- It has totally 130,000 image-depth pairs with the size of .

- To achieve depth labels, the MegaDepth employed COLMAP [148, 150], a state-of-art multi-view stereo method to generate depth maps. To eliminate outliers in the initial depth maps, researchers adopted some refinement techniques, e.g. segmentation and median filter to finally construct dense and plausible depth labels.

DrivingStereo [192]: The DrivingStereo dataset is a large-scale stereo dataset which is developed for outdoor autonomous driving settings.

- The DrivingStereo dataset contains 182,188 RGB images, where the training set has 14,437 stereo pairs and the testing set has 7,751 pairs.

- The image size is 881×400

- High-quality disparity labels are produced by a model-guided filtering strategy from multi-frame LiDAR [134] points.

These datasets enable the training of deep learning models for depth estimation, allowing the models to learn the relationship between RGB image features and the corresponding depth information.

1.3 Evaluation Metrics

To evaluate prediction accuracy of proposed methods in the task of monocular depth estimation, it is essential to employ appropriate evaluation metrics. Different other vision tasks, depth estimation typically encounters the issue of scale inconsistency that the predicted depth has different scales with actual depth. To ensure a fair comparison between different methods, the predicted depth values must be normalized to a uniform scale before calculating their numerical differences with the ground truth. The commonly accepted evaluation metrics are listed in the Tab. 1.1, where d_i represents predicted depth value at pixel i, $\hat{d_i}$ denotes ground truth depth, N is the number of pixels on the image.

These metrics effectively quantify depth evaluation errors, but each emphasizes different depth ranges. For instance, The MAE metric treats all depth ranges equally, making them more sensitive to errors in distant depth values compared to nearby ones. In contrast, AbsRel mitigates the impact of large depth errors by normalizing them relative to the ground truth depth.

Table 1.1: The evaluation metrics for the depth estimation: In the formula column, the d_i represents predicted depth value at pixel i, $\hat{d_i}$ denotes ground truth depth, N is the number of pixels on the image.The $\downarrow$ in the Better value column means the smaller the value better for this metric while $\uparrow$ denotes the bigger value is better.

Metrics	Name	Formula	Better value
MAE	mean absolute error	$\frac{1}{N}\sum_{i=1}^{N}\lvert d_i - \hat{d_i}\rvert$	$\downarrow$
MSE	mean square error	$\frac{1}{N}\sum_{i=1}^{N}(d_i - \hat{d_i})^2$	$\downarrow$
RMSE	root mean square error	$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(d_i - \hat{d_i})^2}$	$\downarrow$
AbsRel	absolute relative error	$\frac{1}{N}\sum_{i=1}^{N}\lvert\frac{d_i-\hat{d_i}}{\hat{d_i}}\rvert$	$\downarrow$
SqRel	square relative error	$\frac{1}{N}\sum_{i=1}^{N}(\frac{d_i-\hat{d_i}}{\hat{d_i}})^2$	$\downarrow$
LogRMSE	root mean square logarithmic error	$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(log(d_i) - log(\hat{d_i}))^2}$	$\downarrow$
δ_i	threshold accuracy	$\frac{1}{N}\sum_{i=1}^{N}max(\frac{d_i}{\hat{d_i}}, \frac{\hat{d_i}}{d_i}) < thre, thre = 1.25, 1.25^2, 1.25^3$	$\uparrow$

Similarly, the RMSE reduces the influence of high-depth discrepancies through performing log operation on the depth value. By evaluating depth through the above metrics simultaneously, it is possible to acquire a more comprehensive evaluation result, thereby enabling a clearer judgment of a model's strengths and weaknesses.

1.4 Sensoring System and Multi-Views for Depth Estimation

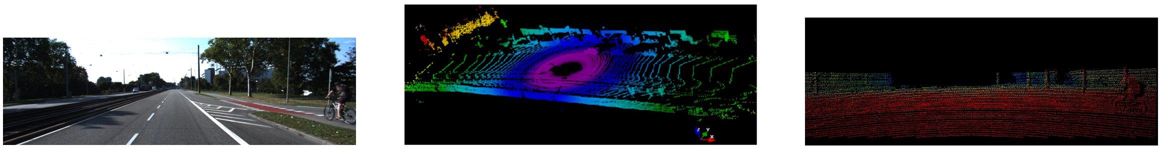
Depth estimation methods can be broadly categorized into stereo vision, multi-view, monocular and sensor-based approaches like LiDAR [134]. This section will presents a comprehensive introduction of LiDAR distance measurement, stereo depth estimation, and multi-views depth estimation subsequently, highlighting their principles, applications and limitations.

1.4.1 LiDAR distance measurement

LiDAR (Light Detection and Ranging) is a remote sensing technology that uses laser pulses to measure distances between the sensor and objects in the environment. It is widely used for creating high-resolution 3D maps and is especially popular in autonomous vehicles, robotics, and geospatial applications due to its high accuracy and ability to capture detailed depth information in real-time.

The LiDAR system is composed of several components: a laser emitter that sends out laser pulses, a receiver that detects the reflected signals, a rotating or scanning mechanism that allows for 360-degree coverage or targeted scanning and a GPS and inertial measurement unit (IMU) for accurately tracking the sensor's position and orientation. It operates by emitting laser beams and measuring the time it takes for the pulses to reflect off objects and return to the sensor. Since the speed of light is constant, the sensor can calculate the distance between the LiDAR and the object based on the time delay, also known as the time of flight (ToF). All of these components work together to acquire accurate depth information, enabling the system to generate detailed 3D maps of the environment.

LiDAR data is typically stored in specialized file formats that preserve the detailed 3D point cloud data. As the diversity of LiDAR systems(e.g Terrestrial LiDAR (TLS), Aerial LiDAR



(a) RGB image

(b) 3D point cloud

(c) Projected point cloud

Figure 1.6: The visualization of LiDAR point cloud: (a) is a 2D RGB image; (b) shows the visualization of the point cloud in 3D space, where purple indicates points closer to the camera, while green represents points farther from the camera; (c) is the result that 3D point cloud is projected onto the 2D image where red indicates closer points while blue denotes farther points.

(ALS), and Bathymetric LiDAR), the document formats storing point cloud data are varying, each having the specific characteristics and requirements of the respective system. These different formats necessitate distinct methods for reading, processing, and interpreting the data. The most commonly used formats involve LAS/LAZ, PLY, PTS and BIN where typically preserving 3D coordinates, intensities, timestamps and RGB information of point clouds. For the KITTI dataset [51], the LiDAR data is saved in a BIN document. After reading the data document and visualizing it, the Fig. 1.6 displays the 3D point cloud of the image in the KITTI dataset.

As a high-accuracy depth sensor, LiDAR excels in capturing detailed 3D spatial data over large areas and being independent from ambient light, allowing it to work effectively in low-light or nighttime conditions. Despite these advantages, the high cost of equipment and maintenance hinders its more widespread applications. Additionally, LiDAR faces challenges in capturing data from objects with high reflectivity and in adverse weather conditions, such as fog, rain, and snow. Furthermore, as shown in the Fig. 1.6, the accuracy and density of the point cloud degrade with increasing distance from the LiDAR, making it difficult to detect fine details at long ranges. Therefore, these limitations of LiDAR motivate the development of machine learning-based depth estimation that tries to incorporate geometric principles into the topic of image processing and machine learning.

1.4.2 Stereo depth estimation

Stereo depth estimation is another crucial method to estimate the depth of objects in a scene. By analyzing two images captured by a paired cameras which positioned at slightly different viewpoints, this technique leverages the disparity between corresponding points in the image pair to compute depth information. Unlike the LiDAR relying on the expensive sensor equipment, stereo depth estimation only employs a calibrated stereo camera, which significantly declining the cost of depth measurement. As shown in Fig. 1.7, the point P in the 3D world can be projected onto 2D images with different viewpoints. According to the coordinate transformation principle introduced in the section 1.1.3, when intrinsic and extrinsic parameters of cameras are known, the transformation relationship between pixel coordinate p on the left camera plane and corresponding coordinate p' on the right can be established through Eq. 1.8, where K_r and K_l represent intrinsic parameters of left camera and right camera respectively, $P_{l->r}$ denotes pose transformation between two cameras which can be easily computed by camera extrinsic parameters, Z is depth value of point P to the left camera.

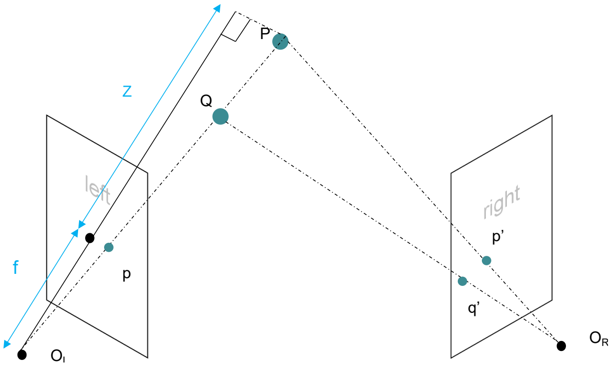


Figure 1.7: The projections from two different viewpoints. The rays $O_L - p$ and $O_R - p'$ intersect at point P while $O_L - p$ and $O_R - q'$ intersect at point Q in the 3D space, which means that once the positions of the projection points on the left image and right image are known, the depth of point can be determined.

$$p' = K_r P_{l->r} K_l^{-1} p Z \tag{1.8}$$

Principle: To estimate depth Z for every pixel on the image, the crucial step is to identify the projection pairs like p and p' from stereo cameras according to the pixel appearance similarity. However, performing a search for matching points across the entire image would make the amount of computation increase exponentially, leading to a lot of computing consumption. As shown in Fig. 1.8, stereo depth estimation can utilize the epipolar line constraint to limit the search domain for corresponding points to a single row of pixels, significantly reducing computational complexity. Specifically, it requires the optical axes of the stereo cameras to be parallel, the image planes to be aligned at the same height, and the cameras to have identical intrinsic parameters. As depicted in Fig. 1.8, given X_L, X_R and B, the distance of L can be computed as $B + X_R - X_L$. Therefore, utilizing similar triangle principle formulated on the left of Eq. 1.9, the depth Z can be calculated based on the baseline B, focus length f and the disparity d which is defined as $d = X_L - X_R$. For a calibrated stereo camera system, both B and f are known constants, meaning the depth calculation depends solely on the disparity d.

$$\frac{B}{B + X_R - X_L} = \frac{Z}{Z - f} \Rightarrow Z = \frac{B * f}{X_L - X_R} \tag{1.9}$$

Basic steps: In practice, stereo depth estimation typically involves the following steps:

- Rectification: The rectification is a series of processes to align the image planes of two cameras so that corresponding points in the stereo image pair lie on the epipolar line. As introduced in the section 1.1.2, pictures captured by camera lens generally suffer from distortions. To eliminate this effect, the first step of rectification is to remove distortions. After performing camera calibration, the distortion parameters as well as other intrinsic parameters can be known, allowing de-distortion operation is accomplished through Eq. 1.1 and 1.2. The next step of rectification is to apply homography transformation on the images to ensure that the image planes are co-planar and corresponding epipolar lines are horizontal. Practically, by performing the homography transformation, i.e. Eq. 1.10

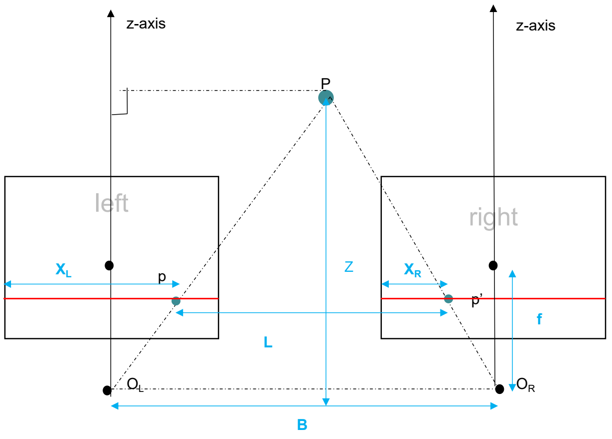


Figure 1.8: The epipolar line constraint of stereo cameras. The projections p on the left image plane and p' on the right plane are located on the same pixel row, marked in red, which is referred to as the epipolar line. Let the x-coordinate of p on the (u, v) coordinate system be X_L and that of p' be X_R, the distance between the two camera optical centers is B, the depth value Z of point P can be calculated according to the principle of similar triangles.

on the original image coordinates (u, v), new coordinates (u', v') can be established to reconstruct rectified images through bilinear interpolation sampling, where $\frac{\lambda}{\lambda'}$ represent scale factor, K, R indicate intrinsics and extrinsics of original camera respectively, K' is the mean of the left and right camera intrinsic parameters, R' is a rotation matrix dedicated for rectification.

$$\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = \frac{\lambda}{\lambda'} K' R'^{-1} R K \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \tag{1.10}$$

- Stereo Correspondence: The stereo correspondence aims at looking for homologous points in the epipolar lines of rectified image pairs, so that to estimate disparity according to the Eq. 1.9. To do this, a matching cost function is used to measure how similar two pixels are. The simplest matching algorithm is to compute pixel similarity one by one. However, due to the effect of noises, occlusion and ambiguous regions, this method commonly generate disappointed disparity predictions. In the light of this, the window-based matching costs [116, 163, 164] are proposed to find the best match by comparing small patches around a pixel in the left image with patches in the right image. Apart from local approaches, global approaches, e.g. graph cuts and dynamic programming, can also optimize the stereo correspondence problem by minimizing an energy function over the whole image.

- Post-processing: After establishing the correspondences for each pixel point, stereo depth estimation requires some post-processing steps, e.g. filtering [119], to remove noise and improve the quality of the final disparity map.

In summary, stereo depth estimation is able to estimate scene depth by using two cameras with a fixed baseline distance. It does not require additional hardware, making it suitable for various environments, and providing high-resolution depth maps. However, stereo depth estimation faces challenges in textureless regions, occlusions, and repetitive patterns, which can make matching difficult and lead to inaccuracies in the depth estimation. Additionally, it may struggle in low-light conditions or with reflective surfaces. Regarding the above issues, another depth estimation approach, i.e multi-view depth estimation, is proposed to improve prediction accuracy.

1.4.3 Multi-view depth estimation

Multi-view depth estimation employs three or more cameras to capture images from different viewpoints, offering a significant improvement over traditional stereo vision methods. By leveraging information from several viewpoints, multi-view systems can eliminate problems caused by occlusions and textureless regions in the stereo settings, thus generating more accurate and detailed 3D representations of a scene. The geometric principles of multi-views depth estimation are based on the perspective projection and feature matching as stereo depth estimation does, but as it involves multiple camera viewpoints, multi-view depth estimation includes many additional processing steps.

The structure from motion (SfM) [149] is the most commonly utilized method for multi-views depth estimation. The pipeline of SfM typically follows a series of steps as below:

- Feature detection and matching: The first step of multi-views depth estimation is to extract keypoints in each image using feature extraction algorithms, e.g. SIFT [101] and SURF [11]. Then the correspondences between the detected keypoints in different images can be established through nearest neighbor search.

- Camera pose estimation: Once correspondences between two images are established, it is mathematically solvable to calculate camera poses. Specifically, given correspondence coordinates p_1, p_2 and intrinsic parameters K, the epipolar constraint can be formulated as E.q 1.11, where $[t]_\times$ represents the skew-symmetric matrix of the translation vector t and R is the rotation matrix of camera pose.

$$p_2^T K^{-T} [t]_\times R K^{-1} p_1 = 0 \tag{1.11}$$

  Based on this equation, the Fundamental Matrix denoted as $K^{-T}[t]_\times RK^{-1}$ will be solved when giving 8 known correspondences, which allows to decompose the translation vector t and rotation matrix R of camera poses.

- Triangulation: When obtaining the relative position relationship between the two cameras, the depth relative to the camera 2 can be calculated according the Eq. 1.12. Here, x_1 and x_2 are normalized coordinates of p_1 and p_2. Consequently, if d_2 is known, the depth relative camera 1 can also be determined.

$$d_2[x_1]_\times R x_2 + [x_1]_\times t = 0 \tag{1.12}$$

- Global Optimization: Through the previous steps, the three-dimensional space information has been constructed. However, there are slight differences among 3D scenes reconstructed from different cameras, multi-views depth estimation has to apply global optimization techniques such as bundle adjustment [106] to refine the estimates of camera poses and 3D points simultaneously. This step minimizes the reprojection error across all images, enhancing the overall accuracy of the reconstruction.

By repeating above procedures on all the image pairs with different viewpoints, the three-dimensional information of real world will be established gradually. Compared with stereo depth estimation, the use of multiple camera views allows for a more accurate depth estimation because it can avoid occlusion issues effectively. Nevertheless, it cannot be ignored that multi-views approaches elevate computational complexity associated with processing multiple images, thereby posing challenges for real-time applications. Furthermore, the requirement for precise calibration and synchronization of multiple cameras also increase computational cost of depth estimation.

1.5 Monocular Depth Estimation

Different from LiDAR, stereo or multi-views depth estimation which either requires complex camera calibration or costly sensor equipment, monocular depth approaches only lie in a single camera, making it more flexible and cost-effective to deploy in various reality environments. Although monocular depth estimation has seen great development potentials in 3D vision applications, it is inherently an ill-posed problem due to depth ambiguity and scale inconsistency. As illustrated in Fig. 1.7, if without a reference image from a second viewpoint, points at different depths in the 3D space will be projected onto the same location on the image plane, which poses challenges for monocular depth estimation to accurately differentiate depth information. Motivated by the advantages of monocular depth estimation, there has been a lot of work attempting to solve these problems. The following content will be divided into into three categories to introduce existing methods from the principles, applications and limitations respectively.

1.5.1 Machine learning-based methods

In 2005, researchers [144] at Stanford University proposed a method that maps RGB images to depth maps using a Markov Random Field(MRF) approach. Specifically, depth estimation model can be defined as a continuous conditional random field like Eq. 1.13 where the posterior probability P is composed of unary and pairwise potentials with parameters σ, θ. Given ground truth depth $\hat{d}$ and RGB image X, the parameters σ, θ will be solved to infer depth through maximizing posterior probability P.

$$\hat{d} = \max_{d \in \Omega} P(d|X, \theta, \sigma) \tag{1.13}$$

Inspired by the MRF-based monocular depth estimation, Ashutosh Saxena et al. [147] assumed that environment is composed of multiple small planes, i.e superpixels. Under this assumption, depth estimation can be achieved by inferring the location and orientation of each plane. To further improve the accuracy of reconstructed 3D structures, the model incorporated some additional visual clues, e.g. neighboring superpixels are more likely to be connected to each other, long straight lines in the image plane are more likely to be straight lines in the 3D model. These geometric priors serve as conditional constraints, facilitating more precise depth estimation.

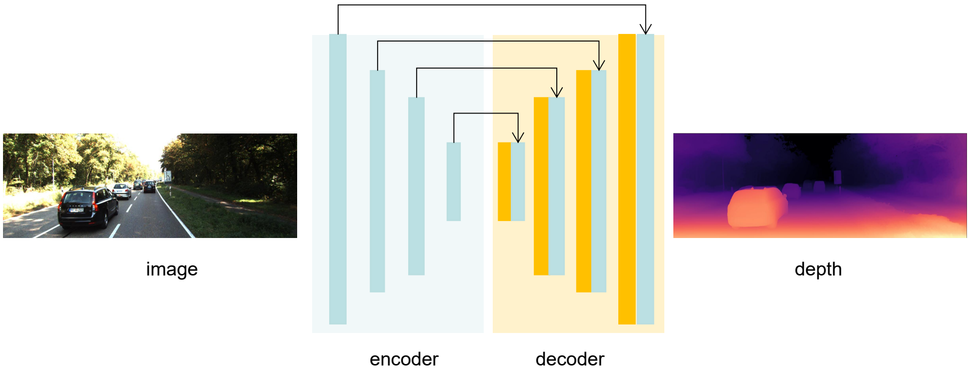


Figure 1.9: The U-net structure for depth estimation. The translation from image to depth is achieved by the end-to-end training paradigm where each layer of encoder will be concatenated with corresponding decoder layer, enabling the preservation of essential features.

Instead of building up the explicit probability models(e.g. Gaussian model or Lagrangian mode) to compute depth, Fayao Liu et al. [105] employed deep convolutional neural networks to model the unary and pairwise potentials, allowing to resolve parameters directly using back propagation.

While these techniques has seen some success in depth estimation, it encounters difficulties in ensuring precision for complex scenarios, as hand-crafted parameter models often fail to accurately capture real-world mapping relationships. With the rise of deep learning and its widespread applications in computer vision, numerous innovative works have emerged, leveraging advanced neural networks to achieve depth regression more effectively.

1.5.2 Supervised methods

In 2014, Eigen et al. [42] first designed an end-to-end depth estimation model based on the multi-scale Convolutional Neural Network (CNN) [88] which combined both coarse and fine-level predictions to capture global scene structure and fine-grained details. To ignore the effect of absolute scale among different images, Eigen et al. proposed the scale-invariant error to better measure relationships between points in the scene. As remarkable abilities of neural networks to extract hierarchical features from raw pixel inputs, the final results performed on the NYU [127] and KITTI [51] datasets proved great improvement in accuracy of monocular depth estimation.

Following this work, numerous methods have been proposed to further improve monocular depth estimation. For example, Ibraheem et al. [4] utilized the U-net structure [141] to map images into depth maps, which effectively mitigates information loss by incorporating skip connections between the encoder and decoder layers. The network structure is illustrated in Fig. 1.9 which has been regarded as the classical depth estimation network. In terms of training loss of network, beyond minimizing the absolute error between predicted depth and ground truth, they put forward the gradient smooth loss. This regularization term encourages smoother depth predictions, particularly in regions with homogeneous texture, thereby enhancing the overall depth map quality.

As the lack of long-range dependencies for the convolutional operations, CNN-based methods typically decline the performance in the global depth consistency. In contrast, models utilizing multi-head self-attention mechanisms, such as the Transformer [168], have proven highly effective in capturing long-range correlations, achieving significant success across various image translation tasks. Building on this insight, Agarwal et al. [2] introduced a Transformer-based architecture for monocular depth estimation to improve depth prediction for distant and small-size objects. To maintain local depth consistency, they retained a CNN branch within the model. Additionally, to facilitate effective fusion between CNN-derived features and Transformer-derived features in the encoder, they introduced a cross-attention module, enabling efficient feature interaction and integration for improved depth estimation accuracy.

Despite more complex network architectures have improved the generalization capabilities of depth estimation models, achieving robust depth predictions still requires training on large and diverse datasets. However, these datasets often exhibit biases in depth range, making it challenging to effectively achieve cross-dataset transfer during training. To overcome this limitation, Rene Ranftl et al. [138] proposed a method that aligns ground truth and predicted depths using a least squares approach before calculating their absolute errors, ensuring both scale and shift invariance. By employing this scale-invariant loss, the network can be trained jointly on five datasets involving various scenes, which demonstrates powerful generalization performance on unseen environments.

Compared with target-driven discriminative models, generative models tend to focus on the overall distribution of data, thereby being expected to have stronger generalization performance. In 2020, Jonathan et al. [64] proposed generative diffusion model, which achieved remarkable success in image synthesis and inspired significant interest in applying generative models to various computer vision tasks. Building on this progress, Bingxin et al. [81] reformulated depth estimation as a conditional generation task, where the generative network directly learns the distribution of depth maps conditioned on RGB inputs. The experimental results demonstrated that this method significantly outperformed Midas [138] and other depth-error-driven methods, achieving superior performance in depth estimation.

Driven by advancements in deep network architectures and the use of diverse datasets, supervised monocular depth estimation has achieved notable success in robustness and accuracy. However, these methods heavily rely on large annotated depth labels, which are costly and time-consuming to collect. Additionally, monocular depth estimation suffers from scale ambiguity due to the lack of metric information in single images, making it difficult to predict absolute depth from relative depth. This limitation reduces its reliability compared to methods like LiDAR or stereo depth estimation, which provide more accurate metric depth measurements.

1.5.3 Self-supervised methods

Compared to data-driven supervised methods, self-supervised monocular depth estimation enables depth prediction from a single image without the need for annotated labels, significantly reducing the dependency on labeled data and making the acquisition of depth information more efficient and easier.

The principle of self-supervised monocular depth estimation is similar to stereo matching that mainly depends on epipolar geometry as formulated in 1.8. However, while the epipolar constraints in stereo vision are applied to stereo image pairs, in monocular depth estimation, these constraints are applied to consecutive frames. Consequently, the epipolar constraint of monocular depth estimation can be expressed as Eq. 1.14 where K represents the camera in-

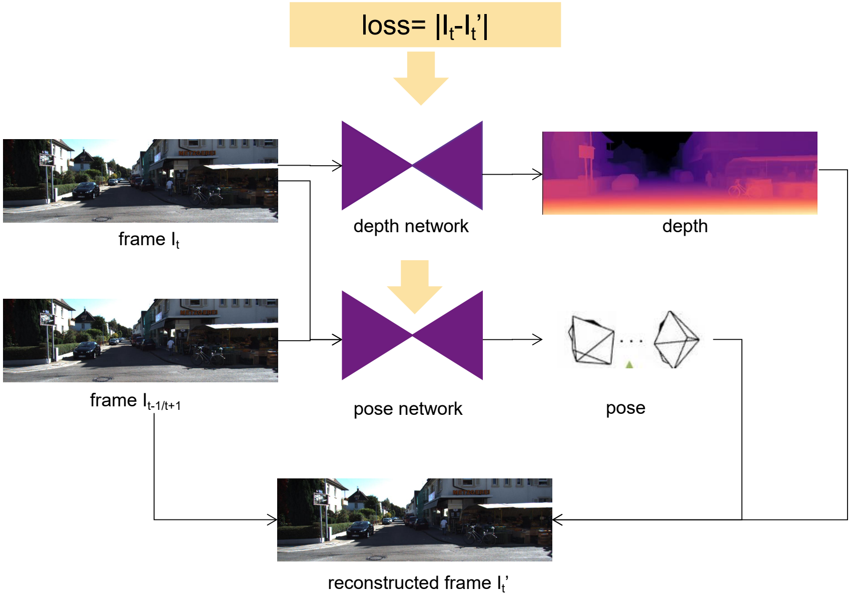


Figure 1.10: The pipeline of self-supervised monocular depth estimation. Both depth network and pose network will be trained by the mean absolute loss between warped image I_t' and target image I_t.

trinsics, $P_{t->t-1}$ refers to camera pose from time t to $t-1$, p_{t-1} and p_t denotes coordinates of image projection on $t-1$ and t respectively, Z is the depth for the p_t.

$$p_{t-1} = KP_{t->t-1}K^{-1}p_tZ \tag{1.14}$$

Inspired by this principle, Garg. et al. [227] first proposed a self-supervised monocular depth estimation paradigm in 2017. This method designed two networks to predict depth and camera pose separately which enable the projection of coordinates from current frame to the previous or subsequent frame based on the constraints in Eq. 1.14. The current view will be reconstructed by inverse warping from previous/next frame where each pixel is sampled according to the reprojected coordinates p_{t-1}/p_{t+1}. The absolute error between target frame and reconstructed frame will serve as objectives to train both depth network and pose network. The overall pipeline of self-supervised monocular depth estimation is illustrated as Fig. 1.10.

Although this approach successfully predicted depth just from single video sequences and established the standard paradigm for self-supervised monocular depth estimation, unreliable reprojected losses caused by occlusion and dynamic objects will disturb network to fit actual depth distribution, declining the prediction accuracy for the network. To further minimize the gap between supervised methods and self-supervised approaches, Godard. et al. [55] introduced a novel reconstruction loss in Monodepth2. Instead of averaging the photometric error over all source images, the reconstructed error of each pixel will be determined by the minimum of each warped frame on the corresponding location. This per-pixel minimum reprojection loss is beneficial to exclude problematic pixels, such as out-of-view pixels and occluded pixels. Besides,they applied the auto-masking algorithm to mask out stationary frames and dynamic pixels when computing warped losses, where the pixel will be included only when reprojection error of the warped image is lower than that of the original unwarped image. Furthermore, to prevent the training objective getting stuck in local minima, they also adopted multi-scale depth estimation where the network will output multi-scale disparities from each layer of decoder and performing reprojected loss on each of them. Benefiting from these improvements, Monodepth2 significantly enhance the predicted accuracy of monocular depth.

Although traditional self-supervised monocular depth estimation like Monodepth2 has eliminated the issues of occlusion, they are limited by the use of previous or subsequent single frame to compute reconstructed losses. These methods neglect much valuable temporal information. In 2021, Watson1 et al. [185] proposed ManyDepth to make use of more historical frames to estimate depth. This work is inspired by multi-view stereo to build up a cost volume [195] which will be fed into the decoder to predict depth. Compared to 2D features, 3D cost volume constructed by historical frames can contain more information, thus effectively improving accuracy of depth estimation. Nevertheless, ManyDepth are more sensitive to the dynamic objects compared with single-frame depth estimation, leading to significant declining results on the moving areas, which makes it have to adopt additional monodepth branch to offset overfitting caused by cost volume.

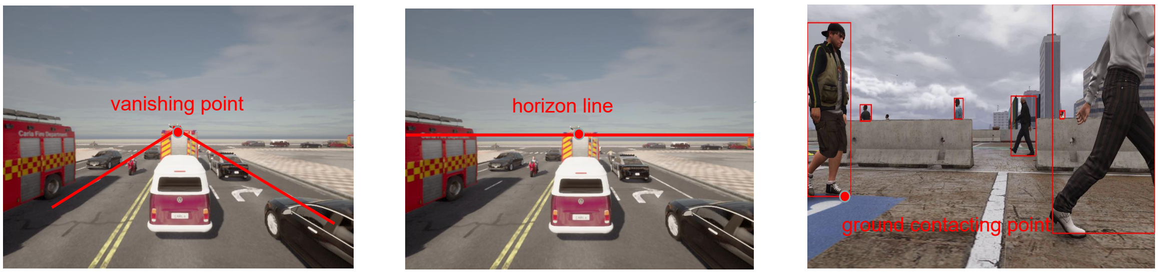
In summary, existing methods of self-supervised monocular depth estimation commonly leverage view consistency principle, allowing to acquire actual scene scales compared to supervised approaches. This characteristic facilitates to restore metric depth from the predicted relative depth. Besides, as mentioned above, they do not require large amount of depth labels but only need readily available video sequences, thus providing a more flexible and cost-efficient solution for applications such as autonomous driving. However, self-supervised monocular depth estimation typically depends on the assumption of static scenes where dynamic objects will be regarded as ill-posed issues. Although several studies have attempted to improve prediction robustness by replacing CNNs with Transformers [220, 226], these challenges are particularly evident in the motion-dense scenarios where moving objects are frequently assigned extreme depth values, either as infinitely distant or close.

Although monocular depth estimation faces many challenges, certain geometric cues within visual scenes are often overlooked. These geometric features can play a crucial role in enhancing depth estimation and contribute to recover accurate depth scales. In light of this, the present work leverages ground geometry information to propose two novel improvement strategies based on monocular depth estimation, each designed specifically for dynamic scenes and static scenes respectively. These methods are elaborated in detail in Chapters 1.6.

1.6 Monocular Depth Estimation Guided by Ground Geometry

In 3D vison tasks, geometric constraints are crucial to retrieve 3D information. These geometric conditions provide additional cues about the scene's structure, compensating for the lack of stereo image pairs or LiDAR data.

For example, vanishing points and horizon lines are crucial geometric elements that can contribute significantly to many 3D vision tasks. Vanishing points represent the intersection of parallel lines in the 3D world when projected onto a 2D image plane, while the horizon line marks the boundary where the sky meets the ground in a scene. In particular, all vanishing points will lie on the horizon line. These geometric cues are particularly useful in determining



(a) Vanishing point **(b)** Horizon line **(c)** Ground plane

Figure 1.11: The illustration of some geometric cues: (a) Vanishing point: the point where two parallel lines in the 3D converge; (b) Horizon line: the boundary where the sky and the ground meet in the distance. It is the farthest visible boundary that an observer can see on the surface of the earth; (c) Ground plane: the plane where most of objects are standing.

the 3D spatial structure of objects. For instance, vanishing point can aid in the estimation of the object's 3D bounding box as the edges of a cuboid in 3D space, being parallel, will converge at the same vanishing point. Additionally, the horizon line, which represents the farthest visible boundary in an image, provides a key geometric constraint: points located above the horizon line can be considered at an infinite distance, resulting in their disparity values approaching zero. This information is particularly valuable in constraining depth estimation in distant regions of the scene.

Another important geometric cue is the planar surface, often employed in outdoor environments. The ground planes can help in determining relative depth and scale, particularly in the environments with structured elements like roads or squares. Furthermore, the ground plane can also present a strong geometric prior about depth estimation, because objects in the scene are generally standing on the ground, indicating consistent depth between them.

In summary, geometric cues can effectively convey 3D scene information for monocular tasks. This insight has inspired some researchers to leverage these cues to enhance the accuracy of monocular depth estimation, such as GEDepth [199], Planedepth [177]. In this thesis, we will further develop ground plane geometry to resolve existing challenges in monocular depth estimation tasks.

1.6.1 Our Contributions

Zero-shot metric depth estimation for static scenes

The legal surveillance cameras in some public areas can help to provide security protections, traffic monitoring and smart management. For instance, estimating crowd density within a scene, monitoring interpersonal distances to ensure compliance with social distancing protocols, or detecting collisions between vehicles are practical applications of video analysis. All of these applications rely on depth estimation, making metric depth estimation in static scenes particularly essential. The static scene refers an environment where a camera is installed at a certain position with a fixed pose and height. In this setting, the background is stationary, while people and vehicles are always moving.

Since the majority of surveillance cameras are monocular, obtaining scene depth from a single captured image is highly required. However, as discussed in Sec. 1.5, estimating metric depth from a single image is inherently an ill-posed problem due to the lack of triangulation cues from multiple viewpoints. Recent advances in supervised monocular depth estimation have enabled depth prediction for static scenes. These data-driven methods are trained on large-scale datasets with precise depth annotations, such as LiDAR data, and employ advanced network architectures, like Transformer [168], thereby achieving strong generalization capabilities in diverse environments. These state-of-art pretrained depth models, such as Midas [138], DPT [137], can provide dense depth maps for any static scene.

However, these models are still subject to failures in specific settings that are under-represented in the training data, e.g., on ambiguous objects such as mirrors, or more commonly, when dealing with images taken from a perspective rarely, or some moving objects that are less observed in the training process. Therefore, directly applying existing depth models on the new scenes will bring out some unexpected predictions, like inconsistent depth on the objects or infinite distance on the moving targets. To address this issue, we propose an on-site adaption technique which are based on a geometric assumption, i.e. most of objects in the scene stand on the ground plane and depth of them should be consistent with their ground contact points. In practice, we first identify all the agents including pedestrians and vehicles by using instance segmentation networks, e.g. Mask R-CNN [1], and then replace depth of each agent with that of ground contact point. In this way, depth predictions can be updated and will serve as pseudo labels to fine-tune depth networks. The experimental results demonstrate that this one-site adaption technique is able to adjust depth for all the agents, allowing to provide more reliable initial dense depth maps for any unseen environments.

Although state-of-the-art depth estimation models can present plausible initial depth predictions, determining the actual scene scale remains ambiguous, which is crucial for converting relative depth to metric depth. Drawing inspiration from the ability of the ground plane to provide essential scale information, we leverage this feature to facilitate scale alignment. A critical step in this process involves aligning the actual ground plane with the predicted ground plane. According to the linear geometric principle, assuming a planar ground surface, the actual ground normal can be estimated by solving a linear equation based on multiple sets(more than three sets) of standing 2D coordinates and the corresponding 2D height of the same agent. In practice, when an individual moves within the monitored field of view, his varying standing coordinates and height in the image can be recorded. The moving individual serves as a geometric probe for computing the actual ground normal. To derive the predicted ground normal, we can mask out the ground area for the image and project them onto a 3D plane using initial depth predictions. This projection enables the calculation of the ground normal through the least squares method. Once the predicted planes are rotated to be aligned with actual ground, the predicted depth will have the same scale as the actual scene depth.

Due to the scale of predicted depth has been aligned with actual scenes, the final step to predict physical distances to the camera is estimating a scaling factor that can convert all relative depths to absolute depths. To achieve this goal, we only use the estimated actual ground normal and the camera's height from the ground to calculate minimum absolute depth which is located in the middle-bottom point of the image. In this way, the scale factor will be known through dividing the absolute depth of this point by the relative depth.

In summary, the proposed method leverages human probes to detect scene scales without reliance on prior information, thereby enhancing the practicality of reconstructing 3D structures for static scenes. Furthermore, all modules within this framework are designed to be efficient and lightweight, ensuring high adaptability across various unseen environments.

Self-supervised depth estimation for moving objects with ground propagation

Self-supervised depth estimation is aimed to eliminate the need for labeled depth data by training models to predict depth using geometric consistency between video frames. Instead of relying on a large number of ground truth depth labels, self-supervised depth estimation learns to minimize reconstruction errors by comparing the original image with a warped or synthesized view generated from the predicted depth. This approach significantly reduces the dependency on expensive labeled datasets and acquires actual 3D structures of scenes, enabling the model easier to restore the metric distances from predictions, thereby it is especially useful for real-world applications and dynamic environments.

However, self-supervised depth estimation heavily relies on the assumption of photometric consistency, which requires all the objects in the scenes should remain static to ensure pixel appearances unchanged between consequent frames. Therefore, dynamic objects, e.g pedestrians and vehicle, present significant challenges on the self-supervised depth estimation as moving speed and directions of dynamic objects are always unpredictable.

Recently, various methods have been developed to address this ill-posed problem in depth estimation. One effective technique involves masking dynamic objects during training, which allows the model to ignore errors from these objects when calculating the loss function. However, this approach can significantly reduce the network's generalization ability due to fewer dynamic features being learned. Another approach introduces additional perceptual information into depth estimation. For instance, SC-DepthV3 [156] refines the depth estimation of moving objects by incorporating pseudo-depth labels, generated by existing depth models [206], as supplementary supervision. Similarly, Klingner et al. proposed a dual-semantic network that uses semantic segmentation to guide depth estimation. While these auxiliary perceptual information can improve accuracy as they share the similar context with depth, they often require additional computational resources, which may not always be accessible. Another intuitive solution is to implement multi-task estimation alongside depth, such as optical flow [160] or scene flow estimation [121]. Since these tasks explicitly capture dynamic object motion, they can help build a complete 3D structure of the scene and mitigate the ill-posed challenges introduced by moving objects. For example, Yihong Sun et al. [157] proposed to learn optical flow, depth and dynamic masks through jointly training three independent networks. However, multi-task estimation inevitably introduces auxiliary network structure, typically including 2-3 encoder-decoder networks, which increases memory usage and the training complexity.

To enhance depth prediction for dynamic objects in self-supervised monocular depth estimation, we introduce a geometric assumption: the depth of an object should match that of its ground contact point, ensuring depth consistency. Based on this principle, we propose a novel ground propagation module, designed to integrate smoothly into any state-of-the-art monocular depth network, such as Monodepth2 [55] or Lite-Mono [220], without requiring additional parameters or specialized training procedures. The concept behind ground propagation draws inspiration from the intrinsic properties of latent spaces within decoder networks, where feature activations can be categorized into depth-aware and detail-aware features. Depth-aware features are responsible for reconstructing the global depth structure, capturing the general depth distribution across the scene, while detail-aware features focus on refining depth edges and preserving finer details. Therefore, by refining the depth-aware feature maps, the accuracy of the final depth predictions can be significantly enhanced.

The first step of ground propagation involves identifying depth-aware feature maps from the decoder outputs. To achieve this, we construct pseudo depth or disparity maps and then evaluate cosine similarity between feature activations and the pseudo depth. Feature maps with higher similarity scores will be identified as depth-related. Subsequently, within the depth-aware feature maps, ground features of each dynamic object will be propagated upward in an iterative way while keeping other features and background areas remained. This process replaces the outlier depths of moving objects with the depth values of their ground contact points, enabling consistent and more accurate depth calibration for dynamic objects. Since ground propagation is applied exclusively to depth-aware feature maps, depth details can be preserved in the final predictions. Compared to other ground-based monocular depth estimation methods [124], our approach supports end-to-end training and flexibly handles objects with surfaces that are not perpendicular to the ground. Finally, we propose that the final feature map should be a weighted sum of the original and updated features, effectively preserving features of objects that already meet the multi-view consistency principle. To accomplish this, we introduce clipping normalization, which calculates the weight based on the differences between the two sets of features, ensuring more reliable updates.

To conclude, our approach effectively address challenges faced by existing self-supervised monocular depth estimation, allowing to improve accuracy for dynamic objects. More importantly, our proposed ground propagation module is light-weight and free parameters, making it possible to be integrated into any existing baseline methods.

Chapter 2

Related Work

Depth estimation has been a significant area of research within computer vision, particularly due to its vital role in understanding 3D scene structure from 2D images. This section outlines key contributions in the field, divided into multi-views and stereo depth estimation, supervised and self-supervised monocular methods, alongside the integration of geometric priors.

2.1 Stereo and Multi-view Depth Estimation

Stereo depth estimation involves the extraction of depth information from two stereo images by computing the disparity between corresponding points. Early methods focused on traditional computer vision techniques, which rely heavily on the carefully designed stages of cost computation, aggregation, optimization, and refinement. For example, Birchfield and Tomasi [20] improved cost computation by introducing a gradient-based matching cost, which became more robust to intensity changes between stereo images. In 2002, Birchfield and Tomasi [21] further proposed a measure of dissimilarity by using the linearly interpolated intensity functions surrounding the pixels, which is insensitive to sampling. After computing the matching cost, Yoon and Kweon [210] introduced an adaptive support-weight approach, where support regions are determined based on color and spatial proximity. A highly influential method is Semi-Global Matching (SGM) [63]. SGM aggregates matching costs by considering multiple paths across the image, which greatly enhances accuracy in both textured and textureless regions, and has become a benchmark in stereo estimation. As for cost volume optimization, techniques like graph cuts [87, 111, 170] and belief propagation [44, 155, 159] became popular for their ability to enforce smoothness while preserving depth discontinuities. To allow disparity maps obtained from initial optimization more accurate, Yang et al. [198] introduced constant-space belief propagation to improve disparity map accuracy with a faster computation time. Bilateral filtering techniques [162] have also been used for post-processing to refine depth maps by smoothing them while preserving sharp edges.

With rapid development of deep networks, they have provided powerful alternatives for stereo depth estimation by learning to extract dense correspondence and depth from stereo pairs. In 2016, Zbontar and LeCun [217] proposed MC-CNN, which leveraged a convolutional neural network (CNN) [88] to compute matching costs between stereo image patches. Luo et al. [111] introduced a faster architecture by simplifying the CNN into a dot-product-based approach, making the network more computationally efficient. Moreover, there are also some work attempting to improve optimization or refinement procedures of the SGM [63] pipeline. For example, Chang and Chen [44] introduced PSMNet, a pyramid spatial pooling module to capture global context, to allow for a more accurate disparity prediction. Shaked and Wolf [159] designed the Global Disparity Network (GDN) to locally optimize cost volume. Yang et al. [13] proposed to combine semantic segmentation and stereo matching into a unified framework to incorporate high-level semantic information for depth estimation.

Furthermore, several works began developing end-to-end stereo matching networks [39, 73, 161, 186, 204, 212]. DispNet [117] by Mayer et al. used an encoder-decoder structure with skip connections to predict dense disparity maps directly from stereo image pairs. Kendall et al. [82] advanced this field with GC-Net, which integrated a 3D cost volume and a differentiable soft argmin layer to aggregate matching costs end-to-end. Similarly, Chang and Chen [27] introduced PSMNet, leveraging a spatial pyramid pooling module and stacked hourglass architecture to improve disparity estimation. Pang et al. [129] developed a cascade residual learning framework that enhanced disparity prediction by reducing errors iteratively. Following this, Cheng et al. [33] proposed to use neural architecture search to optimize network architecture, balancing performance with computational cost. Tonioni et al. [165] focused on online adaptation techniques that allowed stereo networks to adapt to new environments without retraining, improving real-time application. To pursue fast inference, Liang et al. [97] presented an efficient architecture using hierarchical cost volumes. In contrast, Yao et al. [203] proposed MVSNet, a multi-view stereo approach that extended disparity prediction to multiple images, further improving depth estimation for 3D reconstruction. More recently, Zhang et al. [222] applied guided aggregation networks to enhance matching accuracy by learning better feature aggregation strategies.

Multi-view depth estimation extends stereo by incorporating images from multiple viewpoints, often used in more complex 3D reconstruction systems. Traditional approaches often rely on geometric methods such as Structure from Motion (SfM) [149] and Multi-view Stereo (MVS) [47]. Based on this insight, PatchMatch-based MVS introduced by Schönberger et al. [151] provides a more efficient way to perform dense stereo matching, particularly improving speed and accuracy in unstructured image collections.

Recently, deep learning-based methods also revolutionize multi-view depth estimation [48, 154, 171, 194, 201, 224]. A pivotal contribution is MVSNet [203], which established an end-to-end learning framework using a cost volume for depth inference. Following this, DeepMVS [66] introduced a multi-scale approach, effectively combining features from different layers to improve depth estimation. Another important work, DSN [193], utilized a dual-stage network to refine depth predictions through a cascaded learning process. Moreover, MVSNet+ [36] built upon MVSNet by implementing multi-view feature aggregation, resulting in better depth predictions by leveraging diverse viewpoint features. Additionally, Zhou et al. developed a self-supervised framework for multi-view depth estimation, allowing for depth learning without ground truth labels [228].

More advanced methods such as Neural Radiance Fields (NeRF) [31, 122], proposed to synthesize novel views of complex scenes by modeling the volumetric scene as a continuous function of 3D coordinates. Building on this, PlenOctrees [46] combined NeRF with octree structures for faster rendering, thus improving depth estimation in real-time applications. Another significant contribution is KiloNeRF [140], which extended NeRF to handle large-scale scenes, demonstrating that high-quality depth can be inferred from vast datasets. Lastly, DeepVoxels [182] integrated NeRF with voxel-based representations, achieving high-quality depth estimations while preserving details in complex scenes.

Despite advancements, stereo and multi-view depth estimation still faces challenges, including sensitivity to occlusions, reliance on complex camera calibration, and challenges in handling dynamic scenes. These limitations sparks the research about monocular depth estimation.

2.2 Supervised Monocular Depth Estimation

Conventional monocular depth estimation is mainly based on the machine learning process with parameter [65] or non-parameter [79] and explicitly extracted hand-crafted features from single image [80, 144, 147]. These methods struggle to effectively manage more complex real-world scenarios, resulting in deep learning-based monocular depth estimation becoming the prevailing focus of current research [50, 74, 77, 89, 115].

The first work about supervised monocular depth estimation was released by Eigen et al. [54], they utilized multi-scale convolutional neural networks (CNNs) [88] to predict dense depth map, demonstrating the potential of deep learning in this area. Inspired by this work, some researchers extended this idea by introducing deep convolutional neural fields that combine CNNs with conditional random fields(CRFs) [158] to enforce spatial coherence in the predicted depth maps [26, 68, 173, 221]. For instance, Bo Li et al. [92] achieved this by incorporating CRFs into a post-processing step, refining the depth estimation across different scenarios. Beyond basic CNN model, there are also some different variants, such as VGG [152], ResNet [62] and DenseNet [69], to be applied to improve accuracy of depth estimation. Laina et al. [90] introduced residual connections to solve the gradient vanishing problem in deep networks, allowing deeper networks to be trained more stably. Shen et al. [5] proposed to predict depth based on DenseNet, enabling to capture more details in complex scenes. Liu et al. [103] combined the VGG network with conditional random fields for monocular depth estimation. The VGG network used in this model captures global and local information and generates a preliminary depth map, which was then further refined by CRFs.

To improve the accuracy and robustness of depth estimation, multi-task learning (MTL) frameworks has become another trend. It leverages shared representations across distinct vision tasks, allowing the model to enhance performance by learning richer features. The first branch is the combination of semantic segmentation and depth estimation. Due to semantic segmentation and depth estimation typically assign the same value on the small object, many previous works have focused on solving these two tasks in a joint manner [76, 102, 109, 175]. For example, Mahyoub et al. [114] jointly trained two U-Net [141] like networks to predict semantic segmentation and depth maps which shared the same encoder but have different task-specific decoders. Chen et al. [32] further proposed the self–calibration fusion structure to more effectively fuse these two features. In contrast, Pierluigi et al. [136] put forward cross-domain discontinuity loss to enforce feature fusion between depth and semantic domains. Apart from semantic information, monocular depth estimation can also meet significant improvement when combined with other vision clues, such as optical flow [160] and surface normal [180]. Wang et al. proposed TartanVO [179] which is a learning-based visual odometry (VO) system that estimates both optical flow and depth maps simultaneously. It focuses on real-time, efficient performance by combining depth and flow estimation to support visual odometry tasks. Built on state-of-art optical estimation network RAFT [160], Zachary et al. further combined depth and optical flow estimation to understand comprehensive motion representation, i.e 3D scene flow [121]. In 2018, Fu et al. [133] proposed GeoNet to simultaneously estimates depth, surface normals, and optical flow from monocular videos. The main contribution of this work is to introduce a geometry-based loss function that enforces consistency between depth and surface normals. Similarly, Yan et al. [190] present a multi-task CNN to jointly learn surface normal and depth maps but introduced CRFs [158] to refine the superpixel-wise predictions.

With the availability of various large-scale datasets [51, 96, 172] and advancements in state-of-the-art deep networks with strong generalization capabilities [10, 70, 107, 168], an increasing number of studies are focusing on zero-shot monocular depth prediction. One notable contribution in this area is MiDaS [138], which employs a scale-invariant loss to enable training across multiple cross-domain datasets. This approach allows the model to capture richer scene information, resulting in improved accuracy and generalization performance across diverse environments. In 2020, Yin et al. [205] devoted a novel synthesis dateset, Diverse Scene Depth, and proposed a multi-curriculum learning method to enhance generalization performance. Similarly, Eftekhar et al. [40] offered a diverse collection of high-quality, synthetic, and real-world images with ground truth annotations for 3D tasks and integrated multiple 3D vision tasks to improve 3D reconstruction. Although scale-invariant loss enables networks to be trained on the multiple datasets with different ranges, it tends to ignore the scale information of specific scene, posing challenge to reconstruct real scenes. To solve this problem, Yin et al. [206, 207] proposed the dual networks which separately learn depth information and scale factor. Another influential work is inspired by Transformers /citevaswani2023attentionneed, which proposed a dense image prediction frame and reached to state-of-art results in many vision prediction tasks, including monocular depth estimation. Moreover, DepthAnything [196, 197] proposed by Li et al. also achieved awesome generalization performance by predicting depth for unlabeled datasets through a teacher network, and then fine-tuning a student network both on the pseudo labels and real labels. The above work can be reviewed to perform a discriminate task, they heavily depend on the need of large-scale datasets. In contrast, generative models [57, 84, 128] are typically featured by high generalization performance in many tasks. In 2023, by exploiting existing state-of-art generative models, i.e. Stable Diffusion Model [125], Ke et al. proposed Marigold [81] to predict depth, which typically consider depth labels as known distribution.

While monocular depth estimation significantly lowers the cost of depth sensing and is applicable across a wide range of scenarios, it heavily relies on the availability of large-scale training datasets and struggles to recover the actual scale of scenes. These challenges can be effectively addressed by self-supervised monocular depth estimation methods, which alleviate the need for ground-truth depth data and enable the model to learn depth by leveraging geometric and photometric consistency from unlabeled video sequences.

2.3 Self-supervised Monocular Depth Estimation

In recent years, self-supervised monocular depth estimation has emerged as a promising alternative to supervised depth estimation methods. The early work in self-supervised monocular depth estimation was proposed by Garg et al. [49], which introduced a photometric consistency loss for training depth estimation networks using stereo image pairs. To enhance robustness of depth prediction, Godard et al. [53] imposed the left-right geometric consistency loss on the training objective, demonstrating superior performance compared to previous work. Following this, Zhou et al. [54] extended this approach to work with monocular video sequences rather than stereo pairs. They designed the monocular self-supervised framework, where the network simultaneously learns depth and camera pose estimation from consecutive frames, allowing it to predict depth just from consecutive video sequences in an unsupervised way. However, view synthesis loss generally suffer challenges when dealing with repetitive patterns or moving objects. To address this issue, some researchers worked on improving the loss function [55, 219]. Yang et al. [202] proposed smoothness regularization term to force depth consistency on the objects. Mahjourian et al. [113] assumed approximate geometry based matching loss to encourage temporal depth consistency. Godard et al. [53] incorporated an SSIM [184] loss to compute reconstructed differences between target image and warped image. Taking advantages of these improvements, Godard et al. proposed Monodepth2 [55] in 2019. The Monodepth2 improves upon prior approaches by incorporating an auto-masking technique to handle moving objects and occlusions, and introducing the minimum reprojection loss to choose the most photometrically consistent pixels during training. These enhancements allow the model to produce more accurate and robust depth maps, even in challenging real-world scenarios.

In addition, many researchers have also tried to improve accuracy by improving the network structure. For instance, Lyu et al. [112] re-designed bridge connections of U-Net[141], allowing to fuse more high-frequency information, thereby achieving depth estimation of high-resolution images. To acquire more precise and sharper depth maps, Yan et al. [191] introduced channel-wise attention into the decoder, assigning more attention for those highly correlated channels with depth prediction. Moreover, replacing CNNs [88] with Transformers [137] can also benefit improving prediction accuracy, as achieved by [9, 226]. To decrease memory occupation of network, Zhang et al. [220] designed a novel encoder-decoder structure based on self-attention execution and deployment [169] and dilated convolution [211], ensuring precision of prediction as well as lightweight.

Despite self-supervised monocular depth estimation has seen great success, it still faces challenges for dynamic objects as they break the assumption of static scenes, especially when dealing with motion-intensive scenes. To solve this problem, some methods proposed to mask out dynamic objects when computing reconstructed photometric losses. Jiang et al. [75] considered the occluded or dynamic pixels as statistical outliers in the photometric error map and introduced an efficient weighted scheme to reduce the artifacts caused by moving objects. Another solution utilizes a geometric prior, i.e. the appearances of static objects projected to the next frame should remain unchanged. Based on this geometric condition, some approaches identify moving objects by distinguishing if there is consistency between projected instance appearance and original one. For instance, Saunders et al. [143] filtered out objects with small instance overlap rate. Yue et al. [215] proposed minimum instance photometric residual and independently performed moving instance loss on moving object.

Besides, a more intuitive approach is to model motion model for each moving object. Insta-DM [91] exploited camera ego-motion network to predict object motion and established re-projected loss for objects independently. Alternatively, optical flow can also be utilized to predict object motion. Sun et al. [157] established an additional network to estimate optical flow. To regularize the learning of optical flow, they posed the background mask to ensure optical flow of background approaching to zero.

Similar to supervised monocular depth estimation, multi-task estimation also plays a crucial role in improving accuracy for self-supervised depth estimation [120, 120, 135, 208, 230], especially on the dynamic objects. Klingner et al. [86] designed a dual network to estimate depth maps and semantic segmentation which can effectively guide the process of depth prediction.

Zhao et al. [225] jointly trained the depth network and optical flow network which can generate dynamic mask by forward-backward consistency check of the optical flow.

The aforementioned methods utilize scene information from only the preceding and subsequent frames. In contrast, multi-frame monocular depth estimation can substantially improve depth estimation by adopting more historical frames in the prediction process [8, 45, 181]. As the first work in this field, ManyDepth [185], proposed by Jamie Watson et al., is inspired by the principles of stereo geometry to construct a cost volume from video sequences, which serves to predict depth through 3D convolution operations [166] subsequently. Motivated by ManyDepth, Guizilini et al. [59] incorporated attention mechanism into the computation of cost volume, specifically involving cross-attention and self-attention operations. Although multi-frame methods have advantages in static regions, they generally encounter challenges in the dynamic areas as the dynamic objects cause corrupted values in the cost volume. To address this problem, Rui Li et al. [95] proposed the cross-cue fusion (CCF) module to integrate clues from single and multiple views, making up for the weakness of multi-frame methods in dynamic targets. However, the complex network structures and redundant training processes in multi-frame approaches limit their scalability and further development.

Overall, while self-supervised monocular depth estimation significantly enhances the flexibility of depth acquisition, making it suitable for deployment in various autonomous environments, a notable gap remains compared to supervised methods in terms of generalization performance and the restoration of high-frequency details.

2.4 Geometric Priors in 3D Vision

In the monocular depth estimation, some known scene geometric priors can effectively assist the process of depth estimation. As a valuable scene scale probe, ground prior has been widely employed in many 3D vision tasks [35, 142, 178, 213, 229]. In 2023, Xiaodong et al. [200] proposed to improve depth estimation by embedding ground depth which is computed according to the camera intrinsic parameter. The integration process is guided by the ground attention which indicates the possibility of the point belonging to the ground area. Similarly, Aurélien et al. [25] explicitly fused ground depth into self-supervised monocular depth estimation through ground attention. Following in this, Moon et al. [124] put forward the ground smoothness loss to enforce depth consistent between objects and their ground contact points. Cheng et al. [61] replaced the CNNs with Cumulative Convolution to enlarge receptive fields of objects to the whole ground area below them.

In addition, some research has also attempted to adopt vanishing points clues in some of 3D vision task [28, 78, 100, 174]. For instance, Wang et al. [176] proposed to refine the surface normal prediction by fusing with vanishing points. Junsu et al. [83] designed the cross attention guided by vanishing points to solve imbalance issue in the monocular 3D semantic occupancy prediction. Hatem et al. [71] established vanishing point regression model, which are then combined with semantic segmentation results to predict depth according to the hand-crafted rules. Johannes et al. [58] incorporated vanishing points estimation into monocular visual odometry, allowing to solve the issue of scale shift.

Moreover, various other forms of geometric knowledge have been explored to enhance vision tasks. Naderi et al. [126] investigated the potential of constraining models by exploiting the geometric similarity between RGB images and corresponding depth maps, particularly along the edges of 3D scenes, which resulted in more precise depth estimation. Similarly, Genki et al. [85] addressed the issue of metric ambiguity in self-supervised monocular depth estimation by incorporating camera height as a constraint. These geometric knowledge plays the key role in various 3D vision tasks, which are promising to further improve monocular depth estimation in the further.

Building on prior research, the subsequent sections of this article will provide a comprehensive analysis of integrating ground geometry information with monocular depth estimation. We will systematically explore how ground geometry is employed across both dynamic and static scenes, examining the specific methodologies, experiments and constraints applicable to each scenario.

Chapter 3

Zero-shot Metric Depth Estimation for Static Scenes

3.1 Introduction

Surveillance video systems have become essential tools in various sectors, providing significant value in enhancing security, safety, and operational efficiency. One of the primary applications is in public safety, where surveillance cameras are used to monitor public spaces, detect criminal activities, and assist law enforcement agencies in responding to incidents in real-time. In addition to crime prevention, these systems help with crowd management by tracking people density [37, 43], ensuring compliance with safety regulations in high-traffic areas, and even monitoring social distancing in health-related scenarios. Surveillance video also plays a critical role in traffic management [6, 14], where cameras are used to monitor road conditions, detect accidents, and manage congestion. The Fig. 3.1 displays various monitoring scenarios involving traffic and crowd.

With the rapid development of artificial intelligence, intelligent surveillance systems represent a significant advancement in modern security technologies by utilizing multiple vision perceptual techniques, such as object detection [23, 52, 98, 99, 139], semantic segmentation [1, 7, 30, 108, 141, 188] and depth estimation [2, 54, 55, 138], to enhance the capabilities of traditional monitoring. Among them, depth estimation is significantly valuable because it allows surveillance cameras, which are typically monocular, to generate 3D representations of a scene from captured 2D images. With depth estimation, it becomes possible to detect and track objects more accurately [213, 218], assess crowd density [123], and even monitor social distancing with higher precision [3]. In addition, depth information aids in identifying objects' sizes and movements, improving the system's ability to detect anomalies, such as unauthorized access or potential hazards. Furthermore, the use of depth data can significantly enhance video analytics for tasks like collision detection, or object counting [37], making surveillance systems smarter and more responsive to real-world dynamics. Therefore, monocular depth estimation for monitoring scenes is highly required and valuable, they can offer a richer, more detailed understanding of monitored environments, improving security, safety, and operational efficiency.

However, estimating depth from a single image has long posed a significant challenge in computer vision due to the lack of triangulation cues available from multiple views, which introduces inherent scale ambiguity. However, unlike methods that rely on cumbersome and

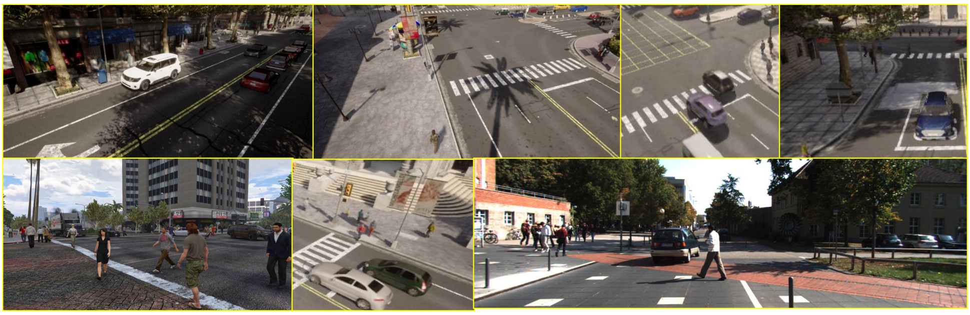


Figure 3.1: Multiple static surveillance scenes

costly active sensors, such as LiDAR [12], multiple, synchronized cameras [132] / a single, moving one [147], monocular approaches require only a single static camera which simplicity makes monocular depth estimation more practical and accessible for various monitoring environments.

The advent of deep learning has enabled the development of the first-ever solutions making it possible to face such a problem [90, 104, 138]. These advancements have been driven by the increasing availability of large-scale datasets annotated with depth labels [51, 96, 127], which are crucial for training, as well as the introduction of alternative self-supervised learning paradigms that replace explicit annotations with synchronized stereo image pairs [49, 54] or monocular video sequences [55, 185]. By leveraging Convolutional Neural Networks (CNNs) [88] and, more recently, Transformers [168], these methods are able to learn depth estimation from visual cues in the scene, such as shadows, perspective distortions, and vanishing lines. Notably, the sensitivity of these models to visual cues is evident from experiments where altering elements like the horizon height or simulating camera tilts relative to the ground plane can lead to significant changes in the estimated depth for the same scene. This demonstrates the crucial role that these visual cues play in inferring depth from monocular images.

Despite some data-driven monocular depth networks [81, 137, 138] have seen significant progress in cross-dataset generalization, these models still meet failures in specific settings that are under-represented in the training data, especially for some less-observed objects or images taken from a perspective rarely. The Fig. 3.2 shows some of outlier cases when employing state-of-art depth model DPT [137] to predict depth for these monitoring scenarios which are very common surveillance setting, with the camera positioned high over the ground and slanted with respect to it. However, as denoted by the red circles, DPT still yields inaccurate depth predictions, e.g. inconsistent depth on moving objects, missing objects in the far distances or infinite depth estimation. These abnormal predictions make scene depth estimation less reliable, thus affecting subsequent absolute depth estimation.

To overcome this issue, we propose an adaptation technique to attenuate prediction errors associated with agents. This methodology is grounded in the geometric assumption that all agents within the scene are in contact with the ground. Leveraging this assumption, we can derive a depth prior, i.e. the depth of each agent should be consistent with that of ground contact point. The process for achieving adaptive depth adjustment begins with the implementation of an instance segmentation network [187] to effectively mask each agent within the scene, e.g. vehicles and pedestrians. Subsequently, we replace the depth values of each agent with those corresponding to their ground contact points, thereby updating the estimated depth maps. These

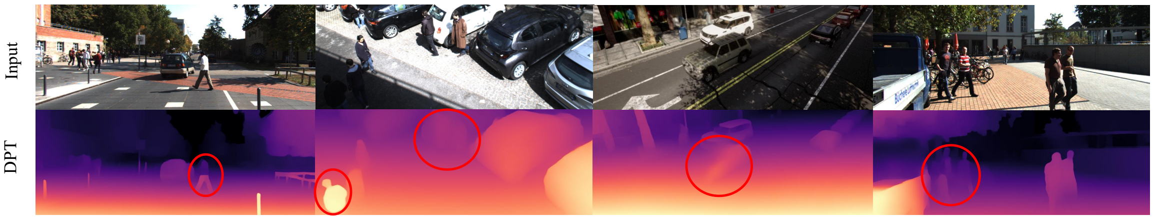


Figure 3.2: Failure cases of monocular depth estimation: all of depth maps are predicted by DPT [137] which is the state-of-art monocular depth network. The red circles emphasize wrong estimations, involving depth inconsistency, infinite values and et al.

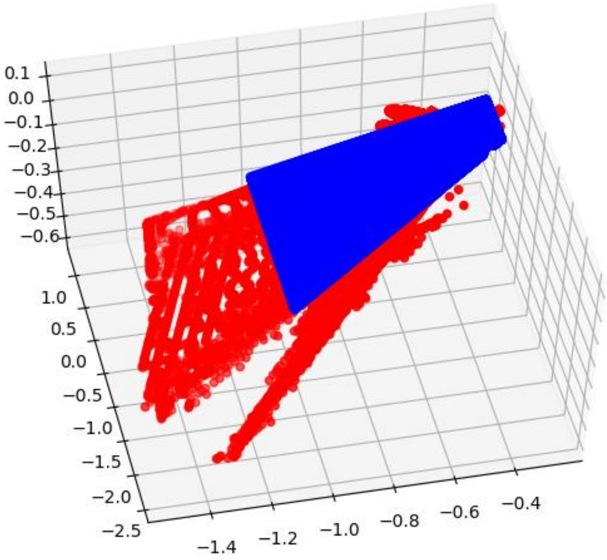


Figure 3.3: The 3D projections of predicted versus actual ground: The red point cloud indicates predicted ground plane while blue represents actual ground. There is a non-linear mapping between them

revised depth maps will serve as pseudo labels to fine-tune the DPT [137]. This adaptation technique allows the fine-tuned DPT to improve its accuracy on the agents at deployment time, without any additional depth labels or scale information.

Despite adaptation technique contributes significant improvement in the depth distribution of static scenes, particularly on the agents, the inherent scale ambiguities of monocular networks continues to result in substantial discrepancies between the predicted and actual scene depths. As illustrated in Fig 3.3, when the predicted ground depth and the actual ground depth are projected into 3D space, a clear gap emerges between the two planes, highlighting the misalignment in depth estimation. It implies that a simple scale factor is inadequate for aligning the two sets of point clouds. Consequently, without achieving proper scale alignment, accurately predicting absolute distances based on the outputs of monocular depth networks remains a challenging task.

Inspired by [183], we exploit the ground normal vector to restore the actual spatial scale. In specific, letting a person move around in the surveillance scene and recording his standing position coordinates and the corresponding height in the image, the actual ground normal can be estimated. Accordingly, when predicted ground normal and actual ground normal are both known, scale alignment is promising to achieve by rotating two planes.

Once the predicted depth is aligned with the actual scale, a straightforward scale factor can be applied to transform all relative depth values into accurate physical distances. To estimate this scale factor, we only have access to a simple prior related to the camera setup, i.e. the height of the camera over the ground. This prior allows us to recover the metric scale for depth maps generated by the monocular network.

To validate our proposal, we run experiments on a subset of the KITTI [51] dataset featuring static camera sequences, as well as two novel datasets composed of both synthetic frames (rendered through CARLA [38]) and real images. In the latter case, the dataset frames indoor and outdoor scenes. The final experimental results indicate our proposed method can effectively improve accuracy of metric depth compared to the baseline method DPT [137].

In conclusion, there are mainly two contributions in this work.

- A novel on-site adaptation framework for monocular depth estimation networks in fixed-camera setups, comprising three key components: (1) a lightweight fine-tuning procedure designed to correct depth estimation errors associated with agents, (2) a scene alignment step facilitated by leveraging the motion of agents to identify the ground plane, and (3) metric scale recovery achieved through the simple prior knowledge of the camera height relative to the ground. Notably, this approach does not require any complex camera calibration procedures or the use of expensive LiDAR depth sensors. As a result, it enables fast and efficient metric depth estimation in static monitoring scenarios. The simplicity and cost-effectiveness of this method make it highly suitable for practical applications.

- Two novel datasets with dense, ground-truth depth labels used to validate the effectiveness of our proposal.

3.2 Proposed Methods

Directly applying monocular depth estimation methods, such as those proposed in [137, 138], to real-world scenarios presents several significant challenges. First, these methods often produce incorrect or blurred depth predictions for dynamic objects within the scene, such as pedestrians, vehicles, or other moving entities. This limitation arises because monocular approaches typically rely on static scene assumptions, which fail to account for the motion of objects. Second, the scales inferred by these methods is frequently misaligned with the actual scene, leading to inaccuracies in depth estimation. This misalignment can result in distorted spatial representations, making it difficult to interpret the scene's true layout. Third, the depth predictions generated by monocular methods are inherently scale-ambiguous, meaning they are accurate only up to an unknown scale factor. This ambiguity complicates the task of estimating precise physical distances between objects, which is critical for applications such as autonomous navigation, robotics, and augmented reality.

To overcome these limitations, we introduce a comprehensive adaptation strategy designed to enhance the robustness and accuracy of monocular depth estimation in real-world environments. Our approach is structured into multiple steps, each addressing a specific challenge within the overall pipeline. As illustrated in Fig. 3.4, our strategy begins with a preprocessing stage that identifies and segments moving objects to minimize their impact on depth prediction.

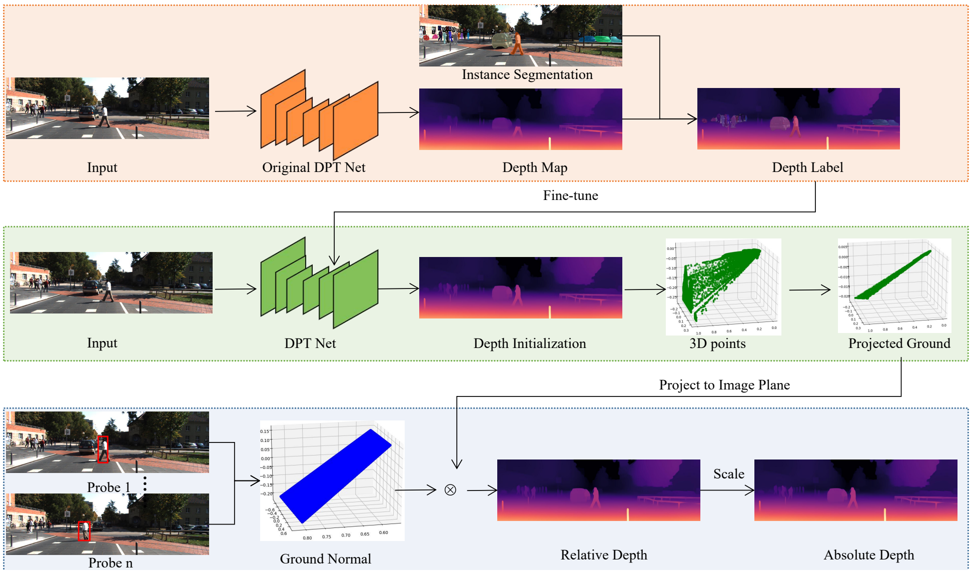


Figure 3.4: Overview of our adaptation scheme: First, (a) we rectify depth for agents in the scene by producing pseudo labels and running a lightweight fine-tuning of the original depth model; then, (b) the ground plane is extracted according to agents' motion, and used to align the overall structure of the depth maps predicted by the model. Eventually, (c) metric scale can be recovered from ground normal vectors by knowing camera height.

Next, we incorporate geometric alignment techniques to ensure that the inferred depth maps are consistent with the real-world scene structure. Finally, we introduce a scale recovery module that leverages additional scene priors to resolve the scale ambiguity, enabling precise distance measurements. By integrating these steps, our method significantly improves the reliability of monocular depth estimation, making it more suitable for practical applications in complex environments.

3.2.1 Lightweight fine-tuning with pseudo network

Although modern monocular depth estimation networks demonstrate strong generalization abilities [81, 138], they sometimes fail when used in ever-seen environments. This issue is particularly evident in fixed-camera installations where the camera viewpoint differs substantially from those seen during the network's training phase. In such cases, the network may miss the presence of agents in the scene, such as pedestrians or vehicles, leading to erroneous or inconsistent depth predictions. To address this limitation, we design a lightweight fine-tuning procedure to improve the perception of such agents by the monocular network by relying on pseudo labels obtained in two steps.

Pseudo labels initialization: We utilize the state-of-the-art monocular deep network, DPT [137], to generate an initial dense depth map. Subsequently, the semantic segmentation network, Detectron2 [187], is employed to perform instance segmentation on the surveillance images, allowing for the identification of possibly moving agents in the scene. Then, by assuming each agent is standing or moving over the ground plane, we generate pseudo labels by replacing the depth of each instance with the depth value of the lowest pixel in the instance itself – i.e., the

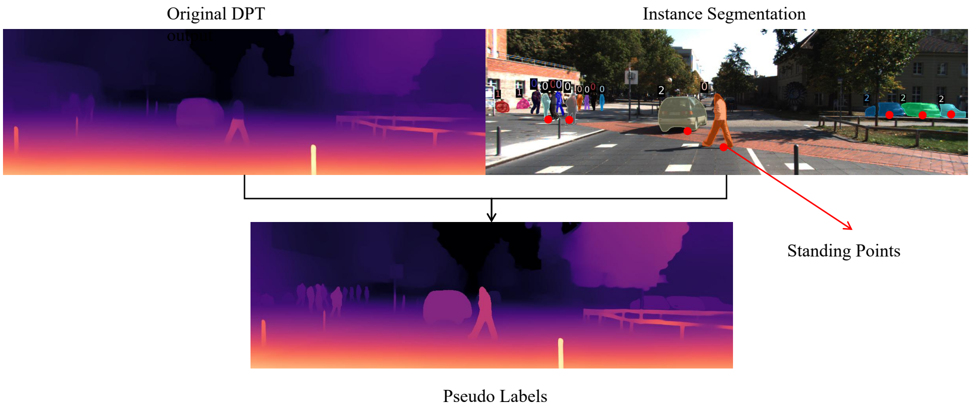


Figure 3.5: The process of pseudo labels initialization: Segment out possibly moving agents in the scene through Detectron2 [187] and then replace the depth of each instance with the depth value of the lowest pixel in the instance itself under assumption that each agent is standing or moving over the ground plane.

Table 3.1: The network fine-tuning parameters

Parameters	**Value**
Network name	dpt_large_384
Optimizer	Adam
Epoch number	20
Batch size	4
Learning Rate	10-5
Loss	MAE error

contact point with the ground. For complex agents, such as bicycles or motorcycles, we approximate the riders' depth to that of the vehicles. While for bags and hand-held items, e.g. umbrellas, the depth will match that of the closest pedestrian. The process of pseudo labels initialization is illustrated by Fig.3.5.

Agents rectification and fine-tuning: Although pseudo-depth labels can be generated using the aforementioned step, the depths of occluded agents may not correspond to their ground contact points, which can lead to inaccuracies in the pseudo label dataset. To ensure the precision of network regulation, we manually remove such unreliable labels, retaining a total of 4,516 pseudo labels in the training dataset. Since the range of pseudo-depth labels is consistent with the DPT predictions, no normalization of the depth labels is required during training. Instead, we directly apply the mean absolute loss between the network outputs and the pseudo labels. Details of the fine-tuning process are provided in Tab. 3.1. Upon completion of the fine-tuning of the DPT network [137], we can obtain more accurate initial dense depth predictions. Figure 3.6 qualitatively compares the results of the original DPT predictions with those of the fine-tuned DPT, demonstrating the effectiveness of our proposed lightweight fine-tuning technique.

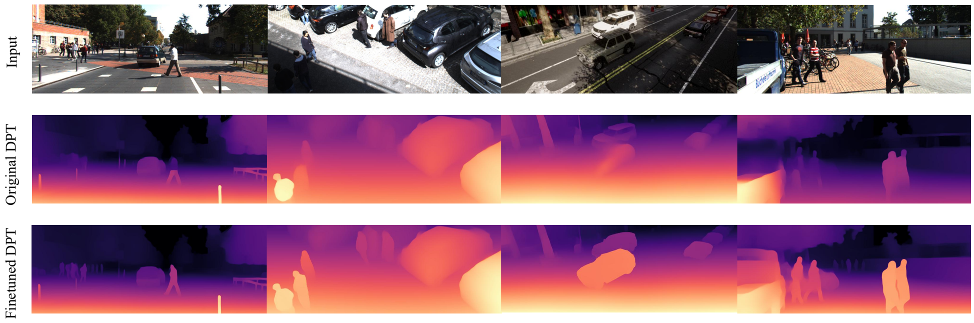


Figure 3.6: Monocular depth estimation before and after adaptation: On fixed-camera settings, state-of-the-art depth estimation models [137] might fail in unseen environments or camera settings. Our adaptation scheme allows us to improve their reliability.

3.2.2 Ground plane estimation and scene alignment

Even after adjusting the depth values for agents within the scene, there are still noticeable discrepancies between the reconstructed scene structures generated by the fine-tuned depth model and the actual physical environment. This misalignment likely arises from the significant differences in camera viewpoints between the training images and the current scene, which leads to a degradation in the network's ability to predict relative depths accurately and perform proper scale recovery when feasible. The problem is particularly pronounced in fixed-camera setups, where viewpoint variation is often limited during training but substantial in deployment. To resolve this issue, we aim to estimate the real ground plane within the sensed environment and incorporate it into the predicted depth map. As actual ground planes embody real scene scales, it is promising to restore geometric structure of the scene guided by the ground normal. By achieve this, we will perform ground normal estimation and scene alignment subsequently.

Ground plane estimation : In many monitoring environments, such as indoor and outdoor, it is common to present a ground surface in the field of view. If assuming the ground is a planar surface, it can be mathematically represented as Eq. 3.1 in 3D space, where (A, B, C) indicates the ground normal of plane, D is the distance from plane to the camera origin while (X, Y, Z) denotes the 3D coordinates of the points on the ground plane.

$$AX + BY + CZ = D \tag{3.1}$$

According to the perspective principle introduced in Section. 1.1.2, the 3D coordinates (X, Y, Z) of the plane are proportionally related with the projected image coordinates (x, y). Besides, the object height H in the 3D space and the projected height h on the image also have the same proportional coefficient. The proportional equation is formulated in Eq. 3.2 where f represents the camera focus length.

$$x = X\frac{f}{Z}, y = Y\frac{f}{Z}, h = H\frac{f}{Z} \tag{3.2}$$

By substituting Eq.3.2 into Eq.3.1, we get:

$$H(Ax + By + Cf) = Dh \tag{3.3}$$

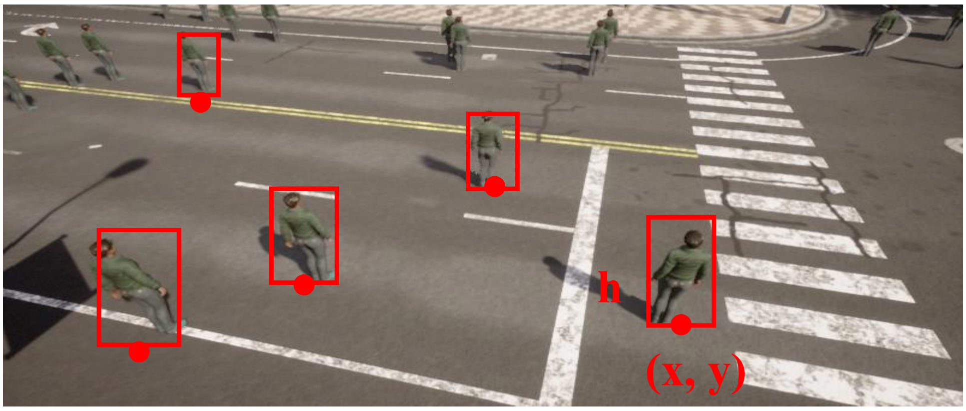


Figure 3.7: People probe for calculation of ground normal: Let a person stand on the different positions of ground and record corresponding image coordinates (x, y) and height h, the ground normal can be estimated by least square method.

The Eq. 3.3 indicates that there is a linear relationship between image coordinates (x, y) and ground plane parameters (A, B, C, D). If height H is assumed as a constant, the (A, B, C, D) can be estimated by solving linear equations when providing three or more sets of non-collinear points (x, y, h). Inspired by the work [183], we adopt people as probe to infer ground normal. In practice, by detecting a single agent in more than four frames, we record their corresponding pixel height h and standing point coordinates (x, y). Due to the sparsity of the standing points, we employ a relatively simple least squares method to compute coefficients (A, B, C, D), with the actual height H of the agent regarded as constant. For synthetic dataset, we can use built-in function to place the same agent at multiple locations at once and record their heights and coordinates as shown in Fig. 3.7. The ground coefficient calculated by the people probe can provide reliable scale information for subsequent spatial alignment.

Ground projection: As discussed in the beginning, there is generally a gap between the actual ground in real space and the ground represented by the predicted depth map. To effectively align the predicted scene with the real-world environment, it is essential to estimate not only the actual ground normal but also the predicted ground normal. Actually, when predicted depth map is given, the predicted ground normal is easy to known. Specifically, we project all ground points in the image onto 3D coordinates (X, Y, Z) based on the predicted depth values Z, then build the Eq. 3.4 to solve the ground normal (A', B', C', D') where A_g represents ground areas of scene. It is important to highlight that, in a monitoring scene, the background remains static over time, and the ground region consistently occupies a fixed area. Therefore, for any monitoring scene, we just manually define the ground area at the beginning of deployment, with no need for repeated delineation in subsequent frames, enabling accessible estimation for any scene.

$$X(A', B', C')^T = \begin{pmatrix} X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \\ \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n \end{pmatrix} \begin{pmatrix} A' \\ B' \\ C' \end{pmatrix} = D' \cdot I_{3\times1}, \quad (X, Y, Z) \in A_g \tag{3.4}$$

The ground normal is determined using the least squares method. Equation 3.5 presents the

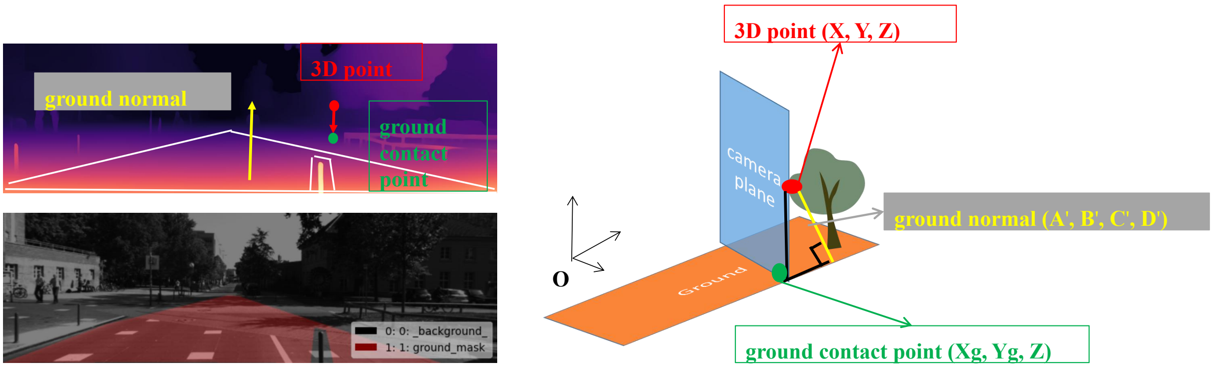


Figure 3.8: The illustration of ground projection: By manually identifying ground areas, the ground normal (denoted as yellow) can be estimated through predicted depth. Then all of 3D points (colored by red) will be projected onto the estimated plane along the camera plane, which are represented as (X_g, Y_g, Z)(green color).

matrix formulation for solving the ground normal, where 1_n represents a vector of ones with dimension n.

$$\begin{cases} (A', B', C')^T = (X^T X)^{-1} X^T 1_n \\ D' = \text{median}(X \times (A', B', C')^T) \end{cases} \tag{3.5}$$

After obtaining the predicted ground normal vector, we proceed to project all 3D points onto the predicted ground plane along the camera plane. This ground projection is carried out in such a way that each 3D point is moved along a line parallel to the camera plane, ensuring that the depths of the projected points remain consistent with their original depths in the predicted depth map. However, the spatial coordinates of these points are adjusted so that they lie on the predicted ground plane.

The primary objective of this ground projection process is to align the predicted 3D scene structure with the actual physical environment. By projecting all points onto the predicted plane, their spatial positions will be adjusted to lie on the same plane, while the depth values are remained. In this way, it allows for accurate scene alignment just by aligning the projection plane with actual ground plane, ensuring scale consistency between the predicted and real-world scenes. The Fig. 3.8 displays the visualization of how the 3D points (X, Y, Z) are projected on the corresponding ground points (X_g, Y_g, Z).

To acquire the projected ground coordinates (X_g, Y_g, Z) for each point, we should first establish two geometric priors: 1) all of projected ground points satisfy plane formula with ground normal (A', B', C', D'). 2) the vector connecting the 3D point and the ground point is orthogonal to both optical axis and ground plane. For the second condition, given the optical axis of camera (0, 0, 1) and ground normal (A', B', C'), the intersection vector of the camera plane and the ground can be denoted as (-B', A', 0) which is orthogonal to projected vector $(X - X_g, Y - Y_g, 0)$. Accordingly, these two geometric priors can be formulated as Eq. 3.6.

$$\begin{cases} -B'(X - X_g) + A(Y - Y_g) = 0 \\ A'X_g + B'Y_g + C'Z = D' \end{cases} \tag{3.6}$$

Through simplifying the above formula, we can get the mathematical functions about X_g, Y_g, Z_g

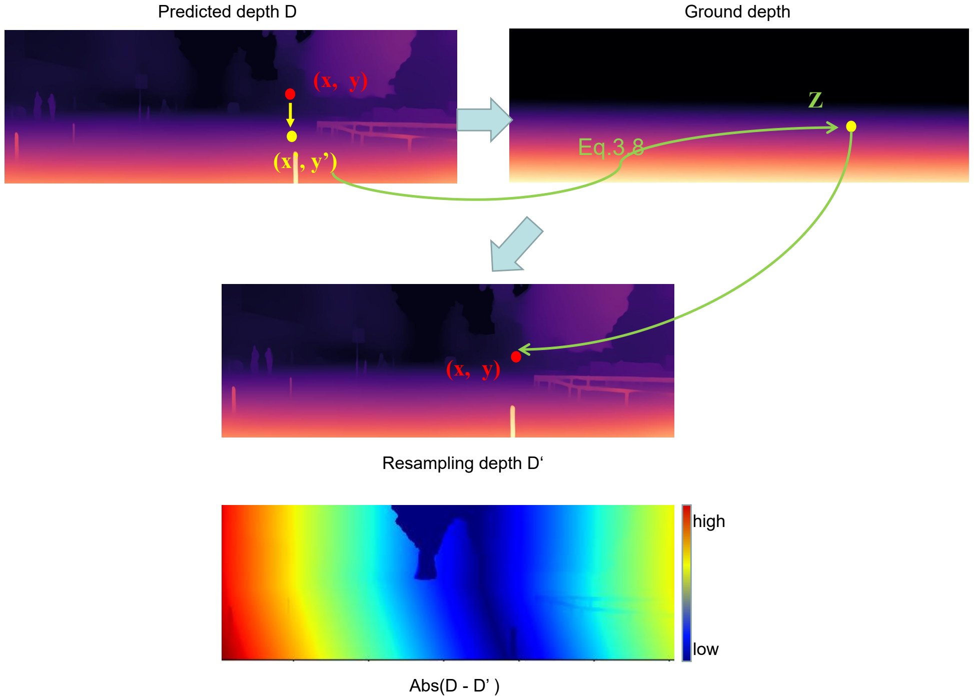


Figure 3.9: The ground depth resampling: For the point (x, y), it will be projected on the corresponding ground location (x', y') through Eq. 3.7. By sampling on the ground depth for (x', y'), the predicted depth will be updated to be aligned with actual scene. The absolute difference map between DPT depth prediction and resampling depth is shown at the bottom picture.

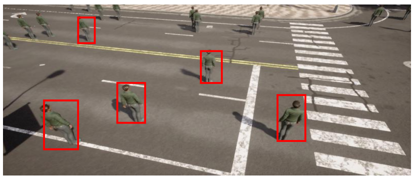
as follows. Based on the Eq. 3.7, all of 3D points can be projected onto the predicted ground plane.

$$\begin{cases} X_g = \frac{A'D' - A'C'Z - A'B'Y + B'^2X}{A'^2 + B'^2} \\ Y_g = \frac{B'D' - B'C'Z - A'B'X + A'^2Y}{A'^2 + B'^2} \\ Z_g = Z \end{cases} \tag{3.7}$$

Ground depth resampling: the crucial step to achieve scene alignment is to make depths of ground contact points consistent with actual values. To accomplish this, the initial step is to project the ground points back to the image plane. This operation provides 2D coordinates (x', y') that represent the ground contact point for each 3D point. By substituting the right-hand side of Eq. 3.3 with the term associated with depth Z, we can derive Eq. 3.8, which provides a practical formulation for estimating the depth Z of each ground point using the actual ground normal vector (A, B, C, D). The original depth will be updated by re-sampling on the ground depth. The overall process of ground depth resampling is depicted in Fig. 3.9, which illustrates the scale of predicted depth has been changed through ground resampling operation.

$$Ax' + By' + Cf = \frac{Df}{Z} \tag{3.8}$$

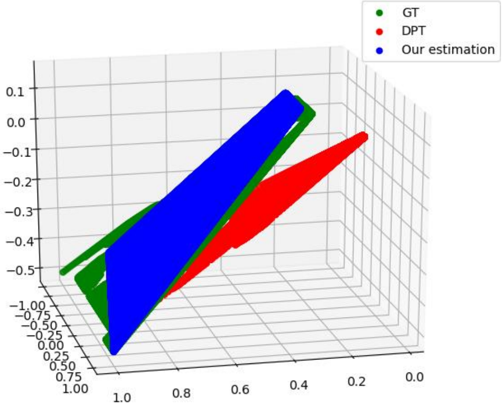
To further visualize the effectiveness of scene alignment, we take the scene on the left of



(a) Plane Estimation(people probes)

Ground normal	a	b	c
GT plane	-0.0002	0.7752	0.6316
DPT normal	-0.0041	0.8910	0.4540
Estimated normal	-0.0266	0.7321	0.6807

(b) Plane normal comparison



(c) Visulization of ground normal

Figure 3.10: Structural misalignment between predicted depth and real scene: For a single scene with pedestrians walking around (a), we report ground normals (a,b,c) obtained from predicted depth (red), our ground plane estimation method (blue), and ground truth (green). On the right, we visualize the misalignment between predicted and real planes and better alignment for plane after performing our proposed scene alignment technique.

Fig. 3.10 as an example. By projecting ground area to the 3D space based on different depths, the green plane in the right figure presents the actual ground plane, the red plane indicates predicted ground from original DPT [137] depth estimation while blue plane reflects adjusted ground through our proposed scene alignment technique. It can clearly demonstrate the plane after scene alignment is closer to the real scene. The Table(b) also prove that, compared to the ground normal predicted by DPT [137], the aligned ground normal exhibits a smaller angle relative to the actual ground normal.

In summary, scene alignment involves ground normal estimation, ground projection and ground depth resampling. All of them only depend on the predicted depth map and actual ground normal which is calculated through people probe, allowing for a practical and feasible way to mitigate scale misalignment between predicted 3D scenes and actual scenes.

3.2.3 Absolute scale recovery

Scale alignment makes the predicted 3D scenes have an approximate linear relationship with the real scenes, making it possible to transform relative depths to absolute depths just by a scale factor. To estimate this scale factor, it is necessary to acquire the corresponding metric depth at any position of depth map. However, in static environments where only a monocular camera is deployed—without the assistance of LiDAR or other physical depth sensing devices—acquiring this scale factor becomes challenging. To achieve true zero-shot metric estimation, we estimate the minimum absolute depth of the scene which is typically located at the middle-bottom position of the image. Actually, previous research [118, 189] have tried to estimate missing scale factor by utilizing camera height over the ground. Unfortunately, they rely on the assumption that the Z-axis of the camera is roughly parallel to the ground plane, a condition not always met in practice.

To deal with arbitrarily oriented cameras, we design a custom method to estimate minimum

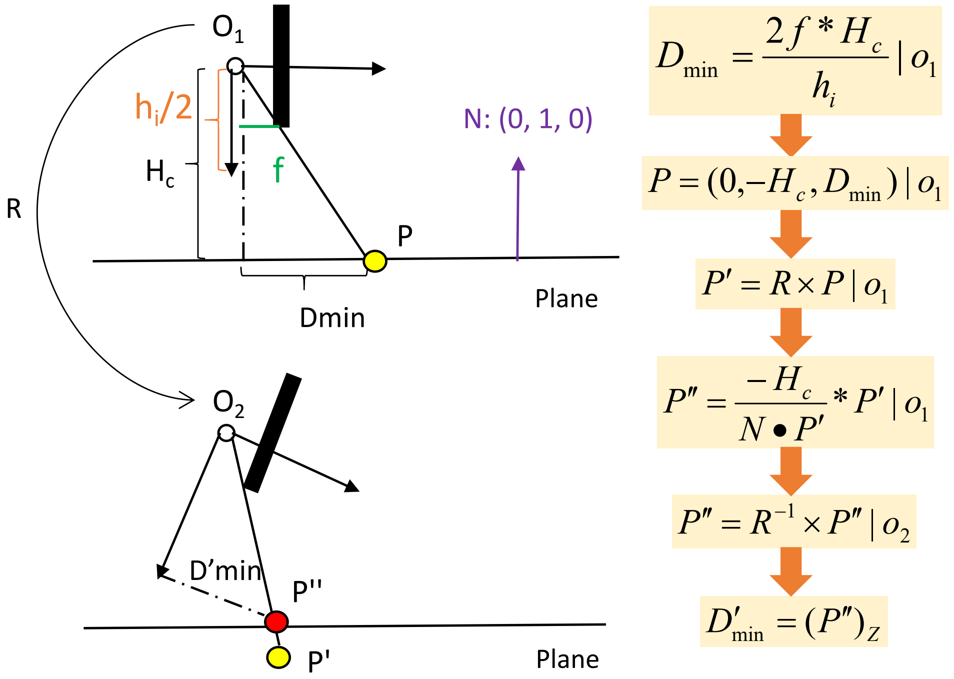

O1
hi/2
Hc
f
R
N: (0, 1, 0)
P
Dmin
Plane
O2
D'min
P''
P'
Plane
$D_{\min} = \frac{2f * H_c}{h_i} | o_1$
$P = (0, -H_c, D_{\min}) | o_1$
$P' = R \times P | o_1$
$P'' = \frac{-H_c}{N \bullet P'} * P' | o_1$
$P'' = R^{-1} \times P'' | o_2$
$D'_{\min} = (P'')_Z$

Figure 3.11: Minimum depth estimation: when the camera is positioned at pose O_1, the minimum depth D_{min} is determined just by camera height and focus length. Then the more common pose O_2 can be obtained by multiplying by rotation matrix which is estimated through current ground normal and N(0, 1, 0). With the rotation matrix, the coordinate of point P" in the O_2 system can be calculated, allowing to infer the minimum depth for pose O_2 according to the projection of P" on the Z-axis. The whole mathematical transformation process is listed on the right side of image.

depth of the image, relying solely on the known camera height. Before addressing more general cases, we initially assume that the camera's optical axis is parallel to the ground plane, as illustrated in the coordinate system O_1 of Fig. 3.11. This scenario represents a special case discussed in previous works [118, 189]. In the O_1 system, point P on the ground plane is projected onto point p', which is located at the center-bottom of the image. Since the camera height H_c, the image height $h_i/2$, the focus length f and the depth D_{min} of point P constitute two similar triangles, D_{min} can be computed according to Eq. 3.9. In the O_1 system, the coordinate of point P can be written as $(0, -H_c, D_{min})$.

$$D_{min} = \frac{2fH_c}{h_i} \tag{3.9}$$

Then by rotating the camera coordinate system from O_1 to O_2 (as shown in the lower-left corner of Fig. 3.11), we can acquire the most common camera pose in the monitoring environments. In the O_1 system, the ground normal will be viewed as (0, 1, 0), whereas in the O_2 system, the ground normal transforms to (A, B, C) which can be estimated through people probes introduced in the Section 3.2.2. Therefore, the rotation matrix between coordinate systems O_1 and O_2 can be derived by aligning the vectors $(0, 1, 0)$ and (A, B, C), representing the ground normals in the respective systems. In specific, given rotation angle θ and rotated axis n between two vectors, the rotation matrix can be established according to the Rodrigues' rotation formula [56] as below:

$$R = \cos\theta I + (1 - \cos\theta)nn^T + \sin\theta[n]_\times \tag{3.10}$$

In the process of rotating normal vectors, the rotation angle θ is given by $\theta = \arccos\frac{B}{\sqrt{A^2+B^2+C^2}}$, and the rotation axis n is computed as the normalized cross product of the two vectors, mathematically expressed as $(-\frac{C}{A^2+C^2}, 0, \frac{A}{A^2+C^2})$. The skew-symmetric matrix representation of n, denoted as $[n]_\times$, is defined as follows:

$$[n]_\times = \begin{bmatrix} 0 & \frac{-A}{A^2+C^2} & 0 \\ \frac{A}{A^2+C^2} & 0 & \frac{C}{A^2+C^2} \\ 0 & \frac{-C}{A^2+C^2} & 0 \end{bmatrix} \tag{3.11}$$

Once the rotation matrix R is obtained, the point P can be transformed to a new position P' through matrix multiplication $R \times (0, -H_c, D_{min}^T)$, where H_c is the camera height and D_{min} represents the minimum depth. Importantly, the new coordinates of P' are still in the coordinate system O_1. However, as illustrated in Fig. 3.10, the point P' will not lie on the ground plane after this rotation. Thus, it is necessary to calculate the intersection point P" between the ground plane and the ray extending from the new camera position O_2 through P'. The coordinate of P" can be determined with the use of assumed ground normal N(0, 1, 0), as formulated in Eq. 3.12 where the P" still takes O_1 as coordinate system.

$$P'' = \frac{-H_c P'}{N \bullet P'} \tag{3.12}$$

To calculate minimum depth in the O_2 system, the coordinate of P" should be transformed to O_2 system by multiplying inverse rotation matrix R^{-1}. Finally, the projected minimum depth under O_2 camera coordinate system will be acquired by retrieving projection of P" on the Z-axis. The overall process of transformation is demonstrated in Fig. 3.11.

Once the minimum depth within the scene has been estimated—a value that typically corresponds to the bottom-center region of the image, as this area often represents the closest point to the camera in many real-world scenarios—the scale factor can be derived. This is achieved by dividing the measured actual minimum depth by the predicted relative depth at the bottom-center position. Since only one absolute depth value can be obtained through the camera height, the conversion from relative depth to metric depth utilizes only a scale factor without considering the offset. The resulting scale factor serves as a critical calibration parameter, enabling the conversion of all relative depth values across the entire image into precise physical distances. This transformation process effectively bridges the gap between relative depth predictions and metric depth estimation, facilitating zero-shot metric depth estimation without the need for additional training or fine-tuning on the target dataset. By leveraging this approach, the system can generalize to new environments and provide accurate depth measurements, even in the absence of prior knowledge about the scene's scale.

3.3 Experimental Results

In this section, we validate the effectiveness of our proposal. We first introduce datasets, metrics and implementation details involved in our evaluation. Then, the effectiveness of the proposed lightweight fine-tuning, scene alignment and absolute scale recovery techniques in enhancing the depth estimation results will be validated through ablation experiments respectively. Besides, we demonstrates the advancements through a qualitative comparison between the predictions generated by the original DPT [137] model and those obtained after applying our proposal. Finally, we compare MonoDepth2 trained with stereo self-supervision and our method on metric depth estimation, proving our proposed monocular method yield close performance with stereo methods in metric recovery.

3.3.1 Datasets

We run our experiments on a mixture of synthetic and real datasets, with a static camera mounted over the scene pointing toward roads, sidewalks, or pedestrian areas. There are a total of seven sequences used– Fig. 3.12 shows an example for each. They are grouped into three categories, i.e. Carla static sequences(C1, C2, C3), KITTI static sequences(K1, K2) and Real sequences(R1, R2).

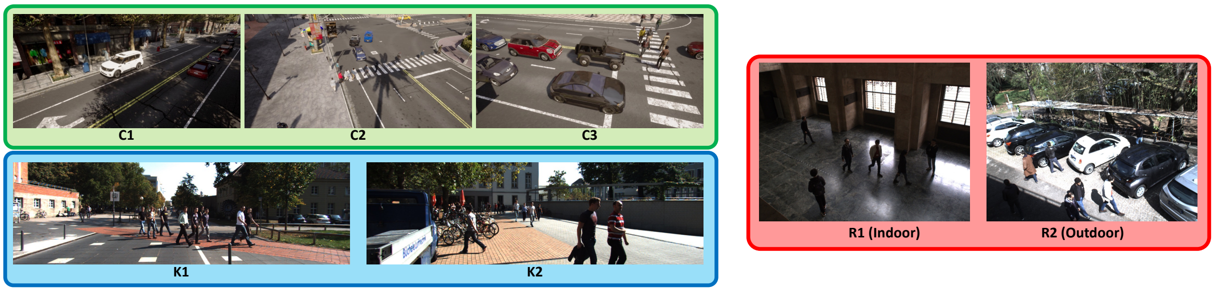


Figure 3.12: Evaluation dataset: We show a sample for each of the seven scenes from CARLA (green), KITTI (blue), or our acquisitions (red) used in our experiments.

Synthetic data (CARLA): In order to obtain video sequences about outdoor vehicle monitoring, we employ the Carla simulator [38] to generate the synthetic dataset.

The CARLA simulator [38] is a game simulator designed to render the real autonomous driving environments. It allows to generate a variety of complex urban and suburban scenes, and users can flexibly customize elements in the simulation environment, such as road layout, traffic signals, buildings, weather conditions (sunny, rainy, foggy, etc.) and lighting changes (such as day and night). Moreover, CARLA provides the built-in sensors to retrieve data for autonomous vehicles. In our experiments, we mainly leverage RGB camera, depth camera and semantic segmentation camera. The following codes introduce how to install and listen these cameras step-by-step by built-in functions of CARLA.

```
import carla
# Deploy the virtual environment
client = carla.Client('localhost', 2000)
world = client.get_world()
spectator = world.get_spectator()
# Get the view of spectator
transform = spectator.get_transform()
# Acquire built-in library containing a variety of sensors
blueprint_library = world.get_blueprint_library()
```

```
# Setting cameras
# RGB camera
camera_rgb = blueprint_library.find(’sensor.camera.rgb’)
camera_rgb_l.set_attribute(’image_size_x’, str(img_w))
camera_rgb_l.set_attribute(’image_size_y’, str(img_h))
# Semantic segmentaion camera
camera_semseg = blueprint_library.find(’sensor.camera.semantic_segmentation
    ’)
camera_semseg.set_attribute(’image_size_x’, str(img_w))
camera_semseg.set_attribute(’image_size_y’, str(img_h))
# Depth camera
camera_depth = blueprint_library.find(’sensor.camera.depth’)
camera_depth.set_attribute(’image_size_x’, str(img_w))
camera_depth.set_attribute(’image_size_y’, str(img_h))

# Spawning cameras at spectator view
camera_rgb = world.spawn_actor(camera_rgb, transform)
amera_semseg = world.spawn_actor(camera_semseg, transform)
camera_depth = world.spawn_actor(camera_depth, transform)

# Listening
# retrieve_data_X will be called each time a new image is generated by the
# camera
camera_rgb.listen(lambda data: retrieve_data_rgb(data))
camera_semseg.listen(lambda data: retrieve_data_semseg(data))
camera_depth.listen(lambda data: retrieve_data_depth(data))
```

By listening for camera events over a period of time, we can obtain video sequences for different scenes to simulate a realistic traffic environment, where pedestrians and vehicles move around, and urban infrastructures are standing, such as trees and buildings. These data not only contain RGB images, and also the corresponding depth and semantic segmentation maps. In this way, we generate three sequences, each consisting of 800 frames at 800 × 400 resolution, dubbed C1, C2, and C3.

KITTI static sequences: The KITTI [51] dataset is a widely used autonomous driving dataset. Among the many samples provided by the KITTI raw dataset, a small amount of short, static sequences, mainly concentrated in the Campus category, are suitable for our experiments. We obtain two main sequences by grouping frames from *2011_09_28_drive_0016* + *2011_09_28_drive_0021* and *2011_09_28_drive_0039* + *2011_09_28_drive_0043*, dubbed K1 and K2, and counting 395 and 506 samples.

Real sequences(R1, R2): To further stress the flexibility of our approach, we collect two real sequences in indoor and outdoor environments, dubbed R1 and R2, counting 3024 and 2952 images. For both, we mounted a camera tilted toward the ground plane at about five meters over it. We collect images with a 27cm baseline stereo camera and use CREStereo [94] to estimate disparity maps and triangulate them into depth to obtain ground truth labels. Although imperfect, we consider these annotations accurate enough for our purposes.

The Tab. 3.2 lists the details of all the data sequences, including image resolution and numbers.

Table 3.2: The details of experimental datasets

Data name	Sequence	Number	Resolution
CARLA Data	C1	800	800×400
	C2	800	800×400
	C3	800	800×400
KITTI Data	K1	395	1224×370
	K2	506	1224×370
Real Data	R1	3027	1024×768
	R2	2952	1024×768

3.3.2 Implementation details and metrics

To implement DPT [137] as the baseline monocular depth estimation network in our experiments, all evaluations were conducted on a single NVIDIA 3090 GPU. The network was adapted to each of the seven sequences using the previously detailed pipeline, which involves lightweight fine-tuning and scene alignment. For the fine-tuning process, we selected the first 342, 463, and 406 frames from three synthetic sequences, the first 273 and 107 frames from the KITTI sequences, and the first 684 and 679 frames from real-world scenes. Pseudo depth labels were generated for each of these frames, and DPT was fine-tuned over 20 epochs. This simulates an on-site adaptation process in which the first frames captured after the camera installation are used for initial fine-tuning.

Following the fine-tuning phase, we evaluated the effectiveness of our proposed pipeline on the remaining frames of each sequence, where we also applied test-time scene alignment to further enhance the depth prediction accuracy. For a comprehensive assessment, we employed several widely-used metrics in the monocular depth estimation field, including Scale-invariant Logarithmic Error (SiLog), Root Mean Squared Error (RMSE), Absolute Relative Error (Abs Rel), and Squared Relative Error (Sq Rel). The evaluation was performed under two scenarios: one using ground truth depth to rescale the predictions as a baseline [137], and the other employing our novel scale restoration technique to recover absolute depth without reliance on external depth sensors. This dual evaluation approach allows us to demonstrate the robustness and accuracy of our method in recovering metric-scale depth information in real-world monitoring scenes.

3.3.3 Experimental evaluation of on-site adaptation

We start our evaluation by examining the effectiveness of the first two components of our pipeline: lightweight fine-tuning and scene alignment. Table 3.3 displays the performance of the original DPT model [137] and results obtained after the application of these techniques. To avoid influences of estimated scale factor, we use median rescaling [54], a widely used method for converting relative depth predictions into metric values, to fairly compare their results in this experiment.

For the model fine-tuned using pseudo labels without scene alignment technique, we denote it as DPT-ft, the goal of it is to enhance depth prediction accuracy by addressing errors that primarily occur in regions where moving agents. From the Table 3.3, we can notice the lightweight fine-tuning can effectively improve the accuracy compared to the original model.

In addition to fine-tuning, we evaluate the impact of applying scene alignment, referred to as DPT-align. Scene alignment corrects the overall scene structure by adjusting the depth predictions to align with the physical environment, specifically through ground plane estimation. As shown in Table 3.3, applying alignment alone leads to a more substantial reduction in overall depth error compared to fine-tuning, particularly in scenes where the ground plane dominates the frame, as it ensures that predicted depth maps better reflect the true geometry of the environment.

By combining both techniques—fine-tuning and scene alignment, we can acquire the best results, denoted as DPT-ft-align. This model demonstrates consistent improvement across the majority of sequences. The fine-tuning process corrects errors related to moving agents, while scene alignment ensures that the overall scene structure is coherent with the real-world layout. The combined model shows a significant reduction in errors across most test cases, confirming that these two methods complement each other in terms of handling dynamic agents and static background elements simultaneously.

However, it is worth noting that in specific sequences, such as C1 and R1, where the ground plane occupies a large portion of the scene, the impact of fine-tuning is less pronounced. In these cases, the ground plane dominates the pixel count, meaning that alignment has a more substantial influence on the overall depth accuracy. The alignment step, by correcting the ground plane, plays a more significant role than fine-tuning, which mainly addresses the smaller, moving components of the scene.

3.3.4 Experimental evaluation of absolute scale recovery

In this experiment, we assess the effectiveness of our scale recovery strategy, comparing it with alternative methods that also utilize camera height but make the assumption that the camera is parallel to the ground plane [118, 189]. To carry out this evaluation, we compute the depth value of anchor point(middle-bottom point of the image) for each sequence using different methods and compare the predicted depths to the corresponding ground truth values. The results are presented in Table 3.4, which includes both the estimated depths and the associated errors relative to the ground truth.

For the KITTI dataset, where ground truth is sparse, we replace the anchor point with the closest available pixel that has a similar ground truth depth. The results show that our approach consistently restores the absolute scale more accurately than the alternative techniques. While the two methods perform similarly on the KITTI dataset—where the camera's optical axis is almost parallel to the ground plane—the advantages of our approach become more apparent when applied to real-world datasets that feature a significant camera tilt. In these scenarios, our method delivers superior accuracy by correctly accounting for the camera's orientation and restoring the true metric scale.

This experiment demonstrates the robustness of our scale recovery method, particularly in challenging cases where the camera's optical axis is not parallel to the ground, thereby validating its applicability in diverse monitoring environments.

3.3.5 Metric depth comparison with stereo method

We conducted a comprehensive comparison between the depth maps predicted by the DPT model [137] after applying our proposed scale recovery technique and those generated by a

Scene	Method	SiLog↓	RMSE↓	Abs rel↓	Sq rel↓
C1	DPT [137]	0.051	4.098	0.184	0.707
	DPT-ft	**0.044**	**4.012**	**0.171**	**0.656**
	DPT-align	**0.018**	**3.421**	**0.103**	**0.399**
	DPT-ft-align	**0.026**	**3.677**	**0.131**	**0.512**
C2	DPT [137]	0.061	5.164	0.221	1.183
	DPT-ft	**0.029**	**3.476**	**0.144**	**0.591**
	DPT-align	**0.011**	**2.301**	**0.041**	**0.307**
	DPT-ft-align	**0.009**	**2.201**	**0.036**	**0.288**
C3	DPT [137]	0.056	2.355	0.214	0.521
	DPT-ft	**0.042**	**2.052**	**0.183**	**0.391**
	DPT-align	**0.018**	**1.282**	**0.112**	**0.154**
	DPT-ft-align	**0.014**	**1.103**	**0.098**	**0.115**
K1	DPT [137]	0.069	5.705	0.219	1.274
	DPT-ft	**0.024**	**3.824**	**0.121**	**0.541**
	DPT-align	**0.026**	**4.537**	**0.105**	**0.767**
	DPT-ft-align	**0.019**	**3.666**	**0.101**	**0.483**
K2	DPT [137]	0.149	4.718	0.337	2.289
	DPT-ft	**0.048**	**2.948**	**0.158**	**0.658**
	DPT-align	**0.057**	**3.686**	**0.235**	**1.441**
	DPT-ft-align	**0.045**	**2.876**	**0.152**	**0.552**
R1 (Indoor)	DPT [137]	0.052	1.626	0.151	0.546
	DPT-ft	**0.049**	**1.579**	**0.148**	**0.507**
	DPT-align	**0.033**	**0.999**	**0.098**	**0.293**
	DPT-ft-align	**0.036**	**1.075**	**0.107**	**0.337**
R2 (Outdoor)	DPT [137]	0.054	3.786	0.198	0.748
	DPT-ft	**0.052**	**3.654**	**0.189**	**0.723**
	DPT-align	**0.051**	**3.604**	**0.167**	**0.671**
	DPT-ft-align	**0.041**	**3.136**	**0.159**	**0.542**

Table 3.3: Quantitative results – on-site adaptation. We report error metrics on the seven sequences, for original DPT [137], DPT after lightweight fine-tuning (DPT-ft) and with test-time scene alignment (DPT-ft-align). We highlight **first**, **second**, and **third** best results.

model that was trained on-site with direct supervision from a stereo camera. In this stereo-supervised setup, the model has access to the true metric scale of the scene, which provides an upper bound on the performance a monocular depth estimation network could theoretically achieve when given perfect scale information during training. For this experiment, we utilized the widely adopted MonoDepth2 model [55] as the baseline for stereo-supervised learning. The results, summarized in Table 3.5, show that our improved DPT model, once fine-tuned and aligned with scale recovery (DPT-ft-align), consistently performs close to the accuracy of MonoDepth2 in terms of depth estimation. In some cases, DPT-ft-align even surpasses the performance of MonoDepth2, underscoring the strength of our scale recovery approach. This demonstrates that by accurately recovering the metric scale, our method allows a monocular network to achieve performance levels comparable to a stereo-based approach, effectively narrowing the gap between unsupervised monocular depth estimation and supervised stereo-based models. These findings provide strong evidence for the effectiveness of our method in real-world applications where accurate metric depth is critical.

3.3.6 Qualitative results

To conclude, Fig. 3.13 shows how our whole frame work dramatically reduces the error being the primary source of failure– i.e., in the presence of agents such as pedestrians and cars, or in

		C1	C2	C3	K1	K2	R1	R2
Ground truth	Depth	6.281	9.799	5.256	5.992	5.895	6.125	6.675
[118, 189]	Depth	10.751	19.431	10.686	6.439	6.439	24.065	25.651
	Error (m)	4.470	9.632	5.430	0.447	**0.544**	17.940	18.976
Ours	Depth	6.102	9.789	5.105	5.772	5.285	6.304	6.421
	Error (m)	**0.179**	**0.010**	**0.151**	**0.220**	0.610	**0.179**	**0.254**

Table 3.4: Scale recovery evaluation – anchor point. From top to bottom: ground truth depth for the anchor point, average depth and its error according to [118, 189] and our method.

Scene	Method	Rescale	SiLog	RMSE	Abs rel	Sq rel
C1	Monodepth2 (S) [55]	Stereo	**0.014**	**4.376**	**0.065**	**0.304**
	DPT-ft-align	Ours	0.038	5.628	0.124	0.727
C2	Monodepth2 (S) [55]	Stereo	**0.011**	**8.231**	**0.051**	**0.383**
	DPT-ft-align	Ours	0.021	8.426	0.096	0.651
C3	Monodepth2 (S) [55]	Stereo	0.031	2.653	0.077	0.256
	DPT-ft-align	Ours	**0.012**	**2.101**	**0.068**	**0.095**
K1	Monodepth2 (S) [55]	Stereo	**0.005**	**2.307**	**0.042**	**0.163**
	DPT-ft-align	Ours	0.019	3.901	0.085	0.504
K2	Monodepth2 (S) [55]	Stereo	0.081	3.181	0.116	0.511
	DPT-ft-align	Ours	**0.042**	**3.026**	**0.139**	**0.553**
R1 (Indoor)	Monodepth2 (S) [55]	Stereo	0.041	1.233	0.101	0.381
	DPT-ft-align	Ours	**0.028**	**1.102**	**0.106**	**0.265**
R2 (Outdoor)	Monodepth2 (S) [55]	Stereo	**0.032**	**3.112**	**0.104**	**0.461**
	DPT-ft-align	Ours	0.081	5.076	0.141	1.112

Table 3.5: Metric depth evaluation – comparison with stereo self-supervision. We report error metrics by MonoDepth2 trained with stereo self-supervision and our method.

the farthest parts of the scene, where the ground plane misalignment between predicted and real depth becomes more prominent. After processing, the error remains slightly higher on objects farther from the ground plane– e.g., structures on the sidewalk (left) or the obstacle in the very foreground (center)– where no optimization is performed by our method.

3.4 Conclusion

In this work, we introduce an innovative pipeline aimed at facilitating the on-site adaptation of a monocular depth estimation network, specifically tailored for applications involving fixed-camera installations. Our proposed method incorporates a lightweight fine-tuning process, enabling the model to be adapted to specific environments effectively. Additionally, it employs test-time scene alignment of the predicted depth maps by utilizing the presence of freely moving agents within the scene. Furthermore, our approach successfully recovers the metric scale of the environment with minimal information, requiring only knowledge of the camera's mounting height. Experimental results demonstrate that a limited number of images collected immediately following deployment can significantly enhance the performance of the DPT network, validating the efficacy of our solution.

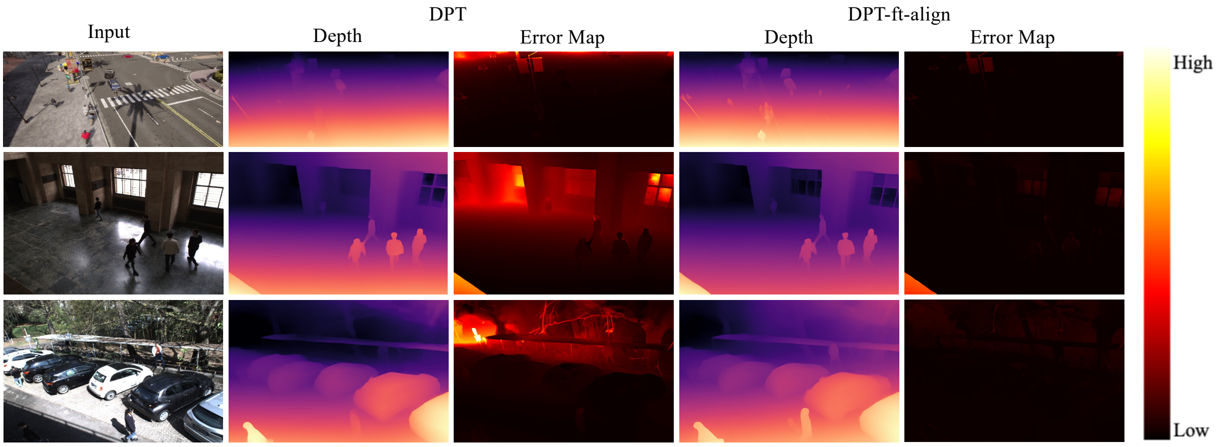


Figure 3.13: Qualitative results of error maps. : We show error maps by DPT and DPT-ft-align.

Chapter 4

Self-supervised Depth Estimation for Moving Objects with Ground Propagation

4.1 Introduction

Monocular depth estimation has become a crucial task in computer vision due to its abilities to reconstruct 3D structures from a single image. The challenge of monocular depth estimation lies in recovering depth information without scale information, making this problem inherently ill-posed. Recently, with advances in deep learning, significant progress has been made in this area. For example, neural networks, particularly convolutional [54, 138] and transformer-based [137, 226] models, have demonstrated remarkable success in accurately predicting depth from single images. Some data-driven approaches [81, 196, 197] have set new benchmarks in terms of accuracy and generalization, utilizing large-scale datasets and diverse forms of supervision to train models that can infer depth from visual cues alone.

Traditionally, these methods rely on supervised learning techniques, where models are trained on using ground-truth depth labels obtained from multi-view stereo cameras [94, 216], LiDAR sensors [12], or synthetic datasets [22, 38]. The process to collect these depth labels is both time-consuming and expensive, as it often requires specialized equipment and complex camera calibration procedures. Moreover, the training scenes are usually captured by monocular cameras with different intrinsic parameters, resulting in depth estimation ambiguity. For instance, as illustrated in Fig. 4.1, two chairs in the images have visually similar appearances, suggesting they would be assigned the same depth value during monocular depth estimation. However, they actually have distinct depths, as these images are taken by cameras with varying focal lengths. This depth ambiguity arising from inconsistent camera focal lengths significantly constrains the accuracy of the predictions. Although there have been some attempts [60, 93, 130] to incorporate camera intrinsic parameters into the monocular depth estimation to enhance precision, existing training datasets cannot involve all real-world scenarios, making it still challenging to recover accurate scale information for unseen environments.

To address the limitations associated with supervised learning, self-supervised monocular depth estimation [54, 55, 185, 220] has emerged as as a promising alternative. This approach eliminates the necessity for explicit depth labels but only depends on arbitrary video sequences by leveraging a training loss based on image reprojection errors. Specifically, self-supervised methods employ view synthesis principle, where images from different viewpoints are gener-

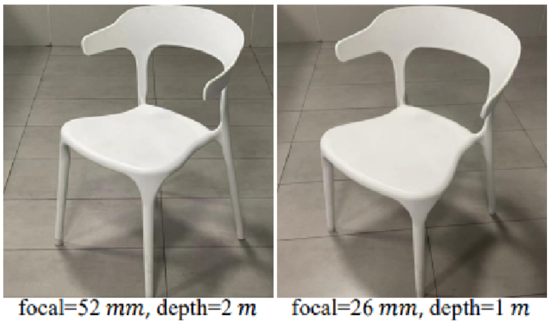


Figure 4.1: The chairs captured by cameras with different focal lengths: The left and right chairs were photographed by cameras with focal lengths of $52mm$ and $26mm$, respectively. Although they have similar visual appearances, their distances from the cameras are $2m$ and $1m$ respectively.

ated by utilizing the predicted depth map and camera pose. During the optimization process, the objective is to minimize the discrepancies between the original image and the reprojected image, making the estimated depth to be an intermediate representation. This enables the network to effectively learn depth estimation without the need for direct supervision from ground-truth labels. Furthermore, self-supervised monocular depth estimation estimates the relative motion of the camera between consecutive frames while simultaneously predicting the depth map for each frame. This approach is particularly appealing because it enables training on arbitrary video sequences, making it more scalable and adaptable to a wide range of environments and tasks. By utilizing the principle of view consistency, the 3D scenes reconstructed from depth estimated by self-supervised monocular methods more accurately present the actual geometric structures of the environment compared to those generated by supervised methods.

Despite the advancements achieved by self-supervised monocular depth estimation, it still faces some challenges, especially when dealing with dynamic scenes of scenes. This issue arises because dynamic objects violate the multi-view consistency assumption, which requires that objects must remain static across frames to ensure their positions in the world coordinate system unchanged. When this assumption is broken, self-supervised methods will introduce inconsistencies that can mislead the depth estimation process, leading to errors in the reconstructed scene. The Fig. 4.2 illustrates several examples about failure cases on the dynamic objects, e.g. vehicles, which indicates the network will assign error values to these moving objects.

To solve this ill-posed problem, there have been some methods attempting to address this by masking out moving objects during training. For example, in the work of Monodepth2 [55], they masked potentially dynamic objects if reprojected error of the warped image is higher than that of the original unwarped image. Alternatively, Casser et al. [24]tried to independently model object motion and ignore moving objects if the results of inverse warping are not aligned with original object appearances. Similarly, Kieran et al. [143] filtered moving objects whose warped masks do not match the original masks. However, due to these networks receive insufficient training samples on dynamic objects, they may lead to poor generalization.

Additionally, some approaches aim to address dynamic objects by integrating auxiliary tasks like 2D optical flow or scene flow estimation. In 2018, Zhichao et al. [209] introduced Res-

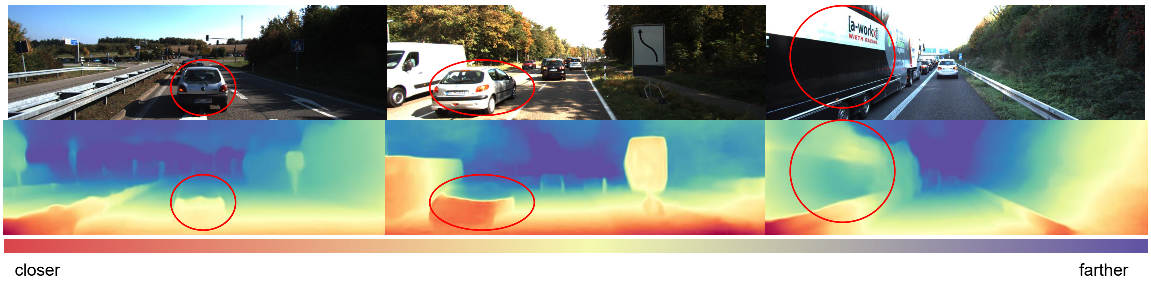


Figure 4.2: Failure cases in self-supervised monocular depth estimation: we employ the existing state-of-art monocular depth estimation network, i.e. LiteMono [220], to predict depth for autonomous driving scenes. It shows unexpected estimation results for moving vehicles.

FlowNet [72] to decompose scene flows from rigid structure reconstructor, facilitating to reason about static and dynamic scene parts separately. Besides, Chenxu et al. [110] adopted MotionNet, DepthNet and OptFlowNet, which are used to predict camera rigid motion, dense depth map and optical flow respectively, to estimate 3D geometric scenes more accurately. By fedding these three information into a holistic 3D motion parser, the rigid background and moving objects can be disentangled. Moreover, Seokju Lee et al. [91] proposed to model independently for every object motion which share the same structure with camera pose network. The whole network follows an end-to-end training paradigm where reprojection loss is composed by rigid background and potentially-moving objects. To address the challenge of jointly predicting depth, scene flow, and motion masks without relying on prior information such as object segmentation, Yihong et al. [157] introduced a novel motion initialization technique combined with regularization strategies, enabling to correct error predictions on dynamic objects.

Another line of research exploits additional perceptual contents, such as semantic segmentation or pseudo depth labels. For example, Marvin et al. [86](SGDepth) leveraged semantic priors to guide the depth learning of dynamic objects by introducing multi-task learning framework, which has been demonstrated to acquire superior performance on the monocular depth estimation. In 2020, based on the previous works [18, 19], Libo et al. [156] proposed SC-DepthV3, which utilized pseudo depth labels generated by pre-trained monocular depth network [206] to guide the estimation results and incorporated two novel losses to further fuse this additional perceptual information. Notably, while these techniques can improve performance, they introduce additional complexity and training challenges due to the larger network architectures.

To decrease the complexity of the network and training process, some research also attempts to utilize the geometric assumption that depth for dynamic objects is generally consistent with the distance of their ground contact points from the camera. Leveraging this idea, some works [61, 124, 199] improved object depth estimation by applying ground consistency loss functions or ground cumulative convolution. However, these approaches may struggle when estimating depth for objects closer to the camera, where surfaces vary vertically, as the ground consistency assumption becomes less reliable. Consequently, they rely on depth labels for supervision or require a supplementary fine-tuning stage to refine results, which precludes end-to-end training.

Building on the idea of using ground information for dynamic object depth estimation, we propose a novel approach that addresses the limitations of previous methods while allowing for end-to-end training. Our method is based on the observation that in the decoder of a depth network, the activated feature maps across different channels can be categorized into depth-aware and detail-aware feature maps: the former provides information concerning the depth distribution in the scene and its smooth behavior, the latter highlights discontinuities and high-frequency details.

We argue that selectively enhancing depth-aware feature maps by propagating depth information from ground regions to moving objects, enables the model to predict depth of objects that remains consistent with the their ground contact points. To implement this strategy, we first identify the depth-aware feature maps that are most relevant to the final depth map's distribution. This is done by computing the cosine similarity between the feature maps and depth pseudo-labels generated from the ground planes. By focusing propagation on the highest-scoring feature maps, we can effectively transfer ground depth information to moving objects, addressing depth ambiguities typically introduced by dynamic scenes. This propagation process is repeated multiple times to ensure ground feature being propagated to the whole objects even when large moving objects are present.

Our approach offers a simple and effective solution to solve the long-standing challenge in dynamic objects depth estimation of monocular settings. Furthermore, the proposed module enables to be seamlessly integrated into any state-of-the-art monocular depth estimation model without introducing any additional network structures and computational complexity. The final experimental results demonstrate that our method achieves state-of-the-art performance when estimating depth for dynamic objects and attains superior generalization compared to existing approaches. In summary, our main contributions in this work are:

- We propose ground propagation, a novel method for dealing with moving objects when training self-supervised monocular depth estimation models.
- Our method is compatible with existing models and requires no additional network parameters.
- Experiments on KITTI [51] and DrivingStereo [192] datasets highlight that our strategy improves the accuracy of any baseline model and achieves state-of-the-art results for dynamic objects.

4.2 Proposed Method

Beyond the multi-view consistency principle, which forms the foundation of self-supervised monocular depth estimation, we propose an additional geometric constraint: the depth of an object should be consistent with its ground contact point. This constraint applies to both stationary and moving objects within a scene, making it particularly useful for handling dynamic objects that disrupt the multi-view consistency assumption. To leverage this insight, we introduce a three-step strategy: 1) depth-aware feature selection, 2) iterative ground feature propagation, and 3) clipping normalization. The following contents will introduce the structures of network framework and implementation details about ground propagation process.

4.2.1 Network structures

The self-supervised monocular depth estimation typically follows encoder-decoder training paradigm containing bridge connections which takes RGB images as inputs and outputs depth maps. In our proposal, we employs two state-of-art monocular depth networks, i.e. Monodepth2 [55] and Lite-mono [220], to achieve depth estimation. These two networks adopt the same decoder structure but leverage different encoder variants, as listed in Tab. 4.1 where Monodepth2 totally yields 4 activations with different sizes to concatenate with downstream decoder layers while Lite-mono just outputs 3 activations. In the Tab. 4.1, the ResBlock, CDCBlock, and LGFI are distinct combinations of convolutional operations. Each is designed to enhance feature extraction efficiency, with varying techniques aimed at capturing different aspects of the input data. These blocks incorporate diverse convolution strategies to improve the model's ability to identify and represent key patterns, ultimately boosting overall performance. The Fig. 4.3 depicts the structure details of Resblock, CDCBlock, LIGF and decoder. By inputting the image into the complete networks, the decoder of Monodepth2 and Lite-mono can output multi-scale disparity maps.

In addition to the depth estimation network, a pose estimation network plays a crucial role in enabling self-supervised depth estimation. The pose network predicts the relative camera motion between consecutive frames by taking adjacent images as input. This is crucial for monocular depth estimation because the camera's movement provides the geometric cues needed to infer depth from single images. The pose network typically learns a transformation matrix that describes the six degrees of freedom (6-DoF) motion between frames, which includes both translation and rotation. In the frameworks used in our work, i.e. Monodepth2 [55] and Lite-Mono [220], the pose network adopts an encoder-decoder architecture, similar to the depth estimation network. Both frameworks utilize ResNet-18 [62] as the encoder and produce 6-DoF motion predictions through convolutional downsampling layers. By jointly training of the depth and pose networks, it enables depth estimation in a self-supervised manner, allowing the depth network to more accurately capture the geometric structure of the scene.

4.2.2 Depth-aware features selection

In the latent space of monocular depth estimation networks, different feature maps are responsible for extracting distinct types of information. Some feature maps capture high-frequency details, such as texture and sharp object boundaries, while others focus on low-frequency, task-specific information. For the task of monocular depth estimation, we classify all the feature activations of decoder as depth-aware feature maps and detail-related feature maps, which corresponds to low-frequency features and high-frequency features respectively. Among them, depth-aware feature maps are sensitive to the global depth structures that contribute significantly to the final depth predictions. As a result, depth-aware maps can be considered more closely aligned with the perceptual contents of depth estimation while detail-aware features more focus on depth edges or object textures.

Based on this insight, we hypothesize that by selectively manipulating these depth-aware feature maps, we can fine-tune the final depth estimation. Therefore, by propagating depth information from the ground contact points to dynamic objects within the scene, it is promising to recalibrate depth for moving objects which often pose challenges on the monocular depth estimation due to their motion and violation of multi-view consistency assumptions.

Table 4.1: The encoder variants for Monodepth2 [55] and Lite-mono [220]: The ConvNxN represents convolutional operations with kernel size of $n \times n$ and symbol $[] \times n$ denotes this block operation will be repeated n times. The Feature column indicates activation of this layer will be concatenated with corresponding layers of decoder. In our experiment, the input image is a three-channel RGB image with resolution of 192×640.

Encoder	**Layer**	**Input Size**	**Output Size**	**Features**
Monodepth2 [54]	Conv7X7	$3 \times 192 \times 640$	$64 \times 96 \times 320$	
	BatchNorm	$64 \times 96 \times 320$	$64 \times 96 \times 320$	
	ReLU	$64 \times 96 \times 320$	$64 \times 96 \times 320$	
	MaxPool	$64 \times 96 \times 320$	$64 \times 48 \times 160$	
	ResBlock	$64 \times 48 \times 160$	$64 \times 48 \times 160$	feature1
	ResBlock	$64 \times 48 \times 160$	$128 \times 24 \times 80$	feature2
	ResBlock	$128 \times 24 \times 80$	$256 \times 12 \times 40$	feature3
	ResBlock	$256 \times 12 \times 40$	$512 \times 6 \times 20$	feature4
Lite-Mono [220]	Conv3X3	$3 \times 192 \times 640$	$48 \times 96 \times 320$	
	[Conv3X3]×2	$48 \times 96 \times 320$	$48 \times 96 \times 320$	
	Conv3X3	$48 \times 96 \times 320$	$48 \times 48 \times 160$	
	[CDCBlock]×3	$48 \times 48 \times 160$	$48 \times 48 \times 160$	
	LGFI	$48 \times 48 \times 160$	$48 \times 48 \times 160$	feature1
	[Conv3X3]×2	$48 \times 48 \times 160$	$48 \times 48 \times 160$	
	Conv3X3	$48 \times 48 \times 160$	$80 \times 24 \times 80$	
	[CDCBlock]×3	$80 \times 24 \times 80$	$80 \times 24 \times 80$	
	LGFI	$80 \times 24 \times 80$	$80 \times 24 \times 80$	feature2
	Conv3X3	$80 \times 24 \times 80$	$128 \times 12 \times 40$	
	[CDCBlock]×9	$128 \times 12 \times 40$	$128 \times 12 \times 40$	
	LGFI	$128 \times 12 \times 40$	$128 \times 12 \times 40$	feature3

To identify these depth-aware feature maps from decoder, we perform inference and extract feature channels from the specific layer in the network, such as the 5th layer in the decoder of the MonoDepth2 or Lite-mono. Specifically, we generate a pseudo depth and disparity maps through Eq. 4.1 where y represents $y-$axis coordinates of image and H denotes image height, that reflects the depth distribution along the ground plane. Next, we compute the cosine similarity between this pseudo depth/disparity map and the feature maps of specific layer. The feature maps exhibiting the highest cosine similarity scores are considered depth-aware since they closely correspond to the depth structure of the scene.

$$pseudo\_disparity = \begin{cases} y - \frac{H}{2}, y > \frac{H}{2} \\ 0, y \leq \frac{H}{2} \end{cases} \tag{4.1a}$$

$$pseudo\_depth = \begin{cases} 1 - y + \frac{H}{2}, y > \frac{H}{2} \\ 1, y \leq \frac{H}{2} \end{cases} \tag{4.1b}$$

This process is illustrated as Fig. 4.4, where the feature maps are extracted by the 5th layer in the decoder in MonoDepth2 [55]. It can be noticed that feature maps with higher cosine-similarity scores are visually closer to the final depth/disparity predictions while that assigned to lower scores are more like related to depth edges, which allows us to propagate cues on the former from ground contact points to dynamic objects.

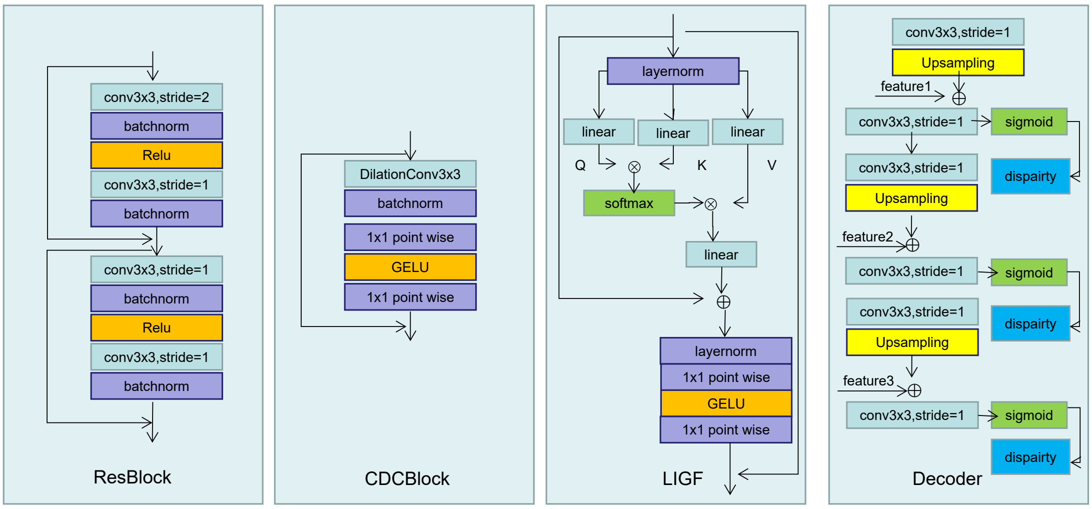


Figure 4.3: Structures of decoder and different convolution blocks in the table4.1: The ResBlock is used in the encoder of Monodepth2[55], CDCBlock and LIGF are for Lite-mono[220] encoder. The Monodepth2 and Lite-mono share the same decoder structure which can output multi-scale disparity maps.

4.2.3 Iterative ground propagation

To adjust the predictions for moving objects, we iteratively propagate ground features to dynamic targets within depth-aware feature maps, making them consistent with the ground contact points. This methodology ensures that the depth information is aligned with the corresponding ground contact points of the objects. Initially, we leverage Mask R-CNN [1] to identify and segment all potential moving objects within the scene, specifically focusing on categories such as vehicles and pedestrians. Following this segmentation, we iteratively propagate the ground feature values across the entirety of each detected object, facilitating a comprehensive update of their depth representations. This iterative process is mathematically formalized in Eq. 4.2:

$$f^r_{i,j} = M_{i,j} * f^{r-1}_{i,j+1} + (1 - M_{i,j}) * f^0_{i,j}, \quad r = 1, 2, 3, \dots, n \tag{4.2}$$

where $f^r_{i,j}$ denotes feature values located at pixel (i, j) during the r-th iteration, and M represents an objects mask obtained from an off-the-shelf semantic segmentation network. The $M_{i,j}$ will be set as 1 if the pixel belongs to moving object. The Eq. 4.2 clearly illustrates that at each iteration, the feature values associated with moving objects are effectively replaced by the features of the pixels located directly beneath them. As a result, after multiple iterations, the feature values of dynamic objects are gradually calibrated by the ground features, while the features of static objects remain unaffected throughout this process.

Furthermore, Figure 4.5 visually illustrates the impact of this ground feature propagation mechanism. As the number of iterations increases, the ground features gradually propagate to the entire moving objects, thereby enforcing a consistency between ground depth/disparity and object depth/disparity. This iterative approach not only rectifies any initial erroneous disparity/depth predictions but also ensures that the depth estimations for dynamic objects are considerably more accurate. Importantly, this iterative ground feature propagation is executed exclusively on the depth-aware feature maps, which allows the method to effectively preserve

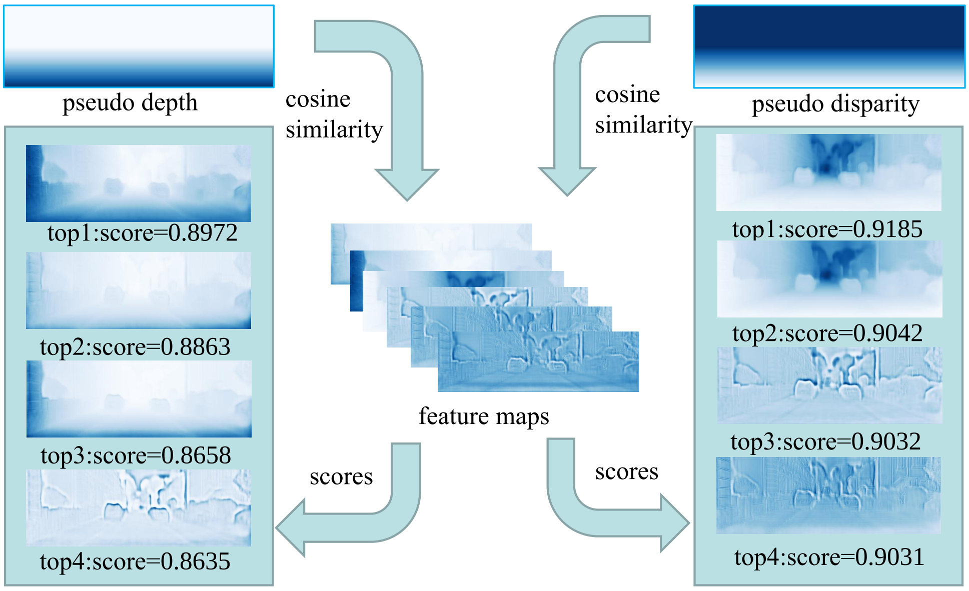


Figure 4.4: Feature maps ranking: We construct pseudo depth/disparity maps to rank the feature maps and identify depth-related maps, according to the cosine similarity with the former. Pseudo depths/disparities are generated to recall the depth/disparity distribution in correspondence with the ground plane.

depth details in scenarios where the surfaces of moving objects are not perfectly perpendicular to the ground, such as in the case of vehicles approaching the camera.

4.2.4 Clipping normalization

We argue that naı̈vely overwriting object features with ground features may be an overly aggressive approach, especially in cases where the objects are not moving. Therefore, proposed ground propagation operation is truly necessary only when there is a substantial disparity between the ground and object features — typically in scenarios involving moving objects.

To mitigate this, we maintain crucial information from the original feature set f^0 by updating the final features f^n using a weighted sum that balances the original and modified features, ensuring a more refined adjustment. This weight is determined by assessing the difference between the calibrated feature values and the original values. When the gap between these values is small, the model is more inclined to retain the original feature values. Conversely, when the gap is larger, the final prediction shifts more significantly towards the values derived after ground feature propagation. As for mathematical expression, the weight is computed through normalized absolute differences between f^0 and f^n as follow:

$$w = \frac{d - min(d)}{max(d) - min(d)}, d = \left|f^n_{i,j} - f^0_{i,j}\right| \tag{4.3}$$

The final feature f^n is the weighted sum of original features and that after performing ground propagation:

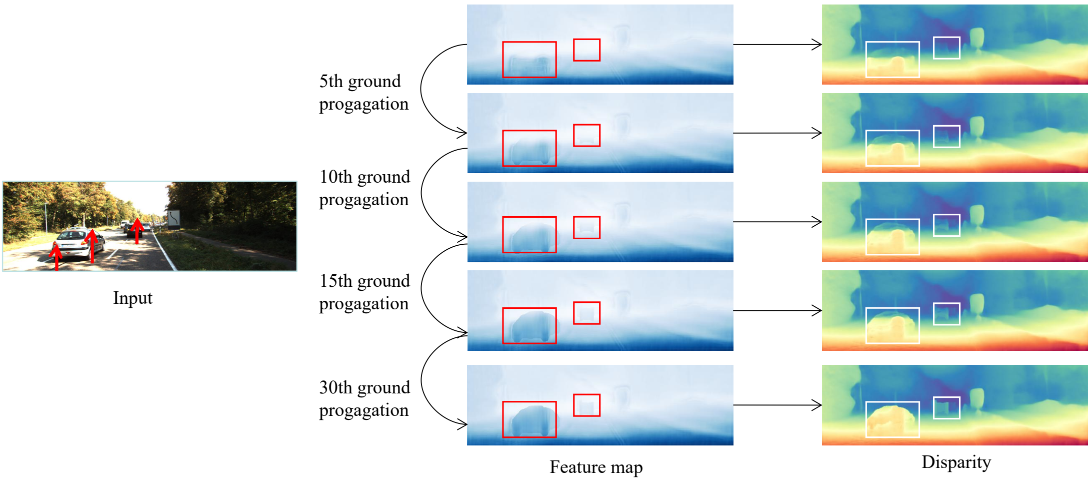


Figure 4.5: Ground Propagation in action: We iteratively perform ground propagation on the feature map of the 5th layer of the depth decoder for 30 steps. The corresponding outputs of reverse depth are shown in the rightmost image column

$$f^n = \min(w, 1) * f^n + (1 - \min(w, 1)) * f^0 \tag{4.4}$$

However, due to the strong generalization capabilities of the current networks, the proportion of incorrectly estimated targets in a scene is significantly smaller than that of correctly estimated ones. This imbalance causes the weight distribution to be skewed towards retaining the original values, resulting in insufficient adjustments by the ground propagation mechanism. Consequently, the depth predictions for the few mis-estimated objects are not effectively corrected, limiting the system's ability to refine predictions in those specific cases.

To address this issue, we introduce clipping normalization to adjust the weight evaluation process, ensuring that objects with inaccurately estimated depth receive higher weight values. As formulated in Eq. 4.5, the weight w is calculated by dividing the absolute difference between the two features by the C -percentile of its maximum value. This ensures that the relative difference is normalized in proportion to the range of observed differences, making it more sensitive to larger discrepancies and allowing the propagation mechanism to adjust accordingly.

$$w_{i,j} = \frac{\left| f^n_{i,j} - f^0_{i,j} \right|}{\max(|f^n - f^0|) * C}, 0 < C \leq 1 \tag{4.5}$$

Accordingly, by performing clipping normalization on the weight computation, we preserve the reliable features learned according to the multi-view consistency principle and correct only the outlier values according to ground depth consistency.

4.2.5 Training loss

Based on the view consistency principle, the depth network and pose network are jointly trained by minimizing the photometric error between the original and reconstructed images, using the predicted depth D and estimated camera pose $T_{t\rightarrow s}$ to reproject pixels p_s through Eq. 4.6 where K denotes camera intrinsic parameters and p_t is the pixel coordinate of target image.

$$p_s \sim KT_{t \to s} D_t(p_t) K^{-1} p_t \tag{4.6}$$

This formula indicates that warped image will be sampled on the source image(i.e. temporal previous frame I_{t-1} or next frame I_{t+1}) according to reprojected coordinates p_s. The total loss for the network is composed of three components: the photometric loss, the structural similarity (SSIM) loss [184], and the smoothness loss.

Photometric Loss: It measures the pixel-wise difference between a target image I_t and reprojected image I'_t, generated by warping a source image I_s. The photometric loss is typically defined as a combination of L1 loss and SSIM [184]:

$$\mathcal{L}_p(I'_t) = \alpha \frac{1 - \text{SSIM}(I_t, I'_t)}{2} + (1 - \alpha)|I_t - I'_t| \tag{4.7}$$

where SSIM (Structural Similarity Index) captures perceptual differences in luminance, contrast, and structure, and α is a weight parameter balancing the contributions of SSIM and L1 loss, typically set as 0.85.

To mitigate the impact of occlusions and stationary frames on depth estimation, we utilize a minimum per-pixel loss to calculate the photometric loss, as outlined in Monodepth2 [55]. This technique evaluates the photometric loss for each pixel by selecting the minimum value from the photometric losses associated with four items listed in the Eq. 4.8 where $I'_{t-1 \to t}$ and $I'_{t+1 \to t}$ represent the reprojected image warped from I_{t-1} and I_{t+1} respectively.

$$\mathcal{L}_p = \min(\mathcal{L}_p(I'_{t+1 \to t}), \mathcal{L}_p(I'_{t-1 \to t}), \mathcal{L}_p(I_{t+1}), \mathcal{L}_p(I_{t-1})) \tag{4.8}$$

This formulation ensures that the photometric loss reflects the best possible alignment for each pixel, thereby enhancing the robustness of the depth estimation process. By focusing on the minimum loss, we effectively reduce the influence of occluded areas and improve the quality of depth predictions.

Smoothness Loss: The smoothness term ensures that the predicted depth map is smooth in regions with low image gradients while allowing for discontinuities at object boundaries. The smoothness loss is defined as:

$$\mathcal{L}_s = |\partial_x D_t| e^{-\partial_x I_t} + |\partial_y D_t| e^{-\partial_y I_t} \tag{4.9}$$

where I_t is the target RGB image and D_t represents corresponding predicted depth map. The exponential terms $e^{-\partial_x I_t}$ and $e^{-\partial_y I_t}$ allow the depth map to remain smooth within objects while preserving depth transitions at edges which are indicated by image intensities, ensuring that the depth predictions accurately capture object boundaries without unnecessary smoothing across them.

Due to the limitation of deep neural networks, continuous upsampling operations can hardly preserve high-frequency information captured by the encoder, leading to blurry of depth estimation. To address this issue, monocular depth estimation networks typically employ multi-scale disparity prediction, as illustrated in Fig. 4.3. This approach allows each layer in the decoder to predict disparities, then these multi-resolution disparities are incorporated into the overall loss calculation:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} (\mathcal{L}_p^i + 2^i * \lambda * \mathcal{L}_s^i) \tag{4.10}$$

where λ is generally set as 1e-3 to control smoothness confidence and N multi-scale disparities are used to compute the photometric loss $\mathcal{L}_p^i$and the smoothness loss $\mathcal{L}_s^i$ respectively. The overall network loss is computed as the average sum of the disparity losses across multiple scales.

4.3 Experimental Results

In this section, we collect the outcome of our experiments to support the effectiveness of our proposed ground propagation strategy with respect to existing solutions.

4.3.1 Implementation details and metrics

We apply our strategy to two self-supervised monocular depth estimation frameworks: Monodepth2 [55] and Lite-Mono [220]. We implement the two variants of our framework in Pytorch, starting from the existing codebases of both models and training them following the original training schedules. Specifically, MonoDepth2 and Lite-Mono variants are trained respectively for 20 and 35 epochs on the KITTI dataset with batch size set to 12. For the remaining training hyper-parameters, losses, and optimizer, we adhered to the original settings detailed in the respective papers [55, 220]. In our experiments, we use a single RTX 3090 GPU and process images at 640×192 resolution. Overall, the network training requires about 15 hours and we adopt the same data augmentation detailed by [55]. Regarding ground propagation, the Monodepth2 variant applies it on the 2nd, 3rd, 4th, and 5th decoder layers for 4, 8, 16, and 32 iterations respectively. The Lite-Mono variant implements ground propagation on the 1st, 2nd, and 3rd decoder layers, using 8, 16, and 32 iterations. Given any layer, we run ground propagation on the $\frac{1}{8}$ and $\frac{1}{16}$ feature maps having the highest cosine similarity with respect to the predicted depth map, for the Monodepth2 and Lite-Mono variants respectively. We set the clipping normalization rate to 0.3 for both. This process occurs during both training and testing phases.

For evaluation, we compute the seven standard metrics (Abs Rel, Sq Rel, RMSE, RMSE log, $\delta_1 < 1.25$, $\delta_2 < 1.25^2$, $\delta_3 < 1.25^3$) proposed by Eigen and Fergus [41] and used by most works in the literature. In each table, we highlight with **bold** or <u>underline</u> the best and second-best results respectively.

4.3.2 Datasets

We conduct our experiments on two popular driving datasets.

KITTI [51]. The KITTI stereo dataset contains 61 scenes, with a typical image size of 1242×375, captured using a stereo rig mounted on a moving car equipped with a LiDAR sensor. Following previous works in this field [55, 220], we use the image split of Eigen *et al.* [41], which consists of 39810 monocular triplets for training and 4424 for validation. To compare with the existing solutions, we evaluate the depth performance on the test split of [41] either using raw LiDAR (697 images) or improved ground truth labels [167] (652 images).

DrivingStereo [192]. It is a large-scale stereo dataset depicting autonomous driving scenarios. Among several sequences, we use the four image splits made available on the website, each made of 500 frames collected under different weather conditions, respectively *foggy*, *cloudy*, *rainy* and *sunny*. We use this dataset to evaluate the generalization capacity of existing solutions and ours.

4.3.3 Depth evaluation

We start by evaluating the overall accuracy of depth maps predicted by our model and existing ones. For all evaluations, we apply median scaling [54] relative to the ground truth to recover the metric scale, which is typically lost in self-supervised training on monocular videos.

Results on KITTI. We evaluate our models on the established KITTI Eigen split [41], comprising 697 images paired with raw LiDAR scans. Although these scans produce several outliers when projected on the image plane, we use them to allow a fair comparison with existing works, before moving to more accurate experiments with the improved ground truth [167]. Table 4.2 collects the outcome of this evaluation, involving several existing frameworks for self-supervised monocular depth estimation, including those specifically designed to handle dynamic objects, such as Dynamo-Depth [157] and From-Ground-To-Objects (FGTO). These latter are grouped at the bottom of the table, depending on the backbone they deploy – either MonoDepth2 or Lite-Mono. In each block, our solution consistently outperforms the original model and achieves more accurate results compared to both Dynamo-Depth and FGTO. Notably, Dynamo-Depth fails to improve the overall accuracy of MonoDepth2 and Lite-Mono, despite enhancing results for dynamic objects, as we will appreciate in the remainder. FGTO, conversely, succeeds in this regard but introduces a two-stage training protocol. Eventually, our strategy further improves the results while maintaining a single-stage training paradigm.

Method	M.N	Data	Abs Rel↓	Sq Rel↓	RMSE↓	RMSElog↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
Geo-Net [209]	✓	K	0.155	1.296	5.857	0.233	0.793	0.931	0.973
Struct2Depth [24]	✓	K	0.141	1.026	5.290	0.215	0.816	0.945	0.979
SC-DepthV3 [156]		K	0.118	0.756	4.709	0.188	0.864	0.960	0.984
Dyna-DM [143]	✓	C+K	0.115	0.785	4.698	0.192	0.871	0.959	0.982
SGDepth [86]	✓	C+K	0.113	0.835	4.693	0.191	0.879	0.961	0.981
Insta-DM [91]	✓	K	0.112	0.777	4.772	0.191	0.872	0.959	0.982
Monodepth2 [55]		K	0.115	0.917	4.880	0.193	0.877	0.959	0.981
Dynamo-Depth (Monodepth2) [157]	✓	K	0.120	0.864	4.850	0.195	0.858	0.956	0.982
FGTO (Monodepth2) [124]		K	0.112	0.866	4.766	0.190	0.879	0.960	0.982
Ours (Monodepth2)		K	0.111	0.797	4.682	0.188	0.880	0.961	0.982
Lite-Mono [220]		K	0.107	0.765	4.561	0.183	0.886	0.963	0.983
Dynamo-Depth (Lite-Mono)	✓	K	0.112	**0.758**	**4.505**	0.183	0.873	0.959	**0.984**
Ours (Lite-Mono)		K	**0.106**	0.761	4.529	**0.181**	**0.888**	**0.964**	0.983

Table 4.2: Results on KITTI Eigen split [41] – raw LiDAR as ground truth. Any network processes 192 × 640 images (except Dyna-DM, SC-Depthv3, and Insta-DM, processing 256 × 892 images). For each method, we report the use of additional motion networks to deal with dynamic objects (M.N).

Results on KITTI – Improved Ground Truth. We repeat the same evaluation using the improved ground truth labels provided by [167], which reduces the number of testing images to 652. Table 4.3 summarizes the results from this evaluation. In general, we can observe lower errors compared to the previous experiments, thanks to the absence of outliers in these improved ground truth labels. In particular, we highlight once again the comparison between the two baseline models, MonoDepth2 and Lite-Mono, the Dynamo-Depth variants and ours. We can observe a trend similar to the one observed in the raw LiDAR evaluation, with Dynamo-Depth being not capable of improving over the baseline models, whereas our models consistently do.

Method	M.N	Abs Rel↓	Sq Rel↓	RMSE↓	RMSElog↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
Insta-DM	✓	0.091	0.506	3.997	0.141	0.907	0.981	0.995
SC-DepthV3		0.099	0.523	4.094	0.144	0.897	0.979	0.995
Dyna-DM	✓	0.092	0.494	3.898	0.140	0.907	0.980	0.995
SGDepth	✓	0.085	0.491	3.755	0.130	0.921	0.984	**0.996**
Monodepth2		0.090	0.545	3.942	0.137	0.914	0.983	0.995
Dynamo-Depth (Monodepth2)	✓	0.096	0.552	4.075	0.145	0.901	0.979	0.995
Ours (Monodepth2)		0.089	0.494	3.843	0.136	0.914	0.983	0.995
Lite-Mono		0.083	**0.455**	3.689	0.128	0.923	0.985	**0.996**
Dynamo-Depth (Lite-Mono)	✓	0.088	0.463	3.692	0.131	0.917	0.984	**0.996**
Ours (Lite-Mono)		**0.081**	0.458	**3.603**	**0.124**	**0.928**	**0.986**	**0.996**

Table 4.3: Results on KITTI Eigen split [41] – improved ground truth [167]. Any network processes 192×640 images (except Dyna-DM, SC-Depthv3, and Insta-DM, processing 256×892 images). For each method, we report the use of additional motion networks to deal with dynamic objects (M.N).

Results on DrivingStereo. Finally, to assess the generalization capability of the models, we evaluate under four different weather conditions, including*foggy*, *cloudy*, *rainy* and *sunny*, from the DrivingStereo [192] dataset. Table 4.4 collects the outcome of this evaluation, conducted by applying KITTI-trained models to DrivingStereo without fine-tuning. This experiment reveals a more significant performance gap between existing methods – such as Insta-DM, SC-DepthV3, Dyna-DM, SGDepth, and Dynamo-Depth – and our solution. Notably, our Lite-Mono variant achieves a substantial improvement over both these methods and the original Lite-Mono backbone.

Method	M.N	Abs Rel↓	Sq Rel↓	RMSE↓	RMSElog↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
Insta-DM	✓	0.217	3.961	12.156	0.296	0.694	0.885	0.950
SC-DepthV3		0.198	2.589	9.685	0.259	4.708	0.914	0.971
Dyna-DM	✓	0.214	4.068	11.766	0.277	0.720	0.898	0.957
SGDepth	✓	0.166	2.231	9.590	0.237	0.770	0.928	0.972
Monodepth2		0.173	2.582	9.753	0.239	0.771	0.929	0.973
Dynamo-Depth (Monodepth2)	✓	0.181	2.966	10.364	0.245	0.766	0.923	0.971
Ours (Monodepth2)		0.169	2.503	9.645	0.234	0.777	0.932	0.975
Lite-Mono		0.160	2.318	9.338	0.225	0.794	0.937	0.976
Dynamo-Depth (Lite-Mono)	✓	0.179	3.169	10.562	0.236	0.778	0.926	0.973
Ours (Lite-Mono)		**0.156**	**2.165**	**9.043**	**0.221**	**0.801**	**0.941**	**0.978**

Table 4.4: Results on DrivingStereo [192] dataset. Any network processes 192×640 images (except Dyna-DM, SC-Depthv3, and Insta-DM, processing 256×892 images). For each method, we report the use of additional motion networks to deal with dynamic objects (M.N).

4.3.4 Depth evaluation for dynamic objects

We now focus on dynamic objects, by measuring the accuracy of existing solutions and ours at estimating their depth. Purposely, we use a pre-trained semantic segmentation network [29]

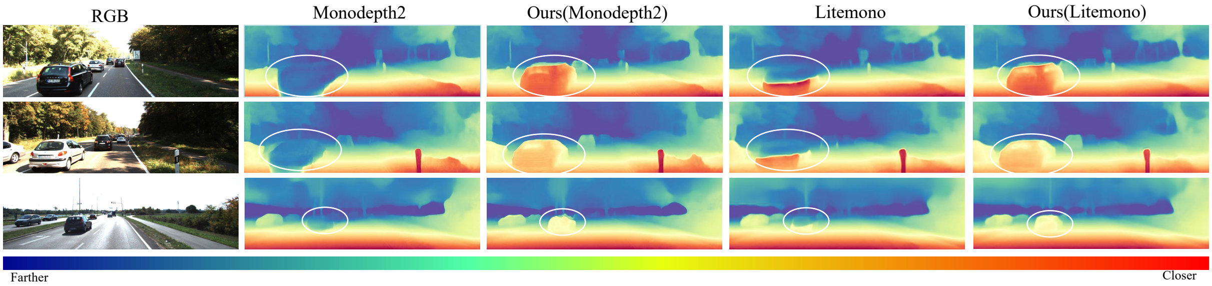


Figure 4.6: Handling dynamic objects with ground propagation.: Our solution effectively boosts the performance of existing self-supervised models such as Monodepth2 [55] and Lite-Mono [220].

to segment cars in the testing images and compute the error metrics only over them during evaluation.

Results on KITTI – Improved Ground Truth. We start this additional evaluation on the KITTI dataset, using the improved ground truth [167]. Table 4.5 presents the results obtained by evaluating only pixels corresponding to cars. We can observe significantly higher error metrics and lower accuracy compared to those in Table 4.4, confirming that the moving objects pose a major challenge to self-supervised monocular depth estimation frameworks. We can appreciate how many of the existing solutions, such as Insta-DM and Dyna-DM, are indeed more effective than MonoDepth2 and Lite-Mono on dynamic objects. Dynamo-Depth improves over the MonoDepth2 baseline but struggles when applied to the Lite-Mono backbone. In contrast, our strategy is effective when applied to both and achieves the best overall results. To further verify the effectiveness of the ground propagation module on the two monocular networks, we apply the improved models to test two dynamic scenes, as shown in the Fig. 4.6. Compared with the original depth estimation networks, our proposed method can effectively correct the depth of dynamic objects.

Method	M.N	Abs Rel↓	Sq Rel↓	RMSE↓	RMSElog↓	$\delta < 1.25$↑	$\delta < 1.25^2$↑	$\delta < 1.25^3$↑
Insta-DM	✓	0.130	1.088	5.242	0.179	0.818	0.949	0.987
SC-DepthV3		0.157	1.381	5.553	0.186	0.791	0.949	0.989
Dyna-DM	✓	0.123	0.935	4.797	0.162	0.845	**0.966**	**0.992**
SGDepth	✓	0.146	1.642	5.743	0.185	0.814	0.950	0.983
Monodepth2		0.147	1.731	6.003	0.188	0.815	0.951	0.981
Dynamo-Depth (Monodepth2)	✓	0.158	1.583	5.916	0.199	0.783	0.941	0.984
Ours (Monodepth2)		0.125	1.082	5.276	0.177	0.834	0.955	0.988
Lite-Mono		0.133	1.207	5.263	0.175	0.828	0.956	0.987
Dynamo-Depth (Lite-Mono)	✓	0.147	1.280	5.283	0.184	0.808	0.947	0.985
Ours (Lite-Mono)		**0.117**	**0.912**	**4.734**	**0.161**	**0.862**	0.964	0.989

Table 4.5: Dynamic Objects Evaluation: results on KITTI Eigen split [41] – improved ground truth [167]. Any network processes 192×640 images (except Dyna-DM, SC-Depthv3, and Insta-DM, processing 256×892 images). For each method, we report the use of additional motion networks to deal with dynamic objects (M.N).

Results on DrivingStereo. We also evaluate the accuracy of estimated depth over dynamic objects on the DrivingStereo dataset. Table 4.6 reports the outcome of this experiment, confirming once again that our models achieve the best results over dynamic objects. The superior accuracy achieved by our method in this setting can also be perceived qualitatively, as in Figure 4.7. Here, we can appreciate how Monodepth2 fails at predicting the correct depth for moving objects in five examples from the DrivingStereo dataset. While DynamoDepth and InstaDM occasionally compensate for these errors, they cannot fully resolve the issue. In contrast, our approach consistently produces satisfactory depth predictions.

Method	M.N	Abs Rel↓	Sq Rel↓	RMSE↓	RMSElog↓	$\delta < 1.25$↑	$\delta < 1.25^2$↑	$\delta < 1.25^3$↑
Insta-DM	✓	0.245	5.578	13.097	0.286	0.620	0.861	0.944
SC-DepthV3		0.268	5.045	12.655	0.279	0.575	0.857	0.958
Dyna-DM	✓	0.244	5.769	12.696	0.271	0.655	0.884	0.951
SGDepth	✓	0.210	4.102	12.703	0.278	0.627	0.850	0.943
Monodepth2		0.208	4.143	12.266	0.262	0.664	0.881	0.957
Dynamo-Depth (Monodepth2)	✓	0.245	5.341	12.771	0.280	0.628	0.864	0.950
Ours (Monodepth2)		0.185	3.376	11.854	0.252	0.676	0.878	0.959
Lite-Mono		0.197	3.483	11.366	0.244	0.680	0.897	0.968
Dynamo-Depth (Lite-Mono)	✓	0.214	3.959	11.261	0.241	0.683	**0.911**	**0.972**
Ours (Lite-Mono)		**0.173**	**2.713**	**10.578**	**0.227**	**0.710**	**0.911**	**0.974**

Table 4.6: Dynamic Objects Evaluation: results on DrivingStereo [192] dataset. Any network processes 192×640 images (except Dyna-DM, SC-Depthv3, and Insta-DM, processing 256×892 images). For each method, we report the use of additional motion networks to deal with dynamic objects (M.N).

4.3.5 Ablation study

We conclude our experiments with ablation studies. Table 4.7 reports, from top to bottom, analyses of (a) the number of features selected for ground propagation and (b) the clipping rate. By focusing on the former aspect (a), we can observe that the most favorable outcomes are achieved when selecting the top $\frac{1}{8}$ of the features map, whereas increasing or decreasing this selection yields drops in accuracy.

Concerning the latter (b), we apply different clipping rate values to determine if retaining part of the original features can further improve the results. We found that setting the clip rate to 0.3 consistently enhances the performance, with other values showing no significant improvements.

4.4 Conclusions

In this thesis, we introduced a novel technique for handling dynamic objects in self-supervised monocular depth estimation. Our approach focuses on propagating feature information from ground contact points up to dynamic objects, allowing for the recalibration of depth-aware features. This enables the decoder to predict consistent depths across both static and dynamic regions of the scene. Unlike prior methods that often rely on masking or require additional prediction networks, our strategy maintains an efficient end-to-end training framework without adding any new network parameters. Extensive experiments on benchmark datasets such as

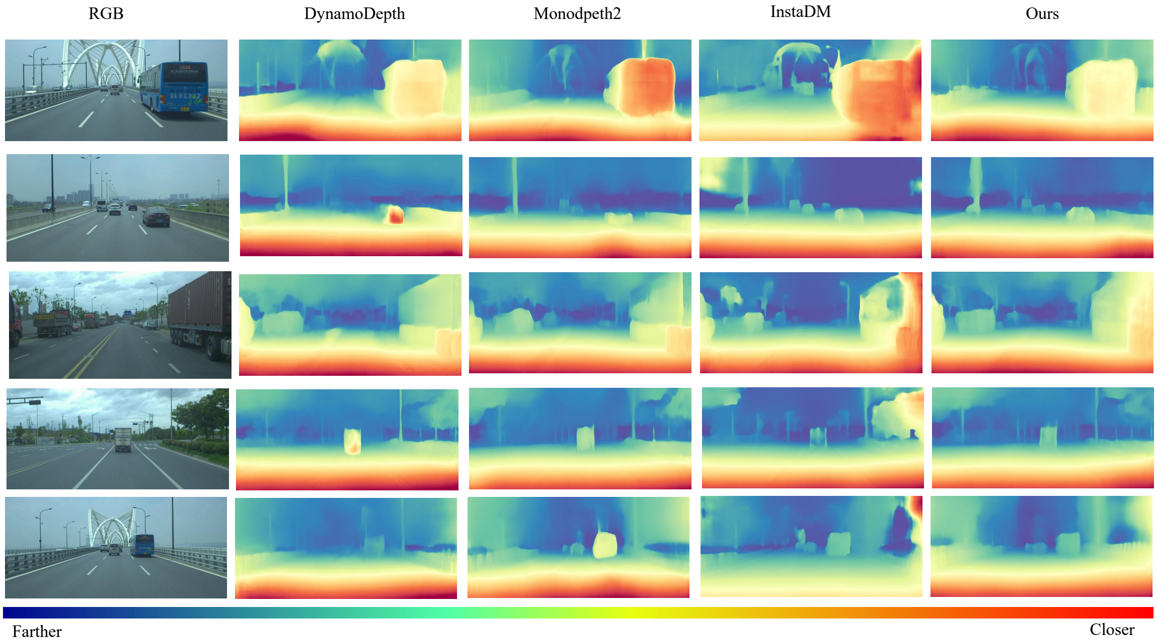


Figure 4.7: Qualitative Results – DrivingStereo dataset [192]. While existing approaches often fail at properly perceiving moving objects, ours predicts consistent depth maps also in the presence of these latter.

KITTI and DrivingStereo validated the effectiveness of our method. The results demonstrated a clear improvement in accuracy over existing approaches, particularly in scenes with complex dynamic objects, while preserving the efficiency of the network. This advancement represents a significant step forward in monocular depth estimation, ensuring reliable depth predictions across various scenarios.

Method	Selected Features	Abs Rel↓	Sq Rel↓	RMSE↓	RMSElog↓	$\delta < 1.25$↑	$\delta < 1.25^2$↑	$\delta < 1.25^3$↑
Ours (Monodepth2)	All	0.115	0.810	4.766	0.196	0.874	0.960	0.980
	$\frac{1}{2}$	0.116	0.829	4.820	0.195	0.872	0.959	0.980
	$\frac{1}{8}$	**0.112**	**0.787**	**4.699**	**0.189**	0.878	**0.961**	**0.982**
	$\frac{1}{16}$	**0.112**	0.826	4.779	0.190	**0.879**	**0.961**	**0.982**

(a)

Method	Clipping rate	Abs Rel↓	Sq Rel↓	RMSE↓	RMSElog↓	$\delta < 1.25$↑	$\delta < 1.25^2$↑	$\delta < 1.25^3$↑
Ours (Monodepth2)	0.8	0.113	0.808	4.752	0.191	0.876	0.960	0.981
	0.5	0.112	0.819	4.720	0.190	**0.880**	0.960	0.981
	0.3	**0.111**	**0.797**	**4.682**	**0.188**	**0.880**	**0.961**	**0.982**
	0.1	0.113	0.824	4.744	0.190	0.879	0.960	0.982

(b)

Table 4.7: Ablation Studies on KITTI Eigen split [41] – raw LiDAR as ground truth. We evaluate the contributions by (a) Ground Propagation and (b) Clipping Normalization with Monodepth2 as the baseline.

Chapter 5

Conclusion

5.1 Summary of Thesis Achievements

In this thesis, we first provide a comprehensive introduction of foundational geometric concepts and theories related to 3D vision, covering essential concepts such as 3D coordinate system, camera calibration parameters, and perspective transformations. These theoretical aspects establish the groundwork for understanding and advancing depth estimation. Following this, we discuss the core research objectives associated with depth estimation, reviewing prominent datasets and standard evaluation metrics that enable comparative analysis of various approaches.

Subsequently, we present an in-depth review of state-of-the-art methodologies in depth estimation, evaluating each approach in terms of accuracy, computational cost, and suitability for real-world applications. Through this analysis, we highlight the strengths and limitations of existing methods, establishing a foundation for our primary research focus: monocular depth estimation. Monocular depth estimation remains a particularly challenging task, as it requires deriving depth information from a single RGB image without the aid of stereo or LiDAR data. Despite this research field has seen significant advancements, it still face several challenges, such as scale ambiguity, dynamic targets processing and more. To solve these problem, we propose to incorporate ground geometry constraints into monocular depth estimation.

We address both dynamic and static scenes by developing distinct strategies for each context. In static scenes, we begin by estimating the ground normal vector using pedestrian probes to achieve scene alignment. The estimated ground normal is subsequently employed to calculate a scale factor, which is able to convert predicted relative depths into metric distances. The proposed method allows to estimate dense absolute depths for static scenes without any complex camera calibration or depth labels, which is promising to measure social distances and vehicle monitoring in any real-world applications.

For dynamic scenes, we designed a ground propagation module within existing state-of-art self-supervised monocular depth estimation networks. This module iteratively propagates ground features to dynamic targets, ensuring that object depths remain consistent with the depths at their ground contact points. Moreover, since the ground propagation module is applied exclusively to depth-aware features, it effectively preserves high-frequency details in depth estimation. This approach operates within an end-to-end training framework without requiring additional training stages or parameters, thereby maintaining computational efficiency while enhancing depth accuracy in dynamic scenes.

In conclusion, the ground geometry plays a crucial role in enhancing depth estimation accuracy. By providing valuable scene context, it is beneficial to identify actual spatial relationships within a scene. The incorporation of ground geometry can mitigate common challenges such as scale ambiguity and improve the consistency of depth predictions across varying environments. Furthermore, leveraging ground geometry facilitates the alignment of depth information with real-world coordinates, leading to more accurate 3D reconstructions. Overall, integrating ground geometry into depth estimation frameworks not only enhances precision but also contributes to a more robust understanding of complex scenes.

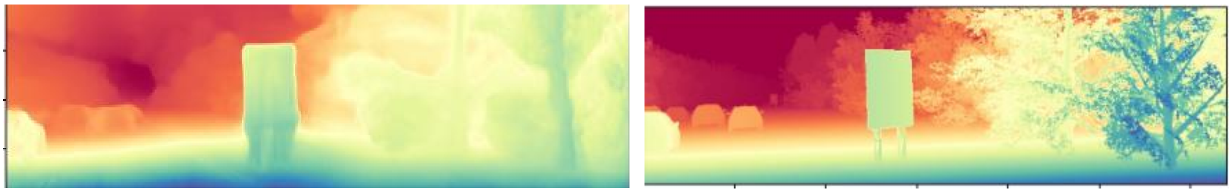
5.2 Future Work

While self-supervised monocular depth estimation has demonstrated commendable accuracy as an alternative approach, self-supervised approaches still show limitations in generalization and high-frequency detail representation compared to supervised methods. Although some recent research [15, 16, 214] have attempted to explore zero-shot metric depth estimation, they heavily depend on extensive datasets, typically millions of images, resulting in unaffordable training costs. Moreover, existing pre-trained models often struggle to adapt to the rapidly growing number of novel scenarios. In contrast, self-supervised monocular depth estimation offers a more cost-effective solution, requiring only tens of thousands of images to predict metric depth, thereby significantly reducing training expenses. However, as discussed in Section 3, this approach is constrained by the pixel projection loss, which hampers its ability to accurately estimate depth in weakly textured regions such as the sky or walls. Additionally, the inherent limitations of the mean absolute error estimation often result in the loss of high-frequency details, leading to blurred depth predictions. Furthermore, self-supervised methods tend to fail in estimating the depth of moving objects. While self-supervised methods can provide valuable scale information, their overall prediction quality generally underperform when compared to state-of-the-art data-driven approaches, such as DepthAnything [197]. To minimize the training costs associated with metric depth estimation, our future work will focus on developing a novel depth refinement framework. This framework will utilize metric depth provided by self-supervised methods as conditional inputs and is expected to produce refined depth predictions with enhanced accuracy for moving objects and improved high-frequency details. The Fig. 5.1 illustrates the original metric depth and refined depth result, where metric depth has accurate metric information but presents blurry edges while our refined depth inherits scale from metric depth and also optimize depth edges and inaccurate predictions.

In future work, our proposed depth refinement network has the potential to significantly enhance depth estimation accuracy while requiring only a minimal number of training samples.

To achieve depth refinement, we will employ a diffusion model [125] as the core of our depth refinement framework. The rationale behind choosing a generative model over discriminative models, such as DPT [137], lies in the diffusion model's superior generalization capabilities. This advantage enables the optimization of networks even when trained on limited data samples, making it particularly suitable for scenarios with fewer training data.

In summarize, our future work will focus on designing a novel depth refinement network based on the diffusion model. This framework will take existing scale-aware depth predictions as conditional inputs, including predictions from self-supervised depth estimation methods and other metric depth estimation methods, such as ZoeDepth [17] and Metric3Dv2 [67]. The pro-



Metric depth(Monodepth2)

Refined Depth

Figure 5.1: A comparison between the original metric depth and the refined depth generated by our refinement network: The left image, estimated using Monodepth2 [55], exhibits accurate scale information but suffers from blurred details. The right image, refined by our proposed method, not only preserves the scale information of metric depth inputs and also achieves significantly sharper object boundaries.

posed network is expected to generate higher-quality depth images that not only retain accurate scale information but also recover more precise depth contours and finer details. By leveraging the generative capabilities of the diffusion model, this approach aims to address the limitations of current methods, particularly in handling weakly textured regions and moving objects, while maintaining computational efficiency and reducing reliance on large-scale training datasets.

Bibliography

[1] W. Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. `https://github.com/matterport/Mask_RCNN`, 2017.

[2] A. Agarwal and C. Arora. Depthformer: Multiscale vision transformer for monocular depth estimation with global local information fusion. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 3873–3877, 2022. doi: 10.1109/ICIP46576.2022.9897187.

[3] M. Aghaei, M. Bustreo, Y. Wang, G. Bailo, P. Morerio, and A. Del Bue. Single image human proxemics estimation for visual social distancing. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2784–2794, 2021. doi: 10.1109/WACV48630.2021.00283.

[4] I. Alhashim and P. Wonka. High quality monocular depth estimation via transfer learning. *arXiv e-prints*, abs/1812.11941:arXiv:1812.11941, 2018. URL `https://arxiv.org/abs/1812.11941`.

[5] A. Ali, R. Ali, and M. Baig. Dense monocular depth estimation with densely connected convolutional networks. In *International Joint Conference on Advances in Computational Intelligence*, pages 393–405. Springer, 2022.

[6] J. Azimjonov, A. Özmen, and M. Varan. A vision-based real-time traffic flow monitoring system for road intersections. *Multimedia Tools Appl.*, 82(16):25155–25174, Feb. 2023. ISSN 1380-7501. doi: 10.1007/s11042-023-14418-w. URL `https://doi.org/10.1007/s11042-023-14418-w`.

[7] 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.

[8] G. Bae, I. Budvytis, and R. Cipolla. Multi-view depth estimation by fusing single-view depth probability with multi-view geometry. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2832–2841, 2022. doi: 10.1109/CVPR52688.2022.00286.

[9] J. Bae, S. Moon, and S. Im. Deep digging into the generalization of self-supervised monocular depth estimation, 2023. URL `https://arxiv.org/abs/2205.11083`.

[10] H. Bao, L. Dong, S. Piao, and F. Wei. Beit: Bert pre-training of image transformers, 2022. URL https://arxiv.org/abs/2106.08254.

[11] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008. ISSN 1077-3142. doi: https://doi.org/10.1016/j.cviu.2007.09.014. URL https://www.sciencedirect.com/science/article/pii/S1077314207001555. Similarity Matching in Computer Vision and Multimedia.

[12] B. Behroozpour, P. A. M. Sandborn, M. C. Wu, and B. E. Boser. Lidar system architectures and circuits. *IEEE Communications Magazine*, 55(10):135–142, 2017. doi: 10.1109/MCOM.2017.1700030.

[13] F. Besse, C. Rhemann, R. Carsten, and K. Kaustav. Real-time stereo matching on gpu with non-parametric belief propagation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2192–2199, 2010.

[14] S. Bhasin, S. Saini, D. Gupta, and S. Mann. Computer vision based traffic monitoring system. In *2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pages 1–6, 2024. doi: 10.1109/ICRITO61523.2024.10522331.

[15] S. F. Bhat, I. Alhashim, and P. Wonka. Localbins: Improving depth estimation by learning local distributions, 2022. URL https://arxiv.org/abs/2203.15132.

[16] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth, 2023. URL https://arxiv.org/abs/2302.12288.

[17] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth, 2023. URL https://arxiv.org/abs/2302.12288.

[18] J.-W. Bian, H. Zhan, N. Wang, T.-J. Chin, C. Shen, and I. Reid. Auto-rectify network for unsupervised indoor depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021.

[19] J.-W. Bian, H. Zhan, N. Wang, Z. Li, L. Zhang, C. Shen, M.-M. Cheng, and I. Reid. Unsupervised scale-consistent depth learning from video. *International Journal of Computer Vision (IJCV)*, 2021.

[20] S. Birchfield and C. Tomasi. Depth discontinuities by pixel-to-pixel stereo. In *Proceedings of the Sixth International Conference on Computer Vision*, pages 1073–1080. IEEE, 1998.

[21] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):401–406, 1998. doi: 10.1109/34.677269.

[22] Y. Cabon, N. Murray, and M. Humenberger. Virtual KITTI 2. *CoRR*, abs/2001.10773, 2020. URL `https://arxiv.org/abs/2001.10773`.

[23] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[24] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos, 2018. URL `https://arxiv.org/abs/1811.06152`.

[25] A. Cecille, S. Duffner, F. Davoine, T. Neveu, and R. Agier. Groco: Ground constraint for metric self-supervised monocular depth, 2024. URL `https://arxiv.org/abs/2409.14850`.

[26] A. Chakrabarti, J. Shao, and G. Shakhnarovich. Depth from a single image by harmonizing overcomplete local network predictions, 2016. URL `https://arxiv.org/abs/1605.07081`.

[27] J.-R. Chang and Y.-S. Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5410–5418, 2018.

[28] E. J. Chappero, R. A. Guerrero, and F. J. Seron. Vanishing point estimation from monocular images. In *9th International Information and Telecommunication Technologies Symposium (I2TS), Proceedings of the*, pages 177–182, 2010.

[29] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[30] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[31] P. Chen, W. Li, N. Gunderson, J. Ruthberg, R. Bly, W. M. Abuzeid, Z. Sun, and E. J. Seibel. Hybrid nerf-stereo vision: Pioneering depth estimation and 3d reconstruction in endoscopy, 2024. URL `https://arxiv.org/abs/2410.04041`.

[32] S. Chen, M. Tang, R. Dong, and J. Kan. Encoder–decoder structure fusing depth information for outdoor semantic segmentation. *Applied Sciences*, 13(17):9924, 2023.

[33] X. Cheng, P. Wang, and R. Yang. Hierarchical neural architecture search for deep stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9260–9269, 2020.

[34] 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. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages

3213–3223, 2016. URL https://api.semanticscholar.org/CorpusID:502946.

[35] P. Dendorfer, V. Yugay, A. Ošep, and L. Leal-Taixé. Quo vadis: Is trajectory forecasting the key towards long-term multi-object tracking?, 2022. URL https://arxiv.org/abs/2210.07681.

[36] Z. Deng, Y. Wu, W. Zhang, and Y. Zhang. Mvsnet+: Depth inference for unstructured multi-view stereo with multi-view feature aggregation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8111–8120, 2020.

[37] F. Dittrich, L. E. S. de Oliveira, A. S. B. Jr., and A. L. Koerich. People counting in crowded and outdoor scenes using an hybrid multi-camera approach. *CoRR*, abs/1704.00326, 2017. URL http://arxiv.org/abs/1704.00326.

[38] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[39] S. Duggal, S. Wang, W.-C. Ma, R. Hu, and R. Urtasun. Deeppruner: Learning efficient stereo matching via differentiable patchmatch. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4383–4392, 2019. doi: 10.1109/ICCV.2019.00448.

[40] A. Eftekhar, A. Sax, J. Malik, and A. Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10786–10796, 2021.

[41] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture, 2015. URL https://arxiv.org/abs/1411.4734.

[42] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2366–2374, Cambridge, MA, USA, 2014. MIT Press.

[43] A. Elaoua, M. Nadour, L. Cherroun, and A. Elasri. Real-time people counting system using yolov8 object detection. In *2023 2nd International Conference on Electronics, Energy and Measurement (IC2EM)*, volume 1, pages 1–5, 2023. doi: 10.1109/IC2EM59347.2023.10419684.

[44] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 261–268, 2006.

[45] Z. Feng, L. Yang, L. Jing, H. Wang, Y. Tian, and B. Li. Disentangling object motion and occlusion for unsupervised multi-frame monocular depth, 2022. URL https://arxiv.org/abs/2203.15174.

[46] T. Fridovich-Keil, Y. Wang, and et al. Plenoctrees for real-time rendering of neural radiance fields. In *arXiv preprint arXiv:2103.00020*, 2021.

[47] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010.

[48] T. Furuya, Y. Nakadate, and T. Saito. Recurrent depth refinement for multi-view stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1457–1465, 2020.

[49] R. Garg, V. K. BG, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue, 2016. URL `https://arxiv.org/abs/1603.04992`.

[50] S. Gasperini, N. Morbitzer, H. Jung, N. Navab, and F. Tombari. Robust monocular depth estimation under challenging conditions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8177–8186, October 2023.

[51] 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.

[52] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[53] C. Godard, O. M. Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6602–6611, 2017. doi: 10.1109/CVPR.2017.699.

[54] 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.

[55] C. Godard, O. M. Aodha, and G. J. Brostow. Digging into self-supervised monocular depth estimation. *CoRR*, abs/1806.01260, 2018. URL `http://arxiv.org/abs/1806.01260`.

[56] H. Goldstein, C. Poole, and J. Safko. *Classical Mechanics*. Addison Wesley, 3rd edition edition, 2002. ISBN 978-0201657029.

[57] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks, 2014. URL `https://arxiv.org/abs/1406.2661`.

[58] J. Gräter, T. Schwarze, and M. Lauer. Robust scale estimation for monocular visual odometry using structure from motion and vanishing points. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 475–480, 2015. doi: 10.1109/IVS.2015.7225730.

[59] V. Guizilini, R. Ambruş, D. Chen, S. Zakharov, and A. Gaidon. Multi-frame self-supervised depth with transformers. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 160–170, 2022. doi: 10.1109/CVPR52688.2022.00026.

[60] V. Guizilini, I. Vasiljevic, D. Chen, R. Ambruș, and A. Gaidon. Towards zero-shot scale-aware monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9233–9243, October 2023.

[61] W. Han, J. Yin, and J. Shen. Self-supervised monocular depth estimation by direction-aware cumulative convolution network. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8579–8589, 2023. doi: 10.1109/ICCV51070.2023.00791.

[62] 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.

[63] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008. doi: 10.1109/TPAMI.2007.1166.

[64] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[65] D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. *ACM Trans. Graph.*, 24(3):577–584, July 2005. ISSN 0730-0301. doi: 10.1145/1073204.1073232. URL `https://doi.org/10.1145/1073204.1073232`.

[66] W. Hsu, S. Kuo, and M. Yang. Deepmvs: Learning multi-view stereo with multi-scale deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1524–1533, 2018.

[67] M. Hu, W. Yin, C. Zhang, Z. Cai, X. Long, H. Chen, K. Wang, G. Yu, C. Shen, and S. Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *arXiv preprint arXiv:2404.15506*, 2024.

[68] Y. Hua and H. Tian. Depth estimation with convolutional conditional random field network. *Neurocomputing*, 214:546–554, 2016. URL `https://api.semanticscholar.org/CorpusID:34719331`.

[69] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks, 2018. URL `https://arxiv.org/abs/1608.06993`.

[70] C. Hwang, W. Cui, Y. Xiong, Z. Yang, Z. Liu, H. Hu, Z. Wang, R. Salas, J. Jose, P. Ram, J. Chau, P. Cheng, F. Yang, M. Yang, and Y. Xiong. Tutel: Adaptive mixture-of-experts at scale, 2022.

[71] H. Ibrahem, A. Salem, and H.-S. Kang. Seg2depth: Semi-supervised depth estimation for autonomous vehicles using semantic segmentation and single vanishing point fusion. *IEEE Transactions on Intelligent Vehicles*, pages 1–11, 2024. doi: 10.1109/TIV.2024.3370930.

[72] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks, 2016. URL https://arxiv.org/abs/1612.01925.

[73] S. Im, H.-G. Jeon, S. Lin, and I. S. Kweon. Dpsnet: End-to-end deep plane sweep stereo, 2019. URL https://arxiv.org/abs/1905.00538.

[74] A. Jan and S. Seo. Monocular depth estimation using res-unet with an attention model. *Applied Sciences*, 13(10):6319, 2023.

[75] H. Jiang, L. Ding, Z. Sun, and R. Huang. Unsupervised monocular depth perception: Focusing on moving objects. *IEEE Sensors Journal*, 21(24):27225–27237, 2021. doi: 10.1109/JSEN.2021.3109266.

[76] J. Jiao, Y. Cao, Y. Song, and R. Lau. Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[77] H. Jung, Y. Kim, D. Min, C. Oh, and K. Sohn. Depth prediction from a single image with conditional adversarial networks. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 1717–1721, 2017. doi: 10.1109/ICIP.2017.8296575.

[78] S.-M. Jung and T.-K. WhangBo. Improved depth map generation using motion vector and the vanishing point from a moving camera monocular image. In Y.-S. Jeong, Y.-H. Park, C.-H. R. Hsu, and J. J. H. Park, editors, *Ubiquitous Information Technologies and Applications*, pages 725–734, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg. ISBN 978-3-642-41671-2.

[79] K. Karsch, C. Liu, and S. B. Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2144–2158, 2014. doi: 10.1109/TPAMI.2014.2316835.

[80] K. Karsch, C. Liu, and S. B. Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2144–2158, 2014. doi: 10.1109/TPAMI.2014.2316835.

[81] B. Ke, A. Obukhov, S. Huang, N. Metzger, R. C. Daudt, and K. Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[82] A. Kendall, H. Martirosyan, et al. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 66–75, 2017.

[83] J. Kim, J. Lee, U. Shin, J. Oh, and K. Joo. Vpocc: Exploiting vanishing point for monocular 3d semantic occupancy prediction, 2024. URL https://arxiv.org/abs/2408.03551.

[84] D. P. Kingma and M. Welling. Auto-encoding variational bayes, 2022. URL https://arxiv.org/abs/1312.6114.

[85] G. Kinoshita and K. Nishino. Camera height doesn't change: Unsupervised training for metric monocular road-scene depth estimation, 2024. URL https://arxiv.org/abs/2312.04530.

[86] M. Klingner, J.-A. Termöhlen, J. Mikolajczyk, and T. Fingscheidt. Self-Supervised Monocular Depth Estimation: Solving the Dynamic Object Problem by Semantic Guidance. In *European Conference on Computer Vision (ECCV)*, 2020.

[87] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *Proceedings of the Eighth IEEE International Conference on Computer Vision (ICCV)*, pages 508–515. IEEE, 2001.

[88] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, May 2017. ISSN 0001-0782. doi: 10.1145/3065386. URL https://doi.org/10.1145/3065386.

[89] A. C. Kumar, S. M. Bhandarkar, and M. Prasad. Depthnet: A recurrent neural network architecture for monocular depth prediction. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 396–3968, 2018. doi: 10.1109/CVPRW.2018.00066.

[90] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks, 2016. URL https://arxiv.org/abs/1606.00373.

[91] S. Lee, S. Im, S. Lin, and I. S. Kweon. Learning monocular depth in dynamic scenes via instance-aware projection consistency, 2021. URL https://arxiv.org/abs/2102.02629.

[92] B. Li, C. Shen, Y. Dai, A. van den Hengel, and M. He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1119–1127, 2015. doi: 10.1109/CVPR.2015.7298715.

[93] B. Li, B. Liu, M. Zhu, X. Luo, and F. Zhou. Image intrinsic-based unsupervised monocular depth estimation in endoscopy. *IEEE Journal of Biomedical and Health Informatics*, pages 1–11, 2024. doi: 10.1109/JBHI.2024.3400804.

[94] J. Li, P. Wang, P. Xiong, T. Cai, Z. Yan, L. Yang, J. Liu, H. Fan, and S. Liu. Practical stereo matching via cascaded recurrent network with adaptive correlation, 2022. URL https://arxiv.org/abs/2203.11483.

[95] R. Li, D. Gong, W. Yin, H. Chen, Y. Zhu, K. Wang, X. Chen, J. Sun, and Y. Zhang. Learning to fuse monocular and multi-view cues for multi-frame depth estimation in dynamic scenes. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21539–21548, 2023. doi: 10.1109/CVPR52729.2023.02063.

[96] Z. Li and N. Snavely. Megadepth: Learning single-view depth prediction from internet photos. *CoRR*, abs/1804.00607, 2018. URL `http://arxiv.org/abs/1804.00607`.

[97] M. Liang, B. Yang, S. Wang, and R. Urtasun. Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 214–230, 2018.

[98] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[99] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[100] Y. Lin, R. Wiersma, S. L. Pintea, K. Hildebrandt, E. Eisemann, and J. C. van Gemert. Deep vanishing point detection: Geometric priors make dataset variations vanish. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6103–6113, June 2022.

[101] T. Lindeberg. Scale invariant feature transform. *Scholarpedia*, 7:10491, 2012. URL `https://api.semanticscholar.org/CorpusID:5900376`.

[102] B. Liu, S. Gould, and D. Koller. Single image depth estimation from predicted semantic labels. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1253–1260, 2010. doi: 10.1109/CVPR.2010.5539823.

[103] F. Liu, C. Shen, G. Lin, and I. Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(10):2024–2039, 2016. doi: 10.1109/TPAMI.2015.2505283.

[104] F. Liu, C. Shen, G. Lin, and I. Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(10):2024–2039, Oct. 2016. ISSN 2160-9292. doi: 10.1109/tpami.2015.2505283. URL `http://dx.doi.org/10.1109/TPAMI.2015.2505283`.

[105] F. Liu, C. Shen, G. Lin, and I. Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(10):2024–2039, 2016. doi: 10.1109/TPAMI.2015.2505283.

[106] Z. Liu and F. Zhang. Balm: Bundle adjustment for lidar mapping. *IEEE Robotics and Automation Letters*, 6(2):3184–3191, 2021. doi: 10.1109/LRA.2021.3062815.

[107] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, F. Wei, and B. Guo. Swin transformer v2: Scaling up capacity and resolution. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[108] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[109] A. Lopez-Rodriguez and K. Mikolajczyk. Desc: Domain adaptation for depth estimation via semantic consistency. *International Journal of Computer Vision*, 131(3):752–771, 2023.

[110] C. Luo, Z. Yang, P. Wang, Y. Wang, W. Xu, R. Nevatia, and A. Yuille. Every pixel counts ++: Joint learning of geometry and motion with 3d holistic understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2624–2641, 2020. doi: 10.1109/TPAMI.2019.2930258.

[111] W. Luo, A. G. Schwing, and R. Urtasun. Efficient deep learning for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5695–5703, 2016.

[112] X. Lyu, L. Liu, M. Wang, X. Kong, L. Liu, Y. Liu, X. Chen, and Y. Yuan. Hr-depth: High resolution self-supervised monocular depth estimation, 2020. URL `https://arxiv.org/abs/2012.07356`.

[113] R. Mahjourian, M. Wicke, and A. Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5667–5675, 2018. doi: 10.1109/CVPR.2018.00594.

[114] M. Mahyoub, F. Natalia, S. Sudirman, A. H. J. Al-Jumaily, and P. Liatsis. Semantic segmentation and depth estimation of urban road scene images using multi-task networks. In *2023 15th International Conference on Developments in eSystems Engineering (DeSE)*, pages 469–474, 2023. doi: 10.1109/DeSE58274.2023.10099504.

[115] M. Mancini, G. Costante, P. Valigi, T. A. Ciarfuglia, J. Delmerico, and D. Scaramuzza. Toward domain independence for learning-based monocular depth estimation. *IEEE Robotics and Automation Letters*, 2(3):1778–1785, 2017. doi: 10.1109/LRA.2017.2657002.

[116] S. Mattoccia, F. Tombari, and L. Di Stefano. Fast full-search equivalent template matching by enhanced bounded correlation. *IEEE Transactions on Image Processing*, 17(4):528–538, 2008. doi: 10.1109/TIP.2008.919362.

[117] N. Mayer, E. Ilg, P. Häusser, et al. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4040–4048, 2016.

[118] R. McCraith, L. Neumann, and A. Vedaldi. Calibrating self-supervised monocular depth estimation, 2021. URL `https://arxiv.org/abs/2009.07714`.

[119] M. J. Mcdonnell. Box-filtering techniques. *Computer Graphics and Image Processing*, 17:65–70, 1981. URL https://api.semanticscholar.org/CorpusID:62639635.

[120] Y. Meng, Y. Lu, A. Raj, S. Sunarjo, R. Guo, T. Javidi, G. Bansal, and D. Bharadia. Signet: Semantic instance aided unsupervised 3d geometry perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[121] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[122] B. Mildenhall, A. Neraditskiy, J. Pritts, and et al. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the European Conference on Computer Vision*, pages 113–129, 2020.

[123] A. Mingozzi, A. Conti, F. Aleotti, M. Poggi, and S. Mattoccia. Monitoring social distancing with single image depth estimation, 2022. URL https://arxiv.org/abs/2204.01693.

[124] J. Moon, J. L. G. Bello, B. Kwon, and M. Kim. From-ground-to-objects: Coarse-to-fine self-supervised monocular depth estimation of dynamic objects with ground contact prior, 2023. URL https://arxiv.org/abs/2312.10118.

[125] S. Mukhopadhyay, M. Gwilliam, V. Agarwal, N. Padmanabhan, A. Swaminathan, S. Hegde, T. Zhou, and A. Shrivastava. Diffusion models beat gans on image classification, 2023. URL https://arxiv.org/abs/2307.08702.

[126] T. Naderi, A. Sadovnik, J. Hayward, and H. Qi. Monocular depth estimation with adaptive geometric attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 944–954, January 2022.

[127] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[128] A. Nichol and P. Dhariwal. Improved denoising diffusion probabilistic models, 2021. URL https://arxiv.org/abs/2102.09672.

[129] J. Pang, W. Sun, J. S. J. Ren, C. Yang, and Q. Yan. Cascade residual learning: A two-stage convolutional neural network for stereo matching. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 878–886, 2017. doi: 10.1109/ICCVW.2017.111. URL https://ieeexplore.ieee.org/document/8265317.

[130] L. Piccinelli, Y.-H. Yang, C. Sakaridis, M. Segu, S. Li, L. Van Gool, and F. Yu. Unidepth: Universal monocular metric depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10106–10116, June 2024.

[131] M. Poggi, F. Tosi, K. Batsos, P. Mordohai, and S. Mattoccia. On the synergies between machine learning and binocular stereo for depth estimation from images: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5314–5334, 2021.

[132] M. Poggi, F. Tosi, K. Batsos, P. Mordohai, and S. Mattoccia. On the synergies between machine learning and binocular stereo for depth estimation from images: a survey, 2021. URL `https://arxiv.org/abs/2004.08566`.

[133] X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 283–291, 2018. doi: 10.1109/CVPR.2018.00037.

[134] X. Qiao, M. Poggi, P. Deng, H. Wei, C. Ge, and S. Mattoccia. Rgb guided tof imaging system: A survey of deep learning-based methods. *International Journal of Computer Vision*, pages 1–38, 2024.

[135] M. Ramamonjisoa, Y. Du, and V. Lepetit. Predicting sharp and accurate occlusion boundaries in monocular depth estimation using displacement fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[136] P. Z. Ramirez, M. Poggi, F. Tosi, S. Mattoccia, and L. D. Stefano. Geometry meets semantics for semi-supervised monocular depth estimation, 2018. URL `https://arxiv.org/abs/1810.04093`.

[137] R. Ranftl, A. Bochkovskiy, and V. Koltun. Vision transformers for dense prediction. *ICCV*, 2021.

[138] 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*, 44(3), 2022.

[139] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[140] R. Revalski and et al. Kilonerf: Speeding up neural radiance fields with a multi-scale architecture. In *arXiv preprint arXiv:2103.00040*, 2021.

[141] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. URL `http://arxiv.org/abs/1505.04597`.

[142] S. Safadoust and F. Güney. Depthp+p: Metric accurate monocular depth estimation using planar and parallax, 2023. URL `https://arxiv.org/abs/2301.02092`.

[143] K. Saunders, G. Vogiatzis, and L. J. Manso. Dyna-dm: Dynamic object-aware self-supervised monocular depth maps, 2023. URL `https://arxiv.org/abs/2206.03799`.

[144] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Proceedings of the 18th International Conference on Neural Information Processing Systems*, NIPS'05, page 1161–1168, Cambridge, MA, USA, 2005. MIT Press.

[145] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Proceedings of the 18th International Conference on Neural Information Processing Systems*, NIPS'05, page 1161–1168, Cambridge, MA, USA, 2005. MIT Press.

[146] A. Saxena, S. H. Chung, and A. Ng. 3-d depth reconstruction from a single still image. *International Journal of Computer Vision*, 76:53–69, 2007. URL `https://api.semanticscholar.org/CorpusID:9620866`.

[147] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5): 824–840, 2009. doi: 10.1109/TPAMI.2008.132.

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

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

[150] 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.

[151] J. L. Schönberger and J. M. Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016.

[152] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. URL `https://arxiv.org/abs/1409.1556`.

[153] S. Singh and A. Kaur. Game development using unity game engine. In *2022 3rd International Conference on Computing, Analytics and Networks (ICAN)*, pages 1–6, 2022. doi: 10.1109/ICAN56228.2022.10007155.

[154] W. Su and W. Tao. Efficient edge-preserving multi-view stereo network for depth estimation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(2):2348–2356, Jun. 2023. doi: 10.1609/aaai.v37i2.25330. URL `https://ojs.aaai.org/index.php/AAAI/article/view/25330`.

[155] J. Sun, H.-Y. Shum, and N.-N. Zheng. Stereo matching using belief propagation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 510–517. IEEE, 2003.

[156] L. Sun, J.-W. Bian, H. Zhan, W. Yin, I. Reid, and C. Shen. Sc-depthv3: Robust self-supervised monocular depth estimation for dynamic scenes, 2023. URL `https://arxiv.org/abs/2211.03660`.

[157] Y. Sun and B. Hariharan. Dynamo-depth: Fixing unsupervised depth estimation for dynamical scenes. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[158] C. Sutton and A. McCallum. An introduction to conditional random fields, 2010. URL `https://arxiv.org/abs/1011.4088`.

[159] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):1068–1080, 2008.

[160] Z. Teed and J. Deng. Raft: Recurrent all-pairs field transforms for optical flow, 2020. URL `https://arxiv.org/abs/2003.12039`.

[161] C. Tian, W. Pan, Z. Wang, M. Mao, G. Zhang, H. Bao, P. Tan, and Z. Cui. Dps-net: Deep polarimetric stereo depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3569–3579, 2023.

[162] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proceedings of the Sixth International Conference on Computer Vision*, pages 839–846. IEEE, 1998.

[163] F. Tombari, L. di Stefano, S. Mattoccia, and A. Galanti. Performance evaluation of robust matching measures. In *International Conference on Computer Vision Theory and Applications*, 2008. URL `https://api.semanticscholar.org/CorpusID:13028327`.

[164] F. Tombari, S. Mattoccia, and L. Di Stefano. Full-search-equivalent pattern matching with incremental dissimilarity approximations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):129–141, 2009. doi: 10.1109/TPAMI.2008.46.

[165] A. Tonioni, M. Poggi, S. Mattoccia, et al. Real-time self-adaptive deep stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 195–204, 2019.

[166] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497, 2015. doi: 10.1109/ICCV.2015.510.

[167] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger. Sparsity invariant cnns, 2017. URL `https://arxiv.org/abs/1708.06500`.

[168] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL `http://arxiv.org/abs/1706.03762`.

[169] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023. URL `https://arxiv.org/abs/1706.03762`.

[170] O. Veksler. Stereo correspondence by dynamic programming on a tree. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 384–390 vol. 2, 2005. doi: 10.1109/CVPR.2005.334.

[171] V. Verma, L. Xie, J. Zhang, L. Xu, and T. Wang. Neural surface reconstruction with multi-view consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 974–982, 2019.

[172] C. Wang, S. Lucey, F. Perazzi, and O. Wang. Web stereo video supervision for depth prediction from dynamic scenes, 2019. URL `https://arxiv.org/abs/1904.11112`.

[173] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. Yuille. Towards unified depth and semantic prediction from a single image. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2800–2809, 2015. doi: 10.1109/CVPR.2015.7298897.

[174] P. Wang, Z. Fang, S. Zhao, Y. Chen, M. Zhou, and S. An. Vanishing point aided lidar-visual-inertial estimator. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13120–13126, 2021. doi: 10.1109/ICRA48506.2021.9561087.

[175] Q. Wang and Y. Piao. Depth estimation of supervised monocular images based on semantic segmentation. *Journal of Visual Communication and Image Representation*, 90:103753, 2023. ISSN 1047-3203. doi: https://doi.org/10.1016/j.jvcir.2023.103753. URL `https://www.sciencedirect.com/science/article/pii/S1047320323000032`.

[176] R. Wang, D. Geraghty, K. Matzen, R. Szeliski, and J.-M. Frahm. Vplnet: Deep single view normal estimation with vanishing points and lines. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[177] R. Wang, Z. Yu, and S. Gao. Planedepth: Self-supervised depth estimation via orthogonal planes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21425–21434, June 2023.

[178] R. Wang, Z. Yu, and S. Gao. Planedepth: Self-supervised depth estimation via orthogonal planes, 2023. URL `https://arxiv.org/abs/2210.01612`.

[179] W. Wang, Y. Hu, and S. Scherer. Tartanvo: A generalizable learning-based vo, 2020. URL `https://arxiv.org/abs/2011.00359`.

[180] X. Wang, D. Fouhey, and A. Gupta. Designing deep networks for surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[181] X. Wang, Z. Zhu, G. Huang, X. Chi, Y. Ye, Z. Chen, and X. Wang. Crafting monocular cues and velocity guidance for self-supervised multi-frame depth learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(3):2689–2697, Jun. 2023. doi: 10.1609/aaai.v37i3.25368. URL `https://ojs.aaai.org/index.php/AAAI/article/view/25368`.

[182] Y. Wang and et al. Deepvoxels: 3d scene reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 403–412, 2020.

[183] Y. Wang, B. Curless, and S. Seitz. People as scene probes, 2020. URL https://arxiv.org/abs/2007.09209.

[184] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. doi: 10.1109/TIP.2003.819861.

[185] J. Watson, O. M. Aodha, V. Prisacariu, G. Brostow, and M. Firman. The Temporal Opportunist: Self-Supervised Multi-Frame Monocular Depth. In *Computer Vision and Pattern Recognition (CVPR)*, 2021.

[186] C. Won, J. Ryu, and J. Lim. End-to-end learning for omnidirectional stereo matching with uncertainty prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3850–3862, 2021. doi: 10.1109/TPAMI.2020.2992497.

[187] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[188] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Advances in neural information processing systems*, volume 34, pages 12077–12090, 2021.

[189] F. Xue, G. Zhuo, Z. Huang, W. Fu, Z. Wu, and M. H. A. J. au2. Toward hierarchical self-supervised monocular absolute depth estimation for autonomous driving applications, 2020. URL https://arxiv.org/abs/2004.05560.

[190] H. Yan, S. Zhang, Y. Zhang, and L. Zhang. Monocular depth estimation with guidance of surface normal map. *Neurocomputing*, 280:86–100, 2018. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2017.08.074. URL https://www.sciencedirect.com/science/article/pii/S092523121731771X. Applications of Neural Modeling in the new era for data and IT.

[191] J. Yan, H. Zhao, P. Bu, and Y. Jin. Channel-wise attention-based network for self-supervised monocular depth estimation, 2021. URL https://arxiv.org/abs/2112.13047.

[192] G. Yang, X. Song, C. Huang, Z. Deng, J. Shi, and B. Zhou. Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[193] G. Yang, L. Xu, and Y. Zhang. Depth supervision networks for multi-view depth estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7898–7907, 2019.

[194] G. Yang, S. Sun, Y. Zhang, and L. Xu. Pvdnet: Multi-view depth estimation with point-wise volumetric depth network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3640–3649, 2020.

[195] J. Yang, W. Mao, J. M. Álvarez, and M. Liu. Cost volume pyramid based depth inference for multi-view stereo. *CoRR*, abs/1912.08329, 2019. URL `http://arxiv.org/abs/1912.08329`.

[196] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024.

[197] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao. Depth anything v2. *arXiv:2406.09414*, 2024.

[198] Q. Yang, L. Wang, and N. A. Yang. A constant-space belief propagation algorithm for stereo matching. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1458–1465, 2010.

[199] X. Yang, Z. Ma, Z. Ji, and Z. Ren. Gedepth: Ground embedding for monocular depth estimation, 2023. URL `https://arxiv.org/abs/2309.09975`.

[200] X. Yang, Z. Ma, Z. Ji, and Z. Ren. Gedepth: Ground embedding for monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12719–12727, October 2023.

[201] Y. Yang, T. Wang, Z. Li, and L. Zhang. Holodepth: Depth estimation for holographic displays. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12357–12366, 2020.

[202] Z. Yang, P. Wang, W. Xu, L. Zhao, and R. Nevatia. Unsupervised learning of geometry with edge-aware depth-normal consistency, 2017. URL `https://arxiv.org/abs/1711.03665`.

[203] Y. Yao, Z. Luo, S. Li, et al. Mvsnet: Depth inference for multi-view stereo with deep learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018.

[204] Y. Yao, Z. Luo, S. Li, T. Shen, T. Fang, and L. Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5520–5529, 2019. URL `https://api.semanticscholar.org/CorpusID:67855970`.

[205] W. Yin, X. Wang, C. Shen, Y. Liu, Z. Tian, S. Xu, C. Sun, and D. Renyin. Diversedepth: Affine-invariant depth prediction using diverse data. *ArXiv*, abs/2002.00569, 2020. URL `https://api.semanticscholar.org/CorpusID:211010487`.

[206] 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 *Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (CVPR)*, 2021.

[207] W. Yin, J. Zhang, O. Wang, S. Niklaus, S. Chen, Y. Liu, and C. Shen. Towards accurate reconstruction of 3d scene shape from a single monocular image. *TPAMI*, 2022.

[208] Z. Yin and J. Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1983–1992, 2018. doi: 10.1109/CVPR.2018.00212.

[209] Z. Yin and J. Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1983–1992, 2018. doi: 10.1109/CVPR.2018.00212.

[210] K.-J. Yoon and I.-S. Kweon. Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):650–656, 2006.

[211] F. Yu, V. Koltun, and T. Funkhouser. Dilated residual networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 636–644, 2017. doi: 10.1109/CVPR.2017.75.

[212] L. Yu, Y. Wang, Y. Wu, and Y. Jia. Deep stereo matching with explicit cost aggregation sub-architecture, 2018. URL `https://arxiv.org/abs/1801.04065`.

[213] H. Yuan, T. Chen, W. Sui, J. Xie, L. Zhang, Y. Li, and Q. Zhang. Monocular road planar parallax estimation. *IEEE Transactions on Image Processing*, 32:3690–3701, 2023. ISSN 1941-0042. doi: 10.1109/tip.2023.3289323. URL `http://dx.doi.org/10.1109/TIP.2023.3289323`.

[214] W. Yuan, X. Gu, Z. Dai, S. Zhu, and P. Tan. New crfs: Neural window fully-connected crfs for monocular depth estimation, 2022. URL `https://arxiv.org/abs/2203.01502`.

[215] M. Yue, G. Fu, M. Wu, X. Zhang, and H. Gu. Self-supervised monocular depth estimation in dynamic scenes with moving instance loss. *Engineering Applications of Artificial Intelligence*, 112:104862, 2022. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2022.104862. URL `https://www.sciencedirect.com/science/article/pii/S0952197622001105`.

[216] Z. K. Z. et al. Stereo matching by training a convolutional neural network to compare image patches. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[217] J. Zbontar and Y. LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17(1):2287–2318, 2016.

[218] L. Zhang, Z. Cao, X. Meng, C. Zhou, and S. Wang. Real-time depth-based tracking using a binocular camera. In *2016 12th World Congress on Intelligent Control and Automation (WCICA)*, pages 2041–2046, 2016. doi: 10.1109/WCICA.2016.7578274.

[219] M. Zhang, X. Ye, X. Fan, and W. Zhong. Unsupervised depth estimation from monocular videos with hybrid geometric-refined loss and contextual attention. *Neurocomputing*, 379:250–261, 2020. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2019.10.107. URL https://www.sciencedirect.com/science/article/pii/S0925231219315516.

[220] N. Zhang, F. Nex, G. Vosselman, and N. Kerle. Lite-mono: A lightweight cnn and transformer architecture for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18537–18546, June 2023.

[221] X. Zhang, J. Wei, A. Moteki, Y. Kobayashi, G. Suzuki, and Z. Tan. Msd-crfs: Multi-scale dual aggregation conditional random fields for monocular depth estimation. In *2024 IEEE International Conference on Image Processing (ICIP)*, pages 2001–2007, 2024. doi: 10.1109/ICIP51287.2024.10647895.

[222] Y. Zhang, Z. Yang, R. Yao, et al. Ganet: Guided aggregation network for end-to-end stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 185–194, 2019.

[223] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 666–673 vol.1, 1999. doi: 10.1109/ICCV.1999.791289.

[224] Z. Zhang, R. Peng, Y. Hu, and R. Wang. Geomvsnet: Learning multi-view stereo with geometry perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21508–21518, June 2023.

[225] B. Zhao, Y. Huang, W. Ci, and X. Hu. Unsupervised learning of monocular depth and ego-motion with optical flow features and multiple constraints. *Sensors*, 22(4):1383, 2022.

[226] C. Zhao, Y. Zhang, M. Poggi, F. Tosi, X. Guo, Z. Zhu, G. Huang, Y. Tang, and S. Mattoccia. Monovit: Self-supervised monocular depth estimation with a vision transformer. In *2022 International Conference on 3D Vision (3DV)*. IEEE, Sept. 2022. doi: 10.1109/3dv57658.2022.00077. URL http://dx.doi.org/10.1109/3DV57658.2022.00077.

[227] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6612–6619, 2017. doi: 10.1109/CVPR.2017.700.

[228] X. Zhou, G. Huang, Y. Liu, and Y. Miao. Self-supervised multi-view depth estimation from a single image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1733–1742, 2019.

[229] Y. Zhou, Q. Liu, H. Zhu, Y. Li, S. Chang, and M. Guo. Mogde: Boosting mobile monocular 3d object detection with ground depth estimation, 2023. URL https://arxiv.org/abs/2303.13561.

[230] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 989–997, 2019. doi: 10.1109/CVPR.2019.00108.