Towards Robust Vision for Self-Supervised Depth and Motion Estimation



Madhu Vankadari

Wolfson College

University of Oxford

A thesis submitted for the degree of Doctor of Philosophy

Summer, 2024

Acknowledgements

Words fall short of expressing the depth of my gratitude to my supervisors, Prof. Niki Trigoni and Prof. Andrew Markham. I am truly fortunate to have their guidance, support, and encouragement to think beyond, without which none of this would have happened. Countless discussions, they always had time, patience, and knowledge. I cannot thank them enough for how instrumental they have been in shaping what I have become.

I am also deeply grateful and privileged to have worked with the most brilliant minds: my co-authors, Sangyun Shin, Sourav Garg, Stuart Golodetz, Samuel Hodgson, and Kaichen Zhou. Your contributions to making my ideas a reality and insights into strengthening our work are invaluable. I also must apologize and thank you for all the sleepless nights you had to spend because of me for our publications. I also sincerely appreciate my other lab members: Jialu Wang, Kai Lu, Jiaxing Zhonng, and Vít Růžička, for all the stimulating discussions, coffee chats, and pub crawls.

Living ten thousand miles away from home would not have been easy if it were not for Wolfson College and my Wolfson family. I cannot express in words how thankful I am to my brotherhood with Samuel Wiese, Frederik Florenz, and Tejo Jehart. They were always there for me during my ups and downs throughout my PhD. This family extends to the most intelligent, compassionate and kind people: Botos Csabi, Francisco Navarro Rosales, Asang Wankhede, Dattavya Aggarwal, Theodor Borrmann, Wan Ting, Celia Hanna, Imogen Bridges, Shubham Deshpande, Caroline Herlin, Alexandra Blackwell, Ayaka Shinozaki, Mei Takehana, Sekoah Kwon-Salkin, Raphael Pietzsch Daniel Poh, Estelle Paulus, Oana Diaconescu, Joel Bellviure, Paulo DeSouza, Shrenik Rao, Sarah Scripps, Sisi Qu, Juliana Mota, Lucho Hildebrandt, Peter Waldert, Rutuja C, Divya Vatsa, Sylvia Gattas, Nilgoun Bahar, Anna Ren, Johannes van de Wetering, Peter Blowfield, Richie Yat Long Lo, and Koichi Noguchi, who made me feel the college home by giving me tremendous emotional support and a sense of belonging.

I am profoundly grateful to my friends and mentors back home for their unconditional support, love, and patience. To my closest and dearest friends, Vinay Kumar Meda, Pooja Paruchuri, and Lavanya Avula, and my mentors, Swagat Kumar, and Imran Shareef, this dream would have been impossible for me to achieve without your presence in my life.

It goes without saying that my deepest thanks go to my parents, Venkateswarlu Vankadari, and Lakshmi Kanthamma Vankadari, and brother, Bhanu Vankadari for all the struggles they went through, always believing in me no matter what, and for giving me that unconditional love, strength, and security. I also must thank my brother for taking care of the family when I was not there at home. It gave me tremendous mental peace to concentrate on my DPhil. I also must thank Christine Dunnett for her love, support, and patience during this important moment in my life.

Finally, I would like to acknowledge the funding and support from the University of Oxford and the Amazon Web Services (AWS) - Lighthouse Graduate Scholarship, without which this research would not have been possible.

My sincere apologies to the people I missed. This PhD journey would not have been possible without the support of many incredible and intelligent people throughout. I thank you all and am forever grateful and indebted to you!

Abstract

Accurate depth and camera motion estimation are critical components in various applications, including autonomous driving, robotics, and augmented reality, where understanding the 3D structure of the environment is essential. Cameras are widely used in these systems due to their affordability, compactness, and ability to capture rich visual information. However, extracting 3D information from 2D images is a challenging problem, especially in dynamic and varying conditions like changes in lighting. Robustness, accuracy, and generalisation across different scenarios, particularly under challenging illumination conditions such as nighttime, are of paramount importance to the successful deployment of these systems in real-world environments.

However, achieving these requirements poses significant challenges. Traditional methods, while effective in controlled environments, often struggle with robustness and adaptability in diverse, real-world settings. Deep learning approaches, particularly those based on supervised learning, perform well on training data but can fail to generalize to new domains, especially under poor illumination conditions. Self-supervised methods, which aim to reduce reliance on ground truth data, often suffer from the limitations of photometric losses that are sensitive to noise, dynamic objects, and varying lighting conditions. Emerging visual foundational models, trained on massive and diverse datasets, offer robust and generalisable feature representations. These models have shown promise in various tasks like classification and semantic segmentation. However, integrating them effectively into self-supervised depth and ego-motion estimation frameworks remains challenging, especially when addressing low-light environments and the need for high precision in keypoint detection and matching.

This thesis presents a series of contributions aimed at overcoming these challenges. Firstly, we enhance photometric losses for self-supervised depth estimation by introducing Neural Intensity Transformation (NIT) to account for illumination changes, extending pixel projection to handle dynamic objects, and incorporating denoising techniques. These improvements enable robust depth

and ego-motion estimation across day and nighttime conditions. Secondly, we propose a stereo-matching approach that leverages visual foundational models, introducing a parameter-free mask for noisy regions and a trainable decoder that improves robustness and generalization in low-light conditions. Finally, we introduce a novel learning-free keypoint detector that utilizes pretrained image encoders for robust keypoint detection and matching, demonstrating superior performance across multiple tasks and datasets.

In summary, this thesis advances the state-of-the-art in depth and camera motion estimation by addressing key challenges associated with varying illumination conditions and noisy environments. The proposed solutions further harness the power of visual foundational models, enhancing robustness, accuracy, and generalization, making them applicable to a wide range of real-world scenarios, particularly in autonomous systems and robotics where reliable 3D perception is critical.

Contents

1 Introduction **1**

1.1 Motivation . . . 2
1.2 Research Challenges . . . 5
1.3 Contributions . . . 9
1.3.1 Thesis Related Publications . . . 11
1.3.2 Additional Co-Authored Publications . . . 11
1.4 Summary . . . 12

2 Background **14**

2.1 Preliminary Concepts - Traditional Methods . . . 15
2.1.1 Depth Estimation . . . 15
2.1.2 Camera Motion Estimation . . . 17
2.1.2.1 Keypoint Detection Methods . . . 22
2.1.3 Joint Depth and Camera Motion Estimation . . . 23
2.2 Deep Learning Methods . . . 26
2.2.1 Supervised Depth Estimation . . . 26
2.2.1.1 Monocular Depth Estimation . . . 26
2.2.1.2 Stereo Depth Estimation . . . 29
2.2.1.3 Multi-View Depth Estimation . . . 31
2.2.2 Supervised Camera Motion Estimation . . . 33
2.2.2.1 Keypoint Detection and Matching Methods . . . 35
2.2.3 Self-Supervised Depth and Camera Motion Estimation . . . 39
2.2.4 Challenges with End to End Learning . . . 44
2.2.4.1 Visual Foundational Models . . . 45
2.3 Datasets and Metrics . . . 47
2.3.1 Datasets . . . 47
2.3.2 Metrics . . . 49
2.4 Summary . . . 52

3 Relaxing Photometric Loss for All Day Depth and Ego-motion Estimation 53
3.1 Introduction . . . 53
3.2 Related Work . . . 56
3.3 Proposed Method . . . 58
3.3.1 Baseline Method . . . 59
3.3.2 Lighting Change Compensation . . . 60
3.3.3 Motion Compensation . . . 60
3.3.4 Image Denoising . . . 61
3.3.5 Full Pipeline . . . 62
3.3.6 Making the Pipeline Bidirectional . . . 63
3.4 Experiments . . . 64
3.4.1 Training and Evaluation Details . . . 64
3.4.2 Depth Evaluation . . . 65
3.4.3 Ablation Study . . . 67
3.4.4 Ego-Motion Evaluation . . . 68
3.4.5 Ease-of-Deployment . . . 69
3.5 Limitations and Future Work . . . 71
3.6 Potential Extensions in the Era of Foundational Depth Estimation . . . 72
3.7 Conclusions . . . 73

4 Self-supervised Nighttime Stereo Depth Estimation using Visual Foundation Models 75
4.1 Introduction . . . 76
4.2 Related Work . . . 78
4.2.1 Nighttime Self-Supervised Depth Estimation . . . 78
4.2.2 Supervised Stereo Depth Estimation . . . 79
4.2.3 Visual Foundation Models . . . 79
4.3 Proposed Method . . . 79
4.3.1 Feature Extractor . . . 80
4.3.2 Projection Head . . . 81
4.3.3 Stereo Matcher . . . 81
4.3.4 Upsampling . . . 83
4.3.5 Training Losses . . . 83
4.4 Experiments . . . 84
4.4.1 Training Details . . . 84
4.4.2 Baselines . . . 85

4.4.3 Depth Evaluation . . . 87
4.4.4 Ablation Studies . . . 91
4.4.5 Comparison with Monocular Methods . . . 92
4.4.6 2D vs 3D Image Warping . . . 93
4.4.7 Effect of Patch Size . . . 93
4.4.8 Adaptability to Other Visual Foundational Models . . . 94
4.4.9 Training with Day and Night Images . . . 95
4.4.10 Generalization to Other Nighttime Datasets . . . 95
4.5 Limitations and Future Work . . . 97
4.6 Conclusions . . . 97

5 Min-Max Detector: Digging into Descriptions for Keypoints 99
5.1 Introduction . . . 100
5.2 Related Work . . . 102
5.2.1 Keypoint Detection and Description . . . 102
5.2.2 Visual Foundation Models . . . 104
5.2.3 Entropy in Keypoint Detection and Description Learning . . . 104
5.2.4 Comparison with Standard Corner Detection Methods . . . 105
5.3 Proposed Method . . . 106
5.3.1 Motivation . . . 106
5.3.2 Min-Max Detector . . . 108
5.3.3 Detection and Matching . . . 109
5.4 Experiments . . . 109
5.4.1 Methods for Comparison . . . 111
5.4.2 Qualitative Results . . . 111
5.4.3 Homography Estimation . . . 113
5.4.4 Camera Pose Estimation . . . 114
5.4.5 Generalization of Our Detector . . . 116
5.4.6 NN-distance vs Match Success Rate . . . 116
5.5 Limitations and Future Work . . . 117
5.6 Conclusions . . . 117

6 Conclusions and Future Work 119

Bibliography 127

Chapter 1

Introduction

Estimating accurate depth and ego-motion in a 3D environment using Cameras is one of the fundamental challenges in computer vision research. Depth estimation involves determining the distance from the camera to various points within a scene, while camera motion (a.k.a ego-motion) estimation focuses on determining the camera's movement in three-dimensional space over time. Together, these tasks enable machines to perceive and navigate the 3D world using only 2D images or video sequences, a capability that is crucial for various domains such as autonomous navigation [1], robotic manipulation [2, 3], and mixed reality [4]. While active depth sensors like LiDAR provide metric depth, they can be power-intensive and produce sparse data. As another type of native depth sensor, RGB-D cameras have a limited depth sensing range (typically ≤ 10 meters) and work poorly under strong ambient illumination. As a non-optical sensor, RADAR data can be noisy and sparse. Visual Inertial Odometry (VIO) can augment camera-based methods by fusing inertial measurements, but this adds hardware complexity and requires precise sensor calibration.

In contrast, cameras, particularly RGB cameras, are an attractive choice for depth and ego-motion estimation due to their cost-effectiveness, low power consumption, and extensive commercial availability. They rely on passive sensing rather than emitting signals, which significantly reduces the risk of interference in multi-sensor environments. Another notable advantage of cameras lies in the dense visual information they capture. Unlike LiDAR or RADAR, which produce sparse point clouds or low-resolution returns, cameras record high-resolution pixel data with textures, colors, and fine structural details—that is essential for accurate scene understanding and precise motion tracking. Moreover, decades of research in computer vision have produced robust methods and libraries for tasks such as feature detection, optical flow, and structure-from-motion, providing a mature foundation for camera-based solutions.

Cameras rely on ambient light to capture clear images. They capture only the 2D projection of the 3D world, which means some depth information is lost. This makes it hard

to estimate the true 3D structure of the scene. Accurately understanding the camera's 3D motion from images alone is a non-trivial task. In low-light conditions or when lighting changes, images can become noisy or lack contrast. Motion blur and image noise are common problems when the camera or objects move quickly. In areas with few textures or repetitive patterns, it is difficult to track reliable features. Processing high-resolution images in real time also requires a lot of computational power. Given these challenges, designing a robust and generalisable system that can input images and accurately estimate scene depth and camera motion has been a long-standing problem in the field.

1.1 Motivation

The motivation behind addressing the problem of depth and ego-motion estimation lies in the need to develop a sophisticated understanding of real-world environments, enabling machines to operate autonomously and safely or enhancing human experiences in immersive virtual or augmented realities. In autonomous navigation [5], for example, accurate depth perception is essential for mapping unfamiliar places [6], planning routes from point A to point B [7], and, most critically, detecting and avoiding obstacles during transit. Likewise, precise ego-motion estimation ensures that robots can navigate accurately and predictably through dynamic environments. These capabilities are equally vital in archaeology for the 3D reconstruction of heritage and cultural sites [8], in robotic manipulation for tasks like object localization and grasping [2, 3], and in medicine for procedures such as minimally invasive robotic surgeries [9]. In recent years, with the rise of virtual and augmented reality applications, the demand for methods that are both computationally efficient and highly accurate has significantly increased. Overall, these real-world applications require methods that are *accurate*, *fast*, *generalisable*, and *robust*.

Accuracy means how close the estimates are to their actual value, for instance, effectively detecting an obstacle at the correct distance or mapping the landmarks of a scene true to their ground truth position. The method should be capable of operating in *real-time* to seamlessly integrate into the physical world. *Generalisability* means that the method should not be affected by changes in scene structure, infrastructure, or objects present, as well as by the location of the scene. *Robustness* implies that the method should not be affected by factors such as texture, sensor or scene noise, imperfect information (like image blur), or varying lighting and weather conditions. However, balancing these competing requirements is challenging. Methods that are accurate in one scene may fail in others, lack generalisation, or may degrade when there is a slight change in scene conditions, lack robustness, or simply take too much time to compute. There is no universally accepted method to meet

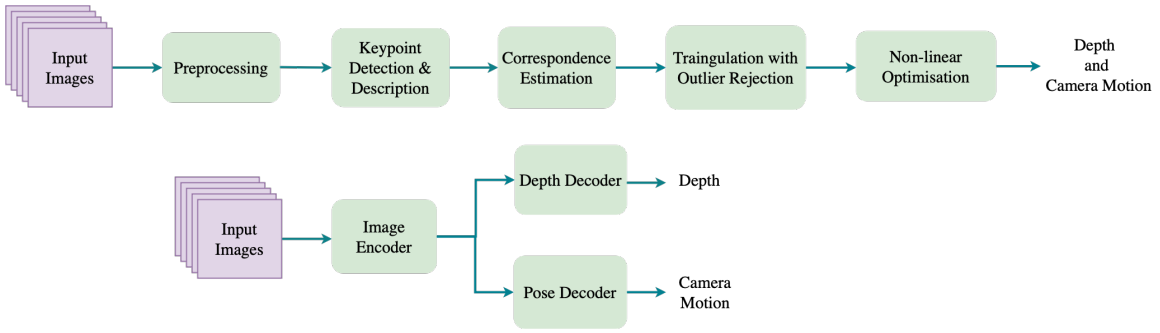


Figure 1.1: Comparison of traditional and deep learning-based approaches for estimating depth and camera motion: Classical methods (top) typically rely on handcrafted features and complex optimization algorithms, making them computationally expensive and sensitive to noise. Deep learning models (bottom) simplify the process by automatically learning hierarchical features from data, enabling faster, more robust, and more accurate estimation, while significantly reducing the need for manual feature engineering.

all these requirements. Although a tremendous amount of excellent work has been done before this thesis, which aimed to design systems that meet these requirements, we are still far from being able to satisfy all these demands. Previous works in this field can broadly be categorized into traditional methods and deep learning-based methods. A typical pipeline of these methods and steps involved is illustrated in Figure 1.1. Traditional methods are grounded in geometry and physics, often using linear or non-linear optimization to estimate depth and camera motion. These methods make numerous assumptions about the scene, sensor, and objectives. For example, frameworks like Simultaneous Localization and Mapping (SLAM) or Structure from Motion (SfM), which jointly estimate both depth and ego-motion, assume the scene is static (nothing moves), rigid (objects do not change shape), and textured (objects are easily distinguishable). Additionally, sensors must be geometrically and photometrically pre-calibrated, with accurate intrinsics and extrinsics ready to use, and cost functions for optimization must be smooth and differentiable. Even when addressing these problems separately, many of these assumptions remain challenging.

Various hand-crafted techniques based on heuristics [10, 11, 12] to describe pixels/ patches/ images, cost functions [13, 14, 15] to capture relationships between input images and expected outputs, and optimization methods [16, 17] for better estimates have been proposed. By design, all these methods consist of several sub-problems, each with its own challenges that must be solved to achieve the desired outcomes. For example, to estimate the ego-motion of the camera between two images, reliable points in the image must first be discovered and then matched. These matched points are then fed into a linear or non-linear optimizer with an outlier rejection module to estimate the camera pose. Any misstep along

the way can cascade and impact final performance. This raises the question: how can we achieve this in a single step? Given a pair of images, can we design a system that directly estimates the desired output without involving all these intermediary steps? The short answer is yes.

In 2012, the introduction of AlexNet [18] led to a significant leap in image classification accuracy. This method used a linear stack of convolutional layers where each layer fed information to the next, and a Multi-Layer Perceptron (MLP) aggregated the information from the convolutional layers. With approximately 60 million trainable parameters (convolution filters and MLP weights), AlexNet could classify a given image into a predefined category without relying on heuristics to describe features that generalize across similar images. This advancement in learning features directly from data, rather than hand-crafting them, was a breakthrough in the field. Since then, there has been extensive research to extend this approach to nearly all computer vision problems, including depth and ego-motion estimation.

A typical deep learning network consists of several layers stacked on top of each other, and it inputs an image and outputs a feature map in a D-dimensional space. These feature maps are then decoded for the task of interest. For example, in an ego-motion task, the feature maps are decoded into 6 or 7 values, while in depth estimation, a 1-channel image with dimensions corresponding to the input image's height and width is estimated. These estimates are compared with ground truth data to guide the network to learn the desired outcomes, typically using a differentiable loss function.

One limitation of these methods is that they require ground truth data to guide the network in solving the task at hand. These methods are known as supervised learning methods. However, it is physically impossible to gather ground truth data for all variations that exist in the real world in any problem domain. This limitation has led the research community to explore self-supervised methods—could also be understood as hybridization of supervised methods—that utilize similar architectures as supervised ones but trained with loss functions grounded by geometry and physics of the scene of interest to address tasks such as depth and ego-motion estimation without requiring ground truth data. This thesis focuses on the challenges of self-supervised depth and ego-motion estimation, and for the sake of brevity, other related problems are not discussed. The major research challenges associated with self-supervised depth and ego-motion estimation, which we aim to address or provide promising directions towards, are explained in the following section.

1.2 Research Challenges

The progress in deep learning has enabled significant advancements over classical methods of depth and camera motion estimation. Deep learning approaches can be broadly divided into two categories: supervised methods [19, 20] and self/unsupervised methods [21]. Supervised approaches typically achieve excellent results on the datasets they are trained on, but their reliance on ground-truth information during training limits their applicability in new domains. In contrast, self- and unsupervised methods often employ geometry-based loss functions inspired by the strong physical principles of traditional methods [22, 23]. These methods alleviate the need for ground-truth depth, with the commonly used loss function referred to as the photometric loss or appearance loss. This loss function is based on the assumptions that the scene is *static*, the illumination is *diffusive* and *temporally consistent*, and the images are *free of noise and occlusions* [22, 23, 24, 25].

There have been numerous efforts to bridge the performance gap between supervised and self-supervised methods by addressing or masking the failures of these assumptions. However, most of these methods are designed for well-illuminated (i.e. daytime) scenarios, where the assumptions are only partially true, leading to poor performance during low-illuminated (i.e. nighttime) conditions [26, 27]. The important observations to make here are first, the *accurate* models [28] trained during the day failed miserably when tested on the same location but at night lacking *robustness* and *generalisation* [27, 29]. Second, it was not possible to train the models on the night data as the photometric losses were confusing and inaccurate. During the day, it is somewhat reasonable to assume that illumination is moderately temporally consistent for image sequences captured outdoors, as the sun is the dominant light source, and its light changes only slowly over a time. However, as shown in Figure 1.2, in low-light conditions such as the nighttime introduces numerous point light sources (e.g., car headlights, lamp posts) that can cause drastic changes in illumination from one frame to the next. Additionally, motion blur associated with dynamic objects (including the ego-vehicle) worsens due to the longer exposure times typically used in low-light conditions [30, 31]. Moreover, the Signal-to-Noise Ratio (SNR) of darker images is much lower than during daytime that is well lit. These conditions which we refer to as low-light conditions inhibit the straightforward use of deep networks based on photometric loss for the images captured in these conditions. Addressing these problems is still a challenge in this area.

Recently, similar to the success of large language models (LLMs) in natural language processing (NLP), there has been significant progress in training massive models with billions of parameters using Vision Transformers [32] as the underlying architecture on massive

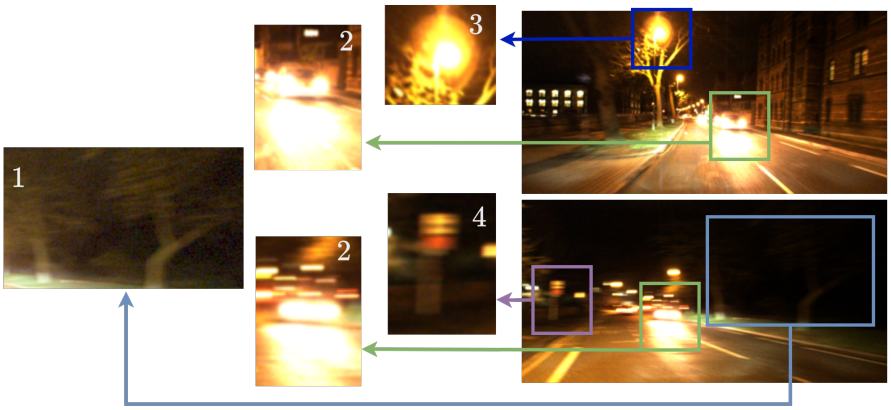


Figure 1.2: Typical challenges faced during night operation: 1. Noise, 2. Moving objects with lights and image saturation, 3. Point light sources, and 4. Image blur. These challenges are not typically encountered during the daytime.

datasets like Facebook LVD-142M [33]. Notable models in this category include CLIP [34], DINO [35], DINO-V2 [33], and AM-RADIO [36]. These models have demonstrated fine-grained semantic and/or spatial understanding of scenes, embedding this information within the features they output. These emergent properties have quickly led to their adaptation into various existing supervised learning-based applications such as classification, semantic segmentation, and depth estimation. Models such as these have a strong learned prior that also addresses issues such as generalization to novel images and robustness to illumination changes. As foundational models become more compact and computing more efficient, such powerful and generalisable feature extractors will likely be applicable to many more tasks.

With these advancements and challenges in mind, this thesis primarily investigates the theoretical question of enabling accurate, robust, and generalisable depth and ego-motion estimation under low-light conditions. To address this objective, it focuses on three specific research questions aimed at developing a system that can reliably perform in diverse scenarios. While practical considerations—such as speed is acknowledged as important for real-world deployment, it is not the central emphasis of this work. Instead, the principal goal is to advance the theoretical understanding and algorithmic development necessary for effective depth and ego-motion prediction in challenging low-light environments. The details are as follows:

1. **Monocular Depth Estimation**

   Monocular depth estimation is an inherently ill-posed problem because a single image does not provide direct information about scale or absolute distance. Unlike stereo or multi-view approaches, where geometric constraints can be used to triangulate 3D

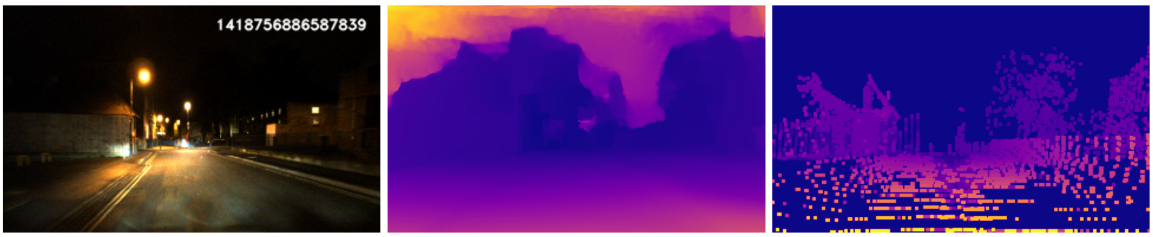


Figure 1.3: Failure of the state-of-the-art method IGEV-stereo [39] when tested against ground truth (accumulated LiDAR points) on a nighttime sample. The method fails to estimate accurate depth, particularly in poorly illuminated areas such as the sky.

structure, monocular systems must rely on visual cues such as perspective, texture gradients, and scene semantics to infer depth. These cues can be ambiguous; objects of different sizes may appear similar in projection, and the same object can look vastly different under varying illuminations or poses. Moreover, occlusions, reflective surfaces, and complex scene layouts can further complicate the estimation process. As a result, robust monocular depth estimation often requires learning-based methods like [37, 28]. While supervised learning methods rely on per-pixel depth maps, self-supervised methods alleviate this problem using photometric loss. However, the photometric loss has assumptions as explained above that make it almost impossible for its direct use in low-light conditions as explained above. To address these limitations, some methods estimate depth from nighttime images using techniques like night-to-day feature translation [26], losses calculated in feature space [27], or GAN-based depth regularisers [38]. However, these approaches do not directly address the limitations of photometric losses. This leads to the first research question:

How can we relax the assumptions made by the photometric loss to use it for robust self-supervised monocular depth and ego-motion estimation?

2. **Stereo Matching**

   Monocular depth estimation relies on temporal parallax from a single camera to triangulate depth. Alternatively, depth information can be captured using two or more cameras simultaneously. Humans, for example, use two eyes (analogous to cameras) to perceive depth through parallax. Stereo matching algorithms replicate this by establishing correspondences between images captured by two cameras, using the disparity between pixel locations to triangulate depth.

   Stereo matching is a challenging problem, especially in regions with blur, noise, reflections, and low illumination. To illustrate the current state of generalization,

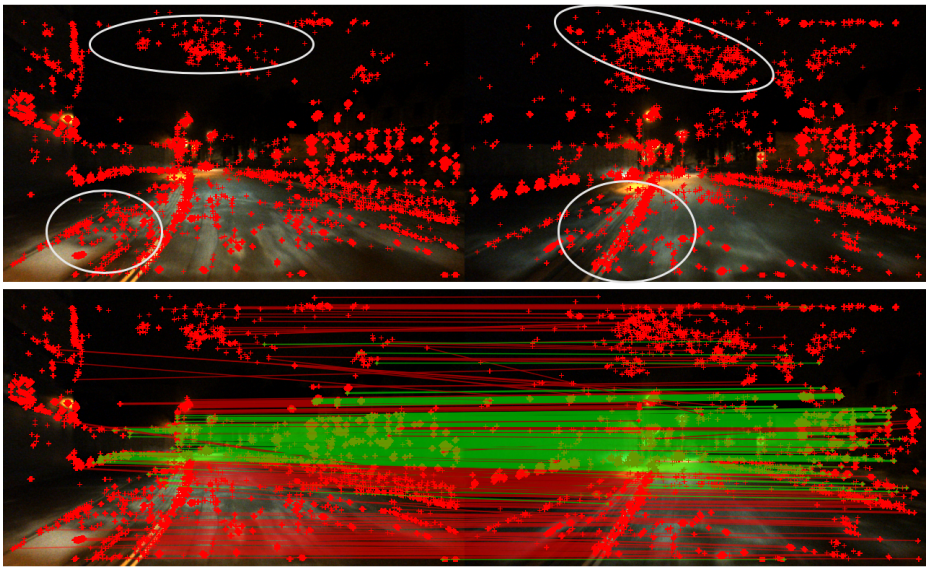


Figure 1.4: Challenges in keypoint detection and matching in low-light conditions using a deep learning-based detector, DeDoDe V2 [40]. Observations include: 1. Many keypoints are detected on the road, in the sky, and on street lights.2. When matched with a nearby image (approx. 5m away), the model fails to establish accurate correspondences, with green indicating good matches and red indicating bad correspondences.

robustness, and accuracy, Figure 1.3 shows a nighttime image along with depth estimates from a state-of-the-art supervised deep learning-based stereo matching method, IGEV-stereo [39]. This method fails to estimate accurate depth under poor illumination (i.e. nighttime) conditions, highlighting issues with robustness to noisy areas such as sky and generalization to low-light environments. This prompts the next question:

Is it possible to make a self-supervised stereo matching system that is accurate, robust, and generalisable?

3. **Camera Motion Estimation**

   Traditional methods like SIFT [10], ORB [12], and SURF [11] rely on handcrafted features to detect and match keypoints, but they often struggle with robustness in real-world conditions such as varying lighting, texture, and scale. These methods are effective in controlled environments but lack adaptability across diverse scenes.

   Deep learning-based approaches, particularly CNNs, improved keypoint detection by learning features directly from data, leading to more robust performance in challenging environments. However, these models require extensive task-specific training and often struggle with generalization, especially when applied to unseen or varied data.

Figure 1.4 shows an example of such a scenario. Although deep learning has brought significant advancements over traditional methods, achieving robustness, repeatability, and generalisability, particularly for tasks like camera motion estimation that demand high precision across a wide range of scenarios, is still challenging.

In contrast, visual foundational models, trained on massive and diverse datasets, excel in generalization and robustness in their features, making them highly effective in downstream tasks like semantic segmentation and depth estimation. However, they are not optimized for keypoint detection and matching, which require us to find distinct and repeatable features. This leads to the final question:

How to harness the feature representations of foundational models for keypoint detection and matching, thereby enhancing tasks like camera motion estimation?

The above-mentioned research challenges have been addressed with novel solutions, leading to state-of-the-art performance in both depth and ego-motion estimation. The contributions made towards solving these research challenges are described in the following section.

1.3 Contributions

In this thesis, the aim is to design an accurate, robust and generalisable method that estimates depth and motion in a completely self-supervised manner. The self-supervision is achieved through the use of photometric loss. However, the assumptions underlying this photometric loss are often violated in poorly illuminated conditions, such as nighttime. This thesis addresses these violations in both monocular and stereo cases, while also incorporating recent advancements in the field of computer vision related to training foundational models. The contributions made in this thesis are as follows:

1. The standard photometric losses are improved with three different modules:

   (a) The primary issue is the observed intensity of the 3D point changing between images during the nighttime. To address this, we propose to estimate how much the illumination changes and compensate for it with a transformation we call Neural Intensity Transformation (NIT).

   (b) The assumption of a static scene is often violated due to dynamic objects. To mitigate this, we extend the reach of the pixels projected by the ego-motion, allowing them to land on the correct pixels that correspond to moving objects, giving extra freedom for photometric loss to account for dynamic objects.

(c) Lastly, noise in the images can lead to inaccurate photometric loss estimation. To address this, we propose to denoise the images while calculating the photometric loss.

These corrections applied to the photometric loss resulted in a model that could be trained and tested across varying illumination conditions, such as day and nighttime scenarios, achieving state-of-the-art results in depth and ego-motion estimation. This work is introduced in Chapter 3 and was published at **CoRL 2022**.

2. We propose a trainable decoder method that inputs stereo images and performs feature matching in a coarse-to-fine manner to estimate per-pixel depth maps. The key contributions are:

   (a) To improve robustness, we introduce a parameter-free mask to avoid noisy depth estimation from low Signal-to-Noise Ratio (SNR) regions, while also encouraging the model to retain useful information from these regions.

   (b) To improve generalization, we use visual foundation models [41, 33, 36] as an encoder and design a trainable decoder rather than training the model end-to-end.

   (c) We propose a more rational set of depth evaluation metrics to allow for more granular and fair evaluation across different depth ranges and conducted comprehensive experimental and ablation studies on two datasets to showcase the efficacy of our approach.

   This is the first method in this line of work to explore self-supervised depth estimation using visual foundational models, achieving performance on par with supervised learning methods. This work is introduced in Chapter 4 and was published at **ICRA 2024**.

3. We introduce a novel learning-free keypoint detector that can be applied to any image encoder to extract high-quality keypoints and descriptors. The contributions are as follows:

   (a) We propose and advocate for a new measure of uniqueness for a feature based on its distance from the nearest neighbour in the feature space.

   (b) We introduce a novel learning-free keypoint detector that can be applied to any feature extraction model to extract high-quality keypoints.

(c) We perform extensive ablation and experimental studies on various datasets, tasks, and configurations to understand the benefits and limitations of the proposed detector and the visual foundational features.

This work is introduced in Chapter 5 and is currently under review at a conference.

1.3.1 Thesis Related Publications

1. **Vankadari M**, Golodetz S, Garg S, Shin S, Markham A, Trigoni N. When the sun goes down: Repairing photometric losses for all-day depth estimation. In Conference on Robot Learning 2022 (pp. 1992-2003). PMLR.

2. **Vankadari M**, Hodgson S, Shin S, Markham Kaichen Z., Trigoni N. Dusk Till Dawn: Self-supervised Nighttime Stereo Depth Estimation using Visual Foundation Models. International Conference on Robotics and Automation, ICRA 2024.

3. **Vankadari M**, Hodgson S, Shin S, Markham, Trigoni N. Min-Max Detector: Digging into visual foundational models for keypoints.—under review.

   As the lead author, I oversaw the entire research and publication process. I developed the research frameworks and coordinated contributions from my co-authors. I ensured all sections of the papers were consistent, managed editing and revisions, and kept communication open with co-authors and reviewers. In all these publications, I wrote key sections, including the methodology and results.

1.3.2 Additional Co-Authored Publications

1. Garg S, **Vankadari M**, Milford M. Seqmatchnet: Contrastive learning with sequence matching for place recognition & relocalization. In Conference on Robot Learning 2022 Jan 11 (pp. 429-443). PMLR.

   Contribution: This work was an external collaboration. In this work, I helped the lead author design the research idea and conducted a few experiments, wrote some parts of the literature, and the experiments section.

2. Golodetz S, **Vankadari M**, Everitt A, Shin S, Markham A, Trigoni N. Real-time hybrid mapping of populated indoor scenes using a low-cost monocular uav. In 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2022 Oct 23 (pp. 325-332). IEEE.

Contribution: In this work, I helped the lead author to collect the dataset on which the method was tested. I also helped in designing the scale-recovery idea that we used to estimate the true metric scale of the map using a drone.

3. Shin S, Golodetz S, **Vankadari M**, Zhou K, Markham A, Trigoni N. Sample, Crop, Track: Self-Supervised Mobile 3D Object Detection for Urban Driving LiDAR. In 2023 IEEE International Conference on Robotics and Automation (ICRA) 2023 May 29 (pp. 7090-7096). IEEE.

   Contribution: I helped the lead author with the research idea design and formulating and evaluating the necessary experiments. I also helped them write the publication.

4. Bhutani V, Majumder A, **Vankadari M**, Dutta S, Asati A, Kumar S. Attentive one-shot meta-imitation learning from visual demonstration. In 2022 International Conference on Robotics and Automation (ICRA) 2022 May 23 (pp. 8584-8590). IEEE.

   Contribution: This paper was another external collaboration, where I helped the lead-author with their core research idea, designing the experiments, and evaluating the results.

5. Shin S, Zhou K, **Vankadari M**, Markham A, Trigoni N. Spherical Mask: Coarse-to-Fine 3D Point Cloud Instance Segmentation with Spherical Representation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2024 (pp. 4060-4069).

   Contribution: In this paper, I contributed to the research framework design, experimental evaluation, and a part of the paper writing.

1.4 Summary

In this thesis, we present several contributions to the field of depth and ego-motion estimation. Chapter 2 provides an in-depth discussion of existing depth estimation and ego-motion estimation literature, covering traditional methods as well as deep learning approaches. We begin by explaining the fundamental concepts and techniques for depth estimation, followed by a detailed review of ego-motion estimation methods. Then, we explore joint depth and ego-motion estimation approaches, comparing traditional and modern deep learning methods. Throughout, we highlight the challenges faced in these areas, such as robustness under varying environmental conditions and generalization to new scenarios. We conclude by discussing the latest advancements, including visual foundational models, and how these

innovations are shaping the future directions in depth and ego-motion research, followed by a discussion about the datasets and the metrics used in this thesis.

Chapter 3 presents our first major contribution: an end-to-end architecture that addresses the issue of photometric error under varied illumination conditions, such as day and night. We propose a method to make photometric error handling robust to changes in lighting, enhancing both depth and ego-motion estimation accuracy. Extensive experiments and ablation studies are conducted to evaluate the performance of our approach across diverse scenarios. We compare our method to existing techniques and demonstrate its superior robustness in challenging lighting environments.

In Chapter 4, we introduce a stereo depth estimation system based on visual foundational model features. By training only the decoder, our approach significantly reduces the computational burden while improving robustness and accuracy, especially in low-light situations like nighttime. This system addresses some of the limitations identified in Chapter 3, offering a promising direction for generalizing to new datasets. We demonstrate the effectiveness of our method in alleviating the issues related to robustness and accuracy under adverse conditions.

Chapter 5 tackles the problem of camera motion estimation through keypoint matching. We propose a novel keypoint detector that leverages local density as a cue for detecting reliable keypoints. Our detector shows strong potential when combined with features from foundational models, paving the way for learning-free detectors. We also test the effectiveness of our detector with a pretrained detector, DISK [42]. Our method is evaluated across two different tasks on three datasets, using four foundational models, showcasing its versatility and effectiveness in real-world scenarios.

Finally, in Chapter 6, the thesis concludes by summarizing the key insights from each chapter. We reflect on the challenges addressed and the contributions made towards improving depth and ego-motion estimation under varied conditions. Future research directions are discussed, focusing on how the findings of this thesis can inspire further advancements in the field.

Chapter 2

Background

This chapter provides the technical and conceptual background necessary to understand the proposed contributions of this thesis. The primary problems addressed are per-pixel depth estimation from one or more images and the estimation of camera motion between consecutive images. Depth estimation involves determining the distance between objects in a scene and the camera, while camera motion estimation focuses on understanding the movement of the camera relative to its environment.

The progress made in this direction can be divided into two parts: traditional methods and deep learning-based methods. Before the advent of deep learning, many traditional algorithms grounded in geometry and physics attempted to tackle these problems. These algorithms often relied on strong assumptions about their optimization objectives, the scene of interest, or the relationships between the corresponding pixels between the images. However, with the development of deep learning, many of these assumptions have been relaxed, enabling the creation of more flexible approaches and leading to more accurate solutions.

The advances in deep learning have led to the proposal of numerous methods that leverage the ability to learn from data using both supervised and self-supervised approaches. These methods have significantly outperformed traditional techniques. However, while learning-based methods have overcome many challenges associated with conventional assumptions, they still struggle with generalization across different scene conditions, such as varying weather, illumination, novel objects, and new infrastructures. For a long time, the goal of the machine learning community has been to design generalized encoders capable of handling a wide range of conditions. This goal has become more attainable with the advent of recent visual foundational models, which are trained on vast amounts of data. The impressive capabilities of these models have prompted us to explore their potential for addressing our specific problems.

This chapter is divided into nine major sections covering the background of topics: depth estimation, camera motion estimation, joint depth and ego-motion estimation, visual foundational models, and datasets and metrics. In the first six sections, we discuss traditional methods, their assumptions, and their challenges, followed by a review of how deep learning approaches have attempted to address them. Later, we explore the visual foundational models, their emergent properties, and their suitability for the problems of depth and ego-motion estimation. In the last two sections, the details of the datasets and the evaluation metrics used in this thesis are explained first, followed by a section summarising the entire chapter.

2.1 Preliminary Concepts - Traditional Methods

Traditional methods for depth estimation, camera motion estimation, and their joint determination have formed the backbone of 3D scene understanding for decades. These techniques generally leverage geometric principles, such as epipolar constraints and triangulation, to infer depth information from multi-view or stereo configurations, while camera motion is estimated through feature-based correspondences and iterative optimization. Despite the surge of learning-based approaches in recent years, a solid grasp of these traditional methods, alongside their benefits and limitations, remains fundamental for advancing modern 3D perception algorithms like the ones proposed in this thesis.

2.1.1 Depth Estimation

Depth estimation has been a crucial research area in computer vision, particularly before the advent of deep learning. Various traditional methods have been developed to estimate depth using cameras, each with its own advantages and challenges.

Stereo vision is a well-established method that involves capturing two images from slightly different viewpoints. A simple illustration of estimating depth from stereo images is shown in Figure 2.1. Depth D is inferred from the disparity d between corresponding points in the images using the formula $D = \frac{fb}{d}$, where f is the focal length and b is the baseline between the cameras. Although this method is straightforward, it heavily relies on accurate stereo matching, which can be challenging in low-texture or repetitive areas. Marr and Poggio [43] were among the pioneers in computational stereo vision, and several algorithms have since been developed to address stereo matching. Block matching, for instance, is a simple and computationally efficient technique, but it struggles in challenging areas of the image [44]. Semi-Global Matching (SGM) offers improved accuracy by minimizing a global cost function that balances data fidelity and smoothness, producing high-quality

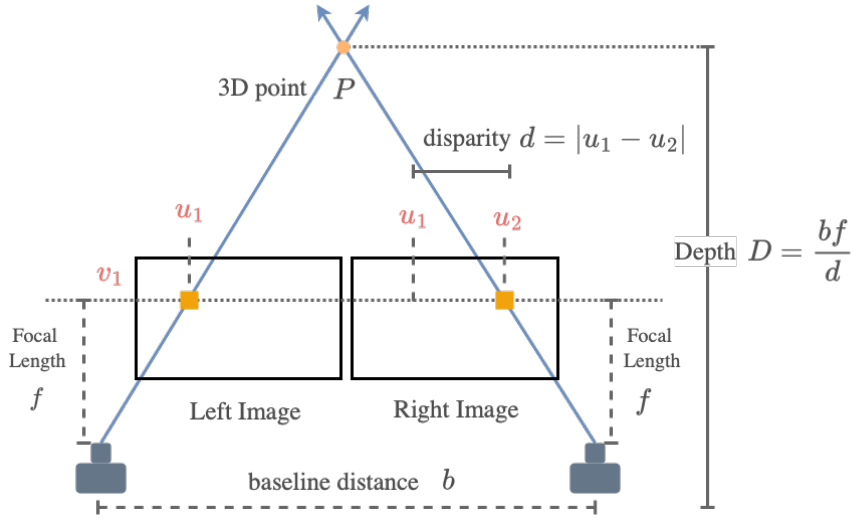


Figure 2.1: Illustration of depth computation using stereo images. Disparity is determined by matching corresponding pixels in the left and right images, which is then converted into depth using the camera's focal length and the baseline distance between the left and right cameras.

disparity maps [45]. Other methods, such as graph cuts, frame stereo matching as an energy minimization problem and solve it using graph cuts, which makes them effective in complex scenes [46]. Dynamic programming approaches optimize the disparity map by minimizing costs along each scanline, providing a balance between efficiency and accuracy [47].

Extending stereo vision to multiple images, Multi-view Stereo (MVS) aims to provide a more detailed 3D reconstruction by analyzing correspondences across multiple views. The depth information is recovered by solving the epipolar constraint across all views [48]. MVS methods are typically more robust than stereo vision but also more computationally intensive. Volumetric methods [49], for example, represent the scene as a 3D voxel grid and iteratively carve out empty space based on visibility constraints. These methods are robust but require significant computational resources. Alternatively, patch-based methods [50] fit small planar patches to the images and merge them into a consistent 3D model, which allows them to handle complex scenes with fine details. Surface-based methods [51] directly reconstruct surfaces by minimizing a global cost function that measures consistency between the surface and the observed images, offering a balance between detail and computational cost.

Another approach to depth estimation is photometric stereo, which estimates surface normals by analyzing multiple images taken under different lighting conditions. This method assumes a Lambertian surface, where the intensity I_i under a light direction $\mathbf{l}_i$ is given by $I_i = \mathbf{n}^\top \mathbf{l}_i$. By solving a system of linear equations, the surface normals $\mathbf{n}$ can be estimated,

and the depth map D can be recovered by integrating these normals [52, 53]. Shape from Shading [54], a special case of photometric stereo, reconstructs 3D shapes from shading patterns in a single image. The main challenge with photometric stereo is that it requires known lighting conditions and assumes constant surface reflectance, which may not hold true in real-world scenarios.

Depth from Focus/Defocus methods estimate depth by analyzing the sharpness or blur in an image, which is directly related to the distance of the object from the camera. The degree of blur is quantified by the *circle of confusion*, and the depth is inferred from that [55]. This method is particularly effective when multiple images are captured with varying focus settings, enabling accurate depth estimation by comparing the defocus levels in these images [56]. However, the accuracy of the method depends on precise control of the focus and is less effective in dynamic scenes where object or camera motion introduces variability.

Shape from Texture infers 3D shape by analyzing texture pattern distortions across a surface. Regular texture patterns will appear distorted when mapped onto a 3D surface due to perspective projection [57]. This technique is effective for surfaces with regular, repetitive textures but may struggle with irregular or complex textures. Additionally, the vanishing point method [58], which uses the convergence of texture lines to infer depth and orientation, is particularly effective in scenes with pronounced perspective effects [59].

Finally, active methods, such as laser scanning and structured light, involve projecting a known pattern onto a scene and analyzing its deformation to estimate depth. Laser scanning, for example, determines depth D by measuring the time-of-flight Δt of a laser pulse, using $D = \frac{c \cdot \Delta t}{2}$, where c is the speed of light [60]. This method is highly accurate and widely used in industrial applications, though it can be expensive and cumbersome for large-scale or outdoor environments [61]. Structured light systems project a known pattern onto the scene and compute depth by analyzing pattern displacement, offering a balance between accuracy and practicality for 3D scanning and motion capture applications [62]. These active methods are accurate but require specialized hardware and can be sensitive to lighting conditions.

2.1.2 Camera Motion Estimation

Camera motion is estimated by integrating several camera poses over time. Camera pose estimation, which involves determining the translation, t, and orientation, R of a camera relative to a scene or the previous camera position, is a fundamental problem in computer vision with applications in robotics, augmented reality, and 3D reconstruction. Traditional methods for camera pose estimation, while groundbreaking, face various challenges in

Method	Functionality	Challenges
Stereo Vision [44, 45, 46]	Estimates depth by matching corresponding points between stereo images.	Struggles with low texture, repetitive patterns, and occluded regions. Global optimization techniques like SGM and Graph Cuts mitigate but do not eliminate these issues.
Photometric Stereo [52]	Recovers surface details by analyzing lighting variations across images.	Assumes Lambertian surfaces and controlled lighting, leading to errors with non-Lambertian surfaces, shadows, and varying albedo.
Depth from Focus/Defocus [55]	Estimates depth by analyzing image sharpness or blur, which is related to the distance from the camera.	Requires multiple images with varying focus settings, complicating use in dynamic scenes. Sensitive to noise and requires high-quality optics.
Active Methods [62]	Uses projected patterns or laser scanning to directly measure depth.	Requires specialized hardware and controlled environments. Not suitable for uncontrolled or outdoor settings. Active illumination limits use in power-sensitive or light-sensitive environments.

Table 2.1: Summary of traditional depth estimation methods, their functionalities, and associated challenges.

complex, real-world scenarios. This section explores these methods, detailing their mathematical foundations and limitations and setting the stage for the need for deep learning-based approaches.

One of the earliest methods for camera pose estimation is the Direct Linear Transformation (DLT) [16]. DLT involves solving a set of linear equations to determine the camera's projection matrix, which encodes both the intrinsics (e.g., focal length, principal point) and extrinsic parameters (rotation and translation) such as translation and rotation. The projection matrix P maps 3D world points, with a slight abuse of notation, $X = (X, Y, Z, 1)^\top$ to their corresponding 2D image points $x = (u, v, 1)^\top$ through a linear transformation

$x = PX$. Given a set of corresponding 3D-2D point pairs, DLT forms a system of linear equations, which can be solved using singular value decomposition (SVD) to estimate P. Despite its historical significance, DLT assumes that all points are exact without considering measurement noise, which limits its robustness. DLT's reliance on solving a linear system makes it sensitive to noise and numerical instability, leading to significant errors in the estimated pose. Moreover, DLT does not inherently enforce the orthogonality of the rotation matrix within P, potentially resulting in non-physical solutions. These challenges necessitate additional steps, such as normalization of points and post-processing to enforce valid rotation matrices, but these adjustments are not always sufficient to overcome the method's inherent limitations.

Building on the limitations of DLT, the Perspective-n-Point (PnP) problem involves determining the pose of a camera given a set of 3D points in the world and their corresponding 2D projections in the image [63]. Given n 3D points $X_i = (X_i, Y_i, Z_i, 1)^\top$ and their corresponding 2D projections $x_i = (u_i, v_i, 1)^\top$, the goal is to find the rotation R and translation t that satisfy $\lambda x_i = K[R|t]X_i^\top$, where K is the camera intrinsic matrix, and λ is a scale factor. The problem is typically solved by minimizing the re-projection error, which is the difference between the observed 2D points and the projected 3D points. Several variants of the PnP algorithm have been developed to improve accuracy and computational efficiency. For example, the Efficient PnP (EPnP) [63] algorithm uses an efficient parametrization of the problem to reduce computational complexity, making it suitable for real-time applications. However, PnP methods remain sensitive to outliers in the point correspondences, which can significantly degrade the quality of the pose estimation.

Another method in camera pose estimation is the Essential Matrix estimation [64, 65], which captures the relationship between two scene views, specifically encoding the rotation and translation between two camera positions. It is fundamental to both structure from motion and stereo vision when the intrinsics of the camera parameters are known. Given a pair of corresponding points x_1 and x_2 in two images taken from different camera positions, the essential matrix E relates these points through the epipolar constraint $x_2^\top E x_1 = 0$. The essential matrix E can be decomposed as $E = [t]_\times R$, with R representing the rotation and $[t]_\times$ being the skew-symmetric matrix of the translation vector t. The essential matrix can be estimated using at least five-point correspondences (the five-point algorithm) [64], though more correspondences typically improve robustness. However, estimating the essential matrix accurately requires precise point correspondences and the presence of noise or outliers can lead to significant errors. Additionally, the decomposition of E into R and t is ambiguous due to the inherent scale ambiguity in the translation vector and the potential for multiple valid decompositions.

The Kabsch algorithm [66, 67] is another important method for calculating the optimal rotation matrix that minimizes the root mean square deviation (RMSD) between two sets of points. This algorithm is particularly useful in scenarios where one needs to align two sets of corresponding 3D points, a common task in pose estimation. Given two sets of corresponding points $\mathbf{P} = \{X_1, X_2, \ldots, X_n\}$ and $\mathbf{Q} = \{\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_n\}$, where each X_i corresponds to $\tilde{X}_i$, the Kabsch algorithm finds the rotation matrix R that minimizes the sum of squared distances between the points. The algorithm proceeds by first centring the point sets by subtracting their respective centroids, resulting, $\mathbf{P}'\,\mathbf{Q}'$, and then finding the optimal rotation matrix R by computing the singular value decomposition (SVD) of the covariance matrix $H = \mathbf{P}'^{\top}\mathbf{Q}'$. The rotation matrix R is given by $R = VU^{\top}$. If the determinant of R is negative, which would correspond to a reflection rather than a proper rotation, the sign of the last column of V is inverted to ensure a proper rotation matrix. While the Kabsch algorithm is robust for calculating rotation, it does not handle translation, and it assumes that the two sets of points are clean and free of outliers, limiting its applicability in more complex scenarios.

The Iterative Closest Point (ICP) [60, 68] algorithm is widely used for refining pose estimation by aligning a set of 3D points (usually obtained from a depth sensor or previous estimation) to a reference model. ICP does not require explicit correspondence estimation between the target and the reference models, making it a versatile tool in pose estimation. ICP iteratively refines the pose $\{R, t\}$ by minimizing the distance between corresponding points in two 3D point clouds. The algorithm alternates between finding the closest points (correspondences) between the two sets and solving for the transformation $\{R, t\}$ that minimizes the error, repeating the process until convergence. However, ICP assumes that a good initial estimate of the pose is available, as it is prone to converging to local minima, especially when the initial alignment is poor. Additionally, ICP relies on the closest point heuristic, which may not be accurate in the presence of noise, occlusions, or partial overlaps between the point clouds. Variants like point-to-plane ICP [69] and robust ICP [70] introduce additional constraints or outlier rejection strategies to mitigate these issues.

Finally, Bundle Adjustment [71] is a comprehensive method used to refine the 3D structure and camera poses by minimizing the overall re-projection error across multiple images. Bundle adjustment simultaneously optimizes the camera parameters (intrinsic and extrinsic) and the 3D structure by minimising the re-projection error. The optimization process typically uses nonlinear methods like Levenberg-Marquardt to iteratively adjust the camera poses and 3D points to reduce the overall error. However, bundle adjustment is computationally intensive, especially as the number of images and 3D points increases.

Method	Key Characteristics	Limitations
Direct Linear Transformation [65]	Solves for the camera projection matrix P using linear equations and SVD.	Sensitive to noise, does not enforce orthogonality in rotation matrix, may result in non-physical solutions.
Perspective-n-Point (PnP) [63]	Estimates camera pose using a set of known 3D points and their 2D projections.	Sensitive to outliers, requires good point correspondences, may struggle with complex scenes.
Essential Matrix Estimation [64, 65]	Encodes the relationship between two views, decomposed into rotation R and translation t.	Sensitive to noise, ambiguity in decomposition, relies on precise point correspondences.
Kabsch Algorithm [66, 67]	Calculates the optimal rotation matrix to align two sets of 3D points.	Does not estimate translation, assumes clean data without outliers.
Iterative Closest Point (ICP) [60, 68]	Aligns two 3D point clouds iteratively to refine pose.	Prone to local minima, requires good initial alignment, sensitive to noise and partial overlaps.
Bundle Adjustment [71]	Optimizes camera parameters and 3D structure by minimizing overall re-projection error.	Computationally intensive, slow convergence, sensitive to initial estimates, not suitable for real-time.

Table 2.2: Summary of traditional camera motion estimation methods

The high dimensionality of the optimization problem can lead to slow convergence, and the process is sensitive to the initial estimates of camera poses and 3D points.

The summary of these methods is shown in Table 2.2 with their potential limitation. All these methods assume that the correspondences between pixel to pixel or pixel to point or point to point exist and are accurate. However, it is not trivial to find an interesting point such that it could be found repeatedly in the other observations of the scene. A large body of literature is there to find these reliable, interesting points that could be tracked between the frames. A few notable traditional works in this direction are discussed below.

2.1.2.1 Keypoint Detection Methods

Keypoint detection is a fundamental task in computer vision, enabling various applications such as image matching, object recognition, and 3D reconstruction. The goal is to detect salient points in an image, known as keypoints, which are distinctive and robust to changes in scale, rotation, or illumination. Several methods have been proposed to address the challenge of detecting keypoints under different conditions, and this section provides an overview of some of the most well-known keypoint detection techniques.

One of the earliest and most widely used techniques is the Harris Corner Detector [72], which is designed to identify corners in an image. Harris corners are detected by analyzing the intensity variations in two orthogonal directions, and a response function, R, is computed to determine the presence of a corner. The response is given by $R = \det(M) - k(\text{trace}(M))^2$, where M is the second-moment matrix computed from image gradients, and k is a constant (typically $k = 0.04$). A large positive value of R indicates a corner, while negative or small values suggest an edge or a flat region. Although the Harris detector is invariant to rotation, it lacks scale invariance, which limits its applicability when significant changes in image scale are present.

The Scale-Invariant Feature Transform (SIFT) [10] is a more advanced method that addresses this issue by offering both scale and rotation invariance. SIFT detects keypoints by constructing a scale-space representation of the image, achieved through Gaussian filtering at multiple scales. Keypoints are identified as local extrema of the Difference of Gaussians (DoG) across these scales. Once a keypoint is detected, its location is refined for stability, and an orientation is assigned based on the direction of the local image gradient around the keypoint. This process ensures the keypoints are invariant to image rotation. A descriptor is then generated by analyzing the local gradients in the surrounding region of the keypoint, creating a discriminative feature vector. Despite its robustness, SIFT is computationally expensive due to its multi-step process, making it slower compared to other methods.

To improve computational efficiency, the Speeded-Up Robust Features (SURF) algorithm was introduced [11]. SURF approximates the Gaussian filtering used in SIFT by employing box filters, which can be computed efficiently using integral images. Keypoints are detected based on the determinant of the Hessian matrix, $\det(H)$, which measures local intensity variations. Like SIFT, SURF is both scale- and rotation-invariant, and it computes a descriptor by analyzing local gradients. However, SURF uses Haar wavelet responses to create a more compact descriptor. As a result, SURF is faster than SIFT but may sacrifice some accuracy in terms of keypoint matching performance.

For real-time applications where speed is critical, the Oriented FAST and Rotated BRIEF (ORB) algorithm [73] offers an efficient alternative. ORB combines the FAST (Features

from Accelerated Segment Test) keypoint detector [74] with the BRIEF (Binary Robust Independent Elementary Features) descriptor [75]. FAST detects keypoints by evaluating the intensity differences between a pixel and its circular neighbourhood, making it extremely fast. BRIEF creates a descriptor by encoding local image patches as binary strings, which is computationally efficient. ORB improves BRIEF by introducing rotation invariance through a modified descriptor known as Rotated BRIEF. This method is well-suited for real-time tasks such as mobile robotics or augmented reality, though the keypoints generated are less distinctive compared to those from SIFT or SURF.

A more recent method, KAZE, employs a nonlinear scale-space approach to keypoint detection [76]. Unlike SIFT and SURF, which rely on a linear scale-space representation, KAZE uses nonlinear diffusion filtering, which preserves more detailed image structures, particularly in textured regions. Keypoints are detected as extrema in this nonlinear scale-space, and a descriptor is computed by analyzing the local image gradients around each keypoint, similar to SIFT. The Accelerated KAZE (AKAZE) [77] variant enhances the speed of KAZE by using faster filtering techniques, making it more practical for real-time applications where both accuracy and speed are important.

All these methods are summarized with potential limitations in Table 2.3. Each of these methods—Harris, SIFT, SURF, ORB, and KAZE—offers different trade-offs in terms of speed, accuracy, and robustness. While SIFT provides highly distinctive and robust keypoints, it is computationally intensive. SURF offers a faster alternative at the expense of some accuracy, while ORB is ideal for real-time applications but may not perform as well in terms of matching precision. KAZE and AKAZE strike a balance between speed and accuracy, especially in scenarios with rich textures.

2.1.3 Joint Depth and Camera Motion Estimation

Joint estimation of depth and camera motion leverages the interdependence between these two tasks, providing complementary information to improve both. When estimating depth alone, ambiguities arise due to the lack of motion cues, making it difficult to resolve scale, occlusions, and texture-less regions [15, 78]. Similarly, ego-motion estimation without detailed scene geometry often results in inaccuracies. By considering the two tasks together, depth estimation benefits from the geometric context provided by motion, while camera motion improves through a more detailed understanding of the structure of the scene [79].

Traditional methods such as Structure from Motion (SfM) tackle this problem by reconstructing the 3D structure of a scene and estimating camera motion based on correspondences between multiple views. SfM tracks feature points across images and seeks to minimize the

Method	Key Characteristics	Limitations
Harris Corner Detector [72]	Detects corners based on intensity variations in orthogonal directions.	Not scale-invariant, sensitive to image scale changes.
Scale-Invariant Feature Transform (SIFT) [10]	Detects keypoints invariant to scale and rotation, and computes distinctive descriptors.	Computationally expensive, slower compared to other methods.
Speeded-Up Robust Features (SURF) [11]	Uses box filters for efficient Gaussian approximation, and computes Hessian-based keypoints.	May sacrifice some accuracy compared to SIFT, still computationally demanding.
Oriented FAST and Rotated BRIEF (ORB) [73, 74, 75]	Combines FAST keypoint detector with BRIEF descriptor, optimized for speed.	Less distinctive keypoints, not as robust as SIFT or SURF.
KAZE / Accelerated KAZE (AKAZE) [76, 77]	Employs nonlinear scale space for keypoint detection, and uses local gradients for descriptors.	More computationally demanding than ORB, not as fast as SURF in its original form.

Table 2.3: Summary of keypoint detection methods

re-projection error,

$$\sum_{j=1}^{m}\sum_{k=1}^{n}\left\|x_{jk}-\pi(P_jX_k)\right\|^2,$$

where m is the number of views, n is the number of points, x_{jk} is the observed point, P_j is the camera projection matrix, and X_k represents the 3D points. Optimization techniques such as bundle adjustment [71] iteratively refine camera motion $\{R_j, t_j\}$ and 3D structure X_k [16]. However, SfM is sensitive to errors in feature matching, assumes static environments, and suffers in dynamic scenes [80]. Incremental SfM [81] and robust methods like RANSAC [82] attempt to improve robustness to outliers and reduce computational load.

Visual Odometry (VO) estimates camera motion by tracking changes in the position of visual features over time [79]. The transformation between consecutive frames is estimated by minimizing the reprojection error

$$\sum_{k=1}^{n}\left\|x'_k-\pi(Rx_k+t)\right\|^2,$$

where R and t are the relative rotation and translation [83]. In monocular VO, depth is recovered via triangulation, while stereo VO uses stereo disparity to directly estimate depth [79]. However, monocular VO suffers from scale ambiguity, as it cannot determine the absolute scale of motion without additional cues. Drift is another common issue, where small errors accumulate over time, leading to inaccurate pose estimates [84]. Techniques such as keyframe-based VO, PTAM (Parallel Tracking and Mapping) [83] and SVO (Semi-direct Visual Odometry) [84] improve robustness, but challenges remain in texture-less environments and under significant motion blur.

Simultaneous Localization and Mapping (SLAM) extends VO by simultaneously building a map of the environment and estimating the motion of the camera [85, 86]. SLAM maintains an estimate of both the map $\mathcal{M}$ and the camera poses $\mathcal{X}$, aiming to maximize the posterior probability $P(\mathcal{X}, \mathcal{M}|\mathcal{Z}, \mathcal{U}) \propto P(\mathcal{Z}|\mathcal{X}, \mathcal{M})P(\mathcal{X}|\mathcal{U})$, where $\mathcal{Z}$ are image observations and $\mathcal{U}$ are control inputs [86]. Although SLAM can handle larger environments than VO, it faces challenges in dynamic scenes where moving objects lead to incorrect map updates [87]. Loop closure detection is critical for reducing drift, but it is computationally expensive, especially in large-scale environments [12, 88].

Direct methods directly operate on the images without extracting keypoints and descriptions to solve the problem of estimation depth and ego-motion in a direct way, hence the name. These methods estimate camera motion by minimizing the photometric error between consecutive frames instead of reprojection error. The photometric error,

$$\sum_{x} \|I_1(x) - I_2(W(x; R, t, D))\|^2 , \tag{2.1}$$

where I_1 and I_2 are pixel intensities, and $W(x; R, t, D)$ warps pixels using the estimated motion parameters, allows for the joint estimation of motion and depth [15]. While direct methods provide highly accurate results, they are sensitive to lighting changes, require good initial estimates, and are computationally expensive, especially for large-scale or high-resolution images [89]. Variants such as LSD-SLAM [15] and Direct Sparse Odometry (DSO) [90] have been developed to address these issues. This error has gained significant attention in recent years in deep learning to address this joint estimation of depth and ego-motion in a completely self-supervised manner.

Despite the success of traditional methods such as SfM, SLAM, and VO, they face significant challenges, including high computational costs, sensitivity to feature tracking, and difficulty handling dynamic scenes. Direct methods, though accurate, are computationally demanding and sensitive to photometric variations, which is one of the research challenges addressed in Chapter 3.

2.2 Deep Learning Methods

The unprecedented success of deep learning in directly learning from data and its ability to generalise has led to the development of numerous deep learning based depth and camera motion estimation methods. The contributions made in this thesis are fully if not partially related to deep learning based methods. These deep learning methods could be divided fundamentally into two types based on how they are trained: Supervised methods; that use ground-truth depth maps, and self-supervised methods; that use appearance and geometry based loss function similar to Equation 2.1. While the proposed contributions in our this thesis are self-supervised, the supervised methods played significant role in developing significant architectures, loss and regularization functions. In the following section, the supervised depth and camera-motion methods are explained and each followed by a short discussion about why the thesis favoured self-supervised methods.

2.2.1 Supervised Depth Estimation

Supervised depth estimation methods generally fall into three categories based on the input: 1) Monocular methods, where only a single image is used as input; 2) Stereo methods, where a pair of images from a calibrated stereo rig is used; and 3) Multi-View Stereo methods, where two or more images are used as input, typically, with known camera motion cues. In the supervised setting, given a dataset of images with corresponding per-pixel depth maps, a model is trained using a loss function that compares the predicted depth with the ground truth depth map.

2.2.1.1 Monocular Depth Estimation

Some notable methods and the progress in terms of structural recovery and estimating plausible depth from a single monocular image are shown in Figure 2.2. A major breakthrough was achieved by Eigen et al. [19], who introduced a multi-scale fusion model for depth regression. This was one of the first methods to apply deep learning to monocular depth estimation (MDE), using a convolutional neural network (CNN) to predict depth maps from a single image at multiple scales. This approach laid the foundation for future deep learning-based methods. Inspired by this, Liu et al. [20] introduced a deep convolutional neural field to learn the unary and pairwise potentials integrating the strengths of deep CNNs and continuous Conditional Random Field (CRFs) to effectively utilize the continuous nature of the depth. Meanwhile, Laina et al. [92] introduced the first full convolutional architecture, introducing upsampling layers trained with Huber loss, reducing the trainable parameters. Li et al. [93] proposed a method to further improve the structural details in the estimate depth

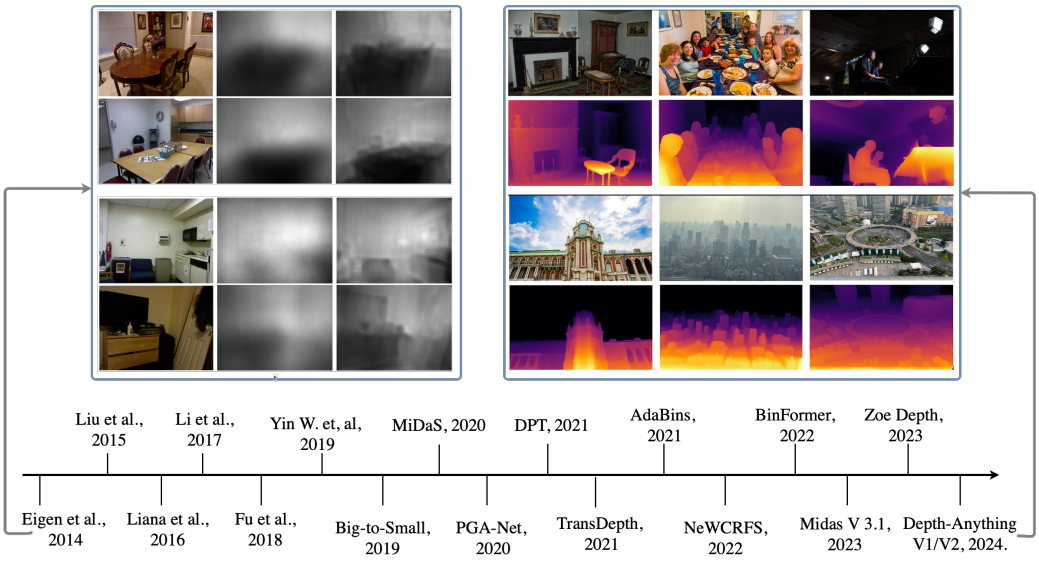


Figure 2.2: Comparison of supervised depth estimation progress over the last decade. The left side displays depth maps estimated by Eigen et al. [19], while the right side showcases more recent results from Depth-Anything [91]. Below, some notable methods in depth estimation from 2014 to 2024 are shown. These results are taken from their respective papers, and the goal here is to show how much progress has been made in recovering structural details accurately. In both depth images, darker colour represents higher depth.

maps by employing two networks: one for estimating the depth and another for estimating the depth gradients. These two information are further fused to estimate the final depth. Fu et al. [94] posed the depth estimation problem as an ordinal regression problem by training the network using ordinary regression loss using a multi-scale network. Similarly, Yin et al. [95] proposed a new loss function that enforces virtual normal directions in the estimated depth maps, further improving the depth estimation accuracy. Further, Big to Small by Lee et al. [96] introduced new local planar guidance layers into the depth decoder for more effective depth decoding from the down-scaled feature maps. Similarly, PGA-Net (Probabilistic Graph Attention Network) [97] extended Conditional Random Fields (CRFs) using probabilistic graph attention network to learn and fuse the multi-scale feature representations in contrast using simple concatenation or weighted average as done in earlier approaches. Later, the introduction of MiDaS by Ranftl et al. [98] marked a significant milestone in the field. MiDaS utilized an affine-invariant loss function, allowing the model to be trained on a diverse set of datasets with varying depth scales and shifts. This approach enabled robust multi-dataset training, significantly improving the model's generalization ability. Meanwhile, with the introduction of vision transformers (ViTs) [32] in 2020, DPT (Dense Prediction Transformer) [99] proposed an end-to-end transformer-based architecture for multiple tasks, including depth estimation. The ability of transformers to capture the global

context significantly boosted the depth estimation performance. Around the same time, TransDepth [100] argued that transformers lose local details in their layers, which convolutional neural networks were good. Therefore, they designed an architecture by fusing the transformer into the ResNet [101] to improve the depth estimation accuracy further. While everyone is treating the depth estimation as a continuous regression task, AdaBins [102] discretised the output depth values into adaptive bins (bin width is estimated per test image) and refined them using a transformer block before linearly combining them into the final output depth map. This method argues that previous methods perform global context fusion at the low-resolution feature level, but processing at the high resolution leads to better results. Again, the CRFs idea was revisited by NeWCRFs (Neural Window Fully Connected CRFs) [103] using transformers to bring in the global context while learning the potentials of CRF. Further, AdaBins was revisited by BinFormer [104] with a new architecture changing the decoder of AdaBins with a transformer-based decoder to calculate the adaptive width of the depth bins. Similarly, MiDas was improved with various versions that trained on more datasets, changed the architecture to transformers, and finally, in Midas-V3.1 [105], extended it to various transformer models leading to much better accuracy than the earlier versions. Even though these models were trained using a large number of datasets and better network architectures like vision transformers, they still suffered from generalising out of the trained datasets. ZoeDepth [106] points out the problem is coming from the depth-scale difference between the datasets (e.g., indoor and outdoor datasets). They address this problem by training the model for relative depth, proposed by [107], a scale agnostic quantity without any metric meaning, and further fine-tuning the results for metric depth using a domain-specific head (a latent classifier is used to decide the domain). This helped them to train the model on massive amounts of data and achieve unprecedented zero-shot generalization performance. Recently, the DepthAnything models, particularly V1 [91] and V2 [108], have made significant strides in monocular depth estimation by emphasizing the value of large-scale, diverse unlabeled data. DepthAnything-V1 introduced a data engine capable of automatically generating depth annotations for massive unlabeled datasets and trained with advanced perturbation techniques and feature alignment with powerful pre-trained models like DINO-V2 [33], significantly enhancing the model's generalization by expanding data coverage. Later, DepthAnything-V2 further trained different models with a much larger number of parameters than DepthAnything-V1 and on large amounts of synthetic data, leading to much better results.

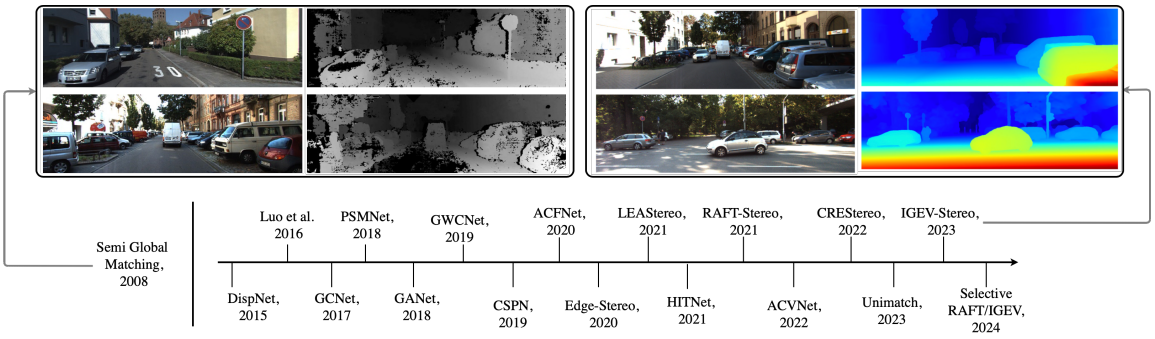


Figure 2.3: Comparison of Supervised Stereo Depth Estimation Progress (2015-2024) with pre-deeplearning method, SGM [109]. The figure on the left illustrates depth maps generated by a traditional method before the advent of deep learning, while the right side displays results from the state-of-the-art IGEV-Stereo [39]. Some notable methods in stereo depth estimation techniques over the last decade are shown below. The Semi-Global Matching results are sourced from the DispNet paper [110], and the more recent results from IGEV-Stereo [39]. The aim is to demonstrate the significant advancements in accurately recovering structural details. In both images, darker colors represent greater depth.

2.2.1.2 Stereo Depth Estimation

Stereo depth estimation involves establishing correspondences between the left and right images, which are then used to estimate depth. Figure 2.3 illustrates several notable methods and the progress made in structural recovery and depth estimation from stereo image pairs. To clearly highlight the advancements of the last decade, the figure includes a comparison with the well-known traditional method, SGM [111]. DispNet [110] was one of the pioneering deep-learning models for stereo-depth estimation. Introduced in 2016, DispNet used an end-to-end convolutional neural network (CNN) architecture, which directly regressed disparity maps from stereo image pairs. It demonstrated the potential of deep learning in stereo matching, outperforming traditional methods in speed and accuracy. Even though there was a huge jump in accuracy, and smoothness of the estimated disparities, the inference speed was still slow. Motivated by this, Luo et al. [112] proposed an efficient stereo-matching algorithm by introducing an inner product layer into their architecture to improve the speed while keeping the accuracy. GCNet (Geometry and Context Network) [113] presents a deep learning model for estimating disparity from stereo images. They create a cost volume using deep features and apply 3D convolutions to capture context. Disparity is then estimated using a soft argmin operation, allowing end-to-end training with sub-pixel accuracy, without needing extra post-processing. PSMStereo (Pyramid Stereo Matching Network) [114] extended the idea of GCNet by incorporating spatial pyramid pooling [115]

and a stacked hourglass architecture. This design allowed PSMStereo to capture global context and fine details, improving performance in challenging scenarios such as large texture-less regions. GANet (Guided Aggregation Network) [116] propose two novel neural network layers for cost aggregation: a semi-global aggregation layer that approximates semi-global matching, and a local guided aggregation layer that refines thin structures using traditional cost filtering. These layers can replace the computationally expensive 3D convolutional layers, significantly reducing complexity. This was further improved by GWCNet (Group-Wise Correlation Network) [117] with group-wise correlation where feature groups from the left and right images are correlated to create efficient matching cost proposals without significant information loss. This method, combined with an improved 3D stacked hourglass network, enhances performance and reduces computational costs compared to previous approaches. In ACFNet (Adaptive Unimodal Cost Volume Filtering Network) [118], the authors argue that directly learning with disparity regression loss can leave the cost volume under-constrained. To address this, they propose constraining the cost volume by filtering it with an unimodal distribution, ensuring more robust and accurate disparity estimation. Later, Edge-Stereo [119] tackled the problem of disparity estimation in challenging areas such as texture less regions, small objects and boundaries with a multi-task learning network combining disparity estimation with edge detection branch demonstrating edge detection and stereo matching can mutually enhance each other. At the same time, LEAStereo (Learning Effective Architecture Stereo) [120] introduced a search-based neural architecture search (NAS) method to design efficient and accurate stereo networks. LEAStereo achieved a good balance between accuracy and computational efficiency, making it suitable for real-time applications. Further, HITNet (Hierarchical Iterative Tile Refinement Network) [121] proposed a novel real-time stereo matching network that avoids explicit cost volume construction and 3D convolutions, instead using multi-resolution initialization, 2D geometric propagation, and warping to infer disparities. The architecture's multi-resolution design and slanted plane hypothesis inference achieve high accuracy with significantly less computation. Later, AANet (Adaptive Aggregation Network) [122] addresses the challenge of high computational cost in stereo matching by proposing a method to replace 3D convolutions, aiming for faster inference while maintaining accuracy. This method, significantly accelerates existing models, achieving up to $41\times$ speed improvements over GCNet, $4\times$ over PSMNet, and $38\times$ over GA-Net. CFNet (Cascade and Fused Cost Volume Network) [123] proposes to fuse multiple low-resolution dense cost volumes to increase the receptive field and propose to refine the disparity in a coarse-to-fine manner. The authors argued that these helped them to mitigate the problems of large domain differences and unbalanced disparity distributions across the datasets which were problems for generalisation.

ACVNet [124] introduces an attention concatenation volume (ACV) that generates attention weights from correlation clues to enhance relevant information and suppress redundancy in stereo matching. This method, significantly improved the accuracy, using only 1/25th of the parameters compared to previous methods like GwcNet. CREStereo (Cascaded REcurrent Stereo matching network) [125] introduces a hierarchical stereo matching network with recurrent refinement and a stacked cascaded architecture to improve fine depth details, along with an adaptive group correlation layer to address rectification errors. Additionally, we present a new synthetic dataset focused on challenging cases to enhance generalization to real-world scenes. RAFT-Stereo (Recurrent All-Pairs Field Transforms) [126] is an efficient and accurate stereo matching network that iteratively refines disparity estimates using a Convolutional Gated Recurrent Units (conv-GRUs), leveraging multi-level correlation volumes to guide the refinement. It builds on the RAFT [127] architecture from optical flow and replaces costly 3D convolutions and large cost volumes. However, this method lacked non-local context, which was later addressed by IGEV-Stereo (Iterative Geometry Encoding Volume) [39] using a combined geometric encoding volume (CGEV) to encode geometry and context. This method, however, reintroduced a 3D convolution network at the cost of increased accuracy. Another approach, Unimatch [128], attempted to address the non-local/global context problem by using coarse-to-fine correlation volumes with an iterative refinement module similar to RAFT-Stereo. Additionally, this method was trained on multiple tasks, such as multi-view-based depth and optical flow estimation, using the features extracted from the same encoder. The authors argued that training the model on all these tasks resulted in a more robust and generalisable model. More recently, to alleviate the problem of inaccurate disparity estimation in texture-less regions, Selective-Stereo [129] proposed Selective Recurrent Unit (SRU) helps the network to aggregate hidden disparity information across multiple frequencies, mitigating the disparity information loss during iterative processes of models like RAFT-Stereo and IGEV-Stereo.

2.2.1.3 Multi-View Depth Estimation

Besides monocular and stereo methods, some approaches that are referred to as Multi-View methods, perform depth estimation using two or more images. Stereo matching could be considered a special case of multi-view stereo, where the cameras are perfectly aligned (horizontally), and images are calibrated with stereo rectification, allowing for a 1D line search for corresponding pixels. With the rise of deep learning, many approaches have been introduced in this direction, alleviating the problems of traditional methods. The majority of these approaches first estimate the camera motion, often using sparse keypoint matching, and then use that camera motion and the sparse 3D point cloud extracted after bundle

adjustment to perform a dense correspondence search to estimate per-pixel depth maps. However, estimating camera motion is not trivial, and noise in this estimate can propagate downstream. To address this, DeMoN (Depth and Motion Network) [130] proposed the first deep learning approach to estimate depth from unconstrained pairs of images. However, the model was constrained to only two input images. Extending this to an arbitrary number of views, MVSNet (Multi-View Stereo Network) [131] proposed constructing cost volumes of features using the homography matrix estimated using the known camera extrinsics and employed 3D convolutions to interpret the cost volume, further refining the estimates with the input image as a reference. Meanwhile, Clark et al. [132] proposed a neural non-linear optimization function to address the difficulties in optimizing the Sum of Squared Differences (SSD) objective. This method required no handcrafted features or priors that were usually employed earlier to make an ill-posed problem well-posed. When tested for this proposed optimization for multi-view stereopsis, the algorithm resulted in improved performance with approximately $3\times$ fewer parameters compared to the nearest baseline, DeMoN [130]. While the methods at the time were directly regressing the depth, DPSNet (Deep Plane Sweep Network) [133] reintroduced a differentiable version of the traditional geometry-based approach, the plane sweep algorithm [134], to build the cost volume using deep features instead of images. This cost volume was further used for regressing dense depth maps. Until this point, unconstrained methods were underperforming compared to constrained methods (methods that use ground truth camera poses). To mitigate this, DeepV2D (Deep Video to Depth) [135] proposed optimizing both depth and motion (an initial pose guess was needed though) in a single framework with a differentiable plane sweeping technique for depth estimation. Further, DeepVideo-MVS [136] inserted Conv-LSTMs [137] into the bottleneck of the feature extractors to embed knowledge from the previous frames into the current one. This simple solution significantly improved depth accuracy with little computational overhead. With a similar motivation of incorporating temporal information, ESTDepth (Epipolar Spatio-Temporal Depth Networks) [138] incorporated a transformer [139] architecture to integrate spatial and temporal geometric context into the estimated cost volumes without the use of computationally heavy 3D convolutional networks, which were a standard method to learn from the cost volumes at the time. In the same spirit of lightweight networks, SimpleRecon [140] proposed replacing 3D convolutional networks with 2D CNNs using an innovative approach. In this approach, they first calculate the cost volumes using plane sweeping and then introduce a lot of additional priors (meta-data) into the cost volume features, then they process this cost volume through a Multi-Layer Perceptron (MLP) to get 3D cost volumes, which are then fused with features extracted from a pre-trained image encoder to estimate depth at multiple scales. This simple approach was found to be fast

and accurate. Similar to RAFT-Stereo in stereo depth estimation, CER-MVS (Cascaded Epipolar RAFT) [141] proposed extending the RAFT architecture to improve depth estimation. Further, Unimatch [128] proposed using a transformer architecture to perform self and cross-feature correlation between images and trained these features with other related tasks such as optical flow and stereo-matching to further improve the features. These features are further used to calculate the weights required to linearly combine the depth planes to estimate the final depth.

Undoubtedly, there has been remarkable progress made in estimating depth from both monocular and stereo-images. Especially, very recently, methods like Depth-Anything [91], UniMatch-Stereo [128] significantly boosted the generalisation capabilities of monocular and stereo models respectively. The only problem with these models is that they require ground truth depth maps to train. In a case where the performance of these methods does not match the expected accuracy or robustness or generalisation, there is no way but to collect the ground truth depth to fine-tune these models. However, collecting ground truth depth data is not trivial. It is costly, cumbersome and sometimes practically not possible. A similar situation of this concern recently observed in works that use Depth-Anything models on estimating consistent depth maps on video-data [142] on which the performance of these methods turned out to be sub-optimal making the researchers to look for methods alternative loss functions for temporal coherence. Another observation was made, in our Chapter 4, where we found that the state of the art stereo methods, IGEV-Stereo [39], Unimatch-Stereo [128] fail miserably when tested on poorly illuminated data. On the contrary, *self-supervised methods* bypass the need for costly ground-truth labels and allow for scalable training on vast amounts of unlabeled data using geometric consistency by enforcing constraints between multiple views of the same scene.

2.2.2 Supervised Camera Motion Estimation

The powerful ability of deep learning methods to automatically learn features from data, without manual feature engineering, has made them an attractive solution for addressing the problem of camera-motion estimation. A typical illustration of estimating pose directly from the images using a deep network is shown in Figure 2.4. Typically, given a pair of images as input, the goal is to design a method that estimates the 6-DoF motion of the camera. Similar to the depth estimation literature, many supervised and self-supervised methods have been developed to tackle this problem. The details of these methods are elaborated on in the following sections.

The approach here is conceptually similar to depth estimation. Given a dataset of image pairs along with their corresponding ground truth camera motions, the task is to design a

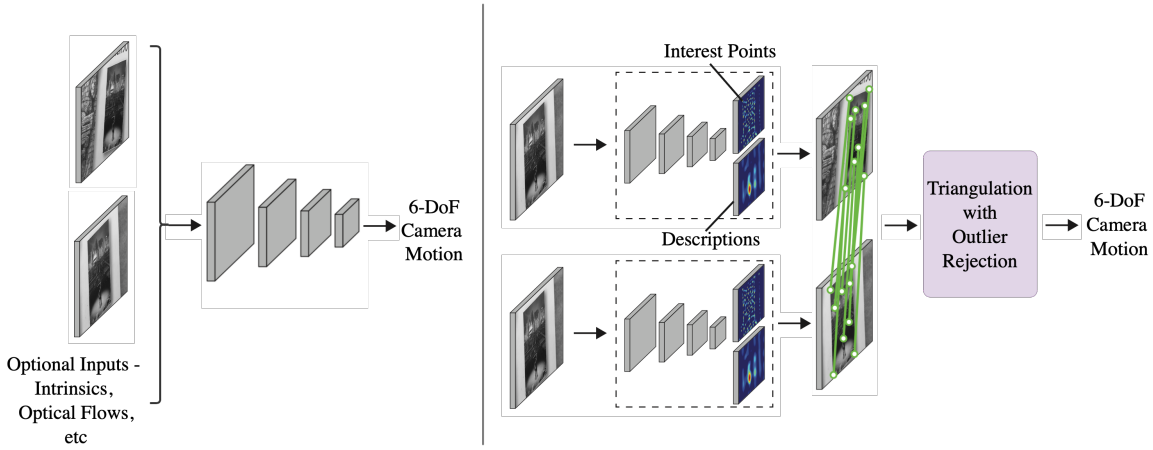


Figure 2.4: Illustration of two distinct deep learning approaches for camera motion estimation. The left side represents a typical end-to-end deep learning method, while the right side illustrates hybrid methods, where camera motion is estimated using point correspondences. Parts of this illustration are adapted from the SuperPoint [143] paper.

neural network that learns to estimate ego-motion by minimizing a loss function based on the difference between the estimated and ground truth motions. There are multiple choices for inputs, network design, and loss functions.

For instance, DeepVO [144] proposed a method that utilizes a CNN-based feature extractor coupled with an LSTM-based decoder to maintain a memory of previous frames while estimating the pose between the current and preceding frames. Another method, VINet [145], which fuses Inertial Measurement Unit (IMU) data and image data at an intermediate feature representation level. This fusion aims to address issues such as IMU synchronization with the camera and trajectory drift (where small pose errors, when integrated into a trajectory, can result in significant deviations from the ground truth trajectory over time). Later, Tartan-VO [146] trained a model conditioned on camera intrinsics using a synthetic dataset with numerous image sequences and their corresponding camera motion ground truth labels. This method significantly outperformed DeepVO [147] in both accuracy and generalization to new scenes. However, requiring intrinsics during inference limited its general applicability. Later, XVO [148] aimed to create a more generalized model by relaxing the intrinsics assumption and training on various auxiliary audio and vision tasks alongside visual odometry prediction to enhance robustness. Similarly, DROID-VO, part of DROID-SLAM [149], incorporated auxiliary supervision with ground truth depth and optical flow during training to improve the accuracy of estimation.

Despite the improvements in generalization and robustness achieved by supervised methods, their dependency on camera intrinsics, ground truth data for training, and additional

ground truth auxiliary data to enhance generalization and robustness limits their wider applicability as a standalone solution. As emphasized by DeepVO [144], these methods are not intended to replace traditional geometric methods but could potentially complement them in camera-pose estimation. The following section explains how deep learning has advanced geometry-based methods for pose estimation.

2.2.2.1 Keypoint Detection and Matching Methods

Deep learning methods excel at learning features that are robust to various nuisances, such as illumination changes, noise, and blur. However, regressing the 6-DoF pose from these features is still found to be challenging. On the other hand, traditional geometry-based methods that rely only on correspondence estimation are highly accurate in pose estimation but struggle with creating feature descriptions that are robust to these nuisances so they can be matched between images. In response, several methods [143, 150, 42, 151] have emerged that use deep learning to estimate accurate correspondences, which are then utilized in a traditional pose estimation pipeline for precise pose estimation, making them hybrid of both traditional and deep learning methods. A typical illustration of this method is shown in Figure 2.4.

These hybrid methods can be further divided into two categories: sparse and dense methods, depending on how correspondences are established. In sparse methods, keypoints are detected individually from images and matched using their descriptors. In contrast, dense methods bypass this step, performing coarse-to-fine feature matching instead. In the rest of this section, sparse methods are described first followed by dense methods. There is no universal definition of what qualifies a pixel to be a keypoint. Ideally, detected keypoints should be *salient* to be well described and *repeatable* to facilitate easy matching. Traditionally, keypoints are detected first, and for each keypoint, a patch around them is selected to describe that keypoint using handcrafted features. With the rise of deep learning, these steps were contested and these two questions were raised: 1. Should we detect and describe like traditional methods? 2. Because we can learn the descriptions for better matching, should we learn the descriptions first and then detect the keypoints? These questions further raised these questions: 1. Should we learn to do detection and description jointly or in a decoupled fashion? 2. If we are training, how do we supervise these detections and descriptions? 3. Do we need to train to detect at all?

In 2015, TILDE (Temporally Invariant Learned DEtector) [152] proposed the first learning-based detector using CNN and showed a significant improvement in detecting robust repeatable keypoints under challenging conditions such as weather and illumination. This method proposed to regress for a score map which could be used to detect keypoints

using non-maximum suppression during the inference. Later, Lenc et al. [153] proposed a covariance constraint to assign the keypoints to the image structures (i.e. corners). This approach was further extended by Zhang et al. [154] by introducing new concepts: standard patch and canonical feature to further improve the keypoint detection. Meanwhile, Quad-Networks [155], raises the question of what is an interesting point and trains a network using corresponding patches and a ranking-based criterion in a completely unsupervised way to detect keypoints. All these approaches process the patches of the image separately as input which made them slow during the inference. MagicPoint [143], the base detector of Superpoint [143], addressed the problems of speed, image patches, and the supervision of keypoints, by trained a fully convolutional network that inputs a full image outputs a score-map similar to [156] and trained using a large synthetic dataset of corners which resulted in real-time performance. Further, Key.Net [157] brought back the handcrafted filters and fused them with learnable CNN filters to train a more repeatable keypoint detection network. While all these methods concentrate purely on detecting only keypoints, some methods concentrate on learning local descriptions. HardNet [158] proposed a metric learning-based loss function with inspiration from Lowe's criterion [10] that pushes the closest negative from the closest-positive in the description space to training a CNN. Extending this metric learning approach with Second Order Similarity (SOS), SOSNet [159] proposed a new local descriptor learning method that performed better than HardNet. All the aforementioned methods only concentrated on learning either detectors or descriptions [1]. Combining these two tasks into a single framework, LIFT (Learned Invariant Feature Transform) [156] proposed a learnable method inspired by SIFT [69] combing all, detection, orientation estimation, and description into a single framework. LFNet [160] proposed to use the depth and motion cues (generated using depth sensors for indoors and using SfM for outdoors) to train a network that does detection first and then extracts the description of the patches around the detections. These methods followed the path of detecting first and describing later. However, in the later years, these methods were significantly outperformed by methods that describe and then detect.

Superpoint [143], proposed to extend MagicPoint with a descriptor network as an end-to-end joint learning method. The method is trained using random homographies generated on the fly for both detections using a cross entropy loss and descriptors using the triplet margin loss simultaneously. D2-Net [151] diverges from traditional score map-based keyoint estimation as done in [143, 152], they learn the descriptors in such a way that the keypoints could be estimated using a local-maxima criterion in spatial and channel-wise descriptor

[1] MagicPoint was proposed in SuperPoint [143] which learned both detections and descriptions. For better readability and explainability, it is mentioned here.

dimensions. R2D2 (Repeatable and Reliable Detector and Descriptor) [161] argued that saliency-based score maps as used by SuperPoint [143] are not necessarily discriminative and could harm the descriptor learning. Instead, encourage descriptors in places where the matching could be performed with high confidence. To enforce this they proposed to train their method with a predictor of the local descriptor discriminativeness to further improve the repeatability and matching performances. DISK (DIScrete Keypoints) [42] argued that despite all the advances in the deep learning-based method at the time, they were found to be working just on par or sometimes underperform than the traditional handcrafted methods when used in a downstream task of 3D reconstruction and they attributed this to the difficulty in the training objectives not being differentiable enough to correspond with the downstream task. To address the discreteness that comes with keypoints, they proposed a probabilistic framework that uses policy gradient-based learning from the Reinforcement Learning (RL) literature. ASLFeat (Accurate Shape and Localization Features) [162] argued that the methods at the time were neglecting to account for the shape properties such as scale and orientation. They proposed a method that uses deformable convolutions [163] to model the geometric details and detect keypoints at multiple scales. Meanwhile, addressing the differentiability problem of keypoint detection, ALIKE [164] and ALIKED [165] introduced another lightweight framework with a differentiable keypoint detection module. Very recently, SILK [166] redesigned the entire framework, from network architecture to the loss function, creating a fully differentiable, lightweight, and flexible keypoint detection network.

Despite these advancements, the aforementioned methods aggregate only local image information, typically within an $n \times n$ pixel patch, to estimate feature descriptions. This approach struggles when there are significant changes in scale, viewpoint, or repetitive patterns in the image—situations that require global context. With the success of transformer architectures, methods like SuperGlue [150] and LightGlue [167] have been developed to incorporate global context into keypoint descriptions. However, these methods primarily extend existing keypoint detection techniques to enhance description quality.

A significant limitation of these keypoint detection methods, even those utilizing transformers, arises when there is insufficient texture in the image to locate keypoints. This challenge is further addressed by dense feature matching methods such as LoFTR [168] and Efficient-LoFTR [169], where the problem is framed as a coarse-to-fine matching task. In these methods, coarse and fine feature maps are extracted for each image. Coarse feature maps are matched across all possible pairs, and once coarse matches are established, the match locations are upsampled and refined within a window centered around the coarse

match location. There are also server other methods such as MatchFormer [170], APSAN-Former [171], DKM [172], ROMA [173], etc that further improved matching using the transformer architectures. While these methods improved the matching performance, the technical concepts are out of the scope of this thesis. Please find their papers for more details.

All the methods mentioned above, including the transformer-based methods, except for the describe-to-detect methods, are trained using ground truth correspondences. These correspondences are typically generated using either homographic adaptation, where homography is employed to match pixels between images, or correspondences derived from traditional Structure from Motion methods like COLMAP [174]. Homographies can only account for 2D variations in the data, while methods like COLMAP, despite being more robust, inherit the challenges of traditional methods and require computationally intensive structure building by estimating depth and pose beforehand. Pointing out this problem, PoSFeat (Camera Pose Supervised Feature) [175] proposed a weakly supervised learning framework to train the network using only camera motion. They further found that joint training of detectors and descriptors was hard to train, so they had to decouple them to train. A similar observation was made before by LIFT [156] when training detectors and descriptors jointly. Another problem with the joint learning of detector and descriptor is that they get co-dependent meaning that they underperform when one is changed. Motivated by these observations, DeDoDe [40] recently proposed a method by training a method in a decoupled fashion that significantly improved over all the joint learning methods and our-performing a transformer-based method SuperGlue [150]. However, this method relies on a base detector to give it a good initial guess and suffers from repeatability issues when sampled in smaller numbers. Despite all the best efforts, previously, it was also found that the generalization of these supervised learning methods was limited to their trained domain and underperforms a traditional method with proper settings when applied to a new domain. We have also shown an example of keypoint detections and their matches in Figure 1.4. When this method was tested for keypoint detection and matching in low-light conditions, many keypoints were detected on the road, showing the variations of the light caused by the headlights of the car as keypoints, in the sky (affected greatly by the low Signal to Noise Ratio), and on street lights. When matched with a nearby image (approx. 5m away), the model was only able to establish a few correspondences including a lot of outlier matches. This is clear evidence of how the SOTA keypoint detection and matching systems fail to generalize and be robust to new and poorly illuminated scenarios.

While the majority of the literature is looking at jointly learning the detections and descriptions, there are a few methods advocated for complete learning-free detection and

descriptions. ELF [176] proposed a method to use pretrained image encoders to reverse compute the gradients from features to images, under the assumption that a well-trained encoder would highlight the salient regions of the image. However, this approach is computationally expensive as it requires backpropagation for every single image. Later, D2D [177] utilized pretrained local feature descriptors, HardNet [158], and SOSNet [159] with a new criterion for detecting keypoints. This method proposed to measure the saliency (a measure to decide a feature is a keypoint) based on the entropy of that feature. While estimating entropy is very expensive, they used a proxy measure by considering the features are from an unimodal Gaussian distribution and the standard deviation as a proxy measure of entropy. This method outperformed the former two categories of methods from detect then describe and describe and detect categories, at the time. However, according to R2D2 [161] saliency itself is not enough for a good keypoint detector, it has to make sure the corresponding descriptors are discriminative enough. Exploring the idea of learning free methods further, our approach in Chapter 5 provides a new definition for keypoints that makes sure the corresponding descriptors are discriminative enough for accurate correspondence estimation.

2.2.3 Self-Supervised Depth and Camera Motion Estimation

One fundamental limitation of supervised methods is their need for ground truth data, which hinders their wider applicability. Ideally, having a method that can be tested and, if necessary, trained on unlabeled data is much more attractive. There are two primary loss functions used to train these self-supervised methods: one in the spatial domain (left and right images) and another in the temporal domain (using a sequence of temporally aligned images). The spatial photometric loss is used when stereo images are available for training, while the temporal photometric loss is used with temporally aligned images, typically when only monocular image sequences are available for training.

A typical illustration of a monocular self-supervised system, taken from SfMlearner [37], is shown in Figure 2.5. One important aspect of monocular frameworks is that they must include either a pose network or an optical flow network to establish correspondences between the pixels of the image sequence. Therefore, every monocular system can be considered a joint depth and motion estimation system. This section describes the methods that concentrated on depth estimation, followed by a discussion about camera motion estimation. Figure 2.6 illustrates several notable methods and the progress made in structural recovery and depth estimation from one or more images over the past eight years, focusing on approaches trained with complete self-supervision. The following section delineates these prominent methods, covering stereo, monocular, and monocular-stereo variants, including those employing both spatio-temporal losses for depth estimation.

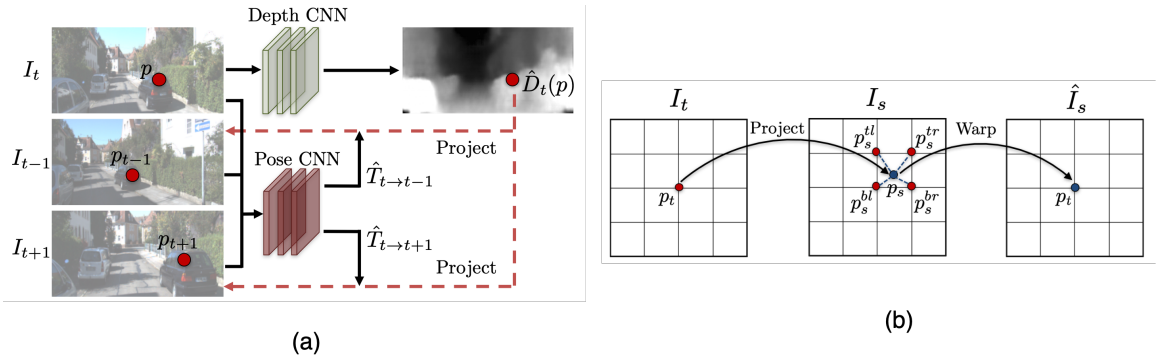


Figure 2.5: Illustration of a typical monocular self-supervised depth and camera motion estimation system in (a) and the inverse image-warping operation used for image reconstruction in (b). Given a triplet of images I_t, I_{t+1}, I_{t-1}, the goal is to train a depth network (Depth CNN) to estimate a per-pixel depth map $\hat{D}_t$ and a pose network (Pose CNN) to estimate the camera motion T between I_t and I_s, where $s \in \{t+1, t-1\}$. This information is then utilized in inverse image-warping, where a pixel p_t in I_t is projected onto I_s using $\hat{D}_t$ and $T_{t\to s}$. The RGB values for p_t are sampled from neighbouring pixels in I_s using bilinear interpolation, as shown in (b), to reconstruct the image $\hat{I}_s$. The key idea is that if the depth and camera motion estimations are accurate, $\hat{I}_s$ should closely match I_t. A photometric loss is calculated by comparing I_t and $\hat{I}_s$, which is then used to train both the depth and pose networks. This illustration is adapted from SfMLearner [37].

In the quest to design a method that could purely learn from images, Garg et al. [178] proposed a geometry-based loss function to train a network in a completely unsupervised fashion using a pair of stereo images. Monodepth [180] improved upon this by using differentiable image warping [181] and a structural similarity-based [182] image comparison loss. SfMLearner [37] used only monocular images to jointly learn depth and ego-motion. This approach was further improved by combining stereo and monocular losses in UnDEMoN (Unsupervised Depth and Ego-Motion Network) [183] and UnDeepVO [147]. Later, GeoNet [184] and EPC (Every Pixel Counts) [185] and EPC++ [186] introduced per-pixel optical flow maps along with depth and ego-motion to mitigate the effects of moving objects. Some methods used GAN-based learning to train their systems, GAN-VO [187], MonoGAN [188], Adv-UnDEMoN [189]. Further, Monodepth2 [190] extended Monodepth into the temporal domain, proposing architectural changes and robust loss functions to achieve state-of-the-art results. Further, SGDepth (Semantically Guided Depth estimation) [191] trained semantic segmentation as an auxiliary task while masking out the dynamic objects that violate the training loss, leading to better depth estimation results. In PackNet-SfM [192], a new encoder-decoder architecture that uses packing and unpacking layers based on sub-pixel convolutions [193] and 3D convolutional layers aimed at better ac-

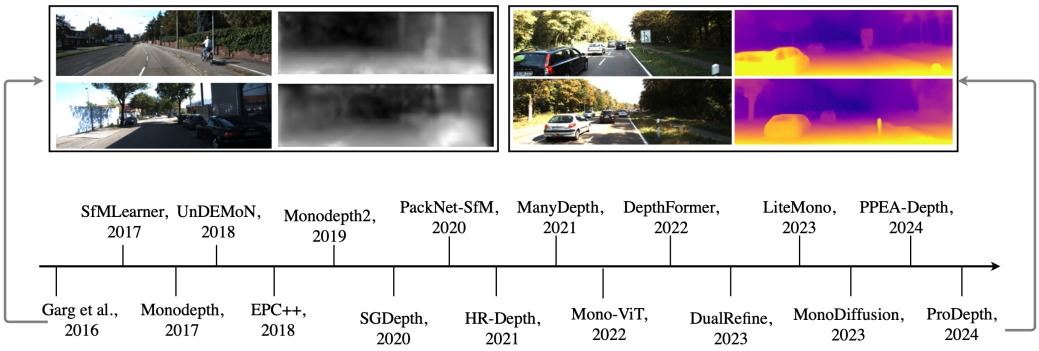


Figure 2.6: Comparison of joint depth and ego-motion estimation progress from (2016 to 2024). The left side displays depth maps estimated by Garg et al., [178], while the right side showcases more recent results from ProDepth[179]. Below, a few notable methods of depth estimation from 2014 to 2024 are highlighted. These methods include self-supervised monocular, stereo and some multi-view methods. The qualitative results are taken from their respective papers, and the goal here is to show how much progress has been made in recovering, structural details accurately. In both images, darker color represents higher depth.

curacy and generalization. They argued that this architecture helped them to preserve the 3D details during the compression (encoding) and decompression (decoding) stages to help them achieve the aforementioned. HR-Depth [194] used an effective skip connection and convolutional block to integrate spatial and semantic information. Meanwhile, ManyDepth [195], extended this self-supervised depth estimation problem to a sequence of images (3, or 5) arguing that often the times we have access to the sequence of images but the models at that time were not making use of that signal. The success of the vision transformers(ViTs) was brought into this problem by MonoViT [21] with the same argument that the standard convolutional neural networks lack the global context, leading to better depth estimation. Similar to the argument made by TransDepth [100] about the transformers lacking global context, so requires the Convolutional Networks for local details, DepthFormer [196], proposes a method with a transformer encoder-decoder and a conv-layer to capture local details for self-supervised depth estimation further improving the accuracy over MonoViT. CADepth-Net (Channel-Wise Attention based Depth Network) [197] extended the attention mechanism from transformers to channels of the feature maps and proposed structure perception and detail emphasis modules for capturing the context of scenes with detail for better depth estimation. Later on, SD-SSMDE (Self Distillation based Self-Supervised Monocular Depth Estimation) [198] adopted the pseudo label concept, where they trained using the high-resolution images at first and then used the same network to generate pseudo labels for later training using a scale-invariant loss. RM-Depth [199] proposed recurrent modulation units

for effective fusion of deep features with fewer parameters and a warping-based motion field for moving objects to improve scene rigidity, leading to enhanced depth estimation. While the majority of the methods concentrate on depth estimation, DualRefine [200] proposed a multi-frame approach and argued that the epipolar constraint on the cost-volumes will be hugely impacted if camera poses are badly estimated, and at the same time better depth estimation helps to refine the bad-poses. Alleviating the problems of heavier transformer architectures, LiteMono [201] proposed an architecture with dilated convolutions [202] for rich multi-scale features and a feature interaction module to capture global context similar to transformers. This method achieved results comparable to those of transformer-based architectures while significantly speeding up the inference. Meanwhile, there is a huge success happened in generative modelling using diffusion models [203]. This success was also brought into the task of self-supervised depth estimation by MonoDiffusion [204]. However, these models suffer from the speed of inference compared to single inference models. Another assumption of photometric loss is that the scene should be static. However, that is not true in many real-world scenarios. To address them instead of masking the dynamic objects, PPEA-Depth (Progressive Parameter-Efficient Adaptation Depth) [205] proposed to train first on static scenes and extend to dynamic scenes using adapters inspired by the parametric efficient training in the natural language processing domain [206]. More recently, ProDepth [179] framework improves depth estimation by addressing issues caused by moving objects through a probabilistic approach. It first estimates uncertainty related to the assumption that everything in the scene is static, identifying potential dynamic areas. Then, it corrects these areas by adjusting the depth predictions based on both single-frame and multi-frame information while also reweighting the training loss to reduce the impact of incorrect supervision in uncertain regions.

However, all these methods have predominantly been tested on standard daytime datasets, with little attention given to poorly illuminated scenarios, such as nighttime conditions. Figure 2.7 illustrates how a monocular method Monodepth2 [207], a method accurate during daytime, fails miserably when applied to nighttime data. Similarly, Figure 1.3 illustrates the failure case where the state of the art stereo methods such as Unimatch-Stereo [128], IGEV-Stereo [39] tested on nighttime data, they also found to be working sub-optimally in the poorly illuminated regions of the images. Very interestingly, these issues persist even when the model is trained specifically on poorly illuminated data with the existing self supervised losses, as demonstrated by ADFA [26] and DeFeat-Net [27], which are early methods proposed for self-supervised depth estimation under poor illumination conditions. This motivated us to pursue the our Research Questions 1 and 2 about how to relax the assumptions of the photometric loss to make it work across various illumination conditions



Figure 2.7: Examples illustrating the performance of the Monodepth2 [207] method. The left side shows results from the original Monodepth2 paper, where the model was trained and tested on well-illuminated images, generating highly plausible depth maps. The right side displays the performance of the model when trained and tested on nighttime images, where it struggled to maintain the majority of the structural details. These nighttime results are from ADFA [26] and DeFeat-Net [27].

in both monocular in Chapter 3 and stereo cases in Chapter 4 to design a system that is accurate, robust and generalisable.

In terms of camera motion, SfMLearner [37], the first self-supervised monocular method, and by design, the method needed a pose network and addressed ego-motion as a sub-problem. This could be considered the first self-supervised deep learning-based end-to-end pose estimation method. In this area, network design choices and the representation of the rotation component of ego-motion are key differentiators.

SfMLearner [37] estimated translation and rotations using Euler angles using a single decoder head. UndeepVO [147] and UnDEMon [183] introduced two heads, one for translations and another for rotations and continued using Euler representation. Monodepth2 [207] shifted to using Axis-angle representation for rotations, a choice that has been widely adopted in subsequent methods. Like supervised methods, some self-supervised approaches also incorporate auxiliary information, such as intrinsics [208]. Extending Monodepth2 formulation as front-end and extending it with back-end non-linear optimisation, [209] proposed to improve the results further. Similarly, SfMLearner was extended by Li et al. [210] to continuously adapt to new environments as a meta-learning algorithm. MotionHint [211] extended Monodepth2 with a motion constraint estimated using a pretrained motion model. Orthogonal to these approaches, CTCNet [212] proposed a self-supervised network that does not use any depth estimation but uses camera motion estimated using traditional methods.

In recent years, many methods have been proposed for self-supervised depth and ego-motion estimation. Interestingly, most of these methods focus heavily on depth estimation, with less emphasis on evaluating ego-motion. Additionally, these methods are often trained and tested on autonomous driving datasets, where vehicles primarily travel in straight lines. This data imbalance hinders the generalization of these ego-motion estimation networks, even within the same dataset, let alone in novel scenes. Another challenge arises from the joint training of depth and ego-motion models, which limits the amount of motion parallax that the ego-motion network can observe, which is usually kept low to maintain image overlap in sequences. This limitation has not been adequately tested to determine how well self-supervised ego-motion networks perform with varying camera speeds.

Moreover, some studies [213] discourage the use of convolutional networks for estimating camera poses, as they struggle to effectively model novel variations between images. These concerns about generalization, robustness, and design ambiguities have led the research community to explore hybrid approaches that combine the strengths of both geometric and deep learning methods within self-supervised pipelines. However, only a few methods, such as those proposed in [214, 215], have adopted these hybrid pose estimation techniques and demonstrated improvements in their estimates. This presents an interesting avenue for exploring self-supervision in correspondence estimation.

2.2.4 Challenges with End to End Learning

Depth and pose estimation methods, whether based on supervised or self-supervised learning, and whether using monocular or stereo approaches, are predominantly trained in an end-to-end manner. This task-specific data training limits the diversity of the data that can be used to train the model. In supervised learning settings, this often leads to overfitting, resulting in poor performance or outright failure when the model is tested on data outside the training distribution, such as data from different geographic locations, infrastructures, illumination conditions, or weather conditions.

To mitigate these problems, supervised models often employ large datasets with data augmentation techniques, such as altering color, hue, and saturation, or cropping images. However, this approach is both expensive and impractical, as the real world is far more diverse in terms of geometry and photometry than can be captured through these methods. The emergence of self-supervision has introduced greater flexibility, improving both robustness and generalization, albeit at the cost of accuracy compared to supervised methods. Many promising end-to-end self-supervised methods are being proposed to bridge the the gap as discussed in previous sections.

Beyond generalization and robustness issues, end-to-end learning is typically very rigid. Once a model is trained, the encoders and decoders become interdependent, making it impossible to interchange them between models trained for the same tasks. This rigidity is partly due to the high-dimensional latent space in which these deep learning methods operate and is often addressed by fine-tuning for new settings. Additionally, end-to-end models are task-dependent; features extracted for depth estimation, for example, cannot be used for other tasks such as pose estimation. The lack of a definition for a generic feature that could be used universally across multiple tasks poses another challenge.

In the quest of making a task-agnostic/general feature encoders, significant progress has been made in learning task-agnostic features through self-supervision, allowing models to be trained on hundreds of millions of images. These models, known as Visual Foundation Models, have demonstrated not only great generalization and robustness but also the potential to perform tasks such as depth estimation and segmentation without retraining the encoder.

In this thesis, we explore the use of these features two tasks: for stereo-depth estimation to increase the robustness and genralisation capabilities of stereo-feature matching Chapter 4, and addressing the Research Question 3 about how to harness the power of these foundational models for estimating keypoints in Chapter 5. More details about these foundational models are provided in the following section.

2.2.4.1 Visual Foundational Models

In 2017, Vaswani et al. introduced transformers, a revolutionary neural network architecture that enabled long-term context correlation and significantly improved the accuracy of sequence-to-sequence tasks, such as natural language translation [139]. Since then, transformers have become the architecture of choice for any task requiring correlation or context.

Three years later, in 2020, Dosovitskiy et al. adapted transformers for the vision domain by dividing images into 16×16 patches and treating them as sequences. This approach led to the development of the Vision Transformer (ViT) [32], designed to address supervised image classification problems. However, despite their massive computational and memory demands, ViTs only showed marginal performance gains compared to the best Convolutional Neural Networks (CNNs) at the time.

Around the same period, Grill et al. introduced a method called BYOL (Bootstrap Your Own Latent) [216] for self-supervised feature learning. BYOL employed a pair of networks—an online network and a target network—to learn visual features without relying on contrastive learning, which was the prevailing approach for self-supervised learning at that time.

Intrigued by the marginal performance of ViTs on supervised tasks like classification, Caron et al. proposed a new architecture called Distillation with No Labels (DINO) [35]. DINO was a self-supervised approach that leveraged the emergent properties of ViTs, drawing inspiration from BYOL. This approach proved to be highly successful, significantly improving how self-supervised features could classify images and enabling the model to automatically capture the semantics of visual features. The features learned by DINO were further explored in subsequent work [217] for tasks like part-segmentation and semantic and geometric keypoint matching by analyzing features from various layers of the DINO ViT. This paper discovered that the early layers of DINO capture local features while the deeper layers progressively capture more semantic information. This insightful finding motivated us to explore the depth estimation problem in Chapter 4. In this chapter, stereo matching is performed semantically using the deeper layers, and any residual errors are further corrected by leveraging local features from the middle layers of DINO. Similar to the approach in [217], we conducted a more in-depth exploration of the local features to understand how suitable these features are for keypoint detection and matching in Chapter 5.

Recently, Facebook scraped the internet for 1 billion images and curated 142 million of them to train the second version of DINO, known as DINO-V2. This model used a fixed patch size of 14×14 and was evaluated on a range of tasks, including semantic segmentation, monocular depth estimation, instance segmentation, and action recognition, all from a frozen encoder. The success of DINO-V2 has sparked further research in areas such as Visual Place Recognition (VPR) [218], feature matching [173], and our own stereo-matching method explained in Chapter 4.

In parallel, other foundational models have emerged in supervised learning for tasks like semantic segmentation and image captioning. Notable examples include Segment Anything (SAM) [219] and CLIP [34]. Most recently, NVIDIA released AM-RADIO [36], a model that distils knowledge from DINO-V2, SAM, and CLIP into a single ViT model.

Despite these advancements, the computational, data, and memory demands of these models remain out of reach for many academic institutions, making it challenging to develop new foundational models. For example, Facebook research team took 22,000 hours for ViT-g, 4,500 hours for ViT-S distillation, 5,300 hours for ViT-B distillation, 8,000 hours for ViT-L distillation [2] on Nvidia-A100 GPU to train these models. Besides the impressive performance of these pre-trained features, the lack of resources motivates us to explore their direct use for our tasks.

[2] https://github.com/facebookresearch/dinov2/blob/main/MODEL_CARD.md

2.3 Datasets and Metrics

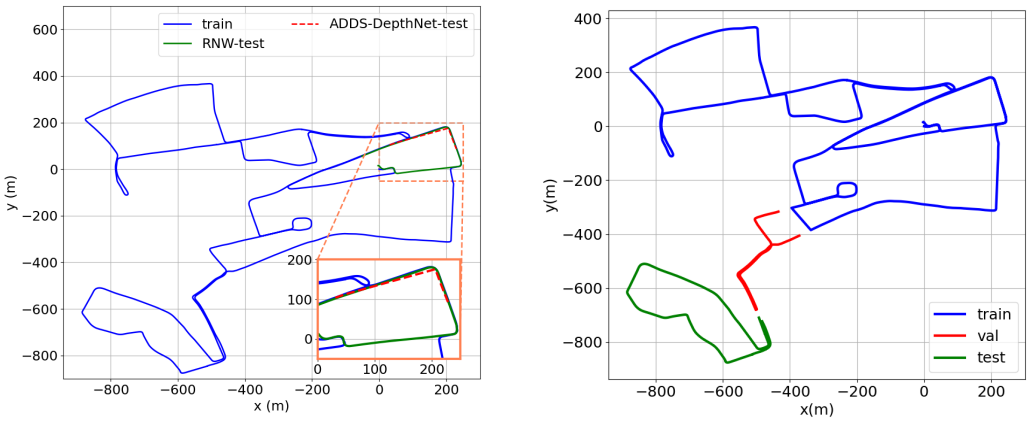
In this section, the datasets that are used in this thesis to perform all the experiments and ablation studies are explained in detail followed by various metrics used to evaluate both depth and ego-motion with an explanation on how to interpret them.

2.3.1 Datasets

Oxford RobotCar Dataset: This dataset was collected over a one-year period by traversing the same route multiple times so as to include a variety of different weather and lighting conditions. We used the six sequences in the `2014-12-09-13-21-02` traversal for our daytime experiments, and the six sequences in the `2014-12-16-18-44-24` traversal for our night-time ones. We cropped the bottom 20% of each image to remove the car hood, and adjusted the camera intrinsics accordingly. We then filtered each sequence to produce a sub-sequence of keyframes such that each consecutive pair of keyframes was at least 0.5m apart. We then constructed our training, validation, and testing splits from triplets of images in the original sequence, each centred on one keyframe (at this stage with a stride of 1). This process had the beneficial side-effect of filtering out most of the images that were captured when the ego-vehicle was stationary.

There are, unfortunately, no standard splits for the RobotCar dataset, and moreover, the splits chosen by other methods [38, 220] overlapped geographically (see Figure 2.8(a)). For these reasons, we constructed our own splits by geographically dividing our triplets (constructed as above) into three daytime sets and three nighttime sets (see Figure 2.8(b)). The nighttime splits are comprised of $19,612$ triplets for training, $2,629$ triplets for validation and $4,559$ triplets for testing. The daytime splits are comprised of $17,790$ triplets for training, $6,693$ triplets for testing and $2,629$ triplets for validation. We used the GPS locations provided with the dataset to ensure that there was no overlap between the training and testing locations in the city. Our hope in constructing these new splits is that they will help enable fairer comparisons between methods on this dataset going forward.

Multi-Spectral Stereo (MS2) Dataset: The MS2 dataset [222] contains 184K data pairs taken from multi-spectral sensors on a vehicle in Daejeon, South Korea. Each sequence contains data from stereo RGB, Near-Infrared (NIR), thermal cameras, LiDARs, and a single GPS with IMU. The dataset is recorded across city, residential, road, campus, and suburban areas. The sequences include various lighting, weather, and traffic conditions, including day and night with clear, cloudy, and rainy weather conditions. Following the evaluation split proposed in [222], we use the *Road*, *City*, and *Campus* nighttime sequences, further sub-sampling them with 5m distance between consecutive test images. This gave 1,470



(a) Splits of RNW [38], ADDS-DepthNet [220]

(b) Our splits

Figure 2.8: A geographic visualisation of the training and testing splits used by (a) RNW [38] and ADDS-DepthNet [220], and (b) our method, based on the GPS locations provided with the Oxford RobotCar dataset [221]. The splits of the methods in (a) clearly suffer from geographical overlap, whereas ours are explicitly constructed to be geographically disjoint, enabling fairer comparisons between methods on this dataset going forward. (Best viewed in colour.)

pairs for evaluation. We use the (filtered) ground truth depth data released with the dataset for the evaluation.

HPatches: This dataset [223] is a widely-used benchmark for evaluating local image feature descriptors and matching algorithms. It contains over 116 sequences of image patches extracted from real-world images, with two types of variations: viewpoint and illumination changes. The *viewpoint* subset with 59 sequences consists of patches taken from different perspectives, enabling the evaluation of feature detectors and descriptors under geometric transformations. The *illumination* subset with 57 sequences includes patches captured under varying lighting conditions, allowing for robustness testing of descriptors across lighting changes. Each sequence in the dataset is accompanied by ground-truth homographies, which facilitate the accurate evaluation of image alignment algorithms based on geometric transformations. The HPatches dataset is commonly used in research areas like feature detection, matching, and image alignment.

ScanNet: This is a popular indoor dataset [224] with 1613 monocular image sequences and ground truth depths and poses. Following the wide-baseline matching protocol of SuperGlue [150], we use their test split of 1500 images. The images here include extreme viewpoint and scale changes.

2.3.2 Metrics

Depth estimation metrics are used to evaluate the accuracy of a model's predicted depth without ground truth supervision. Common metrics include the Absolute Relative Difference, Root Mean Squared Error, Logarithmic Root Mean Squared Error, and Threshold Accuracy. These are defined as follows:

Absolute Relative Difference (Abs Rel): The Absolute Relative Difference measures the average relative error between the predicted and true depth values:

$$\text{Abs Rel} = \frac{1}{N} \sum_{i=1}^{N} \frac{|d_i - \hat{d}_i|}{d_i}$$

where d_i is the ground truth depth, $\hat{d}_i$ is the predicted depth, and N is the total number of pixels. A lower value indicates better performance.

Root Mean Squared Error (RMSE): The Root Mean Squared Error is computed as:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (d_i - \hat{d}_i)^2}$$

RMSE penalizes large errors more significantly, making it particularly important for detecting large depth prediction errors. Lower RMSE values indicate better model performance.

Logarithmic Root Mean Squared Error (Log RMSE): Log RMSE is computed in logarithmic space to account for a wide range of depth values:

$$\text{Log RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\log d_i - \log \hat{d}_i)^2}$$

This metric is scale-invariant and is useful when the predicted depth values vary significantly across the dataset.

Threshold Accuracy: The Threshold Accuracy measures the percentage of predicted pixels that satisfy:

$$\delta = \max \left(\frac{d_i}{\hat{d}_i}, \frac{\hat{d}_i}{d_i} \right)$$

The model is considered accurate if δ is less than a certain threshold. Common thresholds are $\delta < 1.25$, $\delta < 1.25^2$, and $\delta < 1.25^3$. Higher threshold accuracy indicates more accurate predictions over different scales.

Interpretation of Metrics: Lower values for error metrics such as RMSE, Log RMSE, and Abs Rel indicate better depth estimation performance. Higher Threshold Accuracy (i.e., more pixels satisfying the threshold) reflects better performance at different levels

of tolerance: For $\delta < 1.25$, the predicted depth is at most 25% off from the true depth, indicating very high accuracy. A high percentage of pixels within this threshold means the model is highly precise.

Absolute Trajectory Error (ATE): The Absolute Trajectory Error (ATE) measures the difference between the predicted trajectory and the ground truth trajectory after aligning them in the same coordinate system. For two trajectories $\mathbf{T}_{gt} = [\mathbf{t}_{gt}^1, \mathbf{t}_{gt}^2, \ldots, \mathbf{t}_{gt}^N]$ and $\mathbf{T}_{pred} = [\mathbf{t}_{pred}^1, \mathbf{t}_{pred}^2, \ldots, \mathbf{t}_{pred}^N]$, where $\mathbf{t}_{gt}^i$ and $\mathbf{t}_{pred}^i$ are the ground truth and predicted 3D positions at timestamp i, respectively, the ATE is computed as follows:

Alignment: First, align the predicted trajectory $\mathbf{T}_{pred}$ to the ground truth trajectory $\mathbf{T}_{gt}$ by translating the predicted trajectory so that the first points of both trajectories match. The alignment is performed by subtracting the difference between the first points:

$$\mathbf{t}_{pred}^i \leftarrow \mathbf{t}_{pred}^i + (\mathbf{t}_{gt}^1 - \mathbf{t}_{pred}^1), \quad \forall i$$

Scaling: After translation, a scaling factor α is optimized to minimize the alignment error. The scaling factor is computed as:

$$\alpha = \frac{\sum_{i=1}^{N} \mathbf{t}_{gt}^i \cdot \mathbf{t}_{pred}^i}{\sum_{i=1}^{N} \|\mathbf{t}_{pred}^i\|^2}$$

where $\mathbf{t}_{gt}^i \cdot \mathbf{t}_{pred}^i$ denotes the dot product between the ground truth and predicted positions.

RMSE Calculation: The alignment error between the ground truth and scaled predicted positions is computed as:

$$\mathbf{e}^i = \alpha \mathbf{t}_{pred}^i - \mathbf{t}_{gt}^i$$

The Root Mean Square Error (RMSE) is then calculated as:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \|\mathbf{e}^i\|^2}$$

where N is the total number of matched points, and $\|\mathbf{e}^i\|$ is the Euclidean distance between the ground truth and predicted points after scaling. The RMSE represents the final ATE, which quantifies the absolute error between the aligned predicted trajectory and the ground truth trajectory.

Pose Error AUC (Area Under the Curve): The pose error AUC is computed to evaluate the accuracy of pose estimation over a range of angular error thresholds. For a given set of pose errors $\mathbf{e} = [e_1, e_2, ..., e_N]$, where e_i represents the maximum of the rotation and translation errors for sample i, the AUC is computed by integrating the recall-error curve up to a specified threshold τ.

Let $\mathbf{e}$ be sorted in ascending order and augmented with 0 to represent the perfect pose estimate. The recall r_i is defined as:

$$r_i = \frac{i}{N}, \quad i = 0, 1, ..., N$$

where N is the total number of errors. For each threshold τ, the AUC is computed as:

$$\text{AUC}(\tau) = \frac{1}{\tau} \int_0^{\tau} r(e)\, de$$

where $r(e)$ is the recall at error e. The resulting AUC is a scalar value between 0 and 1, representing the fraction of correctly estimated poses within the threshold τ. Higher AUC values indicate better performance in pose estimation.

In our experiments, the AUC is computed for angular thresholds $\tau \in \{2^\circ, 5^\circ, 10^\circ, 20^\circ\}$, denoted as AUC@5, AUC@10, and AUC@20, respectively.

Mean Matching Accuracy (MMA): MMA is a standard metric used to evaluate the performance of feature-matching algorithms. It measures the accuracy with which keypoints detected in one image can be correctly matched to their corresponding keypoints in another image. The accuracy of a match is determined by the Euclidean distance between the matched keypoints and their ground-truth locations, where a match is considered correct if the distance falls below a predefined threshold.

MMA is computed by evaluating the percentage of correct matches within a set of error thresholds, such as 1, 3, 5, or 10 pixels. For each threshold, the accuracy is calculated as the proportion of matches where the error is less than or equal to the threshold. The final MMA score is obtained by averaging the matching accuracy across all thresholds, providing a single metric that reflects the overall performance of the matching algorithm.

Formally, let p_1 and p_2 represent the locations of corresponding keypoints in two images, and let H be the ground-truth homography matrix that maps p_1 to p_2. The matching error is defined as:

$$e(p_1, p_2) = \|Hp_1 - p_2\|$$

A match is considered accurate if $e(p_1, p_2) \leq \tau$, where τ is the error threshold. MMA is the mean fraction of correct matches over several thresholds τ.

MMA is widely used in evaluating feature detectors and descriptors, as it provides a robust measure of how well keypoints are matched under different transformations, such as changes in viewpoint or illumination. It is commonly applied in benchmarks like the HPatches dataset to compare the performance of different matching algorithms.

2.4 Summary

This chapter provides a comprehensive overview of depth and camera motion estimation methods, highlighting the evolution from traditional techniques to modern deep learning approaches. Traditional depth estimation methods, such as stereo vision and structured light, have played a crucial role in advancing computer vision. However, they face significant challenges, including sensitivity to environmental conditions and limited accuracy in complex scenes, emphasizing the need for more robust techniques. In ego-motion estimation, traditional keypoint detection methods are often impacted by noise and lack distinctive features in certain scenes. They also suffer from drift and cumulative errors, making them less reliable in dynamic environments. Moreover, the joint estimation of depth and camera motion (a.k.a ego motion) traditionally involves complex algorithms that struggle with integration and real-time processing, reinforcing the need for enhanced solutions. The transition to deep learning has revolutionized these fields, with neural networks improving depth estimation accuracy and robustness, especially in diverse and challenging environments. Similarly, deep learning techniques in ego-motion estimation have enhanced feature extraction and motion estimation, overcoming many of the limitations seen in traditional methods. Joint deep learning approaches for depth and ego-motion estimation provide more accurate and integrated solutions, pushing the field forward. However, challenges remain, particularly in photometric methods used in self-supervised learning for depth and camera motion, which have not been fully explored in nighttime or low-light conditions. This presents a significant opportunity for future research. As computational resources improve, foundational models that focus on addressing out-of-distribution encoding problems show promise for simplifying and enhancing learning processes. Future advancements may lean towards decoder-only or minimal learning solutions, offering more efficient and scalable methods. In summary, this chapter explains the progress from traditional methods to deep learning innovations, highlights ongoing challenges and points towards future developments that could further advance the field.

Chapter 3

Relaxing Photometric Loss for All Day Depth and Ego-motion Estimation

Self-supervised deep learning methods for joint depth and ego-motion estimation can yield accurate trajectories without needing ground-truth training data. However, as they typically use photometric losses, their performance can degrade significantly when the assumptions these losses make for (e.g. temporal illumination consistency, a static scene, and the absence of noise and occlusions) well-illuminated conditions, such as the daytime, are violated. This limits their use for poorly illuminated conditions, e.g. nighttime sequences, which tend to contain many point light sources (including on dynamic objects) and low signal-to-noise ratio (SNR) in darker image regions. In this chapter, we show how to use a combination of three techniques to allow the existing photometric losses to work for both day and nighttime images. First, we introduce a per-pixel neural intensity transformation to compensate for the light changes that occur between successive frames. Second, we predict a per-pixel residual flow map that we use to correct the reprojection correspondences induced by the estimated ego-motion and depth from the networks. Third, we denoise the training images to improve the robustness and accuracy of our approach. These changes allow us to train a single model for both day and nighttime images without needing separate encoders or extra feature networks like existing methods. We perform extensive experiments and ablation studies on the challenging Oxford RobotCar dataset to demonstrate the efficacy of our approach for both day and nighttime sequences.

3.1 Introduction

An ability to capture 3D scene structure is crucial for many applications, including autonomous driving [1], robotic manipulation [2], and augmented reality [4]. Many methods

use LiDAR or fixed-baseline stereo to acquire the depth needed to reconstruct a scene, but researchers have also long been interested in estimating depth from monocular images, driven by the ubiquity, low cost, low power consumption and ease of deployment of monocular cameras. By contrast, LiDAR can be power-hungry, and stereo rigs must be calibrated and time-synchronised to achieve good performance.

Multi-view monocular depth estimation approaches have long used variable-baseline stereo over multiple images to recover depth [78, 225]. Meanwhile, progress in deep learning has opened up the additional possibility of estimating depth from a single monocular image. Deep learning methods for depth estimation can be broadly divided into two types, namely supervised methods [19, 20], and self/unsupervised methods [178, 226, 187]. Typically, supervised approaches have achieved very good results for the dataset(s) on which they are trained, but their need for ground-truth information during training has often hindered their deployment in new domains.

By contrast, self/unsupervised methods have typically adopted the use of a geometry-based loss function inspired by the strong physical principles of traditional methods [22, 23]. This loss function is commonly referred to as the photometric or appearance loss, and is based on the assumptions that (i) the scene is static (i.e. contains no moving objects), (ii) the illumination in the scene is diffusive (i.e. there are no specular reflections) and temporally consistent (i.e. the pixels to which any scene point projects in any two consecutive frames have the same intensity), and (iii) the images are free of noise and occlusions [22, 23, 24, 25]. In practice, many of these assumptions are at least partly false, which can lead to errors in the estimated depth: scenes are quite likely to contain dynamic objects (e.g. cars, cyclists and pedestrians, in an outdoor driving scenario), surface materials are rarely fully diffusive, and occlusions are common. During the day, it is somewhat reasonable to assume that the illumination is moderately temporally consistent for image sequences captured outdoors, as the sun is by far the dominant light source in that case, and the light it casts changes only slowly over time; however, at night in a scenario like outdoor autonomous driving, the numerous point light sources that are typically turned on after dark (e.g. car headlights, lamp posts, etc.) can cause the illumination to change drastically from one frame to the next. At night, also, the motion blur associated with the movement of dynamic objects in the scene (including the ego-vehicle) becomes worse, owing to the longer exposure times typically used when capturing night-time images [30, 31], and the signal-to-noise ratio of the (darker) images becomes much lower than it would be during daytime. Such issues, as illustrated in Figure 3.1, inhibit the straightforward use of deep networks based on photometric loss for night-time sequences.

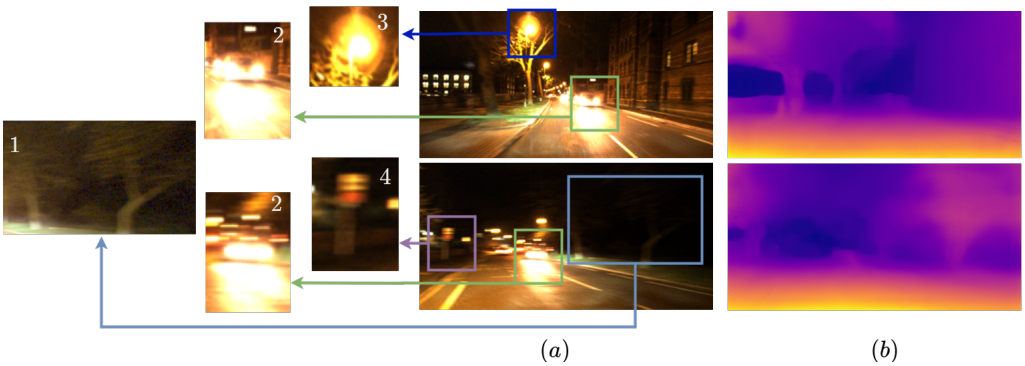


Figure 3.1: (a) The challenges posed by night-time images: (1) low visibility and noise (patch enhanced for better readability); (2) moving light sources with saturating image regions; (3) point light sources; (4) extreme motion blur. (b) Despite these adverse conditions, which violate the assumptions made by the photometric loss, our method can successfully estimate accurate depth maps.

In this chapter, we address this problem by directly targeting violations of the temporal illumination consistency, static scene and noise-free assumptions on which the photometric loss relies. As shown by our day and night results in Table 3.2, these three together account for much of the discrepancy in performance between daytime and night-time. A lack of temporal illumination consistency caused by point light sources in the scene can cause pixels to be incorrectly matched between consecutive frames. To rectify this, we propose a novel per-pixel neural intensity transformation that learns to compensate for these light sources (see Section 3.3.2). Whilst conceptually straightforward, this approach is surprisingly effective, as our results in Section 3.4 demonstrate. Interestingly, they also show that it is able to operate well over wide (motion parallax) baselines, allowing us to leverage the better depth estimation performance that wider baselines offer. To correct for dynamic objects in the scene, as well as motion blur, we predict a per-pixel residual flow map (see Section 3.3.3) that we use to correct the reprojection correspondences induced by the estimated ego-motion and depth from the networks. This improves depth estimation performance at any time of day (see Section 3.4), but has additional theoretical benefits for night-time sequences because of the greater motion blur from which they typically suffer. Lastly, we robustify our approach against noise by incorporating Neighbour2Neighbour [227], a state-of-the-art denoising module, in our photometric loss formulation (see Section 3.3.4).
To summarize, the proposed contributions in this chapter are,

1. We propose to account for the light changes between temporally aligned images in a scene with multiple light sources, like nighttime, using a Neural Intensity Transform.

This module estimates the necessary scale and shift values required in pixel intensities to ensure the photometric loss of a pixel is accurate.

2. To account for the dynamic pixels caused by the moving objects, we propose a module that estimates the residuals needed after reprojecting the pixels between frames using ego-motion. Further, we propose to regularize these residuals to avoid trivial cases and complement the training.

3. Lastly, simple yet effective denoising was applied to the original and reconstructed images to account for the ambiguities in the photometric loss calculation.

4. Extensive experiments and ablation studies are performed on the Oxford RobotCar dataset for depth and ego-motion estimation, exploring the effectiveness of the individual modules in varying illumination conditions, such as day and nighttime.

The rest of the chapter is organized as follows: related work is discussed in Section 3.2 followed by the proposed method and various experiments and ablation studies in Section 3.3 and Section 3.4 respectively. Lastly, the chapter concludes with some promising directions for future work in Section 3.5 and conclusions in Section 3.7.

3.2 Related Work

Estimating depth from images has a long history in computer vision. Several methods use either stereo images [228, 229, 230], or two or more images taken from different viewing angles [231, 232, 233]. We try to solve this problem using a single monocular image without any constraints on the scene of interest. Various methods have addressed this problem using supervised learning [19, 20, 234, 235, 94]. However, it is infeasible to have ground-truth depth maps for training on every scene, which limits the application of these methods and helps motivate unsupervised solutions to this problem.

Unsupervised Methods: Garg et al. [178] proposed a geometry-based loss function to train a network in a completely unsupervised fashion using a pair of stereo images. Monodepth [180] improved this by using differentiable image warping [181] and structural similarity-based [182] image comparison loss. SfMLearner [37] used only monocular images to learn depth and ego-motion jointly. It was further improved by combining stereo and monocular losses in [183, 147]. Later, GeoNet [184] and EPC [185] learnt per-pixel optical flow maps along with depth and ego-motion to mitigate the effect of moving objects. Some methods use GAN-based learning to train their systems [187, 188, 189]. Recently, Monodepth2 [190] extended Monodepth to the temporal domain, proposing a

few architectural changes and robust loss functions to achieve state-of-the-art results. HR-Depth [194] used an effective skip connection and a convolution block to integrate spatial and semantic information. SD-SSMDE [198] introduced a two-stage training strategy to improve scale and inter-frame scale consistency in depth by utilising depth estimation from the first stage as a pseudo-label. Based on channel-wise attention, CADepth-Net [197] proposed structure perception and detail emphasis modules for capturing the context of scenes with the detail for depth estimation. RM-Depth [199] proposed recurrent modulation units for an effective fusion of deep features with fewer parameters and a warping-based motion field for moving objects to improve the scene rigidity, leading to enhanced depth estimation. More broadly, recent years have also seen a wide range of other advances in depth estimation, e.g. changes to the network architecture [192], the addition of extra loss functions [236], and better handling of dynamic objects [237]. However, all of these methods have been tested on standard daytime datasets, whereas our method is designed to work at night as well.

Nighttime Methods: All the methods above are trained using photometric loss as the main supervision signal, and with an assumption of temporal illumination consistency, which is not valid at night. A few methods, such as DeFeat-Net [27], ADFA [26] and [25], have explored how to estimate depth information from nighttime RGB images. DeFeat-Net [27] learns n-dimensional deep feature representations (assumed to be illumination-invariant) using a pixel-wise contrastive loss. The feature maps are simultaneously used along with the input images for photometric loss calculation during training. ADFA [189] mimics a daytime depth estimation model by learning a new encoder that can generate 'day-like' features from nighttime images using a domain adaptation approach. Instead of feature translation as in [26], the authors in [25] propose a joint network for image translation and stereo image-based depth estimation. Recently, photometric losses are again used with an image enhancement module and a GAN-based depth regulariser in [38]. Liu et al. [220] divided the day and nighttime images into view-invariant and variant feature maps using separate encoders and used the view-invariant information for depth estimation. All these methods either need two separate encoders for day and nighttime images [26, 220, 38] or need to learn an illumination-invariant feature space [27]. By contrast, our proposed method learns in a completely self-supervised fashion, without needing stereo images, ground-truth depth information or any additional feature learning.

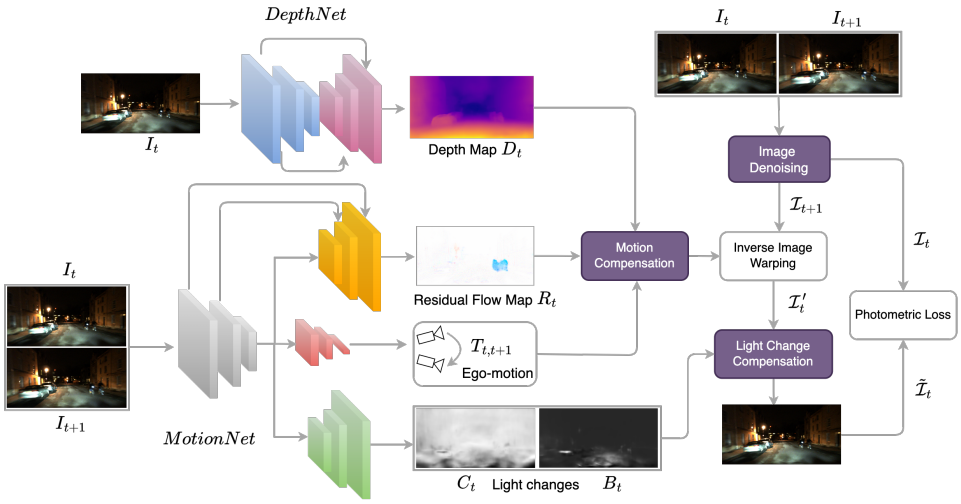


Figure 3.2: The overall architecture of the proposed method. It consists of two primary networks: DepthNet, which is responsible for estimating per-pixel depth maps, and MotionNet, which estimates ego-motion and the residual flow map to capture pixel motion between images. MotionNet also outputs light changes to account for pixel intensity variations caused by static and moving light sources. Rigid optical flow is derived from the depth and ego-motion estimates and then refined using the residual flow map in the motion compensation module. This refined flow aids in reconstructing the image $\mathcal{I}'_t \approx \mathcal{I}_t$ via the inverse image warping module, and light changes are corrected using the light change compensation module. The reconstructed image $\tilde{\mathcal{I}}_t$ is compared with the original $\mathcal{I}_t$ to compute the photometric loss.

3.3 Proposed Method

This section explains the method in more detail regarding the proposed contributions. The overall architecture is shown in Figure 3.2. The proposed method has two networks: DepthNet to estimate per-pixel depth maps and MotionNet to estimate the ego-motion and the residual flow map to account for the motion of all the pixels between the images. The MotionNet also outputs the light changes required to account for the pixel intensity changes that happened because of various static and moving light sources in the image. The depth and ego-motion estimates are used to estimate the rigid optical flow, which is further corrected using the residual flow map using the motion compensation module. This refined flow is fed to the inverse image warping module to reconstruct the $\mathcal{I}'_t \approx \mathcal{I}_t$ from denoised $\mathcal{I}_{t+1}$. Further, the reconstructed image is compensated for light changes between original I_t and I_{t+1} using the light change compensation module. Lastly, this reconstructed and refined image, $\tilde{\mathcal{I}}_t$, is compared with $\mathcal{I}_t$ to estimate the photometric loss. These modules are further described in detail in the following sections.

3.3.1 Baseline Method

We begin by reviewing the main principles behind existing photometric loss methods, which generally involve two networks: a depth network (referred to as *DepthNet*) and a motion network (referred to as *MotionNet*). The *DepthNet* takes a single colour image as input and predicts a depth map D_t for each image I_t in the sequence. The *MotionNet* uses a consecutive pair of images, I_t and I_{t+1}, to estimate the camera's ego-motion $T_{t,t+1}$ between them. With the estimated depth D_t and ego-motion $T_{t,t+1} = [R \,|\, t]$, we can reproject a pixel $\mathbf{u} = [u, v]^\top$ in the frame I_t into frame I_{t+1}, the subsequent image in the sequence. This reprojection is given by:

$$\dot{V}_t(\mathbf{u}) = K T_{t,t+1} D_t(\mathbf{u}) K^{-1} \dot{\mathbf{u}}, \tag{3.1}$$

where $\dot{\mathbf{u}}$ represents the homogeneous form of pixel coordinates $\mathbf{u}$, and $K \in \mathbb{R}^{3\times3}$ is the camera intrinsics matrix. The term $\dot{V}_t(\mathbf{u}) \in \mathbb{R}^3$ is the homogeneous form of $V_t(\mathbf{u}) \in \mathbb{R}^2$, which represents the new 2D position of the pixel in the image plane of I_{t+1}. Note that this reprojected point might not always fall within the actual image bounds.

By using the depth and ego-motion estimates, the pixel $\mathbf{u}$ in frame I_t can be reprojected into frame I_{t+1}, allowing the reconstruction of the image I'_t. This reconstruction is achieved by sampling from I_{t+1} around the reprojected points, using bilinear interpolation [181] to obtain a smoother result. Formally, the reconstructed image I'_t is expressed as:

$$I'_t(\mathbf{u}) = \begin{cases} \mathbf{W}(I_{t+1}, V_t(\mathbf{u})) & \text{if } \mathbf{u} \in M_t, \\ \mathbf{0} & \text{otherwise,} \end{cases} \tag{3.2}$$

Here, $M_t = \{\mathbf{u} : \rho(V_t(\mathbf{u})) \in \Omega(I_{t+1})\}$ represents the set of pixels whose reprojections into I_{t+1}, when rounded to the nearest pixel using ρ, fall within the image bounds $\Omega(I_{t+1})$, $\mathbf{W}$ is the interpolation function. The reconstructed image I'_t is then compared with the original image I_t to compute the loss for training.

The specific loss function used, known as *photometric loss*, has been widely applied in recent deep learning methods for depth estimation [178, 183, 37, 189]. It is typically calculated as a convex combination of pixel-wise differences and single-scale structural dissimilarity (SSIM) [182], formulated as:

$$L_p^{(t)} = \frac{1}{|M_t|} \sum_{\mathbf{u}\in M_t} \left(\alpha \frac{1 - SSIM(I_t(\mathbf{u}), I'_t(\mathbf{u}))}{2} + (1-\alpha) \left| I_t(\mathbf{u}) - I'_t(\mathbf{u}) \right| \right). \tag{3.3}$$

In this equation, α is a weighting factor that balances the SSIM term and the pixel-wise intensity difference. Most current unsupervised methods (e.g. [37, 183, 190, 192]) rely on

this formulation as the core of their models. To provide a fair comparison with state-of-the-art methods for night-time depth estimation [27, 220, 38], we base our modifications on the Monodepth2 framework [190], a commonly used baseline.

3.3.2 Lighting Change Compensation

The numerous point light sources that are typically turned on after dark (e.g. car headlights, lamp posts, etc.) can cause the illumination of a scene to change significantly from frame I_t to frame I_{t+1}. A problematic special case occurs when a light source moves with the camera (e.g. car headlights), which can lead to large holes in the estimated depth directly in front of the ego-vehicle [27, 220]. In our approach, we compensate for the illumination changes by estimating a per-pixel transformation that, when applied to I_{t+1}, can mitigate the changes in lighting that have occurred since I_t. We draw some inspiration from [238, 239, 209], which use a single whole-image transformation based on two scalar values to compensate for the difference in exposure time between a pair of images, based on the observation that such a difference creates approximately uniform intensity changes over the entire image. However, in our case, the intensity changes are far from uniform over the image, owing to both the motions of the ego-vehicle and other objects in the scene and the distances between the ego-vehicle and static point light sources. For this reason, we propose a per-pixel formulation here.

Our approach starts by passing the features produced by the last convolutional layer of the *MotionNet* through a *lighting change decoder* to estimate two per-pixel change images, C_t and B_t (see Figure 3.2). These (respectively) aim to capture the per-pixel changes in contrast (scale) and brightness (shift) that have occurred between the two input frames. As shown in Figure 3.4, the brightness image B_t broadly captures the extra light added to the image by, e.g. vehicle headlights, and the contrast image C_t broadly captures the changes in ambient light due to the motion of the ego-vehicle towards or away from point light sources such as street lamps. We use these images to transform the reconstructed image I'_t via $\tilde{I}_t = C_t \odot I'_t + B_t$, in which $\odot$ denotes the Hadamard product.

3.3.3 Motion Compensation

As seen in Section 3.3.1, the standard photometric loss makes use of correspondences between consecutive frames that have been established via reprojection based on the ego-motion and depth estimated by the networks. Assuming that (i) the ego-motion and depth have been estimated well, (ii) the scene is static, and (iii) there is minimal motion blur, the correspondences established in this way will broadly match those that would have been

established had we used the ground truth optic flow $\Phi_t(\cdot)$ from frame t to frame $t+1$. However, if objects move with respect to the background scene or anything visible in the image moves with respect to the ego-camera (which can cause motion blur), then the reprojection correspondences may be incorrect. To correct for these errors, we predict a residual flow map R_t, such that for each pixel $\mathbf{u} \in \Omega(I_t)$, $R_t(\mathbf{u}) \in \mathbb{R}^2$ is an estimate of $(\mathbf{u} + \Phi_t(\mathbf{u})) - V_t(\mathbf{u})$, the 2D offset from the reprojection correspondence of $\mathbf{u}$, namely $V_t(\mathbf{u})$, to its ground truth correspondence in frame $t+1$, namely $\mathbf{u} + \Phi_t(\mathbf{u})$. We can then add $R_t(\mathbf{u})$ to $V_t(\mathbf{u})$ for each pixel $\mathbf{u}$ to obtain a potentially more accurate correspondence for use in reconstructing I'_t via Equation 3.2.

Some methods [185, 184, 237] already exist that predict residual flow for daytime images. By contrast, we avoid using a separate encoder-decoder network or computationally intensive image warping-based bilinear interpolation for supervision. Instead, we estimate residual flow using an efficient sparsity-based formulation. This involves introducing a *residual flow decoder* that takes the features of the final convolutional layer of the *MotionNet* as input and the features of previous layers in the *MotionNet* via skip connections and outputs residual flow maps $\{R_{t,s} : s \in \{0,1,2,3\}\}$ at four different scales (each $R_{t,s}$ has a width and height that is $1/2^s$ that of I_t, and $R_{t,0} \equiv R_t$).

There is no direct supervision available to learn the residual flow maps. For this reason, we choose instead to encourage sparsity in the residual flow estimates so that the estimated depth and ego-motion can explain the majority of the scene, and the leftover can be explained by the residual flow maps. To achieve this, we adopt the sparsity loss from [237], i.e.

$$L_r^{(t)} = \sum_{s=0}^{3} \frac{\langle |R_{t,s}| \rangle}{2^s} \sum_{\mathbf{u} \in \Omega(I_{t,s})} \sqrt{1 + \frac{|R_{t,s}(\mathbf{u})|}{\langle |R_{t,s}| \rangle}}, \tag{3.4}$$

in which $I_{t,s}$ is a downsampled version of I_t at scale s, and $\langle |R_{t,s}| \rangle$ is the spatial average of the absolute residual flow map $|R_{t,s}|$. By contrast with [237], here we introduce a normalising factor of $1/2^s$ at each scale since the original loss was for scene flow, where the flow magnitude is independent of the resolution of the flow maps, which is not the case for the 2D residual flow we consider.

3.3.4 Image Denoising

Image noise is yet another key factor that affects the performance of the photometric loss. In practice, it is independent of the respective image and is mainly caused by a low SNR in the darker regions of the image. Handling this noise is of crucial importance, as photometric loss is the only training signal and supervises all of the modules we have mentioned thus far.

To remove the noise from the images, we chose to use Neighbour2Neighbour [227], a state-of-the-art unsupervised denoising model trained on ImageNet with zero-mean Gaussian noise. The standard deviation values were varied from 5 to 50 during training. This model can either be used to denoise all images input to the network at both training time and test time, or it can be used solely at training time to denoise the images for the purpose of calculating the loss. In practice, we chose the latter approach, as denoising at test time has two major disadvantages: (i) it can significantly add to the computational burden at runtime, slowing down the depth estimation, and (ii) any errors in the denoising process can lead to downstream errors in the depth maps, even though the depth estimation model itself might have been trained well. By contrast, restricting denoising to training time has the advantage of allowing us to make the depth and motion networks robust to noise by training them on the original images.

3.3.5 Full Pipeline

We can now formulate our full pipeline for depth, motion, and photometric loss estimation as follows: First, the depth map D_t for the image I_t is predicted by the depth network: $D_t = \mathbf{D}(I_t)$, where $\mathbf{D}$ is the DepthNet. Next, motion features f_n are extracted from the image pair $[I_t, I_{t+1}]$ by the motion encoder: $f_n = \mathbf{M}_{1:n}([I_t, I_{t+1}])$, where $\mathbf{M}$ refers to the Motion Encoder. The camera's ego-motion is predicted by the motion decoder: $T_{t,t+1} = \mathbf{P}(f_N)$. The residual flow is computed as $R_t = \mathbf{R}(\{f_n : 1 \leq n \leq N\})$, where $\mathbf{R}$ represents the residual flow decoder, $\mathbf{P}$ represents the ego-motion decoder, and N is the total number of encoder layers in $\mathbf{M}$. The lighting change decoder predicts the contrast and brightness as $(C_t, B_t) = \mathbf{L}(f_N)$. The images I_t and I_{t+1} are denoised using the denoiser: $\mathcal{I}_t = \mathbf{DN}(I_t)$ and $\mathcal{I}_{t+1} = \mathbf{DN}(I_{t+1})$. The next frame is warped using both rigid (V_t) and residual (R_t) flow to reconstruct $\mathcal{I}'_t$:

$$\mathcal{I}'_t = \mathbf{W}(\mathcal{I}_{t+1}, V_t + R_t).$$

The reconstructed image is compensated for light changes as $\tilde{\mathcal{I}}_t = C_t \odot \mathcal{I}'_t + B_t$. Finally, the photometric loss is computed similar to Equation 3.3 but with the denoised original and reconstructed images, $\mathcal{I}_t, \tilde{\mathcal{I}}_t$ respectively as:

$$L_p^{(t)} = \frac{1}{|M_t|} \sum_{\mathbf{u} \in M_t} \left(\alpha \frac{1 - SSIM(\mathcal{I}_t(\mathbf{u}), \tilde{\mathcal{I}}_t(\mathbf{u}))}{2} + (1 - \alpha) \left| \mathcal{I}_t(\mathbf{u}) - \tilde{\mathcal{I}}_t(\mathbf{u}) \right| \right),$$

where α balances the SSIM term and the pixel-wise intensity difference, and M_t is the set of valid reprojections.

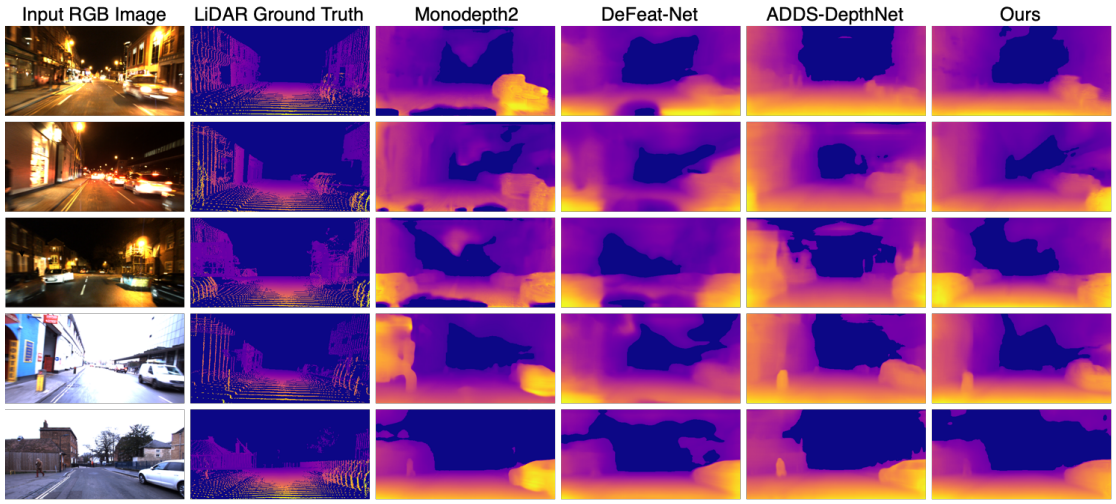


Figure 3.3: A qualitative comparison of our proposed method with the state of the art. Darker colours indicate greater depth. The ground truth data is generated by accumulating several temporal LiDAR frames centred on the input RGB image. As shown, Monodepth2 [28] (trained with standard photometric losses) struggles to estimate accurate depth on the road ahead of the ego-vehicle and fails to address dynamic objects, such as cars. The depth for these dynamic objects displays pixel intensities that differ from their immediate neighbours. A similar issue is observed with DeFeat-Net [27], despite its use of illumination-invariant feature maps. Additionally, ADDS-DepthNet has difficulty capturing the full structure of objects (e.g., the car in the first row) and dynamic objects (e.g., the pedestrian in the last row). Overall, our method recovers the full structure of objects and better handles depth estimation for dynamic objects compared to competing methods.

3.3.6 Making the Pipeline Bidirectional

Monodepth2 [190] calculates its photometric loss not only in the forward direction, from I_t to I_{t+1}, but also in the backward direction, from I_t to I_{t-1}, before combining the losses. This allows us to use the idea of minimum reprojection error to account for occluded pixels, so we do the same. We also adopt the auto-masking losses $L_a^{(t)}$ from Monodepth2 [190], as even though our method can cope with moving objects, it is very difficult to use parallax to disentangle the motion of objects that are moving in the same direction and at the same speed as the ego-vehicle. We further include the commonly used edge-aware gradient smoothing loss $L_g^{(t)}$ [190] to maintain spatial smoothness over the estimated depth maps. Our final loss $L^{(t)}$ then becomes the weighted sum

$$L^{(t)} = \min\left(L_{p-}^{(t)}, L_{p+}^{(t)}, L_{a-}^{(t)}, L_{a+}^{(t)}\right) + \lambda_r\left(L_{r-}^{(t)} + L_{r+}^{(t)}\right) + \lambda_g L_g^{(t)}, \tag{3.5}$$

in which $+/-$ denote the forward/backward versions of the losses, and $\lambda_r, \lambda_g \in \mathbb{R}$ are the weights.

3.4 Experiments

This section starts with the training and evaluation details in Section 3.4.1. In Section 3.4.2, we compare our depth estimation performance to a number of state-of-the-art approaches in a variety of different daytime and/or night-time contexts. In Section 3.4.3, we perform an ablation study to analyse the contributions made by the three individual components of our approach. We also present a study on the effect of parallax to help explain the importance of our neural intensity transformation module. In Section 3.4.4, we present the ego-motion evaluation followed by the ease of deployment of our method compared to the state-of-the-art methods in Section 3.4.5.

3.4.1 Training and Evaluation Details

We evaluated our method's performance on both day and night sequences from the Oxford RobotCar dataset [221]. Automotive scenarios provide a demanding yet essential testbed for monocular depth estimation, as vehicles encounter drastically changing illumination, from bright sunlight and reflections to nighttime conditions with headlights and streetlights, while also interacting with other dynamic agents like cars, cyclists, and pedestrians at various speeds. These conditions yield a broad distribution of pixel intensities, necessitating robust handling of overexposed and underexposed regions alike. Simultaneously, the camera mounted on a moving vehicle introduces motion blur, perspective shifts, and vibrations that further complicate accurate depth prediction. Moreover, complex urban structures such as multi-lane highways, tall buildings, and partially occluded objects add layers of difficulty, requiring algorithms to handle intricate 3D arrangements. This environment is crucial in practice because depth estimation underpins safety-critical functions in advanced driver-assistance systems (ADAS) and autonomous vehicles, including tasks like collision avoidance and pedestrian detection. However, data availability for thoroughly evaluating such algorithms across diverse conditions is limited; the Oxford RobotCar dataset, for instance, is one valuable resource that captures numerous environmental variations. All of our experiments are performed on this dataset.

The networks were trained for 15 epochs using PyTorch. The learning rate was set to 10^{-4}. The loss function weights λ_r and λ_g were both set to 10^{-3}. We used the popular Adam optimiser with $\beta_1 = 0.9$ and $\beta_2 = 0.99$ to train our network.

For depth evaluation, to avoid redundant computation due to the high camera frame rate, we sub-sampled the test set to have a 2m frame separation, resulting in 709 test images for nighttime and 702 images for daytime. We used the (sparse) LiDAR data provided with the dataset as ground truth for the evaluation, as no official per-pixel ground truth

depth is available. As per common practice, we used a script from the Oxford RobotCar SDK to reproject LiDAR points from several nearby frames onto the current test frame to increase the number of points available for evaluation. Like other methods trained using only monocular images, our method can only estimate depth maps up to a scale factor. When reporting results, we scale the predictions of all the methods using the median value approach suggested in [37].

3.4.2 Depth Evaluation

We compare with 4 state-of-the-art unsupervised monocular methods: Monodepth2 [190], DeFeat-Net [27], ADDS-Depth-Night [220] and RNW [38] (see Figure 3.3 and Table 3.2). We present Figure 3.3 to provide a qualitative comparison between our proposed method and state-of-the-art approaches. Darker colours represent greater depth. The ground truth is created by accumulating several temporal LiDAR frames centred around the input RGB image. As seen, Monodepth2 [28], which is trained with standard photometric losses, struggles to accurately estimate the depth of the road ahead (holes in the depth maps) and has difficulty handling dynamic objects, such as cars. For dynamic objects, please observe the varying pixel intensities compared to their nearby pixels. A similar issue is observed in DeFeat-Net [27], despite its use of illumination-invariant feature maps. Furthermore, ADDS-DepthNet [220] struggles to fully capture the structure of objects (e.g., the car in the first row) and dynamic elements (e.g., the pedestrian in the last row). In contrast, our method successfully recovers full object structures and more effectively handles depth for dynamic objects compared to other methods.

Before moving to the quantitative evaluation, we would like to highlight a problem with the current depth evaluation first and then present the comparison of the results later.

Depth Truncation: Typically, a depth estimation benchmark will specify one or more maximum distance thresholds, and then, for each such threshold τ, include only pixels whose ground-truth depth is $\leq \tau$ in the corresponding depth evaluation. This makes sense, since the intention is to determine the ability of a method to predict depths up to certain maximum distances.

However, for each τ, most (if not all) existing unsupervised depth estimation methods also truncate their *estimated* depth values to τ (as opposed to, e.g. counting pixels whose depth is estimated as being $> \tau$ as outliers). Unfortunately, this makes far less sense, and can have the unintended side-effect of biasing the evaluation in their favour. For example, consider a method that estimates a depth $d \gg \tau$ for a pixel $\mathbf{u}$ whose corresponding ground-truth depth is τ. Without truncation, the error for this pixel will be $|d - \tau| \gg 0$, but with truncation, it will be $|\min(d, \tau) - \tau| = 0$. In other words, this practice can lead to low

τ'	*Method*	*Abs. Rel.*	*Sq. Rel.*	*RMSE*	*Log RMSE*	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
50m	RNW [38]	0.180	1.525	6.183	0.259	0.740	0.913	0.962
	Ours	0.171	1.481	6.009	0.242	0.758	0.917	0.965
100m	RNW [38]	0.185	1.710	6.549	0.262	0.733	0.910	0.960
	Ours	0.174	1.637	6.302	0.245	0.754	0.915	0.964

Table 3.1: The effects of truncating the depths estimated by different methods to different thresholds, for a maximum evaluation distance of 50m, evaluated over our test set. The methods' performance appears significantly better when the estimated depths are truncated to 50m rather than 100m.

errors wrongly being ascribed to mispredicted pixels, which can cause the overall error that is reported to be much lower than it otherwise would have been.

To avoid this problem, and as an alternative to explicitly counting outliers, we truncate all estimated depth values in this work to a threshold $\tau' \gg \tau$, thus limiting the maximum penalty for each outlier to at most $\tau' - \tau$ (in practice, we set $\tau' = 100$m, which is much larger than $\tau = 50$m, the largest maximum distance for which we report results). The difference this makes to the evaluation can be seen in Table 3.1.

With this truncated depth evaluation, we tested our model with three different data variations: day only (d), night only (n), and a mix of day and night ($d\&n$). The quantitative evaluation is reported in Table 3.2. Monodepth2 [190] can be trained with all three configurations, although it was already outperformed by DeFeat-Net [27] in the $d\&n$ setting. Hence we skipped the evaluation of this method in this setting. For the day only referring to d and night only referring to n settings, we outperform Monodepth2 by a significant margin in both error and accuracy metrics. RNW [38], another recent method, is also built on Monodepth2 but targets nighttime data only. When compared, our method outperformed this method by 30% relative improvement in the RMSE metric. DeFeat-Net [27] and ADDS-Depth-Night [220] were trained with a $d\&n$ configuration on a split that is different from what we used, so we evaluated the pre-trained models they released on our test split. Our method outperforms both methods by a significant margin. Also, please note that we do not use any additional feature representation-based losses as used in DeFeat-Net [27], or paired day and night images as used in ADDS-Depth-Net [220].

Overall, the qualitative and quantitative evaluations presented in this section demonstrate the effectiveness of our approach. Despite its simple design, the proposed method outperforms competing approaches in addressing challenges related to poorly illuminated areas. In the next section, we present ablation studies conducted with different configurations of the proposed modules.

Test	*Method*	*Train*	*Abs. Rel.*	*Sq. Rel.*	*RMSE*	*Log RMSE*	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Day	Monodepth2 [190]	d	0.219	4.525	7.641	0.285	0.679	0.862	0.930
	Ours	d	**0.191**	**1.710**	**6.158**	**0.253**	**0.713**	**0.904**	**0.962**
	DeFeat-Net [27]	d & n	0.247	2.980	7.884	0.305	0.650	0.866	0.943
	RNW [38]	d & n	0.297	2.608	7.996	0.359	0.431	0.773	0.930
	ADDS-Depth-Night [220]	d & n	0.239	2.089	6.743	0.295	0.614	0.870	0.950
	Ours	d & n	**0.176**	**1.603**	**6.036**	**0.245**	**0.750**	**0.912**	**0.963**
Night	Monodepth2 [190]	n	0.453	21.310	11.420	0.444	0.700	0.873	0.930
	RNW MCIE + SBM [38]	n	0.350	7.934	8.994	0.407	0.674	0.861	0.922
	Ours	n	**0.186**	**1.656**	**6.288**	**0.248**	**0.728**	**0.919**	**0.969**
	DeFeat-Net [27]	d & n	0.334	4.589	8.606	0.358	0.586	0.827	0.911
	ADDS-Depth-Night [220]	d & n	0.287	2.569	7.985	0.339	0.490	0.816	0.946
	RNW [38]	d & n	0.185	1.710	6.549	0.262	0.733	0.910	0.960
	Ours	d & n	**0.174**	**1.637**	**6.302**	**0.245**	**0.754**	**0.915**	**0.964**

Table 3.2: A quantitative comparison of our method. The results of Monodepth2 [190] are reported after retraining it. Those of DeFeat-Net [27] and ADDS-Depth-Night [220] are reported using the checkpoints from their public GitHub repositories. The evaluation uses a maximum depth of 50m. Underlined methods use daytime images as main supervision or for regularisation losses.

3.4.3 Ablation Study

Lighting Change Compensation: In Figure 3.4(a), we show several reference images and their lighting change maps. The intensity changes are non-uniform, so we cannot use the existing correction approaches from [238, 239, 209]. We also observe that our method is able to clearly disentangle both the changes in ambient light resulting from movement towards/away from point light sources (captured by C_t) and the additional light added to the road pixels in the images by the ego-vehicle headlights (captured by B_t). Our neural intensity transform significantly reduces the RMSE error compared to the baseline (see Table 3.3), and is also able to fill in holes in front of the ego-vehicle (see Figure 3.3).

Motion Compensation: In Figure 3.4(b), we show several reference images and their residual flow and depth maps. In the second column, one can clearly see that our method is able to distinguish pixels on moving objects such as cars and pedestrians from static pixels. This effect can be observed for both daytime and nighttime images, showcasing the generality of our approach through a single unified training pipeline. In Table 3.3, it can be seen that correcting the reprojection correspondences using the residual flow map we predict leads to a significant improvement in accuracy.

Image Denoising: Denoising the images while calculating the training loss should ideally reduce the ambiguity in establishing pixel correspondences between the images, giving a robust supervision signal for training our system and thereby achieving lower error and higher accuracy. This effect can be clearly seen in the training error plot shown in

Stride	*Method*	*Abs. Rel.*	*Sq. Rel.*	*RMSE*	*Log RMSE*	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
1	Baseline	0.266	5.647	6.305	0.331	0.759	0.901	0.947
	+ NIT	0.190	1.824	4.848	0.257	0.763	0.919	0.965
	+ Denoising	0.163	1.256	4.193	0.224	0.801	0.935	0.973
	Full Model	0.154	1.174	4.120	0.216	0.811	0.939	0.976
2	Baseline	0.602	63.914	14.726	0.467	0.785	0.902	0.939
	+ NIT	0.169	1.727	4.693	0.236	0.812	0.929	0.967
	Full Model	0.131	0.926	3.731	0.188	0.852	0.949	0.980

Table 3.3: Ablation study demonstrating the impact of various modules in our system. For this study, the maximum depth was set to 30 m. 'Stride' refers to the intra-triplet stride. The top part of the table shows the progressive reduction in error and RMSE metrics as the proposed modules are incrementally introduced, starting from the baseline. In the second part, as the stride increases, the RMSE metric of the baseline model significantly worsens. However, when the Neural Intensity Transform (NIT) module is introduced, the RMSE decreases substantially, highlighting the importance and effectiveness of the proposed module.

Figure 3.4(c), where we compare our model with and without denoising. The denoising results in much more accurate depth maps, improving both the RMSE and accuracy metrics as shown in Table 3.3.

Effect of Parallax: To better understand how depth estimation performance is affected by increasing the average parallax (metric separation) between the images we use to calculate the photometric loss, we constructed a new nighttime training split by increasing the intra-triplet stride (see supplementary material) to 2, which increased the average parallax between the images from 0.353m to 0.706m. Without our neural intensity transformation, the depth estimation performance significantly decreased compared to the original nighttime training split (see the difference between the RMSEs of the baseline in the top and bottom parts of Table 3.3). A key cause of this in night images is likely the headlights of the ego-vehicle, which can cause the pixel intensities to change drastically between frames. However, with our neural intensity transformation, the depth estimation performance was found to instead increase, which we hypothesise to be because by compensating for the lighting changes, we make it possible to exploit the stronger supervision that can be offered by a wider baseline.

3.4.4 Ego-Motion Evaluation

To evaluate the ability of our method to estimate the ego-motion of the camera, we compare the absolute trajectory error (ATE) [37, 189] it can achieve on our nighttime test sequence to the ATEs of three state-of-the-art methods, namely Monodepth2 [190], DeFeat-Net [27] and ADDS-DepthNet [220] (see Table 3.4). We also visualise the overall trajectory we estimate and compare it to those of other methods, as well as the ground truth (see Figure 3.5). Note

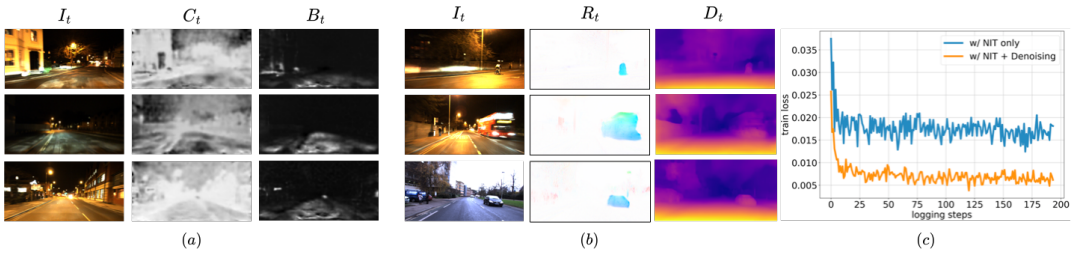


Figure 3.4: Visualizations of (a) estimated light changes: The first column captures overall ambient changes across the entire image, while the second column predominantly captures additive light sources, such as car headlights. (b) Residual flow and estimated depth: The estimated ego-motion does not account for pixels belonging to the object shown in blue. However, when this residual flow is added to the ego-flow, it results in disparity maps that are consistent with neighbouring pixels. (c) The effects of denoising on the training loss over time: The loss value is significantly lower from the start, demonstrating the importance of denoising the images before calculating the photometric loss.

Method	$t_{ate}\ (m) \downarrow$	$r_{ate}\ (rad) \downarrow$
Monodepth2 [190]	0.01317 $\pm$ 0.01517	**0.00036** $\pm$ 0.00040
DeFeat-Net [27]	0.03760 $\pm$ 0.01990	0.00115 $\pm$ 0.00129
ADDS-DepthNet [220]	0.01312 $\pm$ 0.01386	0.00083 $\pm$ 0.00061
Ours	**0.01310** $\pm$ 0.01348	0.00041 $\pm$ 0.00039

Table 3.4: A quantitative comparison of our ATE errors to those of Monodepth2 [190], DeFeat-Net [27] and ADDS-DepthNet [220] on our nighttime test sequence. (The results are in the form mean $\pm$ std.)

that we estimate the ego-motion only up to scale, as our method uses monocular images during training. Moreover, the scale can drift over the course of the sequence, as there is no external constraint enforced to keep it constant.

We used the rescaling approach described in [37] for both our quantitative and qualitative pose estimation results. In both cases, our proposed method achieves results that are competitive with the state-of-the-art.

3.4.5 Ease-of-Deployment

To better understand the relative deployability of our method, we compare it to three state-of-the-art approaches, namely DeFeat-Net [27], ADDS-DepthNet [220] and RNW [38]. Besides outperforming the existing models, our method also has several other advantages over them, including easier data and training requirements, as shown in Table 3.5.

DeFeat-Net [27] is trained with an additional feature space with pixel-wise contrastive learning, which is assumed to be illumination and weather invariant. This feature network is

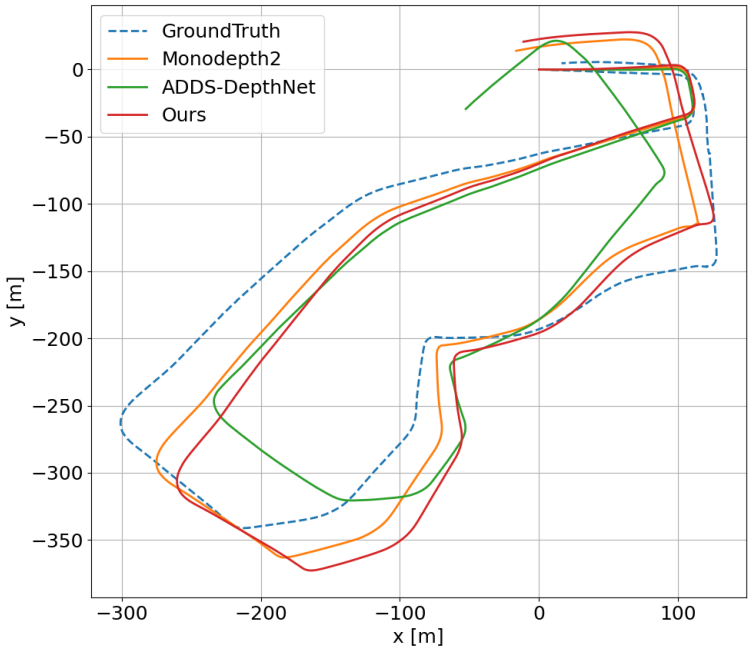


Figure 3.5: Comparing the trajectory estimated by our method for our nighttime test sequence to the trajectories estimated by Monodepth2 [190] and ADDS-DepthNet [220], and to the ground truth.

simultaneously learned with the depth and pose networks. We used their pre-trained model to compare against our method.

ADDS-DepthNet [220] is compared against our method using the pre-trained models released alongside their code on GitHub. The orthogonality loss proposed in their method requires corresponding daytime data. It is not possible to train this model with night-only data.

RNW [38] did not release their pre-trained models alongside their code. We retrained their model with our training data and reported the results. This method uses a GAN-based regularisation loss, through which they achieve an $\approx 93\%$ improvement over the baseline method. To achieve this, they needed to carefully train a model using daytime images, as the nighttime performance directly depends on how well the daytime model is trained.

The results reported in their paper are trained with a ResNet-50 model, whereas the model we use is ResNet-18. When we tried retraining their approach with ResNet-18, we observed a divergence of training after a few epochs. Training instability is a well-known issue that must be overcome when working with GANs, but this does highlight the relative difficulties of making even simple modifications to a GAN-based system such as RNW in

Requirement	*DeFeat-Net* [27]	*ADDS-DepthNet* [220]	*RNW* [38]	*Ours*
No additional features	✗	✓	✓	✓
No daytime images	✓	✗	✗	✓
No paired day-night images	✓	✗	✓	✓
No GAN losses	✓	✓	✗	✓
Night-only training	✓	✗	✗	✓
Works for day and night	✓	✓	✗	✓

Table 3.5: Ease-of-deployment comparison of our method with three state-of-the-art approaches

comparison to our own approach. Notably, in spite of such training difficulties, RNW is the only approach that is very close to our method in terms of its error and accuracy metrics. However, unlike our method, their model can only be used for nighttime depth estimation.

3.5 Limitations and Future Work

Our method has several limitations. First, as is typical with monocular methods, it can only estimate depth up to an unknown scale. Second, like most stereo approaches—whether variable-baseline like ours or fixed-baseline with a rigid stereo rig—it struggles to preserve detail in distant parts of the scene due to limited parallax. Third, the method also has difficulty recovering structural details from very dark image regions (e.g., see Figure 3.1(b)). Lastly, similar to most vision-based depth estimation methods, it does not consistently perform well on transparent surfaces such as glass.

To address the issue of scale, one solution could be to train the model using stereo images with a known baseline. Another limitation of our method is that it requires end-to-end training, meaning it must be retrained for a new scene if it significantly differs from the Oxford RobotCar dataset. Although the self-supervised nature of the method offers flexibility for retraining, it would be more efficient in terms of cost and space to develop a model that works across a variety of scenes without retraining.

Recently, there has been a surge in learning task-agnostic features using self-supervision, showing promise in tasks such as supervised depth estimation and segmentation [33]. This approach could potentially be leveraged to create a more generalisable model. Furthermore, exciting advances in generative learning, such as denoising diffusion models, offer new possibilities. For example, rather than directly regressing depth for transparent objects, we could explore leveraging the generative capabilities of these models to generate the missing depth information for such surfaces.

A potential solution to these challenges, particularly poor illumination, lies in the use of advanced low-light sensors with a wider dynamic range, such as Sony's Starvis technology [1]. Investigating how these cameras can enhance the performance of current depth and ego-motion estimation algorithms under low-light conditions presents an exciting avenue for future research.

3.6 Potential Extensions in the Era of Foundational Depth Estimation

The foundational models for monocular depth estimation have advanced considerably in the recent years. The latest approaches such as UniDepth [240], Metric3D variants [241, 242], and Depth Anything variants [91, 108] that leverage modern architectures like Vision Transformers (ViTs), which offer significant improvements over the CNN-based models discussed in this chapter. Upgrading to these newer architectures could lead to marked gains in accuracy and generalization.

These state-of-the-art foundational models are typically trained in a supervised manner using large datasets that include both synthetic and real-world images paired with ground-truth depth data. This extensive training enables them to learn robust visual representations from single images, yielding superior performance on single images. However, when evaluated for temporal consistency, these models often struggle [243], requiring additional training with video datasets. For instance, the recent Video Depth Anything [142], currently the state-of-the-art for monocular depth estimation in 2025, introduces a depth consistency loss for dynamic scenes based on optical flow. Since ground-truth optical flow is unavailable for every training sample and existing models can be noisy, this method employs proxy losses, such as temporal gradient matching, to enforce temporal consistency. While the focus on achieving temporally consistent depth estimation is growing, the scarcity of large video datasets with ground-truth depth and optical flow data remains a significant challenge. This limitation underscores the need for self-supervised objectives like those proposed in this chapter.

Although rapid advances in deep learning inevitably bring constant changes to network architectures, many of the core principles introduced here are likely to remain central to future depth estimation research. The neural intensity transform plays a pivotal role by preserving the temporal consistency of illumination in video sequences, thereby stabilizing depth estimates across varying lighting conditions. Leveraging a residual flow approach, regularized by sparsity constraints, offers a lightweight yet powerful mechanism to handle

[1] https://www.sony-semicon.com/en/technology/security/index.html

dynamic objects without resorting to full optical flow networks; this not only reduces computational overhead but also effectively addresses the complexities introduced by moving entities in the scene. Meanwhile, even though low-light cameras continue to improve, achieving lower noise levels and capturing higher-quality data, there will inevitably be scenarios with low signal and high noise which could lead to inaccurate the photometric loss. Consequently, denoising will continue to be an essential step in reducing artifacts and preserving depth accuracy, especially in low-light conditions.

Overall, as novel encoders, decoders, and other architectural components evolve to accommodate more demanding tasks and datasets, the underlying objectives such as enforcing temporal consistency, robustly managing dynamic objects, and mitigating low-SNR issues, are going to remain integral to the development of state-of-the-art depth estimation models.

3.7 Conclusions

Photometric loss has become a standard go-to loss function for training deep learning models in depth and ego-motion estimation. While this approach performs well during the day, it struggles significantly with images captured under non-uniform lighting conditions, such as nighttime scenes. These challenges are further compounded by the presence of moving objects, light sources, and noise commonly found in images captured during driving scenarios.

In this chapter, we propose improvements to the photometric loss function to train a more robust and accurate illumination-invariant model. Our first contribution is addressing the lighting changes that occur between temporally aligned consecutive images. We introduce a light change decoder that estimates the amount of lighting change between frames. During our experiments, we found that inconsistent illumination changes negatively affected depth estimation, especially when the time interval between consecutive images increased. Typically, a larger parallax between corresponding pixels leads to better depth estimation. However, lighting changes distorted the photometric loss calculation, thus impacting depth estimation accuracy. Once we addressed these lighting changes, we were able to fully leverage the benefits of larger parallax for improved depth estimation.

Camera motion alone is insufficient to account for every pixel in an image, especially in dynamic scenes where pixels may move or blur. To address this issue, instead of estimating optical flow as in existing methods, we estimated residual flow. This extends rigid flow—calculated using depth and ego-motion—to handle dynamic pixels. This approach helped mitigate errors in loss calculation for dynamic or blurred pixels.

Additionally, low-light conditions cause longer exposure times, leading to noise accumulation in underexposed areas. This noisy data degrades the quality of the depth estimation when using the photometric loss. To counter this, we adopted a simple Gaussian noise assumption and used a pretrained deep learning model to handle the noise. By denoising the images during training, the model received cleaner signals, significantly improving the final depth estimation results.

Finally, we address a gap in the Oxford RobotCar dataset: the absence of a standard train/test/validation split. Previous splits often contained geographic overlaps between sets, resulting in unfair evaluations. We proposed new splits without geographic overlap to ensure a more rigorous evaluation process. Various ablation studies and experiments on these splits demonstrated the effectiveness of our method. When trained with our improved photometric losses, we observed a 30% improvement in the RMSE metric compared to the closest baseline nighttime method, RNW [38].

In conclusion, this chapter addresses the first question of how we can repair and reuse photometric losses, enabling their application in both daytime and nighttime scenarios through a simple yet effective combination of proposed contributions. In the next chapter, we will explore how depth estimation can be achieved using spatial photometric loss by leveraging left and right images at time t, as opposed to the temporal approach used in this chapter, which utilizes images from $t-1$, t, and $t+1$.

Chapter 4

Self-supervised Nighttime Stereo Depth Estimation using Visual Foundation Models

In this work, we address the key limitations of both end-to-end learning and monocular depth estimation methods like ours in Chapter 3. End-to-end approaches often struggle to generalize to out-of-training dataset images, while monocular methods suffer from scale ambiguity, only regressing depth up to a scale factor. To achieve a true metric scale for depth and ego-motion estimation, additional sensors or known reference frames are typically required. Furthermore, self-supervised depth estimation algorithms, which rely on frame-warping relationships, experience significant performance degradation under low-visibility conditions, such as nighttime or in environments with varying illumination.

To overcome these challenges, we propose a novel self-supervised stereo depth estimation algorithm that combines stereo images with features extracted from large pretrained visual foundation models. By leveraging stereo images, we address the scale ambiguity inherent in monocular methods, while the use of robust pretrained features ensures strong generalization, even in low-visibility and noisy nighttime conditions. Our approach introduces an efficient feature-matching technique for integrating stereo frames and includes a novel masking strategy that filters out pixels violating photometric consistency assumptions within the feature space, thereby improving the accuracy of depth predictions.

Additionally, we introduce new evaluation metrics to address the shortcomings in current depth estimation assessments. Extensive experiments on challenging datasets, including Oxford RobotCar and Multi-Spectral Stereo, demonstrate significant improvements in both performance and generalization across diverse environments, showcasing the effectiveness of our algorithm.

4.1 Introduction

Depth estimation is a pivotal subject within computer vision, with wide-ranging implications for applications such as Virtual Reality (VR), Augmented Reality (AR), autonomous driving, and robotics [98, 99]. There has been a significant amount of research in both monocular [19, 105, 96, 102, 244] and stereo settings [112, 39, 128] to estimate depth from images. Despite the accomplishments of supervised depth estimation algorithms, these methods typically depend on acquiring high-resolution ground truth data - a challenging endeavour that necessitates substantial computational resources, costly 3D LiDAR sensors, and heavy computational requirements [180, 207]. Due to this, only limited data is available to train these methods. This often leads to these models overfitting to the training domain and failing to generalize to new scenarios. An example of such a scenario is shown in Figure 4.1. IGEV-Stereo [39], a SOTA stereo method that is trained on a huge amount of ground truth data, was observed to completely fail when tested on nighttime data.

Addressing the need for ground-truth data, recent research has shown interest in self-supervised depth estimation methods [37, 189, 180, 178, 245]. Self-supervised approaches typically warp the source frame to the target frame with the learned depth information and estimated pose in a monocular setting or precalibrated baseline distance in a stereo setting. Photometric loss is computed between the warped and the actual target frame to constrain the learning process. While monocular depth estimation algorithms are widely applicable, they often lack scale information [246] and exhibit limited generalisability [247]. In contrast, self-supervised stereo depth estimation algorithms leveraging the correspondence between left and right frames not only yield more accurate performance [248] but also address the scale problem.

Self-supervised stereo depth estimation, however, relies on photometric consistency assumptions and conventional warping relationships, which are constrained by favourable lighting conditions. These assumptions break down in poorly illuminated scenarios such as nighttime, characterised by low SNR, low texture, and fluctuating illumination, as explained in Chapter 3. There have been some self-supervised methods for nighttime conditions recently in the monocular setting [26, 29, 38, 27], but not many methods for the stereo setting. The only available method was [249], and it was trained in a supervised fashion using auxiliary losses calculated from a GAN-based image translation framework. Also, these stereo depth estimation techniques [249, 248] tend to combine stereo features indiscriminately without addressing regions that have correspondence issues in both supervised and self-supervised settings.



Figure 4.1: A comparison of the estimated disparity using our method (Ours) with the SOTA stereo-matching methods, IGEV-Stereo [39]. Note how the sky (blue rectangle) is incorrectly estimated in IGEV-Stereo as being very near - this is due to the lack of ground truth points for training. Similarly, there is a lack of detail showing the edge of the wall and the lamp-post (green rectangle). In comparison, our method based on self-supervision is able to accurately estimate these depths.

Recently, there has been a huge success in training self-supervised general feature encoders on hundreds of millions of images [33, 36, 41]. These are called visual foundational models, and they provide features that are robust to different illuminations, noise, and geometric variations in the image data, which are some of the problems that the current stereo methods are suffering from. In light of this, our approach concentrates on obtaining accurate self-supervised stereo depth estimation for poorly illuminated conditions such as the nighttime using features extracted from a foundational model. Also, we keep the foundational encoder frozen throughout and train only a stereo-matching decoder. The primary contributions of this chapter can be summarized as follows:

1. We present an architecture capable of efficient self-supervised stereo depth estimation for poorly illuminated scenarios, such as the nighttime, using frozen visual foundation models and a photometric loss function.

2. We introduce a feature-level mask to mitigate the impact of pixels violating illumination assumptions and a distance regularizer to enhance the feature descriptions to estimate accurate depth maps.

3. We provide a more rational set of evaluation metrics to assess the performance of depth estimation. Also, we conducted comprehensive experiments on two datasets to showcase the efficacy of our approach. Detailed ablation studies were performed to substantiate the benefits of each proposed module in our method.

The rest of the chapter is organized as follows: related work is discussed in Section 4.2 followed by the proposed method and various experiments and ablation studies in Section 4.3

and Section 4.4 respectively. Lastly, the chapter concludes with some promising directions for future work in Section 4.5.

4.2 Related Work

Photometric consistency assumes the scene to be static, free of any noise or occlusions, and reflectance to be Lambertian and temporally invariant. Various changes in the appearance of scenes during nighttime cause systems trained only for daytime datasets to fail. Approaches are required to deal the inherent issues with scene lighting, such as lack of the sun as a primary source of light.

4.2.1 Nighttime Self-Supervised Depth Estimation

Loosely, we classify previous work as being based either on domain adaptation or image enhancement. Adaptation-based approaches seek to overcome the domain shift between day and night using synthesised data to align night/day or clear/inclement weather image encodings [250, 251], aiming to obtain the same features despite different conditions. Synthesised data is created using standard datasets (i.e. [221]) and a Generative Adversarial Network (GAN). [252] use a Monodepth2 [28] architecture with a domain adaptation framework using GAN-based training to learn a nighttime image encoder that could extract features that are similar to the daytime from nighttime images. The day-like features are further fed to the pretrained day-decoder to estimate depth for the nighttime images. [253] emulate this approach but train in the output space as well as the feature space. [254] use the GAN outputs to train two feature extractors, a day-night invariant feature extractor that forms the backbone for depth prediction, as well as night- and day-specific style extractors for training. Outside of our broader categorisation, [255] use a feature space (rather than image space) contrastive loss to improve domain generalisation beyond that of [28].

Enhancement-based approaches aim to improve performance on day models by taking greater account of scene lighting, for example, by isolating illumination information within the image. [256] use a learned image enhancement and adapted masking from [28]. [257] use a complex, partially adversarial architecture with a learned image enhancement to estimate illumination and uncertainty, which is then masked from the loss. [29] estimate changes in illumination between frames, which are then used to compensate for such changes when estimating ego-motion and improving depth estimation.

Concurrent state-of-the-art approaches [250] and [251] also consider weather conditions in their depth estimation approaches. Both adapt [28] for their all-day unified networks, using GANs to augment for both weather and lighting conditions. [250] uses a semi-augmented

warping to minimise GAN-induced inconsistency between consecutive frames and uses unaugmented inputs for pose estimation to minimise error propagation. Similarly, [251] add the weak velocity supervision of [258] to the loss used in [28] to improve scale-awareness in the pose model.

4.2.2 Supervised Stereo Depth Estimation

All approaches mentioned so far are monocular during the inference, even if stereo images are used for training. In terms of daytime-only fully-stereo depth estimation, state-of-the-art works include [128] and [39]. [249] present the only other nighttime stereo approach. All of these methods are supervised. [128] use a transformer and 4D correlation feature matrix, including an iterative refinement inspired by [127], to derive depth, flow, and disparity. [39] use similar iterative refinement but split the cost volume into sub-maps to find correlations. [249] augment between both night-and-day and day-and-night, using stereo networks to predict disparity maps for day and night images separately. Our method benefits from not requiring data ground truth depth maps to train as it is self-supervised.

4.2.3 Visual Foundation Models

Visual foundation models can be characterised by large model architectures which are trained on extensive datasets. They can be task-agnostic [33] or have a bias to a given task, for example, semantic segmentation [219]. Many foundational models make use of Vision Transformers (ViT) [32] as a backbone architecture. In general, such models are pre-trained in a self-supervised manner with contrastive losses on what can be, in some cases, hundreds of millions of images [33]. [33] builds on [35], adding various elements to the loss including a KoLeo regulariser [259], which encourages a more uniform distribution across feature space. [36] distills multiple foundation models—[33], [219], and [34]—into a single model. Importantly, each of these models has been observed to have gained a fine semantic or spatial understanding of scenes, resulting in this information being embedded within the features they output [217], which is fundamental for applications like stereo matching.

4.3 Proposed Method

Given a pair of stereo images (I_l, I_r), we aim to estimate a per-pixel depth map $\mathbf{d} \in \mathbb{R}^{H \times W \times 1}$ using a completely self-supervised learning approach. The proposed framework, as shown in Figure 4.2, consists of four main components: the feature extractor, projection head, stereo matcher, and an upsampler. The feature extractor is a foundational model of choice that

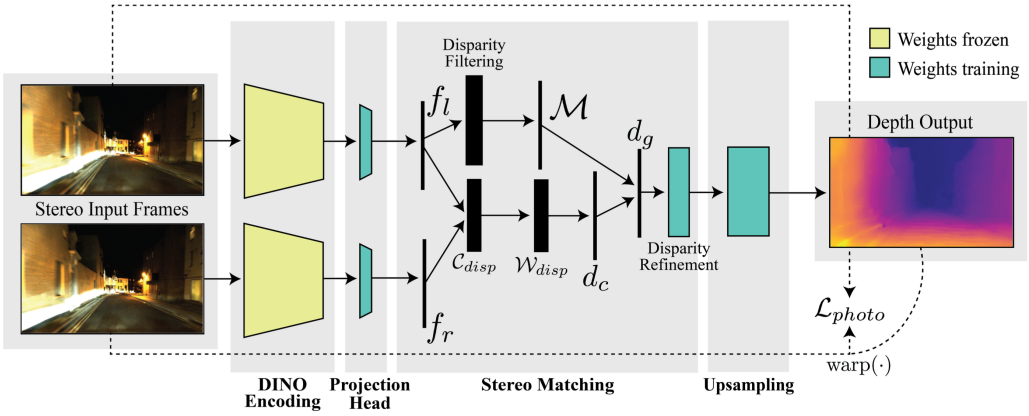


Figure 4.2: Our approach consists of four main components. First, features are independently encoded for each input using DINO [41]. These features are then processed through a learnable projection head, which adapts and reduces their dimensionality, producing f_l and f_r. Next, stereo matching is performed on these features, and disparity filtering generates the mask $\mathcal{M}$. The combination of f_l and f_r produces the correspondence volume $\mathcal{C}_{disp}$. By applying softmax to $\mathcal{C}_{disp}$, we compute the disparity weights $\mathcal{W}_{disp}$, which are used to obtain the coarse disparity d_c. Finally, the coarse disparity and the mask are combined to calculate the global disparity d_g, which is then refined and upsampled to produce the final depth map.

encodes the input stereo images into feature representations. These features are then passed through a projection head that reduces their dimensionality. The downsampled features from both the left and right images are further processed by the stereo matcher to estimate disparity maps. Finally, the upsampler increases the resolution of these disparity maps to match that of the input images. Each of these modules is explained in more detail in the following sections.

4.3.1 Feature Extractor

In contrast to existing approaches, we utilize visual foundation models, such as DINO [41], DINOV2 [33], AM-RADIO [36], or similar architectures, as feature extractors for the input images. These models are pretrained on large-scale datasets like ImageNet [260], Facebook LVD-142M [33], or equivalent datasets, providing rich semantic representations. In our experiments, we employ pretrained *small*-scale models, where the patch size is denoted by P and the stride by S. Specifically, we use $P = 8$ and $C = 384$ for DINO-V1, $P = 14$ and $C = 384$ for DINO-V2, and $P = 16$ and $C = 1280$ for AM-RADIO. For the remainder of the chapter, we refer to the visual foundation model in use DINO-V1 with patch-size 8, as the encoder, unless otherwise specified.

Our encoder consists of an initial *convolutional* layer followed by a series of *transformer* layers. Given an image $I \in \mathbb{R}^{H \times W \times 3}$, the convolutional layer converts the image into

overlapping patches of size $P \times P$ with a stride of S. This results in an intermediate feature map $\hat{f} \in \mathbb{R}^{h' \times w' \times C}$, where the spatial dimensions h' and w' are calculated using the formula for convolutional output size: $h' = \left\lfloor \frac{H-P}{S} + 1 \right\rfloor$ and $w' = \left\lfloor \frac{W-P}{S} + 1 \right\rfloor$.

The resulting feature map $\hat{f}$ is then processed through a subset of transformer layers. The intermediate feature map $\tilde{f} \in \mathbb{R}^{h' \times w' \times C}$ is further subsampled with a stride of 2 along the height and width dimensions, producing approximately $8\times$ downscaled feature map. This map is processed through the remaining transformer layers to yield the final feature map $f \in \mathbb{R}^{h \times w \times C}$, where $h = \lfloor h'/2 \rfloor$ and $w = \lfloor w'/2 \rfloor$.

The choice of transformer layers is informed by insights from [217], which suggest that transformer layers function hierarchically, similar to convolutional neural networks (CNNs), with deeper layers capturing more global semantics and shallow layers capturing finer local details (such as spatial positions). Intermediate layers are believed to balance semantic and positional information. To enhance the accuracy of disparity estimation, both f and $\tilde{f}$ are utilized for stereo matching.

4.3.2 Projection Head

The projection head takes feature vectors $f, \tilde{f}$ from the encoder and projects them into lower dimensional space $\mathbb{R}^C$ where $D < C$. This is done because (1) when PCA is performed on the C dimensions of DINO, we observe the 10 first principal components to explain more than 50% of the total variance, suggesting it is safe to project to lower dimensional space [1] and (2) the computational complexity of the stereo matcher during inference and training will be reduced. The projection head consists of two *conv*-layers with kernel size as 1×1, and ReLU as activation function in the middle. We use $D = 128$ as the output description dimension and the same projection head for both left and right features.

4.3.3 Stereo Matcher

Our stereo-matcher is inspired by Unimatch-Stereo [128]. The stereo matcher takes the multi-scale left and right features $\{(f_l, f_r), (\tilde{f}_l, \tilde{f}_r)\}$ as input and estimates the disparity map $\tilde{d}_r \in \mathbb{R}^{h' \times w' \times 1}$ in three stages. First, features are enhanced with cross-image feature context using a transformer module. Then, a disparity map d_g is estimated at $1/8$ scale using the (f_l, f_r) features and global matching. Lastly, the disparity is bilinearly interpolated by $2\times$ and corrected for the interpolation artefacts with a residual disparity estimated using $(\tilde{f}_l, \tilde{f}_r)$ and local matching. These modules are explained in detail in the following sections.

[1] similar observations were made for the other foundational models.

Transformer: Stereo images are processed through the encoder and projection head individually, meaning no cross-image interaction or information exchange. In previous works [128, 168, 150], it is suggested that adding cross-image information results in better matching accuracy. We, therefore, use a transformer module similar to [128] with fixed positional encoding after the projection head. To further reduce computational complexity, we only aggregate features that are on the respective epipolar line for any given feature. We use rectified stereo-images that make the search problem 1D, as the epipolar lines are strictly straight. Given f_l and f_r, the transformer module outputs $\mathbf{f}_l$ and $\mathbf{f}_r$ in $R^{h \times w \times D}$.

Coarse Disparity Estimation: The output features from the transformer module are used to compute the dense correspondence volume C_{disp} using normalized feature correlation (i.e., global matching) with a simple matrix multiplication

$$\mathcal{C}_{disp} = \frac{\mathbf{f}_l \cdot \mathbf{f}_r^\top}{\sqrt{D}} \in \mathbb{R}^{h \times w \times w}. \tag{4.1}$$

We obtain the matching distribution $\mathcal{W}_{disp}$ using a softmax over the last dimension of $\mathcal{C}_{disp}$. This is then multiplied with a 1D pixel grid $\mathcal{P}_{\text{1D}} \in \mathbb{R}^w$ to obtain the corresponding pixel locations $\mathcal{G}_{\text{1D}} \in \mathbb{R}^{h \times w}$. Finally, the coarse disparity d_c can be computed as the difference between $\mathcal{P}_{\text{1D}}$ and $\mathcal{G}_{\text{1D}}$. Formally, this can be written as:

$$\begin{gathered} \mathcal{W}_{disp} = \text{softmax}(C_{disp}), \text{ and } \mathcal{G}_{\text{1D}} = \mathcal{W}_{disp}\mathcal{P}_{\text{1D}}, \\ d_c = \text{ReLU}(\mathcal{G}_{\text{1D}} - \mathcal{P}_{\text{1D}}), \end{gathered} \tag{4.2}$$

where ReLU ensures the disparity is always positive.

Disparity Filtering: One observation made during our experiments is that features belonging to areas of low texture, particularly the sky, resulted in very noisy disparity estimates similar to [109, 39] as shown in Figure 4.1. We explain this as being due to camera sensitivity and longer exposure timings in low light, causing an accumulation of noise, resulting in erroneous and noisy features. Such features exacerbate incorrect feature matches, causing noisy estimates, as we are extracting disparity as a byproduct rather than estimating it directly as in [249]. To address this, we propose a simple yet effective solution based on intra-image feature description distances to mask the noisy areas. We conjecture that features belonging to noisy areas tend to have a lower minimum distance from their nearest neighbours compared to the features that belong to well-lit areas. In a set of n features, given a feature $f_i \in f$ with $i \in \{1, 2, 3..., n\}$: (1) we normalize them to have unit length; (2) we estimate one to all cosine similarity to find the nearest neighbour f_j, with $j \neq i$; and (3) we calculate l_2 distance p_i between f'_i and f'_j. Formally, this is, $f'_j = \text{argmax}_j(f'_i \cdot f'^\top_j)$, $j \neq i$ where $f' = \frac{f}{||f||}$ and $p_i = \left|\left|f'_i - f'_j\right|\right|$. Estimated distances are then used to filter good

disparity values from noisy ones by estimating a mask $\mathcal{M}$, which is used to find masked disparity d_m as:

$$d_m = \mathcal{M} \cdot d_c, \quad \mathcal{M} = \begin{cases} 1, & \text{if } p_i > \zeta, \\ 0, & \text{otherwise}, \end{cases} \tag{4.3}$$

where ζ is set to 0.2 in our experiments.

Disparity Propagation However, some poorly lit regions that contain useful information may get lost during masking. This creates holes in the estimated disparity. We, therefore, propagate the disparity at the valid pixels to the masked-out pixels by measuring self-similarity, as in [128]. This is done using an attention layer to find global disparity, d_g:

$$d_g = \text{softmax}\left(\frac{f_l \cdot f_l^\top}{\sqrt{D}}\right) \cdot d_m \tag{4.4}$$

Disparity Refinement: The current global disparity d_g is at $1/8\times$ resolution. To increase the resolution, we upsample d_g using bilinear interpolation by $2\times$. Doing so inevitably introduces interpolation artefacts, so we use fine-level features $\tilde{f}_l$, $\tilde{f}_r$ to account for this. We first warp the right-features $\tilde{f}_r$ onto the left frame using the upsampled disparity d_g^{up}. The original left-features $\tilde{f}_l$ and the warped-features $\tilde{f}_l'$ are then passed through the transformer, disparity estimation, and filtering to estimate residual disparity ∇d. Using localised attention on subsampled features in the refinement process is suggested in [128] to improve accuracy through local feature interactions. The residual disparity is added to the upsampled disparity, giving $d_r = d_g^{up} + \nabla d$. Finally, the refined disparity d_r is correlated with $\tilde{f}_l$ to output disparity $\tilde{d}_r \in \mathbb{R}^{h' \times w' \times 1}$.

4.3.4 Upsampling

The disparity output $\tilde{d}_r$ is currently at 1/4 of the image resolution; this is further upsampled to match input image resolution disparity $d \in \mathbb{R}^{H \times W \times 1}$. Instead of bilinear interpolation, which can result in blurry outputs, we use the learnable convex upsampling method proposed by RAFT [127]. The value of each pixel in the final output disparity is a convex combination of values extracted with a 3×3 grid placed on its location at coarse resolution. The weights for this combination are learned using two convolutional layers followed by a softmax operation at the end.

4.3.5 Training Losses

Photometric Loss: The estimated disparity is used to reconstruct the left image from the right using bilinear interpolation. This reconstruction can be performed either using 2D warping directly, based on the estimated disparity as done in [180], or in 3D using the

camera intrinsics, K, and the extrinsics, T_{lr}, as done in [28]. We evaluate our model on both of these warpings in Section 4.4. The reconstructed image is then compared with the original left image to compute the photometric loss. Given the left and right images, (I_l, I_r), and the estimated disparity d, the left-image is reconstructed from the right image as $\hat{I}_l = \text{warp}(I_r, d)$, the photometric loss is defined as:

$$L_{\text{photo}} = \alpha|I_l - \hat{I}_l| + (1-\alpha)\text{SSIM}(I_l, \hat{I}_l), \tag{4.5}$$

where, the term α is a convex combination weight between the L_1 and SSIM losses, and is set to 0.15.

Distance Regularizer: We also encourage all of the features to maximise the minimum distance from their nearest neighbour using a regularization loss inspired by [259]. This allows the features from poorly lit areas to improve. However, there is an imbalance between well and poorly-illuminated areas in the majority of our training split images, similar to the class imbalance problem from classification literature. This is addressed by using a modulation factor, γ, to reduce the concentration of loss on features that already have higher minimum distance, p_i, with their nearest neighbour and to focus more on small feature distances. Formally:

$$L_{\text{reg}} = -\frac{1}{n}\sum_{i=0}^{n}(1-p_i)^{\gamma}\log(p_i), \tag{4.6}$$

where γ is a modulation factor similar to focal-loss in [261], used here to focus more on features that have low minimum distances. We set $\gamma = 2$ in our experiments.

In order to make the estimated disparity spatially smooth while preserving the edges, the common edge-aware Disparity Smoothness Loss L_{smooth} from [262] is used. Finally, the total loss is $L_{\text{total}} = L_{\text{photo}} + \beta_1 L_{\text{reg}} + \beta_2 L_{\text{smooth}}$, where β_1 is set to 1.0, and β_2 is set to 0.1.

4.4 Experiments

4.4.1 Training Details

The entire framework is implemented in PyTorch and trained on a single Nvidia A10 GPU using the Robotcar dataset. Please find the details about the data splits (train/val/test) in Chapter 2.3. We used a batch size of 8 and the Adam [263] optimizer with a constant learning rate of 0.0001. We trained for 20 epochs with an input image resolution of 192×320.

4.4.2 Baselines

To the best of our knowledge, there is no self-supervised system that estimates depth from nighttime stereo images. We, therefore, compare our method with a classical method, Semi-Global Matching (SGM) [109], and three state-of-the-art supervised methods: Sharma et al. [249], UniMatch-Stereo [128], and IGEV-Stereo [39].

Sharma et al. [249] is the only available stereo method for nighttime data. This method was originally trained using Synthia [264], fine-tuned first on KITTI-Stereo [5], and then trained on the RobotCar dataset. On the RobotCar dataset, they use GAN-based image translation to convert daytime images to nighttime and employ cycle consistency between the estimated disparities of the original and translated images to train their system, along with ground truth depth supervision.

UniMatch-Stereo [128] is a supervised method that uses coarse-to-fine feature matching to estimate initial disparities and employs iterative refinement to further refine them. This method aims to build a unified model to solve multiple tasks: Optical Flow, Stereo Matching, and Multi-View Depth estimation. The model was first trained to solve optical flow estimation (matching in 2D) and then fine-tuned on several publicly available datasets with per-pixel disparity ground truth [2] for stereo-matching (matching in 1D). In addition to data diversity, the model was trained with high-resolution images to improve the model's performance.

IGEV-Stereo [39] is another supervised learning-based stereo method. This method uses 3D convolutions, iterative cost volumes and global matching to estimate disparity. The model was first trained on the SceneFlow [110] dataset and further fine-tuned for autonomous driving scenarios on the KITTI [5] dataset. This method also employs iterative refinement (a large number of iterations, 16 during training, 32 during the inference) and training on high-resolution images for better disparity estimation.

Considerations before Comparing Results

The summary of key differences between our method and the baselines is described below and also shown in Table 4.1.

1. **Diversity of data:** All of our baselines, except for SGM, are trained or fine-tuned on more than one dataset, offering greater diversity and a larger baseline distance

[2]There was no specific mention of the datasets in their paper or GitHub repository. We obtained the model from https://github.com/autonomousvision/unimatch/blob/master/MODEL_ZOO.md

between the cameras (e.g., KITTI-Stereo uses 54 cm, whereas RobotCar uses 24 cm). We only trained our method on the Oxford RobotCar dataset.

2. **Low resolution:** During depth evaluation, we observed that the default our test configurations for these models, UniMatch-Stereo and IGEV-Stereo, resulted in very poor depth estimation results (e.g. see Table 4.3) for the same input resolution as ours. However, when the resolution was doubled ($2\times$ our input resolution, 192×320), for Unimatch-Stereo and $4\times$ for IGEV-Stereo, we started seeing plausible depth maps and quantitative evaluation metrics. Please note that these resolutions "do not result in any squashing' of the spatial details as the aspect ratio is maintained. Throughout the evaluation, we used a *best-result resolution* as input to these models and reported the numbers for an unfair but meaningful comparison. Similarly, Sharma et al. were trained with a 256×512 input resolution, and changing this resolution to match ours resulted in very poor depth results. Thus, we maintained their resolution to evaluate the method against ours for a meaningful comparison. Please note that higher resolutions typically result in better outcomes. However, it will be memory-demanding and computationally expensive.

3. **No refinement:** We did not use any iterative refinement modules to further improve the initial disparity. As shown in their papers, the refinement module offers significant improvements in final disparity estimation for models UniMatch-Stereo and IGEV-Stereo. We also reported the quantitative evaluation without refinement for Unimatch-Stereo and with 1 iteration for IGEV-Stereo (the decoder of this model needs at least 1 iteration to estimate depth at full resolution). These models are referred to with * in Table 4.2. Please refer to their original papers for more details.

4. **Lack of ground truth:** The majority of these models failed to estimate correct depth values for poorly or very low-illuminated areas, such as the sky. However, we could not account for this effect in the quantitative evaluation as there is no ground truth available for the sky.

5. **Self-Supervision:** All these methods are trained with one or more tasks aimed at addressing stereo matching in a completely end-to-end and supervised manner. *In contrast, our method is completely self-supervised, training only a decoder and using a frozen encoder, offering greater flexibility to train on any dataset with stereo images.*

Despite all these differences, which could make the comparison unfair for the proposed method, our method still performs significantly better than both SGM and UniMatch-Stereo*.

It also performs better on a few metrics when compared with IGEV-Stereo and Sharma et al., as shown in Table 4.2.

In our experiments, we carefully accounted for the fact that both IGEV-Stereo and UniMatch-Stereo are designed to handle a variety of input resolutions and aspect ratios due to their data augmentation strategies. Specifically, IGEV-Stereo randomly stretches images and their corresponding disparity maps by factors in the range $[2^{-0.2}, 2^{0.4}]$, while UniMatch-Stereo[3] stretches images by factors in the range $[2^{-0.4}, 2^{0.4}]$. This allows each network to learn robust representations over a wide disparity distribution and different image widths.

IGEV-Stereo is originally trained on 320×736 images, and UniMatch-Stereo on 416×1024 images. In their original papers, both methods resize the input images to their respective training resolutions during inference. However, when we resized our test images to these resolutions during evaluation, the results were very poor for both methods. We suspect that the resizing caused a "squashing" of structural details in both height and width dimensions, details for which these models were not trained. These results are noted in Table 4.3 and marked with †.

To address this, we choose the resolutions that closely respect the original aspect ratios while preserving the spatial relationships (kept the original aspect ratio (0.6) of the image the same) which are crucial to stereo matching and avoiding the squashing of image details. To further investigate the sensitivity of these models to changes in aspect ratio, we took the best-performing resolutions for each model, kept the height constant, and varied only the width. These results are shown in the bottom part of Table 4.3 and annotated with ‡. Interestingly, higher resolution did not yield better results for either model. Overall, it appears that both models are more sensitive to distortions (caused when we resized the test images to the respective model's training resolution and when we tried matching the aspect ratio by changing the width) in structural details than to changes in aspect ratio.

4.4.3 Depth Evaluation

The depth-estimation performance is evaluated with DINO-V1 small with patch size 8 unless otherwise stated qualitatively and quantitatively on both the Robotcar [221] and MS2 [222] datasets. Quantitative evaluation is carried out using the metrics explained in Section 2.3 of Chpater 2. For every test image, existing methods compute their metrics as the mean of all valid pixels up to a given depth range ($50\mathrm{m}$ in our evaluation), usually only with a sparse set

[3]This detail is mentioned in their code base at https://github.com/autonomousvision/unimatch/blob/943a99006950ab6c84ea72a72beb147fdd03c19a/dataloader/stereo/transforms.py#L296

Method	Training	Datasets	Resolution
SGM [109]	-	-	192×320
IGEV-Stereo [39]	Supervised end-to-end training with iterative disparity refinement (uses 32 iterations during inference)	KITTI and SceneFlow	768×1280
Unimatch-Stereo [128]	Supervised end-to-end multi-task learning with iterative disparity refinement (uses 3 iterations during inference)	Several Public Datasets (see the caption)	384×640
Sharmal et al. [249]	Supervised end-to-end training with day & night image translation	Synthia, KITTI, and RobotCar	256×512
Ours	Self-Supervised decoder-only training	RobotCar	192×320

Table 4.1: Summary of key differences between the baselines and the proposed method. Resolution refers to the input resolution where the method seems to generate plausible depth maps for evaluation. We used in-the-wild use model of Unimatch-Stereo that was trained on several public datasets. However, there is no mention of the specific details of what they mean by this in the paper or in their GitHub repository.

of LiDAR points. Taking the overall mean is sensible when pixels are uniformly distributed over the entire depth range; however, this is not the case for the RobotCar dataset [221] as shown in Figure 4.3. The same effect is observed for most autonomous driving datasets, including the MS2 dataset [222]. This is due to the majority of pixels in an image occupying points that are very close to the camera. Evaluating performance on this kind of data skews the evaluation, as the performance on nearby points outweighs the performance on the far-away points. To mitigate this, we propose the use of depth bins splitting the given depth range into M bins. Letting x be a metric, unweighted (U) and weighted (W) evaluation metrics for a given test image can be written as:

$$\mathrm{x}_{\text{unweighted}} = \frac{1}{Z}\sum_{i=1}^{Z}\mathrm{x}_i, \quad \mathrm{x}_{\text{weighted}} = \frac{1}{M}\sum_{N=1}^{M}\frac{1}{N_i}\sum_{j=1}^{N_i}\mathrm{x}_{ij}, \tag{4.7}$$

where Z is the total number of valid pixels, M is the total number of bins, and N_i is the total number of valid pixels in the i-th depth-bin. For ease of comparison in the future, we report

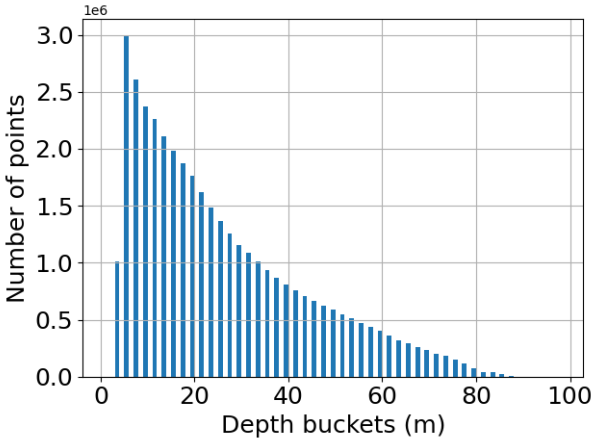


Figure 4.3: The visualization of the ground truth distribution of the test-split of the Oxford RobotCar dataset.

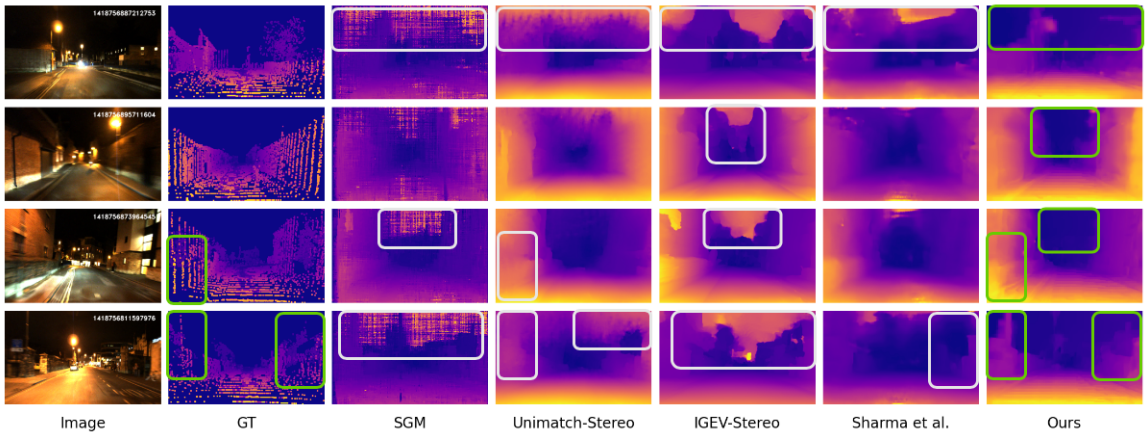


Figure 4.4: A qualitative comparison of the proposed method with SGM [109] and state-of-the-art supervised methods, Unimatch-Stereo [128], IGEV-Stereo [39], and Sharma et al. [249]. In the visualizations, brighter pixels indicate closer proximity to the camera. Please refer to the white bounding boxes to observe the issues of inaccurate depth estimation in the sky and missing structural details in depth maps estimated by the competing methods. In contrast, our method is able to estimate plausible depth maps for poorly illuminated regions, such as the sky, and successfully preserves structural details, as highlighted in the green boxes. While our method accurately estimates the depth of the sky, this improvement is not reflected in Table 4.2 due to the absence of ground truth for evaluation. Additionally, note that the results of IGEV-Stereo and Unimatch-Stereo include the refinement module and are at $2\times$ the resolution compared to ours.

the numbers from both evaluations. Please note that we did not use eigen or garg crops [28] during our evaluation and set $M = 10$, i.e., each bin covers $5m$ depth.

Metric	*Method*	type	*Abs. Rel.* (↓)	*Sq. Rel.* (↓)	*RMSE* (↓)	*Log RMSE* (↓)	$\delta < 1.25$(↑)	$\delta < 1.25^2$(↑)	$\delta < 1.25^3$(↑)
U	UniMatch-Stereo [128]	Sup	0.131	1.380	6.527	**0.262**	0.792	0.911	0.951
	IGEV-Stereo [39]	Sup	**0.123**	**1.341**	**6.300**	0.277	**0.843**	0.907	0.942
	UniMatch-Stereo* [128]	Sup	0.141	1.433	6.656	0.270	0.768	0.905	0.952
	IGEV-Stereo* [39]	Sup	0.127	1.393	6.440	0.288	0.825	0.904	0.940
	SGM [109]	-	0.224	3.164	7.879	0.340	0.735	0.876	0.929
	Sharma et al. [249]	Sup	0.247	1.737	6.430	0.285	0.541	**0.937**	**0.967**
	Ours	Self-Sup	0.162	1.927	6.612	0.266	0.809	0.915	0.951
W	Unimatch-Stereo [128]	Sup	0.171	2.250	6.699	0.275	0.693	0.864	0.924
	IGEV-Stereo [39]	Sup	**0.153**	2.203	**6.504**	0.286	**0.777**	0.861	0.907
	Unimatch-Stereo* [128]	Sup	0.187	2.331	6.906	0.291	0.652	0.849	0.925
	IGEV-Stereo* [39]	Sup	0.157	2.261	6.642	0.298	0.762	0.857	0.905
	SGM [109]	-	0.232	3.388	8.766	0.352	0.685	0.844	0.907
	Sharma et al. [249]	Sup	0.235	**2.051**	6.603	0.285	0.578	**0.912**	**0.949**
	Ours	Self-Sup	0.179	2.441	6.821	**0.273**	0.767	0.889	0.932

Table 4.2: Quantitative evaluation of our proposed method against our baselines on Oxford RobotCar dataset. This evaluation is carried out with a depth range of 50 meters. U refers to the unweighted metric, and W to the proposed weighted metric. In the type column, “sup" refers to supervised training, and “self-sup” refers to self-supervised training. * refers to no refinement for Unimatch-Stereo and one refinement iteration for IGEV-Stereo. **Bold** shows the best performance and underline refers to second best results. As one can see, the proposed self-supervised method performs better in all metrics compared to Unimatch-Stereo and in some metrics compared to IGEV-Stereo and, especially Sharma et al., a supervised method trained on the Oxford Robotcar dataset. Also, please note that our performance in challenging regions, such as the sky, is not accounted for during quantitative evaluation due to the absence of ground truth.

RobotCar: We present a qualitative comparison in Figure 4.4. Our method is capable of extracting crisp details, even in poorly illuminated regions. The effect of masking is evident when observing the disparity estimated for the sky pixels. As shown in Figure 4.4, all other methods estimate inaccurate depth (brighter pixels) for the sky, where they are supposed to be darker, as indicated by the ground truth. Additionally, we visualize the estimated masks and the respective filtered disparity maps in Figure 4.5. Plausible masks are generated for noisy, poorly illuminated areas such as the sky. Quantitative results are provided in Table 4.2. Again, it is important to note that our outstanding performance in challenging regions, such as the sky, is not accounted for during quantitative evaluation due to the absence of ground truth. Despite various differences, such as the diversity of data, being trained and tested on low-resolution images, not utilizing any refinement modules, being unable to evaluate sky regions, and most importantly, being self-supervised, our method performs on par with the baselines in the majority of metrics, and even better in a few, across both the weighted and unweighted variations. Another important thing to note is that dependency on high-resolution images and refinement modules significantly slows down the inference speeds of the baseline methods.

Variation	Resolution	H:W Ratio	*Abs. Rel.* (↓)	*Sq. Rel.* (↓)	*RMSE* (↓)	*Log RMSE* (↓)	$\delta < 1.25$ (↑)	$\delta < 1.25^2$ (↑)	$\delta < 1.25^3$ (↑)
Unimatch-Stereo (768 × 1280)	768 × 1280	0.6	0.180	2.214	8.491	0.349	0.647	0.824	0.916
UniMatch-Stereo (384 × 640)	384 × 640	0.6	**0.131**	**1.380**	**6.527**	**0.262**	**0.792**	**0.911**	**0.951**
Unimatch-Stereo (192 × 320)	192 × 320	0.6	0.370	6.145	11.820	0.399	0.508	0.759	0.898
IGEV-Stereo (768 × 1280)	768 × 1280	0.6	**0.123**	**1.341**	**6.300**	**0.277**	**0.843**	**0.907**	**0.942**
IGEV-Stereo (384 × 640)	384 × 640	0.6	0.170	2.556	9.391	0.394	0.721	0.810	0.880
IGEV-Stereo (192 × 320)	192 × 320	0.6	0.291	5.729	13.778	0.716	0.516	0.612	0.692
IGEV-Stereo† (320 × 736)	320 × 736	0.434	0.150	1.968	8.160	0.332	0.762	0.851	0.915
UniMatch-Stereo† (416 × 1024)	416 × 1024	0.406	0.167	1.956	8.051	0.313	0.681	0.850	0.939
IGEV-Stereo‡ (768 × 1766)	768 × 1766	0.434	0.124	1.436	6.520	0.292	0.827	0.896	0.934
UniMatch-Stereo‡ (384 × 960)	384 × 960	0.40	0.161	1.860	7.845	0.304	0.697	0.860	0.942

Table 4.3: Evaluation of how the input resolution affects the depth estimation performance using unweighted metrics. The new column "W:H Ratio" shows the aspect ratio calculated from the input resolution. † refers to the evaluation with their training resolution, ‡ refers to the resolution obtained using the aspect ratio these models trained on, and **bold** refers to the best results. For UniMatch-Stereo the width had to be divisible by 16, so the closest width we could get was 960 for the entry with ‡.

Method	*Abs. Rel.* (↓)	*RMSE* (↓)	$\delta < 1.25$(↑)	$\delta < 1.25^3$(↑)
Base model	0.220	7.859	0.688	0.945
w/ Mask	0.240	7.916	0.660	0.949
w/ L_{reg}	0.188	7.086	0.730	0.950

Table 4.4: An ablation study was performed to demonstrate the importance of various modules in our system using unweighted evaluation metrics on the Oxford Robotcar dataset. A key observation is the significance of the regularization loss. When only the disparity maps were masked (w/mask), the error and accuracy metrics were worse compared to the baseline model. However, the results improved significantly when regularization loss was incorporated (w/mask + Reg) into the training process.

4.4.4 Ablation Studies

We conducted several ablation studies to assess the impact of different modules, with the results presented in Table 4.4. The Base model uses DINO-V1 as the encoder, incorporating stereo matching and upsampling. The W/ mask variant refers to the model where the coarse disparity is masked using the proposed feature heuristic-based masking. Finally, W/ Mask + L_{reg} refers to the model that includes the regularization loss, L_{reg}, in addition to the masking. The key observation is that the proposed masking was ineffective when features were not encouraged to be informative. It was only through the introduction of the regularization loss that the true potential of the proposed masking was realized. Further, we trained W/ Mask + L_{reg} model with Disparity Smoothness loss before reporting our final results on the test split.

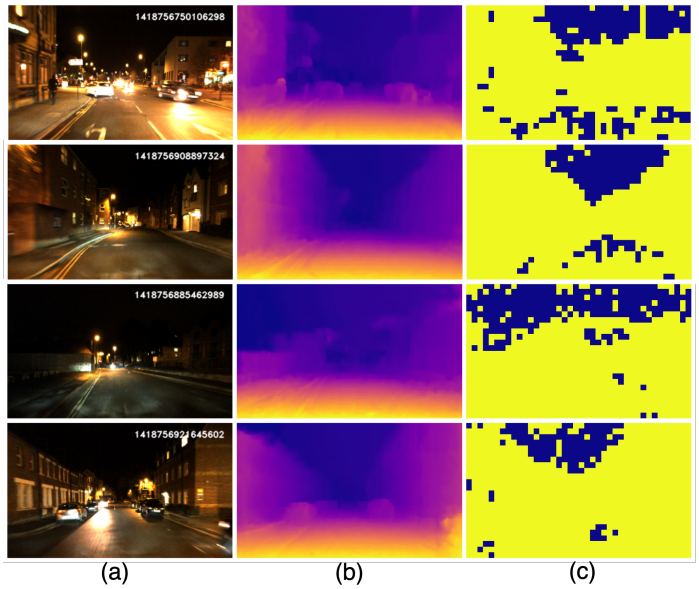


Figure 4.5: (a) Input images, (b) estimated disparity maps, and (c) the corresponding estimated masks. The disparity maps show a cleaner sky, while the masks effectively capture the noise areas in the image.

Baseline	*Abs. Rel.* (↓)	*Sq. Rel.* (↓)	*RMSE* (↓)	*Log RMSE* (↓)	$\delta < 1.25$ (↑)	$\delta < 1.25^2$ (↑)	$\delta < 1.25^3$ (↑)
WSGD (35.3cm)	0.199	1.675	6.344	0.270	0.702	0.902	0.960
Ours (23.8cm)	0.172	2.013	6.764	0.270	0.787	0.912	0.951

Table 4.5: Quantitative comparison against the monocular method proposed in Chapter 3.

4.4.5 Comparison with Monocular Methods

Although there were no stereo methods that were designed for night-time data, there are methods that were designed for a monocular setting, including our method explained in Chapter 3. To further understand the performance comparison between these methods, we compared the proposed method against our monocular method, WSGD. The quantitative comparison is shown in Table 4.5. The closest result to ours was the night-time-only model that was trained with stride-1 configuration between the images, with a baseline distance of ≈ 35.3 cm. There are a few important things to note here while comparing the results. First, WGSD was trained end to end for the task of depth estimation. Second, The baseline distance between the cameras, when the images were captured, is approximately $1.5\times$ than the stereo baseline. Lastly, it also uses higher resolution images as input, 256×512 as an input, while the proposed method uses 192×320. It was already shown in Chapter 3 that the bigger parallax (caused by a bigger baseline distance between the cameras) could lead to better results. However, the proposed stereo method has a better $\delta < 1.25$ metric, meaning

Warping	*Abs. Rel.* (↓)	*Sq. Rel.* (↓)	*RMSE* (↓)	*Log RMSE* (↓)	$\delta < 1.25$ (↑)	$\delta < 1.25^2$ (↑)	$\delta < 1.25^3$ (↑)
3D-warp	0.201	2.047	6.958	0.268	0.702	0.894	0.958
2D-warp	0.172	2.013	6.764	0.270	0.787	0.912	0.951

Table 4.6: Depth evaluation of our proposed method with DINO-V1 model with patch size 8 trained with 2D and 3D warping types during the image reconstruction.

PatchSize	*Abs. Rel.* (↓)	*Sq. Rel.* (↓)	*RMSE* (↓)	*Log RMSE* (↓)	$\delta < 1.25$ (↑)	$\delta < 1.25^2$ (↑)	$\delta < 1.25^3$ (↑)
8×8	0.172	2.013	6.764	0.270	0.787	0.912	0.951
16×16	0.197	2.419	7.264	0.284	0.738	0.899	0.950

Table 4.7: Quantitative evaluation of our method with two different patch sizes, using features from variants of the DINO-V1 model. The results indicate that depth estimation accuracy is significantly influenced by the patch sizes used in the foundational model.

that a larger percentage of the predicted depth values are within 25% of the true values. Essentially, it reflects fewer large errors and more consistent predictions than the monocular method.

4.4.6 2D vs 3D Image Warping

We further evaluated the proposed method with two different types of warping functions—
2D image warping [180]: the predicted disparities are directly used to transform the left-image pixels onto the right-image,
3D image warping [183]: the predicted disparities (for the left image) are first converted into the respective depth maps and then transformed onto the right image using the camera intrinsics and the extrinsics of the stereo rig.

The estimated results are reported in Table 4.6. The 2D warping significantly improves the results compared to 3D warping. This could be because 3D or rigid transformation-based warping requires the images to be perfectly stereo-rectified and the cameras to be perfectly calibrated for intrinsics and extrinsics. On the other hand, pixel warping operates at the image level, enabling finer and more adaptable transformations that can accommodate lens distortions, non-linearities, and depth variations within a scene.

4.4.7 Effect of Patch Size

The success of recent visual foundational models largely stems from their underlying architectures, such as the Vision Transformer (ViT) [32, 41]. These models partition the input image into patches, and the size of these patches is critical for capturing image details. Smaller patches (e.g., 8×8) enable the model to capture finer details and subtle variations

Method	*Abs. Rel.* (↓)	*Sq. Rel.* (↓)	*RMSE* (↓)	*Log RMSE* (↓)	$\delta < 1.25$ (↑)	$\delta < 1.25^2$ (↑)	$\delta < 1.25^3$ (↑)
DINO-VitS-8	0.172	2.013	6.764	0.270	0.787	0.912	0.951
DINOV2-VitS-14	0.196	2.344	7.200	0.282	0.743	0.906	0.950
AM-RADIO-VitH-16	0.217	2.516	7.384	0.293	0.703	0.895	0.950

Table 4.8: Adaptability of our proposed stereo-matching decoder to various foundational models, DINO, DINO-V2 and AM-RADIO.

in texture and edges, which are crucial for dense tasks like feature matching. A finer patch resolution leads to a higher granularity in feature extraction, thereby enhancing the matching accuracy between stereo views. In contrast, larger patches (such as 16×16) summarize larger portions of the image into a single token, potentially missing out on fine-grained details.

To better understand the influence of patch size on depth estimation performance, we trained the DINO-V1 model with both 8×8 and 16×16 patch sizes using a 2D-image warping based photometric loss. As shown in Table 4.7, the model trained with 8×8 patches outperforms the one trained with 16×16 patches across almost all metrics. These findings clearly indicate that a smaller patch size contributes significantly to the model's ability to estimate more accurate depth maps. This finding is further corroborated with the experiments performed with other recent foundational models, DINO-V2 and AM-RADIO with larger patch sizes (14×14, 16×16 respectively) still under perform DINO-V1 with patch size 8×8.

4.4.8 Adaptability to Other Visual Foundational Models

The proposed architecture is not tailored for a single visual foundation model but rather can be easily plugged into any visual foundation model. To validate this claim, we performed experiments with other visual foundational models such as DINO-V2 [33] and AM-RADIO [36]. The depth evaluation results are tabulated in Table 4.8. DINOV2 model is trained with a much larger dataset compared to what DINO was trained. AM-RADIO model is a meta-foundational model that is trained with the distilled knowledge from multiple foundational models, including DINO-V2. Despite this, the performance of DINO-V2 and AM-RADIO seems to be worse than DINO-V1. We largely attribute this to the input-patch sizes of these models. As explained in Section 4.4.7, the proposed method seems to be sensitive to the resolution of the features. Possible solutions towards this include using feature upsamplers [265] or training and fine-level feature extractors alongside [173], which we left for future work.

Condition	*Abs. Rel.* (↓)	*Sq. Rel.* (↓)	*RMSE* (↓)	*Log RMSE* (↓)	$\delta < 1.25$ (↑)	$\delta < 1.25^2$ (↑)	$\delta < 1.25^3$ (↑)
Night-only	0.172	2.013	6.764	0.270	0.787	0.912	0.951
Day & Night	0.166	1.986	6.641	0.266	0.803	0.915	0.952

Table 4.9: Evaluation of the robustness of the approach when trained and tested under extreme lighting conditions, such as day and night. The performance improvement demonstrates that the model effectively leverages diverse training data, becoming more robust in handling varying lighting conditions.

4.4.9 Training with Day and Night Images

We further trained the model, including images collected during the day, to assess its adaptability to extreme lighting variations. We used the daytime training splits of the Oxford RobotCar dataset that share the same geographic overlap as nighttime splits. This model resulted in a significant performance improvement in depth evaluation, as shown in Table 4.9, across almost all metrics. This outcome is expected for several reasons: First, the encoder is inherently effective at extracting meaningful features from daytime images due to its pretraining on datasets rich in well-lit scenes. Additionally, the photometric loss calculated using daytime images is more accurate than that from nighttime images, providing a better learning signal for our decoder. Finally, the diversity in lighting conditions likely acts as a natural regularizer, preventing the decoder from overfitting to the specific noise and artefacts associated with nighttime images.

4.4.10 Generalization to Other Nighttime Datasets

To further evaluate the generalisability of our method, we evaluated it using the test split of the MS2 dataset [222]. We used the model trained on the RobotCar dataset for our experiments. Despite differences in lighting and infrastructure, the model is able to estimate very plausible disparity maps and pixel masks as shown in Figure 4.6 and comparable quantitative performance as shown in Table 4.10. One interesting observation here is that Unimatch-Stereo is the best-performing model in both weighted and unweighted metrics on the MS2 dataset. The same model seemed to struggle on the Oxford Robotcar dataset. This questions the ability of the supervised models to generalize to novel scenes even after being trained on massive amounts of data. However, our model performs better than our closest baseline, Sharma et al. [249], which was trained on the RobotCar Dataset in a supervised manner. This shows a potential for using foundational models to be used as encoders instead of training a new encoder for every scene of interest in self-supervised stereo-matching.

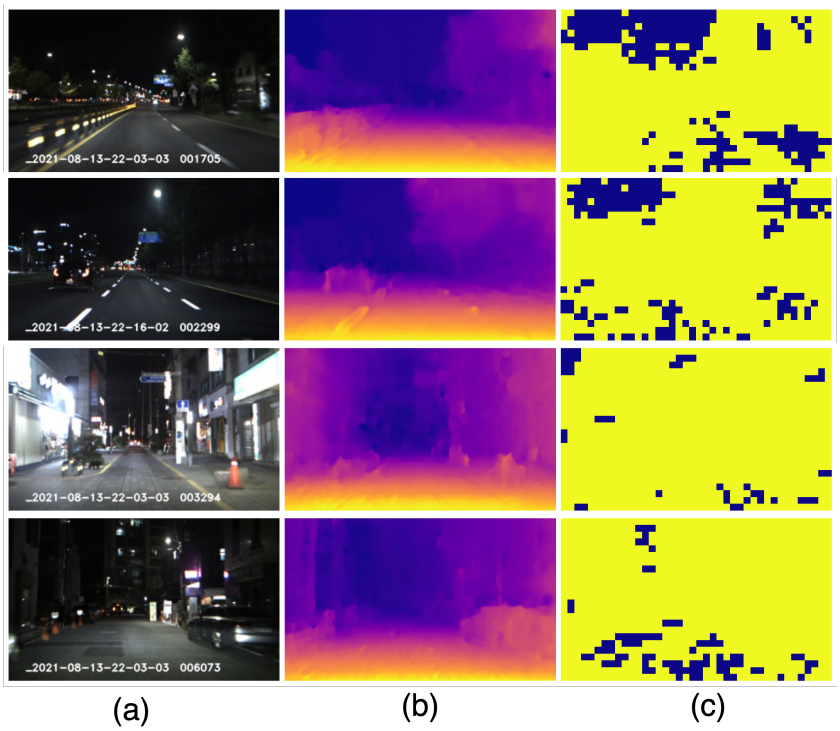


Figure 4.6: Visualization of the estimated disparities in (b) and the masks in (c) for the given input images (a) when tested on the unseen MS2 dataset. For visualization purposes, only the left input is shown. Despite the images being captured in a completely different geographic location (Seoul, Korea), with an entirely different infrastructure compared to Oxford, our method still produces accurate depth estimations.

Metric	*Method*	*Abs. Rel.*	*RMSE*	$\delta < 1.25$	$\delta < 1.25^3$
U	SGM [109]	0.185	6.106	0.773	0.968
	Unimatch-Stereo [128]	0.095	3.396	0.910	0.994
	IGEV-Stereo [39]	0.099	3.918	0.894	0.985
	Sharma et al. [249]	0.193	5.077	0.713	0.990
	Ours	0.151	5.091	0.767	0.989
W	SGM [109]	0.183	6.482	0.764	0.969
	Unimatch-Stereo [128]	0.112	4.182	0.860	0.991
	IGEV-Stereo [39]	0.125	4.977	0.815	0.976
	Sharma et al. [249]	0.180	5.489	0.733	0.987
	Ours	0.172	5.350	0.716	0.982

Table 4.10: Quantitative evaluation of our proposed method (trained on RobotCar) against state-of-the-art methods on the MS2 dataset. "U" refers to the unweighted metric, and "W" refers to the proposed weighted metric. The key takeaway is that our method outperforms the closest supervised baseline (trained on RobotCar), which is trained end-to-end, demonstrating the inherent generalization capabilities of our approach, aided by the use of visual foundational models.

4.5 Limitations and Future Work

The resolution of the input features maps is observed to affect the depth estimation performance greatly. Due to this, it was not possible to benefit from the more recent models, such as DINO-V2 and AM-RADIO, that were trained on orders of magnitude bigger datasets than DINO-V1. There are methods recently, such as FeatUp [265], that use trainable upsamplers to increase the resolution of the foundational features. Employing such a model before feeding it to the stereo-matcher would greatly improve the accuracy of depth estimation. Also, the proposed method seems to struggle with the haze and the glare of the streetlights, and it considers them objects. We leave these works for the future.

4.6 Conclusions

In this chapter, we introduce a self-supervised method for depth estimation from stereo image pairs. The model leverages features extracted from large, pretrained visual foundational models to inherit their robustness and generalisability in challenging conditions and novel scenarios. Depth estimation in this model is a byproduct of stereo-matching these extracted features. We experimented with three foundational models: DINO-V1, DINO-V2, and AM-RADIO, each with different patch sizes, using them as encoders. A learnable decoder, inspired by the Unimatch model, was designed for the task. The model was trained using a standard spatial photometric loss.

One significant challenge was the degradation of depth estimation in noisy regions of the images, such as the sky. This issue arises from poor feature learning when the photometric error is computed on noisy image areas. To address this, we propose a method that estimates a mask from the learned features to exclude these noisy disparity estimates before applying the photometric loss. However, this initially led to the masking of too many regions. To mitigate this, we introduced a regularization loss, inspired by focal loss, which encourages the features to avoid excessive masking. Ablation studies demonstrated the effectiveness of this mask when combined with the proposed regularization loss.

Another interesting observation emerged when using larger patch sizes, such as 16 for DINO-V1 and AM-RADIO and 14 for DINO-V2. The results suggest that features extracted from DINO with a patch size of 8 produced significantly better depth estimation accuracy compared to those from DINO-V2 or AM-RADIO. This is surprising because DINO-V2 was trained on a vastly larger dataset, and AM-RADIO is an ensemble of foundational models, including DINO-V2. Despite this, patch size played a more critical role in performance.

Finally, we evaluated our model on a completely different dataset, captured in a geographically distinct location with varying infrastructure and lighting conditions, to test its generalization ability. Remarkably, our model outperformed a stereo model trained on the same dataset, further underscoring the effectiveness of the strong learned priors in these foundational models. This suggests that end-to-end learning might not be necessary when leveraging such powerful foundational models.

In this chapter, we explored the potential of foundational features for the task of dense stereo matching, focusing on matching features across left and right images captured with a very small baseline distance. In the next chapter, we will further evaluate the full potential of these features across varying baseline distances for the task of correspondence estimation—a critical step in estimating camera motion.

Chapter 5

Min-Max Detector: Digging into Descriptions for Keypoints

While the preceding chapters discussed how depth estimation, whether monocular or stereo, provides valuable spatial information, it alone does not guarantee robust 3D understanding if the motion of the camera or agent itself is unknown. Ego-motion estimation specifically addresses this need by quantifying how the viewpoint changes over time, enabling a system to differentiate between true scene dynamics and mere shifts in perspective. Through repeated observations from varying view-points, previously occluded surfaces become visible, aiding in more accurate scene reconstruction and revealing object details that a single view might miss. By tracking self-motion, perception pipelines can refine their understanding of occlusion boundaries and elevate depth estimates from static scenes to dynamic, temporally coherent representations. Hence, estimating camera motion is another important task for computer vision and autonomy. Typically this is undertaken by estimating the motion transformations between temporally distinct views. We note that keypoint-based approaches often offer greater accuracy and generalization than CNN-based methods. The primary goal of keypoint detection and description is to locate points in an image that are *salient, stable*, and *repeatable*, allowing for accurate matching across image pairs to triangulate the camera-motion information.

In this chapter, we propose a novel learning-free approach for detecting keypoints by utilizing the spatial distribution of features on a D-dimensional unit sphere as a measure of saliency. We further argue that modern visual foundation models are designed to ensure saliency, stability, and repeatability through their task-agnostic feature maps, and can generate high-quality keypoints and their descriptors without requiring additional training. Our method requires no fine-tuning and can be adapted to work with any image encoder. We conduct extensive ablation studies to evaluate the performance of our method under a variety of challenging conditions, such as changes in illumination and viewpoint. The evaluations

span multiple datasets, including HPatches, Oxford RobotCar, and ScanNet. Remarkably, we found that the foundational features perform on par with the trained detectors. Furthermore, when the proposed detector is combined with feature maps generated by a pretrained detector, DISK, it surpasses the performance of the original model. These results highlight the effectiveness of our method and support the exploration of no-learning-based approaches for joint keypoint detection and description.

5.1 Introduction

Detecting keypoints in an image is of paramount importance for many computer vision applications, such as SLAM [266] and Structure from Motion [174]. Traditionally, keypoints are detected using intuitive heuristics based on intensity variations [72, 10, 11] and further described using the pixels around the detected keypoint using techniques such as Histogram of Gradients (HOG) [10] and binary tests [73], for correspondence estimation. With the success of deep learning, many methods such as [156, 160, 162, 151, 40, 143, 42, 151, 161], have tried to address this problem in a data-driven way using neural networks. However, in contrast to the traditional methods that perform detection first and description later, these deep learning methods started questioning this order. Broadly, these methods can be categorized into two types: detect-then-describe [143, 156, 160] and describe-then-detect [151, 161, 42, 162, 166]. The first group are very early deep learning methods. They detect the keypoints first and then for each keypoint a patch around the keypoint is extracted to describe it. These methods are similar to traditional methods [10, 12] but with learnable modules. The second category outperformed the former in recent years by arguing that keypoints do not necessarily lead to discriminative features - matching on these could result in inaccurate correspondence estimation [161, 151]. Several approaches were proposed to jointly learn to describe and detect while prioritising the discriminativeness of the descriptions. All these methods are end-to-end trained using ground truth correspondences estimated using either 2D homographic projections [143] or depth and motion cues [160, 174]. Despite significant research effort, the generalization of these supervised methods is poor outside their training domain [267]. There were also efforts made to weakly supervise using landmarks [268] camera-poses [269, 42, 175] but faced a significant drop in their performance[1].

Orthogonal to learning approaches, there are also some efforts made by methods such as [177] that use pretrained networks and extract keypoints and descriptors from them without any further training. As a seminal work in this area, D2D [177] proposes to revisit

[1]For example, the DISK method has a variant where they trained using weak supervision.

the idea of using Shannon entropy of features to estimate the saliency [270, 271] that can then be used as a metric to detect the keypoints from the features. However, estimating entropy is computationally expensive. The authors thus assumed image features to be drawn from an unimodal Gaussian distribution, and the features with large standard deviations are considered informative. If this informative feature is locally salient in the image space, then the respective pixel location is considered as *salient* to be chosen as keypoint, and that feature as its descriptor. In this work, we focus on this learning-free approach, although we contest two assumptions: that informativeness alone will lead to better matching, and that an informative feature is not confusing to match. The case that violates these assumptions is when an informative feature has a very close nearest neighbour which is also information rich. To be clear, here, we are referring to the nearest neighbours in the latent space rather than in the image space. To improve the matching process, we require features that are just not informative but are discriminative to make it easy to differentiate them in the latent space from other informative features. Inspired by the other proxy measures of entropy using nearest neighbour distances [272], we propose a new criterion based on the local density of a feature in the latent space to be selected as a unique feature. This principle can be understood from the 2-dimensional toy example shown in Figure 5.1. Given a set of n samples, we aim to select m samples, where $m << n$. D2D [177] which makes a unimodal Gaussian assumption about the data and utilizes the standard deviation as a measure to select keypoints resulted in clusters of features. However, matching features that are in clusters accurately is hard. However, the proposed method is able to select points that are more spatially distinct (low local density) which can benefit the correspondence estimation later.

Another problem we identified is that the encoders of learning-based methods, or descriptor networks [158, 159] that D2D was tested on are all trained on relatively small and task-specific datasets. However, the recent success of visual foundation models [33, 34, 32] that are trained on very large scale datasets begs the question: for every specific downstream task, do we still need to learn to encode images anymore? As foundation models shrink and computing gets faster and more efficient, such powerful and generalisable feature extractors will likely be suitable for many applications. This work is one of the first to evaluate the extent to which foundation models can solve this fundamental problem of keypoint detection and description. We also point out some of the limitations and make suggestions for future research considering training the next iteration of these models.
Our main contributions are therefore as follows:

1. We propose and advocate for a new measure of uniqueness for a feature based on its distance from the nearest neighbor in the feature space.

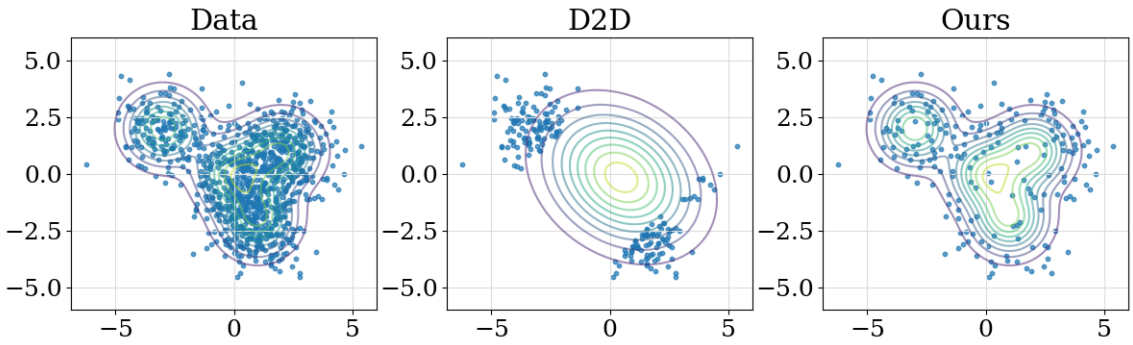


Figure 5.1: Visualisation depicting the difference in selecting using the existing standard deviation-based method D2D [177] and the proposed min-max metric on a 2D toy example. Here, Data refers to the sampled 2D descriptors. Contours (yellow is low and purple is higher) show the standard deviation of values across feature dimensions. When D2D is used, it assumes the data to be unimodal and selects points that are of the highest standard deviation as shown in the middle. As can be seen, this resulted in very tightly clustered points whereas the proposed method is able to sample points that are more distinct from each other.

2. We introduce a novel learning-free keypoint detector that can be applied to any feature extraction model to extract high-quality keypoints.

3. We perform extensive ablation and experimental studies on various datasets, tasks, and configurations to understand the benefits and limitations of the proposed detector and the visual foundational features.

The rest of the chapter is organized as follows, related work is discussed in Section 5.2 followed by the proposed method and various experiments and ablation studies in Section 5.3 and Section 5.4 respectively. Lastly, the limitations with some promising directions for future work are discussed in Section 5.5 followed by conclusions in Section 5.6.

5.2 Related Work

In the following sections, a brief overview of the related methods in the direction of keypoint detection and description using deep learning is explained. However, the majority of this section is repeated from Chapter 2 here for better readability and the sake of completion.

5.2.1 Keypoint Detection and Description

Traditionally, keypoints are detected first, and patches around them are described using handcrafted features. The rise of deep learning prompted questions: 1. Should detection

and description follow traditional methods? 2. Should descriptions be learned first, then keypoints detected? These further led to inquiries about joint or decoupled learning of detection and description, how to supervise them, and whether detection should be learned at all.

TILDE [152] introduced the first CNN-based keypoint detector, improving robustness under challenging conditions by regressing score maps for keypoint detection. Lenc et al. [153] proposed a covariance constraint for assigning keypoints to image structures, while Zhang et al. [154] introduced the concept of standard patch and canonical feature. Quad-Networks [155] took an unsupervised approach using patch ranking for keypoint detection. MagicPoint [143], the base of Superpoint [143], trained a fully convolutional network on synthetic corner data to achieve real-time performance. Key.Net [157] combined handcrafted filters with learnable CNNs to improve repeatability.

While earlier methods focused solely on detecting keypoints, others emphasized learning local descriptors. HardNet [158] used a metric learning loss inspired by Lowe's criterion [10] to train CNN-based descriptors, while SOSNet [159] improved upon HardNet by introducing Second Order Similarity (SOS). Methods such as LIFT [156] combined detection, orientation estimation, and description into one framework, while LFNet [160] leveraged depth and motion cues for detection and description.

Describe-then-detect methods gained prominence with Superpoint [143], which extended MagicPoint with an end-to-end descriptor network. D2-Net [151] moved away from traditional score maps, learning descriptors that determine keypoints through local maxima in spatial and channel dimensions. R2D2 [161] argued that saliency-based score maps harm descriptor learning and instead encouraged descriptors in areas of high confidence for matching. DISK [42] introduced a probabilistic framework using Reinforcement Learning (RL) to address the challenges in differentiability, while ASLFeat [162] utilized deformable convolutions to model geometric details. ALIKE [164] and ALIKED [165] proposed lightweight frameworks with differentiable keypoint detection modules. SILK [166] recently introduced a fully differentiable, lightweight, and flexible framework for keypoint detection.

Most methods rely on ground truth correspondences generated using either homographic adaptation or traditional Structure from Motion (SfM) techniques like COLMAP [174]. However, PoSFeat [175] proposed a weakly supervised learning approach based solely on camera motion and suggested a decoupling detector and descriptor training. DeDoDe [40] also proposed a decoupled training method, outperforming transformer-based methods like SuperGlue [150], though it still suffers from repeatability issues in certain settings. Despite progress, many supervised learning methods still struggle to generalize to new domains [267]. Some methods advocate for learning-free approaches. ELF [176] used

pretrained encoders and backpropagate from features to the image for detecting keypoints but at a high computational cost. D2D [177] utilized pretrained local feature descriptors like HardNet [158] and SOSNet [159], with saliency based on proxy measure for entropy to detect keypoints. However, R2D2 [161] already emphasized that saliency in the image space alone isn't sufficient for good keypoint matching. Our approach in this chapter further explores this idea by redefining keypoints to ensure their corresponding descriptors are discriminative enough for accurate matching.

5.2.2 Visual Foundation Models

Visual foundation models can be characterised by large model architectures which are trained on extensive datasets. They can be task-agnostic [33], or have a bias to a given task, for example, semantic segmentation method, SAM [219] or image caption generation method, CLIP [34]. Many foundational models make use of Vision Transformers (ViT) [32] as a backbone architecture. In general, such models are pre-trained on what can be, in some cases, hundreds of millions of images [33]. For instance, DINO-V2 [33] builds on DINO [35], adding various elements to the loss, including a KoLeo regulariser [259], which encourages a more uniform distribution across feature space. [36] distills multiple foundation models—[33], [219], and [34]—into a single model. Importantly, each of these models has been found to have gained a semantic or spatial understanding of scenes, resulting in this information being embedded within the features they output. [217] explored the use of features extracted from DINO descriptors for task of semantic matching. Motivated by these observations, many methods use these features, especially the features from DINO-V2, for tasks such as depth estimation [91, 108], dense feature matching [173], and even for keypoint detection and description in DeDoDe [40]. Importantly, models such as DINO-V2 and DINO are trained using a loss to ensure the output features are similar to their augmented features. This is similar to the losses used by learning-based keypoint detection and description methods explained in the previous section. We believe that such training and the resultant features are not only useful for descriptions but also for detections via descriptions when used with learning-free approaches.

5.2.3 Entropy in Keypoint Detection and Description Learning

From an information-theoretic perspective, the entropy of a descriptor can be understood as how many bits are required to fully describe the given keypoint. Usually, descriptors are calculated locally, tracing back to a patch around the keypoint [10, 12]. If a patch has high signal complexity or unpredictability, describing it is harder with a lower number of

bits; hence, the patch has higher entropy. This is empirically shown by using Shannon entropy [273] of local attributes such as intensity of the pixels as descriptors and suggested that this high complexity could be used as a measure of saliency [270]. In traditional keypoint description methods, the idea of entropy-motivated works like ORB [12] and BOLD [274] to design binary descriptors. However, the idea of saliency for detection from descriptions, while being very appealing, is not explored further until recently by D2D [177] that uses descriptions extracted from pretrained local descriptor models [159, 158]. This has outperformed the methods that learn to do detection at the time, showing a new direction for learning-free detectors. However, estimating entropy for a continuous D-dimensional random variable (D is the dimension of the feature extracted from the local descriptor model) is very expensive compared to the binary descriptors [12, 274]. So, they assumed that the features are from an unimodal Gaussian distribution and used standard deviation as a proxy measure of entropy similar to binary methods [12, 274]. However, there are also other proxy measures of entropy, Kozachenko–Leonenko differential entropy estimator [272] which uses nearest neighbour distances, for instance. The method does not depend on the underlying distribution of the data. In a related problem, [259] used this proxy entropy estimation method as a regularizer to learn more compact representations while effectively utilizing the descriptor space. Fortunately, the same regularizer is already used to train foundational models like DINO-V2 [33] and has shown its effectiveness in learning highly discriminative features. Inspired by all these, we also used this regularizer with a modification in Chapter 4 to learn better descriptions for stereo-matching. In this chapter, we build on this entropy measured through the nearest neighbour distances as a cue for detecting keypoints from the feature maps extracted from pretrained image encoders.

5.2.4 Comparison with Standard Corner Detection Methods

Traditional corner detectors, such as Harris [72], and FAST [74], find keypoints by identifying pixels with significant intensity variations along at least two directions. These methods typically compute a local gradient matrix (i.e., the structure tensor) or perform patch-level intensity tests, selecting locations where the response function reaches a local maximum. Although these techniques are lightweight and effective for many classic vision tasks (e.g., SLAM, object tracking), they can struggle under conditions of low contrast, noise, or repetitive patterns. Even when complemented with robust descriptors like SIFT [275] or ORB [73], the underlying reliance on local intensity differences may fail to capture broader contextual or semantic information.

In contrast, our proposed min-max detector utilizes deep feature embeddings in place of raw intensity values. In our framework, each pixel (or feature-map location) is represented

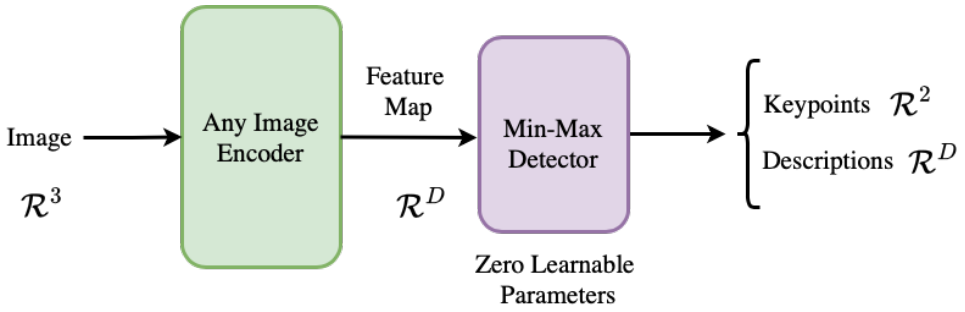


Figure 5.2: Illustration of the proposed keypoint detection pipeline. The Min-Max detector could be plugged into any image encoder to detect unique locations in the input feature map.

by a normalized feature vector in a high-dimensional space, and keypoints are selected by ranking these features based on the distance to their nearest neighbor on the unit sphere. This "feature density" measure effectively quantifies the global distinctiveness of each point features with larger nearest-neighbor distances are considered more unique or corner-like and inherently carry richer information about texture and semantics. Thus, while both traditional corner detectors and our method aim to identify salient keypoints, our approach leverages learned representations in contrast to the pixel intensity variations.

5.3 Proposed Method

Our proposed method is described in Figure 5.2. Given an image, the image encoder, which could be a visual foundational model, a pretrained detector, or a task-specific encoder outputs features at a down-scaled resolution, which are then fed into the proposed learning-free detector that estimates keypoints and their corresponding descriptors. Our method extracts the keypoint locations in a given image without any specific training, unlike existing keypoint detection methods [143, 151, 276]. In the following sections, we first describe two definitions for both detection and matching, followed by the technical details of our detector and matching modules.

5.3.1 Motivation

What is a keypoint? Certain properties, such as stability [72], repeatability [143], and saliency [177] are widely accepted measures for the quality of a point to be considered as a keypoint. Among these, stability and saliency are intra-image properties which many classical keypoint methods [11, 74] are based on. Repeatability is an inter-image property, often used in deep learning-based methods [143, 42, 166]. In this work, we explore the

principle of *local density* in the latent description space as a metric to determine what should be a keypoint. We empirically prove that this is an important property and holds great potential for designing more accurate keypoint detectors. Below, we provide some definitions and plausible explanations to aid understanding of our method.

Definition 1. *A pixel in an image is salient and can be chosen as a keypoint if the corresponding descriptor has a low local density in the latent space.*

Measuring the saliency of a point can be achieved by measuring the entropy of its descriptor [271, 270]. However, it is not straightforward to measure entropy for continuous D-dimensional random variable as it requires computationally expensive probability density estimation. As the entropy gives an idea of how informative a point is, a similar definition for saliency was explored in D2D [177], with standard deviation as a proxy measure of entropy similar to [12, 274]. While this provides points that are informative, it can result in points that are very close to each other, as shown in Figure 5.1. They have a relative saliency-based metric similar to traditional methods [72] to avoid spatial clustering, however, it is in the image space and it cannot guarantee the translation of the effect into the latent space. Also, D2D [177] assumes points to be unimodal, which is not always valid as real-world data is composed of various semantic classes. Identifying these classes and then sampling from them can be computationally expensive, and may lead to points that are less informative [277, 278].

Ideally, we want descriptions to be not just salient (caused by informativeness) but to be discriminative (confusion-free) to match. Note that being discriminative leads to being salient [270, 279], but the converse, as it shown in [161], being salient does not necessarily lead to being discriminative. Thus, in this work, we propose a measure that not only targets *informativeness* but also *local density* in the latent space with a contradiction - we need the descriptions of similar points to be close to each other such that they can be matched across images easily, but also far from other points that are not matches, but could be similar in the same image. For a feature, we quantify local density in the latent local neighborhood by measuring the distance from its nearest neighbours. The features with the highest nearest neighbor distances are chosen as salient and the corresponding pixel locations as keypoints.

This has a connection to estimating the entropy of a distribution as well. It was shown by Kozachenko and Leononenko [272], that the entropy of a distribution could be estimated by using the nearest neighbour distances which further aligns our approach with the entropy-based saliency detection methods, [177, 270]. In Chapter 4, and works like [259], it is shown that the features are encouraged to have a higher nearest-neighbour distance to make them more discriminative and effective in the downstream tasks such as stereo-matching

and similarity search. These works motivated us to explore the nearest neighbour distances further into the problem of keypoint detection. Based on the given definition and the nearest neighbor distances as a measure of the entropy of a descriptor, we design the following detector.

5.3.2 Min-Max Detector

Given an image $I \in \mathbb{R}^{H \times W \times 3}$, we extract a set of $n = h \times w$ features (where h and w are the height and width of the feature map), $\{f_1, f_2, \ldots, f_n\} \in \mathbb{R}^D$, after l layers of the encoder. For each feature f_i, we compute the cosine similarity to all other features to identify the nearest neighbors while excluding self-similarity (i.e., excluding $i = j$). Once the nearest neighbors are identified, we calculate the Euclidean distance d_{ij} between the normalized feature f'_i and its nearest neighbor f'_j, where $i \neq j$ and $j \in \{1, 2, \ldots, n\}$. Formally, this process can be expressed as follows:

$$f'_i = \frac{f_i}{\|f_i\|}, \quad f'_j = \frac{f_j}{\|f_j\|}$$

$$\text{NN}(i) = \underset{j \neq i}{\text{argmax}} \left(f'_i \cdot f'_j \right), \quad j \in \{1, 2, \ldots, n\},\ j \neq i$$

$$d_{ij} = \|f'_i - f'_j\|_2, \quad j \in \text{NN}(i)$$

This distance measure provides an understanding of how a feature f_i is distributed on the D-dimensional unit sphere. The greater the distance from its nearest neighbour, the more separated the feature is in the given space. We conjecture that well-separated features are good keypoints and can result in improved matching. Therefore, we select the top-k distances and use the corresponding features as the descriptors, while their respective pixel locations are chosen as keypoints. Currently, we use the first nearest neighbours in the proposed method, however, one could use multiple nearest neighbours and the mean of their distances to select the keypoints. Lastly, the proposed detector benefits the recent advancements in nearest-neighbour search methods, especially GPU-based methods such as Faiss [280] and cuML [281] to make the computational cost very minimal.

Definition 2. *When preserved in another image, points with low local density are repeatable under non-linear transformations, hence they can be matched.*

Even though our keypoints are spatially distinct and informative, it is not guaranteed that they will be repeated in another image. Repeatability is crucial for downstream applications like keypoint matching. A similar definition was already used in Quad-networks [155] to train a keypoint detection network in a completely unsupervised way. In this work, they

attempted to learn a response function that keeps the ranking of the top and bottom quantiles the same before and after the transformations applied to the image. Further, the learning-based methods either use homographic [143, 161] or depth and motion cues [160, 166, 164] to enforce their descriptions and detections to be repeatable under transformations. Given we use no training in our approach and detect keypoints based on their feature descriptions, we rely on the training of backbone encoders providing suitable descriptions. In this chapter, we evaluate the visual foundation models CLIP [34], DINO [35], DINO-V2 [33], and AM-RADIO [36]. The majority of these are trained with similarity objectives that try to make sure the global feature representations are the same between the input image and its augmented version (usually, cropped and scaled version of the input image). Global representations are composed of local representations, hence this effect is transferred to the local patches as well. By virtue of the way in which foundational models are trained, we believe local density will be preserved between similar images.

5.3.3 Detection and Matching

Given a pair of images (I_q, I_r), we use our encoder to extract the features $(\tilde{f}_q, \tilde{f}_r)$. We upsample these features to the required resolution[2] using bilinear interpolation, resulting (f_q, f_r). The upsampled features are used as input for our min-max detector to get the top k pixel locations with the highest nearest neighbor distances. Since estimating nearest neighbors at full resolution is computationally and memory intensive, we perform this operation patch-wise (4 patches per image) to manage the computational load. These keypoints are then matched using their normalized descriptors (f'_q, f'_r) through a mutual nearest neighbor matcher. Formally it can be written as

$$\text{matches} = \left\{ (i,j) \,\middle|\, j = \underset{k}{\operatorname{argmax}} \left(f'_{q_i} \cdot f'_{r_k} \right),\ i = \underset{l}{\operatorname{argmax}} \left(f'_{q_l} \cdot f'_{r_j} \right) \right\}. \tag{5.1}$$

In this next section, we show the experimental evidence of our definitions and assumptions.

5.4 Experiments

To demonstrate the efficacy of our proposed Min-Max detector, various experimental studies are performed on three different datasets. These are HPatches [223], Oxford-RobotCar [282] and ScanNet [224]. We evaluate the model across a range of conditions that include changes in illumination, viewpoint, and scale between images. Matching is evaluated using three different metrics: Mean-Matching Accuracy on the HPatches dataset and 6DoF camera pose

[2]The upsampling resolution could be the same as the input-image resolution.

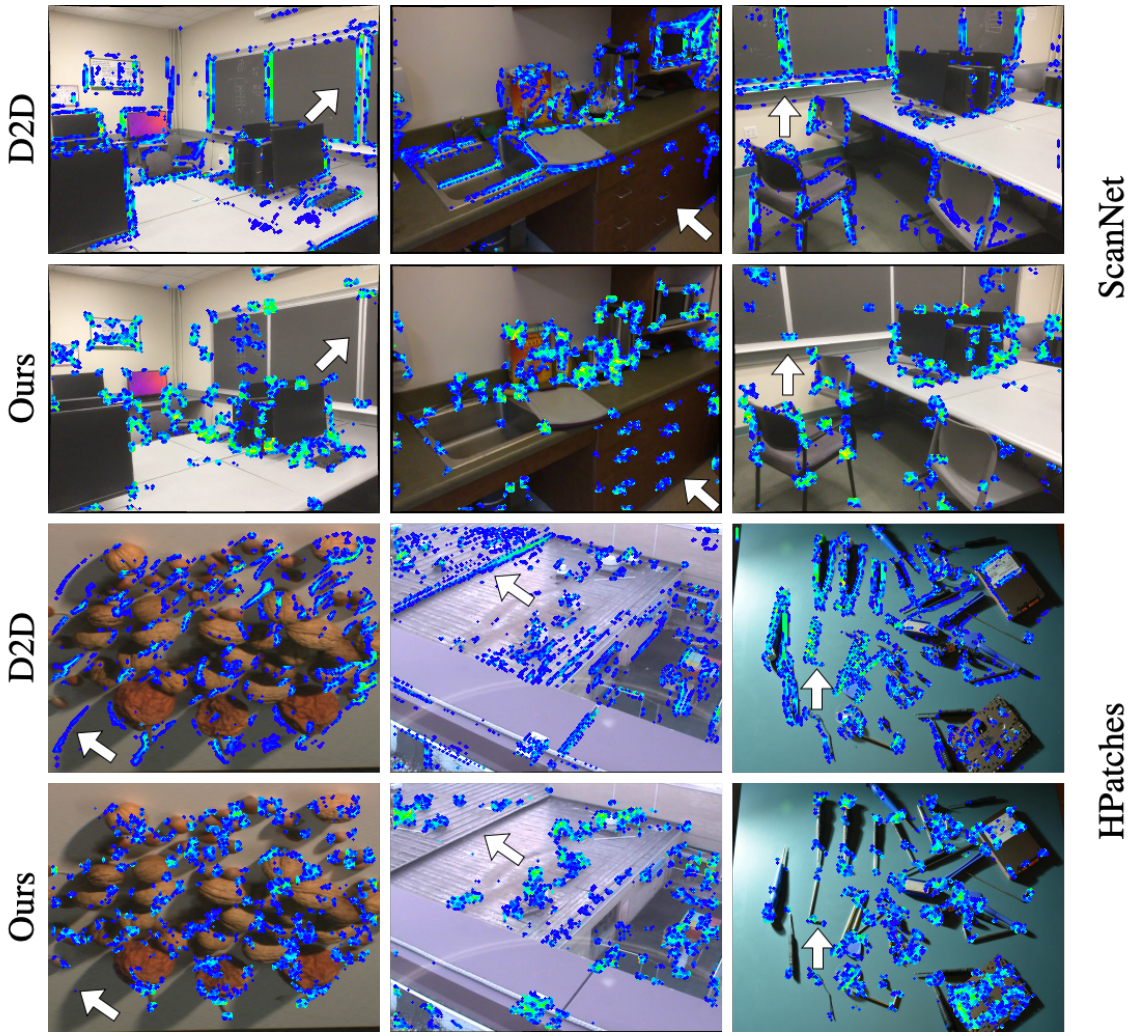


Figure 5.3: Visualisation of detected keypoints using DINO-V2 features with D2D and the proposed Min-Max Detector. Brighter colors indicate higher confidence for a keypoint and we selected the same number of keypoints for this visualisation. Please follow the white arrows to observe the differences more clearly. As seen, D2D clearly focuses on the edges present in the image with high confidence. However, edges are usually unstable for keypoint matching tasks [10]. This assumption is evident from the first image in the third row on the HPatches dataset, where D2D detects edges formed by shadows as keypoints. In contrast, the proposed detector effectively distinguishes corners (which are very reliable for keypoint matching tasks [72]) from edges and detects them with high confidence in both datasets.

estimation on the RobotCar and ScanNet datasets. Note that no training is involved in any of these datasets. For more details about these datasets and metrics, please refer to Chapter 2 and Section 2.3.

5.4.1 Methods for Comparison

During the evaluation, we compare our method with the following methods: SuperPoint [143], a trained method that learns detection and descriptions separately, D2Net [151] in which descriptions and detection are learned jointly and lastly and the keypoint are detection using local-maxima, and lastly, D2D [177] [3], a learning-free approach based on the standard deviation of the local descriptors. We also perform experiments with various foundation models as our feature-extractors: DINO [35] with ViT-Small architecture and patch size 16×16(indicated as DINO) and 8×8 (indicated as DINO†), DINO-V2 [33] with ViT-Small architecture and patch size 14×14, CLIP [34] with VIT-Large with patch size 14×14, and AM-RADIO [36] with VIT-Huge architecture with patch size 16×16. The reported results are generated using the best-performing features from ViT layer-2 for DINO-V2, AM-RADIO and CLIP and layer-5 for DINO-V1 (patch-size 8 and 16).

5.4.2 Qualitative Results

Before moving to a quantitative evaluation, in Figure 5.3 we show the keypoints detected on images sampled from HPatches and ScanNet using our method, compared with our baseline method, D2D. Brighter colors represent higher confidence levels for each keypoint. We selected an equal number of keypoints for this visualization. Follow the white arrows to clearly observe the differences. As demonstrated, D2D predominantly identifies edges in the image with high confidence. This is expected as these edge features are information-rich and potentially relatively salient in the local neighbourhood. However, edges are generally unreliable for keypoint matching tasks [10]. This is particularly noticeable in the first image of the third row on the HPatches dataset, where D2D detects edges formed by shadows as keypoints. In contrast, the proposed detector accurately distinguishes corners (which are highly reliable for keypoint matching tasks [72]) from edges and detects them with high confidence across both datasets.

There are several factors that could explain why our feature density method yields results similar to traditional corner detection techniques. First, deep feature embeddings, even though learned from extensive data, tend to preserve local image structures such as intensity

[3] D2D was implemented for image patch-based descriptor methods and did not make their code publicly available. The details of implementing their relative saliency were unclear. So, the reported results in our experiments are with our implementation of their absolute saliency using the details from their paper.

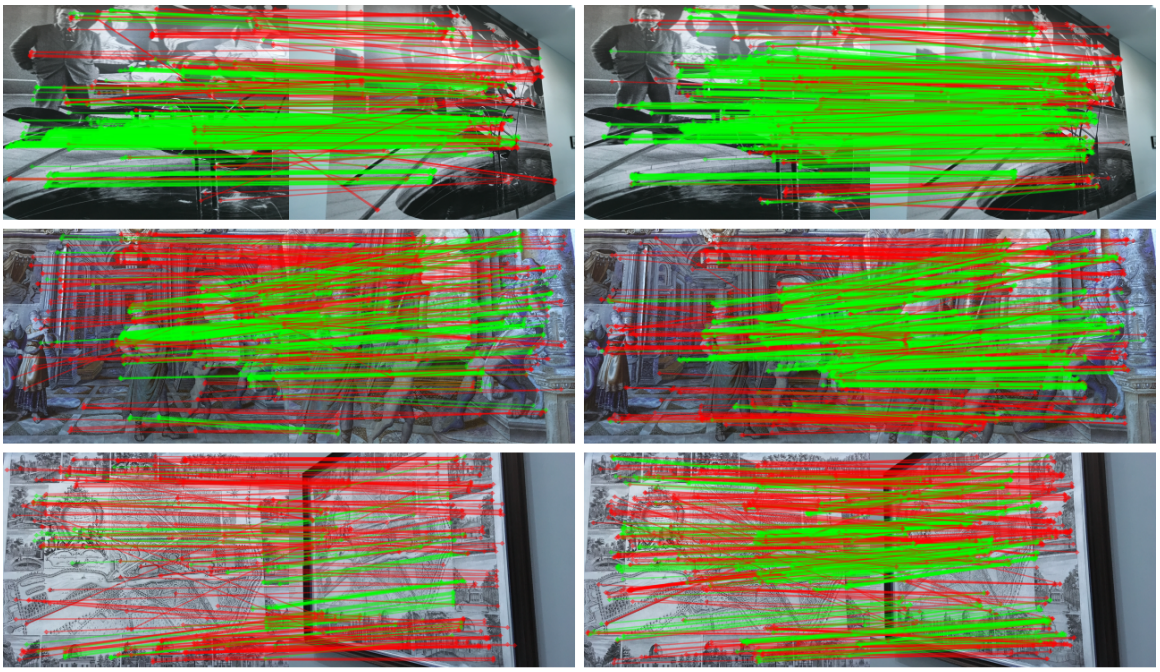


Figure 5.4: Qualitative comparison of the estimated matches on the Hpatches dataset. The good matches are shown in green and the outliers are shown in red. Given the same feature map for detection and description, the proposed detector is able to find more repeatable keypoints than the competing method, D2D, as can be seen from the density of green matches.

variations and edge details, ensuring that regions with strong gradients are still prominently represented. Additionally, both methods share a common goal of identifying regions with high distinctiveness: traditional detectors achieve this by analyzing local gradient responses, while our approach measures the distance to the nearest neighbor in the feature space. Furthermore, the deep network used for feature extraction is often implicitly biased to highlight salient image regions, many of which coincide with classical corners, because such features are critical for various computer vision tasks.

We also show some qualitative matching results in Figure 5.4. Given the same feature map and the same number keypoints selected, the proposed method is able to find more repeatable keypoints, as shown by the density of green matches compared to the competing method, D2D [177]. These matches are further used and evaluated in downstream tasks such as homogrpahy and camera-pose estimation as explained in the next sections.

Model	Illumination		Viewpoint		Overall	
	5px	10px	5px	10px	5px	10px
SuperPoint	0.721	0.840	0.493	0.705	0.602	0.770
D2Net	0.658	0.804	0.458	0.622	0.555	0.710
DINO + D2D	0.877	0.944	0.276	0.558	0.585	0.744
DINO + Ours	0.928	0.967	0.309	0.601	0.607	0.777
DINO†+Ours	0.924	0.979	0.270	0.507	0.585	0.734
CLIP + D2D	0.673	0.694	0.195	0.331	0.425	0.506
CLIP + Ours	0.950	0.966	0.142	0.384	0.531	0.664
AM-RADIO + D2D	0.888	0.900	0.305	0.526	0.585	0.706
AM-RADIO + Ours	**0.981**	**0.984**	0.380	0.649	0.669	0.810
DINO-V2 + D2D	0.887	0.923	0.407	0.612	0.638	0.761
DINO-V2 + ours	0.963	0.980	**0.507**	**0.742**	**0.727**	**0.856**

Table 5.1: Mean Matching Accuracy (MMA) at pixel thresholds of 5 and 10 pixels for various methods with different configurations tested on the HPatches dataset. The goal is to evaluate keypoint repeatability across a wide range of illumination and viewpoint changes. Each matched keypoint is assessed using the ground truth homography between the image pairs.

5.4.3 Homography Estimation

For quantitative evaluation, we follow the experimental evaluation of [151] and report the Mean Matching accuracy (MMA) of our method. MMA measures the accuracy of correspondence estimation with varying pixel thresholds from 1 to 10 pixels. In Table 5.1, we show the performance of different methods and configurations at two thresholds, 5px and 10px. We show that our method outperforms all of the baseline methods and also performs significantly better than the learning-based methods in both the illumination and viewpoint cases. This superior performance can be attributed to the combined ability of foundational features and the proposed detector to find repeatable, robust features that are suitable for matching. Interestingly, in illumination experiments, the proposed method outperforms the competing methods with a significant margin at the 5px threshold and reaches an MMA of 96%. These results suggest that our method can find repeatable keypoints under extreme illumination changes. Overall, the proposed detector performed much better than D2D with all the foundational features. However, the performance difference between illumination and viewpoint changes is substantial. This is expected, as these foundational models were not trained with any homographic projection-based augmentations, meaning the features are not as good for viewpoint changes. We believe that if the model is trained with these transformations, it would be able to perform as well as the illumination case under viewpoint change. We leave an evaluation of fine-tuning in this scenario to future work.

5.4.4 Camera Pose Estimation

In this section, the proposed detector is tested for the task of camera pose estimation between a pair of images. We conducted experiments on two datasets namely, Oxford RobotCar and ScanNet. More details about these experiments and findings are explained as follows.

Oxford RobotCar: In the previous section, we observed that the foundational features extracted using our method are performing well even when there are extreme illumination changes. However, the illumination split of the HPatches dataset is composed of static images. To further evaluate the efficacy of the proposed method, we used the nighttime sequence "2014-12-16-18-44-24" traversal of the RobotCar dataset, with the test-split proposed in [283]. This is a challenging sequence with 3653 images that include many point light sources (both static and dynamic), image blur and noise, extremely dark areas, and moving objects. This experiment aims to check the suitability of the proposed method for visual odometry or smooth camera motion estimation use cases where the camera does not observe extreme variations in the viewpoint, but the robustness to the lighting conditions is preferred more. We follow the evaluation protocol proposed in LoFTR [168] to evaluate the Area Under the Curve (AUC) of monocular pose estimation accuracy at varying thresholds ($2°$, $5°$, and $10°$). We kept hyperparameters for the pose estimation pipeline the same for all the methods. In this experiment, we further explored the best-performing model, DINO-V2 (patch size 14×14) from Table 5.1. To investigate the impact of patch size on performance, we also evaluated using both DINO (patch size 16×16) and DINO†(patch size 8×8) models. The experimental results are shown in Table 5.2(a). Overall, Keypoints extracted using the proposed detector outperform the other methods, further corroborating the effectiveness of the proposed detector in extracting stable and repeatable keypoints under extreme illumination changes.

However, as one can see in Table 5.2(a), DINO† (with patch size 8) consistently outperforms every other method. Similar observations were made in DINO [35], where the smaller patch sizes produced better results. However, as per view-point results in Table 5.1, DINO† underperforms DINO-V1, especially compared to DINO-V2, despite the resolution benefit. A more detailed consideration of whether the recent foundation models (e.g. DINO-V2 or AM-RADIO) with smaller patch sizes are better for correspondence estimation is left to future work, however.

ScanNet: As seen in Table 5.1, all foundational models seem to perform poorly when there are viewpoint changes. To test how this performance degradation translates to pose estimation, we follow a similar experiment to the Oxford RobotCar case on the ScanNet wide-baseline data split [168, 150] and evaluate 6DoF camera poses, but with different pose estimation AUC thresholds, $(5, 10, 20)$, as this split has more variation in camera poses. The

Method	Pose estimation AUC @2°	@5°	@10°
Superpoint	12.69	43.35	68.26
D2Net	5.25	14.53	27.96
DINO† + D2D	11.17	38.16	61.63
DINO + D2D	8.85	33.5	58.13
DINO-V2 + D2D	12.13	41.57	65.63
DINO† + Ours	**15.51**	**45.98**	**68.58**
DINO + Ours	10.69	37.95	62.15
DINO-V2 + Ours	10.91	38.15	62.32

(a)

Method	Pose estimation AUC @5°	@10°	@20°
Superpoint	9.43	21.53	36.40
DINO†+D2D	1.52	5.61	14.05
DINO + D2D	0.77	3.18	8.99
DINO-V2+D2D	1.81	7.64	17.41
DINO†+Ours	2.06	8.55	19.66
DINO + Ours	1.18	5.28	14.82
DINO-V2+Ours	**11.23**	**24.44**	**39.90**

(b)

Table 5.2: Camera pose evaluation results on the (a) Oxford RobotCar dataset and (b) ScanNet dataset. The goal is to assess how the overlap between image pairs impacts the performance of the proposed detector. The Oxford RobotCar split can be considered a narrow-baseline scenario where image overlap is substantial. In contrast, the ScanNet test split represents a wide-baseline scenario with minimal overlap between image pairs. This difference in overlap explains the significant disparity in the 5° and 10° metrics observed between the two datasets.

Model	Illumination 5px	10px	Viewpoint 5px	10px	Overall 5px	10px
DISK	0.940	**0.963**	0.806	**0.842**	0.870	**0.900**
DISK + D2D	0.913	0.935	0.770	0.806	0.839	0.868
DISK + Ours	**0.948**	0.960	**0.826**	0.838	**0.885**	0.897

Table 5.3: Mean matching accuracy(MMA) computed with DISK features and compared against the original DISK implementation.

quantitative results are shown in Table 5.2(b). There are a few interesting findings here. The proposed detector is able to outperform the competing methods. Specifically, the full potential of DINO-V2 features was only realized with our proposed detector. Interestingly, our detector with DINO†, using a patch size of 8, which performed better on RobotCar pose estimation—failed significantly compared to DINO-V2 features on this task. However, the superior performance of DINO-V2 in terms of MMA on HPatches and in wide-baseline matching on ScanNet provides a deeper insight into the effectiveness of the learned prior of DINO-V2 features for future work in this direction.

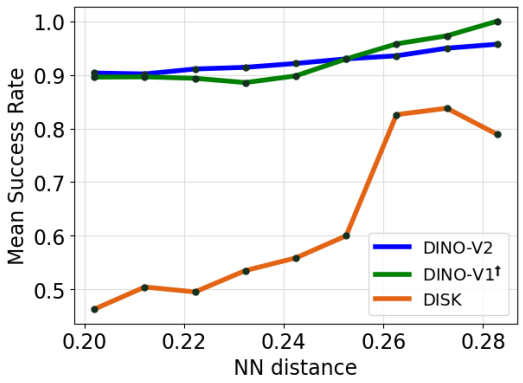


Figure 5.5: Visualization of the success rate of matching against the nearest-neighbour distances for the best-performing features, DINO-V2 in blue, DINO† in green and DISK in orange. As one can see, the success rate of matching increases as the nearest neighbour distance of a feature increases.

5.4.5 Generalization of Our Detector

Our proposed detector can be applied to any visual encoder. All of our experiments so far are using the features extracted from visual foundational models. Now, another question that could be raised is what happens if we give a feature map that is specifically trained for keypoint detection and tested with the proposed detector. Will it improve the performance of the trained detector? To address these questions, we experimented with the full-resolution features extracted from a pretrained keypoint extractor, DISK [42]. The quantitative evaluation on the HPatches dataset is shown in Table 5.3. As one can see, our proposed detector is able to outperform the trained detector in both viewpoint and illumination categories at 5px and the competing method D2D in all metrics, proving the effectiveness of the proposed detector.

5.4.6 NN-distance vs Match Success Rate

We conducted a study to understand how the proposed NN-distance correlates with the success of matching on the HPatches dataset. The success of a match is determined based on the epipolar distance. After detection and matching, the estimated matches were used to compute the fundamental matrix using MAGSAC++ [284] with a threshold of 1px. We used the best-performing feature encoders: DINO-V2, as shown in Table 5.1 for mean matching accuracy and Table 5.2(b) for wide-baseline camera pose estimation; DINO-V1†, as shown in Table 5.2(a) for narrow-baseline camera pose estimation; and DISK features, as shown in Table 5.3, all combined with our proposed detector. The results, depicted in Figure 5.5, show that the mean success rate steadily increases as the nearest-neighbour distance (NN-distance)

increases across all models. This experiment provides additional insight into the potential of the nearest-neighbour criterion as an effective metric for identifying keypoints that can be consistently tracked between frames.

5.5 Limitations and Future Work

The accuracy of keypoint detection relies heavily on the resolution of the feature map. Currently, our method suffers due to the low-resolution feature maps extracted from the foundation models. It is evident from the experiments performed on HPatches and Oxford RobotCar Dataset that DINO† with patch-size 8 performs very well despite being trained with less data compared to the other recent foundational models, DINO-V2 and AM-RADIO. In the future, one could distill models with larger patch sizes into smaller patch sizes or use upsampling methods such as [265] to increase the feature resolution to achieve more precice keypoint detection.

Another problem evident from our experiments is that the foundation model features perform poorly in cases where there is a large viewpoint change. This is likely because such models were never trained with large view-point variations in the data. We also leave this exploration for future work.

5.6 Conclusions

In this chapter, we propose a learning-free keypoint detector that can be applied to any feature map extracted from a CNN or a ViT. Our detector uses the local density of the features in the feature space to determine if a descriptor should be a keypoint or not. In doing so, we are able to detect high-quality keypoints with no prior assumptions about the underlying distribution of the data, unlike alternative methods.

Given the versatility of our approach, we applied it to a multitude of feature encoders. Our feature encoders include the recent foundational encoders DINO, DINO-V2, CLIP, and AM-RADIO—given their strong and general visual representation learning capabilities—as well as a pretrained local descriptor and keypoint detector network, DISK. Our proposed detector was tested with these features on keypoint matching and camera-pose estimation tasks under narrow and wide-baseline configurations across three different datasets. A careful ablation study on the HPatches dataset was performed to understand how positively the local density of the features correlates with successful matching.

During the experimental evaluation, we found that the proposed detector comfortably outperforms the closest state-of-the-art learning free detector D2D, in all tasks and configurations. We also found that no single foundational feature encoder performs best in all tasks. Interestingly, among the foundational features, AM-RADIO was the most illumination-invariant, and DINO-V2 was more view-invariant. Also, despite its extensive training or large distilled knowledge, DINO-V1 with patch size 8×8 performed the best in a narrow-baseline camera pose estimation task. We again attribute this to the spatial resolution of the feature maps, similar to our findings in the stereo-depth estimation task in the previous chapter. Lastly, the foundational features seem to lack certain view-invariant properties that are crucial for keypoint matching task. This in future, could potentially be added by fine-tuning the existing models or including a data augmentation technique like homographic adaptation [143] during the training.

To answer our Research Question 3, our proposed detector—despite never being explicitly trained for keypoint detection or description—can extract robust, high-quality keypoints directly from the feature maps of visual foundational models without requiring additional training. However, when applied to downstream tasks such as homography and camera pose estimation, we find that not all foundational models are equally effective for keypoint detection. In particular, these models often lack the necessary resolution and view invariance, limiting their ability to deliver precise keypoint detection and robust viewpoint-invariant matching.

Chapter 6

Conclusions and Future Work

This thesis explores the problem of self-supervised depth and ego-motion estimation from purely RGB images captured by a standard digital camera. Cameras, as our primary sensor of choice, capture the 2D projection of the world, making it challenging to infer 3D information such as depth and camera motion (a.k.a ego-motion). While these problems have garnered significant attention in the computer vision community, as discussed in Chapter 2, we are still far from achieving industrial-level adequacy. However, the rise of deep learning since 2012 has significantly accelerated progress, narrowing the gap at an unprecedented pace.

Deep learning methods are data-driven, requiring large amounts of data with meaningful signals for training. Many early techniques relied on direct supervision, using ground-truth depth and ego-motion data. Although these methods produced superior results, their success was limited due to the prohibitive costs and constraints associated with acquiring per-pixel ground truth maps and accurate ego-motion data—factors such as cost, time, and lack of diversity posed significant challenges. To alleviate this issue, methods were developed that use a training signal called photometric loss, estimated directly from the input images, instead of relying on external data. However, this approach came with a trade-off: a significant drop in the accuracy of depth and ego-motion estimates compared to supervised methods. In response, many techniques sought to bridge this gap by incorporating geometric and semantic priors, modifying network architectures, and imposing additional constraints on the loss function. Photometric loss computation assumes that the scene of interest is static, illumination invariant and the captured images are blur and noise-free. Hence, most self-supervised methods have primarily focused on well-illuminated data (e.g. daytime data) where scenes are well-lit, and images are almost free from noise and blur. However, when tested on poorly illuminated data (e.g. nighttime data), these methods have been observed to fail miserably in both monocular and stereo camera settings. Moreover, training with poorly illuminated data using this loss function proved even more challenging. Motivated by these observations, we ask the question, how to design a generalisable system

that can estimate accurate depth and camera motion while being robust to all kinds of illuminations, noises and image-blur while estimating. With this motivation in mind, we asked 3 specific research questions and proposed 3 methods with great future potential to answer the aforesaid question. The details are as follows:

Question 1. How can we relax the assumptions made by the photometric loss to use it for robust self-supervised monocular depth and ego-motion estimation?

In Chapter 3, a combination of three modules is proposed to relax the assumptions previously made by the photometric loss, allowing it to be used for training on data with wide-variety of illumination conditions, dynamic objects and noise. The first module, the neural intensity transform, accounts for illumination inconsistencies between consecutive images that may happen because of various light sources. This module is composed of per-pixel scaling and shift terms to address contrast and brightness variations in the images. The second module addresses dynamic changes that could happen in a scene. For example, vehicles, pedestrians, and cyclists in outdoor driving scenarios. It estimates the residual pixel flow necessary to complement the ego-flow for an accurate reconstruction of images containing dynamic pixels. Lastly, a simple yet effective denoising module is applied during the loss calculation using a pretrained denoiser to handle noise present in the original and the reconstructed images. This solution is straightforward and does not require proxy losses computed using external features or generative training as done in the literature. Several ablation and experimental studies are performed with these modules for the task of monocular depth estimation on an autonomous driving dataset to understand their efficacy and relative improvements over the competing approaches. Despite their simplicity, the proposed changes to the existing photometric loss relaxed the assumptions and made it possible to train a single robust model for depth and ego-motion estimation. Then, to further understand the limitations of the photometric loss for stereo-depth estimation we ask the next question,

Question 2. Is it possible to make a self supervised stereo matching system that is accurate, robust, and generalisable?

Existing traditional and supervised methods fail miserably in low SNR regions of the image when tested on the poorly-illuminated data directly. To the best of our knowledge, no self-supervised stereo-matching networks have been trained or tested under nighttime conditions. Besides, similar to large language models (LLMs), there has been huge progress

in visual foundational models that are trained on large corpus of image data could extract features which are proven to generalise in many computer vision tasks. With these advancements in generlisable feature learning and the limitations of the existing stereo-matching methods, we proposed a self supervised method that could use the features extracted from any existing pretrained foundational encoder and a trainable decoder to estimate accurate disparity maps in Chapter 4. To address the issues caused by low SNR, we developed a mask that utilizes features extracted from a visual foundational encoder to filter out uninformative regions. The informativeness of each feature is measured by its distance to the nearest neighbour on a unit sphere. If the distance from the nearest neighbour falls below a certain threshold, those features are excluded from stereo matching. To prevent excessive pixel masking, we encouraged the features to spread more evenly across the unit sphere using a regularisation loss. We performed various ablation studies and experiments using different foundational models, patch sizes, and image warpings to demonstrate the effectiveness of our approach. While answering our objective, we made the following observations.

1. We were able to significantly increase the robustness of the stereo-matching method by addressing the issues with the low-SNR regions in the images and able to train a single model for that is robust to wide variety of illumination variations.

2. We tested the generalisation capability of the model on a completely different dataset, varying in infrastructure, lighting conditions. Our method performed comparably with state-of-the-art stereo methods overall. However, when compared to a specific method that is trained end-to-end on the Oxford RobotCar dataset, our method outperformed it despite being only trained with self-supervision.

3. In terms of accuracy, our method performs comparable if not better to the methods that are trained with full supervision and end to end. The best performing model is DINO-V1 with patch size 8. Despite being trained 142 million images for DINO-V2, or being distilled from various foundational models for AM-RADIO, they performed far worse compared to DINO-V1. We attribute this inferior performance to the bigger patch sizes leading to features with smaller resolutions they use compared to DINO-V1.

This method demonstrates a promising direction for leveraging foundational features instead of training models end-to-end or fine-tuning encoders for every specific scene of interest. Given the impressive performance of the foundational features in the task of stereo matching, we further explored how these features could also be used to estimate the camera motion between images taken with narrow and wide baselines. More specifically, we ask the following question,

Question 3. How to harness the feature representations of foundational models for keypoint detection and matching, thereby enhancing tasks like camera motion estimation?

While CNN-based methods can directly regress motion from extracted features, keypoint-based approaches often achieve greater accuracy and generalizability by leveraging salient, repeatable image points that remain robust under varying illumination and viewpoint conditions. This naturally raises the question: how can we efficiently find these keypoints in a given image? To address this, we introduce a learning-free detector that uses local density in feature space to determine whether a pixel is a keypoint. Concretely, we examine the distribution of features on a unit sphere; the farther a feature lies from its nearest neighbor, the more unique the keypoint is. This simple criterion proves effective, particularly when paired with features that maintain local density consistency across images. We argue that the way foundational models (e.g., DINO-V1, DINO-V2) are trained preserves this local density to some extent. For instance, DINO-based training objectives maximize feature similarity between random crops (resized to the original resolution) and the full image, which makes features robust to scale variations. However, because these training augmentations do not incorporate view-invariant transformations, the resulting features remain sensitive to viewpoint changes. Consequently, while DINO-V2 and AM-RADIO can outperform DISK (a specialized keypoint detector and descriptor) on tasks with no viewpoint changes, such as the HPatches illumination split, they perform poorly when confronted with substantial viewpoint variation, as evidenced on the HPatches viewpoint split and the ScanNet wide-baseline splits.

Another important factor affecting performance is the resolution of feature maps. By design, many foundational models produce feature maps at a resolution determined by the input image size and the model's patch size, often leading to lower-resolution feature maps that must be upsampled for pixel-level keypoint detection. This resolution mismatch can degrade performance, a trend also noted in our dense stereo-matching experiments Chapter 4. To explore the impact of high-resolution feature maps, we tested our detector on full-resolution DISK features; DISK being a trained keypoint detector/descriptor using ground-truth correspondences. On the HPatches dataset, our proposed detector extracted keypoints more robust for matching than DISK's own detector, illustrating the effectiveness of our approach. Overall, Chapter 5 concludes that our local-density-based criterion is promising and can potentially convert any image encoder into a keypoint detector. However, its ultimate performance depends on how the encoder itself was trained, particularly with respect to viewpoint-invariant augmentations and feature map resolution.

This thesis investigates three main areas: (1) relaxing the assumptions of photometric loss to handle varied illuminations, dynamic scenes, and noisy settings; (2) proposing a robust, generalizable system with strong potential for accurate depth estimation via stereo matching; and (3) designing a keypoint detector capable of extracting high-quality keypoints from any feature extractor without additional training. While looking into these, we also highlight an additional promising directions toward approaches that prioritize "less learning, more inferring".

Limitations

In this section, we discuss several current constraints of our approach and outline potential avenues for future improvement.

1. **Computational complexity and efficiency.** Although our methods exhibit promising performance, we have yet to conduct a detailed analysis of their runtime and computational requirements. A thorough evaluation across a range of hardware platforms and real-time scenarios is necessary to better assess its viability for resource-constrained or embedded systems.

2. **Reliance on RGB cameras.** Currently, our approaches are specifically designed for standard RGB cameras, which can encounter difficulties in extremely low-light conditions. Future research could adapt the method to work with alternative sensor types (such as infrared or thermal cameras) or integrate sensor fusion techniques to improve robustness in challenging illumination environments.

3. **Evaluation across broader datasets.** While initial tests on selected datasets have shown encouraging results, the overall generalisability of our methods across varying conditions including: different camera models, environmental factors, and scene types, remains to be proven. Expanding the evaluation to include a wider range of datasets is essential to validate its real-world applicability.

4. **Linear Neural Intensity Transform.** Our current implementation employs a linear intensity transformation on the images directly to address the changes in illumination. However, the image formation from radiance to pixel color conversion is non-linear. Exploring more adaptive or flexible intensity transformation models could enhance performance.

5. **Patch size influence in stereo matching and keypoint detection**

   Foundational models have shown impressive results across various tasks, such as stereo matching and keypoint detection, by leveraging knowledge from large pretrained corpora. However, they often produce low-resolution feature maps caused by the patch-sizes they use to partition the input image. This hinder fine-grained tasks, and retraining them from scratch remains computationally prohibitive. A promising solution is to insert a lightweight adaptor network that learns to upsample or refine the foundational model's features, thus retaining the benefits of the original pretraining while mitigating the high computational demands.

Future Work

The work presented in this thesis also offers many interesting avenues for future research, a few of which are introduced below.

Decoder only learning for task-specific adaptation: We have shown evidence that the generalization and robustness of pretrained visual foundational models can be inherited by using them as encoders. Two of the strongest models currently available are DINO-V2 and AM-RADIO. However, these models are trained to output task-agnostic features. To harness the knowledge from these models and decode it into task-specific information, such as depth estimation, camera motion, or other tasks, one would need to train a dedicated decoder, like the one we propose in Chapter 4. The next question is whether we could train foundational task-specific decoders that can be paired with any foundational encoder. This would create a flexible system where pretrained encoders extract general-purpose features, while task-specific decoders, fine-tuned for particular tasks, such as depth estimation or camera motion detection, interpret those features and produce meaningful outputs. By decoupling the general-purpose representation learning from task-specific decoding, we could make the training process more efficient and improve task performance. Such a system would also reduce the need for retraining or fine-tuning large foundational models. Instead, the encoder remains fixed while the decoder adapts to new tasks. This approach could scale more effectively and make better use of the capabilities of foundational models across various computer vision tasks.

Development of compact yet high-performing models: Although this thesis has primarily concentrated on improving accuracy, robustness, and generalization, computational efficiency remains a critical concern, particularly for applications in augmented reality and mobile computing. Future work should investigate the design of smaller foundational models that retain the rich representational capacity of larger counterparts while significantly

reducing computational, memory, and energy requirements. Techniques such as network pruning, quantization, and knowledge distillation could be employed to compress models without sacrificing performance. Moreover, exploring efficient encoder architectures that are optimized for real-time inference on edge devices will be essential for enabling practical applications.

Sensor fusion for enhanced 3D scene understanding: This thesis has focused on the use of RGB cameras as the primary sensor modality. However, many real-world systems, such as autonomous robots, AR/VR devices, and UAVs, integrate additional sensors like IMUs, RADAR, and LiDAR. Future work could explore the fusion of these heterogeneous data streams to improve the robustness and accuracy of depth and camera-motion estimation. For example, integrating IMU data could provide reliable motion priors that constrain ego-motion estimation, while even sparse RADAR or LiDAR signals might enhance depth predictions in challenging environments. The design of effective sensor fusion architectures, possibly incorporating attention mechanisms to weigh sensor contributions dynamically, represents an exciting area for further research.

Self-Supervised foundational models: Current depth estimation models, such as Depth Anything V1, V2, and others, are predominantly trained in a supervised manner using ground truth real and synthetic images with high-quality depth annotations. However, acquiring such high-quality depth data for every scene of interest is often impractical, especially when the desired expectations are not met in diverse environments. An exciting future direction is the development of self-supervised foundational models that either extend the capabilities of these existing supervised models or learn representations entirely from scratch using vast amounts of unlabeled data.

The self-supervised approaches could leverage objective functions proposed in this thesis to enforce temporal consistency, spatial coherence, and other structural constraints without the need for explicit ground truth depth labels. By utilizing these novel loss functions, the model can learn rich, task-agnostic features that capture the underlying geometry and dynamics of complex scenes. Such models have the potential to learn more generalized and robust representations, leading to improved performance in depth estimation tasks even under challenging conditions. Moreover, a self-supervised foundational model could continuously refine its predictions by integrating new, unlabeled data from varied environments, thereby progressively enhancing its accuracy and adaptability.

Overall, these future research directions emphasize a direction towards systems that are not only highly accurate, robust, and generalizable but also adaptable and efficient. By advancing decoder-only learning, self-supervised foundational models, sensor fusion

strategies, and compact model design, we can significantly extend the applicability and scalability of visual foundational models in complex, real-world scenarios.

Bibliography

[1] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access*, 8:58443–58469, 2020. 1, 53

[2] Jose Sanchez, Juan-Antonio Corrales, Belhassen-Chedli Bouzgarrou, and Youcef Mezouar. Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey. *The International Journal of Robotics Research*, 37(7):688–716, 2018. 1, 2, 53

[3] Songlin Wei, Haoran Geng, Jiayi Chen, Congyue Deng, Cui Wenbo, Chengyang Zhao, Xiaomeng Fang, Leonidas Guibas, and He Wang. D3roma: Disparity diffusion-based depth sensing for material-agnostic robotic manipulation. In *8th Annual Conference on Robot Learning*, 2024. 1, 2

[4] Mark A Livingston, Zhuming Ai, J Edward Swan, and Harvey S Smallman. Indoor vs. outdoor depth perception for mobile augmented reality. In *2009 IEEE Virtual Reality Conference*, pages 55–62. IEEE, 2009. 1, 53

[5] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2, 85

[6] Peteris Racinskis, Janis Arents, and Modris Greitans. Constructing maps for autonomous robotics: An introductory conceptual overview. *Electronics*, 12(13):2925, 2023. 2

[7] Alessandro Gasparetto, Paolo Boscariol, Albano Lanzutti, and Renato Vidoni. Path planning and trajectory planning algorithms: A general overview. *Motion and operation planning of robotic systems: Background and practical approaches*, pages 3–27, 2015. 2

[8] Fabio Remondino. Heritage recording and 3d modeling with photogrammetry and 3d scanning. *Remote sensing*, 3(6):1104–1138, 2011. 2

[9] Jaydeep H Palep. Robotic assisted minimally invasive surgery. *Journal of minimal access surgery*, 5(1):1–7, 2009. 2

[10] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 3, 8, 22, 24, 36, 100, 103, 104, 110, 111

[11] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006. 3, 8, 22, 24, 100, 106

[12] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. 3, 8, 25, 100, 104, 105, 107

[13] 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*, volume 2, pages 674–679, 1981. 3

[14] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981. 3

[15] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014. 3, 23, 25

[16] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 3, 18, 24

[17] Åke Björck. *Numerical methods for least squares problems*. SIAM, 2024. 3

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 4

[19] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014. 5, 26, 27, 54, 56, 76

[20] Fayao Liu, Chunhua Shen, and Guosheng Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5162–5170, 2015. 5, 26, 54, 56

[21] C. Zhao, Y. Zhang, M. Poggi, F. Tosi, X. Guo, Z. Zhu, G. Huang, Y. Tang, and S. Mattoccia. Monovit: Self-supervised monocular depth estimation with a vision transformer. In *2022 International Conference on 3D Vision (3DV)*, pages 668–678, Los Alamitos, CA, USA, Sep 2022. IEEE Computer Society. 5, 41

[22] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Dense visual slam for rgb-d cameras. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2100–2106. IEEE, 2013. 5, 54

[23] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Robust odometry estimation for rgb-d cameras. In *2013 IEEE International Conference on Robotics and Automation*, pages 3748–3754. IEEE, 2013. 5, 54

[24] Andrew I Comport, Ezio Malis, and Patrick Rives. Accurate quadrifocal tracking for robust 3d visual odometry. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 40–45. IEEE, 2007. 5, 54

[25] Aashish Sharma, Loong-Fah Cheong, Lionel Heng, and Robby T Tan. Nighttime stereo depth estimation using joint translation-stereo learning: Light effects and uninformative regions. In *2020 International Conference on 3D Vision (3DV)*, pages 23–31. IEEE, 2020. 5, 54, 57

[26] Madhu Vankadari, Sourav Garg, Anima Majumder, Swagat Kumar, and Ardhendu Behera. Unsupervised monocular depth estimation for night-time images using adversarial domain feature adaptation. In *European Conference on Computer Vision*, pages 443–459. Springer, 2020. 5, 7, 42, 43, 57, 76

[27] Jaime Spencer, Richard Bowden, and Simon Hadfield. Defeat-net: General monocular depth via simultaneous unsupervised representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14402–14413, 2020. 5, 7, 42, 43, 57, 60, 63, 65, 66, 67, 68, 69, 71, 76

[28] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019. 5, 7, 63, 65, 78, 79, 84, 89

[29] Madhu Vankadari, Stuart Golodetz, Sourav Garg, Sangyun Shin, Andrew Markham, and Niki Trigoni. When the sun goes down: Repairing photometric losses for all-day depth estimation, 2022. 5, 76, 78

[30] Travis Portz, Li Zhang, and Hongrui Jiang. Optical flow in the presence of spatially-varying motion blur. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1752–1759. IEEE, 2012. 5, 54

[31] Alex Rav-Acha and Shmuel Peleg. Restoration of multiple images with motion blur in different directions. In *Proceedings Fifth IEEE Workshop on Applications of Computer Vision*, pages 22–28. IEEE, 2000. 5, 54

[32] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 5, 27, 45, 79, 93, 101, 104

[33] 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. *arXiv preprint arXiv:2304.07193*, 2023. 6, 10, 28, 71, 77, 79, 80, 94, 101, 104, 105, 109, 111

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 6, 46, 79, 101, 104, 109, 111

[35] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 6, 46, 79, 104, 109, 111, 114

[36] Mike Ranzinger, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. Am-radio: Agglomerative visual foundation model – reduce all domains into one. In *CVPR*, 2024. 6, 10, 46, 77, 79, 80, 94, 104, 109, 111

[37] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017. 7, 39, 40, 43, 56, 59, 65, 68, 69, 76

[38] Kun Wang*, Zhenyu Zhang*, Zhiqiang Yan, Xiang Li, Baobei Xu, Jun Li, and Jian Yang. Regularizing Nighttime Weirdness: Efficient Self-supervised Monocular Depth Estimation in the Dark. In *ICCV*, pages 16055–16064, 2021. 7, 47, 48, 57, 60, 65, 66, 67, 69, 70, 71, 74, 76

[39] Gangwei Xu, Xianqi Wang, Xiaohuan Ding, and Xin Yang. Iterative geometry encoding volume for stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21919–21928, 2023. 7, 8, 29, 31, 33, 42, 76, 77, 79, 82, 85, 88, 89, 90, 96

[40] Johan Edstedt, Georg Bökman, and Zhenjun Zhao. DeDoDe v2: Analyzing and Improving the DeDoDe Keypoint Detector . In *IEEE/CVF Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2024. 8, 38, 100, 103, 104

[41] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 10, 77, 80, 93

[42] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. *Advances in Neural Information Processing Systems*, 33:14254–14265, 2020. 13, 35, 37, 100, 103, 106, 116

[43] David Marr and Tomaso Poggio. A computational theory of human stereo vision. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 204(1156):301–328, 1979. 15

[44] Takeo Kanade and Masatoshi Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE transactions on pattern analysis and machine intelligence*, 16(9):920–932, 1994. 15, 18

[45] Heiko Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 2, pages 807–814. IEEE, 2005. 16, 18

[46] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239, 2001. 16, 18

[47] Yuichi Ohta and Takeo Kanade. Stereo by intra-and inter-scanline search using dynamic programming. *IEEE Transactions on pattern analysis and machine intelligence*, (2):139–154, 1985. 16

[48] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 1, pages 519–528. IEEE, 2006. 16

[49] George Vogiatzis, Philip HS Torr, and Roberto Cipolla. Multi-view stereo via volumetric graph-cuts. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 391–398. IEEE, 2005. 16

[50] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2009. 16

[51] Jean-Philippe Pons, Renaud Keriven, and Olivier Faugeras. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *International Journal of Computer Vision*, 72:179–193, 2007. 16

[52] Robert J Woodham. Photometric method for determining surface orientation from multiple images. *Optical engineering*, 19(1):139–144, 1980. 17, 18

[53] Katsushi Ikeuchi and Berthold KP Horn. Numerical shape from shading and occluding boundaries. *Artificial intelligence*, 17(1-3):141–184, 1981. 17

[54] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape-from-shading: a survey. *IEEE transactions on pattern analysis and machine intelligence*, 21(8):690–706, 1999. 17

[55] Alex Paul Pentland. A new sense for depth of field. *IEEE transactions on pattern analysis and machine intelligence*, (4):523–531, 1987. 17, 18

[56] Murali Subbarao and Gopal Surya. Depth from defocus: A spatial domain approach. *International Journal of computer vision*, 13(3):271–294, 1994. 17

[57] Andrew P Witkin. Recovering surface shape and orientation from texture. *Artificial intelligence*, 17(1-3):17–45, 1981. 17

[58] Jonas Gårding. Shape from texture for smooth curved surfaces in perspective projection. *Journal of Mathematical Imaging and Vision*, 2:327–350, 1992. 17

[59] Jitendra Malik and Ruth Rosenholtz. Computing local surface orientation and shape from texture for curved surfaces. *International journal of computer vision*, 23(2):149–168, 1997. 17

[60] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. Spie, 1992. 17, 20, 21

[61] Franc¸ ois Blais. Review of 20 years of range sensor development. *Journal of electronic imaging*, 13(1):231–243, 2004. 17

[62] Paul J Besl and Neil D McKay. A method for registration of 3-d shapes. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 14, pages 239–256. IEEE, 1992. 17, 18

[63] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem. *International Journal of Computer Vision*, 81(2):155–166, 2009. 19, 21

[64] David Nister. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770, 2004. 19, 21

[65] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. 19, 21

[66] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976. 20, 21

[67] Wolfgang Kabsch. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 34(5):827–828, 1978. 20, 21

[68] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *Proceedings third international conference on 3-D digital imaging and modeling*, pages 145–152. IEEE, 2001. 20, 21

[69] Kok-Lim Low. Linear least-squares optimization for point-to-plane icp surface registration. *Chapel Hill, University of North Carolina*, 4(10):1–3, 2004. 20, 36

[70] Juyong Zhang, Yuxin Yao, and Bailin Deng. Fast and robust iterative closest point. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3450–3466, 2021. 20

[71] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings*, pages 298–372. Springer, 2000. 20, 21, 24

[72] Chris Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988. 22, 24, 100, 105, 106, 107, 110, 111

[73] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011. 22, 24, 100, 105

[74] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*, pages 430–443. Springer, 2006. 23, 24, 105, 106

[75] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *European conference on computer vision*, pages 778–792. Springer, 2010. 23, 24

[76] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J Davison. Kaze features. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12*, pages 214–227. Springer, 2012. 23, 24

[77] Pablo F Alcantarilla and T Solutions. Fast explicit diffusion for accelerated features in nonlinear scale spaces. *IEEE Trans. Patt. Anal. Mach. Intell*, 34(7):1281–1298, 2011. 23, 24

[78] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *2011 international conference on computer vision*, pages 2320–2327. IEEE, 2011. 23, 54

[79] Davide Scaramuzza and Friedrich Fraundorfer. Visual odometry [tutorial]. *IEEE robotics & automation magazine*, 18(4):80–92, 2011. 23, 24, 25

[80] Berta Bescos, José M Fácil, Javier Civera, and José Neira. Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4):4076–4083, 2018. 24

[81] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. *ACM Transactions on Graphics (TOG)*, 25(3):835–846, 2006. 24

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

[83] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, pages 225–234. IEEE, 2007. 25

[84] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 15–22. IEEE, 2014. 25

[85] H Durrant Whyte. Simultaneous localisation and mapping (slam): Part i the essential algorithms. *Robotics and Automation Magazine*, 2006. 25

[86] Tim Bailey and Hugh Durrant-Whyte. Simultaneous localization and mapping (slam): Part ii. *IEEE robotics & automation magazine*, 13(3):108–117, 2006. 25

[87] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016. 25

[88] Thomas Whelan, Raul F Salas-Moreno, Ben Glocker, Andrew J Davison, and Stefan Leutenegger. Elasticfusion: Real-time dense slam and light source estimation. In *Proceedings of Robotics: Science and Systems*, 2015. 25

[89] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 40, pages 611–625. IEEE, 2017. 25

[90] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2018. 25

[91] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10371–10381, 2024. 27, 28, 33, 72, 104

[92] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 239–248. IEEE, 2016. 26

[93] Jun Li, Reinhard Klein, and Angela Yao. A two-streamed network for estimating fine-scaled depth maps from single rgb images. In *Proceedings of the IEEE international conference on computer vision*, pages 3372–3380, 2017. 26

[94] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018. 27, 56

[95] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5684–5693, 2019. 27

[96] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019. 27, 76

[97] Dan Xu, Xavier Alameda-Pineda, Wanli Ouyang, Elisa Ricci, Xiaogang Wang, and Nicu Sebe. Probabilistic graph attention network with conditional kernels for pixel-wise prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5):2673–2688, 2020. 27

[98] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020. 27, 76

[99] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021. 27, 76

[100] Guanglei Yang, Hao Tang, Mingli Ding, Nicu Sebe, and Elisa Ricci. Transformer-based attention networks for continuous pixel-wise prediction. In *Proceedings of the IEEE/CVF International Conference on Computer vision*, pages 16269–16279, 2021. 28, 41

[101] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 28

[102] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4008–4017, 2021. 28, 76

[103] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. Neural window fully-connected crfs for monocular depth estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3916–3925, 2022. 28

[104] Zhenyu Li, Xuyang Wang, Xianming Liu, and Junjun Jiang. Binsformer: Revisiting adaptive bins for monocular depth estimation. *IEEE Transactions on Image Processing*, 2024. 28

[105] Reiner Birkl, Diana Wofk, and Matthias Müller. Midas v3. 1–a model zoo for robust monocular relative depth estimation. *arXiv preprint arXiv:2307.14460*, 2023. 28, 76

[106] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 28

[107] Jae-Han Lee and Chang-Su Kim. Monocular depth estimation using relative depth maps. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2019. 28

[108] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024. 28, 72, 104

[109] Daniel Hernandez-Juarez, Alejandro Chacón, Antonio Espinosa, David Vázquez, Juan Carlos Moure, and Antonio M López. Embedded real-time stereo estimation via semi-global matching on the gpu. *Procedia Computer Science*, 80:143–153, 2016. 29, 82, 85, 88, 89, 90, 96

[110] 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*, pages 4040–4048, 2016. 29, 85

[111] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30(2):328–341, 2007. 29

[112] Wenjie Luo, Alexander G Schwing, and Raquel Urtasun. Efficient deep learning for stereo matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5695–5703, 2016. 29, 76

[113] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE international conference on computer vision*, pages 66–75, 2017. 29

[114] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5410–5418, 2018. 29

[115] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015. 29

[116] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. Ga-net: Guided aggregation net for end-to-end stereo matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 185–194, 2019. 30

[117] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. Group-wise correlation stereo network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3273–3282, 2019. 30

[118] Youmin Zhang, Yimin Chen, Xiao Bai, Suihanjin Yu, Kun Yu, Zhiwei Li, and Kuiyuan Yang. Adaptive unimodal cost volume filtering for deep stereo matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12926–12934, 2020. 30

[119] Xiao Song, Xu Zhao, Liangji Fang, Hanwen Hu, and Yizhou Yu. Edgestereo: An effective multi-task learning network for stereo matching and edge detection. *International Journal of Computer Vision*, 128(4):910–930, 2020. 30

[120] Xuelian Cheng, Yiran Zhong, Mehrtash Harandi, Yuchao Dai, Xiaojun Chang, Hongdong Li, Tom Drummond, and Zongyuan Ge. Hierarchical neural architecture search for deep stereo matching. *Advances in neural information processing systems*, 33:22158–22169, 2020. 30

[121] Vladimir Tankovich, Christian Hane, Yinda Zhang, Adarsh Kowdle, Sean Fanello, and Sofien Bouaziz. Hitnet: Hierarchical iterative tile refinement network for real-time stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14362–14372, 2021. 30

[122] Haofei Xu and Juyong Zhang. Aanet: Adaptive aggregation network for efficient stereo matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1959–1968, 2020. 30

[123] Zhelun Shen, Yuchao Dai, and Zhibo Rao. Cfnet: Cascade and fused cost volume for robust stereo matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13906–13915, 2021. 30

[124] Gangwei Xu, Junda Cheng, Peng Guo, and Xin Yang. Attention concatenation volume for accurate and efficient stereo matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12981–12990, 2022. 31

[125] Jiankun Li, Peisen Wang, Pengfei Xiong, Tao Cai, Ziwei Yan, Lei Yang, Jiangyu Liu, Haoqiang Fan, and Shuaicheng Liu. Practical stereo matching via cascaded recurrent network with adaptive correlation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16263–16272, 2022. 31

[126] Lahav Lipson, Zachary Teed, and Jia Deng. Raft-stereo: Multilevel recurrent field transforms for stereo matching. In *2021 International Conference on 3D Vision (3DV)*, pages 218–227. IEEE, 2021. 31

[127] 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*, pages 402–419. Springer, 2020. 31, 79, 83

[128] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 31, 33, 42, 76, 79, 81, 82, 83, 85, 88, 89, 90, 96

[129] Xianqi Wang, Gangwei Xu, Hao Jia, and Xin Yang. Selective-stereo: Adaptive frequency information selection for stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19701–19710, 2024. 31

[130] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 5, 2017. 32

[131] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018. 32

[132] Ronald Clark, Michael Bloesch, Jan Czarnowski, Stefan Leutenegger, and Andrew J Davison. Learning to solve nonlinear least squares for monocular stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 284–299, 2018. 32

[133] Sunghoon Im, Hae-Gon Jeon, Stephen Lin, and In So Kweon. Dpsnet: End-to-end deep plane sweep stereo. *arXiv preprint arXiv:1905.00538*, 2019. 32

[134] Robert T Collins. A space-sweep approach to true multi-image matching. In *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 358–363. IEEE, 1996. 32

[135] Zachary Teed and Jia Deng. Deepv2d: Video to depth with differentiable structure from motion. *arXiv preprint arXiv:1812.04605*, 2018. 32

[136] Arda Duzceker, Silvano Galliani, Christoph Vogel, Pablo Speciale, Mihai Dusmanu, and Marc Pollefeys. Deepvideomvs: Multi-view stereo on video with recurrent spatio-temporal fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15324–15333, 2021. 32

[137] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015. 32

[138] Xiaoxiao Long, Lingjie Liu, Wei Li, Christian Theobalt, and Wenping Wang. Multi-view depth estimation using epipolar spatio-temporal networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8258–8267, 2021. 32

[139] 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*, pages 5998–6008, 2017. 32, 45

[140] Mohamed Sayed, John Gibson, Jamie Watson, Victor Prisacariu, Michael Firman, and Clément Godard. Simplerecon: 3d reconstruction without 3d convolutions. In *European Conference on Computer Vision*, pages 1–19. Springer, 2022. 32

[141] Zeyu Ma, Zachary Teed, and Jia Deng. Multiview stereo with cascaded epipolar raft. In *European Conference on Computer Vision*, pages 734–750. Springer, 2022. 33

[142] Sili Chen, Hengkai Guo, Shengnan Zhu, Feihu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video depth anything: Consistent depth estimation for super-long videos. *arXiv preprint arXiv:2501.12375*, 2025. 33, 72

[143] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. 34, 35, 36, 37, 100, 103, 106, 109, 111, 118

[144] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *2017 IEEE*

International Conference on Robotics and Automation (ICRA), pages 2043–2050. IEEE, 2017. 34, 35

[145] Ronald Clark, Sen Wang, Hongkai Wen, Andrew Markham, and Niki Trigoni. Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017. 34

[146] Wenshan Wang, Yaoyu Hu, and Sebastian Scherer. Tartanvo: A generalizable learning-based vo. In *Conference on Robot Learning*, pages 1761–1772. PMLR, 2021. 34

[147] Ruihao Li, Sen Wang, Zhiqiang Long, and Dongbing Gu. Undeepvo: Monocular visual odometry through unsupervised deep learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7286–7291. IEEE, 2018. 34, 40, 43, 56

[148] Lei Lai, Zhongkai Shangguan, Jimuyang Zhang, and Eshed Ohn-Bar. Xvo: Generalized visual odometry via cross-modal self-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10094–10105, 2023. 34

[149] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021. 34

[150] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 35, 37, 38, 48, 82, 103, 114

[151] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint description and detection of local features. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 8092–8101, 2019. 35, 36, 100, 103, 106, 111, 113

[152] Yannick Verdie, Kwang Yi, Pascal Fua, and Vincent Lepetit. Tilde: A temporally invariant learned detector. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5279–5288, 2015. 35, 36, 103

[153] Karel Lenc and Andrea Vedaldi. Learning covariant feature detectors. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pages 100–117. Springer, 2016. 36, 103

[154] Xu Zhang, Felix X Yu, Sanjiv Kumar, and Shih-Fu Chang. Learning spread-out local feature descriptors. In *Proceedings of the IEEE international conference on computer vision*, pages 4595–4603, 2017. 36, 103

[155] Nikolay Savinov, Akihito Seki, Lubor Ladicky, Torsten Sattler, and Marc Pollefeys. Quad-networks: unsupervised learning to rank for interest point detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1822–1830, 2017. 36, 103, 108

[156] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *European conference on computer vision*, pages 467–483. Springer, 2016. 36, 38, 100, 103

[157] Axel Barroso-Laguna, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Key.net: Keypoint detection by handcrafted and learned cnn filters. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5836–5844, 2019. 36, 103

[158] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. *Advances in neural information processing systems*, 30, 2017. 36, 39, 101, 103, 104, 105

[159] Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, and Vassileios Balntas. Sosnet: Second order similarity regularization for local descriptor learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11016–11025, 2019. 36, 39, 101, 103, 104, 105

[160] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. Lf-net: Learning local features from images. *arXiv preprint arXiv:1805.09662*, 2018. 36, 100, 103, 109

[161] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. *Advances in neural information processing systems*, 32, 2019. 37, 39, 100, 103, 104, 107, 109

[162] Zixin Luo, Lei Zhou, Xuyang Bai, Hongkai Chen, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. Aslfeat: Learning local features of accurate shape and localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6589–6598, 2020. 37, 100, 103

[163] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 37

[164] Xiaoming Zhao, Xingming Wu, Jinyu Miao, Weihai Chen, Peter CY Chen, and Zhengguo Li. Alike: Accurate and lightweight keypoint detection and descriptor extraction. *IEEE Transactions on Multimedia*, 2022. 37, 103, 109

[165] Xiaoming Zhao, Xingming Wu, Weihai Chen, Peter CY Chen, Qingsong Xu, and Zhengguo Li. Aliked: A lighter keypoint and descriptor extraction network via deformable transformation. *IEEE Transactions on Instrumentation and Measurement*, 72:1–16, 2023. 37, 103

[166] Pierre Gleize, Weiyao Wang, and Matt Feiszli. Silk: Simple learned keypoints. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22499–22508, 2023. 37, 100, 103, 106, 109

[167] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17627–17638, 2023. 37

[168] 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*, pages 8922–8931, 2021. 37, 82, 114

[169] Yifan Wang, Xingyi He, Sida Peng, Dongli Tan, and Xiaowei Zhou. Efficient loftr: Semi-dense local feature matching with sparse-like speed. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21666–21675, 2024. 37

[170] Qing Wang, Jiaming Zhang, Kailun Yang, Kunyu Peng, and Rainer Stiefelhagen. Matchformer: Interleaving attention in transformers for feature matching. In *Proceedings of the Asian Conference on Computer Vision*, pages 2746–2762, 2022. 38

[171] Hongkai Chen, Zixin Luo, Lei Zhou, Yurun Tian, Mingmin Zhen, Tian Fang, David Mckinnon, Yanghai Tsin, and Long Quan. Aspanformer: Detector-free image matching with adaptive span transformer. In *European Conference on Computer Vision*, pages 20–36. Springer, 2022. 38

[172] Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. Dkm: Dense kernelized feature matching for geometry estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17765–17775, 2023. 38

[173] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Roma: Robust dense feature matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19790–19800, 2024. 38, 46, 94, 104

[174] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 38, 100, 103

[175] Kunhong Li, Longguang Wang, Li Liu, Qing Ran, Kai Xu, and Yulan Guo. Decoupling makes weakly supervised local feature better. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15838–15848, 2022. 38, 100, 103

[176] Assia Benbihi, Matthieu Geist, and Cedric Pradalier. Elf: Embedded localisation of features in pre-trained cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7940–7949, 2019. 39, 103

[177] Yurun Tian, Vassileios Balntas, Tony Ng, Axel Barroso-Laguna, Yiannis Demiris, and Krystian Mikolajczyk. D2d: Keypoint extraction with describe to detect approach. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 39, 100, 101, 102, 104, 105, 106, 107, 111, 112

[178] Ravi Garg, Vijay Kumar BG, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pages 740–756. Springer, 2016. 40, 41, 54, 56, 59, 76

[179] Sungmin Woo, Wonjoon Lee, Woo Jin Kim, Dogyoon Lee, and Sangyoun Lee. Prodepth: Boosting self-supervised multi-frame monocular depth with probabilistic fusion. *arXiv preprint arXiv:2407.09303*, 2024. 41, 42

[180] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6602–6611. IEEE, 2017. 40, 56, 76, 83, 93

[181] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015. 40, 56, 59

[182] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 40, 56, 59

[183] V Madhu Babu, Kaushik Das, Anima Majumdar, and Swagat Kumar. Undemon: Unsupervised deep network for depth and ego-motion estimation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1082–1088. IEEE, 2018. 40, 43, 56, 59, 93

[184] Zhichao Yin and Jianping Shi. GeoNet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2018. 40, 56, 61

[185] Chenxu Luo, Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, Ram Nevatia, and Alan Yuille. Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding. *arXiv preprint arXiv:1810.06125*, 2018. 40, 56, 61

[186] Chenxu Luo, Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, Ram Nevatia, and Alan Yuille. Every pixel counts ++: Joint learning of geometry and motion with 3d holistic understanding. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(10):2624–2641, Oct 2020. 40

[187] Yasin Almalioglu, Muhamad Risqi U Saputra, Pedro PB de Gusmao, Andrew Markham, and Niki Trigoni. Ganvo: Unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2019. 40, 54, 56

[188] Filippo Aleotti, Fabio Tosi, Matteo Poggi, and Stefano Mattoccia. Generative adversarial networks for unsupervised monocular depth prediction. In *15th European Conference on Computer Vision (ECCV) Workshops*, volume 1, page 8, 2018. 40, 56

[189] Madhu Vankadari, Swagat Kumar, Anima Majumder, and Kaushik Das. Unsupervised learning of monocular depth and ego-motion using conditional patchgans. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5677–5684. International Joint Conferences on Artificial Intelligence Organization, 7 2019. 40, 56, 57, 59, 68, 76

[190] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel Brostow. Digging into self-supervised monocular depth estimation. *arXiv preprint arXiv:1806.01260*, 2018. 40, 56, 59, 60, 63, 65, 66, 67, 68, 69, 70

[191] Marvin Klingner, Jan-Aike Termöhlen, Jonas Mikolajczyk, and Tim Fingscheidt. Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 582–600. Springer, 2020. 40

[192] Vitor Guizilini, Rareș Ambruș, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3D Packing for Self-Supervised Monocular Depth Estimation. In *CVPR*, pages 2485–2494, 2020. 40, 57, 59

[193] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016. 40

[194] Xiaoyang Lyu, Liang Liu, Mengmeng Wang, Xin Kong, Lina Liu, Yong Liu, Xinxin Chen, and Yi Yuan. HR-Depth: High Resolution Self-Supervised Monocular Depth Estimation. In *AAAI*, pages 2294–2301, 2021. 41, 57

[195] Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. The temporal opportunist: Self-supervised multi-frame monocular depth. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1164–1174, 2021. 41

[196] Zhenyu Li, Zehui Chen, Xianming Liu, and Junjun Jiang. Depthformer: Exploiting long-range correlation and local information for accurate monocular depth estimation. *Machine Intelligence Research*, 20(6):837–854, 2023. 41

[197] Jiaxing Yan, Hong Zhao, Penghui Bu, and YuSheng Jin. Channel-Wise Attention-Based Network for Self-Supervised Monocular Depth Estimation. In *3DV*, pages 464–473, 2021. 41, 57

[198] Andra Petrovai and Sergiu Nedevschi. Exploiting Pseudo Labels in a Self-Supervised Learning Framework for Improved Monocular Depth Estimation. In *CVPR*, pages 1578–1588, 2022. 41, 57

[199] Tak-Wai Hui. RM-Depth: Unsupervised Learning of Recurrent Monocular Depth in Dynamic Scenes. In *CVPR*, pages 1675–1684, 2022. 41, 57

[200] Antyanta Bangunharcana, Ahmed Magd, and Kyung-Soo Kim. Dualrefine: Self-supervised depth and pose estimation through iterative epipolar sampling and refinement toward equilibrium. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 726–738, 2023. 42

[201] Ning Zhang, Francesco Nex, George Vosselman, and Norman Kerle. Lite-mono: A lightweight cnn and transformer architecture for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18537–18546, 2023. 42

[202] F Yu. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 42

[203] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 42

[204] Shuwei Shao, Zhongcai Pei, Weihai Chen, Dingchi Sun, Peter CY Chen, and Zhengguo Li. Monodiffusion: Self-supervised monocular depth estimation using diffusion model. *arXiv preprint arXiv:2311.07198*, 2023. 42

[205] Yue-Jiang Dong, Yuan-Chen Guo, Ying-Tian Liu, Fang-Lue Zhang, and Song-Hai Zhang. Ppea-depth: Progressive parameter-efficient adaptation for self-supervised monocular depth estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 1609–1617, 2024. 42

[206] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR, 2019. 42

[207] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3828–3838, 2019. 42, 43, 76

[208] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8977–8986, 2019. 43

[209] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1281–1292, 2020. 43, 60, 67

[210] Shunkai Li, Xin Wang, Yingdian Cao, Fei Xue, Zike Yan, and Hongbin Zha. Self-supervised deep visual odometry with online adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6339–6348, 2020. 43

[211] Cong Wang, Yu-Ping Wang, and Dinesh Manocha. Motionhint: Self-supervised monocular visual odometry with motion constraints. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 1265–1272. IEEE, 2022. 43

[212] Ganesh Iyer, J Krishna Murthy, Gunshi Gupta, Madhava Krishna, and Liam Paull. Geometric consistency for self-supervised end-to-end visual odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 267–275, 2018. 43

[213] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3302–3312, 2019. 44

[214] Jiexiong Tang, Rares Ambrus, Vitor Guizilini, Sudeep Pillai, Hanme Kim, Patric Jensfelt, and Adrien Gaidon. Self-supervised 3d keypoint learning for ego-motion estimation. In *Conference on robot learning*, pages 2085–2103. PMLR, 2021. 44

[215] Wang Zhao, Shaohui Liu, Yezhi Shu, and Yong-Jin Liu. Towards better generalization: Joint depth-pose learning without posenet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9151–9161, 2020. 44

[216] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 45

[217] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2(3):4, 2021. 46, 79, 81, 104

[218] Nikhil Keetha, Avneesh Mishra, Jay Karhade, Krishna Murthy Jatavallabhula, Sebastian Scherer, Madhava Krishna, and Sourav Garg. Anyloc: Towards universal visual place recognition. *arXiv preprint arXiv:2308.00688*, 2023. 46

[219] 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*, pages 4015–4026, 2023. 46, 79, 104

[220] Lina Liu, Xibin Song, Mengmeng Wang, Yong Liu, and Liangjun Zhang. Self-supervised Monocular Depth Estimation for All Day Images using Domain Separation. In *ICCV*, pages 12737–12746, 2021. 47, 48, 57, 60, 65, 66, 67, 68, 69, 70, 71

[221] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017. 48, 64, 78, 87, 88

[222] Ukcheol Shin, Jinsun Park, and In So Kweon. Deep depth estimation from thermal image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1043–1053, 2023. 47, 87, 88, 95

[223] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5173–5182, 2017. 48, 109

[224] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 48, 109

[225] Kaixuan Wang and Shaojie Shen. Mvdepthnet: Real-time multiview depth estimation neural network. In *2018 International conference on 3d vision (3DV)*, pages 248–257. IEEE, 2018. 54

[226] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised learning of monocular depth estimation and visual

odometry with deep feature reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 340–349, 2018. 54

[227] Tao Huang, Songjiang Li, Xu Jia, Huchuan Lu, and Jianzhuang Liu. Neighbor2Neighbor: Self-Supervised Denoising from Single Noisy Images. In *CVPR*, pages 14781–14790, 2021. 55, 62

[228] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002. 56

[229] Daniel Scharstein and Chris Pal. Learning conditional random fields for stereo. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 56

[230] Ling Zou and Yan Li. A method of stereo vision matching based on opencv. In *2010 International Conference on Audio, Language and Image Processing*, pages 185–190. IEEE, 2010. 56

[231] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 56

[232] Yuchao Dai, Hongdong Li, and Mingyi He. Projective multiview structure and motion from element-wise factorization. *IEEE transactions on pattern analysis and machine intelligence*, 35(9):2238–2251, 2013. 56

[233] Fisher Yu and David Gallup. 3d reconstruction from accidental motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3986–3993, 2014. 56

[234] Lubor Ladicky, Jianbo Shi, and Marc Pollefeys. Pulling things out of perspective. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 89–96, 2014. 56

[235] Nícolas dos Santos Rosa, Vitor Guizilini, and Valdir Grassi. Sparse-to-continuous: Enhancing monocular depth estimation using occupancy maps. In *2019 19th International Conference on Advanced Robotics (ICAR)*, pages 793–800. IEEE, 2019. 56

[236] Chang Shu, Kun Yu, Zhixiang Duan, and Kuiyuan Yang. Feature-metric Loss for Self-supervised Learning of Depth and Egomotion. In *ECCV*, pages 572–588, 2020. 57

[237] Hanhan Li, Ariel Gordon, Hang Zhao, Vincent Casser, and Anelia Angelova. Unsupervised monocular depth learning in dynamic scenes. *arXiv preprint arXiv:2010.16404*, 2020. 57, 61

[238] Hailin Jin, Paolo Favaro, and Stefano Soatto. Real-time feature tracking and outlier rejection with changes in illumination. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 1, pages 684–689. IEEE, 2001. 60, 67

[239] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International journal of computer vision*, 56(3):221–255, 2004. 60, 67

[240] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. Unidepth: Universal monocular metric depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10106–10116, 2024. 72

[241] Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9043–9053, 2023. 72

[242] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 72

[243] Honghui Yang, Di Huang, Wei Yin, Chunhua Shen, Haifeng Liu, Xiaofei He, Binbin Lin, Wanli Ouyang, and Tong He. Depth any video with scalable synthetic data. *arXiv preprint arXiv:2410.10815*, 2024. 72

[244] Bo Li, Chunhua Shen, Yuchao Dai, Anton Van Den Hengel, and Mingyi He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1119–1127, 2015. 76

[245] Rui Peng, Ronggang Wang, Yawen Lai, Luyang Tang, and Yangang Cai. Excavating the potential capacity of self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15560–15569, 2021. 76

[246] Kaichen Zhou, Lanqing Hong, Changhao Chen, Hang Xu, Chaoqiang Ye, Qingyong Hu, and Zhenguo Li. Devnet: Self-supervised monocular depth learning via density volume construction. In *European Conference on Computer Vision*, pages 125–142. Springer, 2022. 76

[247] Ziyue Feng, Liang Yang, Longlong Jing, Haiyan Wang, YingLi Tian, and Bing Li. Disentangling object motion and occlusion for unsupervised multi-frame monocular depth. In *European Conference on Computer Vision*, pages 228–244. Springer, 2022. 76

[248] Yang Wang, Peng Wang, Zhenheng Yang, Chenxu Luo, Yi Yang, and Wei Xu. Unos: Unified unsupervised optical-flow and stereo-depth estimation by watching videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8071–8081, 2019. 76

[249] Aashish Sharma, Loong-Fah Cheong, Lionel Heng, and Robby T. Tan. Nighttime stereo depth estimation using joint translation-stereo learning: Light effects and uninformative regions. In *2020 International Conference on 3D Vision (3DV)*, pages 23–31, 2020. 76, 79, 82, 85, 88, 89, 90, 95, 96

[250] Kieran Saunders, George Vogiatzis, and Luis Manso. Self-supervised monocular depth estimation: Let's talk about the weather. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 78

[251] Stefano Gasperini, Nils Morbitzer, HyunJun Jung, Nassir Navab, and Federico Tombari. Robust monocular depth estimation under challenging conditions. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 78, 79

[252] Madhu Babu Vankadari, Sourav Garg, Anima Majumder, S. Kumar, and Ardhendu Behera. Unsupervised monocular depth estimation for night-time images using adversarial domain feature adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 78

[253] Chaoqiang Zhao, Yang Tang, and Qiyu Sun. Unsupervised monocular depth estimation in highly complex environments. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6:1237–1246, 2021. 78

[254] Lina Liu, Xibin Song, Mengmeng Wang, Yong Liu, and Liangjun Zhang. Self-supervised monocular depth estimation for all day images using domain separation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 78

[255] J. Spencer, R. Bowden, and S. Hadfield. Defeat-net: General monocular depth via simultaneous unsupervised representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14390–14401, Los Alamitos, CA, USA, Jun 2020. IEEE Computer Society. 78

[256] Kun Wang, Zhenyu Zhang, Zhiqiang Yan, Xiang Li, Baobei Xu, Jun Li, and Jian Yang. Regularizing nighttime weirdness: Efficient self-supervised monocular depth estimation in the dark. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 78

[257] Yupeng Zheng, Chengliang Zhong, Pengfei Li, Huan ang Gao, Yuhang Zheng, Bu Jin, Ling Wang, Hao Zhao, Guyue Zhou, Qichao Zhang, and Dongbin Zhao. Steps: Joint self-supervised nighttime image enhancement and depth estimation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 78

[258] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos, and A. Gaidon. 3d packing for self-supervised monocular depth estimation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2482–2491, Los Alamitos, CA, USA, Jun 2020. IEEE Computer Society. 79

[259] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Spreading vectors for similarity search. *arXiv preprint arXiv:1806.03198*, 2018. 79, 84, 104, 105, 107

[260] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015. 80

[261] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 84

[262] C. Godard, O. Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6602–6611, Los Alamitos, CA, USA, Jul 2017. IEEE Computer Society. 84

[263] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA, 2015. 84

[264] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016. 85

[265] Stephanie Fu, Mark Hamilton, Laura Brandt, Axel Feldman, Zhoutong Zhang, and William T Freeman. Featup: A model-agnostic framework for features at any resolution. *arXiv preprint arXiv:2403.10516*, 2024. 94, 97, 117

[266] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002. 100

[267] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image matching across wide baselines: From paper to practice. *International Journal of Computer Vision*, 129(2):517–547, 2021. 100, 103

[268] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE international conference on computer vision*, pages 3456–3465, 2017. 100

[269] Qianqian Wang, Xiaowei Zhou, Bharath Hariharan, and Noah Snavely. Learning feature descriptors using camera pose supervision. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 757–774. Springer, 2020. 100

[270] Timor Kadir and Michael Brady. Saliency, scale and image description. *International Journal of Computer Vision*, 45:83–105, 2001. 101, 105, 107

[271] Bernt Schiele and James L Crowley. Recognition without correspondence using multidimensional receptive field histograms. *International Journal of Computer Vision*, 36:31–50, 2000. 101, 107

[272] Jan Beirlant, Edward J Dudewicz, László Györfi, Edward C Van der Meulen, et al. Nonparametric entropy estimation: An overview. *International Journal of Mathematical and Statistical Sciences*, 6(1):17–39, 1997. 101, 105, 107

[273] A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948. 105

[274] Vassileios Balntas, Lilian Tang, and Krystian Mikolajczyk. Bold-binary online learned descriptor for efficient image matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2367–2375, 2015. 105, 107

[275] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *International journal of computer vision*, 60(1):63–86, 2004. 105

[276] Changwei Wang, Rongtao Xu, Ke Lu, Shibiao Xu, Weiliang Meng, Yuyang Zhang, Bin Fan, and Xiaopeng Zhang. Attention weighted local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):10632–10649, 2023. 106

[277] Tianxin Shi, Shuhan Shen, Xiang Gao, and Lingjie Zhu. Visual localization using sparse semantic 3d map. In *2019 IEEE international conference on image processing (ICIP)*, pages 315–319. IEEE, 2019. 107

[278] Fei Xue, Ignas Budvytis, and Roberto Cipolla. Sfd2: Semantic-guided feature detection and description. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5206–5216, 2023. 107

[279] Bernt Schiele and James L Crowley. Probabilistic object recognition using multidimensional receptive field histograms. In *Proceedings of 13th International Conference on Pattern Recognition*, volume 2, pages 50–54. IEEE, 1996. 107

[280] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. 2024. 108

[281] Sebastian Raschka, Joshua Patterson, and Corey Nolet. Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. *arXiv preprint arXiv:2002.04803*, 2020. 108

[282] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017. 109

[283] Madhu Vankadari, Stuart Golodetz, Sourav Garg, Sangyun Shin, Andrew Markham, and Niki Trigoni. When the sun goes down: Repairing photometric losses for all-day depth estimation. In *Conference on Robot Learning*, pages 1992–2003. PMLR, 2023. 114

[284] Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. Magsac++, a fast, reliable and accurate robust estimator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1304–1312, 2020. 116