Linköping Studies in Science and Technology
Dissertation No. 2476

Spatiotemporal Learning for Motion Estimation and Visual Recognition

Yushan Zhang

Linköping Studies in Science and Technology
Dissertations, No. 2476

Spatiotemporal Learning for Motion Estimation and Visual Recognition

Yushan Zhang

Linköping University
Department of Electrical Engineering
Computer Vision Laboratory
SE-581 83 Linköping, Sweden

Linköping 2025

Edition 1:1

ISBN 978-91-8118-231-6 (print) 978-91-8118-232-3 (PDF)
ISSN 0345-7524
URL https://doi.org/10.3384/9789181182323

Typeset using LaTeX

Printed by LiU-Tryck, Linköping 2025

POPULÄRVETENSKAPLIG SAMMANFATTNING

Forskningen inom datorseende har genomgått en snabb utveckling. Från att ha fokuserat på igenkänningsuppgifter som klassificering, detektion och segmentering, har trenden inom visuell analys gradvis skiftat mot att lära sig spatio-temporal information. Denna avhandling presenterar arbeten med fokus på spatio-temporalt lärande, särskilt för rörelseskattning och visuell igenkänning.

Först behandlar vi problemet med objektspårning i video. Tidigare metoder för objektspårning har i stor utsträckning förlitat sig på att lära sig bättre representationsmodeller för objektens utseende, medan de spatio-temporala relationerna för varje individuellt objekt har varit mindre utforskade. Vi föreslår att använda optiska flödesegenskaper för att uppnå bättre generaliseringsförmåga i semi-superviserad videoobjektsegmentering. Vi föreslår att direkt använda de optiska flödesegenskaperna i målets representation. Våra experiment och analyser visar att ett rikare funktionsuttryck med spatio-temporal information förbättrar både segmenteringskvalitet och igenkänningsprecision.

Därefter undersöker vi spatio-temporalt lärande i 3D för rörelseskattning, det vill säga scenflödesuppskattning. Scenflödesuppskattning är ett viktigt forskningsområde inom 3D-datorseende och har tillämpningar inom bland annat robotik, autonom körning, navigering i miljöer samt spårning. Vi angriper problemet från olika perspektiv: 1. Vad är den bästa formuleringen för att lösa problemet och hur kan vi lära oss en bättre spatio-temporal funktionsrepresentation? 2. Kan vi införa osäkerhetsuppskattning i uppgiften, vilket är avgörande för säkerhetskritiska tillämpningar? 3. Hur kan vi skala upp beräkningen till stora datamängder, t.ex. autonoma scener, och samtidigt utnyttja temporal information utan att öka beräkningskostnaden för mycket? För att besvara dessa frågor undersöker vi användningen av transformatorer för bättre funktionsrepresentation, diffusionmodeller för osäkerhetsuppskattning samt mer effektiva metoder för funktionsinlärning för att skala upp till flerbildsscener i autonom körning.

Slutligen tar vi ett steg längre och kombinerar visuell segmentering, spårning och öppen vokabulärigenkänning i Lidar-sekvenser, särskilt inom autonoma scenarier. I autonoma körmiljöer är det mycket viktigt att segmentera, spåra och känna igen varje objekt. Med dagens mänskliga annoteringar är det möjligt att spåra trafikanter såsom bilar och fotgängare ganska väl. Men vi vill ta detta ett steg längre: mot att segmentera och spåra vad som helst i Lidar. För detta ändamål föreslår vi en pseudoetiketteringsmotor som använder 2D-visionsmodellen SAM och bildspråksmodellen CLIP för att automatiskt märka Lidar-strömmen. Vi föreslår dessutom modellen SAL-4D för att segmentera, spåra och känna igen objekt i ett zero-shot-sammanhang.

Sammanfattningsvis undersöker vi inlärning av spatio-temporal information i både 2D-bild- och 3D-punktmolnsdomäner. Utifrån bilddomänen visar vi att spatio-temporal information förbättrar kvaliteten på videoobjektsegmentering samt generaliseringsförmågan. I 3D-punktmolnsdomänen visar vi att spatio-temporalt lärande ger mer exakt rörelseskattning och möjliggör den första metoden för zero-shot segmentering, spårning och öppen vokabulärigenkänning av godtyckliga objekt.

ABSTRACT

The field of computer vision has undergone rapid development. Starting from recognition tasks such as classification, detection, and segmentation, the focus of visual analysis has gradually shifted towards learning spatiotemporal information. This thesis presents research on spatiotemporal learning, with a particular emphasis on motion estimation and visual recognition.

First, we address the problem of video object tracking. Previous methods have primarily relied on learning improved appearance representations, while the spatiotemporal relationships of individual objects have been underexplored. We propose leveraging optical flow features to achieve higher generalization in semi-supervised video object segmentation, directly incorporating these features into both the target representation and the decoder network. Our experiments and analysis show that enriching feature representations with spatiotemporal information improves segmentation quality and generalization capability.

Next, we investigate spatiotemporal learning in 3D for motion estimation, specifically scene flow estimation. Scene flow estimation as an important research topic in 3D computer vision is crucial for applications such as robotics, autonomous driving, embodied navigation, and tracking. We investigate the problem in different perspectives: 1. What is the best formulation for solving the problem and how to learn a better spatiotemporal feature representation? 2. Can we introduce uncertainty estimation to the task, which is of crucial importance for safety-critical downstream tasks? 3. How to scale the estimation to large-scale data, e.g., autonomous scenes, and leverage the temporal information without introducing much computation overheads? To answer these questions, we explore the use of transformers for improved feature representation, diffusion models for uncertainty estimation, and efficient feature learning methods for multi-frame, large-scale autonomous driving scenarios.

Finally, we extend our research to joint visual segmentation, tracking, and open-vocabulary recognition in LiDAR sequences, particularly for autonomous scenes. In such environments, precise segmentation, tracking, and recognition of objects are essential for downstream analysis and control. Current human-annotated open-source datasets allow for reasonable tracking of traffic participants such as cars and pedestrians. However, we aim to advance beyond this towards segmenting and tracking any object in LiDAR data. To this end, we propose a pseudo-labeling engine that leverages the 2D visual foundation model SAM v2 and the vision-language model CLIP to automatically label LiDAR streams. We further introduce the SAL-4D model, capable of segmenting, tracking, and recognizing any object in a zero-shot manner.

In summary, we explore the learning of spatiotemporal information in both 2D image and 3D point cloud domains. In the image domain, we demonstrate that spatiotemporal information improves video object segmentation quality and generalization. In the 3D point cloud domain, we show that spatiotemporal learning enables more accurate motion estimation and facilitates the first method for zero-shot segmentation, tracking, and open-vocabulary recognition of arbitrary objects.

Acknowledgments

It has been four years since the beginning of my PhD journey. During this time, I have had the opportunity to conduct exciting research, travel to new places, and meet many inspiring individuals. I am deeply grateful to all those who made this journey possible, and to those who accompanied and supported me throughout these years.

I would like to express my sincere gratitude to my main supervisor, Michael Felsberg, for his excellent guidance and for granting me the freedom to pursue my research interests throughout my studies. I am also deeply appreciative of the inclusive and supportive environment he fostered, which allowed me to feel safe and encouraged to express myself. In addition, I am grateful for his support in finding my internship and research visit opportunities. I would also like to thank my co-supervisor, Maria Magnusson, not only for her valuable academic supervision but also for her genuine concern for aspects of my life beyond research. Furthermore, I am very thankful to Bastian Wandt, who, although not formally my supervisor, provided significant supervision and engaged in many inspiring discussions that greatly enriched my PhD journey.

I would like to acknowledge the Wallenberg AI, Autonomous Systems and Software Program (WASP) for funding my PhD studies. I am also grateful for the computational resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) at C3SE, partially funded by the Swedish Research Council (grant 2022-06725), as well as the Berzelius resource, supported by the Knut and Alice Wallenberg Foundation at the National Supercomputer Center. This research would not have been possible without these contributions.

I would like to thank my lab mates for the many stimulating discussions, fruitful collaborations, enjoyable after-work activities, and memorable board games. Special thanks go to Emil Brissman and Pavlo Melnyk for their generous help during my moves, particularly Emil, who kindly assisted me twice with his new car. I am also grateful to Jie Zhao and Qiyu Sun for their companionship and support during challenging times, especially when dealing with paper reviews. My thanks extend to Andreas Robinson, Arvi Jonnarth, Johan Edstedt, Per-Erik Forssén, and Qingwen Zhang for the joint publications and collaborative efforts. Finally, I would like to thank all my other lab mates, past and present, for contributing to an inspiring working environment.

Finally, I would like to express my deepest gratitude to my parents for their unwavering support throughout my PhD journey. Their encouragement, understanding, and patience during challenging times have been invaluable, and this work would not have been possible without them.

Yushan Zhang

Linköping, October 2025

Contents

Part I

Background and Overview

1

INTRODUCTION

1.1 Motivation

Recent research in computer vision demonstrates significant progress, particularly in static scenarios such as image classification, object detection, and segmentation. As LiDAR sensors become increasingly prevalent, recognition tasks in the LiDAR domain have also seen considerable advancement, enabling more accurate and detailed 3D understanding of environments. These developments reflect a growing capacity of machine intelligence to interpret static scenes with remarkable precision. More recently, focus has shifted toward generative tasks such as image synthesis and scene generation, which go beyond mere understanding. These works empower machines to imagine and create plausible 2D and 3D representations of the world, showcasing a higher level of cognitive ability akin to human imagination. Research in vision-language models further enriches this landscape by connecting visual data with natural language, facilitating multi-modal understanding that merges perception with semantic reasoning.

In contrast, the pursuit of temporal reasoning remains comparatively underdeveloped. Challenges such as handling complex dynamic data formats and the intensive effort required for detailed annotations may hinder progress in this area. Despite these hurdles, understanding the dynamic world is crucial for achieving truly autonomous and intelligent systems. A comprehensive artificial intelligence must grasp both spatial and temporal aspects of real-world scenarios, enabling it to interpret movements, interactions, and changes over time.

This thesis concentrates on spatiotemporal learning within computer vision, aiming to advance the current boundaries by exploring and extending methodologies related to optical flow, scene flow, video object tracking, and LiDAR panoptic tracking. By addressing these challenging areas, the work strives to bring machine perception closer to human-like understanding of dynamic environments.

1.2 Outline

This thesis is divided into two parts. Part I contains seven chapters, providing an overview of relevant research fields and how our contributions are related to the context. The second chapter introduces the *spatiotemporal learning toolbox*, including transformers, diffusion models, segment anything model, and CLIP. The third chapter introduces *optical flow estimation* from 2D images. Both traditional optimization methods and deep learning methods are discussed. The fourth chapter introduces *video object segmentation*, specifically, semi-supervised video object segmentation and how optical flow helps with the performance. The fifth chapter further explores *scene flow estimation* from 3D point clouds. Different machine learning methods are discussed to solve the problem. The sixth chapter discusses *LiDAR panoptic segmentation* with an emphasis on zero-shot learning and open vocabulary recognition. The last chapter concludes the thesis. Part II contains the full version of all the publications presented in the thesis. Paper A leverages optical flow information to learn a better feature representation for video object segmentation. Paper B employs transformers for feature representation learning and formulates scene flow estimation as a global matching problem. Paper C further introduces diffusion models into the scene flow estimation problem, which improves the accuracy and in the mean time enables uncertainty estimation. Paper D explores more efficient feature learning for multi-frame scene flow estimation in large-scale autonomous driving scenes. Paper E leverages the power of vision foundation model SAM 2 and vision-language model CLIP and proposes the first zero-shot 4D LiDAR panoptic segmentation method.

1.3 Included Publications

Paper A: Leveraging Optical Flow Features for Higher Generalization Power in Video Object Segmentation

Yushan Zhang, Andreas Robinson, Maria Magnusson, and Michael Felsberg. "Leveraging Optical Flow Features for Higher Generalization Power in Video Object Segmentation". In: *2023 IEEE International Conference on Image Processing (ICIP)*.

Abstract: We propose to leverage optical flow features for higher generalization power in semi-supervised video object segmentation. Optical flow is usually exploited as additional guidance information in many computer vision tasks. However, its relevance in video object segmentation was mainly in unsupervised settings or using the optical flow to warp or refine the previously predicted masks. Different from the latter, we propose to directly leverage the optical flow features in the target representation. We show that this enriched representation improves the encoder-decoder approach to the segmentation task. A model to extract the combined information from the optical flow and the image is proposed, which is then used as input to the target model and the decoder network. Unlike previous methods, e.g. in tracking where concatenation is used to integrate information from image data and optical flow, a simple yet effective attention mechanism is exploited in our work. Experiments on DAVIS 2017 and YouTube-VOS 2019 show that integrating the information extracted from optical flow into the original image branch results in a strong performance gain, especially in unseen classes which demonstrates its higher generalization power.

Author's contribution: The method was developed jointly with the co-authors. The author was the main contributor of the experiments and the manuscript.

Paper B: GMSF: Global Matching Scene Flow

Yushan Zhang, Johan Edstedt, Bastian Wandt, Per-Erik Forssén, Maria Magnusson, and Michael Felsberg. "GMSF: Global Matching Scene Flow". In: *Advances in Neural Information Processing Systems* 36 (2023)

Abstract: We tackle the task of scene flow estimation from point clouds. Given a source and a target point cloud, the objective is to estimate a translation from each point in the source point cloud to the target, resulting in a 3D motion vector field. Previous dominant scene flow estimation methods require complicated coarse-to-fine or recurrent architectures as a multi-stage refinement. In contrast, we propose a significantly simpler single-scale one-shot global matching to address the problem. Our key finding is that reliable feature similarity between point pairs is essential and sufficient to estimate accurate scene flow. We thus propose to decompose the feature extraction step via a hybrid local-global-cross transformer architecture which is crucial to accurate and robust feature representations. Extensive experiments show that the proposed Global Matching Scene Flow (GMSF) sets a new state-of-theart on multiple scene flow estimation benchmarks. On FlyingThings3D, with the presence of occlusion points, GMSF reduces the outlier percentage from the previous best performance of 27.4% to 5.6%. On KITTI Scene Flow, without any fine-tuning, our proposed method shows state-of-the-art performance. On the Waymo-Open dataset, the proposed method outperforms previous methods by a large margin. The code is available at https://github.com/ZhangYushan3/GMSF.

Author's contribution: The author developed the methods, conducted the experiments, and was the main contributor of the manuscript.

Paper C: DiffSF: Diffusion Models for Scene Flow Estimation

Yushan Zhang, Bastian Wandt, Maria Magnusson, and Michael Felsberg. "DiffSF: Diffusion Models for Scene Flow Estimation". In: *Advances in Neural Information Processing Systems* 37 (2024) (***Spotlight***)

Abstract: Scene flow estimation is an essential ingredient for a variety of real-world applications, especially for autonomous agents, such as self-driving cars and robots. While recent scene flow estimation approaches achieve reasonable accuracy, their applicability to real-world systems additionally benefits from a reliability measure. Aiming at improving accuracy while additionally providing an estimate for uncertainty, we propose DiffSF that combines transformer-based scene flow estimation with denoising diffusion models. In the diffusion process, the ground truth scene flow vector field is gradually perturbed by adding Gaussian noise. In the reverse process, starting from randomly sampled Gaussian noise, the scene flow vector field prediction is recovered by conditioning on a source and a target point cloud. We show that the diffusion process greatly increases the robustness of predictions compared to prior approaches resulting in state-of-the-art performance on standard scene flow estimation benchmarks. Moreover, by sampling multiple times with different initial states, the denoising process predicts multiple hypotheses, which enables measuring the output uncertainty, allowing our approach to detect a majority of the inaccurate predictions. The code is available at https://github.com/ZhangYushan3/DiffSF.

Author's contribution: The author conducted the experiments, and was the main contributor of the methods and the manuscript.

Paper D: DeltaFlow: An Efficient Multi-frame Scene Flow Estimation Method

Qingwen Zhang, Xiaomeng Zhu, Yushan Zhang, Yixi Cai, Olov Andersson, and Patric Jensfelt. "DeltaFlow: An Efficient Multi-frame Scene Flow Estimation Method". In: *arXiv preprint arXiv:2508.17054* (2025)

Abstract: Previous dominant methods for scene flow estimation focus mainly on input from two consecutive frames, neglecting valuable information in the temporal domain. While recent trends shift towards multi-frame reasoning, they suffer from rapidly escalating computational costs as the number of frames grows. To leverage temporal information more efficiently, we propose DeltaFlow (ΔFlow), a lightweight 3D framework that captures motion cues via a Δ scheme, extracting temporal features with minimal computational cost, regardless of the number of frames. Additionally, scene flow estimation faces challenges such as imbalanced object class distributions and motion inconsistency. To tackle these issues, we introduce a Category-Balanced Loss to enhance learning across underrepresented classes and an Instance Consistency Loss to enforce coherent object motion, improving flow accuracy. Extensive evaluations on the Argoverse 2 and Waymo datasets show that ΔFlow achieves state-of-the-art performance with up to 22% lower error and 2× faster inference compared to the next-best multi-frame supervised method, while also demonstrating a strong cross-domain generalization ability.

Author's contribution: The author contributed to the methods that was mainly developed by the Qingwen Zhang. The author contributed to writing the manuscript.

Paper E: Zero-shot 4D Lidar Panoptic Segmentation

Yushan Zhang, Aljoša Ošep, Laura Leal-Taixé, and Tim Meinhardt. "Zero-Shot 4D Lidar Panoptic Segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2025).

Abstract: Zero-shot 4D segmentation and recognition of arbitrary objects in Lidar is crucial for embodied navigation, with applications ranging from streaming perception to semantic mapping and localization. However, the primary challenge in advancing research and developing generalized, versatile methods for spatiotemporal scene understanding in Lidar lies in the scarcity of datasets that provide the necessary diversity and scale of annotations. To overcome these challenges, we propose ***SAL-4D*** (Segment Anything in Lidar–4D), a method that utilizes multi-modal robotic sensor setups as a bridge to distill recent developments in Video Object Segmentation (VOS) in conjunction with off-the-shelf Vision-Language foundation models to Lidar. We utilize VOS models to pseudo-label tracklets in short video sequences, annotate these tracklets with sequence-level CLIP tokens, and lift them to the 4D Lidar space using calibrated multi-modal sensory setups to distill them to our ***SAL-4D*** model. Due to temporal consistent predictions, we outperform prior art in 3D Zero-Shot Lidar Panoptic Segmentation (LPS) over 5 PQ, and unlock Zero-Shot 4D-LPS.

Author's contribution: The author conducted the experiments. The method was jointly developed with all the co-authors. The manuscript was jointly written with Aljoša Ošep and and Tim Meinhardt.

2

SPATIOTEMPORAL LEARNING TOOLBOX

This section presents a set of widely used computer vision models that underpin the spatiotemporal learning approaches explored in this thesis. These models, namely Transformers, Diffusion Models, the Segment Anything Model (SAM), and CLIP, have each demonstrated significant capabilities in understanding and processing visual information, and they provide a diverse set of tools for modeling complex patterns in both space and time. By leveraging the strengths of these models, we aim to develop more robust and generalizable approaches to learning from visual data with temporal dynamics. In the subsequent sections, each model is introduced in detail, including its core architecture and functionality. We also discuss how each model contributes to the key innovations and methodologies proposed in this thesis.

2.1 Transformers in vision

Transformers [120] were originally introduced for machine translation tasks. Unlike previous architectures based on recurrent or convolutional neural networks, it pioneered a framework based entirely on the attention mechanism. This mechanism generates queries, keys, and values from sequential input data, computes an attention map from the queries and the keys, and produces the output as a weighted sum of the values. The core component, attention, can be written as the following equations:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_{model}}})V, \tag{2.1}$$

where Q, K, and V denote the query, key, and value matrices, respectively. d_{model} is the dimension of the input sequence.

Instead of performing a single attention directly on the d_{model} channels, it is often beneficial to first project d_{model} to d_k, d_k, d_v dimensions for query, key, and value, respectively, and then perform multi-head attention:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O, \tag{2.2}$$

where $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$. $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, and $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$ are linear projection parameters for query, key, and value, respectively. $W^O \in \mathbb{R}^{hd_v \times d_{model}}$ maps the output to d_{model} channels.

To incorporate the order of the input sequence, positional encoding is introduced and added to the input features, which is computed as:

$$\begin{aligned} PE_{(pos,2i)} &= \sin(pos/10000^{2i/d_{model}}), \\ PE_{(pos,2i+1)} &= \cos(pos/10000^{2i/d_{model}}), \end{aligned} \tag{2.3}$$

where pos denotes the position, and i denotes the dimension.

Transformers have not only been widely employed in Large Language Models (LLMs) [120, 21, 9], but have also been proven efficient in visual recognition tasks [23, 12, 75, 15]. Vision Transformer (ViT) [23] is the first to apply a pure Transformer architecture to vision tasks. It treats each image as a sequence of 16×16 patches, encodes each patch into a vector, and adds a position embedding to each vector. The resulting sequence of vectors is then processed by a standard Transformer. A multilayer perceptron (MLP) head is added on top of the output feature from the Transformer to perform image classification. DETR [12] applies a Transformer encoder-decoder architecture to the task of object detection [32, 119, 67]. It takes image features extracted by a Convolutional Neural Network (CNN) [63, 109, 39] as input to the Transformer and outputs a set of bounding box predictions that can be supervised given the ground truth. Swin Transformer [75] further improves the Transformer architecture with shifted windows for hierarchical visual learning, and serves as a general-purpose backbone for vision tasks, including object detection, panoptic segmentation [50], etc. MaskFormer [15] challenges the per-pixel classification method for semantic segmentation and proposes a Transformer module for the task, which outputs queries with object information that, together with per-pixel feature embeddings, can be decoded to per-pixel masks. Many other works further employs Transformer into multi-object tracking [84], point cloud classification and segmentation [139, 37], image enhancement [129, 13], image generation [93], and image matching [114], etc.

In Paper A [137], we employ a Transformer to learn a more robust and discriminative feature representation that combines the appearance information and the motion information, which helps with improving the video object segmentation performance. In Paper B [135], we formulate scene flow estimation as a matching process, and employ a local-global-cross Transformer architecture to learn robust features for each point. This results in better feature representations and helps with scene flow estimation performance.

2.2 Diffusion Models in Vision

Diffusion models have gained significant attention in image generation tasks [40, 104]. They are defined by two processes. A forward process, or diffusion process, incrementally adds noise to clean images until they become pure Gaussian noise:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}), \tag{2.4}$$

where $\mathbf{x}_0$ denotes the clean input image, and $\mathbf{x}_t$ denotes the noisy image at timestep t. β_s is the noise scheduler, and $\bar{\alpha}_t := \prod_{s=1}^{t}(1-\beta_s)$, controls the magnitude of the noise added. A reverse process has a closed form conditioned on $\mathbf{x}_0$ that recovers images from Gaussian noise:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t\mathbf{I}), \tag{2.5}$$

where $\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t$, and $\tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$.

This reverse process can be learned by a neural network $q_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ by minimizing the difference between the closed form $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ and the neural network output $q_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$. The training

objective can be formulated either as minimizing the discrepancy between the estimated noise and the injected noise:

$$\mathcal{L} = \|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2, \tag{2.6}$$

or equivalently, as minimizing the difference between the reconstructed clean image and the original clean input:

$$\mathcal{L} = \|\mathbf{x}_0 - f_\theta(\mathbf{x}_t, t)\|^2, \tag{2.7}$$

where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ denotes the added noise, and $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$ denotes the noisy input at the timestep t.

During inference, the learned neural network is used to predict $\mathbf{x}_{t-1}$ from $\mathbf{x}_t$. At each step a random Gaussian noise $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is sampled. Then, $\mathbf{x}_{t-1}$ can be computed either from ϵ_θ:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)) + \sqrt{\tilde{\beta}_t}\mathbf{z}, \tag{2.8}$$

or equivalently from f_θ:

$$\mathbf{x}_{t-1} = \tilde{\mu}_t(\mathbf{x}_t, f_\theta(\mathbf{x}_t, t)) + \sqrt{\tilde{\beta}_t}\mathbf{z}. \tag{2.9}$$

Beyond their success in producing realistic images and videos, researchers have also begun exploring their utility in regression problems. CARD [38], for example, introduces a diffusion-based model capable of both classification and regression, effectively capturing predictive means and associated uncertainties. DiffusionDet [14] reformulates object detection as a denoising diffusion process, transitioning from noisy bounding boxes to accurate object boxes. Baranchuk et al. [6] apply diffusion models to semantic segmentation, particularly in scenarios with limited labeled data. Similarly, DiffusionInst [36] represents instances using instance-aware filters and frames instance segmentation as a denoising task, mapping noise to filters. Jiang et al. [47] extend diffusion models to point cloud registration, operating over the rigid body transformation group. Recent work in optical flow and depth estimation [105] further demonstrates the applicability of diffusion models to dense vision tasks.

In Paper C [138], we employ diffusion models to scene flow estimation. Without having to train the model multiple times, we are able to estimate the prediction uncertainty with the help of diffusion models, which is of crucial importance for safety-critical downstream tasks.

2.3 Minkowski U-Net

Minkowski U-Net [18] has been extensively adopted for efficient point cloud processing. They were proposed to address the limitations of prior methods that process 3D video data frame-by-frame, and instead enable direct processing of 4-dimensional LiDAR sequences using 4D sparse convolutions.

In this framework, the input typically has the format of point coordinates $C_p \in \mathbb{R}^{N_p \times D}$ and their corresponding features $F_p \in \mathbb{R}^{N_p \times N_f}$. These points are first processed by a sparse tensor quantization to generate a sparse tensor, also known as voxelization. The float point coordinates are quantized according to the voxel size v_l:

$$C'_p = \text{floor}(C'_p / v_l). \tag{2.10}$$

This results in integer voxel coordinates. For each voxel coordinate, the sparse quantization keeps only one point, and duplicate points within the same voxel are removed. The output is a set of voxel coordinates $C_v \in \mathbb{R}^{N_v \times D}$ and the corresponding features $F_v \in \mathbb{R}^{N_v \times N_f}$, where N_v is the number of voxels, and D is the 4D coordinate x_n, y_n, z_n, t_n for each voxel.

To perform the sparse convolution, the sparse tensor is first encoded into the Coordinate (COO) format [118]:

$$\mathbf{C} = \begin{pmatrix} x_1 & y_1 & z_1 & t_1 & b_1 \\ & & ... & & \\ x_{N_v} & y_{N_v} & z_{N_v} & t_{N_v} & b_{N_v} \end{pmatrix}, \mathbf{F} = \begin{pmatrix} F_1^T \\ ... \\ F_{N_v}^T \end{pmatrix}, \tag{2.11}$$

where $\mathbf{C}$ stores the 4D coordinates along with batch indices for all non-empty voxels, and $\mathbf{F}$ represents the corresponding feature vectors. To support arbitrary coordinate configurations and kernel shapes, a kernel map is first defined as a list of input-output pairs:

$$\mathrm{K} = \{(I_{\mathbf{in}}, O_{\mathbf{out}})\} \text{ for } I_{\mathbf{in}} \in C^{\text{in}} \text{ and } O_{\mathbf{out}} \in C^{\text{out}}, \tag{2.12}$$

where C^{in} and C^{out} represent the predefined input and output coordinates of the sparse tensors. **in** and **out** denote the indices of the input and output coordinates. A generalized version of sparse convolution [34, 33] is then proposed as:

$$x_{\mathbf{u}}^{\text{out}} = \sum_{\mathbf{i} \in \mathcal{N}^D(\mathbf{u}, C^{\text{in}})} W_{\mathbf{i}} x_{\mathbf{u}+\mathbf{i}}^{\text{in}} \text{ for } \mathbf{u} \in C^{\text{out}}, \tag{2.13}$$

where $\mathbf{u} \in \mathbb{R}^D$ and $\mathbf{i} \in \mathbb{R}^D$ denote the D-dimensional output and offset coordinates, respectively. The neighborhood function $\mathcal{N}^D(\mathbf{u}, C^{\text{in}}) = \{\mathbf{i} \mid \mathbf{u} + \mathbf{i} \in C^{\text{in}}, \mathbf{i} \in \mathcal{N}^D\}$ defines the set of valid offsets, as specified by the kernel map K, from the current output coordinate $\mathbf{u}$ to the input coordinates C^{in}. $W_{\mathbf{i}} \in \mathbb{R}^{N^{\text{out}} \times N^{\text{in}}}$ denotes the convolutional kernel weights for each offset element $\mathbf{i}$, where N^{in} and N^{out} are the input and output feature dimensions, respectively.

The pooling after the sparse convolution is performed by reducing the input features that map to the same output coordinate. Similar to the convolution, a pooling map is defined as:

$$\mathrm{P} = \{(I_{\mathbf{in}}, O_{\mathbf{out}})\} \text{ for } I_{\mathbf{in}} \in C^{\text{in}} \text{ and } O_{\mathbf{out}} \in C^{\text{out}}, \tag{2.14}$$

Then, the max pooling procedure can be written as:

$$x_{\mathbf{u}}^{\text{out}} = \max_{\mathbf{i} \in \mathcal{N}^D(\mathbf{u}, C^{\text{in}})} x_{\mathbf{u}+\mathbf{i}}^{\text{in}} \text{ for } \mathbf{u} \in C^{\text{out}}, \tag{2.15}$$

where all variables are defined similarly as in Equation (2.13).

In Paper D [134] and Paper E [136], we address LiDAR sequence processing in autonomous driving scenarios, targeting both motion estimation and visual recognition tasks. To handle the large-scale and sparsely distributed 4D data, we adopt the Minkowski U-Net architecture, which efficiently encodes and aggregates spatio-temporal information from LiDAR streams while preserving the rich structural and temporal information inherent in the data.

2.4 Segment Anything

There has been a trend in the construction of large foundation models [97, 91, 51, 99, 1] in the computer vision community. The Segment Anything Model (SAM) [51] is one such model released in recent years. Unlike previous models that are mostly limited to specific object categories, SAM aims to segment anything with very high quality. To achieve this, the method is threefold: 1. Promptable segmentation: Given an image to be segmented, the prompt can be points, bounding boxes, masks, or texts. 2. The SAM model: The prompt and the image are encoded by a prompt encoder and an image encoder, respectively. A lightweight mask decoder is then applied to produce the final mask. 3. Data engine: A data engine is used to collect the SA-1B dataset, which contains over 1 billion masks for training.

More recently, SAM 2 [99] was proposed. As an image only captures one moment of the real world and lacks the temporal dimension of information, the newly released model not only

supports segmenting anything in images, but also supports tracking anything in videos. SAM 2 shares a similar structure with SAM and is also threefold: 1. Promptable segmentation: Prompts can be provided in one or multiple frames. 2. Temporal modeling: A memory bank and a memory attention mechanism are introduced to handle temporal information. 3. Data engine: The SA-V dataset is collected, consisting of 642.6K masklets.

In Paper E [136], we develop a 4D LiDAR panoptic segmentation method that aims to segment and track any object in the LiDAR sequence. To reduce the need for manual data annotation, we devise a pseudo-label engine that distills knowledge from SAM 2 in the 2D image domain into the 3D point cloud domain.

2.5 CLIP

Vision Language Models (VLMs) have rapidly advanced in recent years, significantly enhancing multimodal understanding. These models leverage both computer vision models and Large Language Models (LLMs) to jointly interpret images and text, enabling downstream applications such as image captioning [107], visual question answering (VQA) [2], and cross-modal retrieval [132].

A pioneering work in VLMs is CLIP [97], which maps both visual and linguistic features into a shared embedding space. It learns discriminative representations by pulling features of paired image-text samples closer while pushing unpaired features farther apart. The learning process can be formulated as:

$$\mathcal{L}_{I \to T} = -\frac{1}{B} \sum_{i=1}^{B} \log \frac{\exp\left(z_i^I \cdot z_i^T / \tau\right)}{\sum_{j=1}^{B} \exp\left(z_i^I \cdot z_j^T / \tau\right)}, \tag{2.16}$$

$$\mathcal{L}_{T \to I} = -\frac{1}{B} \sum_{i=1}^{B} \log \frac{\exp\left(z_i^T \cdot z_i^I / \tau\right)}{\sum_{j=1}^{B} \exp\left(z_i^T \cdot z_j^I / \tau\right)}, \tag{2.17}$$

where $\mathcal{L}_{I \to T}$ learns to contrast the query image with the text keys, and $\mathcal{L}_{T \to I}$ learns to contrast the query text with the image keys. z_i^I and z_i^T are image and text embeddings. B is the batch size. τ is the temperature hyperparameter that controls the representation density.

Unlike previous state-of-the-art computer vision systems, which are typically limited to a predefined fixed set of categories, CLIP enables visual recognition with an open vocabulary. Moreover, the training data for the CLIP model consists of image-text pairs that are easily obtained from the internet, making large-scale model training more feasible.

In Paper E [136], the goal is to segment any object in the scene, without being confined to predefined object classes, which is often the case in existing datasets. We employ CLIP to annotate object semantics using CLIP features, enabling open vocabulary recognition during inference.

3

Optical Flow Estimation

Understanding the dynamics of the world is fundamental to the construction of artificial intelligence, as it enables machines to interpret, predict, and respond to real-world situations. This involves not only recognizing patterns and relationships within data, but also grasping the complexities of human behavior, physical environments, and ever-changing social contexts. Without this foundational understanding, AI systems risk becoming brittle, unreliable, or misaligned with human values and needs. The first step toward achieving this understanding is to estimate the motions of different objects. This task is nontrivial, as even humans sometimes find it difficult to perceive precise object movement. For machines, the challenge is even greater, as they "see" the world only through sensors. In the early days, machines relied primarily on cameras to perceive their surroundings. However, an image is merely a 2D projection of the real world and inherently loses 3D information, making motion estimation even more difficult. Given these limitations, research on motion estimation began in the 2D domain, specifically with optical flow. In this section, we first present the problem formulation for optical flow, followed by an overview of traditional optimization-based methods and recent deep learning approaches.

3.1 Problem Formulation

The concept of optical flow originated from the psychologist James Gibson in the 1940s. The idea is that, as animals move through their environment, the pattern of luminance changes, providing a visual stimulus. This stimulus not only serves as a control signal for locomotion, but also conveys rich information, including the perception of shape, distance, and movement, that can ultimately lead to a more comprehensive understanding of the environment. The modern definition of optical flow was introduced by Horn and Schunck in the 1980s. It refers to the apparent 2D motion of pixels on the image plane from one frame to the next in a video sequence. Formally, given two images $\mathbf{I}_{\text{source}} \in \mathbb{R}^{W \times H \times 3}$ and $\mathbf{I}_{\text{target}} \in \mathbb{R}^{W \times H \times 3}$, with W and H being the width and height of the image, the goal is to estimate an optical flow vector field $\mathbf{V} \in \mathbb{R}^{W \times H \times 2}$ that describes the movement of each pixel in the x and y direction from $\mathbf{I}_{\text{source}}$ to $\mathbf{I}_{\text{target}}$. An illustrative visualization is shown in Figure 3.1.

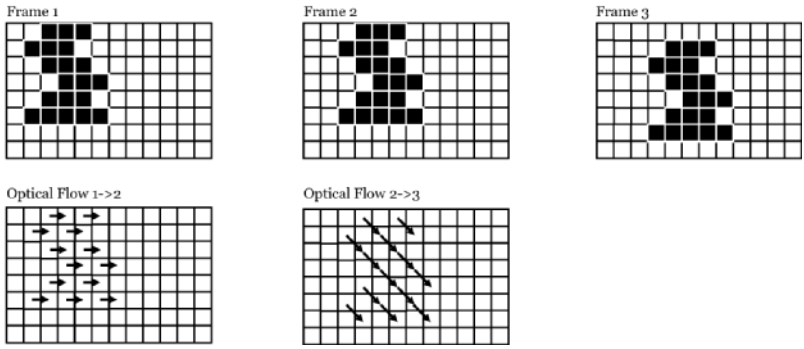

Figure 3.1: Optical flow illustrative visualization. First row left to right: frame 1, frame 2, and frame 3. Second row left to right: optical flow from frame 1 to frame 2, and optical flow from frame 2 to frame 3.

3.2 Traditional Methods

Many efforts have been made by researchers to solve this problem. Traditional methods such as Lucas–Kanade [77], Horn–Schunck [43], and Gunnar–Farnebäck [27] typically optimize the optical flow under three assumptions: First, the movement of objects between consecutive frames should be small. Second, the brightness constancy assumption states that the same object in both frames should have a similar appearance. Finally, the smoothness assumption suggests that changes in the optical flow vector should be smooth. We briefly introduce these three methods in the following to provide an intuition for how traditional optical flow approaches work.

3.2.1 Lucas–Kanade Method

The Lucas–Kanade method [77] assumes local constancy in the flow vector field, meaning that the flow vector is constant within a small neighborhood Ω around the pixel. To compute the optical flow, the following cost function is minimized for each pixel:

$$E_{\mathrm{LK}}(u,v) = \iint\limits_{(x,y)\in\Omega} w(x,y)[(I_x u + I_y v + I_t)^2]dxdy, \tag{3.1}$$

where x and y are the image pixel coordinates, and u and v are the two components of the optical flow vector for the current pixel. I_x, I_y, and I_t are the image partial derivatives with respect to x, y, and t, respectively. $w(x,y)$ is the weighting function within the small neighborhood.

3.2.2 Horn–Schunck Method

The Horn–Schunck method assumes a smooth change in the optical flow vector field and generalizes the smoothness assumption to the entire image, minimizing the following cost function over the whole image:

$$E_{\mathrm{HS}}(u,v) = \iint [(I_x u + I_y v + I_t)^2 + \alpha^2(\parallel \nabla u \parallel^2 + \parallel \nabla v \parallel^2)]dxdy, \tag{3.2}$$

where x and y are the image pixel coordinates, and u and v are the two components of the optical flow vector for the current pixel. I_x, I_y, and I_t are the image partial derivatives with respect to x, y, and t, respectively. The parameter α controls the smoothness of the optical flow vector. $\nabla u = (\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y})$ and $\nabla v = (\frac{\partial v}{\partial x}, \frac{\partial v}{\partial y})$ are the spatial gradients of u and v.

3.2.3 Gunnar-Farneback Method

One limitation of both methods is the assumption that the motion field is temporally consistent. However, the Gunnar–Farnebäck [27] method addresses this limitation by using a polynomial expansion of the images to compensate for such cases:

$$f_i(\mathbf{x}) = \mathbf{x}^T \mathbf{A}_i \mathbf{x} + \mathbf{b}_i^T \mathbf{x} + c_i, \tag{3.3}$$

where $\mathbf{x} = (x, y)$ denotes the image pixel coordinates, and $\mathbf{A}_i$, $\mathbf{b}_i$, and c_i are the parameters of the polynomial expansion for the ith image. Let $\mathbf{d} = (u, v)^T$ represent the pixel displacement, i.e., the optical flow vector. Applying the brightness constancy assumption over a local region, we have $f_2(\mathbf{x}) = f_1(\mathbf{x} - \mathbf{d})$. Under the smoothness assumption, where $\mathbf{A}(\mathbf{x}) \approx \mathbf{A}_1(\mathbf{x}) \approx \mathbf{A}_2(\mathbf{x})$, the problem can be expressed as minimizing the following equation:

$$E_{\mathrm{GF}}(u, v) = \iint_{(x,y)\in\Omega} w(x, y) \parallel \mathbf{b}_2(x, y) - \mathbf{b}_1(x - u, y - v) + 2\mathbf{A}(x, y)(u, v)^T \parallel^2 dxdy. \tag{3.4}$$

To handle large motion, a coarse-to-fine refinement scheme is commonly employed: a coarse flow is first estimated at a low resolution, then up-sampled and refined at higher resolution levels. Another approach is recurrent warping, where in each iteration, the source image is warped according to the previous estimation, and the residual flow is estimated between the warped source image and the target image.

3.3 Deep Learning Methods

Despite the good results that traditional methods can achieve in certain scenarios, they are limited by their assumptions and may fail when illumination changes across consecutive frames, when motion is not smooth (e.g., at edges), or when motion is very large. More recently, deep learning methods for optical flow estimation have been rapidly explored and developed. Numerous datasets for training deep models exist, including synthetic datasets such as FlyingChairs [24], FlyingThings3D [82], MPI Sintel [10], VIPER [101], Infinigen [98], and Spring [83], as well as real-world datasets like Middlebury [5], KITTI [31, 86], and HD1K [53]. A visualization of these datasets is provided in Figure 3.2. In the following, we briefly introduce some representative optical flow estimation architectures and discuss how to train such models.

3.3.1 Encoder-Decoder Methods

The first series of works addressing the optical flow problem are FlowNet [24] and FlowNet 2.0 [45], which employ an encoder-decoder convolutional neural network as a key component of their architecture. The simplest way to input both images into an encoder-decoder network is by concatenating them along the channel dimension. However, FlowNet [24] proposes an alternative approach: features are first extracted separately from each input image, then combined

Figure 3.2: A visualization of optical flow datasets with images and optical flows. Top left: FlyingThings3D [82]. Top right: KITTI [31, 86]. Bottom left: MPI Sintel [10]. Bottom right: Spring [83].

to form a mixed feature, which is finally passed through a decoder network to produce the final prediction. FlowNet 2.0 [45], a follow-up work, proposes combining multiple FlowNets to better handle large displacements. The overall flow of the encoder-decoder methods is shown in Figure 3.3.

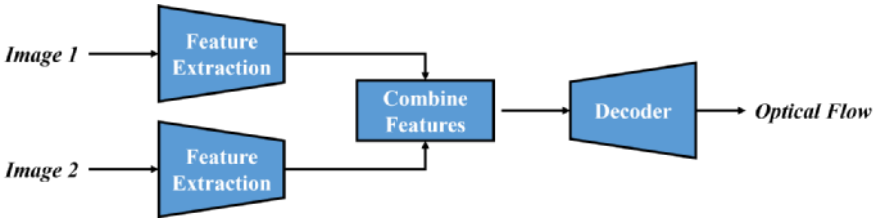

Figure 3.3: The overall flow of the encoder-decoder optical flow methods.

3.3.2 Soft Correspondence Methods

Intuitively, optical flow can be seen as a dense matching problem; that is, for each pixel in the source image, the goal is to find the corresponding pixel in the target image. The optical flow is then simply computed as the displacement between these pixels. One challenge is that optical flow can be subpixel, meaning a pixel in the source image might not correspond exactly to a pixel in the target image. This issue can be addressed using soft correspondence methods. For example, GMFlow [128] treats optical flow estimation as a matching problem by finding multiple soft matches for each pixel and making a weighted decision based on these matches. The overall flow of the soft correspondence methods is shown in Figure 3.4.

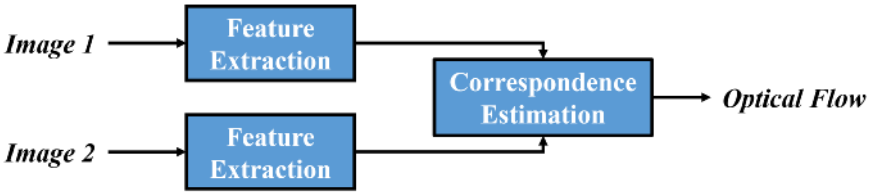

Figure 3.4: The overall flow of the soft correspondence optical flow methods.

3.3.3 Multi-Scale Methods

To handle large motions, multi-scale estimation is commonly employed in deep learning–based optical flow methods. PWC-Net [113], for example, uses a coarse-to-fine refinement scheme. First, multi-scale features are extracted from the two images using a neural network. Then, at each scale, a set of neural networks takes the image features at that level along with the optical flow estimated from the coarser scale as input, and outputs the refined optical flow for the current scale. The overall flow of the multi-scale methods is illustrated in Figure 3.5.

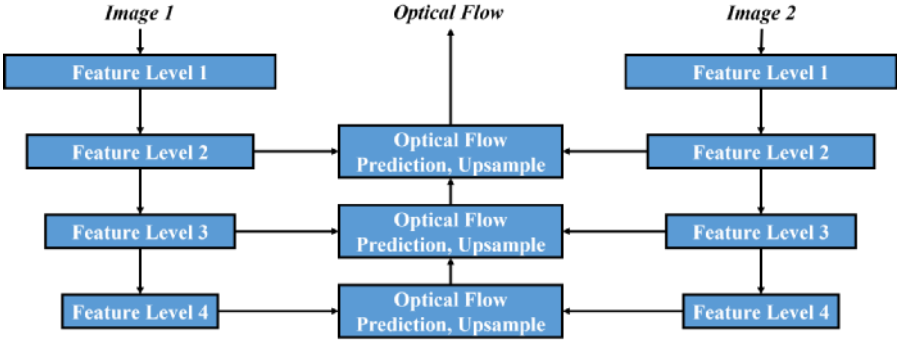

Figure 3.5: The overall flow of the multi-scale optical flow methods.

3.3.4 Recurrent Methods

Recently, RAFT [117] introduced a recurrent network that iteratively refines the optical flow estimation, similar to the recurrent warping procedure used in traditional methods. Many other methods [48, 112, 4, 44, 25, 124] have since been developed based on the RAFT architecture. Specifically, RAFT first constructs 4D correlation volumes. At each iteration, the 4D correlation volumes, context features from the source image, and the optical flow estimation from the previous iteration are concatenated and passed through a set of ConvGRU units. The refined optical flow for the next iteration is then decoded by convolutional neural networks. The overall workflow of recurrent methods is shown in Figure 3.6.

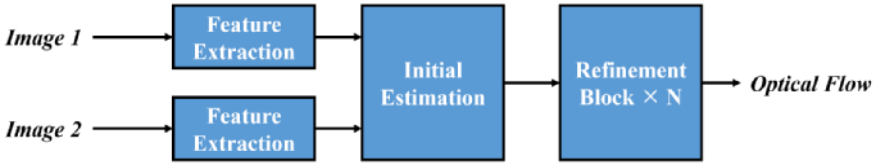

Figure 3.6: The overall flow of the recurrent optical flow methods.

3.4 Training Losses

Optical flow estimation can be trained using both fully-supervised and self-supervised approaches, depending on the availability of ground truth data. In this section, we will present several widely adopted loss functions for training the task.

3.4.1 Fully-supervised Losses

Most of the optical flow methods are fully-supervised. Thanks to the existing datasets [24, 82, 10, 101, 98, 83, 5, 31, 86, 53] this is possible. For a given pair of images $\mathbf{I}_{\text{source}} \in \mathbb{R}^{W\times H\times 3}$ and $\mathbf{I}_{\text{target}} \in \mathbb{R}^{W\times H\times 3}$, the supervision signal is the 2D movement vector for each pixel in the source image $\mathbf{V} \in \mathbb{R}^{W\times H\times 2}$. The loss then can be defined by minimizing the distance between the optical flow prediction and the ground truth.

$$\mathcal{L}_{\text{fully}} = \sum_{\mathbf{x}} \parallel \hat{\mathbf{v}} - \mathbf{v}_{\text{gt}} \parallel_{\text{n}}, \tag{3.5}$$

where $\mathbf{x}$ denotes image pixels. $\hat{\mathbf{v}}$ and $\mathbf{v}_{\text{gt}}$ denote the predicted optical flow and the ground truth for pixel $\mathbf{x}$, respectively. $\|\cdot\|_{\text{n}}$ can be either the L1 norm, the L2 norm, or a robust loss with n less than 1.

3.4.2 Self-supervised Losses

Annotating optical flow requires substantial human effort. Although synthetic datasets can circumvent this challenge, they often differ significantly from real-world data. To address this gap, researchers have investigated self-supervised loss functions for learning optical flow without relying on ground truth annotations. Representative works include Yu [133] and UnFlow [85], which propose various losses. Among these, the three most important losses are data loss, smoothness loss, and consistency loss.

Data loss is based on the observation that the brightness of corresponding pixels in the two images should be similar. It is formally defined as:

$$\mathcal{L}_{\text{data}} = \sum_{\mathbf{x}} \|\mathbf{I}_{\text{source}}(\mathbf{x}) - \mathbf{I}_{\text{target}}(\mathbf{x} + \hat{\mathbf{v}})\|_{\text{n}}, \tag{3.6}$$

where $\|\cdot\|_{\text{n}}$ can be either the L1 norm, the L2 norm, or a robust loss with n less than 1.

Smoothness loss is based on the assumption that the movement of objects should be locally consistent. It is formally defined as:

$$\mathcal{L}_{\text{smoothness}} = \sum_{\mathbf{x}} \|\nabla_{\mathbf{x}}\hat{\mathbf{v}}\|_{\text{n}}, \tag{3.7}$$

where $\|\cdot\|_n$ can be either the L1 norm, the L2 norm, or a robust loss with n less than 1.

Consistency loss compares the forward and backward optical flow. In non-occluded areas, the forward and backward optical flows should be consistent. It is formally defined as:

$$\mathcal{L}_{\text{consistency}} = \|\hat{\mathbf{v}}^{for}(\mathbf{x}) + \hat{\mathbf{v}}^{back}(\mathbf{x} + \hat{\mathbf{v}}^{for}(\mathbf{x}))\|_n + \|\hat{\mathbf{v}}^{back}(\mathbf{x}) + \hat{\mathbf{v}}^{for}(\mathbf{x} + \hat{\mathbf{v}}^{back}(\mathbf{x}))\|_n, \tag{3.8}$$

where $\|\cdot\|_n$ can be either the L1 norm, the L2 norm, or a robust loss with n less than 1.

4

VIDEO OBJECT SEGMENTATION

Having established how to learn 2D motion from videos, this capability can be exploited to enhance visual recognition in video sequences. Early effort in the field of visual recognition starts from visual object tracking [28] of single objects with bounding boxes [58, 29, 103, 55, 59, 57, 61], multiple objects with bounding boxes [66, 87, 19], to multiple objects with segmentation masks [56, 60, 62]. In this section, we focus on video object segmentation (VOS). Since different objects often exhibit distinct motion patterns, the optical flow fields typically reveal clear boundaries between objects. Such motion-derived boundaries are highly valuable for video object segmentation. VOS is a broad and active area of research, encompassing several different branches. We first provide an overview of various types of VOS. We then focus on one specific type, semi-supervised VOS, and illustrate how incorporating optical flow can improve segmentation performance.

4.1 Problem Formulation

Depending on the level of user input, VOS methods are commonly categorized into unsupervised VOS (UVOS), semi-supervised VOS (SVOS), and interactive VOS (IVOS). In UVOS, the task is to segment and track salient objects without any user annotations; given only a video sequence, objects of interest are detected and segmented automatically. In SVOS, the method is provided with segmentation masks of the target objects in the first frame, and the goal is to predict the masks for these objects in all subsequent frames. IVOS involves the highest level of user interaction, where the method receives a video sequence along with user-provided prompts, and must segment and track the specified objects in the following frames. In this chapter, we focus on semi-supervised video object segmentation.

4.2 Semi-supervised VOS Methods

Extensive research has been conducted on semi-supervised video object segmentation (SVOS), and existing approaches can be broadly classified into several categories: (1) online fine-tuning–based methods, (2) propagation-based methods, (3) unsupervised or weakly supervised

methods, and (4) matching-based methods. In the following sections, we outline the core ideas behind each category.

4.2.1 Online Fine-tuning based Methods

Online fine-tuning based methods [11, 79] were among the first methods to address semi-supervised VOS. These methods leverage advancements in more developed image classification and segmentation tasks by transferring learned knowledge from the image domain to the video domain. The process typically begins with pretraining the model on large-scale static image datasets such as ImageNet [20], enabling it to develop a general understanding of semantics. While this provides some segmentation capability, it lacks precision. Starting from the pretrained model, the network is then fine-tuned on segmentation datasets such as DAVIS [95], improving its ability to segment objects more accurately. However, the model at this stage is still optimized for generic object segmentation, whereas SVOS requires segmenting the specific object indicated in the first frame. To achieve this, a final online fine-tuning stage is applied: given a test video sequence, the model is further fine-tuned extensively on the first frame, allowing it to reliably track and segment the same object throughout the video. The overall training scheme is illustrated in Figure 4.1.

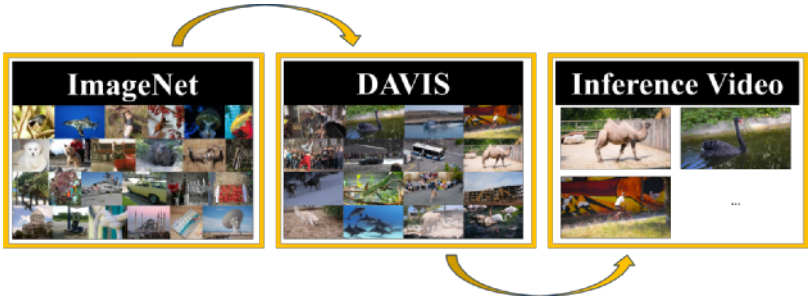

Figure 4.1: The overall training scheme of online fine-tuning based SVOS methods. The training consists of three stages: the first training stage is on image classification datasets such as ImageNet [20], the second training stage is on segmentation datasets such as DAVIS [95], and the final stage is online fine-tuning on the inference video.

4.2.2 Propagation-based Methods

Propagation-based methods [46, 94] address SVOS under the assumption that the same object in consecutive frames remains spatially close. Consequently, starting from the mask of the previous frame and propagating it according to the current frame should yield a reasonably accurate mask prediction. To implement this idea, a network is trained to take the current frame together with the masks from previous frames as input, and to output the predicted mask for the current frame. The general paradigm of propagation-based methods is illustrated in Figure 4.2.

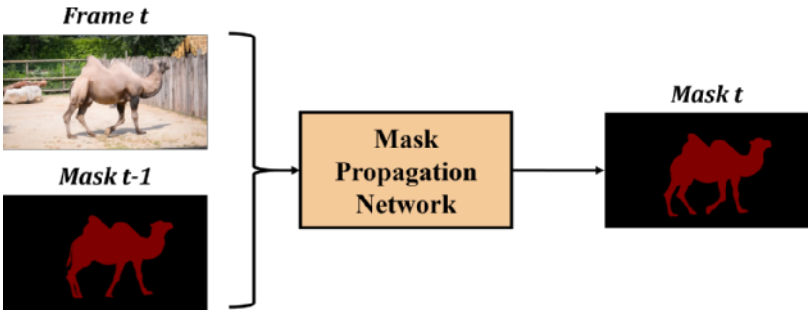

Figure 4.2: General paradigm of propagation-based SVOS. The mask propagation network takes the current frame and the mask of the previous frame as input, and outputs the mask prediction for the current frame.

4.2.3 Un-/Weakly-supervised based Methods

Un-/Weakly-supervised methods [76, 123, 64, 73] aim to alleviate the reliance on extensive pixel-wise annotations required for fully-supervised SVOS, as obtaining such annotations is both time-consuming and labor-intensive. Some approaches adopt a reconstruction loss [64, 73] as the training objective. These methods learn pixel-wise correspondences between consecutive video frames, and then propagate the previous frame using the predicted correspondences to reconstruct the current frame. Other approaches employ a cycle-consistent loss [76, 123], where, after a forward and backward prediction cycle, the reconstructed frame and predicted mask are enforced to be consistent with the original inputs. The overall pipeline of reconstruction-based and cycle-consistent based methods are shown in Figure 4.3.

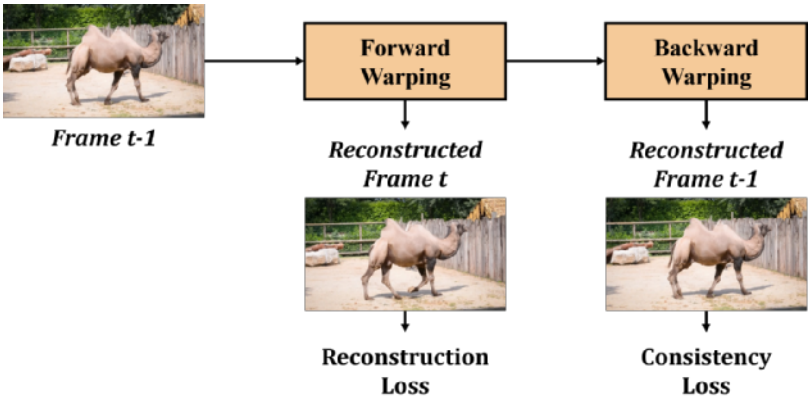

Figure 4.3: The overall pipeline of reconstruction-based and cycle-consistent based methods. The reconstruction loss and the consistency loss are obtained from forward and backward warping.

4.2.4 Generation-Based Methods

Generation-based methods [49] employ a generative appearance module that takes image features as input and estimates the posterior probabilities of foreground and background for each pixel. Specifically, the features of all pixels in the image, denoted as $\mathbf{x}p$, are modeled using Gaussian mixture models, with separate components representing the foreground and background, respectively. Each pixel is assigned to the components via soft labels $\alpha p, k \in [0, 1]$, where p denotes the pixel index and k the Gaussian component index. This produces a coarse mask, which is subsequently refined by a predictor to obtain the final segmentation mask.

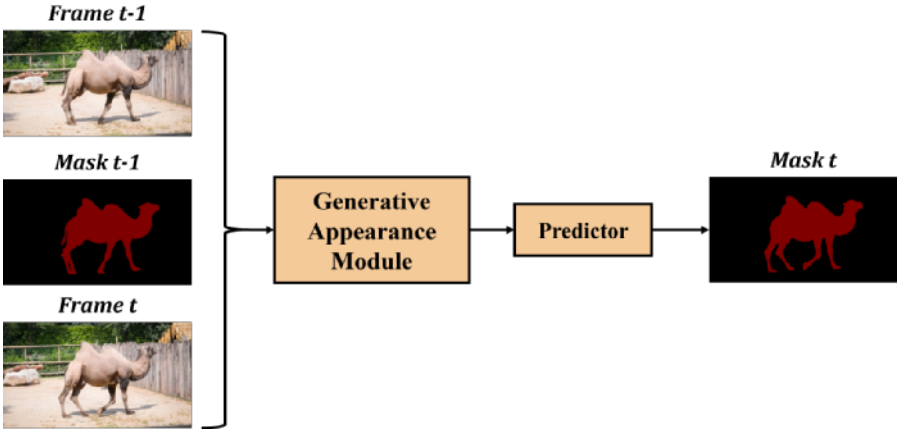

Figure 4.4: The overall flow of generation-based SVOS methods. The current frame, the previous frame, and the previous mask are first input to the generative appearance module. The output are a set of posterior foreground and background probabilities for each pixel.

4.2.5 Matching-Based Methods

Matching-based methods [16, 122, 90, 102, 8] constitute one of the most widely adopted approaches for SVOS. These methods learn an embedding representation of the target object from the first frame and subsequently identify the corresponding foreground regions in later frames by matching against the learned embeddings. This paradigm is conceptually intuitive, as it closely resembles human memory: the network first memorizes the object of interest and then tracks it across the sequence based on feature similarity. The overall flow of matching-based methods is illustrated in Figure 4.5.

4.3 Matching Based Semi-supervised VOS

Among the various approaches to SVOS, we now focus on matching-based semi-supervised VOS, which represents the most widely adopted paradigm. Many existing methods [16, 122, 90, 102, 8] fall into this category, all following the general pipeline illustrated in Figure 4.5. The primary distinctions between these methods lie in how they construct the target embeddings and how they perform the matching between the target and the current input frame. In general,

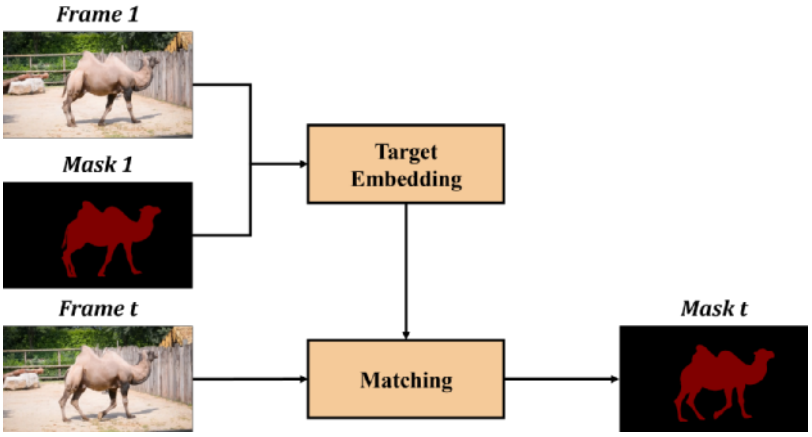

Figure 4.5: The overall flow of matching-based SVOS methods. The neural network learns an embedding representation of the target object from the first frame. The foreground region in the subsequent frames is then identified based on feature similarity with the learned embedding representation.

the target embedding encoder can be formulated as follows:

$$\text{Tgt} = \text{Enc}(\mathcal{I}, \mathcal{M}), \tag{4.1}$$

where $\mathcal{I}$ denotes the set of all previous frames and $\mathcal{M}$ denotes the set of corresponding masks, including the ground-truth mask from the first frame and all subsequently predicted masks. In practice, both the image and mask sets are often restricted to a limited number of frames, or in some cases, the masks may be excluded entirely from the encoder input. The matching stage is typically implemented by a decoder network:

$$\text{Msk} = \text{Dec}(\text{Tgt}, \text{I}_{\text{curr}}), \tag{4.2}$$

where Tgt is the learned target embedding, and I_{curr} is the current input frame.

4.3.1 Learning Fast and Robust Target Models

Given the general framework of matching-based SVOS, we now examine a specific approach [102], which focuses on learning fast and robust target models for SVOS. This method proposes a lightweight target model that first maps the initial frame to its corresponding ground-truth mask, and subsequently updates the target model as new frames from the video are received. Specifically, the encoder network used to learn the target model is implemented as a two-layer convolutional neural network:

$$\text{Tgt}(\mathbf{x}) = \mathbf{w}_2 * (\mathbf{w}_1 * \mathbf{x}), \tag{4.3}$$

where $\mathbf{x}$ denotes the feature extracted from the image I. $\mathbf{w}_1$ serves as a projection layer, mapping the high-dimensional feature $\mathbf{x}$ into a lower-dimensional embedding space. Subsequently, $\mathbf{w}_2$ is implemented as a 3×3 convolutional filter, which transforms the embedding into the predicted mask. The parameters $\mathbf{w}_1$ and $\mathbf{w}_2$ are optimized by minimizing the following objective function:

$$\mathcal{L}_{\text{Tgt}}(\mathbf{w}) = \sum_k \gamma_k \|\mathbf{v}_k \cdot (\mathbf{y}_k - \text{U}(\text{Tgt}(\mathbf{x}_k)))\|^2 + \sum_j \lambda_j \|\mathbf{w}_j\|^2, \tag{4.4}$$

where k indexes the samples used to optimize the target model, and γ_k controls the contribution of each sample to the loss. The parameter $\mathbf{v}_k$ denotes a weight mask for the k-th sample, allowing

different importance to be assigned to foreground and background regions. $\mathbf{x}_k$ and $\mathbf{y}_k$ represent the extracted feature and the corresponding binary mask of the k-th input sample, respectively. The operator U denotes bilinear upsampling, which adjusts the spatial resolution of the target model output to match that of the binary mask. The scalar λ_j controls the strength of the regularization term. The optimization of the cost function is performed using the Gauss–Newton method. To initialize the target model, the feature representation of the first image, $\mathbf{x}_0$, is taken as input. During this stage, both $\mathbf{w}_1$ and $\mathbf{w}_2$ are jointly optimized on the first frame. In subsequent frames, as new images and corresponding masks become available, they are incorporated into the optimization process; however, only $\mathbf{w}_2$ is updated, while $\mathbf{w}_1$ remains fixed. Once the target model is obtained, it is applied to the feature representation of the current input frame to produce the corresponding target score:

$$\mathbf{s} = \mathrm{Tgt}(\mathbf{x}_{\mathrm{curr}}) = \mathbf{w}_2 * (\mathbf{w}_1 * \mathbf{x}_{\mathrm{curr}}). \tag{4.5}$$

The decoder then takes both the target model output score and the current frame feature representation as input, and generates the final mask prediction:

$$\mathrm{Msk} = \mathrm{Dec}(\mathbf{s}, \mathbf{x}_{\mathrm{curr}}). \tag{4.6}$$

The decoder employs a multi-resolution hierarchical architecture. At each layer l, it takes as input the target score from the target model, $\mathbf{s}$, the feature map of the current layer, $\mathbf{x}^l$, and the output from the coarser layer, $\mathbf{z}_{l+1}$. The layer produces as output the mask logits at a finer spatial resolution, $\mathbf{z}_l$. The decoder consists of four such layers in total. The architecture of an individual decoder layer is illustrated in Figure 4.6.

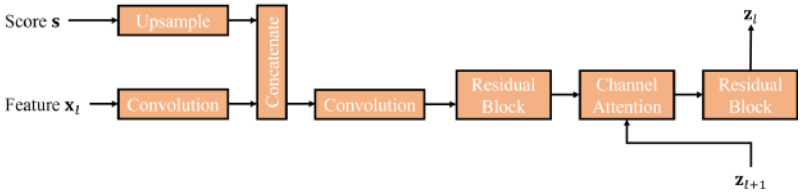

Figure 4.6: The architecture of an individual decoder layer. The network takes as input the target model output score $\mathbf{s}$, the current layer image feature $\mathbf{x}_l$, and the output from the coarser layer $\mathbf{z}_{l+1}$. The output is the mask logits at a finer resolution $\mathbf{z}_l$.

4.3.2 Training Losses

Similar to other segmentation methods, the decoder network is supervised using a binary cross-entropy loss:

$$\mathcal{L}_{\mathrm{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log p_i + (1-y_i)\log(1-p_i)], \tag{4.7}$$

where N denotes the number of observations. y_i is the ground truth label, and p_i is the predicted probability.

4.3.3 Merging Objects

The aforementioned target model and decoder network are applied to a single object. To segment multiple objects in the video, segmentation is performed for each object individually, and the resulting masks are then merged to obtain the final multi-object segmentation. For each

pixel, let the logit scores be $s_1, s_2, ..., s_N$ for N different objects. A Softmax function is applied to these scores to obtain probabilities for the N objects:

$$p_i = \frac{e^{s_i}}{\sum_{j=1}^{N} e^{s_j}}. \tag{4.8}$$

Each pixel is assigned to the object that has the highest probability p_i.

4.4 Contributions

In this section, we aim to improve the performance of the aforementioned method [102] and investigate whether optical flow can enhance SVOS performance in Section 4.4.1. Previous matching-based SVOS methods primarily rely on learning an appearance model for accurate matching, often neglecting the importance of spatio-temporal reasoning. We propose to learn a joint representation that combines both object appearance and motion patterns. By integrating motion information with appearance-based embeddings, our approach aims to achieve more robust and generalizable video object segmentation.

4.4.1 Paper A

In Paper A [137], we propose incorporating optical flow into the SVOS task. Intuitively, motion information is highly informative, as each object typically exhibits motion patterns that differ from its surroundings. This contrast can help achieve more precise and sharper object contours.

Specifically, we integrate optical flow into both the target model and the decoder network. First, the optical flow between the current frame and the previous frame is extracted using a pre-trained RAFT model. The predicted optical flow is then converted into three channels and input to the same backbone network for feature extraction. The resulting optical flow features are combined with the appearance features via an attention block to learn a more informative and robust representation.

Experiments show that incorporating optical flow features improves both the region similarity J and the contour accuracy F of the predictions, with particularly notable improvements in contour accuracy. This is likely because objects often exhibit motion patterns distinct from their surroundings, causing optical flow to change sharply at object boundaries. We also observe higher performance gains on unseen object categories during training, demonstrating improved generalization ability.

5

Scene Flow Estimation

Having discussed motion estimation and object recognition in 2D images and video, we now turn to the 3D domain. Advances in 3D sensors, such as LiDAR, have made it possible to capture the world in three dimensions. In this section, we focus on scene flow estimation, which involves estimating motion from 3D point clouds. This task is analogous to optical flow in 2D images, but with key differences. First, while image pixels are arranged on a regular two-dimensional grid, points in a point cloud have irregular three-dimensional coordinates. Second, features learned from 2D images are typically appearance-based, whereas in 3D point clouds, geometric features are more commonly used. We begin by formally defining the problem of scene flow estimation. We then review different types of estimation methods, which are closely related to optical flow approaches. Finally, we present our contributions to the field.

5.1 Problem Formulation

Given a source point cloud **P**source $\in \mathbb{R}^{N_1 \times 3}$ and a target point cloud **P**target $\in \mathbb{R}^{N_2 \times 3}$, where N_1 and N_2 are the number of points in the source and target point clouds, respectively, the objective is to estimate a scene flow vector field $\mathbf{V} \in \mathbb{R}^{N_1 \times 3}$ that maps each source point to its corresponding position in the target point cloud. An illustration is provided in Figure 5.1.

Figure 5.1: Scene flow estimation illustration. The orange points represent the source points, and the gray points represent the target points. The red point indicates the position where the selected source point should move, while the blue vector represents the scene flow vector for that source point.

Although scene flow estimation may appear straightforward, it involves several challenges. First, point clouds are typically sparse, usually captured by LiDAR sensors, which means there

is no one-to-one correspondence between source and target points. Second, occlusions can occur, objects visible in the source point cloud may be hidden in the target point cloud, making the scene flow of these points difficult to estimate. Other challenges include object deformation, repetitive patterns, and sensor noise.

5.2 Scene Flow Estimation Methods

Since the release of benchmark datasets such as FlyingThings3D [82] and KITTI Scene Flow [86], scene flow estimation has attracted considerable research attention. Many early approaches [7, 78, 86, 100, 111, 121, 130] operate under the assumption that objects in the scene are primarily rigid. These methods typically decompose the problem into subtasks, such as object detection or segmentation, followed by fitting rigid motion models to each object. While this assumption is often valid and effective in autonomous driving scenarios, where scenes generally comprise static backgrounds and rigidly moving vehicles, it overlooks non-rigid elements, such as pedestrians. Moreover, autonomous driving is just one of many application domains for scene flow estimation. In more general settings involving deformable or articulated objects, the rigidity assumption breaks down. Additionally, decomposing the task into multiple subtasks introduces non-differentiable components, which hinders end-to-end training and often necessitates instance-level supervision. Recent scene flow estimation methods explore end-to-end models, many of which are inspired by optical flow methods [24, 45, 113, 117], and can be divided into several categories.

5.2.1 Encoder-Decoder Methods

Encoder-decoder methods, exemplified by HplFlowNet [35] and FlowNet3D [74], first process the two point clouds separately through convolutional neural networks. At a coarse scale, the learned features are combined to form a mixed representation, and finally a decoder network is applied to predict the scene flow. The overall workflow of the encoder-decoder methods is illustrated in Figure 5.1.

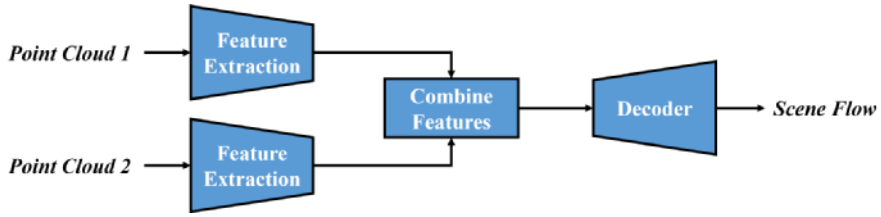

Figure 5.2: The overall flow of the encoder-decoder scene flow estimation methods.

5.2.2 Multi-Scale Methods

Multi-scale methods [17, 70, 127] share a similar idea to traditional multi-scale optical flow techniques: they first estimate the scene flow at a coarse resolution and then progressively refine it at finer scales to capture both slow and fast motions. The overall flow of the multi-scale methods is illustrated in Figure 5.3.

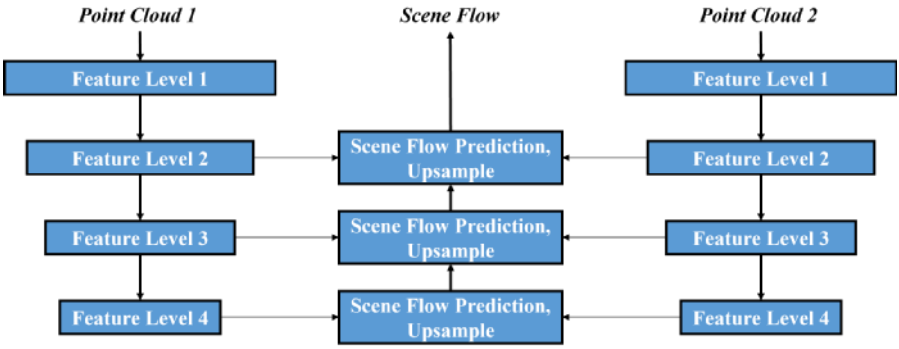

Figure 5.3: The overall flow of the multi-scale scene flow estimation methods.

5.2.3 Recurrent Methods

Recurrent methods [52, 116, 125] typically produce an initial scene flow estimate, which is then iteratively refined through multiple updates to improve accuracy. The overall flow of recurrent methods is illustrated in Figure 5.4.

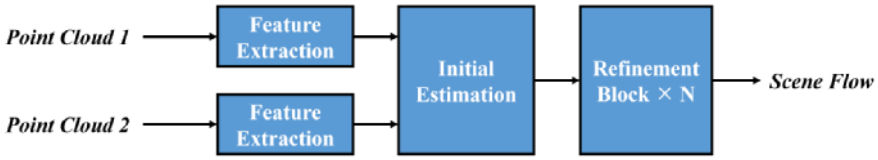

Figure 5.4: The overall flow of recurrent scene flow estimation methods.

5.2.4 Soft Correspondence Methods

Soft correspondence methods [68, 96] view scene flow estimation as a matching problem. This approach is intuitive: once the corresponding location for each source point is found, the scene flow vector can be directly obtained. These methods typically first extract features for each point, and then perform a non-learning soft correspondence estimation to predict the scene flow. The overall flow is illustrated in Figure 5.5.

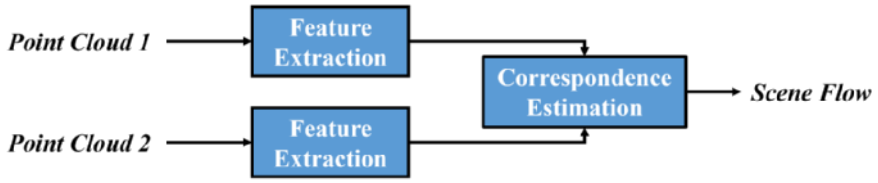

Figure 5.5: Flow chart of soft correspondence scene flow estimation methods.

5.2.5 Runtime Optimization Methods

Runtime optimization methods [71, 72, 65] employ a neural prior to estimate motion during inference. For each input pair, a new neural prior is learned, with the supervisory signal given as the distance between the warped source point cloud and the target point cloud. These methods do not require full supervision with ground-truth annotations, however, optimizing the neural network for each sample pair can be time-consuming, making real-time estimation challenging. The overall flow is illustrated in Figure 5.6.

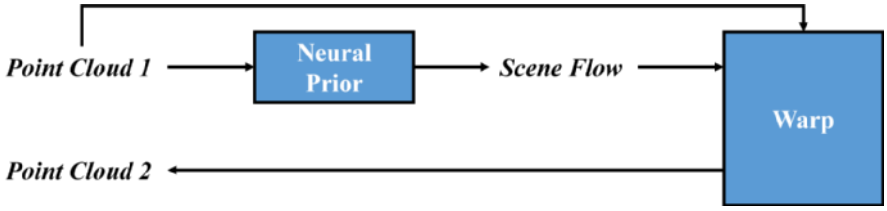

Figure 5.6: Flow chart of runtime optimization scene flow estimation methods.

In practice, different types of methods are often combined to achieve the best performance.

5.3 Training Losses

In this section, we will present the training losses for scene flow estimation, including the fully-supervised losses in Section 5.3.1 and self-supervised losses in Section 5.3.2.

5.3.1 Fully-supervised Loss

To train fully-supervised scene flow estimation methods, there exist synthetic datasets such as FlyingThings3D [82] and real-world datasets such as KITTI Scene Flow [86], Argoverse [126], and Waymo Open Dataset [115]. The supervised loss employed in scene flow estimation is the end-point error between the ground-truth scene flow vector field $\mathbf{V}_{\text{gt}} \in \mathbb{R}^{N\times 3}$ and the predicted scene flow vector field $\hat{\mathbf{V}} \in \mathbb{R}^{N\times 3}$:

$$\mathcal{L}_{\text{fully}} = \sum_{N} \|\hat{\mathbf{v}} - \mathbf{v}_{\text{gt}}\|_{\text{n}}, \tag{5.1}$$

where $\hat{\mathbf{v}} \in \hat{\mathbf{V}}$ and $\mathbf{v}_{\mathbf{gt}} \in \mathbf{V}_{\mathbf{gt}}$ are the prediction and ground-truth of a single point in the point clouds. N is the number of points in the source point cloud. $\|\cdot\|_{\text{n}}$ can be either the L1 norm, the L2 norm, or a robust loss with n less than 1.

5.3.2 Self-supervised Loss

To alleviate the need for manual annotation and address the limitation that synthetic datasets are not sufficiently realistic for training real-world applications, various self-supervised losses have been proposed [108, 65]. These can be broadly categorized into two types: data terms, such as the Chamfer loss, and regularization terms, such as smoothness and consistency losses.

The Chamfer loss minimizes the distance between the warped source point cloud and the target point cloud, and vice versa. It is formally defined as:

$$\mathcal{L}_{Chamfer} = \sum_{x_i \in \mathcal{X}} \min_{y_j \in \mathcal{Y}} \|\hat{x}_i^{'} - y_j\|_{\mathrm{n}} + \sum_{y_i \in \mathcal{Y}} \min_{x_i \in \mathcal{X}} \|x_i - \hat{y}_j^{'}\|_{\mathrm{n}}, \tag{5.2}$$

where $\mathcal{X}$ and $\mathcal{Y}$ denote the source and target point clouds, respectively. x_i and y_j are points from the source and target point clouds. $\hat{x}_i^{'}$ and $\hat{y}_j^{'}$ are the points warped by the estimated forward and backward scene flow.

The smoothness loss assumes that points located close to each other should have similar scene flow vectors. It is formally defined as:

$$\mathcal{L}_{Smoothness} = \sum_{x_i \in \mathcal{X}} \frac{1}{|\mathcal{N}(x_i)|} \sum_{y_j \in \mathcal{N}(x_i)} \|\hat{\mathbf{v}}(x_i) - \hat{\mathbf{v}}(y_j)\|_{\mathrm{n}}, \tag{5.3}$$

where $\mathcal{N}(x_i)$ denotes the nearest neighbors of x_i in $\mathcal{Y}$, and $|\mathcal{N}(x_i)|$ is the number of nearest neighbors. $\hat{\mathbf{v}}(\cdot)$ represents the predicted scene flow.

The consistency loss ensures that the forward prediction and the interpolated backward prediction are consistent, and vice versa. It is formally defined as:

$$\mathcal{L}_{Consistency} = \|\hat{\mathbf{V}}^{for} - \Omega(\hat{\mathbf{V}}^{back})\|_{\mathrm{n}} + \|\hat{\mathbf{V}}^{back} - \Omega(\hat{\mathbf{V}}^{for})\|_{\mathrm{n}}, \tag{5.4}$$

where $\Omega(\cdot)$ denotes the interpolation function. $\mathbf{V}^{for}$ and $\mathbf{V}^{back}$ are the forward and backward scene flow vector field predictions, respectively.

5.4 Contributions

In this section, we investigate scene flow estimation from several perspectives. First, how can the estimates be made more accurate? We address this question in Section 5.4.1, where we follow the soft correspondence methods and formulate scene flow estimation as a global matching problem. To improve the accuracy of the matching process, we employ transformers to learn discriminative features for each point. Second, how can the estimates be made more reliable? Since scene flow estimation can be applied in safety-critical downstream tasks, such as robotics and autonomous driving, reliability is crucial. To this end, we propose employing diffusion models to capture the uncertainty of the estimates in Section 5.4.2. Third, how can temporal information be exploited more efficiently? In Section 5.4.3, we propose a lightweight framework that captures motion information via a delta scheme. This method enables processing multiple frames without a linear increase in memory and computational complexity.

5.4.1 Paper B

In Paper B [135], we follow soft correspondence methods and propose a global matching approach for scene flow estimation. By explicitly formulating scene flow estimation as a matching process, i.e., for each point, identifying its destination in the target point cloud, the model becomes more interpretable compared to regression-based approaches, such as encoder-decoder methods. The key to success is learning high-quality features for each point to make the matching as accurate as possible.

To this end, we employ transformers to learn robust and representative features for individual points. Specifically, we design a local–global–cross transformer paradigm. The local transformer

first captures local features for each point, such as whether it lies on a flat or curved surface, or whether the surrounding point cloud is sparse or dense. Next, the global transformer enables each point to attend to all other points within the point cloud. This is applied separately to the source and target point clouds, allowing each point to be aware of its overall location within the cloud. Finally, a cross transformer performs joint analysis of both point clouds, helping the source points identify the most probable corresponding positions in the target point cloud.

Our experiments demonstrate that both the local transformer and the interleaved global–cross transformers improve performance, highlighting the importance of capturing both local and global information to learn representative features. By learning effective features and formulating scene flow estimation as a matching process, we achieve state-of-the-art performance at the time of publication, outperforming the second-best method by over 50% on the FlyingThings3D dataset.

5.4.2 Paper C

In Paper C [138], building on the work presented in Paper B, we further investigate how to incorporate uncertainty estimation into scene flow prediction. Since scene flow estimation is often applied in safety-critical downstream tasks, equipping the predictions with a measure of confidence is highly valuable.

To model uncertainty, we employ diffusion models, which have gained popularity not only in image and video generation but also in analytical tasks such as classification, detection, and segmentation. The diffusion process begins with the ground truth scene flow vector field and gradually adds noise until only Gaussian noise remains. The reverse process is then learned to recover the scene flow prediction from a randomly sampled noisy vector field. To learn the reverse process, we adopt the architecture introduced in Paper B, with minor modifications. The randomness inherent in the diffusion process facilitates uncertainty estimation during inference without requiring training multiple models. Additionally, as input point clouds can be noisy, the stochastic nature of the diffusion process helps filter out noise, allowing the model to focus on relevant patterns.

Our results show that the estimated uncertainties align well with prediction outliers and correlate positively with the end-point error between the predictions and ground truth. Furthermore, the improved architecture and diffusion-based paradigm enhance both the accuracy and robustness of scene flow predictions, outperforming all other methods at the time of publication.

5.4.3 Paper D

In Paper D, we investigate how to leverage rich temporal information without slowing down prediction or increasing computational complexity. Previous methods primarily focus on estimating the scene flow vector field between two consecutive LiDAR scans, which limits their ability to exploit the richer information available over longer temporal horizons. Our goal is to utilize this temporal information in scene flow estimation while maintaining a lightweight and efficient framework.

We introduce a delta scheme comprising subtraction, temporal weighting, and summation operations that effectively extract compact motion information from voxelized frames. Instead of simply concatenating voxelized point clouds along the feature dimension or stacking them along an additional temporal dimension, our proposed delta scheme operates directly on the voxelized features and introduces significantly less computational overhead than previous ap-

proaches. Combined with a lightweight 3D framework, our method is scalable to longer temporal horizons.

Experiments show that DeltaFlow outperforms all existing methods, both for two-scan-based and multi-scan-based approaches. Thanks to the lightweight architecture, our method runs at approximately 10 FPS on the Waymo dataset, which is substantially faster than other state-of-the-art methods.

6

LiDAR Panoptic Segmentation

We have seen how to perform motion estimation and video object segmentation in 2D images and videos, as well as motion estimation in 3D point clouds. We now extend this discussion to LiDAR panoptic segmentation. With the increasing adoption of LiDAR sensors in applications such as autonomous driving and embodied robotics, research on LiDAR-based computer vision has grown rapidly. One of the most developed areas in this field is LiDAR panoptic segmentation and tracking. In this chapter, our goal is to achieve open-vocabulary panoptic segmentation and tracking of arbitrary objects. We begin with a problem formulation to provide an overview. We then discuss LiDAR Panoptic Segmentation (3D-LPS) in Section 6.2, and its extension to 4D, i.e., simultaneous segmentation and tracking (4D-LPS), in Section 6.3. In Section 6.4, we present how to go beyond pre-defined, closed-class categories to open-vocabulary, zero-shot recognition. Training losses are discussed in Section 6.5. Finally, in Section 6.6, we present our contributions to zero-shot 4D-LPS.

6.1 Problem Formulation

Initially, many studies [42, 69, 140, 30, 80, 106, 110] focused on LiDAR panoptic segmentation on single scans. The objective of these methods is, given a point cloud $\mathbf{P}_t \in \mathbb{R}^{N\times 3}$, to predict both a semantic ID and an instance ID for each point $p \in \mathbf{P}_t$. Subsequently, several works [3, 41, 54, 141, 81, 131] incorporated temporal reasoning and jointly performed segmentation and tracking over LiDAR sequences, a task commonly referred to as 4D LiDAR panoptic segmentation. In this context, each object is expected to maintain a consistent instance ID and semantic ID across the entire sequence $\mathcal{P} = \{\mathbf{P}_t\}_{t=1}^{T}$. More recent studies [92, 89] extended semantic prediction from pre-defined classes to open-vocabulary recognition, wherein the objective is not only to predict semantic IDs but to assign free-form class labels in natural language for each instance. In this chapter, we aim to advance the field by developing the first zero-shot 4D LiDAR panoptic segmentation method. The objective is to assign consistent instance IDs and free-form semantic labels to the same objects throughout an entire LiDAR sequence.

6.2 3D LiDAR Panoptic Segmentation

Early research began with bottom-up methods [42, 69, 140], which perform semantic segmentation followed by clustering as a post-processing step. This was followed by top-down methods [30, 110] that can be trained in an end-to-end manner. More recently, mask-based methods [80, 106] have gained increasing attention. These approaches typically employ a transformer architecture with learnable, randomly initialized queries, where each learned query is used to decode an instance mask and its corresponding semantic label. The overall architecture is illustrated in Figure 6.1.

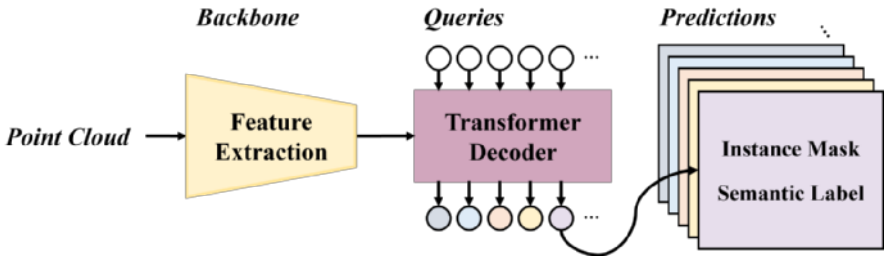

Figure 6.1: The overall architecture of mask-based 3D LiDAR panoptic segmentation methods.

The final prediction is a set of non-overlapping masks $\mathcal{Y} = \{(\mathbf{m}_i, c_i)\}_{i=1}^{K}$, where $\mathbf{m}_i \in \{0,1\}^N$ is the ith predicted binary mask, and $c_i \in \{1, 2, .., L\}$ is the corresponding semantic class for $\mathbf{m}_i$. K is the total number of predicted masks.

6.3 4D LiDAR Panoptic Segmentation

Building up on the development of 3D LiDAR panoptic segmentation methods, many subsequent works [3, 41, 54, 141, 81, 131] try to extend it with temporal dimension, i.e., LiDAR panoptic segmentation and tracking, or 4D LiDAR panoptic segmentation. The most common way of tackling the temporal dimension is to directly accumulate all the LiDAR scans in a limited temporal window to the same coordinate. Then the superimposed point cloud is sent into a neural network to output the 4D instance mask and the corresponding semantic labels. The neural network can be directly taken from the 3D-LPS methods or specifically modified for 4D segmentation. The illustration of the 4D-LPS pipeline is shown in Figure 6.2.

6.4 Zero-shot Recognition in LiDAR

Although substantial progress has been made in 3D-LPS and 4D-LPS, most existing methods remain constrained to a fixed set of predefined classes determined prior to annotation. This limits their ability to recognize a broader spectrum of generic objects when trained on such datasets. Consequently, recent research has shifted towards open-vocabulary, zero-shot recognition of arbitrary objects. This capability is typically achieved through vision–language models [97, 22], which align visual and textual features within a shared embedding space. Such alignment enables semantic prediction directly from visual features, thereby eliminating the need for manual semantic labeling.

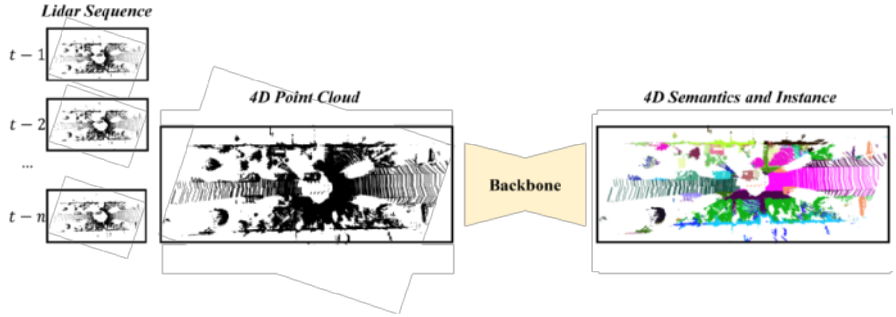

Figure 6.2: An illustration of the 4D-LPS pipeline. Multiple LiDAR scans are first accumulated to form a unified point cloud. The neural network then processes this superimposed point cloud to produce per-point semantic labels and instance IDs.

The most relevant work is SAL [92], which aims to extend 3D-LPS with zero-shot recognition capabilities. Given a point cloud and a set of text prompts, SAL can segment any prompted objects in the scene, including those unseen during training. An overview of the inference process is shown in Figure 6.3.

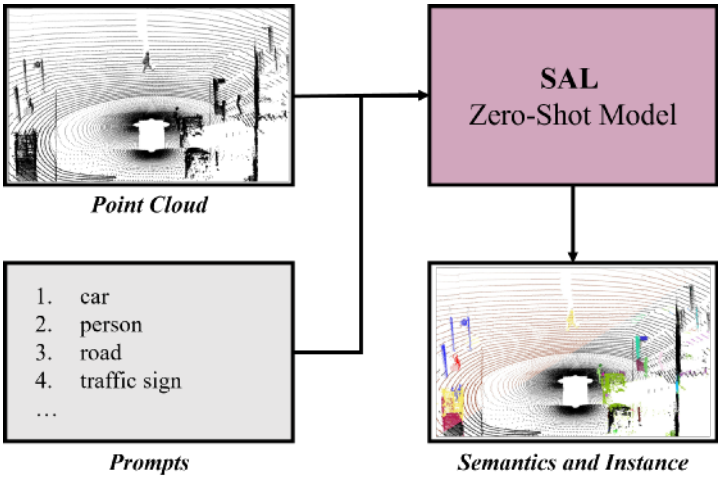

Figure 6.3: An overview of the inference process for SAL (Adapted from SAL [92]).

To enable zero-shot recognition, the SAL model is trained to predict CLIP features for objects, which can subsequently be aligned with text features extracted from prompts during inference. The contributions are twofold: (1) the construction of pseudo-labels to train the SAL zero-shot model, and (2) the design of a model architecture suitable for effective learning from these pseudo-labels.

Pseudo-label Engine The pseudo-label engine takes both the point cloud and the corresponding image as inputs. The Segment Anything Model (SAM) is first applied to the input image to obtain a set of object masks. Each object mask, together with the input image, is processed by the CLIP model [22] to extract a CLIP feature vector for the corresponding object. The 2D masks produced by SAM are then projected into 3D space, with DBSCAN [26] employed to suppress

projection noise. The final annotation format consists of a set of 3D mask–CLIP feature pairs. The overall pseudo-label generation pipeline is illustrated in Figure 6.4.

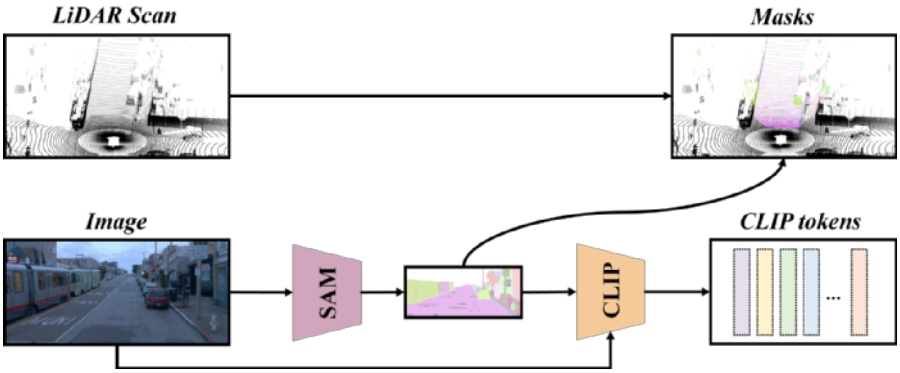

Figure 6.4: The overall pipeline of the SAL pseudo-label engine.

SAL Zero-Shot Model The SAL zero-shot model is built upon prior mask-based 3D-LPS approaches [80, 106]. The primary distinction lies in the training paradigm: whereas previous methods [80, 106] are trained on a fixed, predefined set of semantic classes, SAL [92] enables open-vocabulary recognition of arbitrary objects within the scene. To achieve this, SAL learns to predict CLIP feature embeddings for each segmented mask, allowing alignment with text embeddings during inference. The overall architecture of the SAL zero-shot model is shown in Figure 6.5.

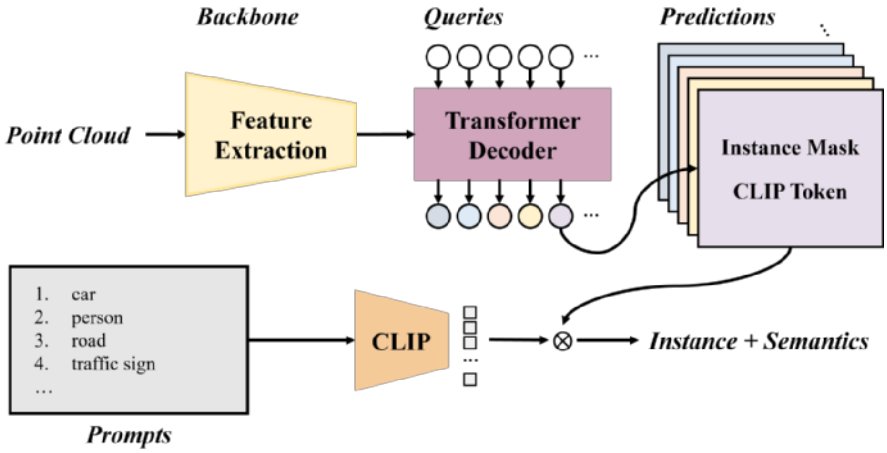

Figure 6.5: The overall architecture of the SAL zero-shot model.

6.5 Training Losses

In this section, we present the loss functions commonly employed in LiDAR panoptic segmentation. These include: 1. Bipartite matching loss: used to associate predicted masks with ground-truth mask annotations. 2. Binary cross-entropy loss and Dice loss: applied to supervise segmentation quality by measuring pixel- or mask-level agreement between predictions and ground truth. 3. Cosine similarity loss: computed between the predicted CLIP embeddings and the annotated CLIP tokens, providing supervision for zero-shot, open-vocabulary recognition.

6.5.1 Bipartite Matching

In transformer-based panoptic segmentation, the output is represented as a set of masks, each associated with semantic and instance identifiers. While traditional per-pixel classification methods can be trained using cross-entropy loss, transformer-based architectures require an additional step: establishing correspondences between predicted and ground-truth masks. MaskFormer [15] addresses this by employing bipartite matching to determine these correspondences. The matching cost is formulated as a weighted sum of the classification loss and the mask loss:

$$\mathbf{C}_{i,j} = -\hat{p}_j(c_i) + \lambda_{\text{dice}}\mathcal{L}_{\text{dice}}(m_i, \hat{m}_j) + \lambda_{\text{ce}}\mathcal{L}_{\text{bce}}(m_i, \hat{m}_j), \tag{6.1}$$

where i and j are the indices of the ground truth mask and the predicted mask, respectively. $\hat{p}_j(c_i)$ denotes the probability that the jth predicted mask belongs to the same class as the ith ground truth mask. The second and third terms correspond to the dice loss and binary cross-entropy loss between the jth predicted mask and the ith ground truth mask. λ_{dice} and λ_{ce} are the weights for the dice loss and binary cross-entropy loss, respectively. Once the matching between the predictions and the ground truth is established, the final loss for training can be computed accordingly.

6.5.2 Segmentation Loss

The segmentation loss typically consists of a binary cross entropy loss and a dice loss. The loss is computed only for the matched pairs obtained from the previous bipartite matching. The binary cross-entropy loss is defined as:

$$\mathcal{L}_{\text{bce}}(y_i, \hat{y}_i) = -\sum_i \left[y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \right], \tag{6.2}$$

The dice loss is defined as:

$$\mathcal{L}_{\text{dice}}(y_i, \hat{y}_i) = 1 - \frac{2 \cdot \sum_i y_i \cdot \hat{y}_i}{\sum_i y_i + \sum_i \hat{y}_i}, \tag{6.3}$$

where y_i and $\hat{y}_i$ denote the ground truth and the predicted value for pixel i, respectively.

6.5.3 Zero-shot Recognition Loss

The zero-shot model predicts the CLIP token embedding for each mask, which can subsequently be aligned with an open class vocabulary. This prediction is supervised by minimizing the cosine distance between the predicted CLIP token and the corresponding pseudo-label [92].

Formally, the loss is defined as:

$$\mathcal{L}_{\text{token}} = 1 - \frac{\mathbf{f} \cdot \hat{\mathbf{f}}}{\|\mathbf{f}\| \cdot \|\hat{\mathbf{f}}\|}, \tag{6.4}$$

where $\mathbf{f}$ and $\hat{\mathbf{f}}$ denote the ground truth and predicted CLIP token embeddings, respectively. The symbol $\cdot$ represents the dot product, and $\|\cdot\|$ denotes the Euclidean norm.

6.6 Contributions

In this section, we investigate how to achieve zero-shot recognition and tracking in the LiDAR domain, i.e., the ability to segment and track arbitrary objects in LiDAR sequences. To the best of our knowledge, no prior work is capable of simultaneously segmenting and tracking all objects in a scene. Existing approaches are either restricted to a closed set of predefined classes [88] or lack the capacity for temporal reasoning [92]. We address this challenge in Section 6.6.1, where we build upon the zero-shot 3D-LPS framework [92] and extend its capabilities to the temporal domain.

6.6.1 Paper E

In Paper E [136], we present the first study on zero-shot 4D LiDAR panoptic segmentation and explore several potential approaches to address this task, which paves the way for our proposed SAL-4D model. SAL-4D leverages annotations derived from a vision foundation model [99] and a vision-language model [22] in the image domain, which, when distilled into the LiDAR domain, enables segmentation and tracking of arbitrary objects.

Our approach comprises two main components: a *pseudo-label engine* for generating training data, and the *SAL-4D model* trained on these labels. For the pseudo-label engine, we employ the SAM 2 model [99] to segment and track objects in video sequences, producing temporally consistent 2D masks. MaskCLIP [22] is then used to extract CLIP features for each mask, providing semantic information. The 2D masks are lifted to 3D following the strategy in SAL [92]. The resulting annotations consist of 4D masklets and their associated CLIP features. For the SAL-4D model, we integrate techniques from Section 6.3 and Section 6.4. The model takes a superimposed point cloud as input and outputs a set of 4D masklets with CLIP features.

Our findings are as follows: (1) Although pseudo-labels contain noise and errors, they are sufficiently decorrelated to allow effective distillation of useful information into the SAL-4D zero-shot model. (2) The proposed SAL-4D model, despite being trained solely on pseudo-labels, achieves over 70% of the performance of fully supervised methods. (3) The temporal consistency introduced by SAL-4D improves single-scan segmentation quality for both the pseudo-labels and the zero-shot model.

7

Conclusions

This thesis investigates spatiotemporal learning in both the 2D image domain and the 3D LiDAR domain, with a particular emphasis on motion estimation and visual recognition. We begin by presenting the spatiotemporal learning toolbox employed throughout this work, which includes Transformers, Diffusion Models, the Segment Anything Model (SAM), Minkowski U-Net, and CLIP. Subsequent chapters provide detailed introductions to motion estimation in the 2D image domain, visual recognition in the 2D image domain, motion estimation in the 3D LiDAR domain, and visual recognition in the 3D LiDAR domain, thereby situating our contributions within their respective research contexts.

In Paper A, we address visual recognition in the image domain and investigate how optical flow can be leveraged to enhance video object segmentation performance. Following matching-based SVOS approaches, we integrate optical flow features into both the target model and the decoder network. Motion information from optical flow and appearance information from image features are fused using an attention mechanism to construct more robust object representations. This joint representation leads to improved segmentation accuracy, particularly in terms of contour precision and model generalization.

In Paper B, we focus on scene flow estimation from LiDAR point clouds. Building on soft correspondence-based approaches, we propose a local-global-cross Transformer architecture to learn discriminative point features for correspondence search. The local Transformer captures local geometric properties such as curvature and density, while the interleaved global-cross Transformers encode global spatial context with respect to both point clouds. By jointly integrating local and global information, the model achieves more accurate correspondence matching and higher estimation precision.

In Paper C, we extend the framework of Paper B by equipping the scene flow prediction model with uncertainty estimation. This is achieved through diffusion models. The stochastic nature of diffusion models can help with indicating estimation uncertainty without having to train the models multiple times. During training, the network learns to predict the scene flow vector field from randomly sampled noisy vector fields. To enable the uncertainty estimation, we perform multiple inferences with different initial random vector fields. The variance of the multiple predictions correlates positively with the endpoint error between the predictions and the ground truths, indicating the effectiveness of the proposed method.

In Paper D, we advance scene flow estimation to the multi-frame setting. We follow the encoder-decoder scene flow estimation methods and employ the Minkowski U-Net as the backbone for spatiotemporal feature extraction. To process the temporal information efficiently, we introduce

a novel delta scheme comprising subtraction, temporal weighting, and summation operations that efficiently encodes motion information across frames while maintaining a fixed feature dimension, thus avoiding the computational overhead typically incurred by multi-frame architectures. The proposed method significantly improves estimation accuracy while remaining computationally efficient compared to state-of-the-art approaches.

In Paper E, we propose the first zero-shot 4D LiDAR panoptic segmentation (4D-LPS) framework for arbitrary object recognition and tracking in autonomous driving scenarios. Our approach builds upon advances in 4D-LPS and zero-shot 3D-LPS, and consists of two core components. First, a pseudo-label engine leverages the vision foundation model, Segment Anything 2 model, and the vision-language model, CLIP, to extract semantically rich information from the image domain, which is subsequently lifted into the LiDAR domain. Second, the distilled pseudo-labels are used to supervise our SAL-4D model, which is capable of segmenting and temporally tracking arbitrary objects in LiDAR sequences in a zero-shot manner.

7.1 Limitations and Future Work

Based on the work presented in this thesis, several limitations remain, and multiple directions exist for future exploration.

In Paper A, the optical flow used for learning the combined feature representation is predicted using a pretrained optical flow model, RAFT. Consequently, the performance of the SVOS method depends on external models trained on different datasets. However, a domain gap typically exists between the training datasets of optical flow and object segmentation, which can hamper segmentation performance when relying on such models. A promising research direction would be to investigate self-supervised optical flow methods that can be trained directly on the same datasets as the SVOS models. This approach could mitigate the domain gap and remove the reliance on additional annotated data.

In Paper B, the proposed scene flow estimation method relies on transformers to extract high-quality features for correspondence search. Currently, these transformers are point-based, which is computationally expensive and becomes prohibitively slow as the number of points increases. A potential solution is to voxelize the point cloud prior to correspondence search, thereby grouping dense points into sparser voxels. This reduces the computational complexity, ensuring that the cost of correspondence search does not scale quadratically with the number of points.

In Paper C, the uncertainty estimation of scene flow prediction is achieved via diffusion models. While this approach avoids retraining the model multiple times, it remains ensemble-based and requires multiple inferences, which can be time-consuming and impractical for real-time applications. A possible future direction is to develop one-pass methods capable of jointly estimating both the scene flow vector field and the corresponding uncertainty in a single inference step.

In Paper D, the proposed method aims to process multiple frames efficiently for scene flow estimation. Current findings indicate that optimal performance is obtained with 10 frames, and incorporating additional frames does not yield further improvement. A potential direction for future research is to investigate the underlying cause of this limitation, either to exploit additional temporal information for enhanced accuracy or to provide a theoretical explanation for why longer temporal horizons fail to improve performance.

In Paper E, the pseudo-label engine is constructed using the vision foundation model SAM 2 and the vision–language model CLIP. Although the labeling process eliminates human annotation effort, the quality of pseudo labels remains limited by the capabilities of the underlying

models. SAM 2 generally performs well in segmenting objects, but it is primarily designed for objects with clear boundaries, making it less suitable for ambiguous structures such as trees. Additionally, semantic annotation relies on CLIP, which is still prone to errors in certain cases. Future work can therefore focus on addressing these issues to improve the quality and reliability of the generated pseudo labels.

BIBLIOGRAPHY

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. "Flamingo: a visual language model for few-shot learning". In: *Advances in neural information processing systems* 35 (2022), pp. 23716–23736 (p. 14).

[2] 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*. 2015, pp. 2425–2433 (p. 15).

[3] Mehmet Aygun, Aljosa Osep, Mark Weber, Maxim Maximov, Cyrill Stachniss, Jens Behley, and Laura Leal-Taixé. "4d panoptic lidar segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 5527–5537 (pp. 41, 42).

[4] Shaojie Bai, Zhengyang Geng, Yash Savani, and J Zico Kolter. "Deep equilibrium optical flow estimation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 620–630 (p. 21).

[5] Simon Baker, Daniel Scharstein, James P Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. "A database and evaluation methodology for optical flow". In: *International journal of computer vision* 92 (2011), pp. 1–31 (pp. 19, 22).

[6] Dmitry Baranchuk, Andrey Voynov, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. "Label-Efficient Semantic Segmentation with Diffusion Models". In: *International Conference on Learning Representations*. 2022 (p. 13).

[7] Aseem Behl, Omid Hosseini Jafari, Siva Karthik Mustikovela, Hassan Abu Alhaija, Carsten Rother, and Andreas Geiger. "Bounding boxes, segmentations and object coordinates: How important is recognition

for 3d scene flow estimation in autonomous driving scenarios?" In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2574–2583 (p. 34).

[8] Goutam Bhat, Felix Järemo Lawin, Martin Danelljan, Andreas Robinson, Michael Felsberg, Luc Van Gool, and Radu Timofte. "Learning what to learn for video object segmentation". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer. 2020, pp. 777–794 (p. 28).

[9] 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". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901 (p. 12).

[10] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. "A naturalistic open source movie for optical flow evaluation". In: *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12*. Springer. 2012, pp. 611–625 (pp. 19, 20, 22).

[11] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. "One-shot video object segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 221–230 (p. 26).

[12] 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*. Springer. 2020, pp. 213–229 (p. 12).

[13] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. "Pre-trained image processing transformer". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 12299–12310 (p. 12).

[14] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. "Diffusiondet: Diffusion model for object detection". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2023, pp. 19830–19843 (p. 13).

[15] Bowen Cheng, Alex Schwing, and Alexander Kirillov. "Per-pixel classification is not all you need for semantic segmentation". In: *Advances in neural information processing systems* 34 (2021), pp. 17864–17875 (pp. 12, 45).

[16] Jingchun Cheng, Yi-Hsuan Tsai, Wei-Chih Hung, Shengjin Wang, and Ming-Hsuan Yang. "Fast and accurate online video object segmentation via tracking parts". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7415–7424 (p. 28).

[17] Wencan Cheng and Jong Hwan Ko. "Bi-pointflownet: Bidirectional learning for point cloud based scene flow estimation". In: *European Conference on Computer Vision*. Springer. 2022, pp. 108–124 (p. 34).

[18] Christopher Choy, JunYoung Gwak, and Silvio Savarese. "4d spatio-temporal convnets: Minkowski convolutional neural networks". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 3075–3084 (p. 13).

[19] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. "Mot20: A benchmark for multi object tracking in crowded scenes". In: *arXiv preprint arXiv:2003.09003* (2020) (p. 25).

[20] 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*. Ieee. 2009, pp. 248–255 (p. 26).

[21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 2019, pp. 4171–4186 (p. 12).

[22] Zheng Ding, Jieke Wang, and Zhuowen Tu. "Open-vocabulary universal image segmentation with MaskCLIP". In: *Proceedings of the 40th International Conference on Machine Learning*. 2023, pp. 8090–8102 (pp. 42, 43, 46).

[23] 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". In: *arXiv preprint arXiv:2010.11929* (2020) (p. 12).

[24] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. "Flownet: Learning optical flow with convolutional networks". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2758–2766 (pp. 19, 22, 34).

[25] Abdelrahman Eldesokey and Michael Felsberg. "Normalized convolution upsampling for refined optical flow estimation". In: *arXiv preprint arXiv:2102.06979* (2021) (p. 21).

[26] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. "A density-based algorithm for discovering clusters in large spatial databases with noise". In: *kdd*. Vol. 96. 34. 1996, pp. 226–231 (p. 43).

[27] Gunnar Farnebäck. "Two-frame motion estimation based on polynomial expansion". In: *Image Analysis: 13th Scandinavian Conference, SCIA 2003 Halmstad, Sweden, June 29–July 2, 2003 Proceedings 13*. Springer. 2003, pp. 363–370 (pp. 18, 19).

[28] Michael Felsberg. "Visual tracking: Tracking in scenes containing multiple moving objects". In: *Advanced Methods and Deep Learning in Computer Vision*. Elsevier, 2022, pp. 305–336 (p. 25).

[29] Michael Felsberg, Amanda Berg, Gustav Hager, Jorgen Ahlberg, Matej Kristan, Jiri Matas, Ales Leonardis, Luka Cehovin, Gustavo Fernandez, Tomas Vojir, et al. "The thermal infrared visual object tracking VOT-TIR2015 challenge results". In: *Proceedings of the ieee international conference on computer vision workshops*. 2015, pp. 76–88 (p. 25).

[30] Stefano Gasperini, Mohammad-Ali Nikouei Mahani, Alvaro Marcos-Ramiro, Nassir Navab, and Federico Tombari. "Panoster: End-to-end panoptic segmentation of lidar point clouds". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 3216–3223 (pp. 41, 42).

[31] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. "Vision meets robotics: The kitti dataset". In: *The international journal of robotics research* 32.11 (2013), pp. 1231–1237 (pp. 19, 20, 22).

[32] Ross Girshick. "Fast r-cnn". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448 (p. 12).

[33] Ben Graham. "Sparse 3D convolutional neural networks". In: *arXiv preprint arXiv:1505.02890* (2015) (p. 14).

[34] Benjamin Graham. "Spatially-sparse convolutional neural networks". In: *arXiv preprint arXiv:1409.6070* (2014) (p. 14).

[35] Xiuye Gu, Yijie Wang, Chongruo Wu, Yong Jae Lee, and Panqu Wang. "Hplflownet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 3254–3263 (p. 34).

[36] Zhangxuan Gu, Haoxing Chen, and Zhuoer Xu. "Diffusioninst: Diffusion model for instance segmentation". In: *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2024, pp. 2730–2734 (p. 13).

[37] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. "Pct: Point cloud transformer". In: *Computational visual media* 7 (2021), pp. 187–199 (p. 12).

[38] Xizewen Han, Huangjie Zheng, and Mingyuan Zhou. "Card: Classification and regression diffusion models". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 18100–18115 (p. 13).

[39] 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*. 2016, pp. 770–778 (p. 12).

[40] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models". In: *Advances in neural information processing systems* 33 (2020), pp. 6840–6851 (p. 12).

[41] Fangzhou Hong, Lingdong Kong, Hui Zhou, Xinge Zhu, Hongsheng Li, and Ziwei Liu. "Unified 3d and 4d panoptic segmentation via dynamic shifting networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.5 (2024), pp. 3480–3495 (pp. 41, 42).

[42] Fangzhou Hong, Hui Zhou, Xinge Zhu, Hongsheng Li, and Ziwei Liu. "Lidar-based panoptic segmentation via dynamic shifting network". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 13090–13099 (pp. 41, 42).

[43] Berthold KP Horn and Brian G Schunck. "Determining optical flow". In: *Artificial intelligence* 17.1-3 (1981), pp. 185–203 (p. 18).

[44] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. "Flowformer: A transformer architecture for optical flow". In: *European conference on computer vision*. Springer. 2022, pp. 668–685 (p. 21).

[45] 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*. 2017, pp. 2462–2470 (pp. 19, 20, 34).

[46] Varun Jampani, Raghudeep Gadde, and Peter V Gehler. "Video propagation networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 451–461 (p. 26).

[47] Haobo Jiang, Mathieu Salzmann, Zheng Dang, Jin Xie, and Jian Yang. "Se (3) diffusion model-based point cloud registration for robust 6d object pose estimation". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 21285–21297 (p. 13).

[48] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. "Learning to estimate hidden motions with global motion aggregation". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 9772–9781 (p. 21).

[49] Joakim Johnander, Martin Danelljan, Emil Brissman, Fahad Shahbaz Khan, and Michael Felsberg. "A generative appearance model for end-to-end video object segmentation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 8953–8962 (p. 28).

[50] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. "Panoptic segmentation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 9404–9413 (p. 12).

[51] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. "Segment anything". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2023, pp. 4015–4026 (p. 14).

[52] Yair Kittenplon, Yonina C Eldar, and Dan Raviv. "Flowstep3d: Model unrolling for self-supervised scene flow estimation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 4114–4123 (p. 35).

[53] Daniel Kondermann, Rahul Nair, Katrin Honauer, Karsten Krispin, Jonas Andrulis, Alexander Brock, Burkhard Gussefeld, Mohsen Rahimimoghaddam, Sabine Hofmann, Claus Brenner, et al. "The hci benchmark suite: Stereo and flow ground truth with uncertainties for urban autonomous driving". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2016, pp. 19–28 (pp. 19, 22).

[54] Lars Kreuzberg, Idil Esen Zulfikar, Sabarinath Mahadevan, Francis Engelmann, and Bastian Leibe. "4d-stop: Panoptic segmentation of 4d lidar using spatio-temporal object proposal generation and aggregation". In: *European Conference on Computer Vision*. Springer. 2022, pp. 537–553 (pp. 41, 42).

[55] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka ˇCehovin Zajc, Tomas Vojir, Goutam Bhat, Alan Lukezic, Abdelrahman Eldesokey, et al. "The sixth visual object tracking vot2018 challenge results". In: *Proceedings of the European conference on computer vision (ECCV) workshops*. 2018, pp. 0–0 (p. 25).

[56] Matej Kristan, Aleš Leonardis, Jiří Matas, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kämäräinen, Hyung Jin Chang, Martin Danelljan, Luka Čehovin Zajc, Alan Lukežič, et al. "The tenth visual object tracking vot2022 challenge results". In: *European Conference on Computer Vision*. Springer. 2022, pp. 431–460 (p. 25).

[57] Matej Kristan, Aleš Leonardis, Jiří Matas, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kämäräinen, Martin Danelljan, Luka Čehovin Zajc, Alan Lukežič, Ondrej Drbohlav, et al. "The eighth visual object tracking VOT2020 challenge results". In: *European conference on computer vision*. Springer. 2020, pp. 547–601 (p. 25).

[58] Matej Kristan, Jiri Matas, Ales Leonardis, Michael Felsberg, Luka Cehovin, Gustavo Fernandez, Tomas Vojir, Gustav Hager, Georg Nebehay, and Roman Pflugfelder. "The visual object tracking vot2015 challenge results". In: *Proceedings of the IEEE international conference on computer vision workshops*. 2015, pp. 1–23 (p. 25).

[59] Matej Kristan, Jiri Matas, Ales Leonardis, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kamarainen, Luka ˇCehovin Zajc, Ondrej Drbohlav, Alan Lukezic, Amanda Berg, et al. "The seventh visual object tracking vot2019 challenge results". In: *Proceedings of the IEEE/CVF international conference on computer vision workshops*. 2019, pp. 0–0 (p. 25).

[60] Matej Kristan, Jiří Matas, Martin Danelljan, Michael Felsberg, Hyung Jin Chang, Luka Čehovin Zajc, Alan Lukežič, Ondrej Drbohlav, Zhongqun Zhang, Khanh-Tung Tran, et al. "The first visual object tracking segmentation vots2023 challenge results". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 1796–1818 (p. 25).

[61] Matej Kristan, Jiří Matas, Aleš Leonardis, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kämäräinen, Hyung Jin Chang, Martin Danelljan, Luka Cehovin, Alan Lukežič, et al. "The ninth visual object tracking vot2021 challenge results". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 2711–2738 (p. 25).

[62] Matej Kristan, Jiří Matas, Pavel Tokmakov, Michael Felsberg, Luka Čehovin Zajc, Alan Lukežič, Khanh-Tung Tran, Xuan-Son Vu, Johanna Björklund, Hyung Jin Chang, et al. "The second visual object tracking segmentation VOTS2024 challenge results". In: *European Conference on Computer Vision*. Springer. 2024, pp. 357–383 (p. 25).

[63] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012) (p. 12).

[64] Zihang Lai, Erika Lu, and Weidi Xie. "Mast: A memory-augmented self-supervised tracker". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 6479–6488 (p. 27).

[65] Itai Lang, Dror Aiger, Forrester Cole, Shai Avidan, and Michael Rubinstein. "Scoop: Self-supervised correspondence and optimization-based scene flow". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 5281–5290 (p. 36).

[66] Laura Leal-Taixé, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler. "Motchallenge 2015: Towards a benchmark for multi-target tracking". In: *arXiv preprint arXiv:1504.01942* (2015) (p. 25).

[67] Mengqi Lei, Siqi Li, Yihong Wu, Han Hu, You Zhou, Xinhu Zheng, Guiguang Ding, Shaoyi Du, Zongze Wu, and Yue Gao. "YOLOv13: Real-Time Object Detection with Hypergraph-Enhanced Adaptive Visual Perception". In: *arXiv preprint arXiv:2506.17733* (2025) (p. 12).

[68] Bing Li, Cheng Zheng, Silvio Giancola, and Bernard Ghanem. "Sctn: Sparse convolution-transformer network for scene flow estimation". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 36. 2. 2022, pp. 1254–1262 (p. 35).

[69] Jinke Li, Xiao He, Yang Wen, Yuan Gao, Xiaoqiang Cheng, and Dan Zhang. "Panoptic-phnet: Towards real-time and high-precision lidar panoptic segmentation via clustering pseudo heatmap". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 11809–11818 (pp. 41, 42).

[70] Ruibo Li, Guosheng Lin, Tong He, Fayao Liu, and Chunhua Shen. "Hcrf-flow: Scene flow from point clouds with continuous high-order crfs and position-aware flow embedding". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 364–373 (p. 34).

[71] Xueqian Li, Jhony Kaesemodel Pontes, and Simon Lucey. "Neural scene flow prior". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 7838–7851 (p. 36).

[72] Xueqian Li, Jianqiao Zheng, Francesco Ferroni, Jhony Kaesemodel Pontes, and Simon Lucey. "Fast neural scene flow". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 9878–9890 (p. 36).

[73] Xueting Li, Sifei Liu, Shalini De Mello, Xiaolong Wang, Jan Kautz, and Ming-Hsuan Yang. "Joint-task self-supervised learning for temporal correspondence". In: *Advances in Neural Information Processing Systems* 32 (2019) (p. 27).

[74] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. "Flownet3d: Learning scene flow in 3d point clouds". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 529–537 (p. 34).

[75] 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*. 2021, pp. 10012–10022 (p. 12).

[76] Xiankai Lu, Wenguan Wang, Jianbing Shen, Yu-Wing Tai, David J Crandall, and Steven CH Hoi. "Learning video object segmentation from unlabeled videos". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 8960–8970 (p. 27).

[77] Bruce D Lucas and Takeo Kanade. "An iterative image registration technique with an application to stereo vision". In: *IJCAI'81: 7th international joint conference on Artificial intelligence*. Vol. 2. 1981, pp. 674–679 (p. 18).

[78] Wei-Chiu Ma, Shenlong Wang, Rui Hu, Yuwen Xiong, and Raquel Urtasun. "Deep rigid instance scene flow". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 3614–3622 (p. 34).

[79] K-K Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. "Video object segmentation without temporal information". In: *IEEE transactions on pattern analysis and machine intelligence* 41.6 (2018), pp. 1515–1530 (p. 26).

[80] Rodrigo Marcuzzi, Lucas Nunes, Louis Wiesmann, Jens Behley, and Cyrill Stachniss. "Mask-based panoptic lidar segmentation for autonomous driving". In: *IEEE Robotics and Automation Letters* 8.2 (2023), pp. 1141–1148 (pp. 41, 42, 44).

[81] Rodrigo Marcuzzi, Lucas Nunes, Louis Wiesmann, Elias Marks, Jens Behley, and Cyrill Stachniss. "Mask4D: end-to-end mask-based 4D panoptic segmentation for lidar sequences". In: *IEEE Robotics and Automation Letters* 8.11 (2023), pp. 7487–7494 (pp. 41, 42).

[82] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 4040–4048 (pp. 19, 20, 22, 34, 36).

[83] Lukas Mehl, Jenny Schmalfuss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn. "Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 4981–4991 (pp. 19, 20, 22).

[84] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. "Trackformer: Multi-object tracking with transformers". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 8844–8854 (p. 12).

[85] Simon Meister, Junhwa Hur, and Stefan Roth. "Unflow: Unsupervised learning of optical flow with a bidirectional census loss". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1. 2018 (p. 22).

[86] Moritz Menze and Andreas Geiger. "Object scene flow for autonomous vehicles". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3061–3070 (pp. 19, 20, 22, 34, 36).

[87] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. "MOT16: A benchmark for multi-object tracking". In: *arXiv preprint arXiv:1603.00831* (2016) (p. 25).

[88] Frank Moosmann and Christoph Stiller. "Joint self-localization and tracking of generic objects in 3D range data". In: *2013 IEEE International Conference on Robotics and Automation*. IEEE. 2013, pp. 1146–1152 (p. 46).

[89] Mahyar Najibi, Jingwei Ji, Yin Zhou, Charles R Qi, Xinchen Yan, Scott Ettinger, and Dragomir Anguelov. "Unsupervised 3d perception with 2d vision-language distillation for autonomous driving". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 8602–8612 (p. 41).

[90] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. "Video object segmentation using space-time memory networks". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 9226–9235 (p. 28).

[91] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. "Dinov2: Learning robust visual features without supervision". In: *arXiv preprint arXiv:2304.07193* (2023) (p. 14).

[92] Aljoša Ošep, Tim Meinhardt, Francesco Ferroni, Neehar Peri, Deva Ramanan, and Laura Leal-Taixé. "Better call sal: Towards learning to segment anything in lidar". In: *European Conference on Computer Vision*. Springer. 2024, pp. 71–90 (pp. 41, 43–46).

[93] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. "Image transformer". In: *International conference on machine learning*. PMLR. 2018, pp. 4055–4064 (p. 12).

[94] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. "Learning video object segmentation from static images". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2663–2672 (p. 26).

[95] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. "A benchmark dataset and evaluation methodology for video object segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 724–732 (p. 26).

[96] Gilles Puy, Alexandre Boulch, and Renaud Marlet. "Flot: Scene flow on point clouds guided by optimal transport". In: *European conference on computer vision*. Springer. 2020, pp. 527–544 (p. 35).

[97] 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*. PmLR. 2021, pp. 8748–8763 (pp. 14, 15, 42).

[98] Alexander Raistrick, Lahav Lipson, Zeyu Ma, Lingjie Mei, Mingzhe Wang, Yiming Zuo, Karhan Kayan, Hongyu Wen, Beining Han, Yihan Wang, et al. "Infinite photorealistic worlds using procedural generation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 12630–12641 (pp. 19, 22).

[99] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. "Sam 2: Segment anything in images and videos". In: *arXiv preprint arXiv:2408.00714* (2024) (pp. 14, 46).

[100] Zhile Ren, Deqing Sun, Jan Kautz, and Erik Sudderth. "Cascaded scene flow prediction using semantic segmentation". In: *2017 International Conference on 3D Vision (3DV)*. IEEE. 2017, pp. 225–233 (p. 34).

[101] Stephan R Richter, Zeeshan Hayder, and Vladlen Koltun. "Playing for benchmarks". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2213–2222 (pp. 19, 22).

[102] Andreas Robinson, Felix Jaremo Lawin, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. "Learning fast and robust target models for video object segmentation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 7406–7415 (pp. 28, 29, 31).

[103] Giorgio Roffo, Simone Melzi, et al. "The visual object tracking vot2016 challenge results". In: *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II*. Springer International Publishing. 2016, pp. 777–823 (p. 25).

[104] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. "High-resolution image synthesis with latent diffusion models". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 10684–10695 (p. 12).

[105] Saurabh Saxena, Charles Herrmann, Junhwa Hur, Abhishek Kar, Mohammad Norouzi, Deqing Sun, and David J Fleet. "The surprising effectiveness of diffusion models for optical flow and monocular depth estimation". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 39443–39469 (p. 13).

[106] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. "Mask3d: Mask transformer for 3d semantic instance segmentation". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 8216–8223 (pp. 41, 42, 44).

[107] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. "Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2018, pp. 2556–2565 (p. 15).

[108] Yaqi Shen, Le Hui, Jin Xie, and Jian Yang. "Self-supervised 3d scene flow estimation guided by superpoints". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 5271–5280 (p. 36).

[109] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014) (p. 12).

[110] Kshitij Sirohi, Rohit Mohan, Daniel Büscher, Wolfram Burgard, and Abhinav Valada. "Efficientlps: Efficient lidar panoptic segmentation". In: *IEEE Transactions on Robotics* 38.3 (2021), pp. 1894–1914 (pp. 41, 42).

[111] Leonhard Sommer, Philipp Schröppel, and Thomas Brox. "Sf2se3: Clustering scene flow into se (3)-motions via proposal and selection". In: *DAGM German Conference on Pattern Recognition*. Springer. 2022, pp. 215–229 (p. 34).

[112] Xiuchao Sui, Shaohua Li, Xue Geng, Yan Wu, Xinxing Xu, Yong Liu, Rick Goh, and Hongyuan Zhu. "Craft: Cross-attentional flow transformer for robust optical flow". In: *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*. 2022, pp. 17602–17611 (p. 21).

[113] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8934–8943 (pp. 21, 34).

[114] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. "LoFTR: Detector-free local feature matching with transformers". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 8922–8931 (p. 12).

[115] 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*. 2020, pp. 2446–2454 (p. 36).

[116] Zachary Teed and Jia Deng. "Raft-3d: Scene flow using rigid-motion embeddings". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 8375–8384 (p. 35).

[117] Zachary Teed and Jia Deng. "Raft: Recurrent all-pairs field transforms for optical flow". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer. 2020, pp. 402–419 (pp. 21, 34).

[118] Parker Allen Tew. "An investigation of sparse tensor formats for tensor libraries". PhD thesis. Massachusetts Institute of Technology, 2016 (p. 14).

[119] Yunjie Tian, Qixiang Ye, and David Doermann. "Yolov12: Attention-centric real-time object detectors". In: *arXiv preprint arXiv:2502.12524* (2025) (p. 12).

[120] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017) (pp. 11, 12).

[121] Christoph Vogel, Konrad Schindler, and Stefan Roth. "3d scene flow estimation with a piecewise rigid scene model". In: *International Journal of Computer Vision* 115 (2015), pp. 1–28 (p. 34).

[122] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. "Feelvos: Fast end-to-end embedding learning for video object segmentation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 9481–9490 (p. 28).

[123] Xiaolong Wang, Allan Jabri, and Alexei A Efros. "Learning correspondence from the cycle-consistency of time". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 2566–2576 (p. 27).

[124] Yihan Wang, Lahav Lipson, and Jia Deng. "Sea-raft: Simple, efficient, accurate raft for optical flow". In: *European Conference on Computer Vision*. Springer. 2024, pp. 36–54 (p. 21).

[125] Yi Wei, Ziyi Wang, Yongming Rao, Jiwen Lu, and Jie Zhou. "Pv-raft: Point-voxel correlation fields for scene flow estimation of point clouds". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 6954–6963 (p. 35).

[126] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. "Argoverse 2: Next generation datasets for self-driving perception and forecasting". In: *arXiv preprint arXiv:2301.00493* (2023) (p. 36).

[127] Wenxuan Wu, Zhiyuan Wang, Zhuwen Li, Wei Liu, and Li Fuxin. "Pointpwc-net: A coarse-to-fine network for supervised and self-supervised scene flow estimation on 3d point clouds". In: *arXiv preprint arXiv:1911.12408* (2019) (p. 34).

[128] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. "Gmflow: Learning optical flow via global matching". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 8121–8130 (p. 20).

[129] Fuzhi Yang, Huan Yang, Jianlong Fu, Hongtao Lu, and Baining Guo. "Learning texture transformer network for image super-resolution". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 5791–5800 (p. 12).

[130] Gengshan Yang and Deva Ramanan. "Learning to segment rigid motions from two frames". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 1266–1275 (p. 34).

[131] Kadir Yilmaz, Jonas Schult, Alexey Nekrasov, and Bastian Leibe. "Mask4former: Mask transformer for 4d panoptic segmentation". In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2024, pp. 9418–9425 (pp. 41, 42).

[132] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions". In: *Transactions of the association for computational linguistics* 2 (2014), pp. 67–78 (p. 15).

[133] Jason J Yu, Adam W Harley, and Konstantinos G Derpanis. "Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness". In: *European conference on computer vision*. Springer. 2016, pp. 3–10 (p. 22).

[134] Qingwen Zhang, Xiaomeng Zhu, Yushan Zhang, Yixi Cai, Olov Andersson, and Patric Jensfelt. "DeltaFlow: An Efficient Multi-frame Scene Flow Estimation Method". In: *arXiv preprint arXiv:2508.17054* (2025) (pp. 8, 14).

[135] Yushan Zhang, Johan Edstedt, Bastian Wandt, Per-Erik Forssén, Maria Magnusson, and Michael Felsberg. "GMSF: Global Matching Scene Flow". In: *Advances in Neural Information Processing Systems* 36 (2023) (pp. 6, 12, 37).

[136] Yushan Zhang, Aljoša Ošep, Laura Leal-Taixé, and Tim Meinhardt. "Zero-Shot 4D Lidar Panoptic Segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2025). © 2025 IEEE. Reprinted, with permission, from the source. (pp. 9, 14, 15, 46).

[137] Yushan Zhang, Andreas Robinson, Maria Magnusson, and Michael Felsberg. "Leveraging Optical Flow Features for Higher Generalization Power in Video Object Segmentation". In: *2023 IEEE International Conference on Image Processing (ICIP)*. © 2023 IEEE. Reprinted, with permission, from the source. IEEE. 2023 (pp. 5, 12, 31).

[138] Yushan Zhang, Bastian Wandt, Maria Magnusson, and Michael Felsberg. "DiffSF: Diffusion Models for Scene Flow Estimation". In: *Advances in Neural Information Processing Systems* 37 (2024) (pp. 7, 13, 38).

[139] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. "Point transformer". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 16259–16268 (p. 12).

[140] Zixiang Zhou, Yang Zhang, and Hassan Foroosh. "Panoptic-polarnet: Proposal-free lidar point cloud panoptic segmentation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 13194–13203 (pp. 41, 42).

[141] Minghan Zhu, Shizhong Han, Hong Cai, Shubhankar Borse, Maani Ghaffari, and Fatih Porikli. “4d panoptic segmentation as invariant and equivariant field prediction”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 22488–22498 (pp. 41, 42).

Part II

Publications

Papers

The papers associated with this thesis have been removed for copyright reasons. For more details about these see:

https://doi.org/10.3384/9789181182323

FACULTY OF SCIENCE AND ENGINEERING

Linköping Studies in Science and Technology, Dissertation No. 2476, 2025
Department of Electrical Engineering

Linköping University
SE-581 83 Linköping, Sweden

www.liu.se