Inferring the 3D information from the Outside World Using Monocular Cameras

Haotian Zhang

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2022

Reading Committee:
Jenq-Neng Hwang, Chair
Radha Poovendran
Blake Hannaford

Program Authorized to Offer Degree:
Electrical and Computer Engineering

University of Washington

Abstract

Inferring the 3D information from the Outside World Using Monocular Cameras

Haotian Zhang

Chair of the Supervisory Committee:
Professor Jenq-Neng Hwang
Electrical and Computer Engineering

Technological advances have made autonomous driving more and more feasible in common driving scenarios. Many large companies such as Waymo, Tesla, GM, and Uber have tested their self-driving vehicles with success in limited capacities. These vehicles employ a combination of camera, radar, sonar, and LiDAR sensors. Yet the high cost of LiDAR as well as the unreliability of sonar and radar makes them unsuitable for quick large-scale deployment. On the contrary, camera-based autonomous driving has the potential to be a cheap and reliable alternative through steadily advancing computer vision and deep learning techniques. A general autonomous driving system incorporates three correlated technologies: 3D-based object detection, tracking, and localization.

While all three components are important, most relevant papers tend to only focus on one single component. In this work, we first propose a *multi-stage* monocular vision-based framework for 3D-based detection, tracking, and localization by effectively integrating all three tasks in a complementary manner. Our system contains an RCNN-based **Localization Network** (LOCNet), which works in concert with fitness evaluation score (FES) based single-frame optimization, to get more accurate and refined 3D vehicle localization. To better utilize the temporal information, we further use a multi-frame optimization technique, taking advantage of camera ego-motion and a 3D TrackletNet Tracker (3D TNT), to improve both accuracy and consistency in our 3D localization.

Moreover, we propose a *joint* framework (JMV3D) that can effectively associates moving objects over time and estimate their 3D localization information as well as segmentation masks from

a sequence of 2D images so as to compensate for the individual drawbacks of each component. We further extend the existing Localization Network (LOCNet) to become **Localization for Tracking Network** (Loc4Trk-Net). A spatial Attention (SA) Neck is added to highlight the foreground (target of interest) and suppress the background with the help of mask segmentation, so that more concentrated appearance features can be obtained. Besides, one additional embedding head is introduced to train a discriminative feature embeddings to leverage deep pairwise contrastive learning and identify objects in various poses and viewpoints with appearance cues. Then, a straightforward combination of a 3D Kalman filter and the Hungarian algorithm is further utilized for robust instance association via both feature similarity and 3D localization information. Overall, both systems outperform the state-of-the-art image-based solutions in diverse scenarios and is even comparable with LiDAR-based methods. The proposed JMV3D pipeline also ranks 1^{st} place on the KITTI-MOTS & KITT-STEP leaderboards and also achieves impressive results among all image-based solutions on nuScenes 3D tracking benchmark.

Futhermore, *monocular 3D object detection* requires decoding 3D predictions solely from a single 2D image. However, by formulating this problem as a region-level understanding task, previous approaches neglect the image-level understanding of depth and semantics. To address this, we present the **Mono**cular **3D** object detection via **C**oarse-to-**F**ine **T**raining (Mono3DCFT), a new transformer-based architecture with an effective two-stage training strategy that can seamlessly handle both levels of tasks: (i) coarse-grained training on the whole image based on monocular depth data; followed by (ii) fine-grained training on specific regions based on 3D bounding boxes annotations. Instead of having dedicated transformer layers for fusion after the uni-modal backbone, Mono3DCFT pushes multi-modal cross-attention fusion into both the vision and depth backbones and achieves significant gains on the KITTI benchmark coupled with two-stage training. Trained solely based on limited publicly available KITTI depth data, our Mono3DCFT performs comparably against the previous best state-of-the-art, which is pre-trained on 15M additional proprietary depth data along with a more compute-intensive architecture. Extensive ablation studies demonstrate

the effectiveness of our approach and its potential to serve as a transformer baseline for future monocular 3D monocular object detection.

The expected contributions of this thesis can be concluded as follows:

- An RCNN-based LOCNet is proposed to simultaneously regress both the 3D orientation and distance of vehicles, which can serve as a good initialization for follow-up optimizations.

- A single-frame optimization technique based on the fitness evaluation score (FES) is applied to ensure the object spatial robustness in the 3D localization.

- We further extend the existing LOCNet to become Loc4Trk-Net. Instance specific features, which are learned jointly with the detection task, utilize the instance masks as spatial attention to emphasize the target of interest explicitly.

- 3D object tracking utilizes the jointly learned instance-aware feature via pairwise contrastive learning and localization information. A straightforward combination of a 3D Kalman filter and the Hungarian algorithm is used for online state estimation and robust data association.

- A novel framework, Mono3DCFT, which consists of a fusion-in-the-backbone encoder and a depth-guided decoder to enable object queries, can adaptively collect rich geometric and appearance information of the scene, resulting in better scene depth estimation to assist object 3D attribute prediction.

- The proposed two-stage coarse-to-fine training on whole sense depth map data, followed by object-wise depth labels, can better capture both scene-level depth cues and region-level appearance cues.

- Experimental results on the KITTI dataset show that our proposed Mono3DCFT achieves near SOTA performance among monocular-based methods with significant gains without extra out-of-domain depth data.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

It couldn't be more enjoyable! I cherish every moment of the five-year graduate study pursuing a Ph.D. with my advisor, Prof. Jenq-Neng Hwang. I came to UW with a deep fascination with computer vision and autonomous systems. Information Processing Lab (IPL) turned out to be the perfect place to fulfill my dream. Surrounded by inspiring faculty, brilliant colleagues, and lovely friends, there is no better place I could imagine.

There are so many things I'd like to thank my advisor, Prof. Jenq-Neng Hwang. He means much more than a perfect advisor to me but as a father. I am still very grateful for his persuading me to pursue a Ph.D. at the time I was enrolled as a Master's student in the BS-MS program. As I look back, this is one of those rare occasions that changes one's life, inviting me onto the journey of keeping chasing my passion in the field at a time when I wasn't sure I could do it. I would also like to thank him for being a great advisor, inspiring, super supportive, and encouraging me throughout the years. I learned so much about traditional 3D geometry, 2D/3D object detection, and multi-object tracking from Prof. Hwang, and it became the key element in this Ph.D. thesis. I'm so grateful to have had the opportunity to work with him, and I'm really excited to see our group growing so much these years, see Figure 1.

I would also like to thank Prof. Radha Proovendran, who is on my graduate committee and also a friend. I learned a lot from him. He not only gave me advice on academic writing and presentation but also backed me up with encouragement. I would also like to thank the rest of my Ph.D. supervisory committee members, Prof. Blake Hannaford and Prof. Yen-Chi Chen, for their time and insightful discussion. Words cannot express my appreciation.

I also want to thank Prof. Hui Liu. Without his support and reference, I would not have obtained the opportunity to study at UW.

Figure 1: The amazing IPL's research group. (This photo was taken at Prof. Hwang's house with family and friends on 2019 Thanksgiving Day.)

Apart from the fantastic faculty, I have worked with so many brilliant colleagues. Yizhou Wang, thank you for teaching me so much about Pycharm and Overleaf techniques and being a good friend to chat with and hang out with. Best of luck in your new job at XPENG after graduation! Dr. Zheng (Thomas) Tang and Dr. Gaoang Wang, as former IPL members, thank you for leading the research success of IPL and introducing me to the world of object detection and tracking. Hung-Min, thank you for staying late at night together in lab debugging, grabbing fries and fried chicken wings at 2 a.m. I extend my thanks to all other current or former IPL members, including but not limited to Dr. Tsung-Wei Huang, Dr. Renshu Gu, Dr. Jiarui Cai, Dr. Pyong-Kun Kim, Jie Mei, Zhongyu Jiang,

Cheng-Yen (Chris) Yang, Yin Jin, Aotian Zheng, Jen-Hao (Andy) Cheng, Yudong Li, Hsiang-Wei Huang, and Samartha Ramkumar, for their support and friendship.

Last but not least, thanks all to my family, who made my years in Seattle much more colorful! Deepest gratitude to my mom and dad - I couldn't have made it without your help, and thank you for the support and care through an emotional and rather stressful peak of my life. Special thanks to my girlfriend, Yifeng, for all the great memories of us exploring cities, hiking in the mountains, and diving in the oceans, and most importantly, for keeping me alive and smiling :)

DEDICATION

to my Mom, Dad, and Yifeng

For always encouraging me to pursue my dreams

and for always be my side :)

Chapter 1

INTRODUCTION

1.1 Monocular 3D Localization of Vehicles in Road Scenes

A general autonomous driving system incorporates three correlated technologies: 3D-based object detection, tracking, and localization. Currently, these three components are explored separately, and work has rarely been done to effectively combine them all so as to compensate for the individual drawbacks and propose a framework solution to the overall system.

Mainstream approaches to 3D-based object detection implement end-to-end architectures. However, there exist two main problems: 1) End-to-end approaches usually require massive amounts of training data and computation resources. 2) Their results are hard to adapt since they are sensitive to training data and cannot be generalized perfectly to different scenarios. To overcome these problems, we propose an integrated system that effectively combines 3D-based detection, tracking and localization in a complementary manner. The system, as shown in Fig. 3.1, begins with an easy-to-train RCNN-based Localization Network (LOCNet), which is only trained with limited amounts of training data to provide reasonable initialization of an object's 3D orientation and distance; Further incorporated with a follow-up single frame optimization method based on the fitness evaluation score (FES) on the 2D raw images, we are able to further improve its 3D localization accuracy in various unreliable detection and localization scenarios.

Frame-by-frame detections are never perfect. Temporal information derived from videos can be employed to associate detections across frames and recover missing or unreliable detections. Traditional tracking methods are usually performed in image coordinates or camera coordinates, which may become problematic for autonomous driving scenarios where the camera encounters translational and rotational movements. To solve this, we take advantage of camera ego-motion to perform tracking in 3D world coordinates. The proposed 3D TrackletNet Tracker (3D TNT)

utilizes accurate spatial object information along with discriminative appearance features to achieve better tracking performance. In addition, we exploit the temporal consistency and use a multi-frame optimization technique based on reliable associations from tracking to obtain the best localization performance.

The main contributions are summarized as follows:

- An RCNN-based LOCNet is proposed to simultaneously regress both the 3D orientation and distance of vehicles, which can serve as a good initialization for follow-up optimizations.

- A single-frame optimization technique based on the fitness evaluation score (FES) is applied to ensure the object spatial robustness in the 3D localization.

- A 3D TrackletNet Tracker, which takes into account both discriminative CNN appearance features and accurate 3D spatial object information from each frame, is introduced to associate detections across frames.

- A multi-frame optimization technique is incorporated to reduce the impact from unreliable or missing detections and generate more accurate 3D object localization by taking into account temporal consistency.

1.2 JMV3D: Joint Monocular Vehicle 3D Localization, Tracking and Segmentation

Monocular 3D localization, tracking, and segmentation are inherently ill-posed. 3D detection method is challenging by itself in the absence of depth measurements or strong priors given a single image, which often requires a large amount of training data and is hard to adapt since they are sensitive to training data. To overcome these problems, our proposed JMV3D framework begins with an easy-to-train RCNN-based Localization for Tracking Network (Loc4Trk-Net), which is only trained with limited amounts of training data, not only to generate a 2D bounding box and an instance mask, but also to provide reasonable initialization of an object's 3D orientation and distance; Further incorporated with a follow-up single frame optimization method based on

the fitness evaluation score (FES) on the 2D raw images, we are able to further improve its 3D localization accuracy in various unreliable detection and localization scenarios.

Frame-by-frame detections are never perfect. Given a strong localization basis, short-term 3D tracking tends to be more robust, and long-term 3D tracking becomes possible. At the same time, 3D tracking information across multiple frames can further assist 3D localization as well by recovering missing/unreliable detections. In addition, self-supervised spatial attention is also applied to our model to learn an instance-aware embedding for each object, which is an instance descriptor represented as a vector in a latent space via deep contrastive learning. Robust tracking results are obtained by associating the detections with the learned features and their historical trajectories using an online 3D Kalman filter and Hungarian matching algorithm.

The main contributions are summarized as follows:

- An RCNN-based Loc4Trk-Net is proposed to not only generate 2D bounding box and instance masks but also simultaneously regress both the 3D orientation and distance of vehicles, which can serve as a good initialization for follow-up FES optimizations.

- Instance-specific features, which are learned jointly with the detection task, utilize the instance masks as spatial attention to emphasize the target of interest explicitly.

- 3D object tracking uses the jointly learned instance-aware feature via pairwise contrastive learning. A straightforward combination of a 3D Kalman filter and the Hungarian algorithm is used for online state estimation and robust data association.

- The proposed JMV3D achieves 1^{st} ranking on the KITTI-MOTS leaderboard. Apart from that, we applied our model to experiment on large-scale urban driving nuScenes dataset and achieve state-of-the-art performance on both 3D detection and tracking benchmark.

1.3 Mono3DCFT: Depth-guided Monocular 3D Object Detection via Coarse-to-Fine Training

Autonomous driving vehicles and robots will transform the modern world just as cars did a century back. It is vital for today's autonomous perception systems to perceive the world the same way

as people do. 3D object detection enables us to capture an object's relative size, pose, and depth information. Among them, depth information can be accessed with the help of LiDAR-scanned point clouds or object-centric stereo matching. However, it usually comes with costly expenses by adding new sensors [229, 324, 325] or higher computational costs [318, 375]. On the other hand, extrapolating depth information from monocular images can be proved to be a viable cost-effective alternative for large-scale deployment with sufficient advancement from its present-day performance.

Conventional monocular 3D object detection involves 2D localization [255, 177, 300, 401] followed by generating 3D object centers from the predicted heatmaps. The model learns the relative size, depth, and pose information with the help of local visual features around the projected 3D object center. This lack of scene-level attention to different objects and contextual cues causes the predictions not to account for inter-object depth relations, which ultimately leads to inadequate performance. Other approaches involve the Pseudo-LiDAR mechanism [207, 318, 335], which convert the estimated dense depth maps to 3D point clouds and run LiDAR-based object detectors on top of them. However, these methods, though better localize the objects with the help of estimated depth, may suffer from the risk of predicting 3D detection on inaccurate depth maps. Additionally, the additional depth estimator incurs a large overhead in inference.

To tackle these issues, we propose a transformer-based framework, the **MONO**cular **3D** object detection via **C**oarse-to-**F**ine **T**raining **(Mono3DCFT)**. It presents a novel depth-guided feature aggregation framework to adaptively estimate each object's 3D attributes based on global context via a two-stage training scheme. The Mono3DCFT mainly consists of a fusion-in-the-backbone encoder and a depth-guided transformer decoder. The fusion-in-the-backbone encoder is modified from Swin-Transformer [187] by adding a multi-scale depth encoder with spatial and cross attention to extract both appearance and depth information. A gating cross-attention fusion block is proposed to learn better coupling features that fuses the geometric and appearance information of the input image. During the 1*st* stage coarse-grained training, the encoder is trained on the whole dense depth map to better capture the depth cues from the high-level semantic information of the image. Then, the depth-guided decoder can be directly concatenated to the encoder in the 2*nd* stage fine-grained training

on the object-wise foreground depth map. This enables few changes to the encoder architecture, which significantly reduces the training cost and avoids obtaining inaccurate depth priors from the pre-trained depth estimator. Furthermore, we introduce the depth positional encoding, and the depth deNoising queries to involve depth-aware hints to the transformer, achieving better performance on monocular 3D object detection. The contributions of this paper can be summarized as follows:

- We propose a novel framework, Mono3DCFT, which consists of a fusion-in-the-backbone encoder and a depth-guided decoder to enable object queries, can adaptively collect rich geometric and appearance information of the scene, resulting in better scene depth estimation to assist object 3D attribute prediction.

- The proposed two-stage coarse-to-fine training on whole sense depth map data, followed by object-wise depth labels, can better capture both scene-level depth cues and region-level appearance cues.

- Experimental results on the KITTI dataset show that our proposed Mono3DCFT achieves near SOTA performance among monocular-based methods with significant gains without extra out-of-domain depth data.

Chapter 2

RELATED WORK

2.1 Monocular 3D Object Detection via Image

Monocular 3D object detection is actually an ill-posed problem. The monocular image lacks depth information because of the principle of perspective transformation. In order to achieve monocular 3D detection well, many algorithms have been developed in recent years.

Some algorithms detect specific kinds of objects, which use some prior hypotheses and template matching. And some algorithms use deep learning to predict the depth map of the image first, which serves as a basis for 3D object detection in the next stage. The PnP based algorithms, which establish the correspondence between 3D key points on the 3D model and the 2D key points on the monocular image, can achieve good detection results. The recent algorithm is to convert the image data format into point clouds data format, and then use the deep learning networks for processing point clouds to predict the 3D information of the objects. In this section, we review some algorithms of monocular 3D object detection.

2.1.1 Single-stage 2D detection based Algorithm

Mousavian [220] proposes a 3D object detection algorithm of Deep3DBox. This algorithm extends the existing 2D detection network, and uses the regression algorithm to directly return the object's spatial size and its yaw angle. A major contribution of Deep3DBox is to propose the MultiBin skill, which calculates the yaw angle of object.

The previous algorithm mainly uses the L2 loss function to directly return to the yaw angle, while MultiBin first discrete the yaw angle into multiple overlapping 3D bins, and then using a convolutional neural network to predict the confidence of each bin and the offset from the rotation residual of the base bin. In the estimation of the object space size, the L2 loss function is directly used

to calculate the offset of the space size. Shift-RCNN [235] further extends idea from Deep3DBox and obtain 3D localization of objects using the geometric constraints between 3D points and 2D box edges. However, by considering geometric projection as the post-processing step, the error from 2D box detection, 3D object orientation and dimension regression can be aggregated in the subsequent distance estimation module.

2.1.2 PnP based Algorithm

The algorithm of using key points is not to directly obtain the pose of the object from the monocular image, but use a two-stage algorithm. The network first predicts the 2D key points of the object, and then calculates the pose of the object by 2D-3D correspondence with the PnP algorithm [5].

2D keypoint detection is relatively easier than 3D localization and rotation estimation, but requires a model of a known 3D object and some predefined keypoints. For objects with rich textures, traditional algorithms can detect local key points more robustly, even in cluttered scenes and severe occlusions. [28, 5] consider the problem as a purely geometric problem, known as the bundle adjustment problem (BA), where closed-form or iterative solutions can be applied by assuming a robust correspondence between 2D semantic keypoints and a 3D model of the object. However, these 2D keypoints largely depend on the training data and can be easily affected by partial occlusions or truncation. Furthermore, such BA iterations can usually be very time-consuming due to random initialization.

2.1.3 Pseudo point cloud based Algorithm

Another popular algorithm is to convert the image information into point cloud information, and then use the point cloud-related network for processing. This algorithms propose that the point cloud data format is more suitable for 3D object detection than image, so it can achieve satisfactory detection results using only the camera.

With the recent success of LiDAR-based 3D object detection, the PointNet network [239] can be used for point cloud classification and semantic segmentation. The way of extracting features in

PointNet is global, which is different from the way of the convolutional neural networks to extract local features layer by layer. Based on this idea, Charles [173] proposes PointRCNN, which can extract feature layers at different scales in local features. PointNet and PointRCNN are mainly used for the classification and detection of point clouds.

Based on these two networks, Pseudo-LiDAR [319] proposes that the accuracy limit of monocular image detection of 3D objects is not because the accuracy of monocular depth estimation is not sufficient, but because the data representation of point clouds is more suitable for 3D object detection than images. Therefore, Pseudo-LiDAR uses the DORN network [78] to estimate the depth and uses the corresponding mathematical relationship to convert the image information into pseudo-point cloud information. And then, it uses two more advanced point cloud processing networks to process the data of the pseudo-point cloud.

2.1.4 Transformer based Algorithm

Transformer [306] was firstly introduced in sequential modeling and has considerable improvement in natural language processing (NLP) tasks. The self-attention mechanism is the core component in the transformer with its capability of capturing the long-range dependencies. Recently, transformer architecture has been successfully leveraged in the computer vision field, such as image classification [64] and human-object interaction [143]. In addition, DETR [24] proposes developing object detection with the transformer without relying on many hand-designed components used in traditional pipelines.

Though the transformer can perform well in most visual tasks, its usage in monocular 3D object detection has not been explored. In the image-based 3D detection task, the object size at far and near distance in the image varies significantly due to the perspective projection [60, 311], which makes it challenging to utilize the learned object query mentioned in DETR[24] to fully represent the object property. MonoDTR [122] is the first transformer-based fusion architecture to integrate image and depth information globally. It uses a Depth-aware Feature Enhancement module that implicitly learns depth-aware features with the help of auxiliary supervision, a Depth-aware Transformer module that performs global integration of context and depth features, and a Depth positional

encoding (DPE) module to inject depth positional hints to the transformers.

2.2 Monocular 3D Object Detection via Depth

2.2.1 Depth-assisted Monocular 3D Object Detection.

Many researchers use depth information to obtain more reliable 3D detections in a direct way, which is named 'depth-assisted.' The gap has been greatly reduced by the proposed pseudo-LiDAR framework [207, 318, 335]. Unlike previous image-based 3D object detection methods, pseudo-LiDAR first utilizes an off-the-shelf depth estimator to convert the image pixels to 3D pseudo point clouds, and then run LiDAR-based 3D object detectors. Upon pseudo-LiDAR, Patch-Net[205] further replaces coordinate transformation with the image representation, which can benefit from the powerful CNNs networks. However, most depth-assisted methods use pre-trained depth estimators and suffer from inaccurate depth, as well as introduce additional computational burden.

2.2.2 Depth-guided Monocular 3D Object Detection

Different from previous depth-assisted methods, depth-guided methods introduce context- and depth-aware features for better 3D reasoning. DD3D [231] considers using a large-scale depth dataset to train the backbone to get more representative 3D features in the pre-training stage only. However, their depth and visual features do not interact with each other for monocular 3D object detection tasks during the fine-tuning stage. Besides, they use convolutional networks as the backbone, which fail to capture the global spatial dependencies in depth prediction. MonoDTR [122] is the first transformer-based fusion architecture to integrate image and depth information globally. It has demonstrated its superior performance with non-local encoding inherited from transformer architecture.

2.3 Multi-Object Tracking

2.3.1 2D Multi-Object Tracking

Recent multiple object tracking (MOT) methods have largely employed tracking-by-detection schemes [134], meaning that tracking is done through the association of detected objects across time. With tracking, object locations can be inferred even in the event of occlusion, truncation, or unreliable detection. Two types of representations, appearance features, and spatial information, are widely used to perform the association accurately and consistently.

Some studies tend to focus more on classical appearance features at the image level (HOG, color histogram, and LBP) [259]. However, it is quite ad-hoc to determine the weighting for each human-crafted feature. In [294, 389], deep convolutional neural networks are used to get discriminate embedding features, which can effectively re-identify the same object in a limited time frame and deal with partial and complete occlusions. Some other frameworks emphasize spatial information, such as network flow [379], multi-hypothesis tracking [144] and quadratic pseudo boolean optimization [69].

2.3.2 3D Multi-Object Tracking

The previously discussed methods in 2D for object tracking in the image domain usually only take visual features and 2D motion into consideration. However, lack of depth information in 2D tracking causes failure in tracking objects long-term due to disappearances and occlusions. Therefore, various research works have proposed to further leverage 3D information to narrow down the search space and stabilize the trajectory of target objects. One approach by Luiten et al. [200] proposes to use dynamic 3D reconstruction for tracklet association in 3D space to improve long-term tracking. Osep et al. [308] study the extension of this paradigm to 3D bounding box tracking using 3D information obtained from stereo cameras.

Further, Given the LiDAR point cloud, recent work by Weng et al. [334] uses standard 3D Kalman filters and Hungarian algorithms to associate detections from LiDAR, which causes fewer ID switches and can perform long-term tracking. Yet LiDAR has its own drawbacks such as

high cost and sensitivity to adverse weather conditions. These limitations suggest that employing LiDAR-based object tracking system is unrealistic in practical, day-to-day applications.

2.3.3 Multi-Object Tracking and Segmentation

MOTS is proposed as a new task to track multiple objects with instance segmentation. Voigtlaende et al. [308] propose a baseline approach Track R-CNN, which can jointly address detection, tracking, and segmentation via a single convolutional network. While the aforementioned method is able to produce tracking outputs with segmentation masks, the network is trained under multiple task, resulting in increasing the tracking performance while degrading the detection and segmentation performance.

2.3.4 Related Autonomous Driving Datasets

Driving datasets have comprised some of the most popular benchmarks for computer vision algorithms in the last decade. Benchmarks like KITTI [90], UA-DETRAC [203], Cityscapes [50] and Oxford RobotCar [209] provide well annotated ground truth for visual odometry, stereo reconstruction, optical flow, scene flow, object detection and tracking as well as semantic segmentation. However, due to the high effort that the annotation of these datasets requires, these benchmarks have been limited in scale. In recent years, the topic of autonomous driving catched on more and more in the industry, providing the resources for new, large-scale driving benchmarks for computer vision, providing annotations for 3D computer vision tasks like 3D object detection and tracking at an unprecedented scale. Therefore, benchmarks like Apollo3D [282], BDD100K [366], NuScenes [20], Argoverse [29] and Waymo Open [288] have attracted a lot of attention by the research community. Still, accurate 3D annotations are challenging to obtain and expensive to measure with 3D sensors like LiDAR.

Chapter 3

MONOCULAR 3D LOCALIZATION OF VEHICLES IN ROAD SCENES

Mainstream approaches to 3D-based object detection implement end-to-end architectures. However, there exists two main problems: 1) End-to-end approaches usually require massive amounts of training data and computation resources. 2) Their results are hard to adapt since they are sensitive to training data and cannot be generalized perfectly to different scenarios. To overcome these problems, we propose an integrated system that effectively combines 3D-based detection, tracking and localization in a complementary manner. The system, as shown in Fig. 3.1, begins with an easy-to-train RCNN-based Localization Network (LOCNet), which is only trained with limited amounts of training data, to provide reasonable initialization of an object's 3D orientation and distance; Further incorporated with a follow-up single frame optimization method based on the fitness evaluation score (FES) on the 2D raw images, we are able to further improve its 3D localization accuracy in various unreliable detection and localization scenarios.

Frame-by-frame detections are never perfect. Temporal information derived from videos can be employed to associate detections across frames and recover missing or unreliable detections. Traditional tracking methods are usually performed in image coordinates or camera coordinates, which may become problematic for autonomous driving scenarios where the camera encounters translational and rotational movements. To solve this, we take advantage of camera ego-motion to perform tracking in 3D world coordinates. The proposed 3D TrackletNet Tracker (3D TNT) utilizes accurate spatial object information along with discriminative appearance features to achieve better tracking performance. In addition, we exploit the temporal consistency and use a multi-frame optimization technique based on the reliable associations from tracking to obtain the best localization performance.

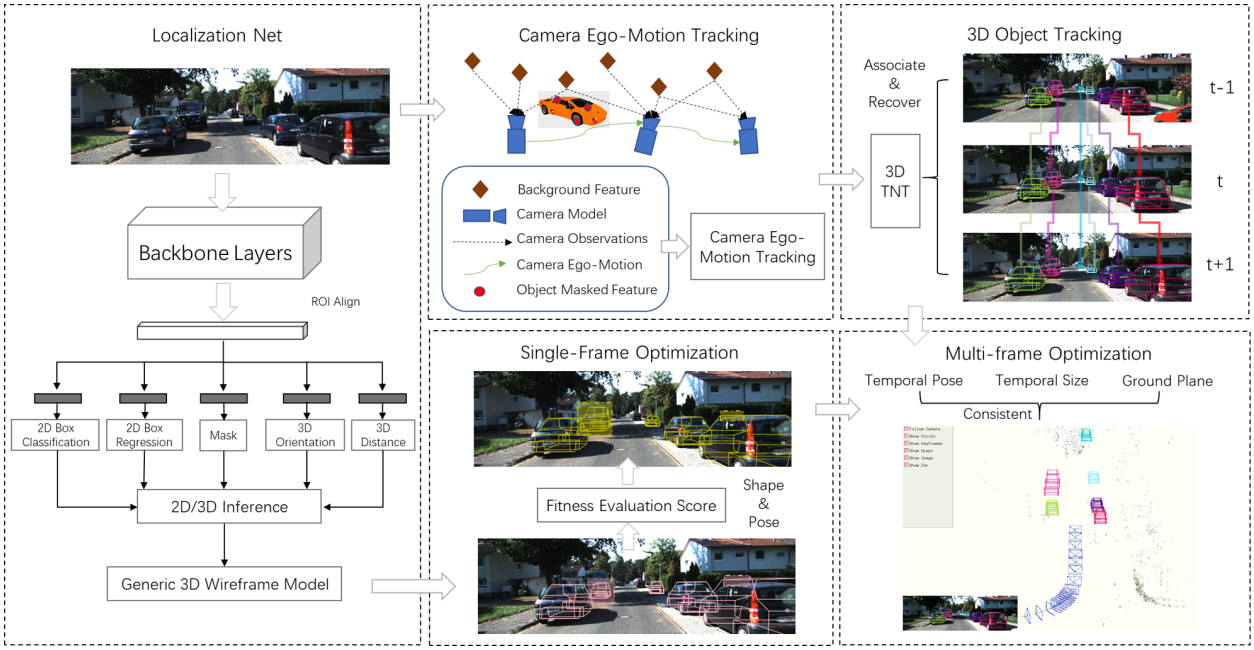

Figure 3.1: System Overview. The system integrates 3D object detection, single-frame optimization, 3D object tracking and multi-frame optimization to achieve the best localization performance.

3.1 LOCNet: 3D Localization Network

3.1.1 Inverse Geometry Interpretation

Deep3Dbox [28] exploits the constraints from projective geometry to estimate full 3D pose and object dimensions from a 2D box. Further estimating the real 3D translation vector is needed in order to accurately reconstruct the 3D bounding box. in 3D camera coordinates. We formulate this estimation in a closed-form, as a least squares solution given by fitting the *geometric constraints* imposed by the camera projection matrix K. Thus, an object described by its 2D bounding box b_{2D} and local orientation angle θ, will have a depth-constrained translation $t = [t_x, t_y, t_z]$ in camera coordinates.

To enforce the 3D bounding box projection to fit tightly into the predicted 2D bounding box, we constrain 2 of the 4 vertical 3D edges to lay on a 2D vertical side and the upper and lower 3D corners to lay on a horizontal 2D side. Assuming that objects lay on the ground plane and that

by fixing one vertical 3D edge the second vertical one must be diagonally opposed, we have 64 configurations - from which we choose the best fit. We reconstruct the 3D bounding box x_{3D_0} at the camera center, then obtain the final 3D box, in camera coordinates, by applying the rotation about the y-axis $R_y(\theta)$ and translation t:

$$x_{3D} = R_y(\theta)x_{3D_0} + t. \tag{3.1}$$

The relation between a 3D point in the world x_{3D} and its 2D projection in the image x_{2D}, using the projection matrix K, is given in homogeneous coordinates:

$$\lambda \cdot \begin{bmatrix} x_{2D} \\ 1 \end{bmatrix} = K \times \begin{bmatrix} x_{2D_{side}} \\ y_{2D_{side}} \end{bmatrix}. \tag{3.2}$$

From Eq. 3.1 and 3.2, we obtain the following system where translation t is the unknown vector:

$$K \times \begin{bmatrix} I & R_y(\theta)x_{3D_0} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} t \\ 1 \end{bmatrix} = \lambda \cdot \begin{bmatrix} x_{2D_{side}} \\ y_{2D_{side}} \\ 1 \end{bmatrix}. \tag{3.3}$$

By substituting each element of b_{2D}, corresponding to a 2D side, and also λ in Eq. 3.3, we propose a least squares solution for the translation. Here $n_i^T = [m_{i1}, m_{i2}, m_{i3}]$ and m_{ij} is the (i, j) element of matrix M.

$$\begin{bmatrix} n_1^T - n_3^T x_{\min} \\ n_2^T - n_3^T y_{\min} \\ n_1^T - n_3^T x_{\max} \\ n_2^T - n_3^T y_{\max} \end{bmatrix} t = \begin{bmatrix} m_{34}x_{\min} - m_{14} \\ m_{34}y_{\min} - m_{24} \\ m_{34}x_{\max} - m_{14} \\ m_{34}y_{\max} - m_{24} \end{bmatrix}. \tag{3.4}$$

The over-constrained Eq. 3.4 can be rewritten as $At = b, b \neq 0$, with a general closed-form solution for the 3D object translation $t = (A^T A)^{-1} A^T b$.

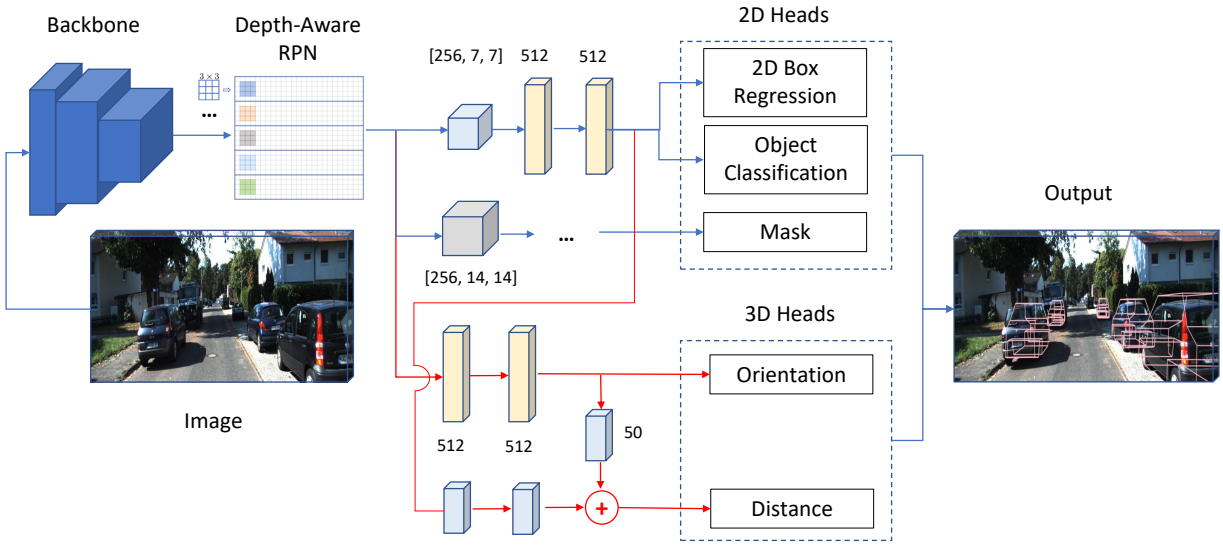

Figure 3.2: Localization Network (LOCNet). The upper part (in blue) is the typical Mask-RCNN detection framework. The bottom part is the added 3D orientation and distance heads (in red).

3.1.2 Network Architecture

The proposed Localization Network (LOCNet) is built upon a popular two-stage object detection network, the Mask-RCNN [108]. LOCNet augments the Mask R-CNN model with a unique depth-aware region proposal network (RPN) [256] and additional learning objectives. In the first stage, we extract and score region proposals by means of anchors based on depth-aware RPN, then ROIAlign for feature cropping is deployed. Based on the top scoring proposals, we use a convolutional encoder to refine the cropped features, then split them up into 5 separate heads. The second stage of the network consists of both classical and customized heads. For the 2D part we use 3 heads for standard multi-class classification, 2D box refinement and (instance segmentation) mask generation respectively. The additional 2 heads are introduced to handle object 3D orientation and distance. The architecture is shown in Fig. 3.2.

Depth-Aware RPN. ResNet-50 is adopted as a convolution body with a feature pyramid network (FPN) as our detection backbone, which takes a single 2D RGB image to extract feature maps as inputs to a 3D-tailored depth-aware RPN. It has been proven [15] that high-level features related to 3D scene understanding are dependent on depth when a fixed camera is assumed. In this case, we separate the feature map into different row bins and apply individual 2D convolutions for each of them. We believe these depth-aware kernels enable the network to develop location specific features and biases for each bin region. We append a proposal feature extraction layer using depth-aware convolutions to generate features for further processing.

Orientation Head. The orientation head takes the same depth-aware ROI-Aligned feature maps (256×14×14) as input to generate the 3D orientation output. Due to the periodic nature of orientation, it is harder to regress angles explicitly. Although Euler angles, yaw, $pitch$, $roll$, are easily understandable and interpretable for 3D orientation, they are sensitive to non-injectivity and gimbal lock [103]. Thus, we instead regress the quarternions [370] since they are continuous, which can be easily enforced through back-propagation. For the orientation head, given the ground truth quaternion $q \in R^4$ and the predicted quaternion $\hat{q}$, the orientation loss is defined as:

$$L_{ori}(q, \hat{q}) = \left\| q - \frac{\hat{q}}{\|\hat{q}\|_2} \right\|_2. \tag{3.5}$$

Distance Head. The distance head takes a concatenated input, from both depth-aware ROIAligned feature maps (256×14×14) and convolved 512-dim features for bounding-box classification/regression, to form more informative input features for 3D distance. The concatenated features are assumed to implicitly encode the 3D orientation information and pre-defined object size information via the incorporation of the convolved 512-dim features. To generate the ground truth for this distance head, we need to transform the 2D detected objects' box center, height and width (u_p, v_p, h_p, w_p) in 2D image coordinates to their corresponding (u_c, v_c, h_c and w_c) in 3D camera coordinates so that the ground truth 3D distances can be determined.

$$\begin{aligned} u_c &= \frac{(u_p - c_x)z_s}{f_x}, h_c = \frac{h_p}{f_x}, \\ v_c &= \frac{(v_p - c_x)z_s}{f_y}, w_c = \frac{w_p}{f_y}, \end{aligned} \tag{3.6}$$

where the parameter vector $[f_x, 0, c_x; 0, f_y, c_y; 0, 0, 1]$ stands for the camera intrinsic K, and z_s is the projective distance [340].

Huber loss is adopted to formulate the penalty in distance estimation: given ground truth distance d and the prediction $\hat{d}$, the distance loss is:

$$L_{dis}(d, \hat{d}) = \begin{cases} \frac{1}{2}(d - \hat{d})^2/\delta & if \left|d - \hat{d}\right| < \delta, \\ \left|d - \hat{d}\right| - \frac{1}{2}\delta & \text{otherwise.} \end{cases} \tag{3.7}$$

where the hyper-parameter δ controls the boundary of outliers.

3.1.3 Multi-Task Loss

The following total loss function L_{total} is minimized to train our proposed LOCNet. The first three loss terms are the standard Mask R-CNN multiclass loss L_{cls}, 2D bounding box regression losses L_{box} and mask loss L_{mask}, respectively as defined in [108]. The last two terms are the orientation loss L_{ori} and distance loss L_{dis} respectively, as defined in Eq. (3.5) and Eq. (3.7).

$$\begin{aligned} L_{total} =& w_{cls}L_{cls} + w_{box}L_{box} \\ &+ w_{mask}L_{mask} + w_{ori}L_{ori} + w_{dis}L_{dis}. \end{aligned} \tag{3.8}$$

We show in the later ablation study Sec. 3.5.4 that our novel formulation for distance regression can produce much more accurate 3D localization estimation compared to methods that treat the distance estimation as a post-processing step [235, 220]. This accurate estimation of both orientation and distance is particularly crucial for the autonomous driving applications, where the location of the objects is of primary importance. Furthermore, the predicted orientation and distance of each object from LOCNet also serve as a good initialization for the subsequent 3D localization optimization part.

3.2 Single-frame Optimization

Although the orientation and distance estimation results from LOCNet can deal with partial occlusions and truncation cases in most of the time, they are not accurate enough for 3D localization. As

you may also notice, LOCNet only focuses on the localization on 2D and 3D without considering the object size, which is an important aspect of 3D detection. In this section, we propose a lightweight optimization pipeline for single-view that refines the initial estimates to ensure localization robustness. Meanwhile, the size of the detected object can also be obtained through this refined optimization. A 3D deformable vehicle model containing 36 shape parameters is set up as prior information and will be described in details in Sec. 3.2.1. An effective fitness evaluation score (FES) is then used to evaluate the fitness between the 2D projection of the 3D deformable vehicle model and raw image data. Moreover, the fitness evaluation is combined into an optimization framework to select better individuals from the combined parameter space based on an iterative population selection strategy.

3.2.1 3D Deformable Vehicle Model

Our deformable model [5] of a vehicle is a 3D wireframe model with 36 shape parameters, which is shown in Fig. 3.3. The shape parameters have respective changeable values and are interdependent. The pose P of a vehicle can be determined by its position (X, Y, Z) and its orientation θ about the vertical axis of the camera coordinates. The projection relation between each vertex of the 3D car model $V_m = (X_m, Y_m, Z_m)$ in object coordinates and its corresponding point v_m in image coordinates is shown in Eq. (3.9) and (3.10).

$$v_m = K \cdot P \cdot V_m. \tag{3.9}$$

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} \cos\theta & -\sin\theta & 0 & X \\ \sin\theta & \cos\theta & 0 & Y \\ 0 & 0 & 1 & Z \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \tag{3.10}$$

With the pose parameters initialized by LOCNet, the 3D vehicle model can then be projected onto the image plane to match with raw image data. An accurate and efficient method is required for fitness evaluation between the projection of 3D vehicle model and image data, which will be described in detail in the next subsection.

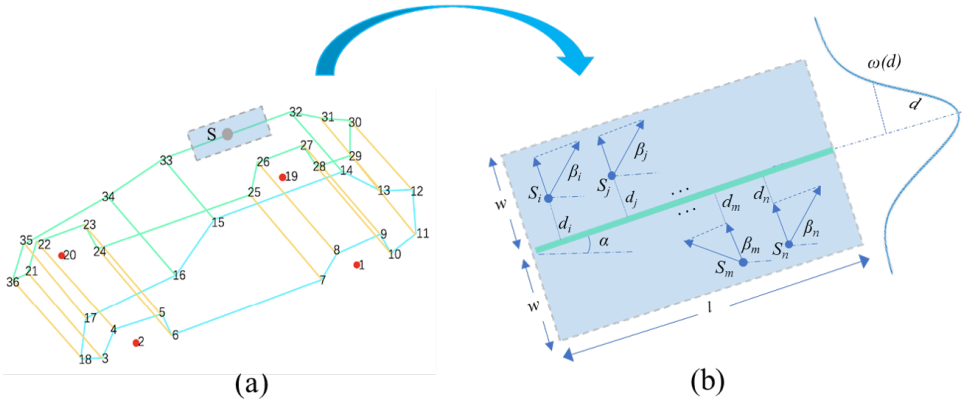

Figure 3.3: (a) A deformable vehicle model with 36 shape parameters. (b) Indication of projected line segments (blue), gradient directions $m(u, v)$ and gradient angle $a(u, v)$.

3.2.2 Fitness Evaluation Score

Fitness evaluation between the projected 3D vehicle model and image data is proposed in [387]. Owing to its effective performance, here we adopt it to our deformable-model-based approaches. Most model-based vehicle localization methods require an initialized pose to project. In this work, the pose initialization is provided by the LOCNet, and the wireframe model can be projected onto 2D image coordinates to form a set of projected line segments. Based on the initial orientation θ, we are able to identify which line segments are visible. For every visible projected line segment, whose direction is denoted as α with length l and width $2w$ in image coordinates, we form a $l \times 2w$ virtual rectangle, as shown in Fig. 3.3. Along the gradient directions of pixels with large gradient magnitude values in the rectangle should coincide with the perpendicular direction of the projected line, if the line fits the image data well. Then, we are able to estimate the fitness score from the gradient information of all pixels within the bounding rectangle. For pixel s_i within the rectangle, we can simply compute its gradient magnitude $m(u, v)$ and gradient angle $a(u, v)$ from pixel differences as follows:

$$\begin{cases} \frac{\partial}{\partial u} I(u,v) = I(u+1,v) - I(u-1,v) \\ \frac{\partial}{\partial v} I(u,v) = I(u,v+1) - I(u,v-1) \\ m(u,v) = \sqrt{\left(\frac{\partial}{\partial u} I(u,v)\right)^2 + \left(\frac{\partial}{\partial v} I(u,v)\right)^2} \\ a(u,v) = \tan^{-1}(\frac{\partial}{\partial v} I(u,v) / \frac{\partial}{\partial u} I(u,v)). \end{cases} \tag{3.11}$$

The fitness error score $E(s_i)$ is calculated by the component of its gradient magnitude perpendicular to the direction in Eq. (3.12). It is also evident that not all pixels in the rectangle have the same weight for fitness evaluation. For those closer to the visible projected line segment, the pixels should contribute more to the FES. In this case, a weight value $\omega(d_i)$ is assigned to every pixel, where d_i is the distance between s_i and projected line segment, and $\omega \sim N(\mu = 0, \sigma = w)$, which is a standard normal distribution. The total FES value, E between the projection of the 3D vehicle model and image data can be obtained from all visible projected line segments, as shown in Eq. (3.13).

$$E(s_i) = \left| m(u,v) \cdot \sin(a(u,v) - \alpha) \right|. \tag{3.12}$$

$$\begin{aligned} E &= \sum_l \log(E_l) \\ &= \sum_l \sum_{s_i} [E(s_i) \cdot \omega(d_i)]. \end{aligned} \tag{3.13}$$

Our approach performs efficiently and accurately for 3D object localization upon a good pose initialization from LOCNet. FES has several advantages comparing with many other existing methods. Compared to [5], whose pose and shape priors are largely dependent on 2D semantic keypoint trained by a neural network. Though they use an iterative re-weighted optimization scheme to tackle erroneously detected keypoints, we outperform them by using stable and invariant edge information in the local region instead of points, and also by avoiding time-consuming keypoint data labeling and network training. Furthermore, we can also easily handle serious occlusion and truncation cases due to good pose initialization.

3.3 Camera Ego-Motion and Object Tracking

Our tracking is performed in the world coordinates. To transform from 3D camera coordinates to 3D world coordinates, the feature-based visual odometry [222] is introduced here to recover the camera pose through ORB features [262] extracted in every frame. Since ORB features must be located on the static background scene, instead of on the highly dynamic objects, we utilize the segmentation masks predicted from LOCNet in Sec. 3.1 to discard those ORB features that are located on the detected objects, keeping those in the static background. We subsequently find correspondence of the background ORB features of the current frame with those of the previous frame. Outliers are further rejected by the RANSAC algorithm [77] as facilitated by the fundamental or homography matrix.

3.3.1 Camera Ego-Motion Estimation

To make it concise for later sections, we define the notations in the following as also shown in Fig. 3.4. ${}^{w}(\cdot)$, ${}^{c}(\cdot)$, and ${}^{i}(\cdot)$ are used to denote the world, camera and image coordinates respectively. For the k^{th} object at time t, we use ${}^{c}O_t^k = \left\{ \begin{array}{cccccccc} {}^{c}X_t^k, & {}^{c}Y_t^k, & {}^{c}Z_t^k, & {}^{c}\theta_t^k, & {}^{c}H_t^k, & {}^{c}W_t^k, & {}^{c}L_t^k \end{array} \right\}$ to describe its distance, orientation and size, which are obtained from Sec. 3.1 and Sec. 3.2. For the camera ego-motion, we use ${}^{w}c_t = \left\{ \begin{array}{cc} {}^{w}T_t, & {}^{w}R_t \end{array} \right\}$ to indicate the camera translation and rotation.

The camera motion is continuously estimated from time 0 to T: ${}^{w}C = \{{}^{w}c_t\}_{t=0:T}$. Given the measurements of the n^{th} sparse ORB features, which are anchored on the background: ${}^{i}p = \{{}^{i}p_t^n\}_{t=0:T}$ and their corresponding 3D positions: ${}^{w}P = \{{}^{w}P_t^n\}_{t=0:T}$. We formulate the camera ego-motion tracking as the following:

$$\begin{aligned} {}^{w}C, {}^{w}P &= \arg\max_{{}^{w}C, {}^{w}P} \prod_{n=0}^{N} \prod_{t=0}^{T} prob({}^{i}p_t^n | {}^{w}c_t, {}^{w}P_t^n, {}^{w}c_0) \\ &= \arg\max_{{}^{w}C, {}^{w}P} \sum_{n=0}^{N} \sum_{t=0}^{T} \log prob({}^{i}p_t^n | {}^{w}c_t, {}^{w}P_t^n, {}^{w}c_0) \\ &= \arg\min_{{}^{w}C, {}^{w}P} \sum_{n=0}^{N} \sum_{t=0}^{T} \left\| r_p({}^{i}p_t^n, {}^{w}c_t, {}^{w}P_t^n) \right\|_{\sum_n^t}^2 \end{aligned} \tag{3.14}$$

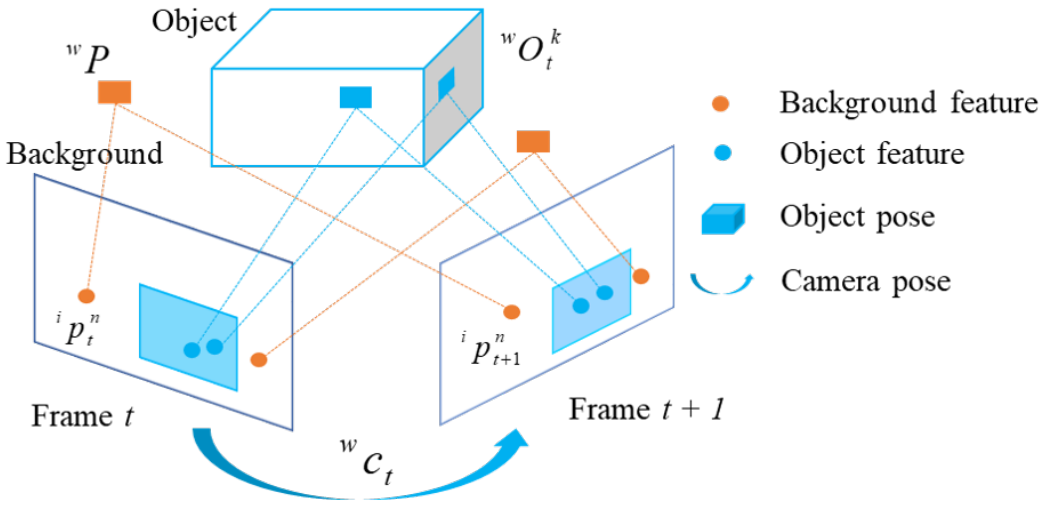

Figure 3.4: Notation visualization.

where $\|r_p()\|^2_\Sigma = {r_p}^T \Sigma^{-1} r_p$, the Mahalanobis norm. This is a common visual odometry formulation, where the camera poses are estimated based on a nonlinear least-squared formulation, also referred to bundle adjustment (BA) [305]. After we solve the camera poses, we can simply convert the object measurements from camera coordinates into world coordinates by using:

$$ {}^{w}O_t^k = {}^{w}C_t^{-1} \cdot {}^{c}O_t^k, \tag{3.15} $$

where the ${}^{w}O_t^k$ stands for object location (distance), orientation and size in world coordinates.

3.3.2 3D TrackletNet Tracker

To take advantage of the temporal consistency for improving the localization performance further, we need tracking to associate corresponding objects along time. The proposed 3D TrackletNet Tracker (3D TNT) takes both discriminative CNN appearance features and accurate object spatial information from each frame to ensure tracking robustness. Inspired by the 2D TNT [310], which builds a graph-based model that takes 2D tracklets as the vertices and use a multi-scale CNN network to measure the connectivity between two tracklets, we further extend the work into 3D tracking scenarios. Our 3D TrackletNet Tracker consists of three key components, as shown in Fig.

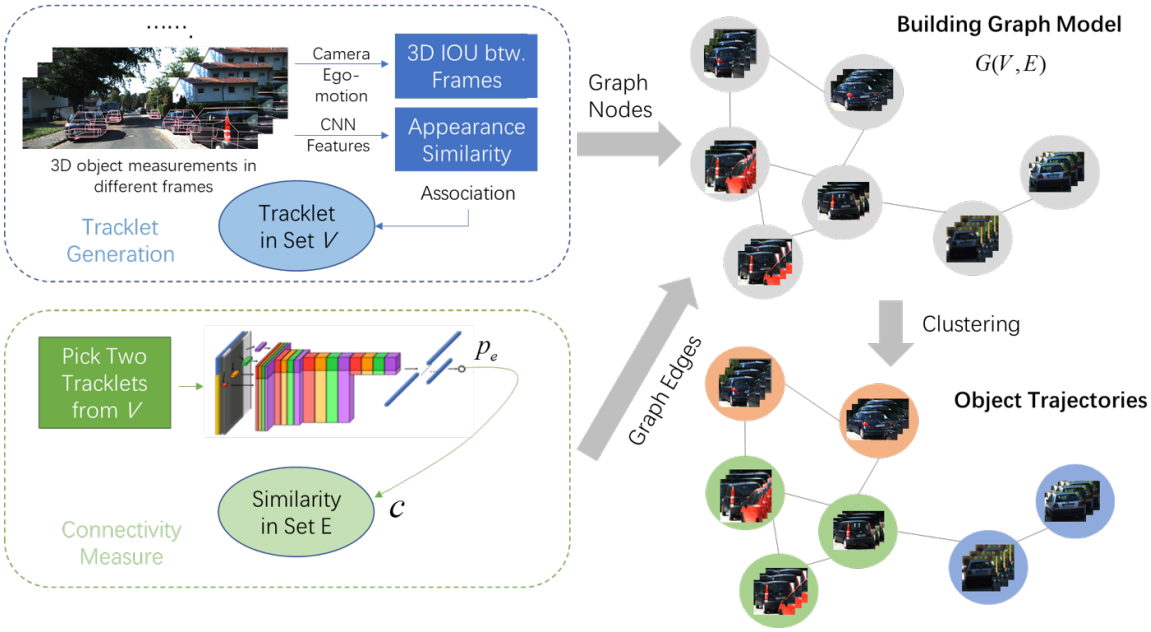

Figure 3.5: 3D TNT framework for object tracking. Given the 3D object measurements in different frames, association is computed to generate tracklets for the Vertex Set V. After that, every two tracklets are put into the TrackletNet to measure the degree of connectivity, which form the similarity on the Edge Set E. A graph model can be derived from V and E. Finally, the tracklets with the same ID are grouped into one cluster using the graph clustering approach..

3.5:

Tracklet Generation. Given the refined vehicle localization of each frame (see Sec. 3.3.1), each tracklet, generated by 2D box appearance similarity based on CNN features derived from FaceNet and 3D intersection-over-union (3D IOU) between adjacent frames, is denoted as a node ($v \in V$) in the graph.

Connectivity Measurement. Between every two tracklets, the connectivity (similarity) $p_e(e \in E)$ is measured and its inverse (dissimilarity) is used as the edge weight in the graph model. To calculate the connectivity, a multi-scale TrackletNet is built as a classifier, which can concatenate both temporal (multi-frame) and appearance features for the likelihood estimation. For each frame t, a vector consisting of the 7-D object measurements ${}^{w}O_t^k$, concatenated by an 512-D embedding appearance feature extracted from the FaceNet, is used to represent an individual feature of the

input frame.

Graph-based Clustering. After the tracklet graph is built, graph partition and clustering techniques, i.e., assign, merge, split, switch, and break operations are iteratively performed to minimize the total cost on the whole graph.

Based on the tracking results from the 3D TNT, we are not only able to associate every object across frames, but also can deal with errors caused by the occlusions and missing detections. This information will be used in the subsequent multi-frame optimization part to further improve the localization performance.

3.4 Multi-frame Optimization

In the context of autonomous driving, the temporal information can be readily exploited to obtain better localization predictions. Based on the 3D object measurements within each frame from Sec. 3.2 and tracking results across frames from Sec. 3.3.1, several temporal consistency constraints can be further imposed to refine the localization results, which are introduced by the following:

Temporal Location and Orientation Consistency. The object location and orientation cannot have a very abrupt change between two adjacent frames, as reflected in the location and orientation consistency regularizer L_P. Here we further denote k^{th} ($k \in K$) object location in frame t as ${}^{w}l_t^k = \left\{ \begin{array}{ccc} {}^{w}X_t^k, & {}^{w}Y_t^k, & {}^{w}Z_t^k \end{array} \right\}$, and object orientation as ${}^{w}\theta_t^k$,

$$\mathcal{L}_P = \sum_{t=0}^{T-1} \sum_{k=1}^{K} \left(\left\| {}^{w}l_{t+1}^k - {}^{w}l_t^k \right\|^2 + \left\| {}^{w}\theta_{t+1}^k - {}^{w}\theta_t^k \right\|^2 \right). \tag{3.16}$$

Temporal Size Consistency. Since the vehicle object of interest is considered as a rigid body, its size (height, width and length) in the 3D world coordinates is supposed to remain the same along time. Here we further denote ${}^{w}s_t^k = \left\{ \begin{array}{ccc} {}^{w}H_t^k, & {}^{w}W_t^k, & {}^{w}L_t^k \end{array} \right\}$ as the object size.

$$\mathcal{L}_S = \sum_{t=0}^{T-1} \sum_{k=1}^{K} (\left\| {}^{w}s_{t+1}^k - {}^{w}s_t^k \right\|^2). \tag{3.17}$$

Ground Plane Consistency. Assume all the observed objects are residing on the same plane, which is usually the case for autonomous driving scenarios. A base plane n_b can be formed by the

roof surface of the 3D car model and its normal vector should have the same direction as the ground plane normal vector n_g computed in [223]. We use the dot product $(\cdot)$ to measure the similarity between two vectors.

$$\mathcal{L}_N = \sum_{t=0}^{T-1} \sum_{k=1}^{K} \left\| (n_g)_t \cdot (n_b)_t^k \right\|. \tag{3.18}$$

Total Optimization Loss. The overall optimization loss $\mathcal{L}_{total}$ consisting all the terms Eq. (3.16), (3.17), (3.18) can be written as

$$\min_{l,\theta} \mathcal{L}_{total} = \omega_P \mathcal{L}_P + \omega_S \mathcal{L}_S + \omega_N \mathcal{L}_N. \tag{3.19}$$

Here $\omega_P, \omega_S, \omega_N$ are the weights to adjust the relative importance for the loss terms. In practice, the loss terms are defined with Huber loss function to avoid the effect of outliers. The above problem can also be minimized using Ceres Solver with a Levenberg-Marquardt optimization method and Iterative Schur as the linear solver. After the multi-frame optimization is performed in world coordinates, we transform the adjusted measurements back to camera coordinates to compare the localization performance.

3.5 Experiments

3.5.1 Dataset

Evaluations are performed on various autonomous driving datasets:

- KITTI [90]: KITTI multi-object tracking dataset contains 20 video sequences for training and 28 sequences for testing. In terms of the data split, we follow [269] and use 1, 6, 8, 10, 12, 13, 14, 15, 16, 18, 19 as the *val* set and other sequences as the *train* set, through our LOCNet training.

- ApolloCar3D [282]: This dataset contains 5,277 driving images with over 60K car instances, aiming at localizing 3D objects in single images.

Figure 3.6: Qualitative examples under diverse scenarios. The top row are the results on the ApolloCar3D instances, and the bottom 2 rows show the results on some image frames of the KITTI tracking dataset. The system is capable of estimating the shape and pose (without scale-factor ambiguity) of objects. The images of the scenes contain the projection of the estimated shapes of cars.

3.5.2 Qualitative Results Under Diverse Scenarios

We demonstrate the system performance on different datasets under various driving scenarios, which include object far distance estimation, occlusion, truncation, and complex road conditions. Some examples of the re-projected images and their corresponding 3D views are shown in Fig. 3.6. We use different colors to represent different vehicles. All the observed cars are visualized in both camera (left side of each column) and world (right side of each column) coordinates for ApolloCar3D and KITTI tracking dataset.

3.5.3 Quantitative Evaluation

For KITTI, we define the true positive of the object 3D localization results if the 3D IOU is greater than 0.5 against the ground truth, as this IoU threshold is widely used and rather strict for image-based methods. For ApolloCar3D, we adopt the official 3D overlap criteria. The quantitative performance are shown in Table 3.1 and 3.2.

KITTI. As the KITTI tracking *test* set ground truth is not released to users, we have to use

Table 3.1: Performance of 3D localization methods using different modality on KITTI *val* set.

Method	Modality	Type	AP_{BEV} (IoU>=0.5)			AP_{3D} (IoU>=0.5)		
			Easy	Mod	Hard	Easy	Mod	Hard
M3D-RPN [15]	Image	Mono	41.53	31.02	26.65	37.41	27.11	23.73
Shift-RCNN [235]	Image	Mono	39.64	30.33	25.90	31.48	24.04	23.60
LOCNet (Ours)	Image	Mono	42.86	30.43	26.35	36.06	25.44	24.19
LOCNet+FES (Ours)	Image	Mono	**50.69**	**36.17**	**31.97**	**48.40**	**38.59**	**32.69**
3DOP [38]	Image	Stereo	54.83	43.36	37.15	53.73	42.27	35.87
Li et al. [167]	Video	Stereo	58.52	46.17	43.97	48.51	37.13	34.54
LOCNet+FES+ 3D TNT+Multi. Opt (Ours)	Video	Mono	**60.37**	**48.49**	**44.36**	**56.54**	**44.23**	**36.91**
Point-RCNN [272]	LiDAR	Pointcloud	66.89	54.91	47.13	62.76	49.13	42.43

the KITTI *val* set for 3D evaluation. Our framework is evaluated on both AP_{BEV} and AP_{3D} metrics and the Car class is split into 3 difficulties: *Easy*, *Moderate* and *Hard*. For 3D localization performance based on single frame images, we compare our LOCNet with/without FES optimization with monocular 3D object detection methods [15, 235]. It can be seen that our method using only LOCNet can achieve 36.06%, 25.44% and 24.19% respectively on AP_{3D}. By adding the FES optimization, we observe significant gains with 48.40% (↑ 12.34%), 38.59% (↑ 13.15%) and 32.69% (↑ 8.5%) on AP_{3D}. Furthermore, by considering the temporal information when dealing with video sequences, we compare our overall system with [38, 167] by adding the proposed 3D TrackletNet and multi-frame optimization methods. We further achieve more gains with 56.54% (↑ 8.14%), 44.23% (↑ 7.1%) and 36.91% (↑ 4.22%) on AP_{3D} and outperform the state-of-the-art image-based methods. Considering the best 3D localization performance, our overall system is even comparable with LiDAR-based methods [272] with reasonable margins ($\sim 4-6\%$).

ApolloCar3D. The 2D evaluation metrics for ApolloCar3D follow similar instance mean AP as the MS-COCO. Instead of using 2D mask IoU to define a true positive, the 3D metric contains the

Table 3.2: Performance of 3D localization methods on ApolloCar3D *val* set.

Method	Modality	2D Evaluation Metrics				3D Evaluation Metrics		
		AP_S	AP_M	AP_L	mAP	shape sim	dist. error (m)	ori. error (°)
LOCNet (Ours)	Image	11.3	12.6	29.7	13.3	0.88	1.13	6.7
LOCNet+FES (Ours)	Image	**11.6**	**13.8**	**33.1**	**14.1**	**0.91**	**1.09**	**6.1**

perspective of shape, 3D distance and orientation. Since there are no available published methods that we can compare with, we only show the performance of baseline and our LOCNet with/without FES optimization. We first provide the 2D evaluation metrics (AP) as shown in Table 3.2. We achieve an mAP of 13.3 by using LOCNet only and we also find that small objects are harder to detect, which commonly indicates the object longitudinal axis distance is far away from the camera. The accurate estimation of large transnational distance value is thus more important. Still, for the 3D evaluation metrics, with the help of FES optimization, the shape similarity, distance and orientation scores are improved by 0.03, 0.04m, 0.6° respectively. Besides, the 2D mAP also increases to 14.1% (↑ 0.8%).

Although we claim that it is not a complete fair comparison between our method and the state-of-the-art image-based 3D object detection methods due to our use of temporal optimization via 3D tracking. However, we stress that our approach only uses a monocular camera and can accurately and efficiently localize the 3D objects with spatial robustness and temporal consistency, which is essential for continuous perception in autonomous driving.

3.5.4 Ablation Study

We perform the ablation study on our LOCNet and the overall system.

Localization Network. To explicitly show the effectiveness of our proposed LOCNet, we perform the ablation study on depth-aware RPN (D-RPN), orientation head (O-H) and distance head (D-H) for both KITTI and ApolloCar3D validation set. O-H+D-H represents we use the features from original RPN in Mask-RCNN to regress the distance and orientation. D-RPN+O-

Table 3.3: Ablation on LOCNet on KITTI and ApolloCar3D *val* set.

Dataset	D-RPN	O-H	D-H	shape sim.	trans dist	rot dist
KITTI		✓	✓	0.93	2.06	6.6
	✓	✓		0.88	5.89	9.2
	✓	✓	✓	**0.94**	**0.98**	**4.3**
ApolloCar3D		✓	✓	0.84	3.67	10.4
	✓	✓		0.78	10.23	12.8
	✓	✓	✓	**0.88**	**1.13**	**6.7**

Table 3.4: Ablation on overall system on KITTI *val* set. (Average precision of bird eye's view and 3D boxes comparison.)

Module	AP_{BEV} (IoU>=0.5)			AP_{3D} (IoU>=0.5)			Time (ms)
	Easy	Mod	Hard	Easy	Mod	Hard	
L	42.86	30.43	26.35	36.06	25.44	24.19	143
L+T	46.89	35.11	28.43	44.22	30.48	27.92	407
L+S	50.69	36.17	31.97	48.40	38.59	32.69	197
L+S+T	57.16	44.72	38.29	54.34	42.88	35.94	457
L+S+T+M	**60.37**	**48.49**	**44.36**	**56.54**	**44.23**	**36.91**	795

H indicates that the network only regresses the orientation, then the distance is obtained by a post-processing stage [220]. D-RPN+O-H+D-H represents both the distance and orientation are regressed simultaneously from the network, where the distance head uses the concatenated features. As seen in Table 3.3, by incorporating both depth-aware RPN and the distance head, the network can achieve the distance and orientation errors within 0.98m and 4.3° for KITTI and 1.13m and 6.7° for ApolloCar3D respectively, which means it is able to exploit the implicit information that is shared between the orientation and distance heads.

Overall System. To see how different modules of our proposed system can contribute to the

localization performance, we further conduct some experiments on KITTI validation set to highlight how they can impact the final results. We use L, S, T, M to represent LOCNet, single frame optimization with FES measure, TrackletNet Tracker, and multi-frame optimization respectively. As shown in Table 3.4, compared to LOCNet-only (L) results, the L+T improves both AP_{BEV} and AP_{3D} by a large margin, which shows that incorporating the temporal information from tracking is helpful to localization accuracy since it can deal with errors caused by occlusions and missing detections. By adding the single-frame FES optimization further brings an improvement of 10.12% and 12.4%, 8.02% respectively. Employing the multi-frame optimization further achieves the best AP_{3D} of 56.54% (↑ 2.2%), 44.23% (↑ 1.35%) and 36.91% (↑ 0.97%). The runtime of the system is also provided based on 8 Core i7-7700k CPUs (S, M) and 2 NVIDIA Titan Xp GPUs (L, T).

3.6 Summary

In this chapter, we propose a monocular vision based autonomous driving framework to perform 3D detection, tracking and localization by effectively integrating all three tasks in a complementary manner. Our LOCNet and FES based single frame optimization provide accurate localization results by utilizing both deep learning approaches and conventional optimization techniques, which are further refined with the help of the 3D TrackletNet Tracker to eventually achieve performance comparable to LiDAR-based localization methods. Quantitative experiments have shown that our system can achieve high accuracy in localization and outperform the state-of-the-art methods. Demonstrations on different datasets also show that our system is robust under different autonomous driving scenarios.

Chapter 4

JOINT MONOCULAR VEHICLE 3D LOCALIZATION, TRACKING AND SEGMENTATION

We phase the 3D MOT/MOTS problems in a supervised manner. Our goal is to jointly infer the 3D localization information from a single monocular video stream and track objects across frames. The 3D localization information includes the distance and orientation of each object instance. During the first stage for Localization for Tracking Network (Loc4Trk-Net), Images are first passed through a backbone network and Region Proposals Network (RPN) to generate 2D object proposals. These 2D proposals are the fed into three multi-head network to infer 3D information and per-instance similarity feature embedding. A lightweight follow-up optimization pipeline for single-view that refines the initial estimates from the network to ensure localization robustness. Meanwhile, the size of the detected object can also be obtained through this refined optimization. Then Hungarian matching is performed between current frame and all tracked tracklets based on feature similarity and 3D intersection-over-union (IoU). Unassigned detected targets are further associated with short-term lost tracklets. Tracklets that miss for longer than threshold are considered as terminated ones to ease computational burden.

Problem Formulation. To make it concise for later sections, we define the notations in the following. ${}^{w}(\cdot)$, ${}^{c}(\cdot)$, and ${}^{i}(\cdot)$ are used to denote the world, camera and image coordinates respectively, e.g., for the k^{th} object at time t in camera coordinates, each detection is represented as a tuple ${}^{c}\mathrm{N}_t^k = [d, \theta, F] \in \mathbb{R}^6$ and we use ${}^{c}S_t^k = [d, \theta, D, \dot{d}, F] \in \mathbb{R}^{10}$ to describe its state, where d defines the 3D localization (X, Y, Z) of the object center, θ for object orientation, D for object size (L, W, H), and $\dot{d}$ for its velocity $(\dot{X}, \dot{Y}, \dot{Z})$. F stands for object-wise appearance feature. For the camera ego-motion on a moving platform, we use ${}^{w}c_t^k = [R|T]$ to indicate the camera translation and rotation, which will be used later in the 3D tracking phase to cancel out the ego-motion of the

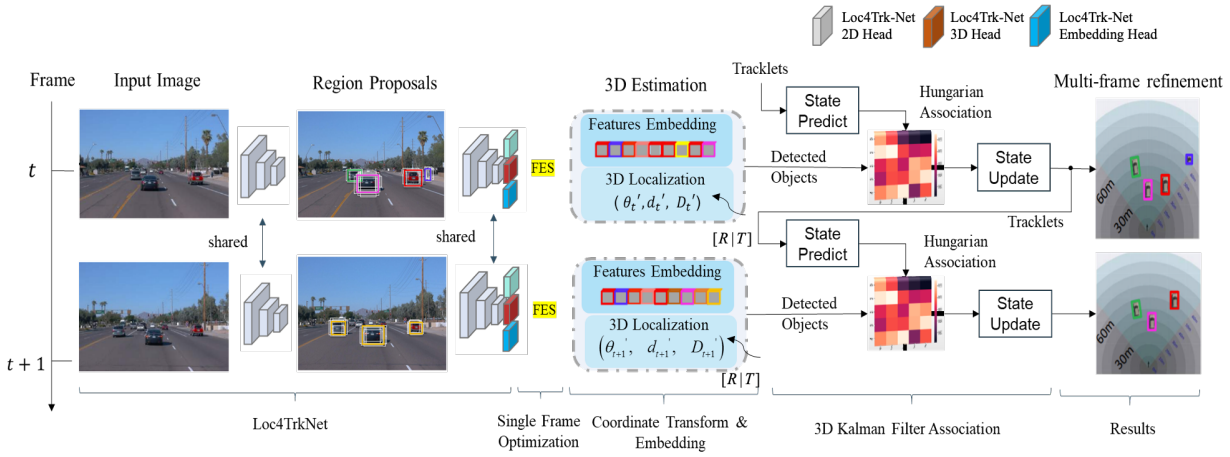

Figure 4.1: **Overview of our pipeline for our JMV3D method.** Our online approach processes monocular frames to estimate and track regions of interest (RoIs) in 3D. Loc4Trk-Net helps to learn the 3D (i.e., orientation, distance) estimation and instance-level feature embedding. Given the initial estimates from the network, FES further ensures the localization accuracy and obtains object size. A 3D Kalman filter produces robust linking across frames leveraging feature similarity and 3D IoU with the help of Hungarian algorithm.

moving platform. The camera intrinsic K can be obtained from camera calibration.

4.1 Loc4Trk-Net: 3D Localization for Tracking Network

Network Architectures. The proposed Localization for Tracking Network (Loc4Trk-Net) is built upon a canonical two-stage object detection network, Mask R-CNN [108]. Loc4Trk-Net augments the Mask R-CNN model with additional learning objectives. The first stage of the network is a multi-stage 2D object detection network, we extract and score region proposals by RPN, then ROIAlign for feature cropping is deployed. Based on the top scoring proposals, we use a convolutional encoder to refine the cropped features, then split them up into five separate heads. The second stage of the network consists of both classical and customized heads. For the 2D part we use three heads for standard multi-class classification, 2D box refinement and mask generation respectively. Two more heads are introduced to handle object 3D orientation and distance. One additional embedding head is introduced to train a discriminative feature embeddings to match detections and tracklets. The

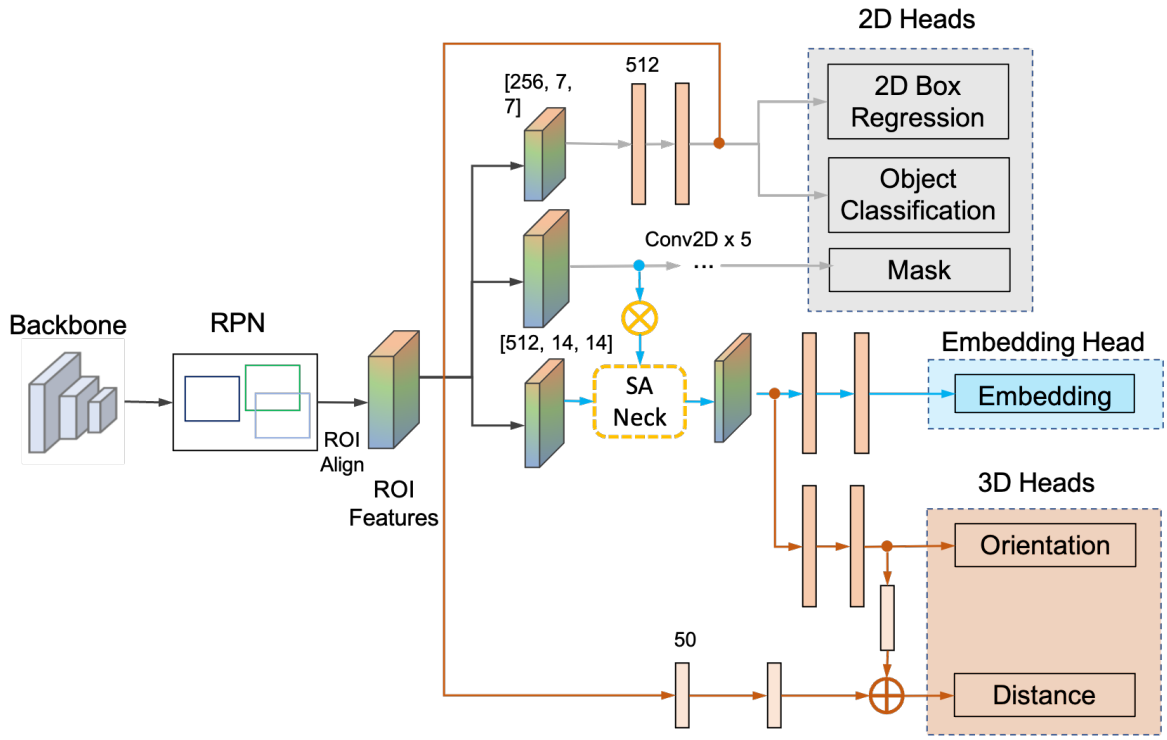

Figure 4.2: **Detailed Architecture for Loc4Trk-Net.** The upper two branches (in gray) are the typical Mask-RCNN detection framework. The middle branch (blue) is the embedding head, with the help of spatial attention (SA) neck (green), will heavily weigh on the foreground object to enhance instance-specific appearance features and suppress the noise in the background. The bottom two branches are the 3D orientation and distance heads (brown).

detailed architecture is shown in Fig. 4.2.

Spatial Attention (SA) Neck. Due to in-plane rotation is unique for a given vehicle class, all vehicles share similar rotational features for the same yaw, pitch, and roll angles. The fixed-size ROIAligned visual cue is not robust for estimating the candidate rotation. In the object detection scenarios, severe occlusions and truncations are usually challenging cases. Therefore, it is likely to include multiple objects within the same 2D bounding box, resulting in erroneous features for orientation regression. The intuition of the spatial attention (SA) neck is to highlight the foreground (target of interest) and suppress the background, so that more concentrated appearance features can be obtained. Details of the SA neck is shown in Fig. 4.2 (green), where ROI features are pooled and

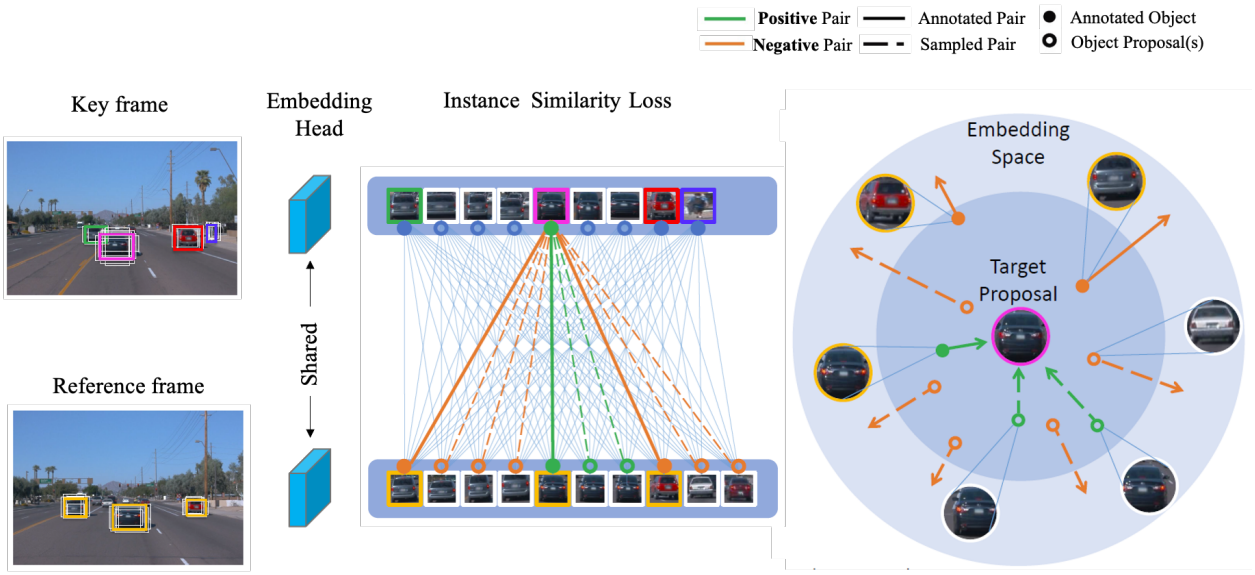

Figure 4.3: **Overview of our training pipeline of our Loc4Trk-Net embedding head**. We leverage all object proposals instead of traditional sparse ground truth (solid circles), to train a discriminative feature space by comparing the region proposal pairs between the key frame and the reference frame. The pairwise contrastive loss pulls the embedding of different identity away from its paired target proposal and draws the embedding of same identity pairs together in a high dimensional space.

flattened for classification and bounding box regression. Simultaneously, they are passed through four 2D convolutional layers and a pixel-wise Sigmoid operation, which is the SA operation, to generate a SA map, indicating the probability of objectness. With the intermediate output of the SA operation, several 2D convolution layers with kernel size 1 are applied to produce the object mask. Meanwhile, the ROI features are multiplied by the SA map to purify pixels which belong to the target and single-dimensional feature is further extracted by fully connected layers.

Embedding Head. Multi-object tracking problem requires distinguishable feature embeddings to match detections and tracklets. Unlike *sparse* metric learning approaches [270, 310], which is widely used in multiple object tracking that learns only from ground-truth bounding boxes, our approach is to utilize all the region proposals generated by RPN to learn the instance similarity by discriminating positive proposals from negative ones with the help of pairwise contrastive loss.

We use RPN to generate RoIs from the two images and RoI Align to obtain their feature maps from different levels in FPN according to their scales. An extra lightweight embedding head is added to extract features for each RoI, shown in Fig. 4.2 (blue). An RoI is defined as positive to an object if they have an IoU higher than 0.7, or negative if they have an IoU lower than 0.3 in our settings. The matching of RoIs on two frames is positive if the two regions are associated with the same object and negative otherwise. Given a key frame at time t, we sample a reference frame within a temporal interval n, where $n \in [-3, 3]$ throughout all the experiments. For each target proposal s_t, we balance the number of positive and negative examples by comparing the target proposal to all positive proposals,

$$L_{emb} = \log[1 + \sum_{F^+_{s_{t+n}}} \sum_{F^-_{s_{t+n}}} \exp(F_{s_t} \cdot F^+_{s_{t+n}} - F_{s_t} \cdot F^-_{s_{t+n}})]. \tag{4.1}$$

The loss term L_{emb} of the above equation minimizes the cosine distance of the target proposal to all positive referenced examples while maximizing the cosine distance to all negative samples. By balancing positive and negative samples, we encourage the network to learn an embedding space that can effectively discriminate between instances, while being invariant to perturbations like changes in viewpoint or lightning.

3D Orientation Head. The orientation head takes the features from SA neck as input to generate the 3D orientation output. Due to the periodic nature of orientation, it is harder to regress angles explicitly. Although Euler angle for yaw θ is easily understandable and interpretable for 3D orientation, they are sensitive to non-injectivity and gimbal lock [103]. Thus, we instead regress the quarternions [370] since they are continuous, which can be easily enforced through back-propagation. For the orientation head, given the ground truth quaternion $q \in R^4$ and the predicted quaternion $\hat{q}$, the orientation loss is defined as:

$$L_{ori}(q, \hat{q}) = \left\| q - \frac{\hat{q}}{\|\hat{q}\|_2} \right\|_2. \tag{4.2}$$

We show in the later experiment results that the our method seems to generate rough masks with different weights on specific vehicle body parts through implicit learning. Fig. 4.6 shows

visualizations of SA maps for orientation estimation. It appears that the network attends to distinct object parts such as tires, lights and side mirror for cars.

3D Distance Head. The distance head takes a concatenated input, from both depth-aware ROIAligned feature maps (256×14×14) and convolved 512-dim features for bounding-box classification/regression, to form more informative input features for 3D distance, which is demonstrated in Fig. 4.2 (brown). The concatenated features are assumed to implicitly encode the 3D orientation information and pre-defined object size information via the incorporation of the convolved 512-dim features. To generate the ground truth for this distance head, we need to transform the 2D detected objects' box center, height and width (u_p, v_p, h_p, w_p) in 2D image coordinates to their corresponding (u_c, v_c, h_c and w_c) in 3D camera coordinates so that the ground truth 3D distances can be determined.

$$\begin{aligned} u_c &= \frac{(u_p - c_x)z_s}{f_x}, h_c = \frac{h_p}{f_x}, \\ v_c &= \frac{(v_p - c_x)z_s}{f_y}, w_c = \frac{w_p}{f_y}, \end{aligned} \tag{4.3}$$

where the parameter vector $[f_x, 0, c_x; 0, f_y, c_y; 0, 0, 1]$ stands for the camera intrinsic K, and z_s is the projective distance [340].

Huber loss is adopted to formulate the penalty in distance estimation: given ground truth distance d and the prediction $\hat{d}$, the distance loss is:

$$L_{dis}(d, \hat{d}) = \begin{cases} \frac{1}{2}(d - \hat{d})^2/\delta & if \left|d - \hat{d}\right| < \delta, \\ \left|d - \hat{d}\right| - \frac{1}{2}\delta & \text{otherwise.} \end{cases} \tag{4.4}$$

where the hyper-parameter δ controls the boundary of outliers.

Multi-Task Learning. The following total loss function L_{total} is minimized to train our proposed Loc4Trk-Net. The first three loss terms are the standard Mask R-CNN multiclass loss L_{cls}, 2D bounding box regression losses L_{box} and mask loss L_{mask}, respectively as defined in [108]. The last two terms are the orientation loss L_{ori}, embedding loss L_{emb} and distance loss L_{dis} respectively, as

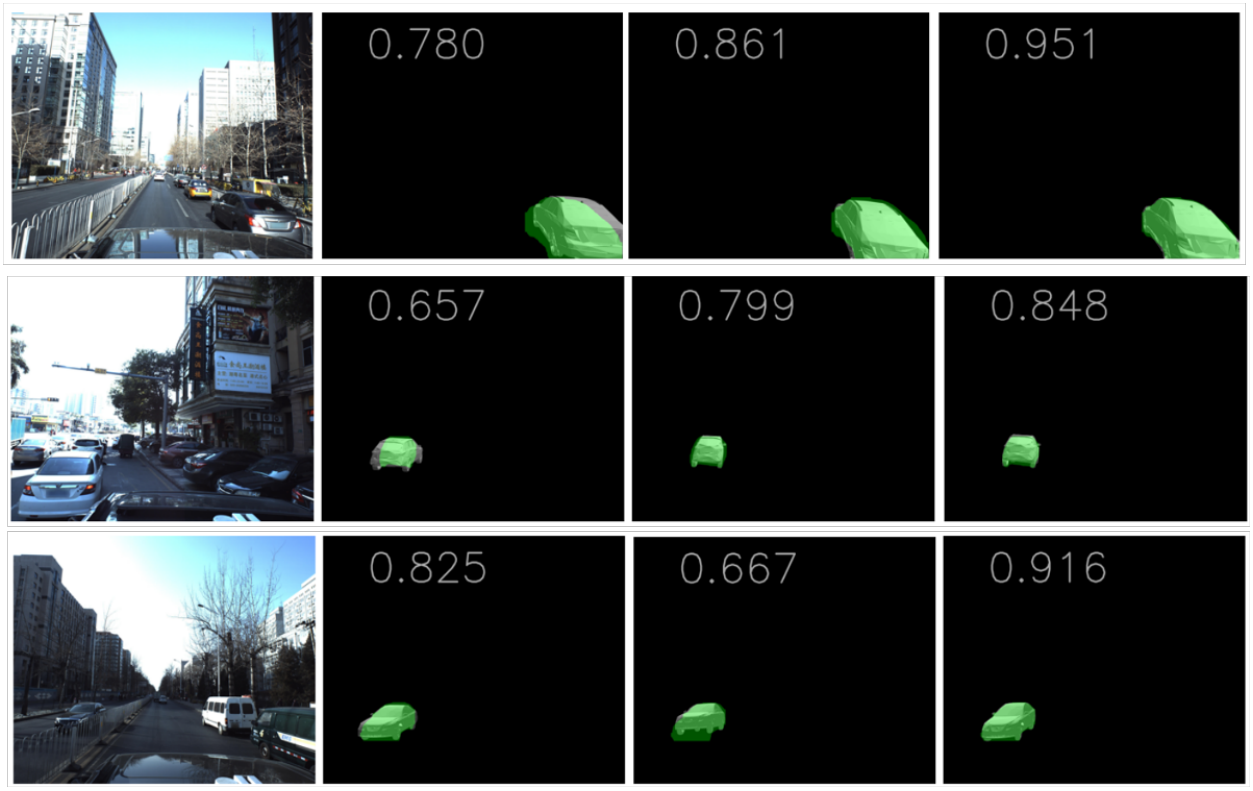

Figure 4.4: Visualization for FES optimization. The learning process takes R, T and D as the optimization variables. The numbers denote IoU score between mask from 3D mesh and mask from the Loc4Trk model, reflecting the accuracy of our pose and size estimation.

defined in Eq. (4.1), (4.2) and (4.4).

$$\begin{aligned} L_{total} =& w_{cls}L_{cls} + w_{box}L_{box} + w_{mask}L_{mask} \\ &+ w_{emb}L_{emb} + w_{ori}L_{ori} + w_{dis}L_{dis}. \end{aligned} \tag{4.5}$$

The outputs from Loc4Trk-Net are particularly crucial for localization and tracking problems, where both feature embedding and 3D information of the objects is of primary importance. Furthermore, the predicted orientation and distance can be further refined and serve as a good initialization for the subsequent 3D localization optimization part.

4.2 Fitness Evaluation Score

Although the orientation and distance estimation results from Loc4Trk-Net can deal with partial occlusions and truncation cases in most of the time, they are not accurate enough for 3D localization. As you may also notice, Loc4Trk-Net only focuses on the localization on 2D and 3D without considering the object size D, which is an important aspect of 3D detection. We adopt a lightweight optimization pipeline from [387] for single-view that refines the initial estimates to ensure localization robustness. Meanwhile, the size of the detected object can also be obtained through this refined optimization. A 3D deformable vehicle model is set up as prior information. An effective image gradient-based fitness evaluation score (FES) is then used to evaluate the fitness between the 2D projection of the 3D deformable vehicle model and raw image data. Moreover, the fitness evaluation is combined into an optimization framework to select better individuals from the combined parameter space based on an iterative population selection strategy. For more details, please refer to Sec: 3.2.

4.3 Data Association and Tracking

Camera Ego-Motion Estimation. Our 3D tracking is performed in the world coordinates. To transform from 3D camera coordinates to 3D world coordinates, the feature-based visual odometry [222] is introduced here to recover the camera pose through ORB features [262] extracted in every frame. Since ORB features must be located on the static background scene, instead of on the highly dynamic objects, we utilize the segmentation masks predicted from Loc4Trk-Net in Sec. 4.1 to discard those ORB features that are located on the detected objects, keeping those in the static background. We subsequently find correspondence of the background ORB features of the current frame with those of the previous frame. Outliers are further rejected by the RANSAC algorithm [77] as facilitated by the fundamental or homography matrix.

The camera motion is continuously estimated from time 0 to T: ${}^{w}C = \{{}^{w}c_t\}_{t=0:T}$. Given the measurements of the n^{th} sparse ORB features, which are anchored on the background: ${}^{i}p = \{{}^{i}p_t^n\}_{t=0:T}$ and their corresponding 3D positions: ${}^{w}P = \{{}^{w}P_t^n\}_{t=0:T}$. We formulate the camera

ego-motion tracking as the following:

$$ {}^{w}C, {}^{w}P = \underset{{}^{w}C, {}^{w}P}{\arg\min} \sum_{n=0}^{N} \sum_{t=0}^{T} \left\| r_p({}^{i}p_t^n, {}^{w}c_t, {}^{w}P_t^n) \right\|^2_{\sum_n^t}, \tag{4.6} $$

where $\|r_p()\|^2_{\Sigma} = {r_p}^T \Sigma^{-1} r_p$, the Mahalanobis norm. This is a common visual odometry formulation, where the camera poses are estimated based on a nonlinear least-squared formulation, also referred to bundle adjustment (BA) [305]. After we solve the camera poses, we can simply convert the object measurements from camera coordinates into world coordinates by using:

$$ {}^{w}S_t^k = {}^{w}{C_t}^{-1} \cdot {}^{c}S_t^k, \tag{4.7} $$

where the ${}^{w}S_t^k$ stands for the object state in world coordinates.

Assignment Problems. For simple design and real-time efficiency, we use a conventional way to solve the association between the predicted 3D Kalman states S and newly arrived measurements, which is to build assignment problems that can be solved using the Hungarian algorithm. The procedure can be described as follows:

- *State Prediction*: A 3D Kalman filter predicts the state of trajectories S_{t-1} to the current frame t as S_{est} during the state prediction step;

- *Data Association*: Into this problem formulation, we integrate motion and appearance information through combination of two appropriate metrics.

  To incorporate motion information, the detections N_t and predicted trajectories S_{est} are associated using the Hungarian algorithm. An affinity matrix is constructed by computing the 3D Intersection of Union (IoU) or negative center distance between every pair of the trajectory ${S_{est}}^i$ and detection N_t^j. To incoporate the appearance information, our second metric measures the smallest *cosine* distance between the i-th track and j-th detection in F appearance space. We combine both metrics to get matched trajectories and detections using a weighted sum following [336].

- *State Update*: The state of each matched trajectory in S_{match} is updated by the 3D Kalman filter based on the corresponding matched detection in N_{match} to obtain the final trajectories S_t.

- *Birth and Death Memory*: A memory buffer takes the unmatched detections $N_{unmatch}$ and unmatched trajectories $S_{unmatch}$ as inputs and creates new trajectories S_{new} and deletes disappeared trajectories S_{lost} from the associated trajectories.

Based on the tracking results, we are not only able to associate every object across frames, but also can deal with errors caused by the occlusions and missing detections. For those missing detections, we use Huber regression for detection interpolations.

4.4 Experiments

We evaluate our 3D detection and tracking pipeline on KITTI-MOT/MOTS benchmark [308] and nuScenes benchmark [20], featuring real-world driving scenarios and various road/ lighting conditions.

4.4.1 Dataset

KITTI-MOT/MOTS. KITTI-MOT is a driving scenario dataset for both Car and Pedestrian tracking task. It consists of 21 training sequences and 29 testing sequences, covering the street view, highway and pavement view. It is based on the KITTI Multi-Object Tracking (MOT) Evaluation and extends the annotations to the Multi-Object and Segmentation (MOTS) task. To this end, dense pixel-wise segmentation labels for every object are added. We only evaluate the performance for Car due to our proposed method.

nuScenes. nuScenes dataset is designed to support the task of 3D Multi-Object tracking (MOT) for autonomous vehicle. It provides 3D annotations for LiDAR data with 10 object classes for detection task, and 7 object classes for tracking task. There are 700 training sequences, 150 validation sequences and 150 test sequences in the nuScenes dataset.Every sequence collects images

at 12 FPS, denoted as full frames, and only those sampled keyframes, annotated at 2 FPS, are used for evaluation. Each sequence consists of about 40 keyframes per camera. Due to the low framerate, the inter-frame motion is significant.

KITTI-STEP. KITTI-STEP [331] is a driving scenario dataset for both car and pedestrian tracking task. It consists of 21 training sequences and 29 testing sequences. The evaluation metrics is the segmentation and tracking quality (STQ) consisting of two factors, association quality (AQ) and segmentation quality (SQ), that measure the tracking and segmentation quality respectively. KITTI-MOTS [308] has the same train and test sequences, and we evaluate our performance using HOTA metrics [198], which accumulates the soft number of true positives, false positives, and ID switches.

4.4.2 Evaluation Metric

3D Object Estimation. For both KITTI-MOT/MOTS and nuScenes, we use the formal evaluation metrics of detection 3D mAP from KITTI [90] in the 3D Object Detection Evaluation. We define the true positive of the object 3D estimation results if the 3D IOU is greater than 0.5 against the ground truth, as this IoU threshold is widely used and rather strict for image-based methods.

Multi-Object Tracking. The recently introduced HOTA for KITTI-MOT/MOTS measure decomposes into two intuitive terms, $DetA$ measuring detection accuracy and $AssA$ measuring association accuracy:

$$HOTA_{\alpha} = \sqrt{DetA_{\alpha} \cdot AssA_{\alpha}}. \tag{4.8}$$

Both terms are evaluated with respect to localization threshold α, and the final HOTA metric is integrated over localization thresholds.

The detection term $DetA$ is evaluated as the ratio of TP detections to the total number of TPs, FPs, and FNs (i.e., as intersection-over-union) and signals how well a tracker performs in terms of detection, ignoring the temporal aspect:

$$DetA_{\alpha} = \frac{|TP_{\alpha}|}{|TP_{\alpha}| + |TP_{\alpha}| + |TP_{\alpha}|}. \tag{4.9}$$

The association accuracy $AssA$ intuitively measures the number of frames in which the predicted track overlaps with the matched ground truth track. For each true positive detection in a predicted track p_t which is matched to a ground truth track g_t, $AssA$ computes the number of TP associations (TPA, detections in p_t which overlap with g_t), FP associations (FPA, detections in p_t which do not overlap with g_t), and FN associations (FNA, ground truth annotations in g_t which do not overlap with p_t). Then, association accuracy is evaluated as intersection-over-union over TPA, FPA and FNA sets, and averaged over TPs:

$$AssA_\alpha = \frac{1}{|TP_\alpha|} \sum_{c \in TP_\alpha} \frac{TPA_\alpha(c)}{TPA_\alpha(c) + FPA_\alpha(c) + FNA_\alpha(c)}. \tag{4.10}$$

nuScenes adopts a AMOTA as the metric for tracking, which is a weighted average of MOTA [10] across different output thresholds. Specifically,

$$AMOTA_\beta = \frac{1}{n-1} \sum_{r \in \left\{\frac{1}{n-1}, \ldots, 1\right\}} MOTA_r. \tag{4.11}$$

where r is a fixed recall threshold. The n = 40 and β = 0.2 (AMOTA@0.2), or β = 1 (AMOTA@1) are set by the official benchmarks.

4.4.3 Experiment Results

We demonstrate our system performance on different datasets under various driving scenarios which include far away objects, occlusions, truncations and adverse weather conditions. Some examples of re-projected images are shown in Fig. 4.5 to demonstrate our 3D object detection and tracking performance. Besides, the corresponding segmentation results for KITTI-MOTS are also shown.

KITTI-MOT/MOTS. For 3D object detection, as the KITTI-MOT tracking *test* set ground truth is not released to users, we have to use *val* set . Our framework is evaluated on both AP_{BEV} and AP_{3D} metrics and the Car class is split into 3 difficulties: *Easy*, *Moderate* and *Hard*. For 3D localization performance based on single frame images, we compare our Loc4Trk-Net with monocular 3D object detection methods. [235] uses CNNs to extract features from the 2D detected bounding boxes to infer orientation and dimension; 3D localization of objects are then obtained using

Table 4.1: Performance of 3D detection methods using different modality on KITTI-MOT/MOTS Car *val* set, ours is marked **bold**.

Method	Modality	Type	AP_{BEV} (IoU>=0.5)			AP_{3D} (IoU>=0.5)		
			Easy	Mod	Hard	Easy	Mod	Hard
M3D-RPN (CVPR '19) [15]	Single Image	Mono	41.53	31.02	26.65	37.41	27.11	23.73
Shift-RCNN (ICIP '19) [235]	Single Image	Mono	39.64	30.33	25.90	31.48	24.04	23.60
LOCNet + FES [374]	Single Image	Mono	50.69	36.17	31.97	48.40	38.59	32.69
3DOP (ICCV '19) [38]	Multiple Images	Stereo	54.83	43.36	37.15	53.73	42.27	35.87
SVB-Stereo (ECCV '20) [167]	Multiple Images	Stereo	58.52	46.17	43.97	48.51	37.13	34.54
Loc4TrkNet + FES (Ours)	Multiple Images	Mono	**57.16**	**44.72**	**38.29**	**48.40**	**38.59**	**32.69**
JMV3D (Ours)	Multiple Images	Mono	**62.49**	**49.91**	**44.13**	**53.76**	**42.13**	**37.91**

the geometric constraints between 3D points and 2D box edges. However, by considering geometric projection as the post-processing step, the error from 2D box detection, 3D object orientation and dimension regression can be aggregated in the subsequent distance estimation module. [28] consider the problem as a bundle adjustment problem (BA), where closed-form or iterative solutions can be applied by assuming a robust correspondence between 2D semantic keypoints and a 3D model of the object. However, these 2D keypoints largely depend on the training data and can be easily affected by partial occlusions or truncation. Furthermore, by considering the temporal information when dealing with multiple frames, we compare our overall JMV3D framework with [167] by adding 3D TrackletNet. Note that our method only requires the monocular input, we further achieve more gains and can be comparable with the state-of-the-art stereo-based image methods.

The performance of JMV3D on multi-object tracking and segmentation in the KITTI-MOT/MOTS is also shown in Table. 4.2. Upon the time of submission, we are the 1^{st} place among all the image-based methods. ViP-DeepLab [242] tries to approach the task by jointly performing monocular depth estimation and video panoptic segmentation, though they require additional ground-truth for training the depth esimation module. ReMOTS [354] proposed an intra-frame self-supervised

Table 4.2: Performance of multi-object tracking and segmentation methods using different modality on KITTI-MOTS Car *test* set, ours is marked **bold**.

Tracker	Modality	**HOTA** ↑	DetA ↑	AssA ↑	DetRe ↑	DetPr ↑	AssRe ↑	AssPr ↑	LocA ↑	IDS ↓
ViP-DeepLab [242]	Image	76.38	82.70	70.93	88.77	75.86	86.00	90.75	81.03	392
ReMOTS [354]	Image	71.61	78.32	65.98	83.51	87.42	68.03	90.61	89.33	716
PointTrackV2 [351]	Image	66.33	83.12	53.38	87.17	90.04	81.79	59.15	90.12	594
TrackR-CNN [308]	Image	56.63	69.90	46.53	74.63	84.18	63.13	62.33	86.60	1058
MOTSFusion [197]	Image	73.63	75.44	72.39	78.32	90.78	75.53	89.97	90.29	572
LidarMOTS [382]	LiDAR	68.11	77.26	60.61	84.50	85.61	64.95	82.35	89.50	835
UW_LIFTS [374]	LiDAR	83.21	81.22	80.37	86.33	90.32	85.28	90.09	90.06	73
JMV3D (Ours)	Image	**79.57**	**79.66**	**80.00**	**83.05**	**90.35**	**83.08**	**91.42**	**90.06**	**114**

triplet construction network to learn mask features for both training and testing set for Re-ID. PointTrack V2[351] distinguish the foreground and background by regarding the object's mask and its surrounding environment as two sets of 2D point clouds. However, these two methods are assuming the accurate initialization of segmentation from a pretrained optical-flow estimation network. UW_LIFTS [374] is one of the leading algorithms in the benchmark. However, their method requires the LiDAR point cloud as the inputs and post-processing steps including optical-flow guided instance mask segmentation and object Re-ID are applied, which can be hardily adapted to the autonomous driving applications.

nuScenes. Our JMV3D method performs well in multi-object tracking among all the published methods, which is shown in Table 4.3. CentertTrack-Vision [400] uses two consecutive frames to generate inter-frame motion for object detection and 3D tracking. CenterTrack-Open [400] fuses LiDAR information in the CenterTrack-Vision pipeline with Megvii-detector [404] to generate 3D detection. The LiDAR-based baselines uses the state-of-the-art LiDAR-based detectors to estimate accurate bounding boxes and feed into a 3D Kalman Filter based 3D tracker, AB3DMOT

Figure 4.5: Qualitative examples under diverse scenarios. The top row are the results on the KITTI-MOT/MOTS dataset, and the bottom row show the results on some image frames of the nuScenes dataset (daytime and night). The system is capable of estimating the shape and pose (without scale-factor ambiguity) of objects.

[334]. Our methods only takes monocular images for 3D object detection and tracking and leverage the discriminative appearance features compared to the prior arts. he nuScenes tracking dataset linearly interpolates GT tracks to avoid track fragments from LiDAR point filtering and removes GT objects without LiDAR points. Both invisible objects with annotation and visible objects without annotation prohibit the camera-based methods from optimizing bounding box estimation. Nevertheless, our JMV3D approach reaches **0.186** AMOTA with near four times tracking accuracy of the best vision-only submission among all published methods.

4.4.4 Ablation Study

Adapt to KITTI-STEP.

We adopted the DeepLabV3+[33], which is the state-of-the-art method in KITTI semantic dataset for semantic segmentation. To improve semantic segmentation results, Atrous Convolution is utilized in DeepLabV3+, for integrating global and local features for the network. Furthermore, Zhu et al.[410] introduce a simple yet efficient data augmentation pipeline for improving Semantic

Table 4.3: Performance of multi-object tracking methods using different modality on nuScenes Car *test* set, ours is marked **bold**.

Tracker	Modality	**AMOTA@1** ↑	AMOTP ↓
CenterTrack-Vision [400]	Image	0.046	1.543
Megvii-AB3DMOT [20]	LiDAR	0.151	1.501
PointPillars-AB3DMOT [20]	LiDAR	0.029	1.703
Mapillarry-AB3DMOT [20]	LiDAR	0.018	1.790
CenterTrack-Open [400]	Multi-Modal	0.108	0.989
QD3DT [120]	Multi-Modal	0.217	1.550
JMV3D (Ours)	Image	**0.186**	**1.429**

Segmentation training. They take the image I_t and label L_t as reference jointly and predict images $I_{t\pm s}$ and labels $L_{t\pm s}$ for data augmentation. As a result, the dataset can be scaled by a factor $2k + 1$. Besides that, Boundary Label Relaxation is introduced for better object semantic boundary estimation. Thus, by combining the above instance results from 4.1 and semantic segmentation methods, we achieve segmentation quality SQ of 64.04 in the KITTI-STEP.

The performance of KITTI-STEP using STQ, which measures segmentation as well as detection and tracking quality. Our method currently ranks the first place among the total valid submissions. The performance of top-selected algorithms among all competitors is shown in Table 4.4.

Effectiveness of joint training with various metric losses on tracking performance. We first demonstrate the benefits for joint training for the embedding head in Table 4.5. *Sparse GT* means the method only considers ground truth labels as matching candidates when learning object association. *Pairwise Contrasitive Loss* with one/multiple positive targets is our method, which is introduced in Sec. 4.1. Compared to learning with sparse ground truths using conventional *Triplet-Hard Loss* [310], our method (**bold**) improves the overall HOTA by 6.41 points. The significant improvement on AssA also indicates our method greatly improves the feature embeddings and enables more accurate associations.

Table 4.4: Competition results on KITTI-STEP *test* set, ours is marked **bold**.

Method	**STQ** ↑	AQ ↑	SQ(IoU) ↑
Motion-DeepLab[331]	52.19	45.55	59.81
HybridTracker	54.99	54.44	55.54
siain	57.87	55.16	60.71
EffPS_MM	62.93	61.49	64.41
REPEAT	67.13	65.81	**68.49**
JMV3D (Ours)	**67.55**	**71.26**	64.04

Table 4.5: Ablation study on joint training with various metric losses on KITTI-MOTS *val* set.

Dataset	Joint	Loss function	**HOTA** ↑	DetA ↑	AssA ↑
KITTI-MOTS		Triplet Hard (Sparse GT) [310]	73.16	80.70	66.32
	✓	Pairwise (One Positive)	78.53	79.59	77.48
	✓	Pairwise (Multiple Positive)	**80.55**	**79.92**	**81.18**
	✓	Cross-Entropy (Sparse GT)	78.27	79.34	77.93

We further analyze the improvements of using various metric loss in details. In Table 4.5, we can observe that when we match each training sample to more negative samples augmented by RPN and train the feature space, the HOTA is significantly improved by 5.37 points. This experiment shows that more contrastive targets, even most of them are negative samples, can improve the feature learning process. The multiple-positive contrastive learning following Equation 4.1 further improves the HOTA by 2 point (78.53% to 80.55%). Moreover, compared to the *Cross-Entropy Loss* which is widely used in multi-object tracking methods [330, 384], our method achieves a gain of 2.28 points.

Effectiveness of amount of training data and FES on localization performance. We train the Loc4Trk-Net with 10%, 50%, and 100% of training data on KITTI-MOTS. The results show how we can benefit from more data in Table 4.6, where a consistent trend of performance improvement

Table 4.6: Ablation study on amount of training data and FES on 3D detection performance.

Dataset	Amount	FES	AP_{3D} (IoU>=0.5)		
			Easy	Mod	Hard
KITTI-MOTS	10%		14.61	8.45	4.10
		✓	27.34	19.68	12.16
	50%		35.57	31.51	24.61
		✓	41.21	35.49	29.44
	100%		48.40	38.59	32.69
		✓	**53.76**	**42.13**	**37.91**

Table 4.7: Ablation study on each component for data association on KITTI-MOTS *val* set.

Dataset	F_{emb}	${}^{w}C$	box_{2d}	box_{3d}	**HOTA** ↑	DetA ↑	AssA ↑
KITTI-MOTS	✓		✓		73.00	78.96	67.49
				✓	76.93	81.02	73.06
		✓		✓	77.52	81.02	74.18
	✓			✓	79.89	79.92	79.88
	✓	✓		✓	**80.55**	**79.92**	**81.18**
	✓	✓	✓	✓	78.44	79.66	77.25

emerges as the number of data increases. The trend of our results indicates that large-scale 3D annotation is helpful, especially with the ground truth of distant and small objects. However, even with limited amount of training data, FES is introduced to correct the mistake made by Loc4Trk-Net by only relying on stable and invariant edge information from raw 2D images and thus avoid time-consuming training and dataset labeling. We design the framework in a complementary manner by utilizing deep learning approaches and conventional optimization techniques, which ensures the accuracy of the localization performance and can further benefits the 3D tracking performance.

Effectiveness of each component for data association. We perform an ablation study on each

component for data association. F_{emb} is for the appearance feature embedding, ${}^{w}C$ is the camera ego-motion, box_{2d} and box_{3d} are the location and motion priors, which stand for the 2D and 3D bounding boxes. As shown in Table 4.7, without appearance features, the tracking performance is consistently improved with the introduction of additional 3D localization information and camera ego-motion. When involving the appearance feature embeddings, the AssA achieves a significant improvement (74.18% to 79.88%), which shows the embedding is of a great importance to the similarity measurement. However, we also notice by taking advantage of both 2D and 3D bounding boxes information as the object state in Kalman Filter, the performance of HOTA drops. This is due to the 2D bounding boxes information is redundant when having 3D estimation since these information can be obtained by projecting the 3D bounding boxes to the image plane using the camera intrinsics K.

4.5 Summary

In this chapter, we propose an joint online monocular 3D localization, tracking and segmentation pipeline, combining with pairwise contrastive learning and 3D instance estimation, to tracking moving vehicles in a 3D world. Our proposed pipeline consists of four parts: an RCNN-based Localization for Tracking Network (Loc4Trk-Net), cross-frames contrastive feature learning modules, a fitness evaluation score (FES) based single-frame optimization, and a simple but effective 3D Kalman filter. Extensive experiments and ablation studies have shown our method is effective and robust under different autonomous driving scenarios. Overall, our method ranks 1^{st} place on the KITTI-MOTS leaderboard and also achieves impressive results among all image-based solutions on nuScenes 3D tracking benchmark.

Figure 4.6: Visualization of the learned attention of the model for orientation estimation. The heatmap shows the image areas that contribute to orientation estimation the most. The network attends to certain meaningful parts of the car such as tires, lights, and side mirrors.

Chapter 5

DEPTH-GUIDED MONOCULAR 3D OBJECT DETECTION VIA COARSE-TO-FINE TRAINING

Autonomous driving vehicles and robots will transform the modern world just as cars did a century back. It is vital for today's autonomous perception systems to perceive the world the same way as people do. 3D object detection enables us to capture an object's relative size, pose, and depth information. Among them, depth information can be accessed with the help of LiDAR-scanned point clouds or object-centric stereo matching. However, it usually comes with costly expenses by adding new sensors [229, 324, 325] or higher computational costs [318, 375]. On the other hand, extrapolating depth information from monocular images can be proved to be a viable cost-effective alternative for large-scale deployment with sufficient advancement from its present-day performance.

Conventional monocular 3D object detection involves 2D localization [255, 177, 300, 401] followed by generating 3D object centers from the predicted heatmaps. The model learns the relative size, depth, and pose information with the help of local visual features around the projected 3D object center. This lack of scene-level attention to different objects and contextual cues causes the predictions not to account for inter-object depth relations, which ultimately leads to inadequate performance. Other approaches involve the Pseudo-LiDAR mechanism [207, 318, 335], which convert the estimated dense depth maps to 3D point clouds and run LiDAR-based object detectors on top of them. However, these methods, though better localize the objects with the help of estimated depth, may suffer from the risk of predicting 3D detection on inaccurate depth maps. Additionally, the additional depth estimator incurs a large overhead in inference.

To tackle these issues, we propose a transformer-based framework, the **MONO**cular **3D** object detection via **C**oarse-to-**F**ine **T**raining **(Mono3DCFT)**. It presents a novel depth-guided feature

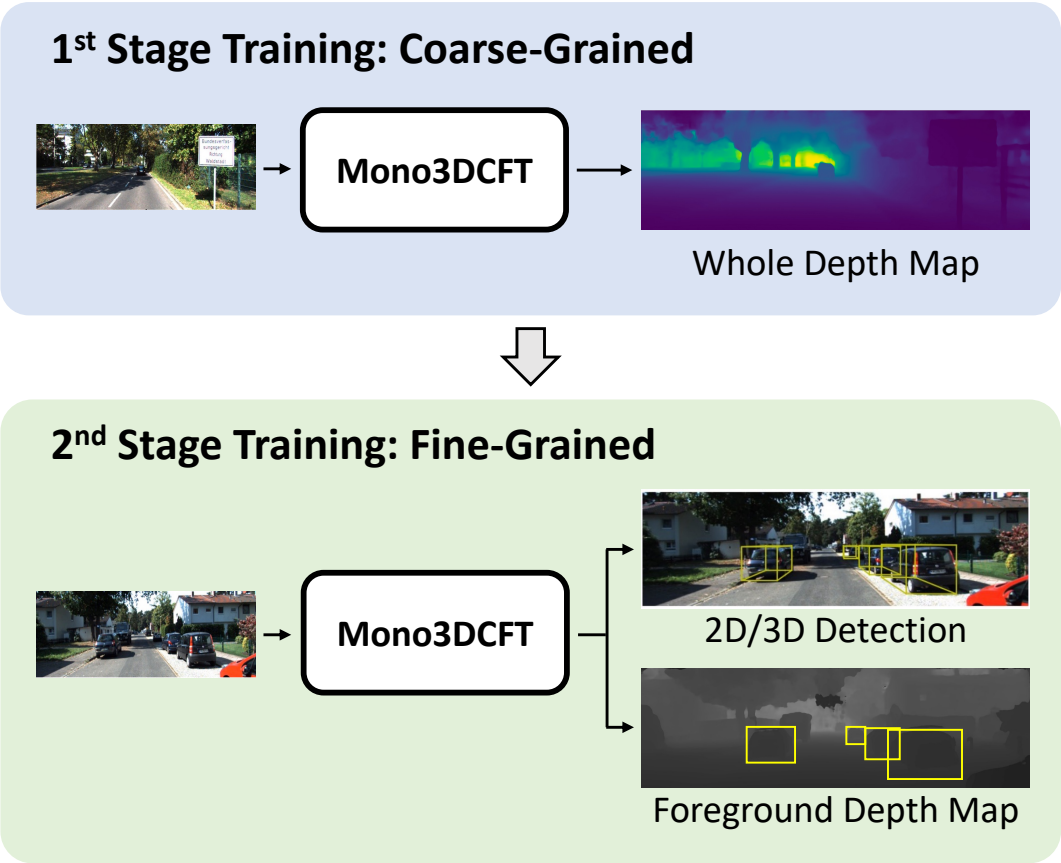

Figure 5.1: The proposed two-stage coarse-to-fine training framework. We first perform *coarse-grained* training with the whole dense depth map to better learn the high-level representation of the depth information and then perform *fine-grained* training with foreground object-wise depth labels. The same encoder architecture is used for both stages.

aggregation framework to adaptively estimate each object's 3D attributes based on global context via a two-stage training scheme. The Mono3DCFT mainly consists of a fusion-in-the-backbone encoder and a depth-guided transformer decoder. The fusion-in-the-backbone encoder is modified from Swin-Transformer [187] by adding a multi-scale depth encoder with spatial and cross attention to extract both appearance and depth information. A gating cross-attention fusion block is proposed to learn better coupling features that fuses the geometric and appearance information of the input image. During the 1*st* stage coarse-grained training, the encoder is trained on the whole dense depth map to better capture the depth cues from the high-level semantic information of the image.

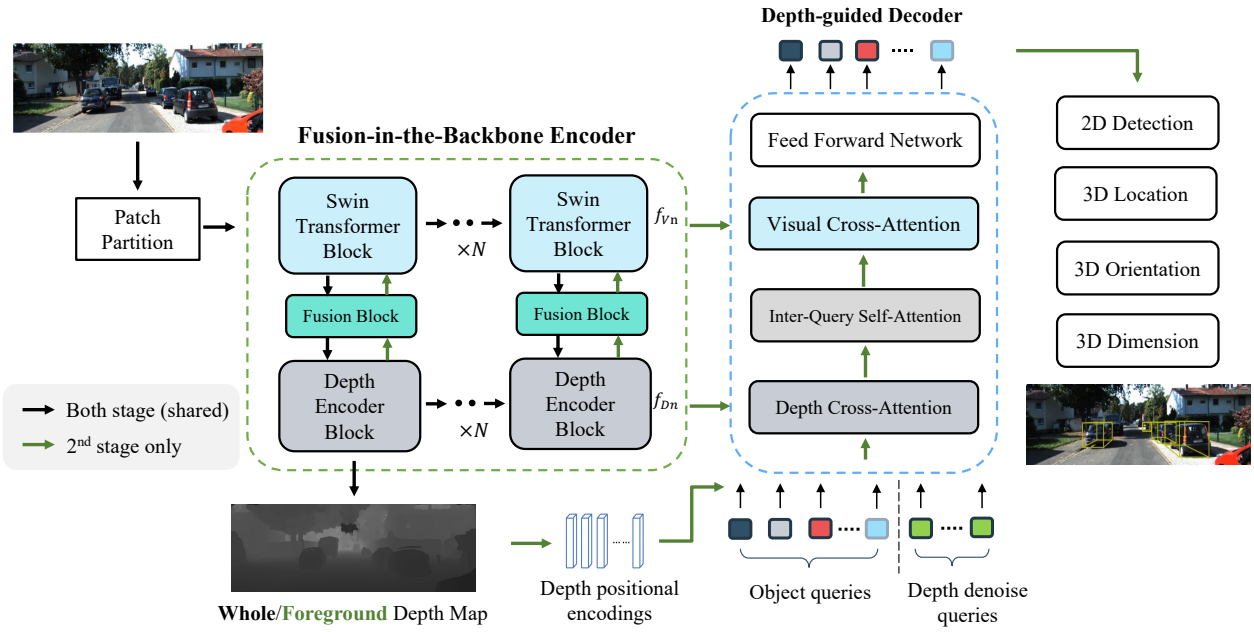

Figure 5.2: Mono3DCFT uses a fusion-in-the-backbone encoder to encode the visual and depth features. Then, a depth-guided decoder is adopted to adaptively aggregate scene-level features for object queries for predicting the 2D and 3D attributes of the objects. The gating cross-attention fusion block is proposed to better learn the fused features for appearance and geometric information in the 2*nd* stage.

Then, the depth-guided decoder can be directly concatenated to the encoder in the 2*nd* stage fine-grained training on the object-wise foreground depth map. This enables few changes to the encoder architecture, which significantly reduces the training cost and avoids obtaining inaccurate depth priors from the pre-trained depth estimator. Furthermore, we introduce the depth positional encoding, and the depth deNoising queries to involve depth-aware hints to the transformer, achieving better performance on monocular 3D object detection.

5.1 Mono3DCFT: a transformer-based monocular 3D object detector via coarse-to-fine training

Figure 5.2 presents the pipeline of Mono3DCFT, which follows the DETR-type [24] and mainly consists of three components: a fusion-in-the-backbone transformer encoder, a depth-guided trans-

former decoder, and several 2D-3D detection heads. In Section 5.1.1, we first introduce the feature extraction of the appearance and depth information. Then, in Section 5.1.2, the depth-guided decoder is proposed to adaptively aggregate the scene-level information for object queries. In Section 5.1.3, we present our prediction of 3D attributes and their loss functions. More specifically, a novel two-stage training based on a coarse-to-fine paradigm is illustrated in Section 5.1.4.

5.1.1 Fusion-in-the-backbone Encoder

Hierarchical Visual Encoder. Our framework utilizes a Swin-Transformer [187] as the visual backbone. The raw image $I \in \mathbb{R}^{H \times W \times 3}$, where H and W denote its height and width respectively, first enters the Patch Partition module to cut the image into patches without overlapping. Each patch is regarded as a token, which represents the features of the corresponding position of the original image. Then through the Linear Embedding module, it maps the channel dimensions of each patch to the specified value C. These tokens are sent to the hierarchical Swin-Transformer Block for processing, where the features pass through the W-MAS/SW-MSA layer in the n-th block. In each block, there are three basic elements of Q(query), K(key), and V(value) for self-attention mechanism:

$$\text{self-attn}(Q, K, V) = Softmax\left(\frac{Q \cdot K^T}{\sqrt{d}} + B\right) \cdot V, \tag{5.1}$$

where d is the dimension of Q and K, and B is the relative position bias in the Swin-Transformer block that is different from the traditional self-attention blocks.

We follow the original Swin-Transformer's implementation with the patch size - 4×4, where we obtain its multi-scale feature maps, $f_{V_{1:4}}$, as shown in Figure 5.2. The feature outputs at each stage then enter the multi-scale depth encoder modules, and the features of different scales are input to the Attention module for subsequent processing of depth embeddings.

Multi-Scale Depth Encoder. Since each stage of the visual encoder models local features, and reduces the height and width of the features to expand the receptive field, we propose a multi-scale depth encoder to capture global information. The depth encoder block is composed of channel attention and spatial attention inspired by [337]. Taking the feature f_{V_n} as an example, the process

is as follows:

$$\begin{aligned} f_{V_n}^{'} &= H_c(f_{V_n}) \otimes f_{V_n}, \\ f_{D_n}^{"} &= H_s(f_{V_n}^{'}) \otimes f_{V_n}^{'}. \end{aligned} \tag{5.2}$$

The channel attention is a 1D mapping $H_c \in \mathbb{R}^{C\times1\times1}$, and the spatial attention is a 2D mapping $H_s \in \mathbb{R}^{1\times\frac{H}{4}\times\frac{W}{4}}$, where$\otimes$ represents the element-wise multiplication. In the multiplication process, the channel attention values are propagated along the spatial dimension, and the spatial attention values are propagated along the channel dimension. The input feature maps first pass through the average-pooling layers and the max-pooling layers, respectively, followed by an MLP. The two parts that pass through different pooling layers are then element-wise summed,

$$\begin{aligned} H_c(f_{V_n}) = \sigma\,(&MLP\,(AvgPool\,(f_{V_n})) \\ &+ MLP\,(MaxPool\,(f_{V_n}))), \end{aligned} \tag{5.3}$$

where σ is a Sigmoid function. There is a ReLU activation function after each pooling layer. Note that, spatial attention follows channel attention; more specifically, convolution is first performed to halve the number of feature map channels, and then the final result is obtained through operations such as average pooling, maximum pooling, and concatenating. The process is as follows:

$$H_s(f_{V_n}^{'}) = \sigma\,(\underset{(7\times7)}{conv}(Concat\begin{bmatrix} AvgPool(\underset{(1\times1)}{conv}(f_{V_n}^{'})) \\ MaxPool(\underset{(1\times1)}{conv}(f_{V_n}^{'})) \end{bmatrix})), \tag{5.4}$$

where $conv_{(1\times1)}$ is a convolutional layer with kernel size 1×1, and the $conv_{(7\times7)}$ is the one with the kernel size 7×7. After passing through the depth encoder attention blocks, the depth features of different scales f_{D_n} up-sampled and then concatenated in the channel dimension to obtain the depth feature map $f_D \in \mathbb{R}^{\frac{H}{16}\times\frac{W}{16}\times C}$.

To supervise the depth features, we predict the depth map $D \in \mathbb{R}^{\frac{H}{16}\times\frac{W}{16}\times(k+1)}$ by applying a 1×1 convolution on top of f_D. Here, we discretize the depth into $k + 1$ bins following [249], where the first ordinal k bins represent foreground depth and the last one denotes the background. We adopt linear-increasing discretization (LID) since the depth estimation of farther objects inherently yields

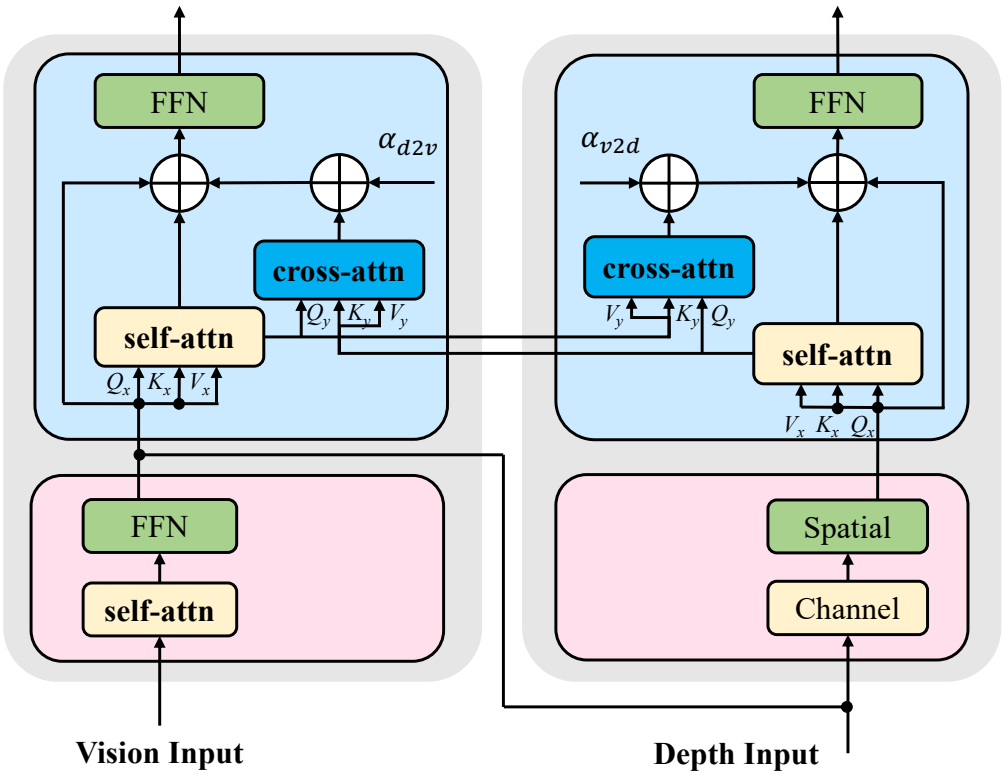

Figure 5.3: Illustration of gating cross-attention fusion block. (x, y) are the (vision, depth), and both α are learnable scalars.

larger errors, which can be suppressed with a wider categorization interval. We limit the depth values within $[d_{min}, d_{max}]$ and set both the first interval length and the common difference of LID as δ. We then categorize a ground-truth depth label d of an object into the k-th bin as

$$k = \lfloor -0.5 + 0.5\sqrt{1 + \frac{8(d - d_{min})}{\delta}} \rfloor, \quad \text{where } \delta = \frac{2(d_{max} - d_{min})}{k(k+1)}. \tag{5.5}$$

Focal loss [177] is used here to supervise the categorical depth prediction for the pixels in D, denoted as $\mathcal{L}_D$.

Gating Cross-Attention Fusion Block. By channel-wise attention and spatial-wise attention, the depth encoder explores long-range dependencies of depth values from different image regions,

which provide the network with non-local cues of the entire space. Additionally, the coupling of depth and visual encoders can allow the model to better learn features that can fuse the spatial and appearance information of the input image. Instead of having few dedicated transformer layers on top of the image and depth encoders for fusion [377], we propose to directly insert cross-attention modules into the vision and depth encoders in the backbone for fusion and include gating mechanism for the cross modal layers (shown in Figure 5.3). Specifically, at each encoding layer, we have:

$$\begin{aligned} \tilde{x} &= \text{self-attn}(x), \\ x &= x + \tilde{x} + \alpha * \text{cross-attn}(\tilde{x}, y), \\ x &= x + \text{FFN}(x), \end{aligned} \tag{5.6}$$

where α is a learnable parameter initialized to zero. By inserting cross-attention layers with gating mechanism, we enable the cross-modal interactions without affecting the original computational flow of the backbones at the beginning of the model training. We can switch from a single interaction (vision-to-depth) to dual interactions (vision-to-depth & depth-to-vision) easily.

5.1.2 DeNoising Depth-guided Decoder

In the image-based 3D detection task, the object size at far and near distances in the image varies significantly due to the perspective projection [401], which makes it challenging to utilize the traditionally learned object queries. Thus, we propose adopting depth-aware features as the input of the transformer decoder to fully represent the object's attributes and handle complex scale-variant situations. The decoder is built upon [381], where cross-attention modules inside can efficiently model the relationship between context- and depth-aware features, as shown in Figure 5.2.

Depth Positional Encodings (DPE). Previous methods only use visual information and thus, lack depth cues for the detector. In the decoder cross-attention layer, we add learnable positional encodings, instead of using sinusoidal functions, as an alternative attention mechanism, as shown in Figure 5.2. According to [381, 122], inserting depth hints as depth embedding into the detector provides more detailed information for small objects. The set of learnable embeddings, $p_D \in \mathbb{R}^{(d_{max}-d_{min}+1)\times C}$ is obtained by the weighted summation of the depth-bin confidences and their

corresponding depths, where each row encodes the depth positional information in terms of meters, ranging from d_{min} to d_{max}. By pixel-wise addition of f_D with such encodings, object queries can capture more sufficient depth cues in the depth cross-attention layer and better understand the 3D scene.

Depth DeNoising Queries. The dynamic nature of DETR-like models introduces an instability problem in training due to discrete bipartite matching and stochastic training. This causes the slow convergence issue and creates inconsistent optimization goals for decoder queries in the training. The training process can be assumed to learn good anchors as well as good relative offsets, both of which are interdependent. We propose to add noised depth queries as learnable parameters into the transformer decoder along with object queries. The denoising task similar to [164], which is introduced to remove the noise, acts as a training shortcut to enable quicker learning of relative offset since the denoising task bypasses bipartite matching completely. The noised depth can be treated as a good value indicating the closeness of the query to the actual depth of the object of focus, thus creating a more distinct optimization goal for the model. To put it differently, it introduces a new loss parameter that encourages the model to reconstruct original depth values, i.e., z_{3D}. As far as we know, this is the first time deNoising is used to integrate into monocular 3D object detection.

5.1.3 2D-3D Detection Heads

The depth-aware object embeddings are fed into a series of MLP-based heads for 2D and 3D attribute estimation following the depth-guided transformer. We integrate the attributes during inference to directly generate 3D bounding boxes as outputs, requiring no non-maximum suppression (NMS) post-processing. We use the Hungarian algorithm [24] to match the orderless queries with ground-truth labels for trainingwithout using any rule-based label assignment.

Object Category and 2D Bbox (cls, x, y, w, h)**.** We detect objects of three categories in KITTI [91], car, pedestrian and cyclist, and adopt Focal loss [177] for optimization, denoted as $\mathcal{L}_{cls}$. We obtain the 2D bounding box of an object by predicting four parameters, x, y, w, h. We apply L1 loss for the distances and generalized IoU (GIoU) loss [257] for the recovered 2D bounding box following DETR [24], denoted as $\mathcal{L}_{xywh}$ and $\mathcal{L}_{GIoU}$, respectively.

3D Location (x_{3D}, y_{3D}, z_{3D})**.** We directly output the 3D coordinate (x_{3D}, y_{3D}, z_{3D}) of each query. Following [373], Huber loss is adopted to describe the penalty in location estimation: given predicted 3D translation $t = [x_{3D}, y_{3D}, z_{3D}]$ and ground truth $\hat{t}$, the 3D location loss is

$$\mathcal{L}_{loc}(t, \hat{t}) = \begin{cases} \frac{1}{2}(t - \hat{t})^2 / \Delta & \text{if } \left|t - \hat{t}\right| < \Delta, \\ \left|t - \hat{t}\right| - \frac{1}{2}\Delta & \text{otherwise,} \end{cases} \tag{5.7}$$

where the hyper-parameter Δ controls the range of outliers.

3D Dimension and Orientation. We predict the residuals to the mean shape values for 3D sizes and divide the heading angle into multiple bins with residuals and adopt MultiBin loss [43, 401] to optimize the prediction of orientation. The two losses are respectively denoted as $\mathcal{L}_{dim}$ and $\mathcal{L}_{ori}$.

Bipartite Matching. To correctly match each query with a ground-truth object, we calculate the loss for each query-label pair and utilize the Hungarian algorithm to find the globally optimal matching. For each pair, we integrate the losses of five attributes into two groups. The first group contains the object category and the 2D size. The second group consists of 3D location, size, and orientation, which are 3D spatial attributes of the object. We respectively sum the losses of two groups and denote them as $\mathcal{L}_{2D}$ and $\mathcal{L}_{3D}$. With the help of depth positional encoding as well as depth deNoise queries to stabilize the training, the network predicts reasonable 3D attributes even at the beginning of the training. Thus, we utilize both $\mathcal{L}_{2D}$ and $\mathcal{L}_{3D}$ as the matching cost during matching each query-label pair.

5.1.4 Two-Stage Training Mechanism and Loss

We realize that better performance of region-level monocular 3D detection needs better dense depth prediction/initialization which in turn requires a global-level semantic understanding of the image. Though these two kinds of tasks seem different, they are both inherently similar and require the fusion of vision and depth modalities. Therefore, sharing as many parameters as possible between the models used for these two tasks can be beneficial. Here, we propose a two-stage training paradigm, where we first train the models with global-level objectives and then perform further training with region-level goals. In this way, the coarse-grained supervision from the first stage can

Table 5.1: **Performance of the car category on KITTI *Test* and *Val* sets.** We use bold numbers to highlight the best results and use blue-colored numbers for the second-best outcome.

Method	Extra data	AP_{3D}@IoU=0.7, Test			AP_{BEV}@IoU=0.7, Test			AP_{3D}@IoU=0.7, Val		
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
SMOKE [188]	None	14.03	9.76	7.84	20.83	14.49	12.75	14.76	12.85	11.50
MonoPair [43]		13.04	9.99	8.65	19.28	14.83	12.89	16.28	12.30	10.42
RTM3D [168]		13.61	10.09	8.18	-	-	-	19.47	16.29	15.57
PGD [315]		19.05	11.76	9.39	26.89	16.51	13.49	19.27	13.23	10.65
IAFA [394]		17.81	12.01	10.61	25.88	17.88	15.35	18.95	14.96	14.84
MonoDLE [208]		17.23	12.26	10.29	24.79	18.89	16.00	17.45	13.66	11.68
MonoRCNN [275]		18.36	12.65	10.03	25.48	18.11	14.10	16.61	13.19	10.65
MonoGeo [385]		18.85	13.81	11.52	25.86	18.99	16.19	18.45	14.48	12.87
MonoFlex [386]		19.94	13.89	12.07	28.23	19.75	16.89	23.64	17.51	14.83
GUPNet [195]		20.11	14.20	11.77	-	-	-	22.76	16.46	13.72
Kinematic3D [17]	Multi-frames	19.07	12.72	9.17	26.69	17.52	13.10	19.76	14.10	10.47
MonoRUn [30]	LiDAR	19.65	12.30	10.58	27.94	17.34	15.24	20.02	14.65	12.61
CaDDN [249]		19.17	13.41	11.46	27.94	18.91	17.19	23.57	16.31	13.84
AutoShape [190]	CAD	22.47	14.17	11.36	30.66	20.08	15.59	20.09	14.65	12.07
PatchNet [205]	Depth	15.68	11.12	10.17	22.97	16.86	14.97	-	-	-
D4LCN [60]		16.65	11.72	9.51	22.51	16.02	12.55	-	-	-
DDMP-3D [311]		19.71	12.78	9.80	28.08	17.89	13.44	-	-	-
MonoDTR [122]		21.99	15.39	12.73	28.59	20.38	17.14	24.52	18.57	15.51
DD-3D w. 15M [231]		23.19	**16.87**	**14.36**	32.35	**23.41**	**20.42**	30.89	23.92	21.10
Mono3DCFT (Ours)	Depth	**24.69**	16.82	14.13	**33.62**	22.30	18.90	29.63	22.01	16.96

provide good initialization for the second stage for all the shared parameters.

1*st*-stage Coarse-grained Training. In the 1*st*-stage of training, only the fusion-in-the-backbone transformer encoder is adopted. We use per-pixel depth predictions and pixels that have valid ground-truth dense depth. The α_{v2d} and α_{d2v} in the gating cross-attention fusion block separately

denote the cross-modal interactions for visual-to-depth and depth-to-visual. The α_{d2v} is set to 0 (switched-off) in the first stage while we keep the α_{d2v} learnable (switched-on), as indicated in the green arrow in Figure 5.2. The entire fusion-in-the-backbone encoder is trained on the whole dense depth map to better learn the high-level semantic representation of the image, so the overall loss for stage one is just the categorical depth focal loss $\mathcal{L}_{1st} = \mathcal{L}_{D_{map}}$.

2*nd*-stage Fine-grained Training. After the encoder is trained in 1*st*-stage, the depth-guided transformer decoder can be directly concatenated to the encoder, with few changes to the encoder architecture. This enables the sharing of as many parameters as possible between the two stages while avoiding introducing extra computational cost and inaccurate depth priors from off-the-shelf depth estimators [318, 365]. We switch on the interactions and set α_{v2d}, and α_{d2v} learnable, thus better representations to fuse the geometric and appearance information can be learned through the coupling. Unlike the 1*st* stage trained on the whole depth map, the 2*nd* stage is trained on foreground depth and supervised only by object-wise depth labels. After query-label matching, we obtain N_{gt} valid pairs out of N queries, where N_{gt} denotes the number of ground-truth objects. Then, the overall loss in the 2*nd* fine-grained stage is formulated as,

$$\mathcal{L}_{2nd} = \frac{1}{N_{gt}} \cdot \sum_{n=1}^{N_{gt}} (\mathcal{L}_{2D} + \mathcal{L}_{3D}) + \mathcal{L}_{Dobj}, \tag{5.8}$$

where $\mathcal{L}_{Dobj}$ represents the loss of the predicted categorical foreground depth.

5.2 Experiments

5.2.1 Datasets and Implementation

Datasets.

KITTI-3D. The KITTI-3D detection benchmark [91] consists of urban driving scenes with 8 object classes. The benchmark evaluates 3D detection accuracy on three classes (Car, Pedestrian, and Cyclist) using two average precision (AP) metrics computed with class-specific thresholds on intersection-over-union (IoU) of 3D bounding boxes or Bird-Eye-View (2D) bounding boxes. We refer to these metrics as 3D AP and BEV AP. We use the revised AP|R40 metrics. The training

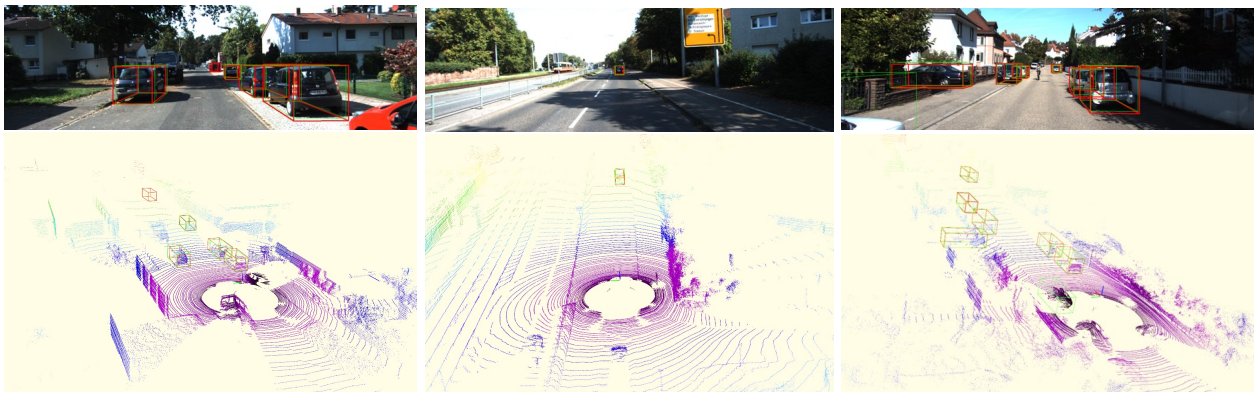

Figure 5.4: The above figure represents the qualitative results on KITTI *val* set. Our predictions are shown in red 3D boxes, while ground truths are represented by green 3D boxes. LiDAR signals are only used for visualization. It can be best viewed in color with zoom-in.

set consists of 7481 images, and the test set consists of 7518 images. The objects in the test set are organized into three partitions according to their difficulty level (easy, moderate, hard), and are evaluated separately. We follow the common practice of splitting the training set into 3712 and 3769 images and report validation results on the latter. We refer to these splits as KITTI-3D train and KITTI-3D val.

KITTI-Depth. We use the KITTI-Depth dataset to perform the 1*st*-stage training. It contains over 93 thousand depth maps associated with the images in the KITTI raw dataset. The standard monocular depth protocol is to use the Eigen splits. However, as described in [276], up to a third of its training images overlap with KITTI-3D images, leading to biased results for models. To avoid this bias, we follow [231] to generate a new split by removing geographically close training images (i.e., within 50m) to any of the KITTI-3D images. We denote this split by Eigen-clean and use it to fine-tune the depth predictor of our model.

Implementation Details. As mentioned in Section 5.1, we utilize a Swin-Tiny [187] for our visual encoder in the fusion-in-the-backbone encoder. We follow the same practice in [381] and build upon it to use 3 Transformer decoder blocks and 8 heads for all attention modules in the depth-guided decoder. For the depth denoising, we add uniform noise to depth and set the noise hyper-parameters as 0.2 and 0.1 as in [164] and use 5 depth denoising groups in total. We set the number of depth

queries N as 50. The Δ in the $\mathcal{L}_{loc}$ is set to 1.6m. We use the AdamW optimizer with weight decay of 1×10^{-4} and train our model on 1 Nvidia V100 GPU. The batch size is set to 8, and the initial learning rate is 1×10^{-4}. We train our model with 50 epochs in the 1*st*-stage coarse-grained training on the dense depth map. In the 2*nd*-stage fine-grained training, we drop LR at 65 and 90 epochs by a factor of 0.1.

5.2.2 Main Results

Result on the KITTI *test* set. As shown in Table 5.1, we report our results of **Car** category on KITTI *test* set. The proposed method obtains **4.58/2.62/2.36** improvements in AP_{3D} at IoU threshold 0.7, surpassing the best method GUPNet[195] which has no extra data assisted. This indicates the limitations of the purely image-based methods and strengthens the importance of depth cues. More importantly, compared with the methods with extra depth data, such as DD3D, our proposed method still gets comparable performance. DD3D (current SOTA) takes advantage of extra 15M additional proprietary depth data, while our approach achieves on-par performance with only much smaller amount of in-domain KITTI-Depth data.

MonoDTR[122] also applies transformers to fuse the depth features with additional depth supervision to benefit the detection performance. However, our Mono3DCFT outperforms it on all AP_{3D} metrics (with **2.70/1.43/1.40** improvements on Easy/Mod./Hard), by introducing a better feature fusion encoder with contextual depth priors inherently from the coarse-stage training, and several careful designs on object-wise depth in the decoder for the fine-grained training. Overall, our Mono3DCFT achieves superior results over previous methods across all settings under fair conditions and demonstrates its simplicity and effectiveness for monocular 3D object detection. The performance of the Pedestrian and Cyclists categories on the KITTI test set at 0.5 IoU threshold is also provided in the supplementary material.

Result on the KITTI *val* set. We conduct experiments of **Car** category on KITTI *val* set, also as listed in Table 5.1. Our approach achieves better performance over several image-only and depth-assisted methods. Specifically, our method outperforms the transformer-based MonoDTR[122] by a huge gap with **5.11/3.44/1.45** improvements on AP_{3D} at IoU threshold 0.7. Also, note that our

method shows better performance consistency between the validation set and the test set, when compared to DD3D[231]. This indicates that our method has better generalization ability, which is of great significance in autonomous/assisted driving.

Qualitative Results. We provide the qualitative results on the KITTI validation set in Figure 5.4. To clearly show the object's position in the 3D world space, we also visualize the LiDAR point clouds. It can be observed that our model produces remarkably accurate 3D bounding boxes for the cases at a reasonable distance. More qualitative results are included in the supplementary material.

5.3 Ablation Study

Effectiveness of the 1*st* stage Coarse-grained Training. In Table 5.2, we conduct an ablation study to analyze the effectiveness of the 1*st* stage coarse-grained training on KITTI *val* set. We fix the model to be trained with $\mathcal{L}_{Dobj}$ for the 2*nd* stage fine-grained training since the losses make more sense when dealing with region-level attributes. The first row indicates the model without the 1*st*-stage training for the encoder and directly performs the monocular 3D object detection using the whole Mono3DCFT architecture, which has already achieved comparable performance with MonoDTR[122]. Furthermore, if the fusion encoder is pre-trained under the 1*st* stage but is only supervised by foreground object-wise depth, the detection performance is better than the model without 1*st* stage training. The best result is obtained by using $\mathcal{L}_{Dmap}$ during the 1*st* stage training. This demonstrates that the depth cues are much better learned and become more contextual when training with global-level depth information, which also provides a good initialization for the latter fine-grained 3D detection. In addition, we also provide the final column for depth (z_{3D}) root-mean-squared-error (RMSE). It can be seen from Table 5.2, depth estimation can well-benefit from our proposed training method.

Fusion-in-the-backbone Encoder. In Table 5.3, we first explore using different attention mechanisms for the proposed depth encoder. By comparing the first row and second row, using the vanilla self-attention achieves slightly lower performance than the channel + spatial attention. The main reason of using channel + spatial attention instead of self-attention in our depth encoder is due to its computational efficiency and fewer epochs needed for 1*st* stage training. It is also worthy of

$w.$ 1*st* stage	$\mathcal{L}_{Dmap}$/$\mathcal{L}_{Dobj}$	AP_{3D}@IoU=0.7			Depth RMSE
		Easy	Mod.	Hard	
×	-	25.16	18.53	14.46	1.41
✓	$\mathcal{L}_{obj}$	28.83	20.49	15.07	1.33
✓	$\mathcal{L}_{map}$	**29.63**	**22.01**	**16.96**	**1.09**

Table 5.2: **Effectiveness of the 1*st* stage coarse-grained training**. $\mathcal{L}_{Dmap}$/$\mathcal{L}_{Dobj}$ indicates whether the encoder is trained on the whole depth map or foreground object-wise depth.

Settings		AP_{3D}@IoU=0.7		
Attn. Mechanism	Gating α	Easy	Mod.	Hard
Vanilla Self-Attn.	Single	27.18	20.86	18.14
Channel + Spatial Attn.	Single	27.85	20.78	18.31
Channel + Spatial Attn.	Dual	**29.63**	**22.01**	**16.96**

Table 5.3: **Fusion-in-the-backbone Encoder.** We explore different attention mechanisms and switch on/off gating cross-attention.

mentioning that this attention mechanism is reasonable for multi-scale features to capture global depth information. Besides, during the 2*nd* stage training, we also try to use different interactions in gating cross-attention fusion blocks. Switching from a single interaction (v2d) to dual interactions (v2d & d2v) gives us more performance gain (**1.23**↑ in Mod. between Row 2 and Row 3), which shows that both depth and visual information are important and need to be jointly incorporated to achieve higher performance.

Depth-guided Decoder. We investigate the depth-guided decoder in two aspects, (i) the positional encodings, and (ii) the depth deNoising queries (shown in Table 5.4). Compared with many commonly used positional encodings, including depth positional encodings (DPE)[381], absolute positional encoding (APE)[64], sinusoidal positional encoding[306], and without positional encoding (No PE), Mono3DCFT can achieve the best performance with DPE, echoing the findings in

Settings		AP_{3D}@IoU=0.7		
DeNoise Queries	Positional Enc.	Easy	Mod.	Hard
×	No PE	28.90	21.21	16.70
×	Sinusoidal[306]	28.69	21.10	16.53
×	APE[64]	29.02	21.15	16.74
×	DPE[381]	29.47	21.42	16.83
✓	Sinusoidal[306]	29.45	21.33	16.71
✓	DPE	**29.63**	**22.01**	**16.96**

Table 5.4: **Depth-guided Decoder.** We compare difference position encoding mechanisms and add depth denoising queries.

previous literatures[122, 381]. Interestingly, the model without the positional encoding outperforms the one with sinusoidal encoding. We believe that encoding the depth-aware cues is more effective for learning the positional representations of 3D tasks than pixel-level encodings.

Furthermore, in Table 5.4, we add the depth denoising queries to further boost the detection performance, especially, a **0.59** improvement on Mod. and a **0.13** improvement on Hard. This shows that adding depth deNoising queries indeed helps the object queries to learn more accurate depth to the object of focus. This is the first time that this kind of denoising approach has been used for 3D cases.

Bipartite Matching. We explore different combinations of discrete loss as the matching cost for each query-label pair. As reported in Table 5.5, besides L_{2D} that is commonly used in [24] to predict the 2D bounding boxes and object categories, we append more 3D losses into the matching cost. We find the best solution is to utilize both L_{2D} and L_{3D} into the matching cost. This is also contrary to the findings from [381], where they also append 3D losses into the matching cost but lead to ill-posed problems and even cause training failure. We claim the training stability is improved by two takeaways: a) the 1st-stage training gives better depth priors for 3D monocular object detection. 2) adding the depth denoising queries makes the training consistency optimization goals for decoder

Matching Cost	AP_{3D}@IoU=0.7		
	Easy	Mod.	Hard
$\mathcal{L}_{2D}$	27.98	21.16	15.91
$\mathcal{L}_{2D}$ $w.$ $\mathcal{L}_{ori}$	26.23	19.60	15.74
$\mathcal{L}_{2D}$ $w.$ $\mathcal{L}_{loc}$	28.44	21.58	16.62
$\mathcal{L}_{2D}$ $w.$ $\mathcal{L}_{dim}$	27.12	21.04	16.70
$\mathcal{L}_{2D}$ $w.$ $\mathcal{L}_{3D}$	**29.63**	**22.01**	**16.96**

Table 5.5: **Bipartite Matching.** We set different losses $\mathcal{L}_{2D}$ and $\mathcal{L}_{3D}$ as the matching cost of each query-label pair.

queries.

5.4 Summary

In this chapter, we propose an innovative transformer-based monocular 3D object detection framework with an effective two-stage training mechanism. The proposed framework can capture better scene-level depth cues through the 1st stage of coarse-grained training, which can later be adapted to region-level attributes understanding in the 2nd stage of fine-grained training. Besides, the proposed fusion-in-the-backbone encoder and depth-guided decoder are well-designed for parameter sharing and avoid inaccurate depth priors. Extensive experiments and analyses on the KITTI dataset have demonstrated the effectiveness of our approach. Future work will investigate scaling our transformer-based model with extra depth data to learn even better depth representations, as well as integrating the framework with multi-object tracking methods for 3D autonomous driving applications.

Chapter 6

CONCLUSIONS

Detecting safety-critical objects in urban scenes using a single monocular camera in both 2D and 3D space is extremely challenging due to scale variability, occlusion, and depth ambiguity. The need to detect such objects in an accurate and efficient manner is growing in importance as urban autonomous driving applications and technology continue to develop rapidly. Specifically, monocular camera-based object detection methodology remains the most cost-effective and is therefore widespread among the available sensors (LiDAR, Radar, and stereo cameras).

Throughout this dissertation, we have proposed and demonstrated the effectiveness of various monocular object detection techniques while keeping both accuracy and efficiency in mind. We apply this goal firstly to LOCNet (Ch. 3) and propose a monocular vision-based autonomous driving framework to perform 3D detection, tracking, and localization by effectively integrating all three tasks in a complementary manner. Our LOCNet and FES-based single frame optimization provide accurate localization results, which are further refined with the help of the 3D TrackletNet Tracker to eventually achieve performance comparable to LiDAR-based localization methods. Moreover, in Ch. 4, we propose a *joint* framework (JMV3D) that can effectively associate moving objects over time and estimate their 3D localization information as well as segmentation masks from a sequence of 2D images so as to compensate for the individual drawbacks of each component. We further extend the existing Localization Network (LOCNet) to become Localization for Tracking Network (Loc4Trk-Net). spatial Attention (SA) Neck is added to highlight the foreground (target of interest) and suppress the background with the help of mask segmentation so that more concentrated appearance features can be obtained. Besides, one additional embedding head is introduced to train discriminative feature embeddings to leverage deep pairwise contrastive learning and identify objects in various poses and viewpoints with appearance cues. In Ch. 5, we further address the

problem for monocular 3D object detection by neglecting image-level understanding of depth and semantics. To address this, we present the Mono3DCFT to use transformer-based architecture with an effective two-stage training strategy. The proposed framework can capture better scene-level depth cues through the 1st stage of coarse-grained training, which can later be adapted to region-level attributes understanding in the 2nd stage of fine-grained training. Besides, the proposed fusion-in-the-backbone encoder and depth-guided decoder are well-designed for parameter sharing and avoid inaccurate depth priors.

As a whole, extensive experiments and ablations have demonstrated the effectiveness of all the approaches in this thesis. Future work will investigate scaling our transformer-based model with extra depth data to learn even better depth representations, as well as integrating the framework with multi-object tracking methods for 3D autonomous driving applications.

BIBLIOGRAPHY

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018.

[2] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. `http://ceres-solver.org`.

[3] Shubhra Aich, Jean Marie Uwabeza Vianney, Md Amirul Islam, and Mannat Kaur Bingbing Liu. Bidirectional attention network for monocular depth estimation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11746–11752. IEEE, 2021.

[4] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018.

[5] Junaid Ahmed Ansari, Sarthak Sharma, Anshuman Majumdar, J Krishna Murthy, and K Madhava Krishna. The earth ain't flat: Monocular reconstruction of vehicles on steep and graded roads from a moving camera. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8404–8410. IEEE, 2018.

[6] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[7] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. *arXiv preprint arXiv:2202.03555*, 2022.

[8] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[9] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[10] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008.

[11] Gedas Bertasius and Lorenzo Torresani. Classifying, segmenting, and tracking object instances in video with mask propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9739–9748, 2020.

[12] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding. *arXiv preprint arXiv:2102.05095*, 2(3):4, 2021.

[13] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017.

[14] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pages 446–461. Springer, 2014.

[15] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9287–9296, 2019.

[16] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *IEEE International Conference on Computer Vision*, 2019.

[17] Garrick Brazil, Gerard Pons-Moll, Xiaoming Liu, and Bernt Schiele. Kinematic 3d object detection in monocular video. In *Proceedings of the European Conference on Computer Vision*, 2020.

[18] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[19] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *CoRR*, abs/1903.11027, 2019.

[20] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[21] Jiarui Cai, Yizhou Wang, Hung-Min Hsu, Haotian Zhang, and Jenq-Neng Hwang. Dior: Distill observations to representations for multi-object tracking and segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 520–529, 2022.

[22] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

[23] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1611.08050*, 2016.

[24] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision*, 2020.

[25] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[26] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[27] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[28] Florian Chabot, Mohamed Chaouch, Jaonary Rabarisoa, Céline Teuliere, and Thierry Chateau. Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2040–2049, 2017.

[29] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8748–8757, 2019.

[30] Hansheng Chen, Yuyao Huang, Wei Tian, Zhong Gao, and Lu Xiong. Monorun: Monocular 3d object detection by reconstruction and uncertainty propagation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[31] Jiahui Chen, Hao Sheng, Yang Zhang, and Zhang Xiong. Enhancing detection model for multiple hypothesis tracking. In *Conf. on Computer Vision and Pattern Recognition Workshops*, pages 2143–2152, 2017.

[32] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[33] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig 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.

[34] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[35] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

[36] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *arXiv preprint arXiv:2202.03026*, 2022.

[37] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[38] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew G Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals for accurate object class detection. In *Advances in Neural Information Processing Systems*, pages 424–432, 2015.

[39] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew G. Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals for accurate object class detection. In *Conference on Neural Information Processing Systems*, 2015.

[40] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[41] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021.

[42] Yinbo Chen, Xiaolong Wang, Zhuang Liu, Huijuan Xu, and Trevor Darrell. A new meta-baseline for few-shot learning. *arXiv preprint arXiv:2003.04390*, 2020.

[43] Yongjian Chen, Lei Tai, Kai Sun, and Mingyang Li. Monopair: Monocular 3d object detection using pairwise spatial relationships. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[44] Hsu-kuang Chiu, Antonio Prioletti, Jie Li, and Jeannette Bohg. Probabilistic 3d multi-object tracking for autonomous driving. *arXiv preprint arXiv:2001.05673*, 2020.

[45] Wongun Choi. Near-online multi-target tracking with aggregated local flow descriptor. In *Proceedings of the IEEE international conference on computer vision*, pages 3029–3037, 2015.

[46] Qi Chu, Wanli Ouyang, Hongsheng Li, Xiaogang Wang, Bin Liu, and Nenghai Yu. Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism. In *2017 IEEE International Conference on Computer Vision (ICCV).(Oct 2017)*, pages 4846–4855, 2017.

[47] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021.

[48] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014.

[49] MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. `https://github.com/open-mmlab/mmdetection3d`, 2020.

[50] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[51] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *IEEE International Conference on Computer Vision*, 2017.

[52] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1601–1610, 2021.

[53] Zihang Dai, Hanxiao Liu, Quoc Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems*, 34, 2021.

[54] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[55] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[56] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[57] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11162–11173, 2021.

[58] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[59] Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. *arXiv preprint arXiv:1909.02729*, 2019.

[60] Mingyu Ding, Yuqi Huo, Hongwei Yi, Zhe Wang, Jianping Shi, Zhiwu Lu, and Ping Luo. Learning depth-guided convolutions for monocular 3d object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[61] Xiaohan Ding, Xiangyu Zhang, Yizhuang Zhou, Jungong Han, Guiguang Ding, and Jian Sun. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. *arXiv preprint arXiv:2203.06717*, 2022.

[62] Xiaoyi Dong, Jianmin Bao, Ting Zhang, Dongdong Chen, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. Peco: Perceptual codebook for bert pre-training of vision transformers. *arXiv preprint arXiv:2111.12710*, 2021.

[63] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[64] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[65] Zi-Yi Dou, Aishwarya Kamath, Zhe Gan, Pengchuan Zhang, Jianfeng Wang, Linjie Li, Zicheng Liu, Ce Liu, Yann LeCun, Nanyun Peng, et al. Coarse-to-fine vision-language pre-training with fusion in the backbone. *arXiv preprint arXiv:2206.07643*, 2022.

[66] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6569–6578, 2019.

[67] Stéphane d'Ascoli, Hugo Touvron, Matthew L Leavitt, Ari S Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *International Conference on Machine Learning*, pages 2286–2296. PMLR, 2021.

[68] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*, 2014.

[69] Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc Van Gool. A mobile vision system for robust multi-person tracking. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[70] Kuan Fang, Yu Xiang, Xiaocheng Li, and Silvio Savarese. Recurrent autoregressive networks for online multi-object tracking. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 466–475. IEEE, 2018.

[71] Yuxin Fang, Shusheng Yang, Shijie Wang, Yixiao Ge, Ying Shan, and Xinggang Wang. Unleashing vanilla vision transformer with masked image modeling for object detection. *arXiv preprint arXiv:2204.02964*, 2022.

[72] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004.

[73] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to track and track to detect. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3038–3046, 2017.

[74] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.

[75] Jianfeng Feng, Errui Ding, and Shilei Wen. Monocular 3d object detection via feature domain adaptation.

[76] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[77] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[78] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018.

[79] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[80] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[81] Zeyu Fu, Pengming Feng, Federico Angelini, Jonathon Chambers, and Syed Mohsen Naqvi. Particle phd filter based multiple human tracking using online group-structured dictionary learning. *IEEE Access*, 6:14764–14778, 2018.

[82] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*, 2021.

[83] Peng Gao, Zhengkai Jiang, Haoxuan You, Pan Lu, Steven CH Hoi, Xiaogang Wang, and Hongsheng Li. Dynamic fusion with intra-and inter-modality attention flow for visual question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6639–6648, 2019.

[84] Peng Gao, Jiasen Lu, Hongsheng Li, Roozbeh Mottaghi, and Aniruddha Kembhavi. Container: Context aggregation network. *arXiv preprint arXiv:2106.01401*, 2021.

[85] Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fast convergence of detr with spatially modulated co-attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3621–3630, 2021.

[86] Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fast convergence of detr with spatially modulated co-attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3621–3630, October 2021.

[87] Ravi Garg, Vijay Kumar BG, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *Proceedings of the European Conference on Computer Vision*, 2016.

[88] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *International Conference on Machine Learning*, pages 1243–1252. PMLR, 2017.

[89] Andreas Geiger, Martin Lauer, Christian Wojek, Christoph Stiller, and Raquel Urtasun. 3d traffic scene understanding from movable platforms. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):1012–1025, 2014.

[90] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012.

[91] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[92] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Dropblock: A regularization method for convolutional networks. *Advances in neural information processing systems*, 31, 2018.

[93] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[94] Ross Girshick. Fast r-cnn. In *IEEE International Conference on Computer Vision*, 2015.

[95] Clement Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with leftright consistency. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[96] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9224–9232, 2018.

[97] Benjamin Graham and Laurens van der Maaten. Submanifold sparse convolutional networks. *arXiv preprint arXiv:1706.01307*, 2017.

[98] Edouard Grave, Moustapha Cissé, and Armand Joulin. Unbounded cache model for online language modeling with open vocabulary. *arXiv preprint arXiv:1711.02604*, 2017.

[99] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

[100] Jianyuan Guo, Kai Han, Han Wu, Chang Xu, Yehui Tang, Chunjing Xu, and Yunhe Wang. Cmt: Convolutional neural networks meet vision transformers. *arXiv preprint arXiv:2107.06263*, 2021.

[101] Xiaoyang Guo, Hongsheng Li, Shuai Yi, Jimmy Ren, and Xiaogang Wang. Learning monocular depth by distilling cross-domain stereo networks. In *Proceedings of the European Conference on Computer Vision*, 2018.

[102] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *Advances in Neural Information Processing Systems*, 34, 2021.

[103] Andrew J Hanson. Visualizing quaternions. In *ACM SIGGRAPH 2005 Courses*, pages 1–es. 2005.

[104] Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3018–3027, 2017.

[105] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[106] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

[107] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[108] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[109] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[110] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[111] Tong He and Stefano Soatto. Mono3d++: Monocular 3d vehicle detection with two-scale 3d hypotheses and task priors. In *AAAI Conference on Artificial Intelligence*, 2018.

[112] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

[113] Roberto Henschel, Laura Leal-Taixé, Daniel Cremers, and Bodo Rosenhahn. Fusion of head and full-body detectors for multi-object tracking. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018.

[114] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[115] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016.

[116] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Putting objects in perspective. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[117] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *ICML*, 2019.

[118] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[119] Hou-Ning Hu, Qi-Zhi Cai, Dequan Wang, Ji Lin, Min Sun, Philipp Krähenbühl, Trevor Darrell, and Fisher Yu. Joint monocular 3d vehicle detection and tracking. 2019.

[120] Hou-Ning Hu, Yung-Hsu Yang, Tobias Fischer, Trevor Darrell, Fisher Yu, and Min Sun. Monocular quasi-dense 3d object tracking. *arXiv preprint arXiv:2103.07351*, 2021.

[121] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[122] Kuan-Chih Huang, Tsung-Han Wu, Hung-Ting Su, and Winston H Hsu. Monodtr: Monocular 3d object detection with depth-aware transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4012–4021, 2022.

[123] Lichao Huang, Yi Yang, Yafeng Deng, and Yinan Yu. Densebox: Unifying landmark localization with end to end object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[124] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4634–4643, 2019.

[125] Tony Huang, Jack Chu, and Fangyun Wei. Unsupervised prompt learning for vision-language models. *arXiv preprint arXiv:2204.03649*, 2022.

[126] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017.

[127] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5001–5009, 2018.

[128] Ajay Jain, Pieter Abbeel, and Deepak Pathak. Locally masked convolution for autoregressive models. In *Conference on Uncertainty in Artificial Intelligence*, pages 1358–1367. PMLR, 2020.

[129] Muhammad Abdullah Jamal and Guo-Jun Qi. Task agnostic meta-learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11719–11727, 2019.

[130] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021.

[131] Xiaohu Jiang, Ze Chen, Zhicheng Wang, Erjin Zhou, et al. Guiding query position and performing similar attention for transformer-based detection heads. *arXiv preprint arXiv:2108.09691*, 2021.

[132] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2019.

[133] Eskil Jörgensen, Christopher Zach, and Fredrik Kahl. Monocular 3d object detection and box fitting trained end-to-end using intersection-over-union loss. *CoRR*, abs/1906.08070, 2019.

[134] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1409–1422, 2011.

[135] Kai Kang, Hongsheng Li, Junjie Yan, Xingyu Zeng, Bin Yang, Tong Xiao, Cong Zhang, Zhe Wang, Ruohui Wang, Xiaogang Wang, et al. T-cnn: Tubelets with convolutional neural networks for object detection from videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

[136] Kai Kang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Object detection from video tubelets with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 817–825, 2016.

[137] Alex Kendall. *Geometry and Uncertainty in Deep Learning for Computer Vision*. PhD thesis, University of Cambridge, 2018.

[138] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[139] Margret Keuper, Siyu Tang, Bjorn Andres, Thomas Brox, and Bernt Schiele. Motion segmentation & multiple object tracking by correlation co-clustering. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[140] Margret Keuper, Siyu Tang, Yu Zhongjie, Bjoern Andres, Thomas Brox, and Bernt Schiele. A multi-cut formulation for joint segmentation and tracking of multiple objects. *arXiv preprint arXiv:1607.06317*, 2016.

[141] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172*, 2019.

[142] Hilke Kieritz, Stefan Becker, Wolfgang Hübner, and Michael Arens. Online multi-person tracking using integral channel features. In *Advanced Video and Signal Based Surveillance (AVSS), 2016 13th IEEE International Conference on*, pages 122–130. IEEE, 2016.

[143] Bumsoo Kim, Junhyun Lee, Jaewoo Kang, Eun-Sol Kim, and Hyunwoo J Kim. Hotr: End-to-end human-object interaction detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 74–83, 2021.

[144] Chanho Kim, Fuxin Li, Arridhana Ciptadi, and James M Rehg. Multiple hypothesis tracking revisited. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4696–4704, 2015.

[145] Chanho Kim, Fuxin Li, and James M Rehg. Multi-object tracking with neural gating using bilinear lstm. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 200–215, 2018.

[146] Doyeon Kim, Woonghyun Ga, Pyungwhan Ahn, Donggyu Joo, Sehwan Chun, and Junmo Kim. Global-local path networks for monocular depth estimation with vertical cutdepth. *arXiv preprint arXiv:2201.07436*, 2022.

[147] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. *arXiv preprint arXiv:1805.07932*, 2018.

[148] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[149] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. *arXiv preprint arXiv:1912.08193*, 2019.

[150] Jannik Kossen, Neil Band, Clare Lyle, Aidan N Gomez, Tom Rainforth, and Yarin Gal. Self-attention between datapoints: Going beyond individual input-output pairs in deep learning. *arXiv preprint arXiv:2106.02584*, 2021.

[151] Wouter M Kouw and Marco Loog. An introduction to domain adaptation and transfer learning. *arXiv preprint arXiv:1812.11806*, 2018.

[152] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[153] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[154] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven Waslander. Joint 3d proposal generation and object detection from view aggregation. In *International Conference on Intelligent Robots and Systems*, 2018.

[155] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[156] Ratnesh Kumar, Guillaume Charpiat, and Monique Thonnat. Multiple object tracking by efficient graph partitioning. In *Asian Conference on Computer Vision*, pages 445–460. Springer, 2014.

[157] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[158] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *European Conference on Computer Vision*, 2018.

[159] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[160] Aoxue Li, Weiran Huang, Xu Lan, Jiashi Feng, Zhenguo Li, and Liwei Wang. Boosting few-shot learning with adaptive margin loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12576–12584, 2020.

[161] Bohao Li, Boyu Yang, Chang Liu, Feng Liu, Rongrong Ji, and Qixiang Ye. Beyond max-margin: Class margin equilibrium for few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7363–7372, 2021.

[162] Buyu Li, Wanli Ouyang, Lu Sheng, Xingyu Zeng, and Xiaogang Wang. Gs3d: An efficient 3d object detection framework for autonomous driving. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[163] Chunyuan Li, Jianwei Yang, Pengchuan Zhang, Mei Gao, Bin Xiao, Xiyang Dai, Lu Yuan, and Jianfeng Gao. Efficient self-supervised vision transformers for representation learning. *arXiv preprint arXiv:2106.09785*, 2021.

[164] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13619–13627, 2022.

[165] Kunchang Li, Yali Wang, Junhao Zhang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unifying convolution and self-attention for visual recognition. *arXiv preprint arXiv:2201.09450*, 2022.

[166] Liunian Harold* Li, Pengchuan* Zhang, Haotian* Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022.

[167] Peiliang Li, Tong Qin, et al. Stereo vision-based semantic 3d object and ego-motion tracking for autonomous driving. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 646–661, 2018.

[168] Peixuan Li, Huaici Zhao, Pengfei Liu, and Feidao Cao. Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving. In *European Conference on Computer Vision*, 2020.

[169] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. *arXiv preprint arXiv:2110.05208*, 2021.

[170] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. *arXiv preprint arXiv:2203.16527*, 2022.

[171] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Improved multiscale vision transformers for classification and detection. *arXiv preprint arXiv:2112.01526*, 2021.

[172] Yanghao Li, Saining Xie, Xinlei Chen, Piotr Dollar, Kaiming He, and Ross Girshick. Benchmarking detection transfer learning with vision transformers. *arXiv preprint arXiv:2111.11429*, 2021.

[173] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *Advances in neural information processing systems*, pages 820–830, 2018.

[174] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7083–7093, 2019.

[175] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[176] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[177] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[178] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[179] Bin Liu, Yue Cao, Yutong Lin, Qi Li, Zheng Zhang, Mingsheng Long, and Han Hu. Negative margin matters: Understanding margin in few-shot classification. In *European Conference on Computer Vision*, pages 438–455. Springer, 2020.

[180] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European conference on computer vision (ECCV)*, pages 85–100, 2018.

[181] Lijie Liu, Jiwen Lu, Chunjing Xu, Qi Tian, and Jie Zhou. Deep fitting degree scoring network for monocular 3d object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[182] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021.

[183] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. *arXiv preprint arXiv:2201.12329*, 2022.

[184] Songtao Liu, Di Huang, and Yunhong Wang. Adaptive nms: Refining pedestrian detection in a crowd. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6459–6468, 2019.

[185] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[186] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision*, 2016.

[187] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In

Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 10012–10022, 2021.

[188] Zechen Liu, Zizhang Wu, and Roland Tóth. SMOKE: single-stage monocular 3d object detection via keypoint estimation. *CoRR*, abs/2002.10111, 2020.

[189] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*, 2022.

[190] Zongdai Liu, Dingfu Zhou, Feixiang Lu, Jin Fang, and Liangjun Zhang. Autoshape: Real-time shape-aware monocular 3d object detection. *CoRR*, abs/2108.11127, 2021.

[191] C Long, A Haizhou, Z Zijie, and S Chong. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. ICME, 2018.

[192] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[193] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

[194] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*, 2019.

[195] Yan Lu, Xinzhu Ma, Lei Yang, Tianzhu Zhang, Yating Liu, Qi Chu, Junjie Yan, and Wanli Ouyang. Geometry uncertainty projection network for monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3111–3121, October 2021.

[196] Yongyi Lu, Cewu Lu, and Chi-Keung Tang. Online video object detection using association lstm. In *Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy*, pages 22–29, 2017.

[197] Jonathon Luiten, Tobias Fischer, and Bastian Leibe. Track to reconstruct and reconstruct to track. *IEEE Robotics and Automation Letters*, 5(2):1803–1810, 2020.

[198] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision*, pages 1–31, 2020.

[199] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision*, pages 1–31, 2020.

[200] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. In *Asian Conference on Computer Vision*, pages 565–580. Springer, 2018.

[201] Shujie Luo, Hang Dai, Ling Shao, and Yong Ding. M3dssd: Monocular 3d single stage object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6145–6154, 2021.

[202] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[203] Siwei Lyu, Ming-Ching Chang, Dawei Du, Wenbo Li, Yi Wei, Marco Del Coco, Pierluigi Carcagnì, Arne Schumann, Bharti Munjal, Doo-Hyun Choi, et al. Ua-detrac 2018: Report of avss2018 & iwt4s challenge on advanced traffic monitoring. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2018.

[204] Cong Ma, Changshui Yang, Fan Yang, Yueqing Zhuang, Ziwei Zhang, Huizhu Jia, and Xiaodong Xie. Trajectory factory: Tracklet cleaving and re-connection by deep siamese bi-gru for multiple object tracking. *arXiv preprint arXiv:1804.04555*, 2018.

[205] Xinzhu Ma, Shinan Liu, Zhiyi Xia, Hongwen Zhang, Xingyu Zeng, and Wanli Ouyang. Rethinking pseudo-lidar representation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[206] Xinzhu Ma, Shinan Liu, Zhiyi Xia, Hongwen Zhang, Xingyu Zeng, and Wanli Ouyang. Rethinking pseudo-lidar representation. In *European Conference on Computer Vision*, pages 311–327. Springer, 2020.

[207] Xinzhu Ma, Zhihui Wang, Haojie Li, Pengbo Zhang, Wanli Ouyang, and Xin Fan. Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6851–6860, 2019.

[208] Xinzhu Ma, Yinmin Zhang, Dan Xu, Dongzhan Zhou, Shuai Yi, Haojie Li, and Wanli Ouyang. Delving into localization errors for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4721–4730, June 2021.

[209] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017.

[210] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[211] Fabian Manhardt, Wadim Kehl, and Adrien Gaidon. Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[212] Mingyuan Mao, Renrui Zhang, Honghui Zheng, Peng Gao, Teli Ma, Yan Peng, Errui Ding, and Shumin Han. Dual-stream network for visual recognition. *arXiv preprint arXiv:2105.14734*, 2021.

[213] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3651–3660, 2021.

[214] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional DETR for fast training convergence. *CoRR*, abs/2108.06152, 2021.

[215] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

[216] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.

[217] Anton Milan, Seyed Hamid Rezatofighi, Anthony R Dick, Ian D Reid, and Konrad Schindler. Online multi-target tracking using recurrent neural networks. In *AAAI*, volume 2, page 4, 2017.

[218] Anton Milan, Konrad Schindler, and Stefan Roth. Multi-target tracking by discrete-continuous energy minimization. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2054–2068, 2016.

[219] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2906–2917, 2021.

[220] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7074–7082, 2017.

[221] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[222] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[223] J Krishna Murthy, Sarthak Sharma, and K Madhava Krishna. Shape priors for real-time monocular object localization in dynamic environments. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1768–1774. IEEE, 2017.

[224] Ramin Nabati and Hairong Qi. Centerfusion: Center-based radar and camera fusion for 3d object detection. In *IEEE Winter Conference on Applications of Computer Vision*, 2020.

[225] Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. In *International Conference on Learning Representations*, 2020.

[226] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

[227] A Emin Orhan. A simple cache model for image recognition. *arXiv preprint arXiv:1805.08709*, 2018.

[228] Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. Scaling neural machine translation. *arXiv preprint arXiv:1806.00187*, 2018.

[229] Su Pang, Daniel Morris, and Hayder Radha. Clocs: Camera-lidar object candidates fusion for 3d object detection. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10386–10393. IEEE, 2020.

[230] Christos H Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *Journal of computer and system sciences*, 43(3):425–440, 1991.

[231] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3142–3152, 2021.

[232] Dennis Park, Jie Li, Dian Chen, Vitor Guizilini, and Adrien Gaidon. Depth is all you need for monocular 3d detection. *arXiv preprint arXiv:2210.02493*, 2022.

[233] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012.

[234] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[235] Vlad Paunescu, Andretti Naiden, Gyeongmo Kim, ByeongMoon Jeon, and Marius Leordeanu. Shift r-cnn: Deep monocular 3d object detection with closed-form geometric constraints. 2019.

[236] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters–improve semantic segmentation by global convolutional network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4353–4361, 2017.

[237] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 918–927, 2018.

[238] Charles R. Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J. Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[239] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[240] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[241] Rui Qian, Divyansh Garg, Yan Wang, Yurong You, Serge Belongie, Bharath Hariharan, Mark Campbell, Kilian Q Weinberger, and Wei-Lun Chao. End-to-end pseudo-lidar for image-based 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5881–5890, 2020.

[242] Siyuan Qiao, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Vip-deeplab: Learning visual perception with depth-aware video panoptic segmentation. *arXiv preprint arXiv:2012.05258*, 2020.

[243] Zengyi Qin, Jinglu Wang, and Yan Lu. Monogrnet: A geometric reasoning network for monocular 3d object localization. In *AAAI Conference on Artificial Intelligence*, 2019.

[244] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[245] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[246] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[247] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[248] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

[249] Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. Categorical depth distribution network for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8555–8564, 2021.

[250] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[251] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[252] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[253] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[254] Mengye Ren, Andrei Pokrovsky, Bin Yang, and Raquel Urtasun. Sbnet: Sparse blocks network for fast inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8711–8720, 2018.

[255] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[256] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[257] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019.

[258] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision*, pages 17–35. Springer, 2016.

[259] Ergys Ristani and Carlo Tomasi. Features for multi-target multi-camera tracking and re-identification. *arXiv preprint arXiv:1803.10859*, 2018.

[260] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic feature transform for monocular 3d object detection. *arXiv preprint arXiv:1811.08188*, 2018.

[261] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic feature transform for monocular 3d object detection. *CoRR*, abs/1811.08188, 2018.

[262] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.

[263] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[264] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. *arXiv preprint arXiv:1701.01909*, 4(5):6, 2017.

[265] Amir Sadeghian, Khashayar Khosravi, and Alexandre Robicquet. End-to-end learning of motion, appearance and interaction cues for multi-target tracking.

[266] Ricardo Sanchez-Matilla, Fabio Poiesi, and Andrea Cavallaro. Online multi-target tracking with strong and weak detections. In *European Conference on Computer Vision*, pages 84–99. Springer, 2016.

[267] 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.

[268] Ashutosh Saxena, Sung Chung, and Andrew Ng. Learning depth from single monocular images. In *Advances in Neural Information Processing Systems*, 2005.

[269] Samuel Scheidegger, Joachim Benjaminsson, Emil Rosenberg, Amrit Krishnan, and Karl Granström. Mono-camera 3d multi-object tracking using deep learning detections and pmbm filtering. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 433–440. IEEE, 2018.

[270] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[271] Sarthak Sharma, Junaid Ahmed Ansari, J Krishna Murthy, and K Madhava Krishna. Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3508–3515. IEEE, 2018.

[272] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–779, 2019.

[273] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[274] Shaoshuai Shi, Zhe Wang, Xiaogang Wang, and Hongsheng Li. Part-aˆ 2 net: 3d part-aware and aggregation neural network for object detection from point cloud. *arXiv preprint arXiv:1907.03670*, 2019.

[275] Xuepeng Shi, Qi Ye, Xiaozhi Chen, Chuangrong Chen, Zhixiang Chen, and Tae-Kyun Kim. Geometry-based distance decomposition for monocular 3d object detection. In *IEEE International Conference on Computer Vision*, 2021.

[276] Andrea Simonelli, Samuel Rota Bulò, Lorenzo Porzi, Peter Kontschieder, and Elisa Ricci. Demystifying pseudo-lidar for monocular 3d object detection. *ArXiv*, abs/2012.05796, 2020.

[277] Andrea Simonelli, Samuel Rota Bulò, Lorenzo Porzi, Elisa Ricci, and Peter Kontschieder. Towards generalization across depth for monocular 3d object detection. In *Proceedings of the European Conference on Computer Vision*, 2020.

[278] Andrea Simonelli, Samuel Rota Rota Bulò, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. In *IEEE International Conference on Computer Vision*, 2019.

[279] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[280] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

[281] Xibin Song, Wei Li, Dingfu Zhou, Yuchao Dai, Jin Fang, Hongdong Li, and Liangjun Zhang. Mlda-net: Multi-level dual attention-based network for self-supervised monocular depth estimation. *IEEE Transactions on Image Processing*, 30:4691–4705, 2021.

[282] Xibin Song, Peng Wang, Dingfu Zhou, Rui Zhu, Chenye Guan, Yuchao Dai, Hao Su, Hongdong Li, and Ruigang Yang. Apollocar3d: A large 3d car instance understanding benchmark for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5452–5462, 2019.

[283] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[284] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16519–16529, 2021.

[285] Edgar Sucar and Jean-Bernard Hayet. Bayesian scale estimation for monocular slam based on generic object detection for correcting scale drift. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7. IEEE, 2018.

[286] Sainbayar Sukhbaatar, Edouard Grave, Guillaume Lample, Herve Jegou, and Armand Joulin. Augmenting self-attention with persistent memory. *arXiv preprint arXiv:1907.01470*, 2019.

[287] Jianyong Sun, Qingfu Zhang, and Edward PK Tsang. De/eda: A new evolutionary algorithm for global optimization. *Information Sciences*, 169(3-4):249–262, 2005.

[288] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020.

[289] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14454–14463, 2021.

[290] Zhiqing Sun, Shengcao Cao, Yiming Yang, and Kris M Kitani. Rethinking transformer-based set prediction for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3611–3620, 2021.

[291] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018.

[292] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019.

[293] Siyu Tang, Bjoern Andres, Miykhaylo Andriluka, and Bernt Schiele. Subgraph decomposition for multi-target tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5033–5041, 2015.

[294] Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Multi-person tracking by multicut and deep matching. In *European Conference on Computer Vision*, pages 100–111. Springer, 2016.

[295] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple people tracking by lifted multicut and person reidentification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3539–3548, 2017.

[296] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David Anastasiu, and Jenq-Neng Hwang. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8797–8806, 2019.

[297] Zheng Tang, Gaoang Wang, Hao Xiao, Aotian Zheng, and Jenq-Neng Hwang. Single-camera and inter-camera vehicle tracking and 3d speed estimation based on fusion of visual and semantic features. In *CVPR Workshop (CVPRW) on the AI City Challenge*, 2018.

[298] Bo Tao, Xinbo Chen, Xiliang Tong, Du Jiang, and Baojia Chen. Self-supervised monocular depth estimation based on channel attention. In *Photonics*, volume 9, page 434. MDPI, 2022.

[299] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 266–282. Springer, 2020.

[300] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019.

[301] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019.

[302] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems*, 34, 2021.

[303] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[304] Hugo Touvron, Matthieu Cord, Alaaeldin El-Nouby, Jakob Verbeek, and Hervé Jégou. Three things everyone should know about vision transformers. *arXiv preprint arXiv:2203.09795*, 2022.

[305] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999.

[306] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[307] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

[308] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7942–7951, 2019.

[309] Gaoang Wang, Jenq-Neng Hwang, Kresimir Williams, and George Cutter. Closed-loop tracking-by-detection for rov-based multiple fish tracking. In *Computer Vision for Analysis of Underwater Imagery (CVAUI), 2016 ICPR 2nd Workshop on*, pages 7–12. IEEE, 2016.

[310] Gaoang Wang, Yizhou Wang, Haotian Zhang, Renshu Gu, and Jenq-Neng Hwang. Exploit the connectivity: Multi-object tracking with trackletnet. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 482–490, 2019.

[311] Li Wang, Liang Du, Xiaoqing Ye, Yanwei Fu, Guodong Guo, Xiangyang Xue, Jianfeng Feng, and Li Zhang. Depth-conditioned dynamic message propagation for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 454–463, 2021.

[312] Tai Wang, Conghui He, Zhe Wang, Jianping Shi, and Dahua Lin. Flava: Find, localize, adjust and verify to annotate lidar-based point clouds. In *Adjunct Publication of the 33rd Annual ACM Symposium on User Interface Software and Technology*, UIST '20 Adjunct, page 31–33, New York, NY, USA, 2020. Association for Computing Machinery.

[313] Tai Wang, Xinge Zhu, and Dahua Lin. Reconfigurable voxels: A new representation for lidar-based point clouds. In *Conference on Robot Learning*, 2020.

[314] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 913–922, 2021.

[315] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Probabilistic and geometric depth: Detecting objects in perspective. In *Conference on Robot Learning*, 2021.

[316] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 568–578, 2021.

[317] Xinlong Wang, Wei Yin, Tao Kong, Yuning Jiang, Lei Li, and Chunhua Shen. Task-aware monocular depth estimation for 3d object detection. In *AAAI Conference on Artificial Intelligence*, 2020.

[318] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object

detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8445–8453, 2019.

[319] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8445–8453, 2019.

[320] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q. Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[321] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. *arXiv preprint arXiv:2109.07107*, 2021.

[322] Yizhou Wang, Yen-Ting Huang, and Jenq-Neng Hwang. Monocular visual object 3d localization in road scenes. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 917–925, 2019.

[323] Yizhou Wang, Jenq-Neng Hwang, Gaoang Wang, Hui Liu, Kwang-Ju Kim, Hung-Min Hsu, Jiarui Cai, Haotian Zhang, Zhongyu Jiang, and Renshu Gu. Rod2021 challenge: A summary for radar object detection challenge for autonomous driving applications. In *Proceedings of the 2021 International Conference on Multimedia Retrieval*, pages 553–559, 2021.

[324] Yizhou Wang, Zhongyu Jiang, Xiangyu Gao, Jenq-Neng Hwang, Guanbin Xing, and Hui Liu. Rodnet: Radar object detection using cross-modal supervision. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 504–513, 2021.

[325] Yizhou Wang, Gaoang Wang, Hung-Min Hsu, Hui Liu, and Jenq-Neng Hwang. Rethinking of radar's role: A camera-radar dataset and systematic annotator via coordinate alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2815–2824, 2021.

[326] Yizhou Wang, Haotian Zhang, Zhongyu Jiang, Jie Mei, Cheng-Yen Yang, Jiarui Cai, Jenq-Neng Hwang, Kwang-Ju Kim, and Pyong-Kun Kim. Hvps: A human video panoptic segmentation framework.

[327] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7278–7286, 2018.

[328] Yu-Xiong Wang and Martial Hebert. Learning to learn: Model regression networks for easy small sample learning. In *European Conference on Computer Vision*, pages 616–634. Springer, 2016.

[329] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, pages 180–191. PMLR, 2022.

[330] Zhongdao Wang, Liang Zheng, Yixuan Liu, and Shengjin Wang. Towards real-time multi-object tracking. *arXiv preprint arXiv:1909.12605*, 2(3):4, 2019.

[331] Mark Weber, Jun Xie, Maxwell Collins, Yukun Zhu, Paul Voigtlaender, Hartwig Adam, Bradley Green, Andreas Geiger, Bastian Leibe, Daniel Cremers, et al. Step: Segmenting and tracking every pixel. *arXiv preprint arXiv:2102.11859*, 2021.

[332] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. *arXiv preprint arXiv:2112.09133*, 2021.

[333] Longyin Wen, Wenbo Li, Junjie Yan, Zhen Lei, Dong Yi, and Stan Z Li. Multiple target tracking based on undirected hierarchical relation hypergraph. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1282–1289, 2014.

[334] Xinshuo Weng and Kris Kitani. A baseline for 3d multi-object tracking. *arXiv preprint arXiv:1907.03961*, 2019.

[335] Xinshuo Weng and Kris Kitani. Monocular 3d object detection with pseudo-lidar point cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[336] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017.

[337] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[338] Mitchell Wortsman, Gabriel Ilharco, Mike Li, Jong Wook Kim, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. *arXiv preprint arXiv:2109.01903*, 2021.

[339] Raymond E Wright. Logistic regression. 1995.

[340] Di Wu, Zhaoyong Zhuang, Canqun Xiang, Wenbin Zou, and Xia Li. 6d-vnet: End-to-end 6-dof vehicle pose estimation from monocular rgb images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[341] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22–31, 2021.

[342] Yu Xiang, Alexandre Alahi, and Silvio Savarese. Learning to track: Online multi-object tracking by decision making. In *Proceedings of the IEEE international conference on computer vision*, pages 4705–4713, 2015.

[343] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010.

[344] Tete Xiao, Piotr Dollar, Mannat Singh, Eric Mintun, Trevor Darrell, and Ross Girshick. Early convolutions help transformers see better. *Advances in Neural Information Processing Systems*, 34, 2021.

[345] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 418–434, 2018.

[346] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34, 2021.

[347] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[348] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. *arXiv preprint arXiv:2111.09886*, 2021.

[349] Zhenda Xie, Zheng Zhang, Xizhou Zhu, Gao Huang, and Stephen Lin. Spatially adaptive inference with stochastic feature sampling and interpolation. In *European conference on computer vision*, pages 531–548. Springer, 2020.

[350] Bin Xu and Zhenzhong Chen. Multi-level fusion based 3d object detection from monocular images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[351] Zhenbo Xu, Wei Zhang, Xiao Tan, Wei Yang, Xiangbo Su, Yuchen Yuan, Hongwu Zhang, Shilei Wen, Errui Ding, and Liusheng Huang. Pointtrack++ for effective online multi-object tracking and segmentation. *arXiv preprint arXiv:2007.01549*, 2020.

[352] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10), 2018.

[353] Boyu Yang, Chang Liu, Bohao Li, Jianbin Jiao, and Qixiang Ye. Prototype mixture models for few-shot semantic segmentation. In *European Conference on Computer Vision*, pages 763–778. Springer, 2020.

[354] Fan Yang, Xin Chang, Chenyu Dang, Ziqiang Zheng, Sakriani Sakti, Satoshi Nakamura, and Yang Wu. Remots: Self-supervised refining multi-object tracking and segmentation. *arXiv e-prints*, pages arXiv–2007, 2020.

[355] Fan Yang, Wongun Choi, and Yuanqing Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2129–2137, 2016.

[356] Shuo Yang, Lu Liu, and Min Xu. Free lunch for few-shot learning: Distribution calibration. *arXiv preprint arXiv:2101.06395*, 2021.

[357] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *IEEE International Conference on Computer Vision*, 2019.

[358] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *IEEE International Conference on Computer Vision*, 2019.

[359] Zhuyu Yao, Jiangbo Ai, Boxun Li, and Chi Zhang. Efficient detr: Improving end-to-end object detector with dense prior. *arXiv preprint arXiv:2104.01318*, 2021.

[360] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. *CVPR*, 2021.

[361] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[362] Young-chul Yoon, Abhijeet Boragule, Kwangjin Yoon, and Moongu Jeon. Online multi-object tracking with historical appearance matching and scene adaptive detection filtering. *arXiv preprint arXiv:1805.10916*, 2018.

[363] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659, 2016.

[364] Yurong You, Yan Wang, Wei-Lun Chao, Divyansh Garg, Geoff Pleiss, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. *arXiv preprint arXiv:1906.06310*, 2019.

[365] Yurong You, Yan Wang, Wei-Lun Chao, Divyansh Garg, Geoff Pleiss, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. In *ICLR*, 2020.

[366] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2(5):6, 2018.

[367] Jiahui Yu, Yuning Jiang, Zhangyang Wang, and Thomas Cao, Zhimin amd Huang. Unitbox: An advanced object detection network. In *ACM Multimedia Conference*, 2016.

[368] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1307–1315, 2018.

[369] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 558–567, 2021.

[370] Fuzhen Zhang. Quaternions and matrices of quaternions. *Linear algebra and its applications*, 251:21–57, 1997.

[371] Gongjie Zhang, Zhipeng Luo, Yingchen Yu, Kaiwen Cui, and Shijian Lu. Accelerating detr convergence via semantic-aligned matching. *arXiv preprint arXiv:2203.06883*, 2022.

[372] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, Yue Sun, Tong He, Jonas Mueller, R. Manmatha, Mu Li, and Alexander Smola. Resnest: Split-attention networks. *CoRR*, abs/2004.08955, 2020.

[373] Haotian Zhang, Haorui Ji, Aotian Zheng, Jenq-Neng Hwang, and Ren-Hung Hwang. Monocular 3d localization of vehicles in road scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2855–2864, 2021.

[374] Haotian Zhang, Yizhou Wang, Jiarui Cai, Hung-Min Hsu, Haorui Ji, and Jenq-Neng Hwang. Lifts: Lidar and monocular image fusion for multi-object tracking and segmentation.

[375] Haotian Zhang, Yizhou Wang, Jiarui Cai, Hung-Min Hsu, Haorui Ji, and Jenq-Neng Hwang. Lifts: Lidar and monocular image fusion for multi-object tracking and segmentation. In *BMTT Challenge Workshop, IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[376] Haotian Zhang, Yizhou Wang, Zhongyu Jiang, Cheng-Yen Yang, Jie Mei, Jiarui Cai, Jenq-Neng Hwang, Kwang-Ju Kim, and Pyong-Kun Kim. U3d-molts: Unified 3d monocular object localization, tracking and segmentation. In *ICCV Segmenting and Tracking Every Point and Pixel: 6th Workshop on Benchmarking Multi-Target Tracking*, volume 6, 2021.

[377] Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Harold Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding. *arXiv preprint arXiv:2206.05836*, 2022.

[378] Hongyi Zhang, Andreas Geiger, and Raquel Urtasun. Understanding high-level semantics by modeling traffic patterns. In *Proceedings of the IEEE international conference on computer vision*, pages 3056–3063, 2013.

[379] Li Zhang, Yuan Li, and Ramakant Nevatia. Global data association for multi-object tracking using network flows. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[380] Renrui Zhang, Rongyao Fang, Peng Gao, Wei Zhang, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*, 2021.

[381] Renrui Zhang, Han Qiu, Tai Wang, Xuanzhuo Xu, Ziyu Guo, Yu Qiao, Peng Gao, and Hongsheng Li. Monodetr: Depth-aware transformer for monocular 3d object detection. *arXiv preprint arXiv:2203.13310*, 2022.

[382] Wenwei Zhang, Hui Zhou, Shuyang Sun, Zhe Wang, Jianping Shi, and Chen Change Loy. Robust multi-modality multi-object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2365–2374, 2019.

[383] Yanting Zhang, Haotian Zhang, Gaoang Wang, Jie Yang, and Jenq-Neng Hwang. Bundle adjustment for monocular visual odometry based on detections of traffic signs. *IEEE transactions on vehicular technology*, 69(1):151–162, 2019.

[384] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *arXiv preprint arXiv:2004.01888*, 2020.

[385] Yinmin Zhang, Xinzhu Ma, Shuai Yi, Jun Hou, Zhihui Wang, Wanli Ouyang, and Dan Xu. Learning geometry-guided depth via projective modeling for monocular 3d object detection. *arXiv preprint arXiv:2107.13931*, 2021.

[386] Yunpeng Zhang, Jiwen Lu, and Jie Zhou. Objects are different: Flexible monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3289–3298, June 2021.

[387] Zhaoxiang Zhang, Kaiqi Huang, Tieniu Tan, and Yunhong Wang. 3d model based vehicle tracking using gradient based fitness evaluation under particle filter framework. In *2010 20th International Conference on Pattern Recognition*, pages 1771–1774. IEEE, 2010.

[388] Zhaoxiang Zhang, Tieniu Tan, Kaiqi Huang, and Yunhong Wang. Three-dimensional deformable-model-based localization and recognition of road vehicles. *IEEE transactions on image processing*, 21(1):1–13, 2011.

[389] Zhimeng Zhang, Jianan Wu, Xuan Zhang, and Chi Zhang. Multi-target, multi-camera tracking by hierarchical clustering: Recent progress on dukemtmc project. *arXiv preprint arXiv:1712.09531*, 2017.

[390] Minghang Zheng, Peng Gao, Xiaogang Wang, Hongsheng Li, and Hao Dong. End-to-end object detection with adaptive clustering transformer. *arXiv preprint arXiv:2011.09315*, 2020.

[391] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6881–6890, 2021.

[392] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

[393] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019.

[394] Dingfu Zhou, Xibin Song, Yuchao Dai, Junbo Yin, Feixiang Lu, Miao Liao, Jin Fang, and Liangjun Zhang. Iafa: Instance-aware feature aggregation for 3d object detection from a single image. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[395] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021.

[396] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *arXiv preprint arXiv:2109.01134*, 2021.

[397] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *arXiv preprint arXiv:2109.01134*, 2021.

[398] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. *arXiv preprint arXiv:2203.05557*, 2022.

[399] Tinghui Zhou, Matthew Brown, Noah Snavely, and David Lowe. Unsupervised learning of depth and ego-motion from video. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[400] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European Conference on Computer Vision*, pages 474–490. Springer, 2020.

[401] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

[402] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[403] Yunsong Zhou, Yuan He, Hongzi Zhu, Cheng Wang, Hongyang Li, and Qinhong Jiang. Monocular 3d object detection: An extrinsic parameter free approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7556–7566, June 2021.

[404] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. *arXiv preprint arXiv:1908.09492*, 2019.

[405] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. *CoRR*, abs/1908.09492, 2019.

[406] Ji Zhu, Hua Yang, Nian Liu, Minyoung Kim, Wenjun Zhang, and Ming-Hsuan Yang. Online multi-object tracking with dual matching attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 366–382, 2018.

[407] Xinge Zhu, Yuexin Ma, Tai Wang, Yan Xu, Jianping Shi, and Dahua Lin. Ssn: Shape signature networks for multi-class object detection from point clouds. In *Proceedings of the European Conference on Computer Vision*, 2020.

[408] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *Proceedings of the European Conference on Computer Vision*, 2021.

[409] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

[410] Yi Zhu, Karan Sapra, Fitsum A Reda, Kevin J Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8856–8865, 2019.

[411] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. Eco: Efficient convolutional network for online video understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 695–712, 2018.