Robust 2D and 3D Registration with Deep Neural Networks



Zirui Wang
Linacre College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy
Michaelmas 2024

For my family, my inspiration and strength.

Abstract

Recovering 3D geometry is a crutial task in computer vision, essential for accurate world reconstruction and perception. Modern applications in AR, VR, autonomous driving, and medical imaging rely heavily on 3D and 4D reconstruction techniques. This thesis aims to enhance registration methods, which play a key role in reconstruction, by fusing classical multi-view geometry and deep neural networks. We explore this theme in three primary directions, each distinguished by registration dimensionality: 3D – 3D, 3D – 2D, and 2D – 2D.

First, we focus on improving the alignment of 3D point clouds in both rigid and non-rigid scenarios. In non-rigid 3D registration, traditional methods directly optimise a motion field between a source and target surface. This often leads to slow convergence and being trapped to local minimums. We introduce a neural network-based scene flow to initialise the optimisation, providing a more efficient and robust solution. Additionally, we present a novel surface normal estimation technique that aids both rigid and non-rigid registration. Unlike conventional methods that use a fixed global neighbour parameter, our approach employs a self-attention mechanism to adapt to local geometry variations.

Second, we address the challenge of registering 2D images to 3D Neural Radiance Fields (NeRF) through joint optimisation of NeRF and camera parameters. Original NeRF training mandates pre-processed camera parameters, creating a bottleneck in the workflow. Our approach allows for end-to-end camera parameter estimation during NeRF training while reusing the existing photometric loss in NeRF. We further extend this to account for larger camera movements by incorporating a monocular depth prior.

Lastly, we propose a method for interest point discovery, which is beneficial for 2D image registration. Unlike existing interest point identification methods that suffer from significant viewpoint changes and occlusion boundaries, we propose a multi-view interest point discovery approach to address these limitations. Our method is trained in a self-supervised fashion with pure-geometric constraints that encourage point identification repeatability, sparsity, and multi-view consistency.

In summary, this thesis explores the fusion of traditional multi-view geometry concepts with deep learning priors in various registration tasks, including point cloud registration, image-to-NeRF registration, and image-to-image registration.

Keywords – 3D Registration, 3D Reconstruction, Multi-view Geometry, NeRF, Point Cloud, Interest Point Discovery.

Declaration

This thesis is submitted to the Department of Engineering Science, University of Oxford, in fulfilment of the requirements for the degree of Doctor of Philosophy. This thesis is entirely my own work, and except where otherwise stated, describes my own research.

Zirui Wang, December 2023.

Acknowledgement

The most exciting part of a DPhil is discovering new things and seeing novel ideas succeed. But such moments are rare, and much of the journey involves working alone and continuous dedication. I've been lucky to have strong support and friendships throughout this journey.

First, I express my deepest gratitude to my respected supervisors, Professor Min Chen and Professor Victor Adrian Prisacariu. Their guidance and wisdom have been invaluable and will serve me well in the years to come.

Second, special thanks go to Shuda Li, a friend from before my DPhil journey. After completing my master's degree in Bristol with a passion for SLAM, Shuda provided patient and invaluable guidance. I wouldn't be on this DPhil path without his support.

Third, my thanks to my collaborators in the Active Vision Lab, including Shuda Li, Wenjing Bian, Shuai Chen, Xinghui Li, Kejie Li, and Henry Howard-Jenkins. Working on numerous interesting projects has been rewarding. Additional thanks to Shangzhe Wu and Weidi Xie of the Visual Geometry Group for insightful discussions and support. I'm also grateful to my mentors, Yipu Zhao and Jakob Engel at Meta Reality Lab, for their invaluable assistance during and after my internship.

A warm thank you goes to my friends Shangzhe Wu, Weidi Xie, Tengda Han, Yuqing Tang, Yang Zhang, Yizhe Wu, Xinghui Li, Yiwen Li, Michael Hobley, Gratianus Wesley Putra Data, Marcelo Gennari do Nascimento, and Theo Costain. We have enjoyed countless memorable moments, from sharing meals to playing football and frisbee. An additional recognition goes to Michael, who dedicated significant effort to help me polish this thesis. Furthermore, a special thanks go to long-time friends Peijie Li and Kenan Yang, who have always been there to listen to my many complaints.

Lastly, my deepest appreciation goes to my family and Xinxia for their consistent love and encouragement throughout my life. I'll be forever thankful for their support.

While this list is far from complete, the experiences and people mentioned have profoundly enriched my DPhil journey. Thank you all.

Contents

1 Introduction **6**
- 1.1 Key Contributions . . . 8
  - 1.1.1 3D – 3D Registration in Dynamic Scenes . . . 8
  - 1.1.2 3D – 2D Registration with NeRF . . . 9
  - 1.1.3 2D – 2D Registration with Interest Point Discovery . . . 11
- 1.2 Thesis Outline . . . 11
- 1.3 List of Publications . . . 12

2 Literature Review **15**
- 2.1 3D Scene Representations . . . 15
  - 2.1.1 Point Cloud . . . 16
  - 2.1.2 Level Set . . . 17
  - 2.1.3 NeRF . . . 17
- 2.2 3D – 3D Registration . . . 18
  - 2.2.1 Registration in Static Scenes . . . 18
  - 2.2.2 Registration in Dynamic Scenes . . . 19
  - 2.2.3 Point Cloud Normal Estimation . . . 20
- 2.3 3D – 2D Registration . . . 21
- 2.4 2D – 2D Registration . . . 22

3 3D – 3D Registration in Dynamic Scenes **24**
- 3.1 FlowNet3D++: Geometric Losses For Deep Scene Flow Estimation . . . 25
- 3.2 Neighbourhood-Insensitive Point Cloud Normal Estimation Network . . . 40

4 3D – 2D Registration with NeRF **59**
- 4.1 NeRF−−: Neural Radiance Fields Without Known Camera Parameters 60
- 4.2 NoPe-NeRF: Optimising Neural Radiance Field with No Pose Prior . . . 80

5 2D – 2D Registration with Interest Point Discovery **98**

6 Conclusion **113**
- 6.1 Achievement . . . 114
- 6.2 Future Work . . . 115

Chapter 1

Introduction

Bridging the gap between our physical reality and the digital world, 3D modelling plays a pivotal role in advancing digital and automated technologies. This demand arises from two key areas. Firstly, precise 3D modelling is vital for automating complex tasks within three-dimensional environments, evident in applications such as object detection, environmental inspections, and autonomous driving. Secondly, it satisfies our inherent need to interact with digital content as intuitively as we do in the real world, impacting industries like telecommunications, education, entertainment, and healthcare. With the growing accessibility of computational resources and advancements in algorithms, 3D modelling has evolved to be more practical and accurate than ever before.

One way to model the 3D world is reconstructing 3D structures from 2D images. Depending on whether the scene is static or dynamic, this process is termed as 3D or 4D reconstruction, respectively. Numerous systems have been developed for these purposes. For static scenes, example systems include Simultaneous Localisation and Mapping (SLAM) [1, 2, 3, 4, 5, 6, 7], Structure-from-Motion (SfM) [8, 9, 10], Multi-view Stereo (MVS) [11, 12, 13], and Neural Radiance Field (NeRF) [14, 15, 16, 17]. For dynamic scenes, example systems include 4D recon-

struction [18, 19, 20] and non-rigid SfM systems [21, 22, 23].

This thesis focuses on a critical aspect of both 3D and 4D reconstruction: **registration**. From a dimensionality perspective, we classify the registration task into three categories: i) 3D – 3D, ii) 3D – 2D, and iii) 2D – 2D. In the 3D – 3D context, classical algorithms like Iterative Closest Point (ICP) are used for registering point clouds [24, 25, 26, 27], and level set evolution methods [28, 29, 30] are employed for aligning implicit surfaces. In the 3D – 2D domain, Perspective-n-Points (PnP) methods [31, 32, 33] and photometric-based approaches [34, 35] are used to register a 3D model with a 2D image. In 2D – 2D direction, rigid registration is often achieved using homography and fundamental matrices [36, 37, 38, 39, 40], while non-rigid registration utilises optical flow methods [41, 42].

While classical registration algorithms are known for their lightweight nature and interpretability they tend to focus on local structures and textures, which often leads to local minima and slower convergence in the optimisation process. In contrast, recent developments in deep learning have introduced data-driven alternatives [43, 44, 45, 46, 47]. By leveraging high-level understanding, these method offers superior results, but they can sometimes make elementary errors due to a lack of geometric grounding. Therefore, it is crucial to balance both high-level and low-level understanding. For instance, high-level information from data-driven approaches, such as categorical or part-based priors, can serve as coarse registration guidance. Conversely, low-level or piece-wise geometric consistency can refine these initial results.

Hence, the goal of this thesis is to study how to combine strengths from both classical and learning-based techniques and develop robust and accurate registration methods. Specifically, Chapter 3 presents our study in 3D – 3D registration: where i) Section 3.1 discusses how classical point-to-plane ICP registration can enhance learning-based scene flow estimation, and how learning-based scene flow

estimation is beneficial to complex non-rigid registration, and ii) Section 3.2 introduces a novel point cloud normal estimation method that takes advantage of self-attention [48] to weight points softly in this task. This learning-based method aids both rigid and non-rigid registration.

Chapter 4 explores 3D – 2D registration between a Neural Radiance Field (NeRF) and 2D images, in which Section 4.1 proposes a joint camera parameter estimation and NeRF training pipeline, removing camera pre-processing overhead, and Section 4.2 shows how adding a monocular depth network improves handling of large camera movements.

Lastly, Chapter 5 proposes an interest point discover network. We demonstrate that introducing geometric constraints to interest point discovery network enables consistent and repeatable interest point predictions under large view angle changes and occlusions. This further leads to better 2D – 2D registration performance in term of related pose estimation.

This chapter is structured as follows. We start with Section 1.1 that expands the key contributions of this thesis, followed by a thesis outline in Section 1.2, and end with a publication list in Section 1.3.

1.1 Key Contributions

This thesis focuses on enhancing both 2D and 3D registration techniques through the fusion of deep neural networks and multi-view geometry concepts. Specifically, our research advances the registration field in three principal directions.

1.1.1 3D – 3D Registration in Dynamic Scenes

Traditional methods for dynamic and non-rigid registration [49, 50, 29, 30] often suffer from slow optimisation and prone to local minima. These methods typically

operate well only when the motion between source and target is small.

Our research tackles these limitations from two perspectives. In Section 3.1, we introduce a motion prior in the form of scene flow to guide the optimisation-based registration process. This coarse registration serves as a useful initial step for a second stage non-linear optimisation. The scene flow is estimated using a neural network trained solely on synthetic data, and we enhance its performance by incorporating an Iterative Closest Point (ICP) [25] loss.

Normal estimation in point clouds plays a crucial role in surface alignment loss metrics such as ICP. Existing methods like Principal Component Analysis (PCA) [51], Nesti-Net [52], and PCPNet [53] are sensitive to the selection of the neighbourhood range. For instance, flat regions benefit from larger surface neighbourhoods which reduce local noise, while complex regions require smaller neighbourhoods to prevent blurring. In Section 3.2, we introduce a self-attention-based method for normal estimation that is insensitive to neighbourhood selection. This approach soft-focuses on relevant neighbour points and substantially mitigates the sensitivity issue. The method is advantageous for both rigid and non-rigid 3D registration.

In summary, Sections 3.1 and 3.2 discuss our contributions to enhancing 3D registration through point cloud scene flow and normal estimation networks.

1.1.2 3D – 2D Registration with NeRF

Neural Radiance Fields (NeRF) methods [14] offer photo-realistic novel view synthesis by optimising a radiance and a density volume from multi-view images using a photometric loss function. However, training NeRF requires pre-computed camera parameters, typically obtained from Structure from Motion (SfM) tools like COLMAP [9]. This step can be computationally intensive when dealing with numerous images, adding significant overhead to NeRF optimisation.

In this line of research, we focus on the simultaneous training of NeRF and camera parameter estimation, aiming to alleviate this pre-processing burden. Given that NeRF operates in a 3D volumetric space and the input images reside in a 2D domain, we frame this as a 3D-to-2D registration problem.

Our work offers two key contributions. First, in Section 4.1, we introduce NeRF$--$, a straightforward yet effective method for jointly optimising NeRF and camera parameters, both extrinsic and intrinsic, for forward-facing scenes. We achieve this by simply considering the camera parameters as trainable parameters and applying the same photometric loss used in the original NeRF framework. Given that ray sampling in NeRF depends on camera parameters, the rendering loss can be back-propagated to these parameters during NeRF training. Our approach introduces little overhead and maintains comparable rendering performance to a NeRF model trained with pre-computed cameras.

Second, in Section 4.2, we extend NeRF$--$ with NoPe-NeRF, a model that incorporates monocular depth measurements for joint NeRF and camera parameter optimisation. Specifically, we learn to undistort monocular depth for each input image during joint optimisation, and constrain the optimisation process with three factors: i) a point cloud registration constraint through chamfer distance, ii) a depth-based surface rendering loss, and iii) a depth consistency loss between rendered depth and monocular depth. As a result, NoPe-NeRF can accommodate more complex camera movements.

In summary, our contributions in Sections 4.1 and 4.2 present a novel 3D-to-2D framework that jointly optimises camera parameters and monocular depth distortions during NeRF training.

1.1.3 2D – 2D Registration with Interest Point Discovery

The goal of this work is to enhance the process of interest point identification, a crucial step in affine transformation estimation for image registration. Currently, local keypoints based on low-level image features are the norm. These are detected using either traditional algorithms [54, 55, 56] or neural networks [57, 58, 59, 60]. However, a significant limitation of this approach is that these low-level features are predominantly surface-based, limiting their detectability across various viewing angles, distances, and lighting conditions. This limitation has led to the development of interest points at a higher, more abstract level, especially in specific categories like human and hand imagery [61, 62, 63, 64]. Yet, extending this abstraction to a broader range of categories remains a challenge.

In response, Chapter 5 proposes a concept of Abstract Interest Point (AIP), denoting an abstract-level keypoint that is repeatable across diverse viewing conditions and applicable to a wide range of scenes. To train a neural network to discover AIPs, we design a geometric-based self-supervised training strategy, including a contrastive term and a triangulation term. Our pipeline avoids relying heavily on local surface texture, instead focusing on broader scene structures, similar to approaches in human keypoint discovery literature but without assumptions on scene content.

To sum up, our contribution is enhancing image-to-image registration by introducing a multi-view framework for interest point discovery.

1.2 Thesis Outline

This integrated thesis is structured such that Sections 3.1, 3.2, 4.1 and 4.2 and Chapter 5 each present a research paper. Additionally, in Chapter 2 we offer an overall literature review of current developments in related fields, and in Chapter 6 we

summarise our contributions, evaluate their impact, and outline possible avenues for future research.

For a coherent presentation of the core concepts, the research papers are organised into three chapters, mirroring Sections 1.1.1 to 1.1.3 outlined above. In **Chapter 3**, we delve into the notion of 3D – 3D registration in dynamic scenes, encapsulating Sections 3.1 and 3.2. **Chapter 4** explores the concept of 3D – 2D registration with NeRF, incorporating Sections 4.1 and 4.2. Lastly, **Chapter 5** discusses 2D – 2D registration with interest point discovery.

1.3 List of Publications

Section 3.1: "FlowNet3D++: Geometric Losses For Deep Scene Flow Estimation". **Zirui Wang**, Shuda Li, Henry Howard-Jenkins, Victor Adrian Prisacariu, Min Chen. In *Winter Conference on Applications of Computer Vision (WACV)*, 2020.

Section 3.2: "Neighbourhood-Insensitive Point Cloud Normal Estimation Network". **Zirui Wang**, Victor Adrian Prisacariu. In *British Machine Vision Virtual Conference (BMVC)*, 2020.

Section 4.1: "NeRF−−: Neural Radiance Fields Without Known Camera Parameters". **Zirui Wang**, Shangzhe Wu, Weidi Xie, Min Chen, Victor Adrian Prisacariu. In *arXiv preprint arXiv:2102.07064*, 2021.

Section 4.2: "NoPe-NeRF: Optimising Neural Radiance Field with No Pose Prior". Wenjing Bian, **Zirui Wang**, Kejie Li, Jia-Wang Bian, Victor Adrian Prisacariu. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Chapter 5: "AIP: Abstract Interest Point Discovery from Multi-view Self-supervision". **Zirui Wang***, Yipu Zhao*, Tanner Schmidt, Tomasz Mal-

isiewicz, Jakob Engel. *Technical Report*, 2023. Under peer review. *Equal Contribution.

Publications Not Included

During my DPhil I also contributed to the following publications. We do not include them since they are only loosely connected to this thesis.

⇒ **"Ray-ONet: Efficient 3D Reconstruction From A Single RGB Image".** Wenjing Bian, **Zirui Wang**, Kejie Li, Victor Prisacariu. In *British Machine Vision Conference (BMVC)*, 2021.

Abstract: Ray-ONet enhances monocular 3D modelling by predicting 3D occupancies along a ray with one query based on image feature, offering greater accuracy and faster reconstruction compared to Occupancy Networks [65].

⇒ **"Direct-PoseNet: Absolute Pose Regression with Photometric Consistency".** Shuai Chen, **Zirui Wang**, Victor Prisacariu. In *International Conference on 3D Vision (3DV)*, 2021

Abstract: Direct-PoseNet is a relocalisation pipeline, combing an absolute pose regression (APR) network with a view synthesis-based direct matching module, and achieving top accuracy on single-image APR benchmarks.

⇒ **"DFNet: Enhance Absolute Pose Regression with Direct Feature Matching".** Shuai Chen, Xinghui Li, **Zirui Wang**, Victor Adrian Prisacariu. In *European Conference on Computer Vision (ECCV)*, 2022.

Abstract: DFNet further extends the previous work Direct-PoseNet with an exposure-adaptive view synthesis, improving APR accuracy in challenging outdoor setups.

⇒ **"NeFeS: Refinement for Absolute Pose Regression with Neural Feature Synthesis".** Shuai Chen, Yash Bhalgat, Xinghui Li, Jiawang Bian, Kejie Li, **Zirui Wang**, Victor Adrian Prisacariu. In *arXiv preprint 2303.10087*, 2023.

Abstract: Evolving from DFNet, NeFeS directly renders feature maps during novel view synthesis, leading to more robust and faster APR.

Chapter 2

Literature Review

This literature review serves as a high-level overview of work relevant to the theme of registration. We initially examine common 3D representations in Section 2.1, followed by a discussion of both 2D and 3D registration techniques in Sections 2.2 to 2.4. As Sections 3.1, 3.2, 4.1 and 4.2 and Chapter 5 each correspond to an individual research paper, readers are directed to these chapters for more in-depth literature reviews in each respective field.

2.1 3D Scene Representations

The field of 3D shape representations has evolved substantially since its beginning in the 1960s and remains a vibrant area of research. Early works started from techniques like wire-frame models [66], Constructive Solid Geometry (CSG) [67], and parametric surfaces [68], driven largely by applications in computer-aided design. Subsequently, point clouds [69] and polygonal meshes [70] were introduced to accommodate more complex shapes. Implicit surfaces and level sets [71] were developed to naturally handle changes in topology. The Surfel representation [72] extended point clouds to provide richer surface details without the complexity of

mesh affinity.

Recent advancements in deep learning have given rise to neural-based representations, such as the Occupancy Network (OcNet) [65], Neural Radiance Field (NeRF) [14], and Multi-plane Images (MPI) [73]. These representations offer several advantages including ease of differentiation during training, high-quality view synthesis, and efficient storage.

In the context of this thesis, level sets, point clouds, and NeRF are especially relevant. Section 3.1 delves into point cloud scene flow networks, demonstrating improvements in dynamic reconstruction using level set representations. Section 3.2 introduces a network for point cloud normal estimation. Sections 4.1 and 4.2 build upon NeRF, and Section 4.2 additionally incorporates point clouds back-projected from monocular depth maps. Chapter 5 discusses a self-supervised interest point discovery method that relies on point clouds for data preparation and triangulation during training. A general review of these representations is provided in the subsequent sections, with more targeted domain-specific reviews available in each specific section.

2.1.1 Point Cloud

A point cloud is a collection of data points defined within a coordinate system. Although point clouds are versatile in 3D computer vision field, this thesis focuses on their use as a simple and efficient representation of surfaces. Common ways to obtain point cloud include a) gathering raw data from depth sensors such as LiDAR, radar, depth cameras, ultrasonic sensors; b) extracting data from other representations such as meshes, distance functions; and c) recovering data through image-based algorithms such as Structure from Motion (SfM) [8, 9, 10] and Simultaneous Localisation and Mapping (SLAM) [1, 2, 3, 4, 5, 6, 7]. With point clouds, it is possible to perform and conduct various tasks, for example, relative pose es-

timation [24, 25, 74, 75, 76], shape classification and segmentation [77, 78, 79], object tracking [80, 81], and scene flow estimation [43, 82, 83]. We refer readers to [84] for a more comprehensive survey on point clouds and their applications.

2.1.2 Level Set

A level set is a collection of points in a space where a particular function takes on a specific value. In simpler terms, it's like drawing a line or a surface where a function remains constant. For example, on a weather map, a line showing all the locations with the same temperature is a level set. These sets help visualise and analyse the behaviour of functions in various fields like mathematics, physics, and engineering. In this thesis we narrow the scope to its applications in implicit 3D surface representation [71], which offers more flexibility during 3D and 4D reconstruction. A commonly utilised level set function for this purpose is the Signed Distance Function (SDF), which assigns positive distance value to points located between the surface and the observer, and negative distance value to those behind the surface. Subsequent advancements introduced Truncated SDF (TSDF) [85] and voxel hashing [86], enabling more efficient storage and faster computational operations. Level sets are also closely related to recently developed neural representations such as occupancy network[20, 87] and NeRF [14, 88], which all share similar concept of occupancy fields and density fields. We direct readers to [89] for a more complete review on level sets and their applications.

2.1.3 NeRF

Neural Radiance Field (NeRF) [14] is a neural volumetric representation that utilises a Multi-layer Perceptron (MLP) to map 3D positions and viewing angles to volume density and radiance. This allows for the generation of photo-realistic

novel views based on a set of input images and corresponding camera parameters. Numerous enhancements have been made to NeRF, including accelerated training with explicit voxel structures [90, 91, 15], volume decomposition [92, 93], Gaussian splatting [16]. To eliminate per-scene optimisation, generalisable NeRF methods [94, 95, 96, 97] have been proposed. Further extensions include, improving surface reconstruction accuracy [88, 98, 99], dynamic reconstruction [100, 101, 102], camera parameter estimation [103, 104, 105, 106, 107, 108], and integration with large language models [109, 110, 111, 112]. NeRF remains a vibrant and rapidly advancing field. For the latest updates and comprehensive insights, readers are advised to consult a recent survey [113].

2.2 3D – 3D Registration

Accurately aligning a source and target surface is a crutial step in 3D and 4D reconstruction. In this thesis, Section 3.1 introduces a scene flow estimation method that provides a motion prior to non-rigid 3D registration in dynamic environments. Additionally, Section 3.2 presents a surface normal estimation technique that enhances general 3D surface registration. Here, we present a general review of three related topics in 3D to 3D registration: static registration, dynamic registration, and surface normal estimation, whereas Sections 3.1 and 3.2 contain more detailed paper specific literature review.

2.2.1 Registration in Static Scenes

Registration in static scenes primarily involves pose estimation between two surfaces, where one surface is a rigidly transformed version of the other. The objective is to estimate a rigid transformation for aligning the surfaces.

Numerous methods exist in the literature, depending on the type of 3D rep-

resentations. Examples include registration between two level sets [114, 115, 28], between level sets and point clouds [116, 117], and between two point clouds [24, 25, 74, 75, 76]. Recent advances also cover neural representations [118, 119].

In this thesis, Section 3.2 introduces a novel normal estimation method that aids surface-based registration. Meanwhile, Section 4.2 utilises a Chamfer Distance [120] based loss function for point cloud registration. Therefore, we narrow our focus to 3D registration between two point clouds.

Methods for registration are commonly categorised based on whether correspondence between surfaces is pre-established. First, correspondence-based methods [74, 75, 121] directly compute a rigid transformation when the correspondence is known. Second, when the correspondence is not specified, Iterative Closest Point (ICP) methods are widely used for local alignment. For instance, point-to-point ICP [24] minimises the distance between the closest points on the two surfaces to estimate rotation and translation. Point-to-plane ICP [25] stabilises the optimisation using point-to-plane distance metrics. Coloured ICP [27] incorporates photometric information. In the era of deep learning, approaches like PointNetLK [44] minimise feature distances in ICP formulations, while DCP [45] leverages transformer architectures [48] to register point sets.

2.2.2 Registration in Dynamic Scenes

Dynamic scene registration entails non-rigid alignment between source and target surfaces. Two primary categories exist: intrinsic methods, which work in transformed domains, and extrinsic methods that function in 3D space $\mathbb{R}^3$. This thesis emphasises the extrinsic approach. A more comprehensive survey can be found in [122].

The essence of non-rigid registration is estimating a displacement field linking source and target surfaces. Simple methods employ scene flow, depicted as 3D

translation vectors [123]. Affine fields [124] extend simple scene flow, capturing local deformations like rotation and scaling. These methods, however, need to model motion for each point individually, overlooking spatial correlations among neighbouring points and leading to expensive computation cost.

To introduce spatial smoothness and improve efficiency, Deformation Graphs [125] use a sparse set of nodes to model the displacement field of a surface. Each node has an affine transformation attached that transforms a local region and is linked to neighbour nodes, enabling local smoothness enforcement. Despite its efficiency, defining and maintaining the graph is challenging.

Recently, the integration of neural networks into the non-rigid 3D registration domain has seen notable progress, particularly in areas like scene flow estimation. This includes advancements in point clouds [43, 126, 127, 128], occupancy fields [20], and NeRFs [101, 102]. In Section 3.1, we explore how scene flow estimation networks and conventional surface registration techniques can be mutually beneficial. For instance, using network-estimated scene flow as an initial step in dynamic reconstruction tasks can enhance registration convergence. Conversely, incorporating a point-to-plane Iterative Closest Point (ICP) registration objective during scene flow network training can significantly improve the training efficiency and scene flow prediction accuracy.

2.2.3 Point Cloud Normal Estimation

Normal estimation is crucial for tasks like surface reconstruction [129, 130], rendering [131], and registration [25]. Classical methods like PCA [51] employ eigen decomposition on the covariance matrix of a local point cloud patch to find the normal vector. Jet [132] uses a truncated Taylor expansion to estimate both the surface normal and principal curvatures.

Recent learning-based methods like PCPNet [53] use PointNet [77] on a local

patch to directly regress a surface normal. Nesti-Net [52] employs a Mixture of Experts (MoE) system [133, 134] that processes various neighbour scales around a target point, with each expert specialising in different geometric properties.

Although these learning-based approaches outperform PCA and Jet, they require larger models and longer inference times. They also limit information flow due to their manual handling of multiple scales. These shortcomings are addressed in Section 3.2 via a new attention-based lightweight normal estimation technique.

2.3 3D – 2D Registration

The task of aligning a reconstructed 3D model with a 2D image is known as 3D – 2D registration. The Perspective-n-Point (PnP) algorithm [31, 32, 33, 135] is a conventional method in this area, mapping a 3D point cloud to corresponding 2D points on an image plane. However, PnP is dependent on estimated correspondences, often obtained through feature matching, which can be both slow and prone to errors.

An alternative approach is intensity-based registration, which registers two entities using photometric information, such as RGB, gray-scale, or object silhouettes. This line of work is initially developed for medical applications such as image-guided therapy (IGT) [34, 136, 137, 138], which aligns 3D volumes such as computed tomography (CT) and magnetic resonance imaging (MRI) volumes and 2D scans such as X-ray. Similar idea has also been incorporated into Simultaneous Localisation and Mapping (SLAM) systems such as DTAM [35], LSD-SLAM [5], and DSO [139].

With the advent of neural representations like Neural Radiance Fields (NeRF), which require costly pre-processing of camera parameters, there has been a growing focus on reducing this computational burden. Section 4.1 introduces our

approach that jointly optimises a NeRF model and camera parameters, minimising the overhead while maintaining high-quality novel view synthesis. Section 4.2 extends this work by incorporating monocular depth to accommodate larger camera movements. Evolving from our research, notable contributions in this domain include SPARF [107], which leverages dense correspondence from optical flow networks to estimate poses in sparse views pose scenarios, and RoDynRF [140], which focuses on optimising camera parameters in dynamic NeRF setups. To further facilitate large scale 3D understanding, there is a growing interest in replacing per-scene pose optimisation with generalisable pose estimation pipelines [141, 142, 143, 144, 145, 146], and with pose-free pipelines, for instance, LEAP [147].

Additional methods in 3D – 2D registration include absolute pose regression [47, 148, 149] and scene coordinate regression [150, 151, 152]. In this thesis we work closely with NeRF-based registration, and we refer readers to [138, 153] for more detailed review in 3D – 2D registration domain.

2.4 2D – 2D Registration

Registering two images involves estimating a transformation that aligns one image with another. This field can be categorised based on the type of transformation involved: either rigid or non-rigid.

In the context of non-rigid transformations, the problem is often framed as motion estimation. Various techniques have been proposed for this, including optical flow [41, 42, 154, 155, 156], diffusion process [157], level sets evolution [158, 159], and correspondence flow [160, 161, 162, 163, 164].

Conversely, estimating rigid transformations between two images is equivalent to estimating epipolar relationships. This is typically parameterised using

either fundamental matrices or homography transformations [36, 37, 38, 39, 40]. A common pipeline for these estimations includes interest point detection, feature extraction, feature matching, and ultimately solving a set of linear equations. In Chapter 5, we introduce a method for discovering highly repeatable interest points from multi-view inputs, contributing to the initial stage of image-to-image registration. We direct readers to extensive reviews [165, 153, 166] for a full landscape in this topic.

Chapter 3

3D – 3D Registration in Dynamic Scenes

This chapter concerns non-rigid 3D registration, a key component in dynamic reconstruction systems. Considering two surfaces, a source surface and a deformed target surface, we aim to recover a non-rigid transformation between these two surfaces. Classical methods in this area mainly include level set evolution [28, 29, 30], non-rigid ICP [49, 50, 19], and scene flow estimation [167, 168, 29, 30]. However, due to the inherent complexity of non-rigid registration, which involves a high degree of freedom in optimisation, the methods mentioned above can only handle only limited motion between two surfaces.

We address this issue from two perspectives. First, in Section 3.1, we propose to initiate classical non-rigid registration with a scene flow initial solution. The initial solution is estimated from a point cloud neural network. We additionally enhance the scene flow estimation network by introducing several geometry constraints during training. Second, in Section 3.2, we contribute to the 3D registration task, including both rigid and non-rigid, by introducing a novel point cloud normal estimation method. Our method consists of a self-attention based network that

weights each point softly during normal estimation, resulting in more accurate normal estimation and faster surface registration.

In summary, we introduce three contributions in non-rigid registration. First, we propose a two-stage non-rigid reconstruction pipeline, where the first stage provides a initial alignment through a point cloud scene flow estimation network, and the second stage conducts a classical optimisation-based refinement. Second, we enhance existing scene flow network generalisation capability by introducing a self-supervision signal from classical ICP. Third, we introduce a novel point cloud normal estimation method, further contributing to non-rigid 3D registration task.

3.1 FlowNet3D++: Geometric Losses For Deep Scene Flow Estimation

In this section, we present our research paper on scene flow estimation. Our work builds on FlowNet3D [43], the pioneering point cloud scene flow estimation network introduced in 2019. FlowNet3D estimates 3D scene flows, describing the movement of each point between two point clouds. It works by first using a PointNet-style encoder to down-sample and extract local features from both point clouds. These features are then aggregated via a neural network to determine the implicit correspondence between the point clouds. Finally, the network up-samples and decodes this correspondence into scene flow vectors. FlowNet3D is trained with L2 loss on the FlyingThing3D [169] synthetic dataset using ground truth scene flows.

While this pipeline demonstrates the feasibility of estimating scene flow between two point clouds using a neural network, we enhance the existing FlowNet3D model by integrating geometric principles, specifically the point-to-plane distance and angular alignment. These geometric constraints not only improve the accu-

racy of scene flow estimation but also contribute to training stability and faster convergence.

In addition to enhancing scene flow estimation, we propose improving 3D-3D registration in dynamic scenes by incorporating our improved scene flow network into an established dynamic reconstruction pipeline [29]. We employ our network as an initial step, followed by further refinement stages. This two-stage approach notably elevates the accuracy of dynamic reconstruction.

In summary, our contributions to the field of non-rigid 3D registration are twofold. First, we introduce two additional loss terms to enhance the training of scene flow estimation networks. Second, we integrate our improved scene flow network into an established dynamic reconstruction pipeline, resulting in increased accuracy in dynamic reconstruction.

FlowNet3D++: Geometric Losses For Deep Scene Flow Estimation

Zirui Wang
ryan@robots.ox.ac.uk

Shuda Li
shuda@robots.ox.ac.uk

Henry Howard-Jenkins
henryhj@robots.ox.ac.uk

Victor Adrian Prisacariu
victor@robots.ox.ac.uk

Min Chen
min.chen@oerc.ox.ac.uk

Abstract

We present FlowNet3D++, a deep scene flow estimation network. Inspired by classical methods, FlowNet3D++ incorporates geometric constraints in the form of point-to-plane distance and angular alignment between individual vectors in the flow field, into FlowNet3D [21]. We demonstrate that the addition of these geometric loss terms improves the previous state-of-art FlowNet3D accuracy from ***57.85****% to* ***63.43****%. To further demonstrate the effectiveness of our geometric constraints, we propose a benchmark for flow estimation on the task of dynamic 3D reconstruction, thus providing a more holistic and practical measure of performance than the breakdown of individual metrics previously used to evaluate scene flow. This is made possible through the contribution of a novel pipeline to integrate point-based scene flow predictions into a global dense volume. FlowNet3D++ achieves up to a* ***15.0%*** *reduction in reconstruction error over FlowNet3D, and up to a* ***35.2%*** *improvement over KillingFusion [32] alone. We will release our scene flow estimation code later.*

1. Introduction

Scene flow is defined as a 3D vector field that provides a low level representation of 3D motion. It is analogous to optical flow, which describes the pixel movements on a 2D image plane. Optical flow can be considered the projection of scene flow into 2D. Applications such as object detection, object tracking, point cloud registration, correspondence estimation and motion capture can benefit from this low-level information for better performance.

Although the vector field representation is simple, scene flow estimation is far from an easy task. This is due to the requirement of accurate depth estimation and also the need to deal with occlusion. Traditionally, scene flow estimation is computed by optimising photometric error [21], or through matching hand-crafted features [3], each applied over multiple view geometry or RGB-D images. With the fast development of deep learning, some works bring CNN to scene flow estimation. This allows for the scene flow estimation to benefit from the semantic information and powerful deep feature extraction. Recently, PointNet [5] and PointNet++ [29] enabled the direct point cloud processing for deep learning. These works are particularly interesting since they are point cloud-based networks that can directly process the rich 3D geometric information, rather than implicitly learning 3D geometry from 2D images. Built on top of PointNet++, FlowNet3D [21] tackles the scene flow estimation problem on point cloud directly, achieving the state-of-art scene flow estimation results. Despite the impressive results of FlowNet3D, point cloud based scene flow estimation is still at the very beginning of its development. FlowNet3D trains the network with a naive supervision signal, which is the L_2 loss between predicted flow and ground truth vectors.

In this work, we apply geometric principles from classical point cloud registration algorithms in order to mature deep scene flow estimation beyond the simple L_2 norm between prediction and ground truth. In particular, we investigate two geometric constraints including: 1) point-to-plane distance and 2) the cosine distance between predicted flow vector and ground truth vector. The point-to-plane distance is a common loss term in Iterative Closest Point (ICP) [2, 26] algorithm, which is known for fast convergence. Cosine distance can penalise the angle between two vectors directly. As a result of this, cosine distance encourages correct alignment of our predicted scene flow vectors, not only that they lie on the L_2 norm surface. The application of these geometrically principled constraints not only demonstrates improved accuracy over the state-of-art, but also improved convergence speed and stability of training.

Further, we introduce a novel benchmark for investigating the practical performance of scene flow estimators, through the proxy task of dynamic 3D reconstruction. Application of scene flow to dynamic reconstruction provides a holistic combination of the individual metrics previously used to evaluate 3D flow estimation. We contribute a novel pipeline for integrating point-based scene flow into a global

dense volume.

To summarise, our main contributions are:

1. We improved the state-of-art point cloud-based scene flow estimation accuracy from 57.85% to 63.43% by combining point-to-plane loss, cosine distance loss with L_2 loss.

2. We propose an average angle error metric to evaluate the flow direction deviation to supplement End Point Error (EPE), which is not sufficient alone to evaluate the angle difference between two vectors.

3. We propose dynamic 3D reconstruction, alongside a novel dynamic integration pipeline, as a benchmarking task for scene flow estimation. 3D reconstruction provides a holistic and inherently geometric measure of flow estimation performance. Within our deformable scene flow benchmark task, our FlowNet3D++ achieves up to 15.0% less reconstruction error than FlowNet3D, and up to a 35.2% improvement over KillingFusion.

The remainder of the paper is organised as follows: in Section 2, we briefly describe related work. Section 3 describes our modifications to FlowNet3D, as well as our method to integrate the sparse scene flow vector field in dense dynamic reconstruction. In the experiment section, we evaluate the effect of adding different geometric constraints to scene flow estimation and our dynamic reconstruction results with several public datasets. Section 6 concludes our work and describes potential future work.

2. Related Work

2.1. Tradition Scene Flow Estimation

Scene flow is a low-level representation of 3D motion of points within a scene. It is a 3D extension from the 2D optical flow [3], which itself describes the pixel movements on a 2D image plane. Many works have focused on estimating scene flow using multi-view geometry [35] by associating salient image key points. Later works [27, 11, 37, 36] tackle this problem with joint variational optimisation of image registration and motion estimation. [37] compute dense scene flow from stereo cameras and achieved 5fps on a CPU. SphereFlow [10, 15] is the first real-time scene flow estimation system using RGB-D input. [16] proposed to process rigid and non-rigid segments differently.

2.2. Deep Flow Estimation

The recent development in deep neural networks provides an alternative to address the problem of associating points over deformed depth maps. One group of deep methods can be viewed as the successors of the classic 2D optical flow methods. For example, FlowNet [7] and its variants [12]. Instead of using hand-crafted feature for tracking pixel locations, these methods rely on learned deep features for tracking and then back-project into depth maps to fetch the 3D scene flow. For better training and evaluation, Mayer et al [24] created three synthetic scene flow datasets. They also proposed a network for disparity and scene flow estimation. [30] assume a dynamic scene contains foreground objects and background and apply instance segmentation masks over foreground to treat foreground and background differently.

Similarly, [31] developed a neural network that jointly estimates object segmentation, trajectories of objects, and the object scene flow from two consecutive RGB-D frames. Ilg et al. [13] proposed a network based on FlowNet [7] to estimate occlusions and disparity together. [23] integrates three vision cues to estimate scene flow for rigid objects in self-driving tasks. The three vision cues are segmentation masks, disparity map, and optical flow and they are extracted by existing networks, i.e. Mask R-CNN [9], PSMNet [4], and PWC-Net [34].

All the above approaches are mostly image-based so that appearance features can be conveniently extracted using 2D convolution. However, some sensory data such as laser scanners is unstructured and therefore conventional convolution is not applicable. To address the problem, Behl et al [1] evaluated the performance of scene flow estimation when integrating bounding box and integrating pixel-wise segmentation to scene flow estimation pipeline. [1, 22, 8] are designed for scene flow estimation on the point cloud. PointFlowNet [1] proposed to estimate scene flow, ego-motion and rigid object motion at the same time. In comparison to the PointFlowNet, FlowNet3D [22] and HPLFlowNet [8] are more general scene flow estimation frameworks that do not rely on rigid object assumption. More specifically, FlowNet3D extracts features with PointNet++ [29], mixes features and computes a coarse scene flow using a flow embedding layer, and propagates coarse scene flow to finer level using a set-upconv layer. HPLFlowNet, instead of using PointNet++, states that using permutohedral lattice[20] and Bilateral Convolutional Layer (BCL) [17] can improve global information extraction and faster performance.

3. Method

FlowNet3D is a neural network for estimating 3D scene flow $\mathcal{V}_{ts}$ given two point clouds, namely the source point cloud $\mathcal{X}_s$ and the target point cloud $\mathcal{X}_t$, where $\mathcal{X}_s$ and $\mathcal{X}_t$ are two sets of unordered 3D points. For generality, the numbers of both point clouds do not have to be identical, i.e. $|\mathcal{X}_s| \neq |\mathcal{X}_t|$, but the predicted vector field always has the same dimension as the source point cloud. FlowNet3D adopts the Siamese architecture that first extracts downsampled point features for each point cloud using the PointNet++, and then mixes the features in the flow embedding

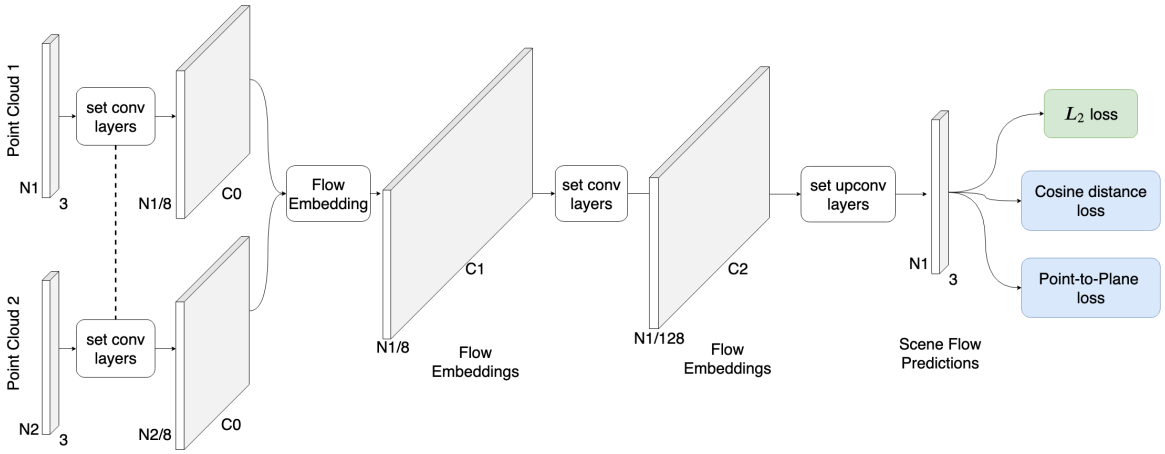


Figure 1. We adapt FlowNet3D structure and highlight the loss terms we added with blue boxes. **Set conv layers**: down-sampling and encoding. **Flow embedding layers**: aggregate spatially close features of two point clouds. **Set upconv layers**: Up-sampling and decoding implicit correspondences to scene flow vectors.

layer. In the end, the output features of the flow embedding are imposed with the regularisation and up-sampled into the same dimensionality as the $\mathcal{X}_s$. The network is trained using the loss function of the L_2 norm between predicted $\|\mathcal{V}_{ts} - \mathcal{V}_{gt}\|_2$ where $\mathcal{V}_{gt}$ is the groundtruth scene flow field.

FlowNet3D has been successfully applied in rigid scenes. In this paper, we further explore the potential of it when applied to non-static scenes, even the scenes dominated by deformable objects. More importantly, we introduce two loss terms that improves the accuracy of the prediction in both dynamic scenes while maintains the performance in rigid scene (measured using KITTI dataset). The new loss term also speeds up and stabilises the training procedure. Fig 1 illustrates the general idea of FlowNet3D and loss terms we applied. More details on the original FlowNet3D structure can be found in its paper.

3.1. Geometric Constraints

Point-to-Plane Loss is inspired by the popular point-to-plane distance metric for point cloud registration, such as the Iterative Closest Point (ICP) algorithm. Specifically, we can use the set $\mathcal{X}_l^n$ and $\mathcal{X}_w^n$ to represent two 3D point clouds at the n^{th} frame, where the labels l and w represent the live camera and the world coordinate system, respectively. Each point $\mathbf{x}_{\in}\mathcal{X}_w^n$ is a 3D homogeneous coordinate. The point $\mathbf{x}_w^n$ can be transformed from the world coordinate into the camera coordinate using $\mathbf{x}_l^n = \mathbf{T}_{cl}^n \mathbf{x}_w^n$, where $\mathbf{T}_{lw}^n \in \mathbb{SE}(3)$ is a 3D rigid transformation.

Given $\mathcal{X}_l^n$ and $\mathcal{X}_w^n$, $\mathbf{T}_{cw}^n$ can be estimated by minimising the following error function from a typical ICP algorithm [26]:

$$\mathbf{T}_{cw}^n = \arg\min_{\hat{\mathbf{T}}_{cw}^n} \sum_{\mathbf{x}_w^n \in \mathcal{X}_w^n} \|\mathbf{n}(\mathbf{x}_c^n)^\top (\mathbf{T}_{cw}^n \mathbf{x}_w^n - \mathbf{x}_c^n)\|^2, \quad (1)$$

where $\mathbf{n}(\mathbf{x}_c^n)$ is the function to calculate the surface normal at $\mathbf{x}_c^n$. $\mathbf{x}_c^n$ is the closet point to $\mathbf{T}_{cw}^n \mathbf{x}_w^n$. The dot product between the surface normal and the closet distance measures the distance from $\mathbf{T}_{cw}^n \mathbf{x}_w^n$ to the plane defined by $\mathbf{x}_c^n$ and its normal, hence it is known as the point-to-plane metric.

Inspired by the point-to-plane metric, we introduce a new loss for training the FlowNet3D, which is defined as follows:

$$\mathcal{L}_{pp} = \sum_{\mathbf{x}_s \in \mathcal{X}_s} \|\mathbf{n}(\mathbf{x}_t)^\top (\mathbf{x}_s - \mathbf{x}_t)\|^2, \quad (2)$$

where $\mathbf{x}_t$ is the closest point in the target set $\mathcal{X}_t$ to the source point $\mathbf{x}_s \in \mathcal{X}_s$. The scene flow may encode any rigid transformation or simply the non-static motion field, which is ultimately determined by the the samples provided during training. During training on FlyingThings, both $\mathcal{X}_s$ and $\mathcal{X}_t$ are in the same coordinate system and therefore, the trained model naturally learns to represent segments of rigid motion fields. Interestingly, we found that the same model can generalise to the point clouds extracted from consecutive frames of a deforming object so well that it outperforms the state-of-the-art dynamic fusion algorithm.

Cosine Distance Loss aims at constraining the angle between predicted flow field and the ground truth. From the scene flow predictions of FlowNet3D, we noticed that some of the predicted motion vectors differ greatly in direction from the groundtruth. As a result, we introduce the cosine distance loss which aims to minimise the angle between prediction and ground truth. We compute the cosine distance directly between a predicted vector and its groundtruth. This provides extra penalisation to vectors with directions with deviate from the groundtruth, even if they have the same $\mathcal{L}_2$ loss. Fig 2 illustrates the effect of applying $\mathcal{L}_2$ loss and cosine distance together.

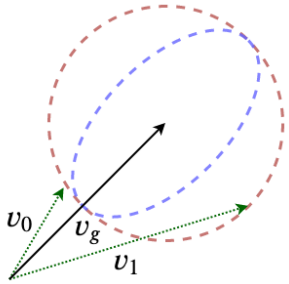


Figure 2. The red circle denotes the energy contour that $\mathcal{L}_2$ loss penalises equally. The blue circle denotes the energy contour after combining $\mathcal{L}_2$ and $\mathcal{L}_{\cos}$. $\mathcal{L}_{\cos}$ explicitly punishes vectors with large angle deviations. In this figure, v_1 is penalised more than v_0 after adding $\mathcal{L}_{\cos}$.

Combined Loss includes all three different loss terms using a weighted summation:

$$\mathcal{L} = \mathcal{L}_2 + \lambda_p \mathcal{L}_{pp} + \lambda_{\cos} \mathcal{L}_{\cos} \tag{3}$$

where λ_p and λ_{cos} are the weight to balance among the loss terms, the $\mathcal{L}_2 = \frac{1}{|\mathcal{V}'_{ts}|} \sum_{\mathbf{v} \in \mathcal{V}'_{ts}} \|\mathbf{v} - \mathbf{v}_{gt}\|_2$ and $\mathcal{L}_{cos} = \frac{1}{|\mathcal{V}'_{ts}|} \sum_{\mathbf{v} \in \mathcal{V}'_{ts}} \frac{\mathbf{v} \cdot \mathbf{v}_{gt}}{\|\mathbf{v}\| \|\mathbf{v}_{gt}\|}$. The $\mathcal{V}'_{ts}$ is the predicted vector field and $\mathbf{v}_{gt}$ is an individual ground truth vector corresponding to $\mathbf{v}$. It is worth noting that the cosine loss and $\mathcal{L}_2$ loss weigh over the angles and lengths of the predicted vector field, respectively.

3.2. Scene Flow for Dynamic 3D Reconstruction

The performance of flow field estimation on a dynamic scene is normally evaluated through the counting of inliers, which are determined through a set threshold. However, this evaluation scheme depends heavily on the threshold, which must be set heuristically. We propose to benchmark scene flow framework based on a state-of-the-art dynamic 3D reconstruction system, so that the scene flow can be evaluated by viewing a 3D model. This provides a more holistic performance measure, as well as a practical application for 3D flow estimation.

Dynamic 3D reconstruction is recently introduced for recovering non-static objects, including deformable objects such as moving animals or human beings [25, 14, 32, 33, 18, 19]. KillingFusion [32] and its variant SobolevFusion [33] represent the state-of-the-art dynamic fusion method directly estimating a dense vector field between two TSDF volumes. However, this variational optimisation process is easily trapped in local minimum when the search space is large. Our benchmark framework, which can also be considered as a dynamic reconstruction system, significantly outperforms KillingFusion in terms of quality by a 35.2% reduction in mean error.

Particularly, our benchmark framework takes in a sequence of point clouds with corresponding scene flow predictions to recover a 3D model. The reconstruction error can be visualised comparing with the ground truth model. In experiments, the FlowNet3D++ reduces up to 15.0% error in the dynamic reconstruction task compare to FlowNet3D.

We show the essential steps of the dynamic reconstruction pipeline below. An illustration of this process is also available in the supplementary material.

1. Compute the rigid $\mathbb{SE}(3)$ transformation between the live point cloud and the canonical model using point-to-plane ICP algorithm [2] as shown in Eq. (1). It compensates the overall movement of the target object by absorbing it into the current camera pose.

2. Predict the scene flow between the live and the canonical point cloud. In this paper we experimented the FlowNet3D[21] and our FlowNet3D++.

3. Warp the live point cloud using the scene flow computed in the last step and create the synthetic depth map by projecting the warped live point cloud into the compensated camera pose.

4. Construct a live TSDF volume ϕ_n from the synthetic depth map using the widely used depth to volume integration method, which is first introduced in KinectFusion [26].

5. Refine the vector field Ψ_V between the live volume and canonical volume using a simple variational voxel based scene flow refinement.

6. Update the ϕ_{global} by taking the voxel-wise weighted average between ϕ_n and ϕ_{global} for the TSDF values live and accumulate the weight [6].

Step 2 introduces the deep scene flow to warp the live point cloud so that a virtual TSDF volume that is much easier for the KillingFusion to optimise and therefore reduces the computation complexity and quality of the recovered model. Step 3, 4 and 5 formulate our novel scene flow integrator that integrates a scene flow in point cloud resolution to the full TSDF volume resolution with very little artefacts.

Scene Flow Integrator merges multiple point cloud into a single 3D volumetric representation from which the 3D model can be extracted. Specifically, assuming $\mathcal{X}_l^n$ represents point cloud of live frame in camera coordinate and $\mathcal{X}_g^n$ represents the ray-casted point cloud from the canonical model, the scene flow predictor computes a scene flow field Ψ_S^n that associate $\mathcal{X}_l^n$ with $\mathcal{X}_g^n$. The warping from $\mathcal{X}_l^n$ to $\mathcal{X}_g^n$ can be formulated as follows:

$$\mathcal{X}_g^{n\prime} = \{\mathbf{x} | \mathbf{x} := v(\mathbf{x}_l^n) + \mathbf{x}_l^n\}, \tag{4}$$

where $\mathbf{x}_l^n \in \mathcal{X}_l^n$. Note that Ψ_S^n and $\mathcal{X}_l^n$ share the same resolution and $\mathcal{X}_g^{n\prime}$ and $\mathcal{X}_g^n$ are different. Therefore, our

target is to integrate $\mathcal{X}_g^{n\prime}$ into the canonical volume ϕ_{global} smoothly.

Naively integrating $\mathcal{X}_g^{n\prime}$ into the global TSDF volume seems a reasonable solution, however, in our experiments we discover this causes significant artefacts. This is because scene flow computed on point clouds is only capable of inferring the motion on the object surface or the zero level set. To deform a volumetric TSDF, the vector field has to cover the entire 3D region within the truncated area while maintaining the property of TSDF to be a precise level set function so that the artefacts are minimised.

Therefore, we tackle the problem by: 1) creating a synthetic depth map $\tilde{D}^n = \{\tilde{d}(u,v)|(u,v) \in \Omega\}$, where Ω is a set of pixel locations on the depth map, by projecting $\mathcal{X}_g^{n\prime}$ onto the depth map $\tilde{D}^n$. 2) creating a synthetic live volume ϕ_n from $\tilde{D}^n$. This is equivalent to integrating a depth map to an empty TSDF volume.

By converting the deformed point cloud $\mathcal{X}_g^{n\prime}$ into a TSDF volume ϕ_n, we have acquired a coarse alignment between the ϕ_n and ϕ_{global}. The next step is to refine this coarse alignment with a simple variational vector field refinement.

Voxel Based Vector Field Refinement: The concept of running variational optimisation directly on TSDF volume was first introduced in the KillingFusion [32] and simplified in SobolevFusion [33]. It solves the vector field by evolving the source TSDF into target TSDF iteratively. This approach enjoys the advantage of being capable of dealing with topological changes but a drawback of this variational SDF evolution lies in that it can easily get trapped in some local minima. This is because it lacks explicit correspondences associating level set functions. By providing a good initial solution from our deep scene flow estimator, only a few iterations of voxel based vector field refinement is needed. Specifically, for a voxel at position $\mathbf{x} \in \mathcal{V}$ and a 3D vector $\mathbf{v} = v(\mathbf{x})$ associates with this voxel, our energy is simply defined as:

$$E(\Psi_V) = \frac{1}{2} \sum_{\mathbf{x} \in \mathcal{V}} \left(\phi_n(\mathbf{x}+\mathbf{v}) - \phi_{global}(\mathbf{x})\right)^2 \qquad (5)$$

where $\phi(\mathbf{x})$ represent TSDF value at voxel centre $\mathbf{x}$ and the energy can be optimised using gradient descent easily:

$$\mathbf{v}^{(k+1)} = \mathbf{v}^{(k)} - \alpha E'(\mathbf{v}^{(k)}) \qquad (6)$$

where $\mathbf{v}^{(k)} \in \Psi_V^{(k)}$ represents the vector field Ψ_V^n at its k^{th} iteration and $E'(\mathbf{v})$ is the gradient with respect to the $\mathbf{v}$ and can be computed efficiently using the following calculus of variations:

$$E'(\mathbf{v}) = (\phi_n(\mathbf{x}+\mathbf{v}) - \phi_{global}(\mathbf{x}))\nabla\phi_n(\mathbf{x}+\mathbf{v}), \qquad (7)$$

where the $\nabla\phi_n(\mathbf{x}+\mathbf{v})$ is the spatial gradient at the voxel position $(\mathbf{x}+\mathbf{v})$ in the live volume ϕ_n.

It is worth noting that the vector field computed from above optimisation is only meaningful in local regions and the purpose is two-fold: (i) to register a roughly aligned live volume to canonical model; (ii) to remove artefacts introduced in the coarse non-rigid point cloud registration. Thanks to the quality of the deep scene flow estimator, we no longer require a regularisation term, such as those in KillingFusion and SobolevFusion.

The above energy will produce a scene flow vector for each SDF voxel. In general, the magnitude of this vector field should be small because the main evolution has already been compensated when warping the live point cloud to the canonical point cloud. As a result, with a small number of iterations, typically ranging from 3 to 70, we can mediate the artefacts and noise introduced in from scene flow predictor.

We are aware that having variational refinement may affect the deep scene flow benchmarking result. However, this variational refinement is necessary for complex tasks like dynamic reconstruction. If not present, the tracking can fail after a few frames due to the large accumulated error. To eliminate the effect of this variational refinement in benchmarking, we explicitly set a fixed iteration number for all experiments. For the Snoopy and Duck dataset, we use 30 iterations for all deep scene flow benchmarking.

4. Experiments

In this section, we evaluate our modifications to FlowNet3D and validate their effectiveness quantitatively in two subsections. In the first subsection, we benchmark our FlowNet3D++ result using the existing scene flow datasets FlyingThings and KITTI, which are pre-processed and provided by FlowNet3D. For pre-processing details, we refer reader to the FlowNet3D supplementary material. We also provide a graph to analyse the time taken for our training to converge. In the second subsection, we quantitatively evaluate the performance of FlowNet3D++ in our novel dynamic reconstruction benchmark. This is performed on two reconstruction datasets (Snoopy and Duck), both of which are provided by KillingFusion[32]. Further qualitative results can be found in our supplementary material.

To enable the point-to-plane loss term in Eq. 2, we also pre-compute per-point surface normal for the FlyingThings dataset but we do **not** use surface normals as input features.

Our model is trained from scratch using the training split of FlyingThings dataset and testing is performed on the test split. We directly transfer our model that was trained on FlyingThings to KITTI without any fine-tuning. For the dynamic reconstruction benchmark, we also directly deploy the model that was trained on FlyingThings dataset to the pipeline, again without fine-tuning. For hyper-parameters, in most experiments we use exactly same hyper-parameters the FlowNet3D used to show the effectiveness of our loss

Input channels	Model	ACC (0.05)	ACC (0.10)	EPE	ADE (degree)
xyz	F3D	23.71%	56.05%	0.1705	22.83
	F3D++	**28.50%**	**60.39%**	**0.1553**	**20.78**
rgb	F3D	25.37%	57.85%	0.1694	22.58
	F3D++	**30.33%**	**63.43%**	**0.1369**	**21.14**

Table 1. F3D is a shorthand for FlowNet3D. We evaluate our FlowNet3D++ in two input settings. Input setting *xyz* means we only use point position as input features. Input setting *rgb* means both point position and the colour features are fed into the network. For EPE and ADE, the lower the better.

terms. For the best result we show in Table 4.2, we trained 200 epochs.

4.1. Metrics

We report our results using 3D End-Point-Error (EPE) and an accuracy metric (ACC) with two thresholds. These three metrics are also used in FlowNet3D to provide fair comparison. We also propose the average angle deviation error (ADE) for this task for the evaluation of the predicted scene flow vectors' direction. **EPE**: the EPE is the L_2 norm between an estimated flow vector and its ground truth vector. **ADE**: we define the ADE as $\arccos(\frac{1}{N}\sum\cos(v_p, v_{gt}))$, where v_p and v_{gt} are predicted vector and its ground truth vector.

4.2. FlyingThings Dataset

In FlowNet3D++, we apply both the cosine distance loss and point-to-plane loss alongside the original L_2 loss. The results listed in Table 4.2 show that our modifications improve all metrics that we test. In fact, the geometric-only XYZ-FlowNet3D++ even outperforms RGB-FlowNet3D, which is allowed to incorporate colour information. We use $\lambda_{pp} = 1.3$ and $\lambda_{cos} = 0.9$ for this test, but we found $\lambda_{pp,cos} \in [0.5, 1.5]$ generates good results in the general case. As the FlowNet3D did not evaluate ADE, we compute FlowNet3D's ADE with the pre-trained model provided by [21].

4.3. KITTI Dataset

As KITTI scene flow dataset only provides a colourless, LiDAR-scanned point cloud, we only show the results for geometry-only models.

Model	Outlier	EPE	ADE
F3D (with our eval script)	7.53%	0.3259	42.60
F3D++	**4.81%**	**0.2530**	**36.86**

Table 2. KITTI scene flow benchmark. We report that our model has significantly lower error in all tested error metrics.

We propose a more simple evaluation procedure on the KITTI dataset than was used in [21]. Instead of cutting the KITTI point cloud into numerous chunks and having to deal with overlapping regions, we resize the KITTI dataset to the size of FlyingThings scenes, which is $x \in [-15m, +15m], y \in [-8m, +8m], z \in [0m, 35m]$, before feeding it to networks. Although this produces differing results than in [21], we ensure a fair comparison by training both FlowNet3D and FlowNet3D++ on FlyingThings and transferring to our resized KITTI without fine-tuning. We report our results in table 2.

4.4. Dynamic Dense Reconstruction

In this section, we demonstrate the effectiveness of FlowNet3D++ within our proposed dynamic dense reconstruction benchmark.

4.4.1 Configuration

Our depth-only dynamic reconstruction system is implemented on top of InfiniTAM [28], an open sourced RGBD dense SLAM system with modern CUDA support. The volume resolution is set as 256^3 and voxel size 3 mm or 5 mm for all of our experiments. Specifically, we use 3mm for small scenes like *Snoopy* and *Duck* dataset [32] and 5 mm for the VolumeDeform datasets [14]. The truncated distance δ is set to ± 4 times of the voxel size. The step size α for optimiser is set to 0.1. We also implement a SobolevFusion system for comparison (comparison images can be found in Appendix). Similar to the KITTI scene, applying FlowNet3D++ to videos that captured with different cameras requires the scene to be resized to the range of FlyingThings dataset, i.e. $x \in [-15m, +15m], y \in [-8m, +8m], z \in [0m, 35m]$. The choice of scaling factors depends on different voxel size, SDF volume size and camera intrinsics. However, in practice, we found a rough estimation of the scaling factors works well for all the experiments. In particular, for the *Snoopy* sequence, the scaling factors are set as $s_x = 25, s_y = 25, s_z = 30$. The good results achieved through this resizing method in dynamic reconstruction provide evidence that the resizing in KITTI evaluation is also valid.

4.4.2 Results

The KillingFusion dataset (Snoopy and Duck) provides a ground truth mesh. Thus, we can quantitatively analyse the benefit of adding deep scene flow estimation to the dynamic reconstruction. We also present more images of running our systems on VolumeDeform dataset and a video sequence we record by ourselves in Appendix, to illustrate the benefit of our pipeline qualitatively. Our Snoopy and Duck evaluation result is reported in Table 3 and Fig 3.

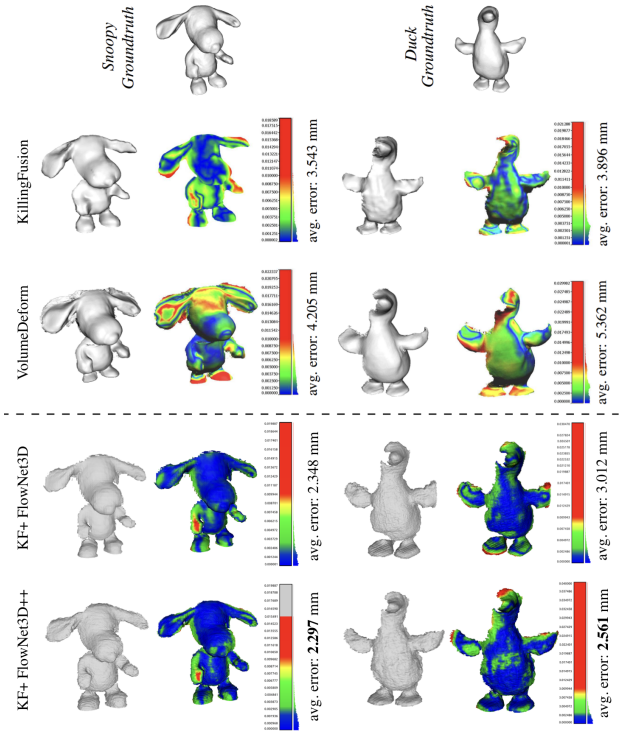


Figure 3. Mean mesh-to-mesh error. Blue regions represent low error. We use red to represent an error that is larger than 1cm. Top: ground truth mesh. 2^{nd} row and 3^{rd}: results of KillingFusion and VolumeDeform (image copied from [32]). 4^{th} row: Our reconstruction results with unmodified *FlowNet3D*. Bottom row: our reconstruction result after applying our *FlowNet3D++*. It can be seen that most of our model appears as blue indicating low overall error. The bottom two rows demonstrate that FlowNet3D++ offers an improvement over FlowNet3D. Quantitatively, our average error on *Snoopy* and *Duck* scene is **35.2**% and **34.3**% lower than KillingFusion reconstructions, respectively.

	Mean Error To Ground Truth (mm)			
Scenes	Volume Deform	Killing Fusion	KF + FlowNet3D	KF + FlowNet3D++
Snoopy	4.205	3.543	2.348	**2.297**
Duck	5.362	3.896	3.012	**2.561**

Table 3. Evaluation of FlowNet3D and FlowNet3D++ in our dynamic reconstruction benchmark

5. Ablation study

To validate the individual benefit derived from each of our geometric constraints, as well as their combination, we perform ablation tests for both the geometric-only models and colour models. Unless otherwise stated, we use exactly same training procedure as described in [22].

Results are shown in Table 4 and Table 5. The results in the bottom rows of Table 4 and Table 5, we trained for 200 epochs, instead of 150 epochs.

In addition to the overall performance of the geometric loss terms, it is also worth noting that in the RGB setting, simply combining $\mathcal{L}_{pp}$ + $\mathcal{L}_{\cos}$ does not yield the best ACC and EPE after 150 epochs of training. Instead, the best result acquired after this schedule is the model trained with $\mathcal{L}_{pp}$. However, we found that the accuracy of FlowNet3D + $\mathcal{L}_{pp}$ plateaus after 150 epochs, whereas the accuracy of model with both $\mathcal{L}_{pp}$ and $\mathcal{L}_{\cos}$ still grows until 200 epochs. Therefore, the combination of geometric losses in the RGB setting provides our best configuration. In the XYZ setting, however, the combination of geometric loss terms provides the best result, even after the 150 epoch schedule.

Models (XYZ)	ACC (0.05)	ACC (0.10)	EPE	ADE
F3D	23.71%	56.05%	0.1705	22.83
F3D + $\mathcal{L}_{pp}$	27.79%	60.06%	0.1567	21.96
F3D + $\mathcal{L}_{\cos}$	25.30%	58.15%	0.1615	21.17
F3D + $\mathcal{L}_{pp}$ + $\mathcal{L}_{\cos}$	**28.22**%	**60.11**%	**0.1556**	**20.75**
F3D + $\mathcal{L}_{pp}$ + $\mathcal{L}_{\cos}$	28.50%	60.39%	0.1553	20.77

Table 4. Ablation study for FlowNet3D geometry-only model. The last row is the result trained for 200 epochs. Other rows including the original FlowNet3D are all trained for 150 epochs.

Models (RGB)	ACC (0.05)	ACC (0.10)	EPE	ADE
F3D	25.37%	57.85%	0.1694	22.58
F3D + $\mathcal{L}_{pp}$	**28.52**%	**62.75**%	**0.1391**	21.74
F3D + $\mathcal{L}_{\cos}$	26.84%	61.57%	0.1454	**20.96**
F3D + $\mathcal{L}_{pp}$ + $\mathcal{L}_{\cos}$	26.05%	60.53%	0.1492	21.27
F3D + $\mathcal{L}_{pp}$ + $\mathcal{L}_{\cos}$	30.33%	63.43%	0.1369	21.14

Table 5. Ablation study for FlowNet3D colour model. The last row is the result trained for 200 epochs. Other rows including the original FlowNet3D are all trained for 150 epochs.

6. Conclusion

In this paper, we introduced FlowNet3D++, which to the best of our knowledge is the state-of-art point cloud-based deep scene flow estimator. We contribute two principled geometric constraints that each improve the accuracy of the state-of-art of point cloud based deep scene flow from **57.85**% to **63.43**%. We also contribute a novel geometric based scene flow benchmark pipeline in dynamic reconstruction context. Within our deformable scene flow benchmark, our FlowNet3D++ achieves up to 15.0% less reconstruction error than FlowNet3D, and up to a 35.2% improvement over KillingFusion alone.

Acknowledgements

We gratefully acknowledge the European Commission Project Multiple-actOrs Virtual Empathic CARegiver for the Elder (MoveCare) grant for financially supporting the authors of this work.

References

[1] A. Behl, D. Paschalidou, S. Donné, and A. Geiger. Pointflownet: Learning representations for rigid motion estimation from point clouds. In *CVPR*, 2019. 2

[2] P. Besl and N. D. McKay. A method for registration of 3-d shapes. *PAMI*, 1992. 1, 4

[3] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *PAMI*, 2010. 1, 2

[4] J.-R. Chang and Y.-S. Chen. Pyramid stereo matching network. In *CVPR*, 2018. 2

[5] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 1

[6] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques*, 1996. 4

[7] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 2

[8] X. Gu, Y. Wang, C. Wu, Y. J. Lee, and P. Wang. Hplflownet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds. In *CVPR*, 2019. 2

[9] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. *PAMI*, 2018. 2

[10] M. Hornacek, A. Fitzgibbon, and C. Rother. Sphereflow: 6 dof scene flow from rgb-d pairs. In *CVPR*, 2014. 2

[11] F. Huguet and F. Devernay. A variational method for scene flow estimation from stereo sequences. In *ICCV*, 2007. 2

[12] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017. 2

[13] E. Ilg, T. Saikia, M. Keuper, and T. Brox. Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. In *ECCV*, 2018. 2

[14] M. Innmann, M. Zollhöfer, M. Nießner, C. Theobalt, and M. Stamminger. Volumedeform: Real-time volumetric non-rigid reconstruction. In *ECCV*, 2016. 4, 6

[15] M. Jaimez, M. Souiai, J. Gonzalez-Jimenez, and D. Cremers. A primal-dual framework for real-time dense rgb-d scene flow. In *ICRA*, 2015. 2

[16] M. Jaimez, M. Souiai, J. Stückler, J. Gonzalez-Jimenez, and D. Cremers. Motion cooperation: Smooth piece-wise rigid scene flow from rgb-d images. In *3DV*, 2015. 2

[17] V. Jampani, M. Kiefel, and P. V. Gehler. Learning sparse high dimensional filters: Image filtering, dense crfs and bilateral neural networks. In *CVPR*, 2016. 2

[18] H. Joo, H. Liu, L. Tan, L. Gui, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *ICCV*, 2015. 4

[19] H. Joo, T. Simon, and Y. Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *CVPR*, 2018. 4

[20] M. Kiefel, V. Jampani, and P. V. Gehler. Permutohedral lattice cnns. *arXiv preprint arXiv:1412.6618*, 2014. 2

[21] X. Liu, C. R. Qi, and L. J. Guibas. Flownet3d: Learning scene flow in 3d point clouds. In *CVPR*, 2019. 1, 4, 6

[22] X. Liu, C. R. Qi, and L. J. Guibas. Flownet3d: Learning scene flow in 3d point clouds. In *CVPR*, 2019. 2, 7

[23] W.-C. Ma, S. Wang, R. Hu, Y. Xiong, and R. Urtasun. Deep rigid instance scene flow. In *CVPR*, 2019. 2

[24] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016. 2

[25] R. A. Newcombe, D. Fox, and S. M. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *CVPR*, 2015. 4

[26] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, 2011. 1, 3, 4

[27] J.-P. Pons, R. Keriven, and O. Faugeras. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *IJCV*, 2007. 2

[28] V. A. Prisacariu, O. Kähler, S. Golodetz, M. Sapienza, T. Cavallari, P. H. Torr, and D. W. Murray. Infinitam v3: a framework for large-scale 3d reconstruction with loop closure. *arXiv preprint arXiv:1708.00783*, 2017. 6

[29] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017. 1, 2

[30] Z. Ren, D. Sun, J. Kautz, and E. Sudderth. Cascaded scene flow prediction using semantic segmentation. In *3DV*, 2017. 2

[31] L. Shao, P. Shah, V. Dwaracherla, and J. Bohg. Motion-based object segmentation based on dense rgb-d scene flow. *Robotics and Automation Letters (RAL)*, 2018. 2

[32] M. Slavcheva, M. Baust, D. Cremers, and S. Ilic. Killingfusion: Non-rigid 3d reconstruction without correspondences. In *CVPR*, 2017. 1, 4, 5, 6, 7

[33] M. Slavcheva, M. Baust, and S. Ilic. Sobolevfusion: 3d reconstruction of scenes undergoing free non-rigid motion. In *CVPR*, 2018. 4, 5

[34] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 2

[35] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *ICCV*, 1999. 2

[36] C. Vogel, K. Schindler, and S. Roth. 3d scene flow estimation with a rigid motion prior. In *ICCV*, 2011. 2

[37] A. Wedel, C. Rabe, T. Vaudrey, T. Brox, U. Franke, and D. Cremers. Efficient dense scene flow from sparse or dense stereo data. In *ECCV*, 2008. 2

FlowNet3D++: Geometric Losses For Deep Scene Flow Estimation (Supplementary Material)

Zirui Wang
ryan@robots.ox.ac.uk

Shuda Li
shuda@robots.ox.ac.uk

Henry Howard-Jenkins
henryhj@robots.ox.ac.uk

Victor Adrian Prisacariu
victor@robots.ox.ac.uk

Min Chen
min.chen@oerc.ox.ac.uk

1. Additional Details on Dynamic Reconstruction Pipeline

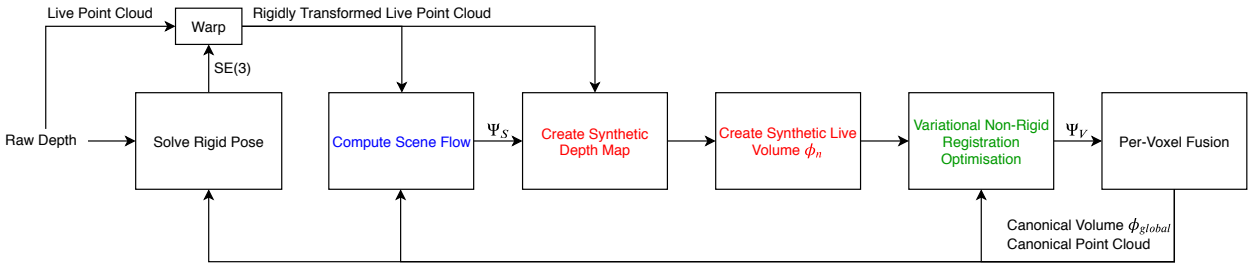


Figure 1. Dynamic Reconstruction Pipeline. This figure is supplements Section 3.2 in the main paper.

2. Additional FlowNet3D++ Results With Our Dynamic Reconstruction Benchmark

We present more qualitative results to demonstrate that integrating FlowNet3D++ is useful for high-level dynamic reconstruction applications. We utilise our dynamic reconstruction benchmark pipeline to make qualitative comparisons between dynamic reconstruction results with FlowNet3D++ integrated and without FlowNet3D++. For dynamic reconstruction baseline, we implement SobolevFusion[4], which is an improved version of KillingFusion[3].

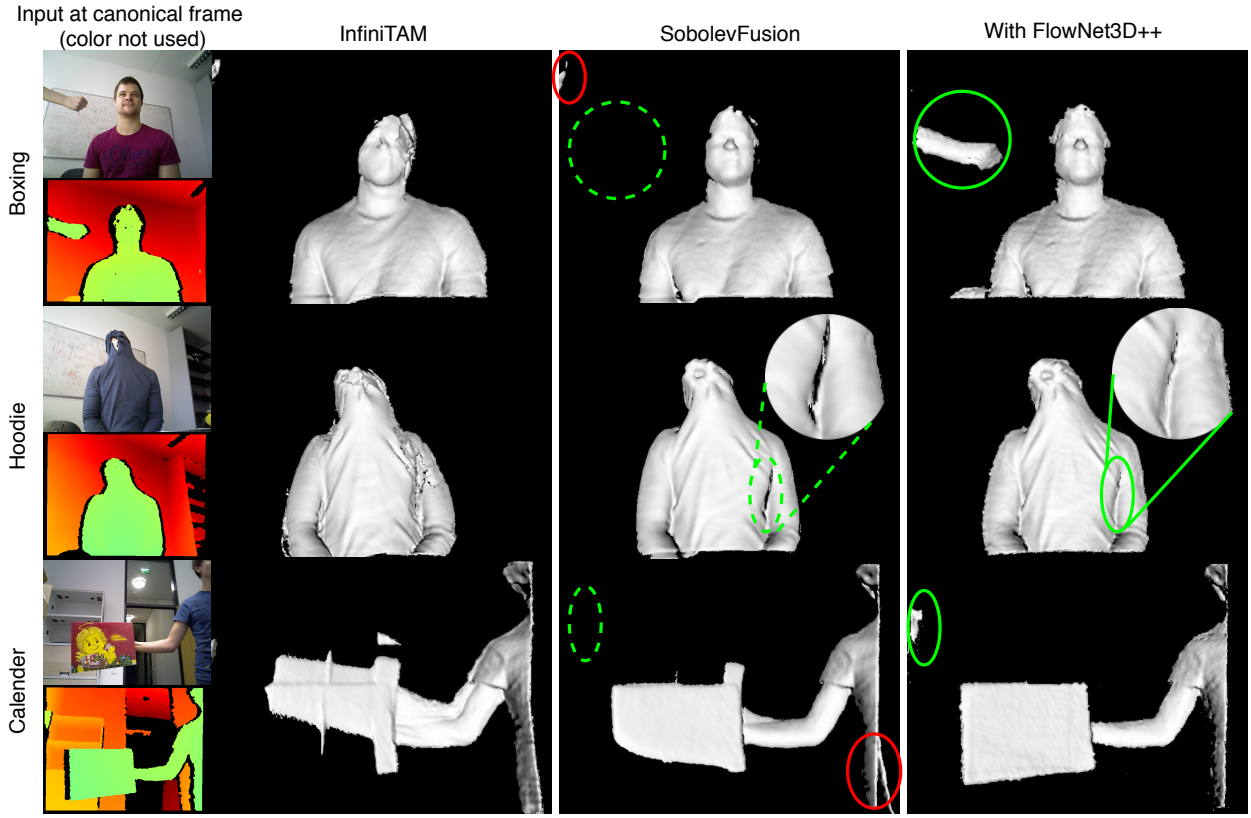


Figure 2. Results of running our benchmark on VolumeDeform [1] Dataset. Aside from obvious differences, we use red circles to highlight places where integrating FlowNet3D++ produces less distortion and green circles to highlight places integrating FlowNet3D++ preserves more detail. Note there is no foreground mask, the background is missing because of volume size. For scenes with more rigid components (the hoodie sequence), we have similar results with SobolevFusion because the rigid tracker lifts the most weight.

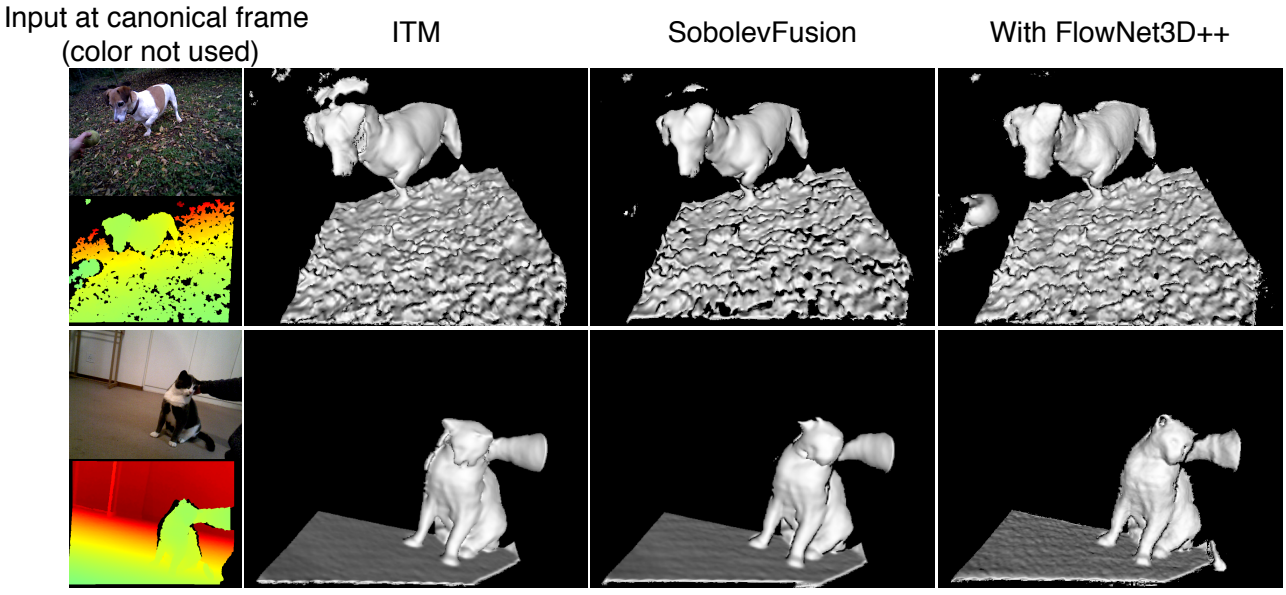


Figure 3. Results of running our benchmark on a dataset used in [2]. Our results have fewer artefacts and more details than SobolevFusion after few hundreds of frames.

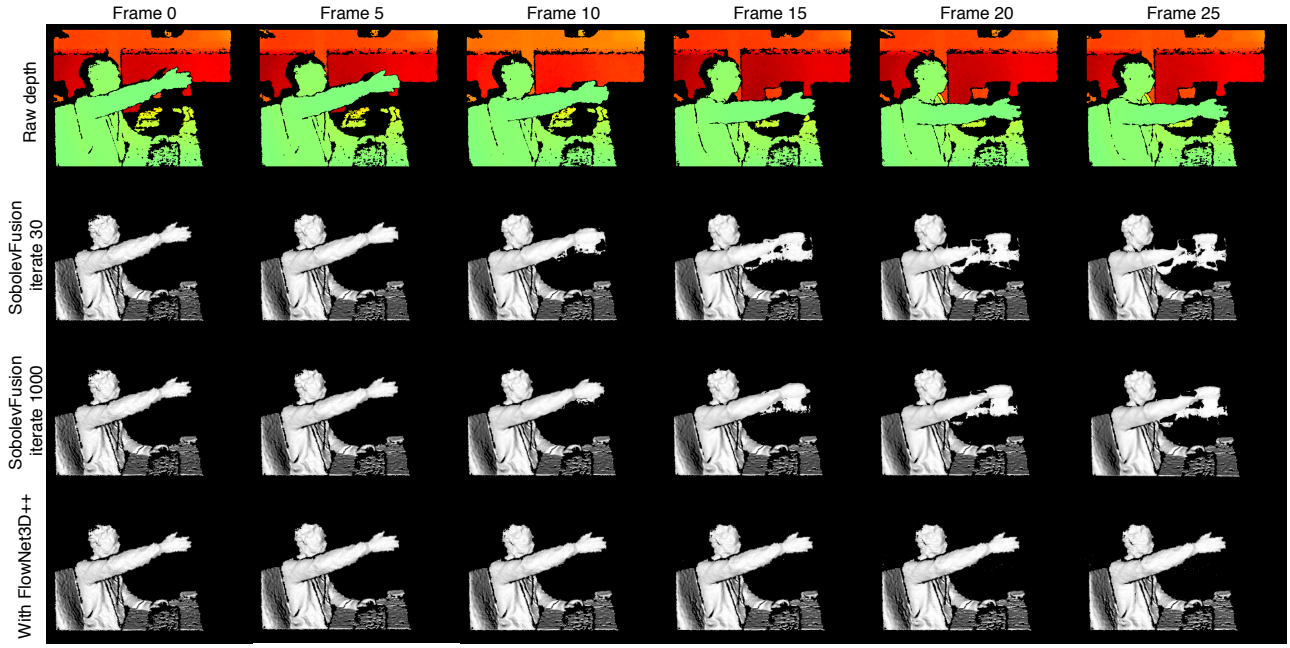


Figure 4. A comparison of the different number of iterations on a video we recorded. We show the canonical model at different time stamps. From the figure we can see that when the iterations increase, the SobolevFusion results improve. However, it is still difficult to register the live volume to the canonical model. In contrast, integrating FlowNet3D++ enables the handling of large motion.

3. FlowNet3D++ Non-Rigid Registration

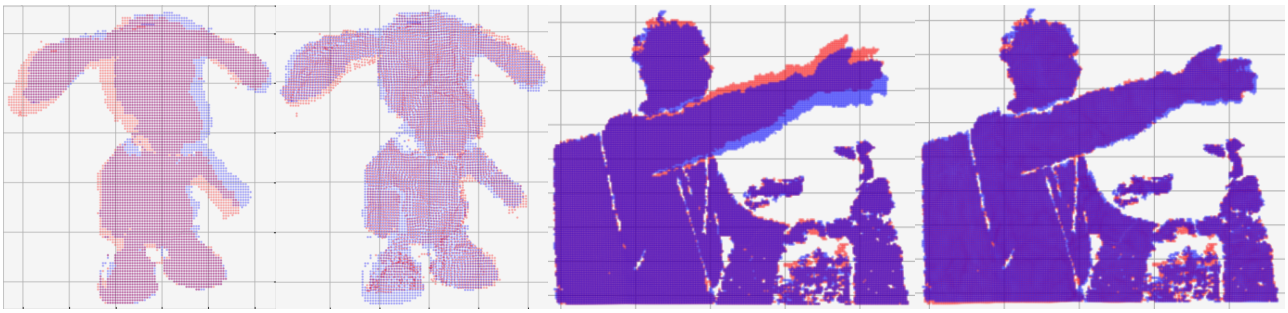


Figure 5. Point clouds warped using the deep scene flow. The left and right pairs are the results using the *Snoopy* sequence and *Waving* sequence, respectively. It can be seen that FlowNet3D++ has aligned the source point cloud (blue) with the target point cloud (red) effectively. Note that although the colour scheme for each of the pairs appears different, this is simply caused by the differing densities of their respective point clouds.

References

[1] M. Innmann, M. Zollhöfer, M. Nießner, C. Theobalt, and M. Stamminger. Volumedeform: Real-time volumetric non-rigid reconstruction. In *ECCV*, 2016. 2

[2] C. Malleson, J.-Y. Guillemaut, and A. Hilton. Hybrid modelling of non-rigid scenes from rgbd cameras. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018. 2

[3] M. Slavcheva, M. Baust, D. Cremers, and S. Ilic. Killingfusion: Non-rigid 3d reconstruction without correspondences. In *CVPR*, 2017. 1

[4] M. Slavcheva, M. Baust, and S. Ilic. Sobolevfusion: 3d reconstruction of scenes undergoing free non-rigid motion. In *CVPR*, 2018. 1

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Flownet3d++: Geometric Losses for Deep Scene Flow Estimation
Publication Status	☑ Published □ Accepted for Publication □ Submitted for Publication □ Unpublished and unsubmitted work written in a manuscript style
Publication Details	**Zirui Wang**, Shuda Li, Henry Howard-Jenkins, Victor Adrian Prisacariu, Min Chen. In Winter Conference on Applications of Computer Vision (WACV), 2020.

Student Confirmation

Student Name:	Zirui Wang		
Contribution to the Paper	First-author contribution: • Conception of research ideas. • Design and implementation of models. • Running of large-scale experiments. • Writing and presentation of the paper.		
Signature	王梓睿	Date	20/Sep/2023

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	Professor Victor Adrian Prisacariu		
Supervisor comments:	The description is accurate.		
Signature	VAP	Date	20/Sep/2023

This completed form should be included in the thesis, at the end of the relevant chapter.

3.2 Neighbourhood-Insensitive Point Cloud Normal Estimation Network

In this section, we present our research paper on point cloud normal estimation, an essential step for surface registration and reconstruction. This topic is particularly relevant to ICP algorithms, which play a key role in our scene flow network training as detailed in Section 3.1.

Traditional techniques such as Principal Component Analysis (PCA) [51] and Jet [132] requires a manually tuning of a global hyper-parameter, specifically the local neighbourhood radius, for optimal performance. This poses a challenge when handling scenes with complex geometries. Subsequent methods [53, 52] address this limitation by utilising multi-scale patches at a cost of complex system design.

We introduce a novel self-attention-based normal estimation network. This architecture allows each point contributes softly in normal estimation task, offering more accurate results while eliminating the need for manual neighbourhood range tuning and the use of multi-scale patches. Moreover, our design leads to significantly smaller model size and faster inference time. We further demonstrate the effectiveness of our model in the point-to-plane ICP registration task, showing faster convergence without neighbourhood range tuning.

arXiv:2008.09965v3 [cs.CV] 15 Jan 2021

Neighbourhood-Insensitive Point Cloud Normal Estimation Network

Zirui Wang
ryan@robots.ox.ac.uk
Victor Adrian Prisacariu
victor@robots.ox.ac.uk

Active Vision Lab,
University of Oxford,
Oxford, UK

Abstract

We introduce a novel self-attention-based normal estimation network that is able to focus softly on relevant points and adjust the softness by learning a temperature parameter, making it able to work naturally and effectively within a large neighbourhood range. As a result, our model outperforms all existing normal estimation algorithms by a large margin, achieving 94.1% accuracy in comparison with the previous state of the art of 91.2%, with a 25x smaller model and 12x faster inference time. We also use point-to-plane Iterative Closest Point (ICP) as an application case to show that our normal estimations lead to faster convergence than normal estimations from other methods, without manually fine-tuning neighbourhood range parameters. Code available at https://code.active.vision.

1 Introduction

Normal estimation is an important low-level task that forms the foundation for many high-level applications, such as tracking, reconstruction, and rendering. Principle Component Analysis (PCA) [11], the classic normal estimation method, estimates a normal vector at a point by collecting its k nearest neighbours (k-NN) and fitting a plane on them. It is worth noting that, the neighbourhood range k, is a global hyper-parameter requiring manual tuning for PCA to achieve best accuracy. In practice, a single global k is often not suitable to an entire scene, where different regions have different geometric structures. Therefore, an algorithm that can determine the neighbourhood range adaptively based on local geometric structures and is insensitive to the neighbourhood variance is desirable, as downstream tasks can enjoy high-quality normal estimations without manually tuning neighbourhood range parameters.

One possible approach to achieve this is to identify appropriate neighbourhood scales to estimate normals under different geometric properties. One line of research [2, 8] aims to solve this problem by integrating information from multi-scale patches. Here, the experts that specialise at small scales are responsible for sharper regions and the experts that specialise at large scales are responsible for smoother regions. A scale manager can choose to weight estimations from different experts. This approach is effective and does not require a manual fine-tuning of the neighbourhood range, but it comes at the expenses of a large

© 2020. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

neural network, i.e. for a model that processes L scales, the model is L times as large as a single scale model, for example, the 7-expert Nesti-Net [2] model is about 1.1GB.

In this paper, we propose a normal estimation network that selects neighbour points softly according to its local geometric properties, by applying a multi-head self-attention module to the neighbour points around the current point. Our model allows a soft selection of neighbour points by learning a temperature parameter that controls the softmax function within the attention module. As a result, our method does not need a carefully picked k. Given a fairly large neighbourhood range, for example, setting k to 30, 40, or even 50 when the best k for PCA is 8, our network is able to focus on relevant points adaptively and outperforms all existing normal estimation algorithms consistently, with a significantly smaller model and faster inference time. We also provide a point-to-plane ICP [6] application study to demonstrate how downstream algorithms like ICP can benefit from our neighbourhood-insensitive normal estimation.

The rest of the paper is organised as follows: Sec. 2 introduces related works on point cloud reasoning, normal estimation, and attention mechanisms. Sec. 3 describes our network in detail. We present our results in Sec. 4, together with a point-to-plane ICP application study, and conclude in Sec. 5.

2 Related Work

In this section, we present approaches related to our work, starting with neural network point cloud reasoning, moving to normal estimation from point clouds and ending with selected attention-based methods.

Neural Network Point Cloud Reasoning Learning-based 3D point cloud reasoning is an active research area. Su et al. [20] first proposed to process point clouds using CNNs on their 2D projections. This method can extract semantic information but lose geometric details significantly. Qi et al. [18] proposed to analyse point clouds using volumetric representations so that 3D CNNs could be applied directly. This method preserves 3D geometric structures but has an expensive computational cost. Geometric details are also lost during the volumetric quantization. To overcome these issues, PointNet [5] was proposed to consume point clouds directly. The follow-up work PointNet++ [19] improves the performance using a hierarchical strategy. More recent works, such as DGCNN [25], PointCNN [14], and KPConv [21] combine graph neural networks and deformable convolutions with point cloud processing. These approaches focus on standard tasks like classification and segmentation. The next section focus on approaches for normal estimation.

Point Cloud Surface Normal Estimation PCA [11] is a classic method for normal estimation. Given a target point and its neighbours, PCA treats these points as samples from a multivariate distribution, of which the three spatial coordinates x, y, z are three independent random variables. By applying eigen-decomposition to the covariance matrix of this point cloud patch, the eigenvector that has the smallest eigenvalue points to the normal direction or its reverse direction. Jet [4] estimates a normal vector by fitting a truncated Taylor expansion to the local point cloud patch. Geometric properties such as surface normal and the two principle curvatures can be computed from the Taylor expression.

Recently, learning-based point cloud normal estimation approaches have been proposed. PCPNet [8] applies PointNet to a local patch and regresses a surface normal directly. To adapt to local geometries better, it also introduces a multi-scale model that runs PointNet [5] at three different local neighbourhood scales. Similarly, Nesti-Net [2] is a Mix of Expert

(MoE) [9, 12] system that processes 7 different scales of neighbours around a target point. Each expert learns to specialise at regions with certain geometric properties and outputs confidences at inference time. To obtain the final result, a manager network integrates normal estimations from all experts and outputs a final result. Both methods tackle normal estimation by exploring more scale levels and balancing results in the multi-scale inferences. This line of approaches achieves better performance than classical PCA and Jet but also requires a large model and a long inference time. The multi-scale setting also prohibits information flow among sub-networks. Our method improves both performance and efficiency by using a single network that learns to select point information softly and can adapt to different geometries without quantizing a local patch to a number of scales.

Attention Mechanism Attention was first introduced by Bahdanau et al. [1] to handle long-range dependencies in language processing tasks together with Recurrent Neural Networks (RNN). Self-attention or intra-attention [7, 15] was proposed to understand relationships within a sequence itself. Vaswani et al. [22] designed multi-head attention and replaced RNNs with self-attention to enable parallel processing, which is difficult to achieve in RNNs. Recently, attention mechanisms have been introduced to the point cloud domain to explore relationships within a point cloud. These methods use attention mechanism to enhance semantic understanding over the entire point cloud. [26, 27] uses attention mechanism to sample points. Wang et al. [23] employs self-attention in semantic segmentation tasks. DCP [24] applies attention mechanism to find correspondences between two point clouds in a registration task. Inspired by prior work, we propose applying attention to normal estimation, a task that requires an understanding of local geometric structures.

3 Method

Given a point cloud, our model predicts a surface normal for every point within a point cloud. To achieve this, our model takes k nearest neighbours around a point, which we call a local patch, as input and outputs the predicted surface normal for this local patch as a three dimensional unit vector. The rest of the section introduces the overall normal estimation pipeline, elaborates our key attention module, and provides implementation details in the end.

Pipeline Our prediction pipeline consists of five steps: **1)** The network takes a local patch $\mathcal{N}_a$ at a point a as input and extract a set of per-point features $\mathbf{F}_a := \{f_i, i \in \mathcal{N}_a\}, \mathbf{F}_a \in \mathbb{R}^{k \times D}$

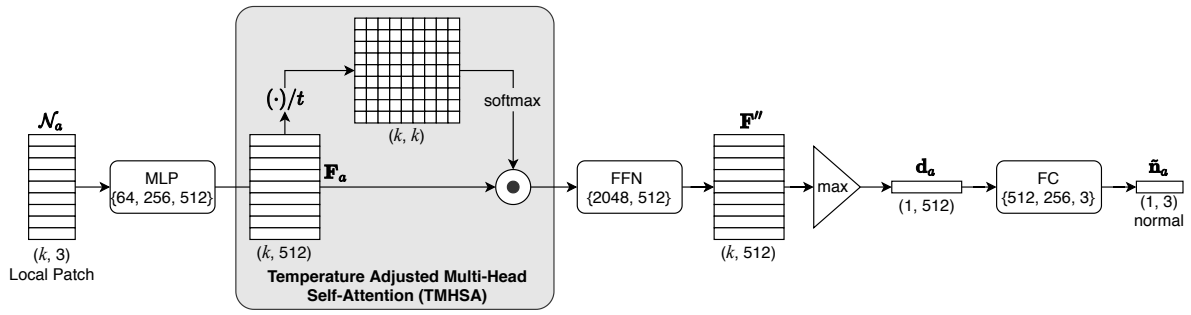


Figure 1: Our attention-based normal estimation model. We introduce TMHSA module that enables our network to focus on relevant neighbours, adapt to various geometric structures, and adjust the attention strength, given a large neighbourhood range. The MLP denotes the Multi-Layer Perceptron and the FFN denotes the Feed Forward Network introduced in [22].

for $\mathcal{N}_a$ using a 3-layer Multi-Layer Perceptron (MLP) where f_i denotes a per-point feature and D is the feature dimension of f_i. **2)** We define a Temperature-adjusted Multi-Head Self-Attention (TMHSA) module and apply it on $\mathbf{F}_a$ extracted in the first step. The TMHSA softly mingles per-point information from different aspects and outputs an integrated feature set $\mathbf{F}' \in \mathbb{R}^{k\times D}$. This is the key step that enables soft neighbourhood selection, points information integration, and attention strength adjustment. **3)** A Feed Forward Network (FFN) that is introduced in [22] is applied to transform $\mathbf{F}'$ to $\mathbf{F}'' \in \mathbb{R}^{k\times D}$ in order to introduce non-linearity. **4)** We apply a point-wise max pooling $g : \mathbb{R}^{k\times D} \to \mathbb{R}^D$, a symmetric function introduced in PointNet [5], on $\mathbf{F}''$ to extract a patch descriptor $\mathbf{d}_a$ for $\mathcal{N}_a$. This patch descriptor is designed to capture the geometric information of this local patch and **5)** a number of fully-connected (FC) layers are applied to the $\mathbf{d}_a$ to finally regress a normal estimation $\tilde{\mathbf{n}}_a$ from $\mathbf{d}_a$. With this pipeline, our model learns to estimate normals for different surface properties and to adjust the softness in the self-attention. The full process is illustrated in Fig. 1.

Training Loss Our network is trained by minimising the unoriented angle difference between estimated normals $\tilde{\mathbf{n}}$ and ground truth normals $\mathbf{n}$. We use the sine distance $\sin(\mathbf{n}, \tilde{\mathbf{n}})$ to measure the unoriented angle difference:

$$L := \sin(\mathbf{n}, \tilde{\mathbf{n}}) = \frac{\|\tilde{\mathbf{n}} \times \mathbf{n}\|}{\|\tilde{\mathbf{n}}\|\|\mathbf{n}\|} \tag{1}$$

TMHSA Module The key component of our network architecture is the TMHSA module, which enables our network to decide which neighbours are useful and how they should be combined. We first introduce the Temperature-adjusted Self-Attention (TSA) module, the building block of the TMHSA module. The TSA takes a set of per-point features around a point as input and produces a set of weighted per-point features. Features that are potentially useful to the task are assigned with higher weights, based on the dot product between features [22]. Specifically, we define TSA as follows:

$$\mathrm{TSA}(\mathbf{F}) := \mathrm{softmax}(\frac{\mathbf{F}_t \mathbf{W}_q \mathbf{W}_k^\top \mathbf{F}_t^\top}{\sqrt{D}})\mathbf{F}_t \mathbf{W}_v \tag{2}$$

where $\mathbf{W}_{q,k,v} \in \mathbf{R}^{D\times D}$ are learnable linear projections that transform $\mathbf{F}$ to other feature spaces and $\mathbf{F}_t = \mathbf{F}/t$ is the temperature-adjusted feature set for a local point patch. In the TSA module, the softmax function produces weights for each feature vector and the temperature t controls the smoothness of the softmax function, for example, the softmax gets sharper when the temperature decreases. By setting t to a learnable parameter, the network can control the sharpness of the softmax function and we denote this behaviour as attention strength adjustment.

The linear transformations $\mathbf{W}_{q,k,v}$ enable the TSA module to attend to one aspect of $\mathbf{F}$. By stacking more linear transformations, we can build the THMSA module that attend to multiple aspects of $\mathbf{F}$:

$$\mathrm{TMHSA}(\mathbf{F}) := \mathrm{concat}_{h\in H}[\mathrm{TSA}_h(F)]\mathbf{W}_o \tag{3}$$

where H is the number of heads or the number of linear transformations we apply to $\mathbf{F}$ and $\mathbf{W}_o$ is a learnable projection matrix that combines outputs from all attention heads.

Implementation Our network is implemented using PyTorch [17]. In the pre-processing stage, we scale all objects to a unit sphere. For each point a, we find its k nearest neighbour and shift this k-point patch to the origin by subtracting the mean of the patch. We use the

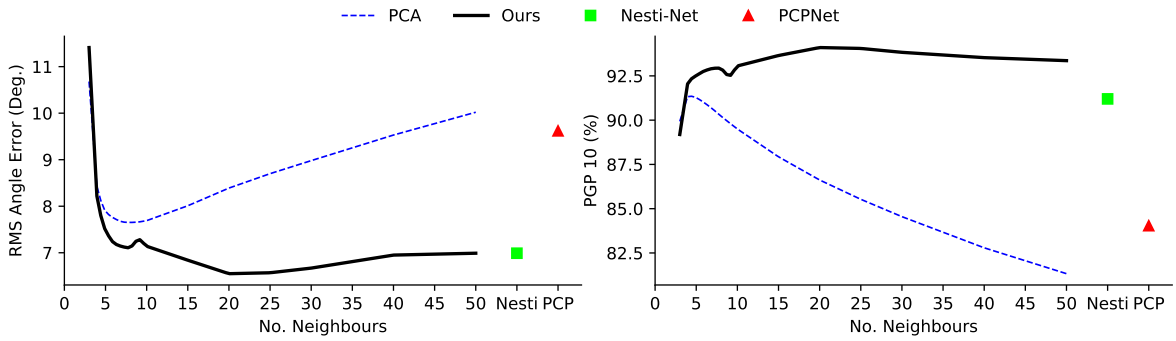


Figure 2: Model performance comparison, given the different number of neighbours. Nesti-Net and PCPNet are plotted on the side since they take multi-scale patches as input.

Adam optimiser with a start learning rate of 5e-4, a momentum set to 0.9 and no weight decay. The learning rate is scheduled to divide by 10 at epoch 400 and 800 and we train the model for 900 epochs in total. We use a batch size of 12000 and found no significant performance difference with other batch sizes. The convolutional layers are initialised using Kaiming Uniform [10] and the temperature parameter is initialised to 1.0.

4 Experiments

In this section, we demonstrate two benefits of our neural network: 1) our method is much less sensitive to the neighbourhood range in comparison with other approaches. 2) our method achieves state of the art performance with a much smaller model and a much faster inference time. We organise this section as follows: Sec. 4.1 introduces the dataset details and evaluation metrics to compare with other approaches. Sec. 4.2 presents our main results on stability, performance, model size, and inference time. Sec. 4.4 discusses the reason why our model is insensitive to the neighbourhood range variance using attention maps. To validate that learning a temperature parameter is helpful, we provide an ablation test in Sec. 4.3. Lastly, Sec. 4.5 provides an application study that uses the ICP algorithm as an example to show how downstream tasks can benefit from the neighbourhood-insensitive property.

4.1 Dataset and Metrics

We train our model using the PCPNet dataset [8] that consists of 27 objects with each object containing 100,000 points. The training/validation/test splits contain 8/3/19 objects respectively. We do not use the noise-augmented data provided in the PCPNet dataset and we follow the train/test splits of PCPNet [8] and Nesti-Net [2] to ensure a fair comparison.

We evaluate our normal estimation network using root mean square error (RMSE) and the proportion of good points (PGPα) metric, the same as Nesti-Nets[2] and PCPNet[8]. Both RMSE and PGP are computed using the unoriented angle error β. In the definition of unoriented angle error, a 179° angle difference is considered as accurate as a 1° angle difference. We compute the unoriented angle error using $\beta = \arccos\left(\left|\frac{\tilde{\mathbf{n}}\cdot\mathbf{n}}{\|\tilde{\mathbf{n}}\|\|\mathbf{n}\|}\right|\right)$ where $\tilde{\mathbf{n}}$ is a predicted normal vector and $\mathbf{n}$ is a ground truth vector. PGPα computes the percentage of angle differences between the predicted result and ground truth is less than $\alpha°$, we compute the PGPα using $\text{PGP}\alpha = \frac{1}{N}\sum_{a\in P}\mathbb{1}(\beta_a < \alpha)$, where $\mathbb{1}(\cdot)$ is an indicator function that yields 1 when the condition inside the operator holds and N is the cardinality of the point cloud P.

4.2 Results

Stability We show that our network is insensitive to the range of neighbourhood in Fig. 2. Our model outperforms PCA by a large margin and most importantly, the performance of our method does not degrade as the number of neighbours increases. Nesti-Net [2] and PCPNet [8] are shown separately in this figure because they use patches from multi-scales.

Our method consistently performs better than PCA and PCPNet as long as more than 3 neighbour points are used, and better than Nesti-Net when more than 15 neighbours are available. PCA achieves the best performance using 7 or 8 neighbours and our model outperforms the top PCA performance when more than 5 neighbours are given. This performance is desirable for downstream algorithms because it ensures that high-quality normal estimations could be found by simply setting a large enough neighbourhood range, and our network can choose neighbours it needs adaptively based on local geometric structures.

State of the Art Performance In addition to achieving a neighbourhood-insensitive property, equipped with the attentional neighbourhood integration model, our method also outperforms other methods by a large margin with a much smaller model and a faster inference time (Tab.1).

Models	PGP 5 (%)	PGP 10 (%)	RMSE (deg)
PCA [11]	81.84	90.23	7.65
Jet [4]	79.05	88.02	7.60
PCPNet [8]	69.86	84.04	9.62
Nesti-Net [2]	80.57	91.20	6.99
Ours	**86.24**	**94.10**	**6.55**

Table 1: Normal estimation results on PCPNet dataset. Our model achieves the state of the art results in all metrics.

Model Size and Inference Time As mentioned above, our model is much smaller and faster than the previous state of the art model Nesti-Net [2], which has 8 sub-networks (7 experts and a manager). Our model takes 41.1MB and Nesti-Net takes 1051.6MB (25x larger). Regarding inference time, our model takes 6.5 seconds to evaluate 100,000 normals (50 neighbours for each normal), while the multi-scale Nesti-Net takes 81.1 seconds. We also report that PCA consumes 0.27 seconds given 8 neighbours, where PCA produces best normal estimations according to Fig. 2. For learning-based models, we benchmark timings on an NVIDIA-1080Ti GPU. For PCA, we use NumPy's SVD function and an Intel i9-9900K CPU. Detailed model sizes and inference time are listed in Tab. 2.

Model	Model Size(MB)	Inference Time (sec / 100K points)
PCA [11]	N/A	0.27 (nb8)
PCPNet [8]	90	165.5
Nesti-Net [2]	1052	81.1
Ours	41	6.5 (nb50)

Table 2: Model sizes and mean inference time. Our model is 25x times smaller and 12x faster than Nesti-Net. The time shown in the table only accounts for forward passing time, i.e. no data loading, neighbour searching, pre-processing and post-processing time included. PCPNet is slow because it learns a quaternion transformation and converts it to a rotation matrix in an inefficient way. nb denotes the number of neighbours used in normal estimation.

4.3 Ablation Study

We verify that introducing a temperature parameter improves the network performance and makes the network more robust to large neighbourhood ranges in Tab. 3 and Fig. 3. By learning a temperature, our network learns to control the softmax in the TMHSA module and to control the attention strength.

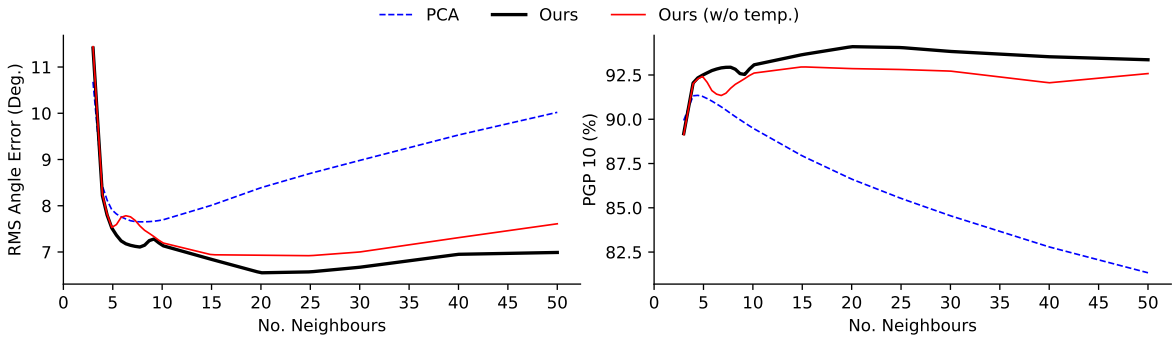


Figure 3: Ablation study on the learnable temperature. Learning a temperature parameter is beneficial to improving stability and performance.

Model	PGP 5 (%)	PGP 10 (%)	RMSE (deg)
w/o temperature	85.50	92.81	6.92
w temperature	**86.24**	**94.10**	**6.55**

Table 3: Ablation study on learning the temperature parameter.

4.4 Attention Weights Analysis

In this section, we discuss why the performance of our network does not degrade when the number of neighbours are increasing, as in PCA. In fact, our network performs better when more neighbours are available. To explain this, we show 4 attention maps from our models in Fig. 4. The four models we use to generate these attention maps are trained and tested using local patches contain 5, 10, 25, and 50 neighbours and achieves PGP10 92.5%, 93.0%, 94.0%, and 93.3%, from left to right, while the best PCA performance is 91.4%. In an attention map, each local patch is represented by a row of pixels and attention weights are colour-coded by pixel intensity.

From the attention map over 5 and 10 neighbours, (a) and (b), we can see that the network pays more attention to neighbours that are close to the patch centre and ignores points farther than 8^{th} neighbours. This agrees with the PCA performance in Fig. 2. From the attention map over 25 and 50 neighbours, (c) and (d), we observe that the network pays extra attention to points at the right-end in a row. These are points far from the patch centre. To illustrate this better, we show the attention weights predicted by our network in 3D in Fig. 5. These attention weights visualisations suggest that the network learns to identify geometric properties around the current point using points from larger scales, by inspecting, for example, whether the patch contains a sharp edge or a corner. Therefore, our network maintains a relatively stable performance despite different numbers of neighbours by focusing on relevant points.

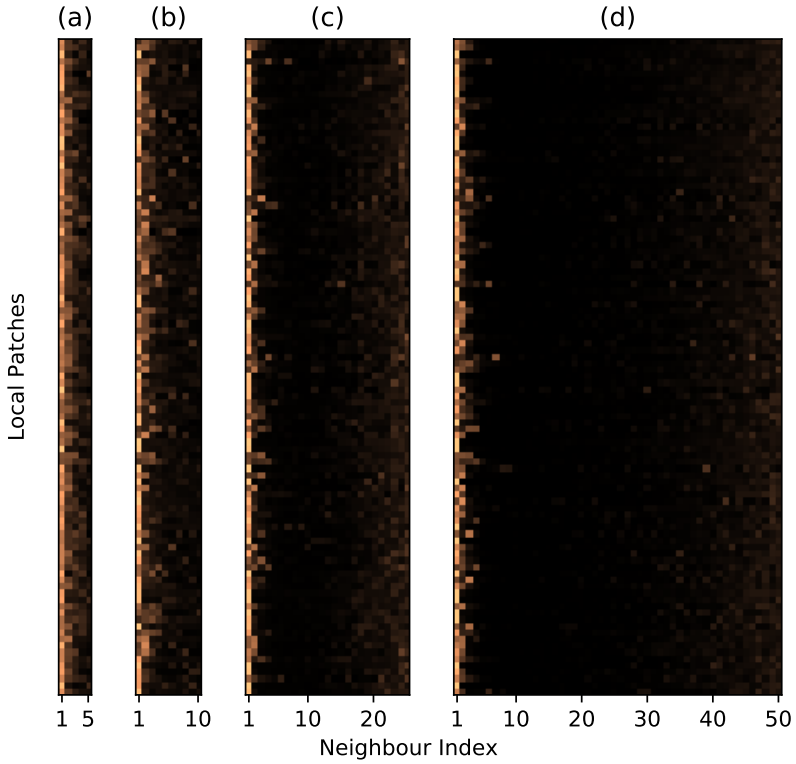


Figure 4: Attention weights visualisation. Each row in the attention map is a local patch that contains 5, 10, 25, 50 neighbours, from to left to right. The x-axis denotes the indices of closest neighbours within a patch, i.e., the point that is the closest to the patch centre is represented by the 1^{st} pixel in a row.

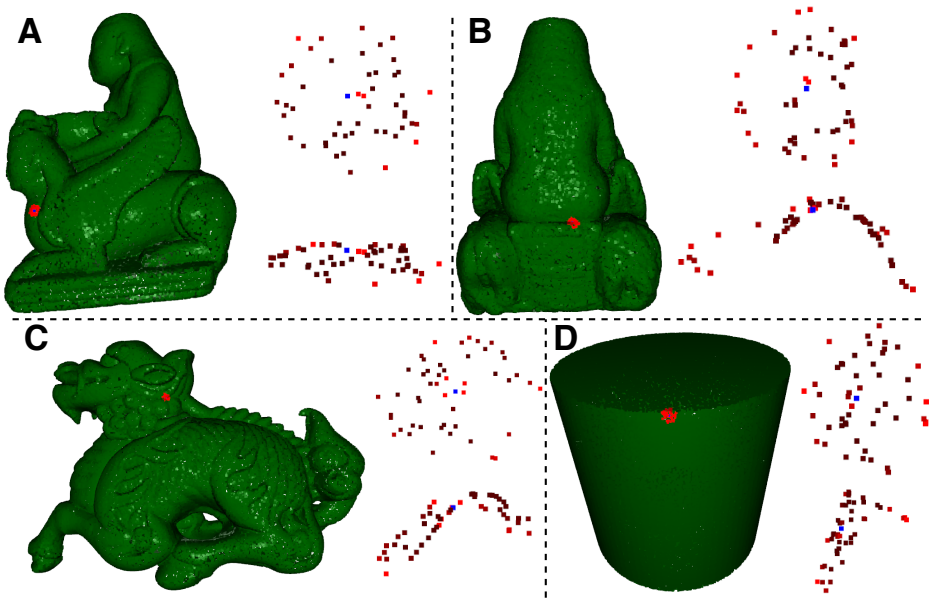


Figure 5: Attention weights on various patches evaluated on the test set. The point of our interest p is coloured with blue and its 49 neighbours are coloured with red. For each patch, we first highlight its points within the entire point cloud, then we show two colour-coded views of the patch, in which brighter red denotes higher attention weight. Patch A is at a flat area while patch B, C, and D contain more complicated geometry structures. The figure is best viewed when zoomed in. We also attached a high resolution version in the supplementary material.

Model	s01	s02	s03	s04	s07	s08	s09	s10	s11	s13	s14	s15	s16	s19	Mean
GT	7	16	6	10	6	7	9	11	7	26	10	10	6	22	10.9
PCA-nb8	15	52	13	23	20	13	10	24	**9**	37	64	16	**7**	F	23.3
PCA-full	14(4)	31(4)	12(4)	16(4)	14(15)	10(3)	**9(7)**	14(7)	**9(3)**	33(3)	17(20)	13(3)	**7(3)**	**12(15)**	15.1(6)
Nesti-Net	35	F	11	F	16	13	12	84	F	F	F	F	F	F	28.5
Ours-nb50	**8**	**17**	**8**	**10**	**7**	**8**	**9**	**11**	9	**29**	**12**	**12**	7	F	**11.3**
Ours-nb25	**8**	**17**	**8**	**10**	**7**	**8**	10	16	**8**	**29**	13	**12**	**6**	F	11.7

Table 4: The number of iterations that the point-to-plane ICP takes to converge, using normals from different sources. s01-s19 denotes the 19 shapes in the test-set of PCPNet [8] dataset. Shapes s05-06, s12, and s17-18 are not shown because ICP fails to converge even using ground truth normals. We denote failure to converge with F. The number in parentheses is the number of neighbours that produces the smallest number of iterations for PCA, for the specific shape. GT denotes ground truth normals. PCA-nb8 denotes the PCA result using the best global $k = 8$ (according to Fig. 2). nb25 and nb50 denote models trained with 25 and 50 neighbours.

4.5 Application Study: ICP Point Cloud Registration

In this section, we demonstrate how other downstream algorithms can benefit from our neighbourhood-insensitive normal estimation network. We take the point-to-plane ICP [3, 6] algorithm as an example and show that, by simply setting the number of neighbours k to a large enough number, for example, $k = 50$ in our network, ICP with our normal estimation converges faster than with other normal estimation methods. Moreover, ICP with classical normal estimation methods like PCA requires fine-tuning k manually to achieve best performance, which is still outperformed by our network.

ICP Preliminary ICP is a widely used point cloud registration and pose estimation method. Many variants have been proposed in past decades. The original point-to-point ICP [3] optimises the distance between closest point pairs directly. The point-to-plane ICP [6] is proposed to improve the converging speed and robustness against noise. Given two point clouds, a source point cloud $\mathbf{X}_s$ and a rigid transformed destination point cloud $\mathbf{X}_d = \mathbf{T}_{ds}\mathbf{X}_s$, ICP computes the rigid transformation $\mathbf{T}_{ds} \in \mathbb{SE}(3)$ to register $\mathbf{X}_d$ and $\mathbf{X}_s$ iteratively. The point-to-point energy is defined as

$$E_{pt-pt} = \sum(\|\mathbf{x}_s - \mathbf{x}_d\|^2) \tag{4}$$

and the point-to-plane energy is defined as

$$E_{pt-plane} = \sum(\mathbf{n}_d \cdot (\tilde{\mathbf{T}}_{ds}\mathbf{x}_s - \mathbf{x}_d))^2 \tag{5}$$

where $\tilde{\mathbf{T}}_{ds}$ denotes the estimated transformation, $\mathbf{x}_s$ is a point in $\mathbf{X}_s$, $\mathbf{x}_d$ is the corresponding point in $\mathbf{X}_d$, and $\mathbf{n}_d$ is the surface normal at $\mathbf{x}_d$.

Results We run the point-to-plane ICP on the test-set of PCPNet dataset [8], using normals from different estimation methods (Tab. 4). With normal estimation from our model, the point-to-plane ICP converges faster than using normals from PCA [11] and Nesti-Net [2]. By making comparison between the nb25 and nb50 models at the last two rows in Tab. 4, we can conclude our model is stable given the different number of neighbours. For PCA, we run an exhaustive search over the different numbers of neighbours and report the smallest iteration number in Tab. 4 (the PCA-full row). The full ICP iteration table for PCA and the full shape names s1-s19 can be found in the supplementary material.

ICP Setup To ensure fair comparisons, we set an early stopping criteria using the point-to-point energy E_{pt-pt} since this stopping criterion is immune to the normal quality when evaluating registration results. A perfect registration should produce $E_{pt-pt} = 0$ but the $E_{pt-plane}$ can still be larger than zero because the normals estimations are not perfect. We set the stopping threshold E^{stop}_{pt-pt} to 10^{-5}. For simplicity, we set the rotation angles between $\mathbf{X}_s$ and $\mathbf{X}_d$ to 10° in all three axes (x, y, z), and the translation between $\mathbf{X}_s$ and $\mathbf{X}_d$ to $[0.01, 0.01, 0.01]$. We scale all point clouds to a unit sphere and shift them to the point cloud centre before running ICP. We use the Levenberg-Marquardt [13, 16] solver to optimise $E_{pt-plane}$. Our ICP code comes along with the training code at https://code.active.vision.

5 Conclusion

In this work, we propose an attention-based normal estimation network that is insensitive to neighbourhood range and provides state of the art normal estimation performance. We achieve this by applying an attention module to the local neighbourhood so our network can focus on relevant points and integrate information softly. With a learnable temperature parameter, our network can also control how attentive it should be under current neighbourhood settings. As a result, our network is 12x faster and 25x smaller than the previous state of the art models [2] and provides 2.9% higher accuracy (PGP 10) on the PCPNet dataset [8]. We also use point-to-plane (ICP) as an application to show that our normal estimations lead to faster convergence than normal estimations from other methods, without manually fine-tuning neighbourhood range parameters.

Acknowledgement

The authors would like to thank Min Chen, Tengda Han, Shuda Li, Tim Yuqing Tang and Shangzhe Wu for insightful discussions and proofreading.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

[2] Yizhak Ben-Shabat, Michael Lindenbaum, and Anath Fischer. Nesti-net: Normal estimation for unstructured 3d point clouds using convolutional neural networks. In *CVPR*, 2019.

[3] Paul J Besl and Neil D McKay. A method for registration of 3-d shapes. *TPAMI*, 1992.

[4] Frédéric Cazals and Marc Pouget. Estimating differential quantities using polynomial fitting of osculating jets. *Computer Aided Geometric Design*, 2005.

[5] Ruizhongtai Qi Charles, Hao Su, Mo Kaichun, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017.

[6] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 1992.

[7] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. In *Empirical Methods in Natural Language Processing*, 2016.

[8] Paul Guerrero, Yanir Kleiman, Maks Ovsjanikov, and Niloy J. Mitra. PCPNet: Learning local shape properties from raw point clouds. *Computer Graphics Forum*, 2018.

[9] John B Hampshire II and Alex Waibel. The meta-pi network: Building distributed knowledge representations for robust multisource pattern recognition. *TPAMI*, 1992.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015.

[11] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics*, 1992.

[12] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural Computation*, 1991.

[13] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 1944.

[14] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *NeurIPS*, 2018.

[15] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. In *ICLR*, 2017.

[16] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of The Society for Industrial and Applied Mathematics*, 1963.

[17] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

[18] Charles Ruizhongtai Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *CVPR*, 2016.

[19] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017.

[20] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *ICCV*, 2015.

[21] Hugues Thomas, Charles Ruizhongtai Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *ICCV*, 2019.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[23] Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. Graph attention convolution for point cloud semantic segmentation. In *CVPR*, 2019.

[24] Yue Wang and Justin M. Solomon. Deep closest point: Learning representations for point cloud registration. In *ICCV*, 2019.

[25] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ToG*, 2019.

[26] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *CVPR*, 2020.

[27] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian. Modeling point clouds with self-attention and gumbel subset sampling. In *CVPR*, 2019.

Neighbourhood-Insensitive Point Cloud Normal Estimation Network (Supplementary)

Zirui Wang
ryan@robots.ox.ac.uk
Victor Adrian Prisacariu
victor@robots.ox.ac.uk

Active Vision Lab
University of Oxford
Oxford, UK

1 ICP Iterations Using PCA

To find the best ICP performance with normals from PCA, we run a set of exhaustive experiments given different numbers of neighbours as input. This is similar to a neighbourhood parameters tuning process and our method can help applications such as ICP to avoid such parameter searching procedures.

Full PCA	s01	s02	s03	s04	s07	s08	s09	s10	s11	s13	s14	s15	s16	s19
nb3	15	71	25	21	15	**10**	10	19	**9**	**33**	F	**13**	**7**	F
nb4	**14**	**31**	**12**	**16**	16	12	11	16	**9**	35	F	14	**7**	F
nb5	15	41	15	F	20	11	11	19	**9**	36	73	14	**7**	F
nb6	18	49	15	19	F	**10**	10	16	**9**	34	27	17	**7**	F
nb7	15	F	18	19	15	13	**9**	**14**	**9**	38	21	19	**7**	38
nb8	15	52	13	23	20	13	10	24	**9**	37	64	16	**7**	F
nb9	14	41	16	18	16	13	**9**	20	**9**	85	22	20	**7**	F
nb10	14	F	14	19	15	41	17	21	**9**	42	34	18	**7**	24
nb15	14	83	15	19	**14**	12	12	20	**9**	**33**	24	15	**7**	**12**
nb20	18	45	14	19	**14**	13	14	18	**9**	55	**17**	F	**7**	21
nb25	14	65	**12**	17	F	F	10	F	**9**	37	35	16	**7**	26
nb30	17	F	20	18	16	F	15	17	**9**	F	63	16	**7**	17
nb40	14	F	18	18	F	F	13	16	**9**	F	28	16	8	F
nb50	25	F	29	17	F	F	10	18	**9**	F	30	22	8	24
Min iter.	14	31	12	16	14	10	9	14	9	33	17	13	7	12

Table 1: Exhaustive number of neighbours search for PCA to achieve the best point-to-plane ICP performance. We reported the minimum iteration numbers for each object in our ICP convergence study, after trying all number of neighbours.

© 2020. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

2 Shape Names s1-s19

s1 galera100k	s2 icosahedron100k	s3 netsuke100k	s4 Cup34100k	s5 sphere100k
s6 cylinder100k	s7 star_smooth100k	s8 star_halfsmooth100k	s9 star_sharp100k	s10 Liberty100k
s11 boxunion2100k	s12 pipe100k	s13 pipe_curve100k	s14 column100k	s15 column_head100k
s16 Boxy_smooth100k	s17 sphere_analytic100k	s18 cylinder_analytic100k	s19 sheet_analytic100k	

3 Attention Weight Visualisation

Here we present a high resolution version of the attention weight visualisation (Fig. 5 in the main paper).

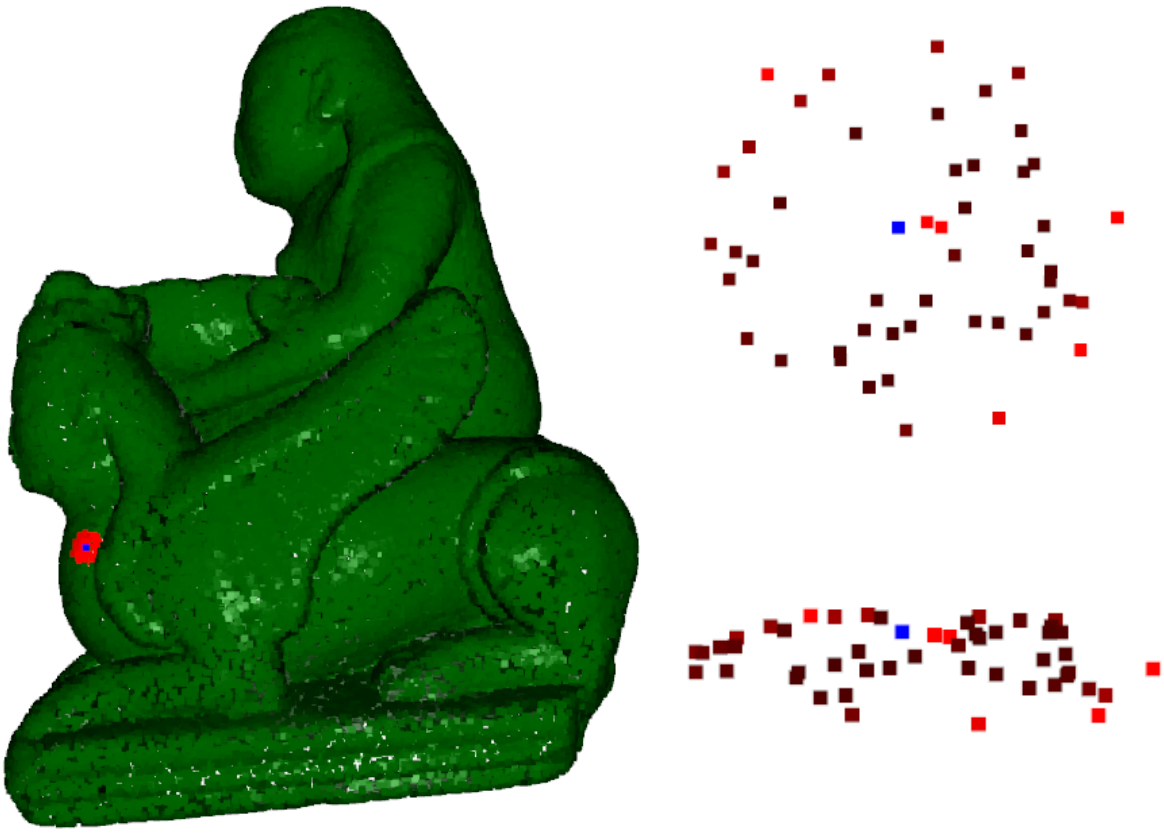


Figure 1: Patch A attention weight visualisation.

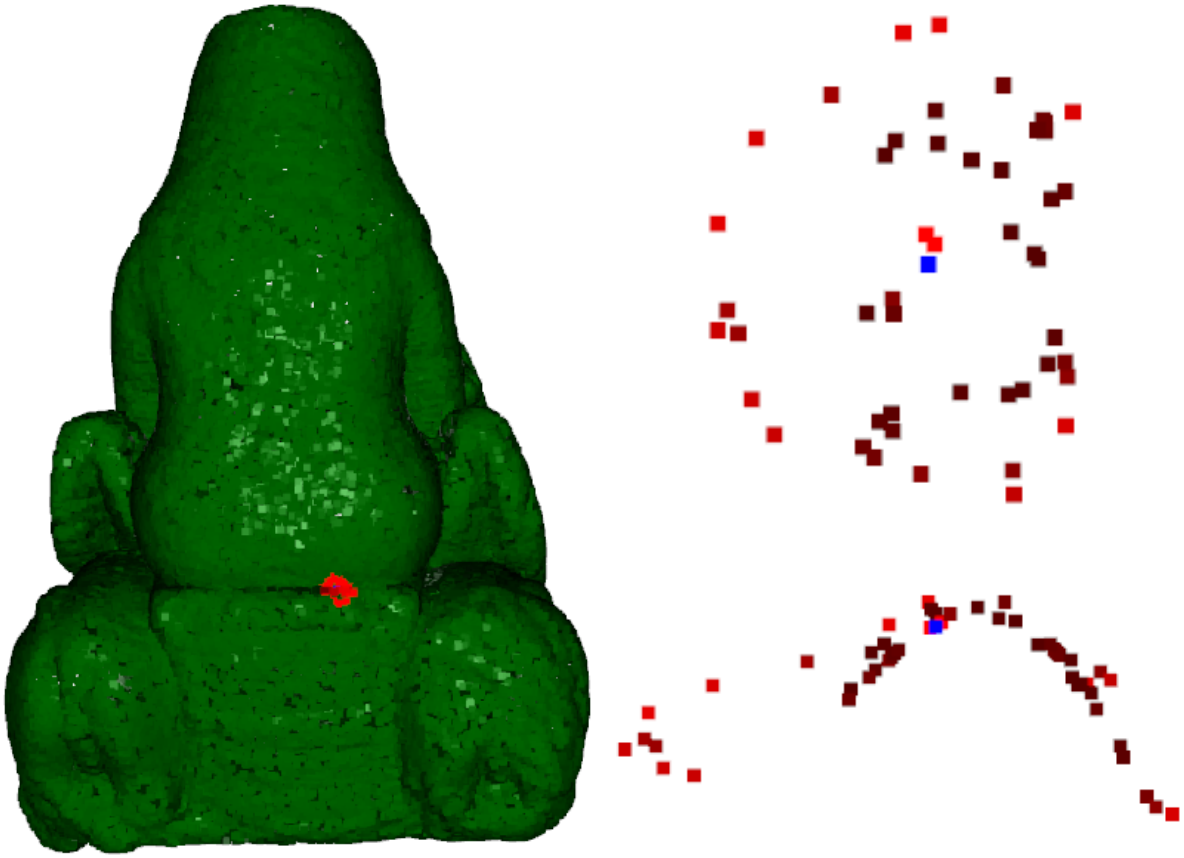


Figure 2: Patch B attention weight visualisation.

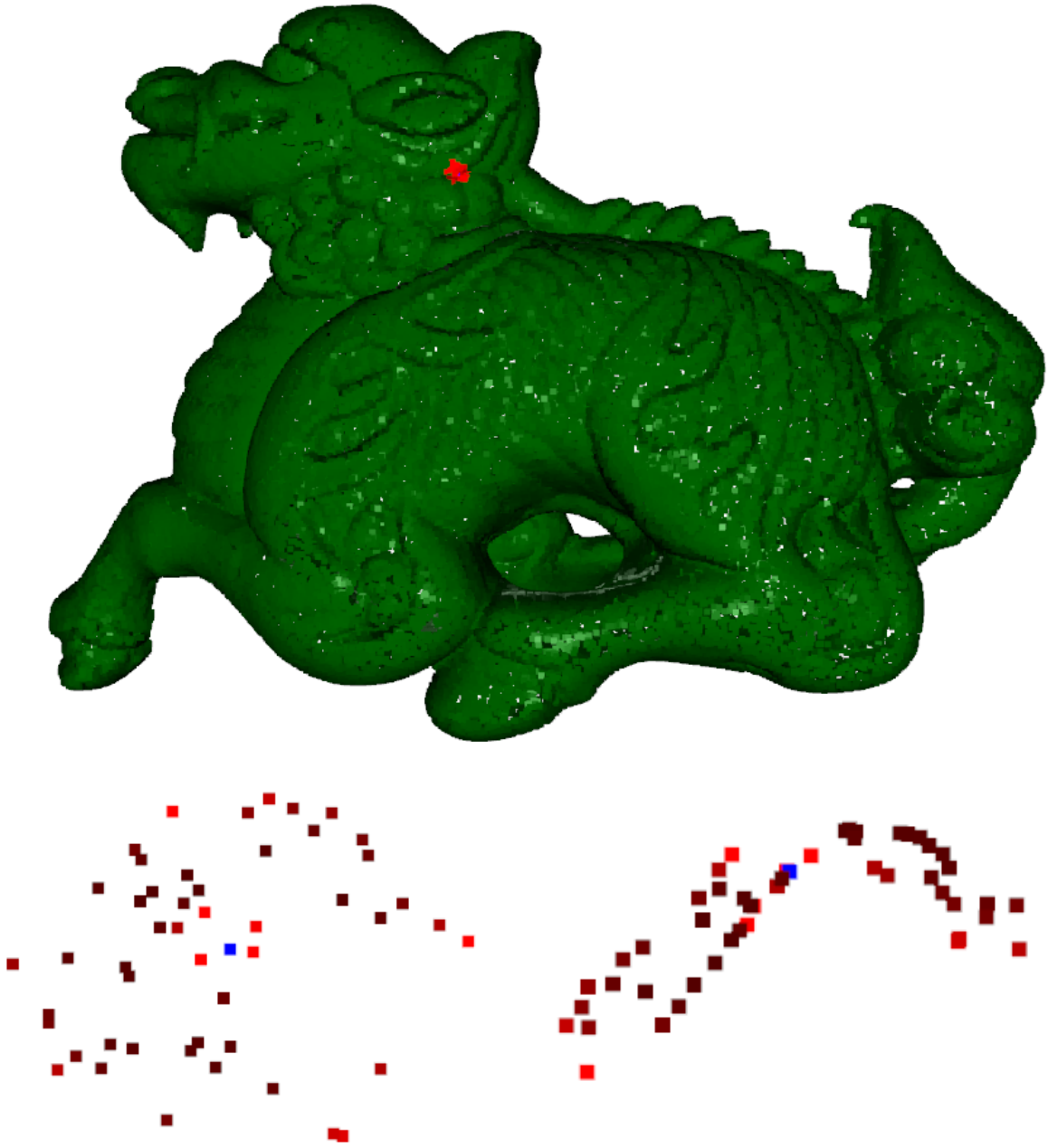


Figure 3: Patch C attention weight visualisation.

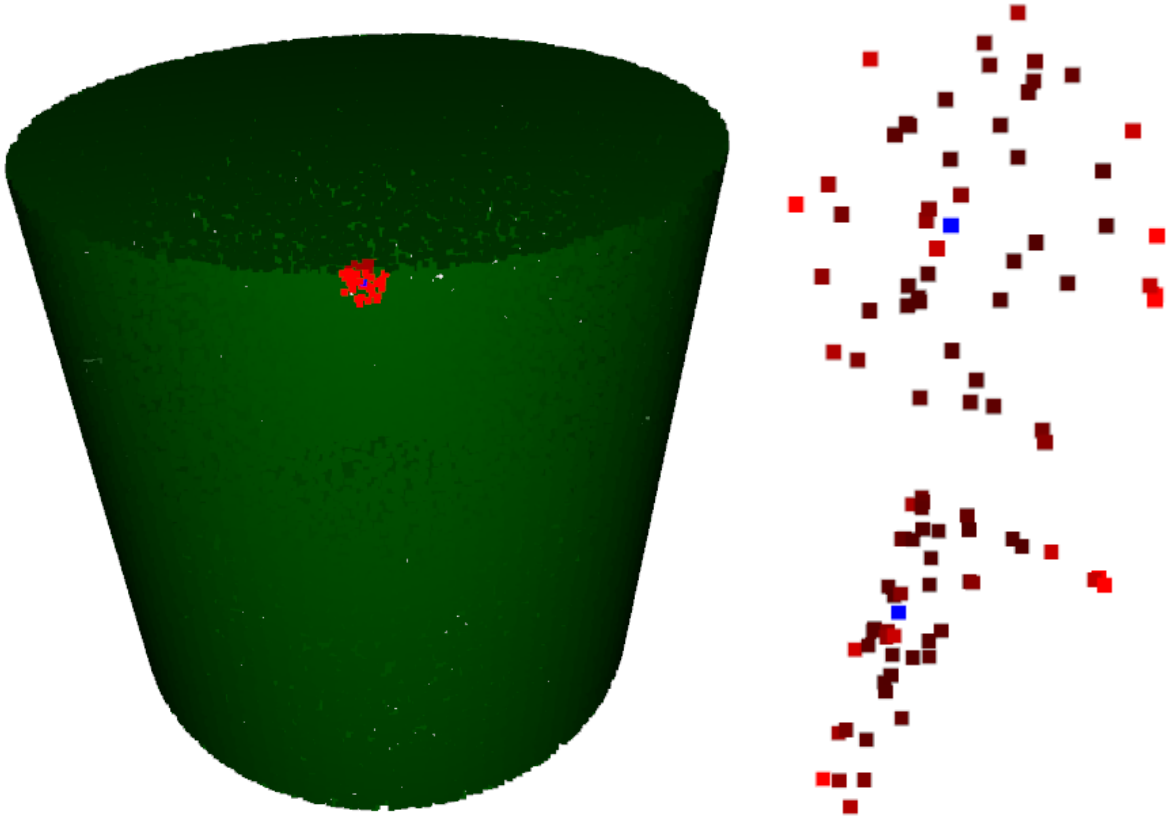


Figure 4: Patch D attention weight visualisation.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Neighbourhood-Insensitive Point Cloud Normal Estimation Network
Publication Status	☑Published □ Accepted for Publication □Submitted for Publication □ Unpublished and unsubmitted work written in a manuscript style
Publication Details	**Zirui Wang**, Victor Adrian Prisacariu. In British Machine Vision Virtual Conference (BMVC), 2020.

Student Confirmation

Student Name:	Zirui Wang		
Contribution to the Paper	First-author contribution: • Conception of research ideas. • Design and implementation of models. • Running of large-scale experiments. • Writing and presentation of the paper.		
Signature	王梓睿	Date	20/Sep/2023

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	Professor Victor Adrian Prisacariu		
Supervisor comments:	The description is accurate.		
Signature	VPA	Date	20/Sep/2023

This completed form should be included in the thesis, at the end of the relevant chapter.

Chapter 4

3D – 2D Registration with NeRF

This chapter addresses the problem of aligning a set of 2D images with a global 3D model, namely 3D – 2D registration. Representative approaches in this field include Perspective-n-Point (PnP) [31, 32, 33], absolute pose regression (APR) [47, 148, 149], scene coordinate networks [150, 151, 152], and photometric methods [34, 136, 137, 35, 139]. However, these methods primarily rely on classical representations such as point cloud and voxel.

Recently, the advent of Neural Radiance Fields (NeRF) has significantly improved performance in various 3D tasks, including novel view synthesis [14, 15, 92], 3D reconstruction [88], and generative models [112]. One limitation in NeRF training is that it often requires a time-consuming pre-processing step to estimate camera parameters for input images, for example, camera extrinsics and intrinsics.

This chapter presents a series of two papers aimed at eliminating the need for this pre-processing step. In Section 4.1, we propose to optimise camera parameters along with NeRF training. In specific, our method consider camera parameters as learnables during NeRF training and update them with photometric loss during NeRF training. Since the photometric loss is shared with the original NeRF pipeline, our method add little overhead to NeRF training, while removing the

pre-processing step.

In Section 4.2, we propose to extend Section 4.1 to handle larger camera motion by leveraging monocular depth. We integrate monocular depth with joint camera and NeRF training pipeline from three perspectives: i) providing geometry prior to NeRF training, ii) supporting a 3D registration between depth projected point clouds, and iii) enabling surface rendering in addition to original volumetric rendering. Since monocular depth can be estimated efficiently, our method introduces little overhead while dramatically enhance the performance under large camera motion.

In summary, we propose a line of work in joint camera parameter estimation and NeRF optimisation, contributing to the theme of 3D – 2D registration with NeRF representation.

4.1 NeRF−−: Neural Radiance Fields Without Known Camera Parameters

In this section, we present a paper that focuses on aligning 2D images with Neural Radiance Fields (NeRF) volumes. Standard NeRF training employs ray casting and volumetric rendering to achieve multi-view photometric consistency. This approach inherently depends on pre-computed camera parameters.

In this work, we eliminate the requirement for such pre-computation, streamlining the NeRF training process. Our approach treats camera parameters as trainable variables during NeRF optimisation. This allows us to recover camera parameters by back-propagating the photometric error in the ray-casting procedure. As a result, our method integrates seamlessly with the existing NeRF training pipeline, introducing minimal computational overhead while providing comparable novel view synthesis quality to the original NeRF method that requires

pre-defined camera parameters.

Moreover, we contribute a synthetic dataset to aid in evaluating the performance of concurrent camera parameter and NeRF optimisation. Existing datasets fall into two main categories: one includes real-life images with pseudo ground truth camera parameters obtained from SfM [9] or SLAM [6] systems; the other contains ground truth camera parameters but with over-simplified synthetic scenes. Our dataset fills this gap by offering both accurate ground truth and photo-realistic scenes, thereby facilitating assessment of joint estimation techniques.

NeRF−−: Neural Radiance Fields Without Known Camera Parameters

Zirui Wang
Active Vision Lab
University of Oxford
ryan@robots.ox.ac.uk

Shangzhe Wu
Visual Geometry Group
University of Oxford
szwu@robots.ox.ac.uk

Weidi Xie
Visual Geometry Group
University of Oxford
weidi@robots.ox.ac.uk

Min Chen
e-Research Centre
University of Oxford
min.chen@oerc.ox.ac.uk

Victor Adrian Prisacariu
Active Vision Lab
University of Oxford
victor@robots.ox.ac.uk

Abstract

Considering the problem of novel view synthesis (NVS) from only a set of 2D images, we simplify the training process of Neural Radiance Field (NeRF) on forward-facing scenes by removing the requirement of known or pre-computed camera parameters, including both intrinsics and 6DoF poses. To this end, we propose *NeRF−−*, with three contributions: *First*, we show that the camera parameters can be jointly optimised as learnable parameters with NeRF training, through a photometric reconstruction; *Second*, to benchmark the camera parameter estimation and the quality of novel view renderings, we introduce a new dataset of path-traced synthetic scenes, termed as Blender Forward-Facing Dataset (BLEFF); *Third*, we conduct extensive analyses to understand the training behaviours under various camera motions, and show that in most scenarios, the joint optimisation pipeline can recover accurate camera parameters and achieve comparable novel view synthesis quality as those trained with COLMAP pre-computed camera parameters. Our code and data is available at https://nerfmm.active.vision.

1 Introduction

Generating photo-realistic images from arbitrary viewpoints not only requires to understand the 3D scene geometry but also complex viewpoint-dependent appearance resulted from sophisticated light transport phenomena. One way to achieve this is by constructing a 5D plenoptic function that directly models the light passing through each point in space [1] (or a 4D light field [2, 3] if we restrict ourselves outside the convex hull of the objects of interest). Unfortunately, physically measuring such a plenoptic function is usually not feasible in practice. As an alternative, Novel View Synthesis (NVS) aims to generate the unseen views from a small set of images captured from diverse viewpoints.

Recently, Neural Radiance Fields (NeRF) [4, 5, 6, 7] and Multi-Plane Images (MPI) [8, 9, 10, 11, 12, 13] have demonstrated that, by optimising a volumetric scene representation on a sparse set of images, one can re-render the scene from novel viewpoints with impressive visual quality, including sophisticated view-dependent effects, such as specularity and transparency.

One limitation for these methods is the requirement of the camera parameters of the images at the training time, which is rarely accessible in real scenarios, *e.g.,* images taken with a mobile phone. In practice, these methods usually pre-compute the camera parameters via conventional techniques. For example, NeRF and its variants [4, 5, 6, 7] adopt COLMAP [14] to estimate the camera parameters (both intrinsics and extrinsics) for each input image. This pre-processing step,

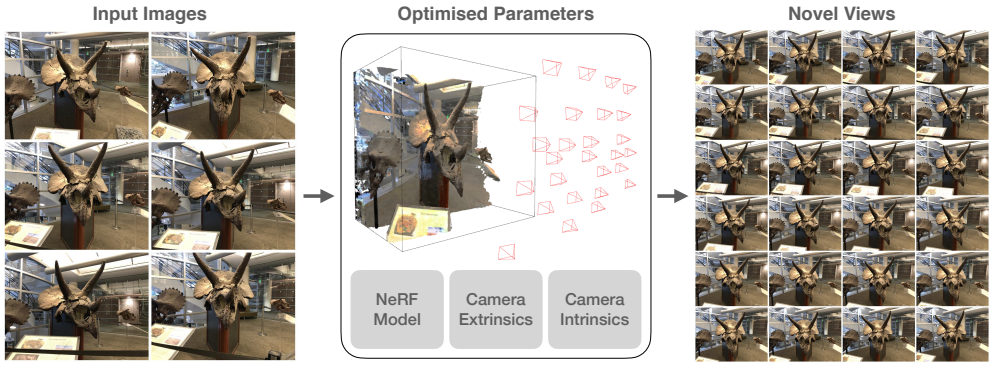


Figure 1: We propose *NeRF−−*, a NeRF-based framework for novel view synthesis without pose supervision. Specifically, our method jointly optimises camera poses and intrinsics of a set of forward-facing input images while training a NeRF model.

apart from introducing additional complexity, can be potentially erroneous or simply fail, due to homogeneous regions, fast-changing view-dependent appearance, or degenerate solutions in linear equations from highly ambiguous camera trajectories [15, 16, 17].

Our goal in this paper is to investigate novel view synthesis for forward-facing scenes with unknown camera parameters. We introduce *NeRF−−*, a framework that treats camera poses and intrinsics as learnable parameters, and jointly optimises them with a 3D scene representation (as shown in Fig. 1). Given only a sparse set of images captured in a forward-facing setup, our system can be trained 1) to estimate camera parameters for each image and 2) to learn a volumetric scene representation through a photometric reconstruction loss, achieving high fidelity NVS results comparable to existing two-stage COLMAP-NeRF pipelines while removing the pre-processing step for camera parameters.

In order to establish a comprehensive evaluation of the method, we make two additional contributions: 1) we introduce a high-quality path-traced synthetic dataset, named Blender Forward Facing (BLEFF), to benchmark camera parameter estimation and novel view synthesis quality, which is publicly accessible for future research; 2) we conduct thorough analyses on our method and COLMAP-NeRF under various camera motions, showing that both systems can tolerate up to $\pm 20^\circ$ of rotation and $\pm 20\%$ translation perturbations, and that the joint optimisation is more favourable than COLMAP in translational perturbations, but less competitive in rotational perturbations.

2 Related Work

We divide the related work roughly into two categories, one assuming camera parameters are input to NVS representations, and the other jointly estimating camera parameters with an NVS representation from uncalibrated images.

With Known Camera Parameters Images and their camera parameters are usually required by many novel view synthesis systems. In this line of work, classical methods Light Field Rendering [3] and Lumigraph [2] model a simplified plenoptic function. [18] integrates sophisticated hand-crafted material BRDF models into Signed Distance Functions (SDF). FVS[19] and SVS[20] combine meshes and deep features from images. Recently, dense volumetric representations have been proposed to enable smooth gradients for photometry-based optimisation and has shown to be promising for photo-realistic novel view synthesis of highly complex shapes and view-dependent appearance. These representations include Soft3D[21], Multi-Plane Images (MPI)[8, 9, 10, 11, 12, 13], Scene Representation Networks (SRN)[22], implicit surface [23, 24], and Neural Radiance Fields (NeRF)[4, 6, 7, 5]. Despite these approaches differ in the way of scene representations, they all require input images to be associated with known camera parameters.

With Unknown Camera Parameters The second line of work, mostly consisting of SLAM and SfM systems, aims to reconstruct the scene directly from RGB images by jointly estimating camera parameters and 3D geometries. For example, MonoSLAM[25] and ORB-SLAM [26] reconstruct a point cloud map and estimate camera poses in real-time by associating feature correspondences, Similarly, DTAM [27], SVO[28], and LSD-SLAM [29] approach the SLAM task by minimising a photometric loss. SfM systems Bundler[30] and COLMAP [14] can reconstruct a global consistent

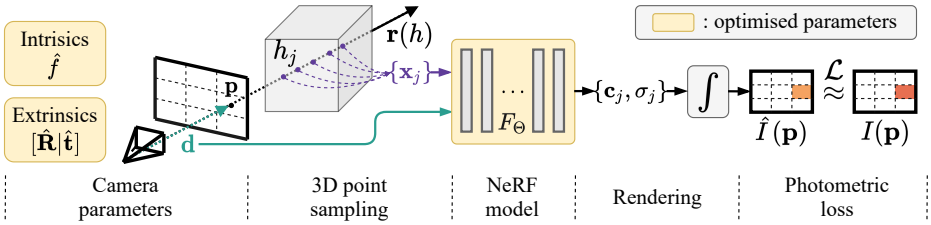


Figure 2: Our pipeline jointly optimises a NeRF model and the camera parameters of the input images by minimising the photometric reconstruction errors. To render a pixel $\mathbf{p}$ from NeRF, given the optimised camera parameters ($\hat{f}$, $\hat{\mathbf{R}}$, $\hat{\mathbf{t}}$), we feed 3D points $\mathbf{x}_j$ sampled along the camera ray together with the viewing direction $\mathbf{d}$ into NeRF F_Θ, and aggregate the output radiance $\mathbf{c}_j$ and densities σ_j to obtain its colour $\hat{I}(\mathbf{p})$. The entire pipeline can be trained end-to-end using only RGB images with unknown cameras as input.

map and estimate camera parameters for large image sets, but sensitive to initial image pairs. Although these methods only require RGB images as input, they often assume diffuse surface appearance in order to establish correspondences and cannot recover view-dependent appearance, hence resulting in unrealistic novel view rendering.

Concurrent to our work, several methods [31, 32, 33] have developed similar ideas on estimating camera parameters with a NeRF model. These methods either requires intrinsics as input [32, 33], or assumes a pre-trained NeRF is available [31], In contrast, we propose to jointly optimise camera poses, intrinsics and a NeRF in an end-to-end manner on forward-facing scenes. As a result, we show the proposed framework is able to produce photo-realistic view synthesis that are comparable to two-stage NVS systems.

3 Preliminary

Given a set of images $\mathcal{I} = \{I_1, I_2, \dots, I_N\}$ captured from N sparse viewpoints of a scene, with their associated camera parameters $\Pi = \{\pi_1, \pi_2, \dots, \pi_N\}$, including both intrinsics and 6DoF poses, the goal of novel view synthesis is to come up with a scene representation that enables the generation of realistic images from novel and unseen viewpoints.

In this paper, we follow the approach proposed in NeRF [4]. Specifically, NeRF adopts an neural representation to construct a radiance field from sparse input views, where the view-dependent appearance is modelled by a continuous function $F_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$, which maps a 3D location $\mathbf{x} = (x, y, z)$ and a viewing direction $\mathbf{d} = (\theta, \phi)$ to a radiance colour $\mathbf{c} = (r, g, b)$ and a density σ.

To render an image from the NeRF model, the colour at each pixel $\mathbf{p} = (u, v)$ on the image plane $\hat{I}_i$ is obtained by a rendering function $\mathcal{R}$, aggregating the radiance along a ray shooting from the camera position $\mathbf{o}_i$, passing through the pixel $\mathbf{p}$ into the volume [34, 2]:

$$\hat{I}_i(\mathbf{p}) = \mathcal{R}(\mathbf{p}, \pi_i|\Theta) = \int_{h_n}^{h_f} T(h)\sigma\left(\mathbf{r}(h)\right)\mathbf{c}\left(\mathbf{r}(h), \mathbf{d}\right)\, dh, \tag{1}$$

where $T(h) = \exp(-\int_{h_n}^{h} \sigma(\mathbf{r}(s))\, ds)$ denotes the accumulated transmittance along the ray, *i.e.,* the probability of the ray travelling from h_n to h without hitting any other particle, and $\mathbf{r}(h) = \mathbf{o} + h\mathbf{d}$ denotes the camera ray that starts from camera origin $\mathbf{o}$ and passes through $\mathbf{p}$, controlled by the camera parameter π_i, with near and far bounds h_n and h_f. In practice, the integral in Eq. (1) is approximated by accumulating radiance and densities of a set of sampled points along a ray. With this implicit scene representation $F_\Theta(\mathbf{x}, \mathbf{d})$ and a differentiable renderer $\mathcal{R}$, NeRF can be trained by minimising the photometric error $\mathcal{L} = \sum_i^N ||I_i - \hat{I}_i||_2^2$ between the observed views and synthesised ones $\hat{\mathcal{I}} = \{\hat{I}_1, ..., \hat{I}_N\}$ under known camera parameters:

$$\Theta^* = \arg\min_\Theta \mathcal{L}(\hat{\mathcal{I}}|\mathcal{I}, \Pi). \tag{2}$$

To summarise, NeRF represents a 3D scene as a radiance field parameterised by MLPs, which is trained by minimising the discrepancy between the observed and rendered images. Note that, the camera parameters π_i for these given images are required for training, which are usually estimated by SfM packages, such as COLMAP [14]. For more details of NeRF, we refer the readers to [4].

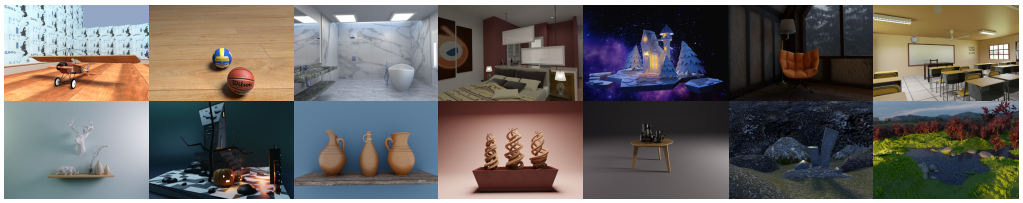


Figure 3: Thumbnails of BLEFF dataset.

4 Method

In this paper, we investigate the training framework for novel view synthesis without known camera parameters. Our framework jointly optimises the camera parameters and a scene representation from a set of input RGB images $\mathcal{I}$. We make two assumptions: *First*, all images are captured in a forward-facing setup with certain amount of rotation and translation flexibility, as described in LLFF [10] and original NeRF[4]. *Second*, all images are captured with same intrinsic parameters. Mathematically, our proposed framework *NeRF−−* can be formulated as:

$$\Theta^*, \Pi^* = \arg\min_{\Theta,\Pi} \mathcal{L}(\hat{\mathcal{I}}, \hat{\Pi}|\mathcal{I}), \tag{3}$$

where Π refers to the camera parameters, including both intrinsics and 6DoF poses. In addition to simplifying the original two-stage approach, this joint optimisation approach enables globally consistent reconstruction results, similar to the bundle adjustment used in conventional SfM pipelines [35] and SLAM systems [25, 27, 29].

In the following sections, we first introduce the representations for the camera parameters and then describe the process of the joint optimisation.

4.1 Camera Parameters

Camera Intrinsics can be expressed as a focal length f and principle points c_x and c_y for a pinhole camera model. For simplicity, we consider the centre of sensor as the camera principle points, *i.e.,* $c_x \approx W/2$ and $c_y \approx H/2$, where H and W denote the height and the width of the image, and all input images are taken by the same camera. Thus, estimating the camera intrinsics only refers to finding the focal length f in this case.

Camera Poses can be expressed as a camera-to-world transformation matrix $\mathbf{T}_{wc} = [\mathbf{R}|\mathbf{t}]$ in SE(3), where $\mathbf{R} \in$ SO(3) and $\mathbf{t} \in \mathbb{R}^3$ denote the camera rotation and translation respectively. Specifically, optimising the translation vector $\mathbf{t}$ can be as simple as setting it as trainable parameters as it is defined in the Euclidean space. Whereas for the camera rotation, which is in SO(3) space, we adopt the axis-angle representation: $\boldsymbol{\phi} := \alpha\omega$, $\boldsymbol{\phi} \in \mathbb{R}^3$, where ω and α denote a normalised rotation axis and a rotation angle α respectively. A rotation matrix $\mathbf{R}$ can be recovered from the Rodrigues' formula $\mathbf{R} = \mathbf{I} + \frac{\sin(\alpha)}{\alpha}\boldsymbol{\phi}^\wedge + \frac{1-\cos(\alpha)}{\alpha^2}(\boldsymbol{\phi}^\wedge)^2$, where $(\cdot)^\wedge$ is the skew operator that converts a vector $\boldsymbol{\phi}$ to a skew matrix. Till here, we can optimise the camera poses for each input image I_i, with trainable parameters $\boldsymbol{\phi}_i$ and $\mathbf{t}_i$.

4.2 Joint Optimisation of Camera Parameters and Scene Representation

Now we introduce the details for training *NeRF−−*, which jointly optimises camera parameters along with a NeRF model on a sparse set of forward-facing views.

Specifically, to render a pixel p on a given image I_i, we shoot a ray $\hat{\mathbf{r}}_{i,p}(h) = \hat{\mathbf{o}}_i + h\hat{\mathbf{d}}_{i,p}$, from the camera position through the pixel p into the radiance field, with the current estimates of the camera parameters $\hat{\pi}_i = (\hat{f}, \hat{\boldsymbol{\phi}}_i, \hat{\mathbf{t}}_i)$, and the ray direction

$$\hat{\mathbf{d}}_{i,p} = \hat{\mathbf{R}}_i \begin{pmatrix} (u - W/2)/\hat{f} \\ -(v - H/2)/\hat{f} \\ -1 \end{pmatrix}, \tag{4}$$

where $\hat{\mathbf{o}}_i = \hat{\mathbf{t}}_i$ and $\hat{\mathbf{R}}_i$ is computed from $\hat{\boldsymbol{\phi}}_i$ using the Rodrigues' formula. Along the ray, we sample a number of 3D points $\{\mathbf{x}_j\}$ and evaluate the radiance colours $\{\mathbf{c}_j\}$ and the density values $\{\sigma_j\}$ of

these points from the NeRF model F_Θ. To get the colour for that pixel, the rendering function Eq. (1) is applied to aggregate the predicted radiance and densities along the ray.

During training, we randomly render M pixels for each input image, and minimise the reconstruction loss between the rendered colours against the ground-truth colours on these pixels. Note that, the entire pipeline is fully differentiable, which enables the jointly optimisation of NeRF and the camera parameters $\{\pi_i\}$ by minimising the reconstruction loss.

5 Blender Forward Facing Dataset

Evaluating the performance of a joint optimisation system like ours requires both images and their ground truth camera parameters, which are not easy to acquire for real captured images. To that end, we introduce a synthetic dataset, Blender Forward Facing (BLEFF), containing 14 path-traced scenes[1], with each rendered in multiple levels of rotation and translation perturbations. We will make this dataset publicly available, expecting to foster future research on the joint optimisation of camera parameters and scene representations.

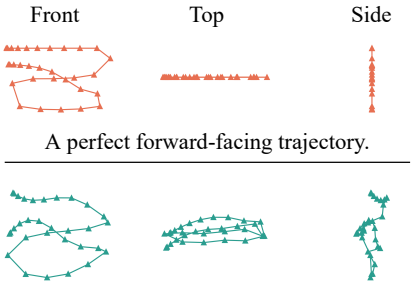


Figure 4: Dataset trajectory generation.

We now detail how we create the dataset. Starting with a perfect forward-facing camera trajectory (top row in Fig. 4), *i.e.,* all cameras looking forward and moving in xy-plane (z=0), we generate 16 trajectory variants for each scene by adding 5/5/6 levels of rotation/translation/full-6DoF perturbations on the perfect trajectory. We denote these trajectory variants via a $t_{aaa}r_{bbb}$ notation, where t_{aaa} and r_{bbb} denote $\pm aaa\%$ of translation and $\pm bbb$ degree of rotation perturbed, respectively. Since each synthetic scene is in different scale, we add translation perturbation proportional to the maximum dimension of a trajectory, hence the $\%$ notation. Fig. 3 shows image examples of BLEFF and Fig. 4 presents a t_{010} trajectory example.

Background It is surprisingly difficult to find suitable datasets to jointly evaluate the performance of camera parameters and a NVS model at the same time. In general, three types of dataset are available, but cannot meet our requirements. The first type focuses on NVS rendering quality with their camera parameters estimated from SfM and SLAM packages[14, 26]. These datasets include LLFF-NeRF[10], RealEstate10K[8], Spaces[12], Shiny[13], and Tanks & Temples[36]. Without ground truth camera parameters, it is impossible to evaluate the camera pose estimation accurately. The second type is designed to benchmark SLAM systems, *e.g.,* TUM-RGBD[37] and ETH3D[38], with "ground truth" camera parameters being estimated from an external motion capture system. However, we find the time lags between motion capture system and the captured images are unignorable in NVS tasks and often lead to blurry renderings. The third type, *e.g.,* SceneNet RGB-D[39] and InteriorNet[40], offers large-scale simulated indoor images but lacks of scene diversities.

6 Experiments

6.1 Setups

Dataset Including our BLEFF dataset, we evaluate *NeRF−−* on three datasets: **LLFF**: We first conduct experiments on the same forward-facing dataset as that in NeRF, namely, LLFF-NeRF [10], which has 8 forward-facing scenes captured by mobile phones or consumer cameras, each containing 20-62 images. In all experiments, we follow the official pre-processing procedures and train/test splits, *i.e.,* the resolution of the training images is 756×1008, and every 8-th image is used as a test image. **RealEstate10K**: To understand the behaviour of NVS in real-life capture scenarios, we additionally collected a number of diverse scenes extracted from the short video segments in RealEstate10K [8]. In particular, the image resolution in these sequences is 1080×1920 and the frame rate is 30 fps. We sub-sample the frames and reduce the frame rates to 5 fps, with each sequence containing 14-43 images. **BLEFF**: We use BLEFF to evaluate camera parameter estimation accuracy

[1]We modified open sourced blender files downloaded from https://www.blendswap.com/. Most blender files come with CC-BY-3.0 licence. Details see our supplementary.

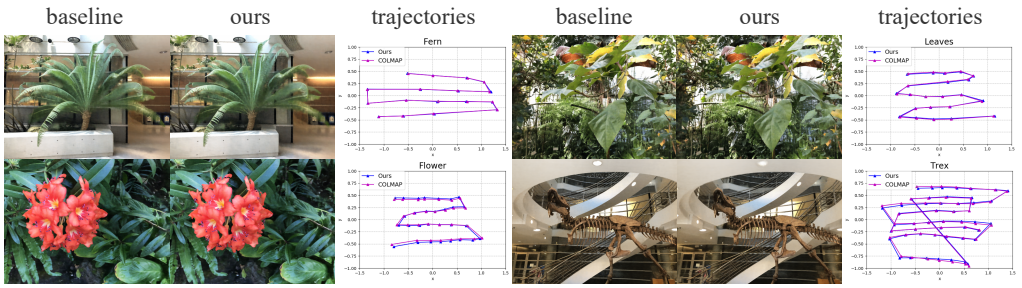


Figure 5: Qualitative comparison between our *NeRF−−* model with unknown cameras and the *baseline NeRF* on LLFF-NeRF dataset. For each example, on the left, we show the synthesised novel views from the *baseline NeRF* and from our model; on the right, we compare our optimised camera trajectories with the ones estimated from COLMAP, aligned with a Sim(3) transformation. Our proposed *NeRF−−* recovers similar camera poses with COLMAP and produces high quality novel views comparable to the *baseline NeRF*. Full results can be found in the supplementary.

and NVS rendering quality, as well as to explore the breaking point of the *NeRF−−* and *baseline NeRF*. All scenes in BLEFF come with 31 images in 1040×1560 resolution, with 27 for training and 4 for testing, and focal lengths set to 1040 pixels in both horizontal and vertical directions. In the experiments below, all BLEFF images are downsized by half, *i.e.,* images are in 520×780 and the ground truth focal length f_{gt} is 520 for training and testing.

Metrics We evaluate the proposed framework from two aspects: First, to measure the quality of novel view rendering, we adopt common metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure(SSIM) [41] and Learned Perceptual Image Patch Similarity (LPIPS) [42]; Second, we evaluate the accuracy of the optimised camera parameters, including focal length, rotation and translation. For focal length evaluation, we report the absolute error in the metric of pixels. For the camera poses, we follow the evaluation protocol of Absolute Trajectory Error (ATE) [43, 37], aligning two sets of pose trajectories globally with a similarity transformation Sim(3) and reporting the rotation angle between two rotations and the absolute distance between two translation vectors.

Implementation Details We implement our framework in PyTorch following the same architecture as original NeRF, except that, for computation efficiency, we: (a) do not use the hierarchical sampling strategy; (b) reduce the hidden layer dimension from 256 to 128; and (c) sample only 1024 pixels from every input image and 128 points along each ray. We use Kaiming initialisation [44] for the NeRF model, and initialise all cameras to be at origin looking at $-z$ direction, with focal length f to be the image width. We use three separate Adam optimisers for NeRF, camera poses and focal lengths respectively, all with an initial learning rate of 0.001. The learning rate of the NeRF model is decayed every 10 epochs by multiplying with 0.9954 (equivalent to a stair-cased exponential decay), and learning rates of the pose and focal length parameters are decayed every 100 epochs with a multiplier of 0.9. All models are trained for 10000 epochs unless otherwise specified. More technical details are included in the supplementary material.

6.2 Results

6.2.1 On Novel View Synthesis Quality

We compare the novel view rendering results from *baseline NeRF*, where camera parameters are estimated from COLMAP, and our proposed model *NeRF−−*, which jointly optimises the camera parameters and the 3D scene representation. We report the quantitative evaluations in Table 1 and visual results in Fig. 5 and Fig. 6. Overall, our joint optimisation model, which does not require camera parameters as inputs, achieves similar NVS quality compared to the *baseline NeRF* model (perceptual quality metric ΔSSIM and Δ LPIPS = 0.05). This confirms



Figure 6: Qualitative comparison of NVS quality between our model and the baseline on RealEstate10K. Our model produces better results on the first examples but slightly worse results on the second example, indicating comparable performance overall compared to baseline.

Scene	SSIM↑		LPIPS↓		PSNR↑		Camera Parameters Difference		
	colmap	ours	colmap	ours	colmap	ours	Δrot (deg)	Δtran	Δfocal (pixel)
Fern	0.64	0.61	0.47	0.50	22.22	21.67	1.78	0.029	153.5
Flower	0.71	0.71	0.36	0.37	25.25	25.34	4.84	0.016	13.2
Fortress	0.73	0.63	0.38	0.49	27.60	26.20	1.36	0.025	144.1
Horns	0.68	0.61	0.44	0.50	24.25	22.53	5.55	0.044	156.2
Leaves	0.52	0.53	0.47	0.47	18.81	18.88	3.90	0.016	59.0
Orchids	0.51	0.39	0.46	0.55	19.09	16.73	4.96	0.051	199.3
Room	0.87	0.84	0.40	0.44	27.77	25.84	2.77	0.030	331.8
Trex	0.74	0.72	0.41	0.44	23.19	22.67	4.67	0.036	89.3
Mean	0.68	0.63	0.42	0.47	23.52	22.48	3.73	0.031	143.3

Table 1: Quantitative comparison between our model and the *baseline NeRF* on LLFF-NeRF dataset. One the left, we show the NVS quality of our method with unknown cameras, achieving comparable results to *baseline NeRF*: ΔSSIM and ΔLPIPS = 0.05, ΔPSNR = 1.0. Note that the *baseline NeRF* numbers in this table is lower than the numbers reported in original paper, as we employed a smaller NeRF model for both *baseline NeRF* and *NeRF−−* for computation efficiency. On the right, we report the difference between our optimised camera parameters and ones computed from COLMAP. The result suggests our optimised camera poses are close to COLMAP estimations.

that jointly optimising the camera parameters and 3D scene representation is indeed possible. Nevertheless, we observe that for both the *Orchids* and the *Room*, our *NeRF−−* model produces slightly worse results compared to the *baseline NeRF*. One reason can be our joint optimisation for these two scenes might have trapped into local minima, ending up with inaccurate intrinsics, as we notice from Table 1 that the difference between optimised camera focal lengths and COLMAP estimation are most noticeable for these two scenes (199.3 and 331.8).

6.2.2 On Camera Parameter Estimation

Considering a forward-facing image capturing process in real life, where a user is likely to introduce unstable camera motions, to simulate this effect, we select $t_{010}r_{010}$ subset in BLEFF [2]. Table 2 lists the L1 difference on the estimated focal length, and metrics on camera rotation and translation computed with the ATE toolbox [43]. In summary, our rotation estimation error is about $5°$, and focal length error is about 25 pixels, where as COLMAP has a slightly larger rotation error but lower focal length error. As a result, the NVS quality of our method is comparable with *baseline NeRF*.

Method	Camera Parameters Error (vs. GT)			NVS Quality	
	Δfocal (pixel)	Δrot (deg)	Δtran	SSIM↑	PSNR↑
colmap	14.89	13.65	0.0127	0.90	33.92
ours	20.55	4.45	0.0654	0.90	33.24

Table 2: Quantitative evaluation of camera parameters estimation for both COLMAP and our method. We show that both the camera parameters estimation and NVS performance are comparable to COLMAP in BLEFF $t_{010}r_{010}$. A comprehensive list of results for each scene can be found in the supplementary.

Visualisation of Pose Optimisation For a better understanding of the optimisation process, we visualize the camera poses at various training epochs for the scene *Flower* from LLFF-NeRF dataset (Fig. 7). Starting from identity, our optimised camera poses gradually converge towards COLMAP estimations after about 1000 epochs, subject to a similarity transformation between optimised camera parameters and those estimated from COLMAP.

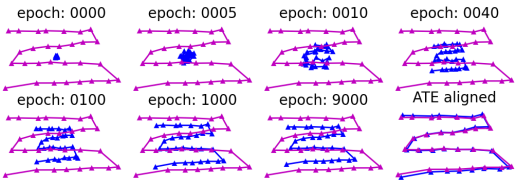


Figure 7: Pose optimisation during training, visualised on xy-plane (purple: COLMAP, blue: ours).

6.3 Discussion: Effects of Camera Motions

Considering *baseline NeRF* and *NeRF−−* as two alternatives, this section demystifies when and why one method are more favourable than the other under various camera motion patterns. We expand

[2] Suppose a trajectory of interest is bounded by a $1m \times 1m$ square, $\pm 10\%$ translation perturbation offers about $\pm 10\% \times 1.414m \Rightarrow 28.3cm$ flexibility in z direction during capturing.

this section from two aspects: First, we study the performance of our system and COLMAP under multiple levels of camera trajectory perturbations and explore the breaking point of both systems; Second, we examine system behaviours when a camera follows certain motion patterns during image capturing. As this section unfolds, we can see the necessity of discussing these common aspects in real-life capturing.

6.3.1 Breaking Point in Forward-Facing Scenes

The forward-facing assumption we made, which in an ideal case, assumes all cameras looking to $-z$ direction and moving within xy-plane. However, this assumption is often over optimistic in real-life image capturing, it is thus interesting to study the robustness of both COLMAP and *NeRF*$--$ under multiple levels of trajectory perturbations (Fig. 8).

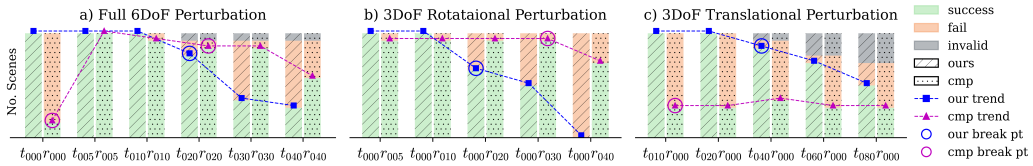


Figure 8: Breaking point analysis for camera parameter estimation in three groups of perturbation experiments: a) full 6DoF, b) 3DoF rotation, and c) 3DoF translation. COLMAP is shorten to "cmp" in the graph. We define a breaking point to the trajectory variant where a method starts failing in an experiment group. In group a) with full 6DoF noise, both methods start failing at $t_{020}r_{020}$. In 3DoF perturbations, our method performs more stable in translation perturbations but less stable in rotation perturbations. Note that COLMAP also faces degenerate issues with $t_{000}r_{000}$. Check Section 6.3.2 for more details on degenerate cases.

Setup Three groups of perturbation experiments are conducted in this section (Fig. 8): a) full 6DoF, b) 3DoF rotation, and c) 3DoF translation. We train all models for 3K epochs (not full 10K) and compare camera parameter estimations with ground truth with a criteria of mean rotation error $< 20^\circ$ and focal length error $< 50\%$ ground truth focal, as we observe severe NVS performance drop when the criteria is violated.

Result With full 6DoF perturbation added, both *NeRF*$--$ and COLMAP start breaking when translation noise $> \pm 20\%$ and rotation noise $> \pm 20^\circ$. Our method is favourable than COLMAP in large translation perturbations, but underperforms COLMAP in large rotation cases. Two reasons account for this result. First, it is well known that feature descriptors can handle appearance changes better than photometric methods, hence the COLMAP's superior performance in large rotation perturbation. However, feature-based method can also be vulnerable to degenerate cases, resulting in the failure cases in translation perturbation experiments, as will be discussed in Section 6.3.2.

6.3.2 Controlled Motion Patterns

In this section, we study three controlled camera motion patterns (Fig. 9): a) Rotational, where a camera motion contains mostly rotation with little translation. Note this is different from rotation perturbation experiment in Section 6.3.1, where the camera follows a planar trajectory but with perturbed viewing directions; b) Translational, in which a camera moves parallel to the camera sensor; c) Track-to-object, when a camera tracks an object. At the first glance, one may doubt if these controlled motions only exist in the synthetic world. In fact, they can often happen in real-life image capturing, especially when a camera is attached to a gimbal, dolly, or a drone.

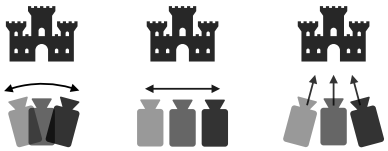
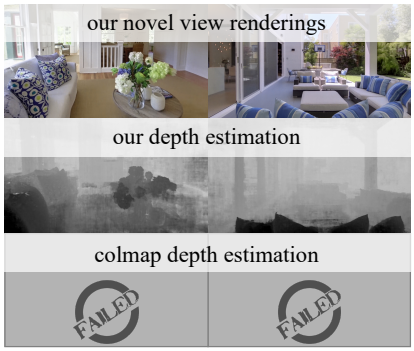


Figure 9: Three common camera motions in real-life capturing.

In short, our method outperforms COLMAP in rotational and translation camera motions, but underperforms COLMAP when a camera tracks an object.

Rotational and Translational As shown in Fig. 8c) and $t_{000}r_{000}$ column, COLMAP gives inaccurate camera parameter estimations in all translational experiments. Similar results are also observed in rotational motion. Fig. 10 presents two examples from the RealEstate10K dataset, where COLMAP fails to initialise or produces inaccurate estimations. For explanation, these results are known as degenerate cases while solving fundamental matrices from the point correspondences [15, 17]. Moreover, in [16], Torr et al. further point out that these degenerate cases are even more difficult to identify when matching outliers exist. Consequently, feature-matching-based SfM and SLAM methods tend to be vulnerable to these cases, whereas our photometry-based method handles them well.



a) rotational b) translational

Figure 10: Rotational and translational scenes that are problematic to COLMAP in real image dataset RealEstate10K.

Track-to-object We observe an inaccurate joint optimisation from our model, when a camera is set to track to an object during capturing, even when the camera rotation is well below our breaking point ($\pm 20°$). Fig. 11 illustrates such a situation. One hypothesis is, such track-to-object motion results in an effect that the object position and the image composition remain almost unchanged during capturing, which is not a problem for systems that work with high-res images directly, *e.g.,* COLMAP and SLAM systems. However, our method is supervised from the photometric loss between captured images and synthesised images. When the model is trained from scratch, synthesised images are blurry during most of the training time, producing little supervision for camera parameter optimisation.


a) track-to-object b) regular forward-facing

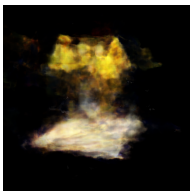
Figure 11: Rendered training views in two types of scenes at epoch 20. Unlike regular forward-facing scenes, where blurry rendered images are distinguishable even at early training stage, track-to-object scenes cannot provide enough photometric supervision from blurry rendered images, as the object position remains mostly unchanged (hinted by red dashed line) in all views.

6.4 Limitations and Future Work

Our system faces two limitations. *Firstly*, as motioned in section Section 6.3.2, our method does not estimate meaningful camera parameters when images are captured in a track-to-object motion, as there is not enough supervision to guide the optimisation of camera parameters in such motions. *Secondly*, we design our system to handle forward-facing scenes, which allows us initialising camera poses from identity matrices and optimising camera poses via gradient descent. However, this assumption restricts our pose estimation capability when the camera undergoes large rotations. As a result, camera motions with rotation perturbations larger than $\pm 20°$ might cause a failure (Fig. 8), and our method certainly cannot handle $360°$ scenes (Fig. 12). As for future work, exploring relative pose estimation between image pairs and temporal relationships in sequences might enable the processing of such $360°$ scenes.

7 Conclusions

Our contribution in this work is threefold. *First*, we present *NeRF*$--$, a framework that jointly optimises camera parameters and scene representation for forward-facing scenes, eliminating the need of pre-computing camera parameters, while achieving view synthesis results on par with the COLMAP-enabled NeRF baseline. *Second*, we introduce the BLEFF datasets for evaluating such jointly optimised novel view synthesis system with ground-truth camera parameters and high-quality path-traced images. *Third*, we present extensive experimental results and demonstrate the viability of this joint optimisation framework under different camera trajectory patterns, even when COLMAP fails to estimate the camera parameters. Despite its current limitations discussed above, our proposed joint optimisation pipeline has demonstrated promising results on this highly challenging task, and advanced one step towards end-to-end novel view synthesis on more general scenes.


Figure 12: Failed *Lego* $360°$ scene.

Acknowledgement

Shangzhe Wu is supported by Facebook Research. Weidi Xie is supported by Visual AI (EP/T028572/1). The authors would also like to thank Tim Yuqing Tang for insightful discussions and proofreading.

Appendix A Additional Results

In this section, we provide additional results from 4 directions: 1) NVS rendering quality; 2) Camera parameter estimation accuracy; 3) Full breaking point analysis results; and 4) Results of an optional refinement step.

A.1 On NVS Quality

Fig. 13 presents additional qualitative results on LLFF-NeRF dataset. Our method achieves comparable novel view synthesis quality to the *baseline NeRF* while taking RGB images as the only inputs.

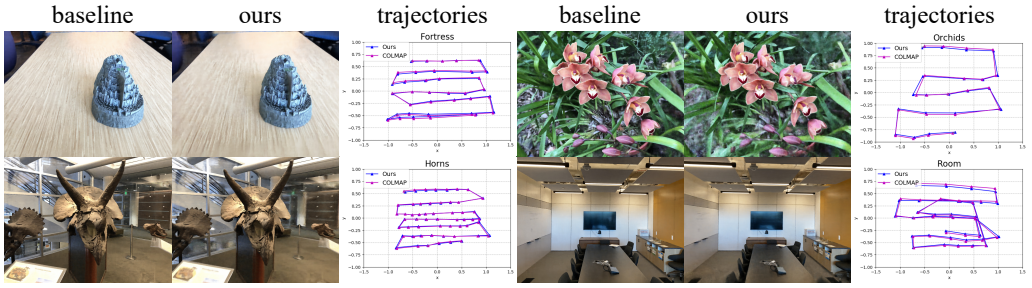


Figure 13: Additional Qualitatively Results on LLFF-NeRF Dataset. Our method achieves comparable NVS quality to the COLMAP-enabled NeRF while taking RGB images as the only input.

A.2 On Camera Parameter Estimation

We report the full pose estimation results for Section 6.2.2. In short, our method produces accurate pose estimations and comparable NVS rendering quality to COLMAP-enabled NeRF (ΔSSIM $= 0$).

Pose Ambiguity We notice a special case *Roundtable*, where there is a 179.10° rotation error between COLMAP predicted poses and ground truth poses but the COLMAP-NeRF model still produces high quality NVS renderings. This is a case of ambiguous pose estimation as stated in [45, 46], where two sets of pose estimations are mirror images with respect to a plane. Nevertheless, as the estimated mirrored poses and the estimated 3D geometries from COLMAP are self-consistent, a NeRF model trained in this case can still provide plausible novel view synthesis, hence the high PSNR and SSIM in evaluation.

A.3 Breaking Point Analysis

In Fig. 14 we present a per-scene breakdown for the breaking point analysis in Section 6.3.1. Our method performs better than COLMAP in translation perturbation but worse in rotation perturbation.

A.4 Optional Refinement

We propose an optional refinement step to further improve the quality of the synthesised images. Specifically, after the first training process is completed, we drop the trained NeRF model and re-initialise it with random parameters while keeping the pre-trained camera parameters. We then repeat the joint optimisation using the pre-trained camera parameters as initialisation. We find this additional refinement step generally leads to sharper images and improves the synthesis results, as evidenced by the comparison in Table 4.

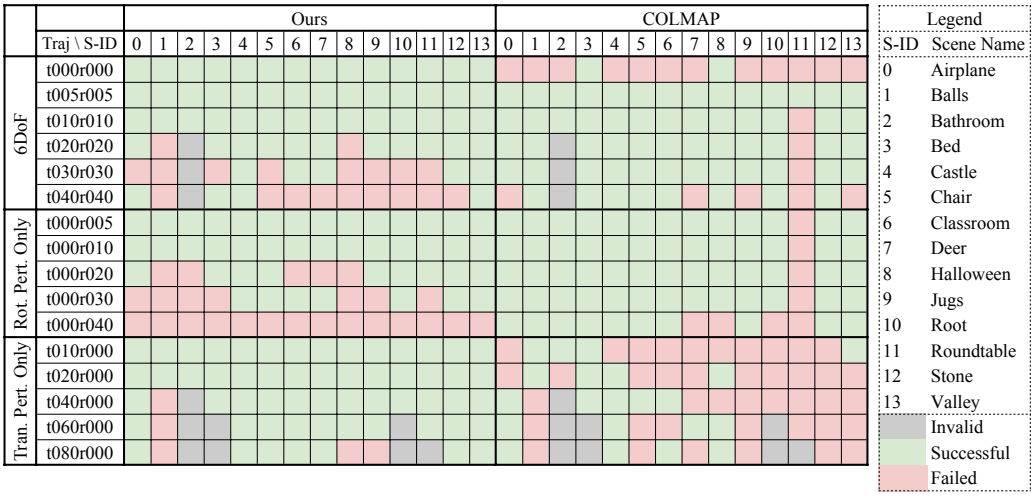


Figure 14: Per-scene camera parameter estimation results for breaking point analysis. Left: ours. Right: COLMAP. For each row (trajectory variant), the larger green area is, the better performance a method achieves in that trajectory. *Invalid* (grey) denotes that a part of a camera trajectory is out of a blender scene after perturbed, resulting in meaningless image renderings, from which it is impossible for any method to recover camera parameters or scene geometries. We set a criteria of $\pm 20^\circ$ rotation error and $\pm 0.5 f_{gt}$ focal length error to separate *successful* (green) and *failed* (red) camera parameter estimations.

Additionally, the camera parameters can also be initialised with estimated values from external toolboxes, where they are available, and jointly refined during the training of the NeRF model. We conduct experiments to refine the camera parameters estimated using COLMAP during NeRF training, and find the novel view results slightly improved through the joint refinement, as shown in Table 4.

Scene	Focal Err. (pixel)		Rot. Err. (deg)		Tran. Err.		SSIM↑		PSNR↑	
	colmap	ours	colmap	ours	colmap	ours	colmap	ours	colmap	ours
Airplane	0.09	0.87	0.02	0.61	0.000	0.003	0.83	0.87	28.19	30.57
Balls	0.21	15.44	0.06	13.43	0.001	0.285	0.89	0.81	34.56	32.12
Bathroom	0.02	0.52	1.49	1.50	0.001	0.004	0.93	0.94	32.64	31.58
Bed	0.26	0.39	2.21	2.21	0.003	0.004	0.94	0.94	33.28	32.41
Castle	0.04	3.23	0.04	3.17	0.001	0.020	0.88	0.89	32.44	32.74
Chair	0.01	6.12	0.03	3.52	0.000	0.078	0.84	0.81	33.42	32.24
Classroom	0.15	2.29	0.35	8.14	0.002	0.032	0.90	0.86	27.49	25.14
Deer	1.93	6.29	0.42	6.17	0.015	0.166	0.99	0.99	41.44	42.01
Halloween	0.69	9.77	0.13	6.74	0.003	0.064	0.94	0.91	31.59	29.30
Jugs	0.05	0.16	0.22	2.30	0.004	0.065	0.98	0.99	40.76	42.53
Root	15.27	36.42	6.95	4.51	0.078	0.054	0.97	0.97	36.49	35.45
Roundtable*	189.64	206.10	179.10	9.68	0.069	0.139	0.99	0.99	45.13	39.88
Stone	0.04	0.11	0.04	0.12	0.000	0.001	0.79	0.86	29.95	31.74
Valley	0.02	0.05	0.04	0.25	0.000	0.001	0.73	0.75	27.53	27.66
Mean	14.89	20.55	13.65	4.45	0.013	0.065	0.90	0.90	33.92	33.24

Table 3: Detailed quantitative camera parameter estimation results for Section 6.2.2. Our method produces accurate camera parameter estimations and comparable NVS quality to COLMAP-NeRF (ΔSSIM $= 0$). *Roundtable** is a special case where COLMAP estimated poses are mirrored with the ground truth. We tried 10 random seeds for COLMAP on this scene, 9/10 trials produce 179.1° rotation error and 1/10 trials produce 0.71° rotation error. Hence we report the 179.1° here. More details see the text above.

Appendix B Additional Details

B.1 Pseudo Code

We provide a pseudo code to show the key steps in our pipeline. In addition to set *require_grad* to *True*, the online ray construction step is another key that enables the direct joint optimisation via back-propagation.

ALGORITHM 1: *NeRF*$--$ Pipeline

Input: N images $\mathcal{I} = \{I\}_{i=1}^{N}$
Output: NeRF model F_Θ, camera parameters $[\hat{\boldsymbol{\phi}}_i]$, $[\hat{\mathbf{t}}_i]$, and $\hat{f}$

import torch.nn as nn

// Initialisation
$[\hat{\boldsymbol{\phi}}_i]$ = nn.Parameter(shape=(N, 3), require_grad=True)
$[\hat{\mathbf{t}}_i]$ = nn.Parameter(shape=(N, 3), require_grad=True)
$\hat{f}$ = nn.Parameter(shape=(1,), require_grad=True)
// NeRF structure see our supp.
F_Θ = NeRF_Module(require_grad=True)

// Training
for *i in* `range` *(N)* **do**
  for *m in* `range` *(M)* **do**
    $\hat{\mathbf{d}}_{i,m}$ = construct_ray($\hat{\boldsymbol{\phi}}_i$, $\hat{\mathbf{t}}_i$, $\hat{f}$, $\mathbf{p}_{i,m}$) // Eq. 4
    for h *from* h_n *to* h_f **do**
      $\mathbf{x}_j$ = sample_point($\hat{\mathbf{d}}_{i,m}$, $\hat{\mathbf{t}}_i$, h)
      $\mathbf{c}_h, \sigma_j = F_\Theta(\mathbf{x}_h, \hat{\mathbf{d}}_{i,m})$ // forward NeRF
    end
    $\hat{I}_{i,m}$ = render_ray($[\mathbf{c}_h]$, $[\sigma_h]$)
  end
  L = loss($\hat{I}_i$, I_i) // Eq. 3
  L.backward()
  optimiser.update($[\hat{\boldsymbol{\phi}}_i]$, $[\hat{\mathbf{t}}_i]$, $\hat{f}$, $\hat{\Theta}$)
end

B.2 NeRF Architecture

We employ a smaller NeRF [4] network than the original NeRF paper proposed without a hierarchical structure. Specifically, our NeRF implementation shrinks all hidden layer dimensions by half and follows the same positional encoding and skip connections as implemented in the original NeRF. The network architecture is presented in Fig. 15.

B.3 Training Details

Apart from the implementation details we mentioned in Section 6.1, we report more training details here. Specifically, a BLEFF scene with 27 training images can be trained in 5.5 hours on a single

Scene	SSIM↑				LPIPS↓				PSNR↑			
	colmap	ours	colmap + r	ours + r	colmap	ours	colmap + r	ours + r	colmap	ours	colmap + r	ours + r
Fern	0.64	0.61	0.64	0.63	0.47	0.50	0.47	0.48	22.22	21.67	22.28	21.93
Flower	0.71	0.71	0.72	0.72	0.36	0.37	0.36	0.35	25.25	25.34	25.53	25.58
Fortress	0.73	0.63	0.72	0.71	0.38	0.49	0.38	0.38	27.60	26.20	27.13	27.37
Horns	0.68	0.61	0.68	0.64	0.44	0.50	0.44	0.48	24.25	22.53	24.44	23.18
Leaves	0.52	0.53	0.53	0.53	0.47	0.47	0.47	0.46	18.81	18.88	18.94	18.96
Orchids	0.51	0.39	0.52	0.41	0.46	0.55	0.45	0.53	19.09	16.73	19.31	17.03
Room	0.87	0.84	0.87	0.85	0.40	0.44	0.39	0.42	27.77	25.84	27.96	26.28
Trex	0.74	0.72	0.74	0.73	0.41	0.44	0.41	0.43	23.19	22.67	23.13	22.95
Mean	0.68	0.63	0.68	0.65	0.42	0.47	0.42	0.44	23.52	22.48	23.59	22.91

Table 4: An optional camera parameter refinement (denote by '+r') can further improve NVS quality. During the refinement, camera parameters are first initialised from COLMAP estimations or from previous joint optimisations, and jointly optimised with a randomly initialised NeRF model.

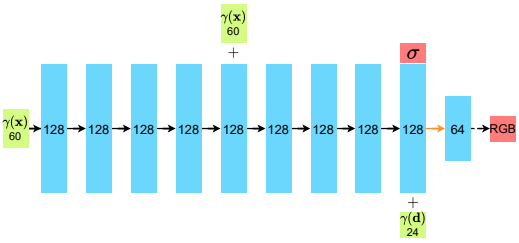


Figure 15: The NeRF implementation used in our paper. Green: position-encoded input. Blue: MLP hidden layers, with channel dimension shown inside. Red: radiance and density outputs. + denotes feature concatenation. Solid black arrow denotes layers with ReLU activation. Dashed black arrow denotes layers with Sigmoid activation. Orange arrow denotes layers without activation. We shrink all hidden layer dimensions in the original implementation by half, i.e. from 256 to 128 for the first 9 layers, and from 128 to 64 for the last layer. We employ the same number of positional encoding frequencies and the same skip links as in original NeRF. The figure style is borrowed from the original NeRF paper so readers can easily make comparisons.

1080Ti GPU. This training time is slightly longer than the training time (5 hours) of the original NeRF with our simplifications. The extra half hour training time mainly comes from the online ray direction construction from the learned camera poses and focal length.

B.4 Evaluation Details

Since our optimised camera parameters might lie in different spaces from the ones estimated using COLMAP, for evaluation, we first align the two trajectories globally with a Sim(3) transformation using Umeyama algorithm[47] in an ATE toolbox [43], followed by a more fine-grained gradient-driven camera pose alignment by minimising the photometric error on the synthesised image, while keeping the NeRF model *fixed*. Finally, we compute the metrics between the test image and our synthesised image rendered from the best possible viewpoint. Simply put, all the above mentioned processing aims to eliminate the effect from camera mis-alignment and make a fair comparison on quality of the 3D scene representation.

B.5 Focal Length Parameterisation

As mentioned in Section 6.1, we initialise our focal length f to be input image width W. In this section, we discuss the focal length parameterisation details, and compare three types of parameterisations.

In short, we choose to parameterise focal length as a single focal length f with a so-called 2nd-order trick. The details are elaborated in the following subsections.

B.5.1 f vs. $f_x f_y$

We choose to optimise a focal length f instead of a horizontal f_x and vertical f_y, as we found no performance difference between these choices (Table 5).

B.5.2 1st-order vs. 2nd-order

A naive way to optimise focal length f is to initialise it with image width W directly, but this poses difficulties to the direct optimisation as W is usually in a much larger numerical magnitude than other learnable parameters.

A better way to parameterise f is through a scale factor s:

$$f = sW, \tag{5}$$

and initialise with $s = 1.0$. By parameterising focal length with s, the network avoids optimising f in pixel unit directly, whose value are often large and pose numerical difficulties in optimisation.

In practice, we found that optimising the square root of s, denoted by $\tilde{s}$, leads to slightly better results, *i.e.,*

$$f = \tilde{s}^2 W, \tag{6}$$

Scene	PSNR↑		SSIM↑		LPIPS↓	
	f	$f_x f_y$	f	$f_x f_y$	f	$f_x f_y$
Fern	21.67	21.79	0.61	0.62	0.50	0.50
Flower	25.34	25.24	0.71	0.71	0.37	0.37
Fortress	26.20	26.24	0.63	0.65	0.49	0.46
Horns	22.53	23.08	0.61	0.63	0.50	0.50
Leaves	18.88	18.79	0.53	0.52	0.47	0.47
Orchids	16.73	16.48	0.39	0.37	0.55	0.56
Room	25.84	25.72	0.84	0.84	0.44	0.44
Trex	22.67	22.53	0.72	0.72	0.44	0.44
Mean	22.48	22.48	0.63	0.63	0.47	0.47

Table 5: NVS quality on LLFF-NeRF dataset when using f and $f_x f_y$. We choose to optimise a single focal length f for simplicity, as no performance gain acquired by optimising two focal lengths separately.

where $\tilde{s}$ is initialised to 1.0 too. We refer this parameterisation as the 2nd-order trick. Table 6 shows a quantitative comparison of the novel view rendering quality using these two parameterisations.

Scene	PSNR↑		SSIM↑		LPIPS↓	
	$\tilde{s}^2$	s	$\tilde{s}^2$	s	$\tilde{s}^2$	s
Fern	**21.67**	21.48	**0.61**	0.60	**0.50**	0.53
Flower	25.34	**25.44**	0.71	0.71	0.37	0.37
Fortress	**26.20**	24.92	0.63	**0.58**	**0.49**	0.58
Horns	22.53	22.53	0.61	0.61	0.50	0.50
Leaves	**18.88**	18.85	**0.53**	0.52	0.47	0.47
Orchids	**16.73**	16.67	0.39	0.39	0.55	0.55
Room	**25.84**	25.74	**0.84**	0.83	0.44	0.44
Trex	**22.67**	22.45	**0.72**	0.71	0.44	0.44
Mean	**22.48**	22.26	**0.63**	0.62	**0.47**	0.49

Table 6: Quantitative comparison (PSNR on novel views) of two different focal length parameterisations on LLFF-NeRF dataset. The 2nd-order parameterisation $\tilde{s}^2$ produces slightly better results than the 1st-order s.

B.6 RealEstate10K Details

We use two sequences from RealEstate10K [8] in Section 6.2.1 and Section 6.3.2. The details of the these videos and frame rate down-sampling procedures are listed in Table 7.

YouTube video ID	Original fps	Original res.	Training fps	Training res.
MVVJodQ50HQ	30	1920x1080	5	1920x1080
OT04jHhqYyw	30	1920x1080	5	1920x1080

Table 7: Details for selected sequences in RealEstate10K.

B.7 BLEFF Asset Licences

Our BLEFF dataset is made of 14 blender scenes, which we modified and downloaded from https://www.blendswap.com. The detailed licence info is listed in Table 8.

Scene	Licence	Author	Link
Airplane	CC-BY-3.0	fabien	https://blendswap.com/blend/15016
Balls-basketball	CC-BY-3.0	CGMasters	https://blendswap.com/blend/14758
Balls-volleyball	CC-BY-3.0	Shri	https://blendswap.com/blend/16958
Bathroom	CC-BY-SA-3.0	ORBANGeoffrey	https://blendswap.com/blend/20604
Bed	CC-BY-SA-3.0	Warcos	https://blendswap.com/blend/20975
Castle	CC-BY-0	BySky	https://blendswap.com/blend/27390

Chair	CC-BY-0	akashdlfps	https://blendswap.com/blend/25057
Classroom	CC-BY-3.0	SwastikDas	https://blendswap.com/blend/21410
Deer	CC-BY-0	Spine69	https://blendswap.com/blend/26863
Halloween	CC-BY-0	Yash Deokar	https://blendswap.com/blend/26793
Jugs	CC-BY-0	BigBadCat	https://blendswap.com/blend/23234
Root	CC-BY-0	AceTop	https://blendswap.com/blend/24472
Roundtable	CC-BY-0	gandre82	https://blendswap.com/blend/26159
Stone	CC-BY-0	rajatg8008	https://blendswap.com/blend/22490
Valley	CC-BY-0	ShadowCrystol	https://blendswap.com/blend/27325

Table 8: BLEFF licence info.

Appendix C Broader Impact

Joint Optimisation Method Our method focuses on simplifying the classical two-stage NVS systems, by jointly optimising a NeRF model and camera parameters for forward-facing images. We expect our method to be applied in a wide ranged of applications, for example, AR and VR content creation, 3D modelling, camera pose estimation and etc. As this method is designed to be trained on a per-scene basis and is designed for static scenes and cannot handle dynamic objects such as animals or humans, we expect the potential negative social impact (*i.e.,* bias in models and human privacy) that can be introduced by our method is very limited, as long as the image capturing procedure confronts local legal requirements.

BLEFF Dataset We also introduce a synthetic dataset BLEFF, which contains high-quality path-traced images rendered from 3D Blender models. The BLEFF scenes are selected to cover various common real-life subjects, such as furniture, decorations, toys, rooms, or outdoor landscapes and etc. There are two main purposes for this dataset: 1) to evaluate pose estimation and NVS rendering quality as the same time; and 2) to evaluate the robustness of a joint optimisation system similar to ours. Therefore, we expect our dataset to be beneficial to other joint optimisation systems (NVS and camera parameters) in the future.

References

[1] Edward H. Adelson and James R. Bergen. The plenoptic function and the elements of early vision. In *Computational Models of Visual Processing*, 1991. 1

[2] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *SIGGRAPH*, 1996. 1, 2, 3

[3] Marc Levoy and Pat Hanrahan. Light field rendering. In *SIGGRAPH*, 1996. 1, 2

[4] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 2, 3, 4, 12

[5] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. *arXiv preprint arXiv:2103.13415*, 2021. 1, 2

[6] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. *arXiv preprint arXiv:2008.02268*, 2020. 1, 2

[7] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, 2021. 1, 2

[8] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *SIGGRAPH*, 2018. 1, 2, 5, 14

[9] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *CVPR*, 2020. 1, 2

[10] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ToG*, 2019. 1, 2, 4, 5

[11] Inchang Choi, Orazio Gallo, Alejandro Troccoli, Min H Kim, and Jan Kautz. Extreme view synthesis. In *CVPR*, 2019. 1, 2

[12] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In *CVPR*, 2019. 1, 2, 5

[13] Suttisak Wizadwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. Nex: Real-time view synthesis with neural basis expansion. In *CVPR*, 2021. 1, 2, 5

[14] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1, 2, 3, 5

[15] Q-T Luong and Olivier D Faugeras. A stability analysis of the fundamental matrix. In *ECCV*, 1994. 2, 9

[16] Philip HS Torr, Andrew Zisserman, and Stephen J Maybank. Robust detection of degenerate configurations while estimating the fundamental matrix. *Computer Vision and Image Understanding*, 1998. 2, 9

[17] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. 2003. 2, 9

[18] Zhenglong Zhou, Zhe Wu, and Ping Tan. Multi-view photometric stereo with spatially varying isotropic materials. In *CVPR*, 2013. 2

[19] Gernot Riegler and Vladlen Koltun. Free view synthesis. In *ECCV*, 2020. 2

[20] Gernot Riegler and Vladlen Koltun. Stable view synthesis. *arXiv preprint arXiv:2011.07233*, 2020. 2

[21] Eric Penner and Li Zhang. Soft 3d reconstruction for view synthesis. In *SIGGRAPH Asia*, 2017. 2

[22] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *NeurIPS*, 2019. 2

[23] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 2019. 2

[24] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *NeurIPS*, 2020. 2

[25] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *TPAMI*, 2007. 2, 4

[26] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *TRO*, 2015. 2, 5

[27] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *ICCV*, 2011. 2, 4

[28] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *ICRA*, 2014. 2

[29] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *ECCV*, 2014. 2, 4

[30] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *SIGGRAPH*, 2006. 2

[31] Lin Yen-Chen, Pete Florence, Jonathan T. Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. *arXiv preprint arXiv:2012.05877*, 2020. 3

[32] Quan Meng, Anpei Chen, Haimin Luo, Minye Wu, Hao Su, Lan Xu, Xuming He, and Jingyi Yu. Gnerf: Gan-based neural radiance field without posed camera. *arXiv preprint arXiv:2103.15606*, 2021. 3

[33] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. *arXiv preprint arXiv:2104.06405*, 2021. 3

[34] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1995. 3

[35] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment — a modern synthesis. In *Vision Algorithms: Theory and Practice*, 2000. 4

[36] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ToG*, 2017. 5

[37] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *IROS*, 2012. 5, 6

[38] Thomas Schops, Torsten Sattler, and Marc Pollefeys. Bad slam: Bundle adjusted direct rgb-d slam. In *CVPR*, 2019. 5

[39] John McCormac, Ankur Handa, Stefan Leutenegger, and Andrew J Davison. Scenenet rgb-d: 5m photorealistic images of synthetic indoor trajectories with ground truth. In *ICCV*, 2016. 5

[40] Wenbin Li, Sajad Saeedi, John McCormac, Ronald Clark, Dimos Tzoumanikas, Qing Ye, Yuzhong Huang, Rui Tang, and Stefan Leutenegger. Interiornet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. In *BMVC*, 2018. 5

[41] 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*, 2004. 6

[42] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6

[43] Zichao Zhang and Davide Scaramuzza. A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry. In *IROS*, 2018. 6, 7, 13

[44] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015. 6

[45] Denis Oberkampf, Daniel F DeMenthon, and Larry S Davis. Iterative pose estimation using coplanar feature points. *Computer Vision and Image Understanding*, 1996. 10

[46] Gerald Schweighofer and Axel Pinz. Robust pose estimation from a planar target. *TPAMI*, 2006. 10

[47] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *TPAMI*, 1991. 13

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	NeRF--: Neural Radiance Fields Without Known Camera Parameters
Publication Status	□Published □ Accepted for Publication ☑Submitted for Publication □ Unpublished and unsubmitted work written in a manuscript style
Publication Details	**Zirui Wang**, Shangzhe Wu, Weidi Xie, Min Chen, Victor Adrian Prisacariu. In arXiv preprint arXiv:2102.07064, 2021.

Student Confirmation

Student Name:	Zirui Wang		
Contribution to the Paper	First-author contribution: • Conception of research ideas. • Design and implementation of models. • Running of large-scale experiments. • Writing and presentation of the paper.		
Signature	王梓睿	Date	20/Sep/2023

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	Professor Victor Adrian Prisacariu		
Supervisor comments:	The description is accurate.		
Signature		Date	20/Sep/2023

This completed form should be included in the thesis, at the end of the relevant chapter.

4.2 NoPe-NeRF: Optimising Neural Radiance Field with No Pose Prior

While Section 4.1 demonstrates the feasibility of jointly estimating camera parameters and NeRF, its applicability is constrained to scenes with limited camera motion for two primary reasons: **i)** NeRF training suffers from inherent shape-radiance ambiguity [170], which is further amplified while optimising additional camera parameters, and **ii)** existing methods estimate camera poses for individual images without accounting for their relative transformations.

Although classical epipolar geometry can potentially address the second limitation, it requires a series of pre-processing steps including interest point detection, feature matching, and transformation estimation. This contradicts our goal to eliminate complex and time-consuming pre-processing.

To overcome both challenges and avoid complex pre-processing, we introduce monocular depth estimation into the joint camera and NeRF optimisation pipeline. This integration serves two purposes: **i)** it offers robust geometric cues for NeRF training and camera parameter optimisation, and **ii)** it allows NeRF, which acts as a global volumetric model, to correct distortions in monocular depth estimation. As a result, monocular depth and NeRF complement each other.

Specifically, we integrate monocular depth from three perspectives: **i)** explicitly optimising distortion parameters for each depth map, un-distorting during NeRF training; **ii)** employing on-fly un-distorted depth for surface rendering and 3D registration between adjacent images, enhancing camera parameter estimation by considering relative transformations; and **iii)** supervising NeRF rendered depth with monocular depth to resolve shape-radiance ambiguity. As a result, our method effectively handles large camera motions and outperforms existing methods in both novel view rendering quality and camera pose estimation accuracy.

arXiv:2212.07388v3 [cs.CV] 14 Apr 2023

NoPe-NeRF: Optimising Neural Radiance Field with No Pose Prior

Wenjing Bian Zirui Wang Kejie Li Jia-Wang Bian
Victor Adrian Prisacariu
Active Vision Lab, University of Oxford
{wenjing, ryan, kejie, jiawang, victor}@robots.ox.ac.uk



Figure 1. **Novel view synthesis comparison**. We propose NoPe-NeRF for joint pose estimation and novel view synthesis. Our method enables more robust pose estimation and renders better novel view synthesis than previous state-of-the-art methods.

Abstract

Training a Neural Radiance Field (NeRF) without pre-computed camera poses is challenging. Recent advances in this direction demonstrate the possibility of jointly optimising a NeRF and camera poses in forward-facing scenes. However, these methods still face difficulties during dramatic camera movement. We tackle this challenging problem by incorporating undistorted monocular depth priors. These priors are generated by correcting scale and shift parameters during training, with which we are then able to constrain the relative poses between consecutive frames. This constraint is achieved using our proposed novel loss functions. Experiments on real-world indoor and outdoor scenes show that our method can handle challenging camera trajectories and outperforms existing methods in terms of novel view rendering quality and pose estimation accuracy. Our project page is https://nope-nerf.active.vision.

1. Introduction

The photo-realistic reconstruction of a scene from a stream of RGB images requires both accurate 3D geometry reconstruction and view-dependent appearance modelling. Recently, Neural Radiance Fields (NeRF) [24] have demonstrated the ability to build high-quality results for generating photo-realistic images from novel viewpoints given a sparse set of images.

An important preparation step for NeRF training is the estimation of camera parameters for the input images. A current go-to option is the popular Structure-from-Motion (SfM) library COLMAP [35]. Whilst easy to use, this pre-processing step could be an obstacle to NeRF research and real-world deployments in the long term due to its long processing time and its lack of differentiability. Recent works such as NeRFmm [47], BARF [18] and SC-NeRF [12] propose to simultaneously optimise camera poses and the neural implicit representation to address these issues. Nevertheless, these methods can only handle forward-facing scenes when no initial parameters are supplied, and fail in dramatic camera motions, *e.g.*a casual handheld captured video.

This limitation has two key causes. First, all these methods estimate a camera pose for each input image individually without considering relative poses between images. Looking back to the literature of Simultaneous localisation and mapping (SLAM) and visual odometry, pose estimation can significantly benefit from estimating relative poses be-

tween adjacent input frames. Second, the radiance field is known to suffer from *shape-radiance* ambiguity [56]. Estimating camera parameters jointly with NeRF adds another degree of ambiguity, resulting in slow convergence and unstable optimisation.

To handle the limitation of large camera motion, we seek help from monocular depth estimation [22, 28, 29, 52]. Our motivation is threefold: First, monocular depth provides strong geometry cues that are beneficial to constraint *shape-radiance* ambiguity. Second, relative poses between adjacent depth maps can be easily injected into the training pipeline via Chamfer Distance. Third, monocular depth is lightweight to run and does not require camera parameters as input, in contrast to multi-view stereo depth estimation. For simplicity, we use the term *mono-depth* from now on.

Utilising mono-depth effectively is not straightforward with the presence of scale and shift distortions. In other words, mono-depth maps are not multi-view consistent. Previous works [9, 17, 48] simply take mono-depth into a depth-wise loss along with NeRF training. Instead, we propose a novel and effective way to thoroughly integrate mono-depth into our system. First, we explicitly optimise scale and shift parameters for each mono-depth map during NeRF training by penalising the difference between rendered depth and mono-depth. Since NeRF by itself is trained based on multiview consistency, this step transforms mono-depth maps to undistorted multiview consistent depth maps. We further leverage these multiview consistent depth maps in two loss terms: a) a Chamfer Distance loss between two depth maps of adjacent images, which injects relative pose to our system; and b) a depth-based surface rendering loss, which further improves relative pose estimation.

In summary, we propose a method to jointly optimise camera poses and a NeRF from a sequence of images with large camera motion. Our system is enabled by three contributions. **First**, we propose a novel way to integrate mono-depth into unposed-NeRF training by explicitly modelling scale and shift distortions. **Second**, we supply relative poses to the camera-NeRF joint optimisation via an inter-frame loss using undistorted mono-depth maps. **Third**, we further regularise our relative pose estimation with a depth-based surface rendering loss.

As a result, our method is able to handle large camera motion, and outperforms state-of-the-art methods by a significant margin in terms of novel view synthesis quality and camera trajectory accuracy.

2. Related Work

Novel View Synthesis. While early Novel View Synthesis (NVS) approaches applied interpolations between pixels [3], later works often rendered images from 3D reconstruction [1, 6]. In recent years, different representations of the 3D scene are used, *e.g.*meshes [30, 31], Multi-Plane Images [42, 60], layered depth [43] *etc*. Among them, NeRF [24] has become a popular scene representation for its photorealistic rendering.

A number of techniques are proposed to improve NeRF's performance with additional regularisation [13, 26, 55], depth priors [7, 32, 48, 54], surface enhancements [27, 44, 50] or latent codes [41, 45, 53]. Other works [2, 10, 25, 34] have also accelerated NeRF training and rendering. However, most of these approaches require pre-computed camera parameters obtained from SfM algorithms [11, 35].

NeRF With Pose Optimisation. Removing camera parameter preprocessing is an active line of research recently. One category of the methods [33, 39, 61] use a SLAM-style pipeline, that either requires RGB-D inputs or relies on accurate camera poses generated from the SLAM tracking system. Another category of works optimises camera poses with the NeRF model directly. We term this type of method as *unposed-NeRF* in this paper. iNeRF [51] shows that poses for novel view images can be estimated using a reconstructed NeRF model. GNeRF [21] combines Generative Adversarial Networks with NeRF to estimate camera poses but requires a known sampling distribution for poses. More relevant to our work, NeRFmm [47] jointly optimises both camera intrinsics and extrinsics alongside NeRF training. BARF [18] proposes a coarse-to-fine positional encoding strategy for camera poses and NeRF joint optimisation. SC-NeRF [12] further parameterises camera distortion and employs a geometric loss to regularise rays. GARF [4] shows that using Gaussian-MLPs makes joint pose and scene optimisation easier and more accurate. Recently, SiNeRF [49] uses SIREN [37] layers and a novel sampling strategy to alleviate the sub-optimality of joint optimisation in NeRFmm. Although showing promising results on the forward-facing dataset like LLFF [23], these approaches face difficulties when handling challenging camera trajectories with large camera motion. We address this issue by closely integrating mono-depth maps with the joint optimisation of camera parameters and NeRF.

3. Method

We tackle the challenge of handling large camera motions in unposed-NeRF training. Given a sequence of images, camera intrinsics, and their mono-depth estimations, our method recovers camera poses and optimises a NeRF simultaneously. We assume camera intrinsics are available in the image meta block, and we run an off-the-shelf mono-depth network DPT [7] to acquire mono-depth estimations. Without repeating the benefit of mono-depth, we unroll this section around the effective integration of monocular depth into unposed-NeRF training.

The training is a joint optimisation of the NeRF, camera poses, and distortion parameters of each mono-depth

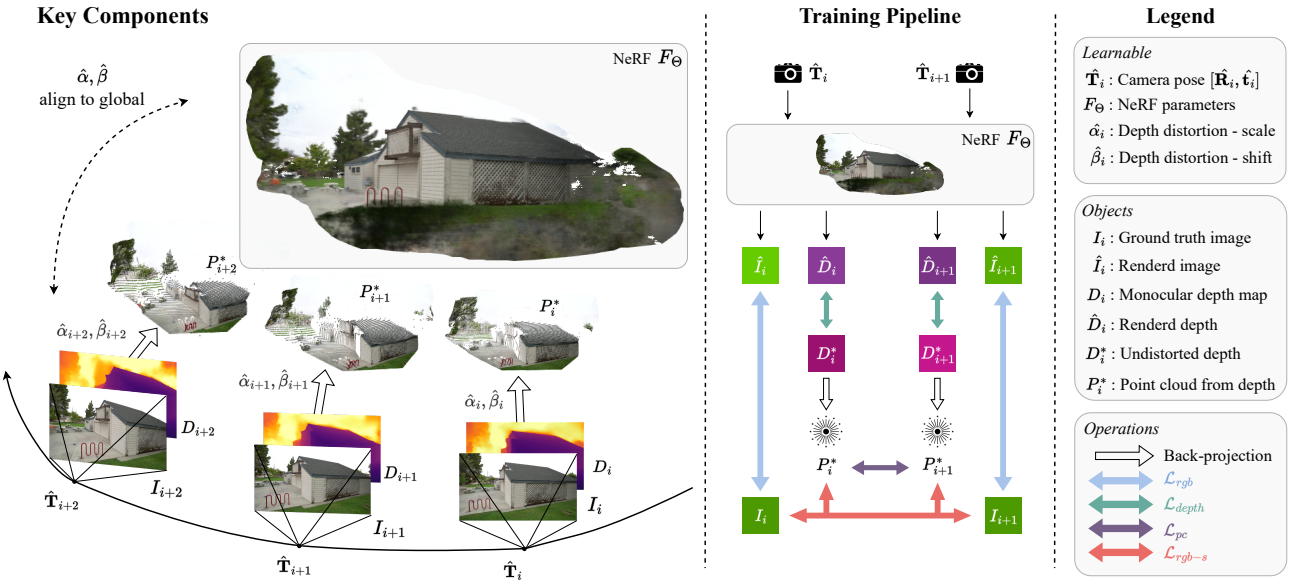


Figure 2. **Method Overview.** Our method takes a sequence of images as input to reconstruct NeRF and jointly estimates the camera poses of the frames. We first generate monocular depth maps from a mono-depth estimation network and reconstruct the point clouds. We then optimise NeRF, camera poses, and depth distortion parameters jointly with inter-frame and NeRF losses.

map. The distortion parameters are supervised by minimising the discrepancies between the mono-depth maps and depth maps rendered from the NeRF, which are multiview consistent. The undistorted depth maps in return effectively mediate the *shape-radiance* ambiguity, which eases the training of NeRF and camera poses.

Specifically, the undistorted depth maps enable two constraints. We constrain global pose estimation by supplying relative pose between adjacent images. This is achieved via a Chamfer-Distance-based correspondence between two point clouds, back-projected from undistorted depth maps. Further, we regularise relative pose estimation with a surface-based photometric consistency where we treat undistorted depth as surface.

We detail our method in the following sections, starting from NeRF in Sec. 3.1 and unposed-NeRF training in Sec. 3.2, looking into mono-depth distortions in Sec. 3.3, followed by our mono-depth enabled loss terms in Sec. 3.4, and finishing with an overall training pipeline Sec. 3.5.

3.1. NeRF

Neural Radiance Field (NeRF) [24] represents a scene as a mapping function $F_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$ that maps a 3D location $\mathbf{x} \in \mathbb{R}^3$ and a viewing direction $\mathbf{d} \in \mathbb{R}^3$ to a radiance colour $\mathbf{c} \in \mathbb{R}^3$ and a volume density value σ. This mapping is usually implemented with a neural network parameterised by F_Θ. Given N images $\mathcal{I} = \{I_i \mid i = 0 \dots N-1\}$ with their camera poses $\Pi = \{\pi_i \mid i = 0 \dots N-1\}$, NeRF can be optimised by minimising photometric error $\mathcal{L}_{rgb} = \sum_i^N \|I_i - \hat{I}_i\|_2^2$ between synthesised images $\hat{\mathcal{I}}$ and captured images $\mathcal{I}$:

$$\Theta^* = \arg\min_\Theta \mathcal{L}_{rgb}(\hat{\mathcal{I}} \mid \mathcal{I}, \Pi), \tag{1}$$

where $\hat{I}_i$ is rendered by aggregating radiance colour on camera rays $\mathbf{r}(h) = \mathbf{o} + h\mathbf{d}$ between near and far bound h_n and h_f. More concretely, we synthesise $\hat{I}_i$ with a volumetric rendering function

$$\hat{I}_i(\mathbf{r}) = \int_{h_n}^{h_f} T(h)\sigma(\mathbf{r}(h))\mathbf{c}(\mathbf{r}(h), \mathbf{d})dh, \tag{2}$$

where $T(h) = \exp(-\int_{h_n}^{h} \sigma(\mathbf{r}(s))ds)$ is the accumulated transmittance along a ray. We refer to [24] for further details.

3.2. Joint Optimisation of Poses and NeRF

Prior works [12, 18, 47] show that it is possible to estimate both camera parameters and a NeRF at the same time by minimising the above photometric error $\mathcal{L}_{rgb}$ under the same volumetric rendering process in Eq. (2).

The key lies in conditioning camera ray casting on variable camera parameters Π, as the camera ray $\mathbf{r}$ is a function of camera pose. Mathematically, this joint optimisation can be formulated as:

$$\Theta^*, \Pi^* = \arg\min_{\Theta,\Pi} \mathcal{L}_{rgb}(\hat{\mathcal{I}}, \hat{\Pi} \mid \mathcal{I}), \tag{3}$$

where $\hat{\Pi}$ denotes camera parameters that are updated during optimising. Note that the only difference between Eq. (1)

and Eq. (3) is that Eq. (3) considers camera parameters as variables.

In general, the camera parameters Π include camera intrinsics, poses, and lens distortions. We only consider estimating camera poses in this work, *e.g.*, camera pose for frame I_i is a transformation $\mathbf{T}_i = [\mathbf{R}_i \mid \mathbf{t}_i]$ with a rotation $\mathbf{R}_i \in \mathrm{SO}(3)$ and a translation $\mathbf{t}_i \in \mathbb{R}^3$.

3.3. Undistortion of Monocular Depth

With an off-the-shelf monocular depth network, *e.g.*, DPT [28], we generate mono-depth sequence $\mathcal{D} = \{D_i \mid i = 0 \dots N-1\}$ from input images. Without surprise, mono-depth maps are not multi-view consistent so we aim to recover a sequence of multi-view consistent depth maps, which are further leveraged in our relative pose loss terms.

Specifically, we consider two linear transformation parameters for each mono-depth map, resulting in a sequence of transformation parameters for all frames $\Psi = \{(\alpha_i, \beta_i) \mid i = 0 \dots N-1\}$, where α_i and β_i denote a scale and a shift factor. With multi-view consistent constraint from NeRF, we aim to recover a multi-view consistent depth map D_i^* for D_i:

$$D_i^* = \alpha_i D_i + \beta_i, \tag{4}$$

by joint optimising α_i and β_i along with a NeRF. This joint optimisation is mostly achieved by enforcing the consistency between an undistorted depth map D_i^* and a NeRF rendered depth map $\hat{D}_i$ via a depth loss:

$$\mathcal{L}_{depth} = \sum_{i}^{N} \left\| D_i^* - \hat{D}_i \right\|, \tag{5}$$

where

$$\hat{D}_i(\mathbf{r}) = \int_{h_n}^{h_f} T(h)\sigma(\mathbf{r}(h))dh \tag{6}$$

denotes a volumetric rendered depth map from NeRF.

It is important to note that both NeRF and mono-depth benefit from Eq. (5). On the one hand, mono-depth provides strong geometry prior for NeRF training, reducing *shape-radiance* ambiguity. On the other hand, NeRF provides multi-view consistency so we can recover a set of multi-view consistent depth maps for relative pose estimations.

3.4. Relative Pose Constraint

Aforementioned unposed-NeRF methods [12, 18, 47] optimise each camera pose independently, resulting in an overfit to target images with incorrect poses. Penalising incorrect relative poses between frames can help to regularise the joint optimisation towards smooth convergence, especially in a complex camera trajectory. Therefore, we propose two losses that constrain relative poses.

Point Cloud Loss. We back-project the undistorted depth maps $\mathcal{D}^*$ using the known camera intrinsics, to point clouds $\mathcal{P}^* = \{P_i^* \mid i = 0 \dots N-1\}$ and optimise the relative pose between consecutive point clouds by minimising a point cloud loss $\mathcal{L}_{pc}$:

$$\mathcal{L}_{pc} = \sum_{(i,j)} l_{cd}(P_j^*, \mathbf{T}_{ji} P_i^*), \tag{7}$$

where $\mathbf{T}_{ji} = \mathbf{T}_j \mathbf{T}_i^{-1}$ represents the related pose that transforms point cloud P_i^* to P_j^*, tuple (i, j) denotes indices of a consecutive pair of instances, and l_{cd} denotes Chamfer Distance:

$$l_{cd}(P_i, P_j) = \sum_{p_i \in P_i} \min_{p_j \in P_j} \|p_i - p_j\|_2 + \sum_{p_j \in P_j} \min_{p_i \in P_i} \|p_i - p_j\|_2. \tag{8}$$

Surface-based Photometric Loss. While the point cloud loss $\mathcal{L}_{pc}$ offers supervision in terms of 3D-3D matching, we observe that a surface-based photometric error can alleviate incorrect matching. With the photometric consistency assumption, this photometric error penalises the differences in appearance between associated pixels. The association is established by projecting the point cloud P_i^* onto images I_i and I_j.

The surface-based photometric loss can then be defined as:

$$\mathcal{L}_{rgb-s} = \sum_{(i,j)} \| I_i \langle \mathbf{K}_i P_i^* \rangle - I_j \langle \mathbf{K}_j \mathbf{T}_j \mathbf{T}_i^{-1} P_i^* \rangle \|, \tag{9}$$

where $\langle \cdot \rangle$ represents the sampling operation on the image and $\mathbf{K}_i$ denotes a projection matrix for i_{th} camera.

3.5. Overall Training Pipeline

Assembling all loss terms, we get the overall loss function:

$$\mathcal{L} = \mathcal{L}_{rgb} + \lambda_1 \mathcal{L}_{depth} + \lambda_2 \mathcal{L}_{pc} + \lambda_3 \mathcal{L}_{rgb-s}, \tag{10}$$

where $\lambda_1, \lambda_2, \lambda_3$ are the weighting factors for respective loss terms. By minimising the combined of loss $\mathcal{L}$:

$$\Theta^*, \Pi^*, \Psi^* = \arg\min_{\Theta, \Pi, \Psi} \mathcal{L}(\hat{\mathcal{I}}, \hat{\mathcal{D}}, \hat{\Pi}, \hat{\Psi} \mid \mathcal{I}, \mathcal{D}), \tag{11}$$

our method returns the optimised NeRF parameters Θ, camera poses Π, and distortion parameters Ψ.

4. Experiments

We begin with a description of our experimental setup in Sec. 4.1. In Sec. 4.2, we compare our method with pose-unknown methods. Next, we compare our method with the COLMAP-assisted NeRF baseline in Sec. 4.3. Lastly, we conduct ablation studies in Sec. 4.4.



(a) BARF (b) NeRFmm (c) SC-NeRF (d) Ours (e) Ground Truth

Figure 3. **Qualitative results of novel view synthesis and depth prediction on Tanks and Temples.** We visualise the synthesised images and the rendered depth maps (top left of each image) for all methods. NoPe-NeRF is able to recover details for both colour and geometry.

4.1. Experimental Setup

Datasets. We conduct experiments on two datasets *Tanks and Temples* [15] and *ScanNet* [5]. **Tanks and Temples**: we use 8 scenes to evaluate pose accuracy and novel view synthesis quality. We chose scenes captured at both indoor and outdoor locations, with different frame sampling rates and lengths. All images are down-sampled to a resolution of 960×540. For the *family* scene, we sample 200 images and take 100 frames with odd frame ids as training images and the remaining 100 frames for novel view synthesis, in order to analyse the performance under smooth motion. For the remaining scenes, following NeRF [24], 1/8 of the images in each sequence are held out for novel view synthesis, unless otherwise specified. **ScanNet**: we select 4 scenes for evaluating pose accuracy, depth accuracy, and novel view synthesis quality. For each scene, we take 80-100 consecutive images and use 1/8 of these images for novel view synthesis. For evaluation, we employ depth maps and poses provided by ScanNet as ground truth. ScanNet images are down-sampled to 648×484. We crop images with dark orders during preprocessing.

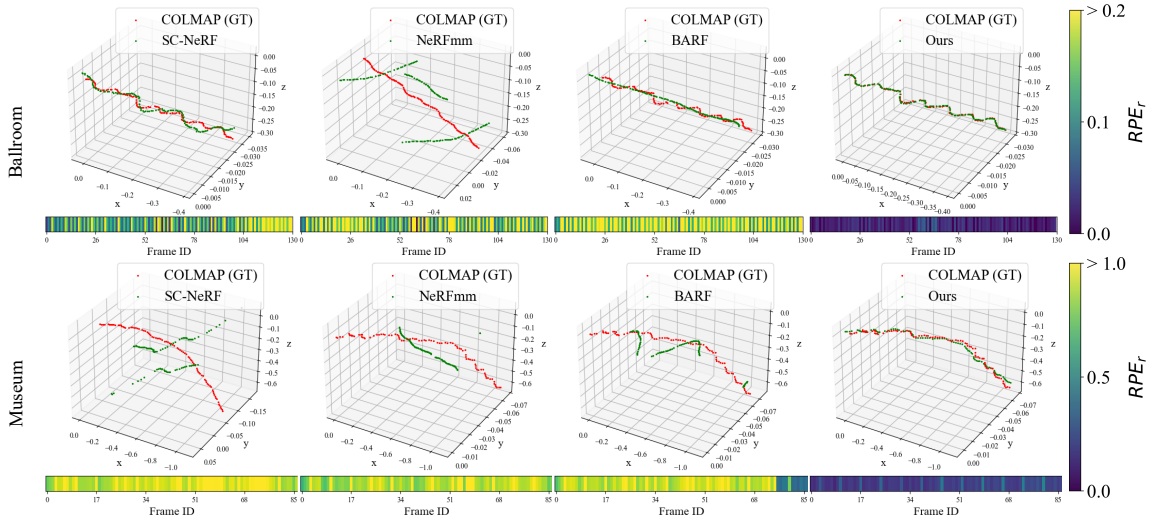
Metrics. We evaluate our proposed method in three aspects. For **novel view synthesis**, we follow previous methods [12,18,47], and use standard evaluation metrics, including Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) [46] and Learned Perceptual Image Patch Similarity (LPIPS) [57]. For **pose** evaluation, We use standard visual odometry metrics [16, 38, 58], including the Absolute Trajectory Error (ATE) and Relative Pose Error (RPE). ATE measures the difference between the estimated camera positions and the ground truth positions. RPE measures the relative pose errors between pairs of images, which consists of relative rotation error (RPE_r) and relative translation error (RPE_t). The estimated trajectory is aligned with the ground truth using Sim(3) with 7 degrees of freedom. We use standard depth metrics [8,19,20,40] (Abs Rel, Sq Rel, RMSE, RMSE log, δ_1, δ_2 and δ_3) for **depth** evaluation. For further detail, please refer to the supplementary material. To recover the metric scale, we follow Zhou *et al*. [59] and match the median value between rendered and ground truth depth maps.

Implementation Details. Our model architecture is based on NeRF [24] with a few modifications: a) replacing ReLU activation function with Softplus and b) sampling 128 points along each ray uniformly with noise, between a predefined range $(0.1, 10)$. We use 2 separate Adam optimisers [14] for NeRF and other parameters. The initial learning rate for NeRF is 0.001 and for the pose and distortion is 0.0005. Camera rotations are optimised in axis-angle representation $\phi_i \in \mathfrak{so}(3)$. We first train the model with all losses with constant learning rates until convergence. Then, we decay the learning rates with different schedulers and gradually reduce weights for the inter-frame losses and depth loss to further train for 10,000 epochs. We balance the loss terms with $\lambda_1 = 0.04$, $\lambda_2 = 1.0$ and $\lambda_3 = 1.0$. For each training step, we randomly sample 1024 pixels (rays) from each input image and 128 samples per ray. More details are provided in the supplementary material.

4.2. Comparing With Pose-Unknown Methods

We compare our method with pose-unknown baselines, including BARF [18], NeRFmm [47] and SC-NeRF [12].

View Synthesis Quality. To obtain the camera poses of test views for rendering, we minimise the photometric error



(a) BARF (b) NeRFmm (c) SC-NeRF (d) Ours

Figure 4. **Pose Estimation Comparison.** We visualise the trajectory (3D plot) and relative rotation errors RPE_r (bottom colour bar) of each method on *Ballroom* and *Museum*. The colour bar on the right shows the relative scaling of colour. More results are in the supplementary.

	scenes	Ours			BARF			NeRFmm			SC-NeRF		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
ScanNet	0079_00	**32.47**	**0.84**	**0.41**	32.31	0.83	0.43	30.59	0.81	0.49	31.33	0.82	0.46
	0418_00	**31.33**	**0.79**	**0.34**	31.24	**0.79**	0.35	30.00	0.77	0.40	29.05	0.75	0.43
	0301_00	**29.83**	**0.77**	0.36	29.31	0.76	0.38	27.84	0.72	0.45	29.45	**0.77**	**0.35**
	0431_00	**33.83**	**0.91**	**0.39**	32.77	0.90	0.41	31.44	0.88	0.45	32.57	0.90	0.40
	mean	**31.86**	**0.83**	**0.38**	31.41	0.82	0.39	29.97	0.80	0.45	30.60	0.81	0.41
Tanks and Temples	Church	**25.17**	**0.73**	**0.39**	23.17	0.62	0.52	21.64	0.58	0.54	21.96	0.60	0.53
	Barn	**26.35**	**0.69**	**0.44**	25.28	0.64	0.48	23.21	0.61	0.53	23.26	0.62	0.51
	Museum	**26.77**	**0.76**	**0.35**	23.58	0.61	0.55	22.37	0.61	0.53	24.94	0.69	0.45
	Family	**26.01**	**0.74**	**0.41**	23.04	0.61	0.56	23.04	0.58	0.56	22.60	0.63	0.51
	Horse	**27.64**	**0.84**	**0.26**	24.09	0.72	0.41	23.12	0.70	0.43	25.23	0.76	0.37
	Ballroom	**25.33**	**0.72**	**0.38**	20.66	0.50	0.60	20.03	0.48	0.57	22.64	0.61	0.48
	Francis	**29.48**	**0.80**	**0.38**	25.85	0.69	0.57	25.40	00.69	0.52	26.46	0.73	0.49
	Ignatius	**23.96**	**0.61**	**0.47**	21.78	0.47	0.60	21.16	0.45	0.60	23.00	0.55	0.53
	mean	**26.34**	**0.74**	**0.39**	23.42	0.61	0.54	22.50	0.59	0.54	23.76	0.65	0.48

Table 1. **Novel view synthesis results on ScanNet and Tanks and Temples**. Each baseline method is trained with its public code under the original settings and evaluated with the same evaluation protocol.

of the synthesised images while keeping the NeRF model fixed, as in NeRFmm [47]. Each test pose is initialised with the learned pose of the training frame that is closest to it. We use the same pre-processing for all baseline approaches, which results in higher accuracy than their original implementations. More details are provided in the supplementary material. Our method outperforms all the baselines by a large margin. The quantitative results are summarised in Tab. 1, and qualitative results are shown in Fig. 3.

We recognised that because the test views, which are sampled from videos, are close to the training views, good results may be obtained due to overfitting to the training images. Therefore, we conduct an additional qualitative evaluation on more novel views. Specifically, we fit a bezier curve from the estimated training poses and sample interpolated poses for each method to render novel view videos. Sampled results are shown in Fig. 5, and the rendered videos are in the supplementary material. These results show that our method renders photo-realistic images consistently, while other methods generate visible artifacts.

Camera Pose. Our method significantly outperforms other baselines in all metrics. The quantitative pose evaluation results are shown in Tab. 2. For ScanNet, we use the camera poses provided by the dataset as ground truth. For Tanks and Temples, not every video comes with ground truth poses, so we use COLMAP estimations for reference. Our estimated trajectory is better aligned with the ground truth than other methods, and our estimated rotation is two

	scenes	Ours			BARF			NeRFmm			SC-NeRF		
		$RPE_t \downarrow$	$RPE_r \downarrow$	ATE$\downarrow$	RPE_t	RPE_r	ATE	RPE_t	RPE_r	ATE	RPE_t	RPE_r	ATE
ScanNet	0079_00	**0.752**	**0.204**	**0.023**	1.110	0.480	0.062	1.706	0.636	0.100	2.064	0.664	0.115
	0418_00	**0.455**	**0.119**	**0.015**	1.398	0.538	0.020	1.402	0.460	0.013	1.528	0.502	0.016
	0301_00	**0.399**	**0.123**	**0.013**	1.316	0.777	0.219	3.097	0.894	0.288	1.133	0.422	0.056
	0431_00	**1.625**	**0.274**	**0.069**	6.024	0.754	0.168	6.799	0.624	0.496	4.110	0.499	0.205
	mean	**0.808**	**0.180**	**0.030**	2.462	0.637	0.117	3.251	0.654	0.224	2.209	0.522	0.098
Tanks and Temples	Church	**0.034**	**0.008**	**0.008**	0.114	0.038	0.052	0.626	0.127	0.065	0.836	0.187	0.108
	Barn	**0.046**	**0.032**	**0.004**	0.314	0.265	0.050	1.629	0.494	0.159	1.317	0.429	0.157
	Museum	**0.207**	**0.202**	**0.020**	3.442	1.128	0.263	4.134	1.051	0.346	8.339	1.491	0.316
	Family	**0.047**	**0.015**	**0.001**	1.371	0.591	0.115	2.743	0.537	0.120	1.171	0.499	0.142
	Horse	**0.179**	**0.017**	**0.003**	1.333	0.394	0.014	1.349	0.434	0.018	1.366	0.438	0.019
	Ballroom	**0.041**	**0.018**	**0.002**	0.531	0.228	0.018	0.449	0.177	0.031	0.328	0.146	0.012
	Francis	**0.057**	**0.009**	**0.005**	1.321	0.558	0.082	1.647	0.618	0.207	1.233	0.483	0.192
	Ignatius	**0.026**	**0.005**	**0.002**	0.736	0.324	0.029	1.302	0.379	0.041	0.533	0.240	0.085
	mean	**0.080**	**0.038**	**0.006**	1.046	0.441	0.078	1.735	0.477	0.123	1.890	0.489	0.129

Table 2. **Pose accuracy on ScanNet and Tanks and Temples**. Note that we use COLMAP poses in Tanks and Temples as the "ground truth". The unit of RPE_r is in degrees, ATE is in the ground truth scale and RPE_t is scaled by 100.

	Abs Rel $\downarrow$	Sq Rel $\downarrow$	RMSE $\downarrow$	RMSE log $\downarrow$	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$
Ours	**0.141**	**0.137**	**0.568**	**0.176**	**0.828**	**0.970**	**0.987**
BARF	0.376	0.684	0.990	0.401	0.490	0.751	0.884
NeRFmm	0.590	1.721	1.672	0.587	0.316	0.560	0.743
SC-NeRF	0.417	0.642	1.079	0.476	0.362	0.658	0.832
DPT	0.197	0.246	0.751	0.226	0.747	0.934	0.975

Table 3. **Depth map evaluation on ScanNet**. Our depth estimation is more accurate than baseline models BARF [18], NeRFmm [47] and SC-NeRF [12]. Compared with DPT [59], we show our depth is more accurate after undistortion.

orders of magnitudes more accurate than others. We visualise the camera trajectories and rotations in Fig. 4.

Depth. We evaluate the accuracy of the rendered depth maps on ScanNet, which provides the ground-truth depths for evaluation. Our rendered depth maps achieve superior accuracy over the previous alternatives. We also compare with the mono-depth maps estimated by DPT. Our rendered depth maps, after undistortion using multiview consistency in the NeRF optimisation, outperform DPT by a large margin. The results are summarised in Tab. 3, and sampled qualitative results are illustrated in Fig. 3.

4.3. Comparing With COLMAP Assisted NeRF

We make a comparison of pose estimation accuracy between our method and COLMAP against ground truth poses in ScanNet. We achieve on-par accuracy with COLMAP, as shown in Tab. 4. We further analyse the novel view synthesis quality of the NeRF model trained with our learned poses to COLMAP poses on ScanNet and Tanks and Temples. The original NeRF training contains two stages, finding poses using COLMAP and optimising the scene representation. In order to make our comparison fairer, in this section only, we mimic a similar two-stage training as the original NeRF [24]. In the first stage, we train our method with all losses for camera pose estimation, *i.e.*, mimicking the COLMAP processing. Then, we fix the optimised poses and train a NeRF model from scratch, using the same settings and loss as the original NeRF. This evaluation enables us to compare our estimated poses to the COLMAP poses indirectly, *i.e.*, in terms of contribution to view synthesis.

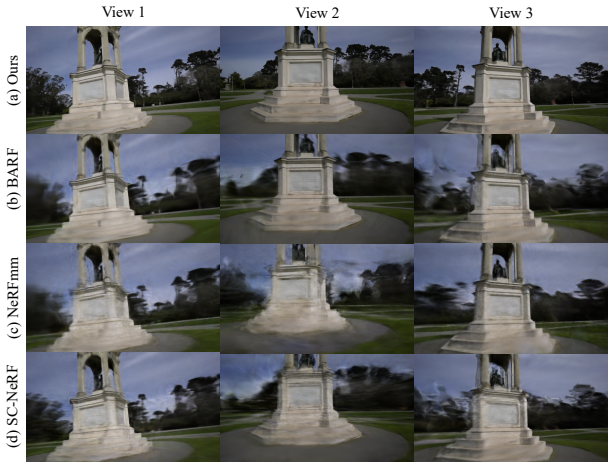


Figure 5. **Sampled frames from rendered novel view videos.** For each method, we fit the learned trajectory with a bezier curve and uniformly sample new viewpoints for rendering. Our method generates significantly better results than previous methods, which show visible artifacts. The full rendered videos and details about generating novel views are provided in the supplementary.

Our two-stage method outperforms the COLMAP-assisted NeRF baseline, which indicates a better pose estimation for novel view synthesis. The results are summarised in Tab. 5.

As is commonly known, COLMAP performs poorly in low-texture scenes and sometimes fails to find accurate camera poses. Fig. 6 shows an example of a low-texture scene where COLMAP provides inaccurate pose estimation that causes NeRF to render images with visible artifacts. In contrast, our method renders high-quality images, thanks to robust optimisation of camera pose.

Interestingly, this experiment also reveals that the two-

scenes	Ours			COLMAP		
	$RPE_t \downarrow$	$RPE_r \downarrow$	ATE$\downarrow$	RPE_t	RPE_r	ATE
0079_00	0.752	**0.204**	0.023	**0.655**	0.221	**0.012**
0418_00	**0.455**	**0.119**	**0.015**	0.491	0.124	0.016
0301_00	**0.399**	**0.123**	0.013	0.414	0.136	**0.009**
0431_00	1.625	0.274	0.069	**1.292**	**0.249**	**0.051**
mean	0.808	**0.180**	0.030	**0.713**	0.182	**0.022**

Table 4. **Comparison of pose accuracy with COLMAP on ScanNet.**



(a) COLMAP+NeRF (b) Ours (c) Ground Truth

Figure 6. **COLMAP failure case.** On a rotation-dominant sequence with low-texture areas, COLMAP fails to estimate correct poses, which results in artifacts in synthesised images.

	scenes	Ours			Ours-r			COLMAP+NeRF		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
ScanNet	0079_00	32.47	0.84	0.41	**33.12**	**0.85**	**0.40**	31.98	0.83	0.43
	0418_00	**31.33**	**0.79**	**0.34**	30.49	0.77	0.40	30.60	0.78	0.40
	0301_00	29.83	0.77	0.36	**30.05**	**0.78**	**0.34**	30.01	**0.78**	0.36
	0431_00	33.83	0.91	0.39	**33.86**	**0.91**	**0.39**	33.54	**0.91**	**0.39**
	mean	31.86	**0.83**	**0.38**	**31.88**	**0.83**	**0.38**	31.53	0.82	0.40
Tanks and Temples	Church	25.17	0.73	0.39	**26.74**	**0.78**	**0.32**	25.72	0.75	0.37
	Barn	26.35	0.69	0.44	26.58	**0.71**	**0.42**	**26.72**	**0.71**	**0.42**
	Museum	26.77	0.76	0.35	26.98	0.77	0.36	**27.21**	**0.78**	**0.34**
	Family	26.01	0.74	0.41	26.21	0.75	0.40	**26.61**	**0.77**	**0.39**
	Horse	27.64	**0.84**	**0.26**	**28.06**	**0.84**	**0.26**	27.02	0.82	0.29
	Ballroom	25.33	0.72	**0.38**	**25.53**	**0.73**	**0.38**	25.47	**0.73**	**0.38**
	Francis	29.48	0.80	**0.38**	29.73	**0.81**	**0.38**	30.05	**0.81**	**0.38**
	Ignatius	23.96	0.61	0.47	23.98	**0.62**	**0.46**	24.08	0.61	0.47
	mean	26.34	0.74	0.39	**26.73**	**0.75**	**0.37**	26.61	**0.75**	0.38

Table 5. **Comparison to NeRF with COLMAP poses.** Our two-stage method (Ours-r) outperforms both COLMAP+NeRF and our one-stage method (Ours).

stage method shows higher accuracy than the one-stage method. We hypothesise that the joint optimisation (from randomly initialised poses) in the one-stage approach causes the NeRF optimisation to be trapped in a local minimum, potentially due to the bad pose initialisation. The two-stage approach circumvents this issue by re-initialising the NeRF and re-training with well-optimised poses, resulting in higher performance.

4.4. Ablation Study

In this section, we analyse the effectiveness of the parameters and components that have been added to our model. The results of ablation studies are shown in Tab. 6.

Effect of Distortion Parameters. We find that ignoring depth distortions (*i.e.*, setting scales to 1 and shifts to 0 as constants) leads to a degradation in pose accuracy, as inconsistent distortions of depth maps introduce errors to the estimation of relative poses and confuse NeRF for geometry reconstruction.

	NVS			Pose		
	PSNR↑	SSIM ↑	LPIPS ↓	$RPE_t \downarrow$	$RPE_r \downarrow$	ATE $\downarrow$
Ours	**31.86**	**0.83**	**0.38**	**0.801**	**0.181**	**0.031**
Ours w/o α, β	31.46	0.82	0.39	1.929	0.321	0.066
Ours w/o L_{pc}	31.73	0.82	**0.38**	2.227	0.453	0.101
Ours w/o L_{rgb-s}	31.05	0.81	0.41	1.814	0.401	0.156
Ours w/o L_{depth}	31.20	0.81	0.40	1.498	0.383	0.089

Table 6. **Ablation study results on ScanNet.**

Effect of Inter-frame Losses. We observe that the inter-frame losses are the major contributor to improving relative poses. When removing the pairwise point cloud loss L_{pc} or the surface-based photometric loss L_{rgb-s}, there is less constraint between frames, and thus the pose accuracy becomes lower.

Effect of NeRF Losses. When the depth loss L_{depth} is removed, the distortions of input depth maps are only optimised locally through the inter-frame losses. We find that this can lead to drift and degradation in pose accuracy.

4.5. Limitations

Our proposed method optimises camera pose and the NeRF model jointly and works on challenging scenes where other baselines fail. However, the optimisation of the model is also affected by non-linear distortions and the accuracy of the mono-depth estimation, which we did not consider.

5. Conclusion

In this work, we present NoPe-NeRF, an end-to-end differentiable model for joint camera pose estimation and novel view synthesis from a sequence of images. We demonstrate that previous approaches have difficulty with complex trajectories. To tackle this challenge, we use mono-depth maps to constrain the relative poses between frames and regularise the geometry of NeRF, which leads to better pose estimation. We show the effectiveness and robustness of NoPe-NeRF on challenging scenes. The improved pose estimation leads to better novel view synthesis quality and geometry reconstruction compared with other approaches. We believe our method is an important step towards applying the unknown-pose NeRF models to large-scale scenes in the future.

Acknowledgements

We thank Theo Costain, Michael Hobley, Shuai Chen and Xinghui Li for their helpful proofreading and discussions. Wenjing Bian is supported by the China Scholarship Council - University of Oxford Scholarship.

References

[1] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. In *SIGGRAPH*, 2001. 2

[2] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *ECCV*, 2022. 2

[3] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *SIGGRAPH*, 1993. 2

[4] Shin-Fang Chng, Sameera Ramasinghe, Jamie Sherrah, and Simon Lucey. Garf: Gaussian activated radiance fields for high fidelity reconstruction and pose estimation. *arXiv e-prints*, 2022. 2

[5] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 5, 11

[6] Paul E Debevec, Camillo J Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *SIGGRAPH*, 1996. 2

[7] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *CVPR*, 2022. 2

[8] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018. 5

[9] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *ICCV*, 2021. 2

[10] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *ICCV*, 2021. 2

[11] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. 2003. 2

[12] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Anima Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. In *ICCV*, 2021. 1, 2, 3, 4, 5, 7, 11

[13] Mijeong Kim, Seonguk Seo, and Bohyung Han. Infonerf: Ray entropy minimization for few-shot neural volume rendering. In *CVPR*, 2022. 2

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015. 5

[15] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 2017. 5, 11

[16] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *CVPR*, 2021. 5

[17] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, 2021. 2

[18] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *ICCV*, 2021. 1, 2, 3, 4, 5, 7, 11

[19] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *TPAMI*, 2015. 5

[20] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ToG*. 5

[21] Quan Meng, Anpei Chen, Haimin Luo, Minye Wu, Hao Su, Lan Xu, Xuming He, and Jingyi Yu. Gnerf: Gan-based neural radiance field without posed camera. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 2

[22] S Mahdi H Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yagiz Aksoy. Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In *CVPR*, 2021. 2

[23] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 2019. 2, 12

[24] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 2021. 1, 2, 3, 5, 7

[25] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.* 2

[26] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *CVPR*, 2022. 2

[27] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *ICCV*, 2021. 2

[28] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021. 2, 4

[29] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *TPAMI*, 2020. 2

[30] Gernot Riegler and Vladlen Koltun. Free view synthesis. In *ECCV*, 2020. 2

[31] Gernot Riegler and Vladlen Koltun. Stable view synthesis. In *CVPR*, 2021. 2

[32] Barbara Roessle, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *CVPR*, 2022. 2

[33] Antoni Rosinol, John J Leonard, and Luca Carlone. Nerf-slam: Real-time dense monocular slam with neural radiance fields. *arXiv preprint arXiv:2210.13641*, 2022. 2

[34] Sara Fridovich-Keil and Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022. 2

[35] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1, 2

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 11

[37] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *NeurIPS*, 2020. 2

[38] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *IROS*. IEEE, 2012. 5

[39] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew Davison. iMAP: Implicit mapping and positioning in real-time. In *ICCV*, 2021. 2

[40] Libo Sun, Jia-Wang Bian, Huangying Zhan, Wei Yin, Ian Reid, and Chunhua Shen. Sc-depthv3: Robust self-supervised monocular depth estimation for dynamic scenes. *arXiv preprint arXiv:2211.03660*, 2022. 5

[41] Alex Trevithick and Bo Yang. Grf: Learning a general radiance field for 3d representation and rendering. In *ICCV*, 2021. 2

[42] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *CVPR*, 2020. 2

[43] Shubham Tulsiani, Richard Tucker, and Noah Snavely. Layer-structured 3d scene inference via view synthesis. In *ECCV*, 2018. 2

[44] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *NeurIPS*, 2021. 2

[45] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021. 2

[46] 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*, 2004. 5, 11

[47] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. NeRF−−: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 1, 2, 3, 4, 5, 6, 7, 11

[48] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In *ICCV*, 2021. 2

[49] Yitong Xia, Hao Tang, Radu Timofte, and Luc Van Gool. Sinerf: Sinusoidal neural radiance fields for joint pose estimation and scene reconstruction. 2022. 2

[50] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *NeurIPS*, 2021. 2

[51] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. In *IROS*, 2021. 2

[52] Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Simon Chen, Yifan Liu, and Chunhua Shen. Towards accurate reconstruction of 3d scene shape from a single monocular image. *TPAMI*, 2022. 2

[53] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021. 2

[54] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *NeurIPS*, 2022. 2

[55] Jian Zhang, Yuanqing Zhang, Huan Fu, Xiaowei Zhou, Bowen Cai, Jinchi Huang, Rongfei Jia, Binqiang Zhao, and Xing Tang. Ray priors through reprojection: Improving neural radiance fields for novel view extrapolation. In *CVPR*, 2022. 2

[56] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2

[57] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 5, 11

[58] Zichao Zhang and Davide Scaramuzza. A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry. In *IROS*. IEEE, 2018. 5, 11

[59] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017. 5, 7

[60] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. 2018. 2

[61] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *CVPR*, 2022. 2

Appendix A. Implementation Details

The following sections include more details about the datasets we use, our training procedure and evaluation metrics.

A.1. Dataset

We select sequences containing dramatic camera motions from ScanNet [5] and Tanks and Tamples [15] for training and evaluation. Tab. 7 lists details about these sequences, where *Max rotation* denotes the maximum relative rotation angle between any two frames in a sequence. The sampled images are further split into training and test sets. Starting from the 5*th* image, we sample every 8*th* image in a sequence as a test image. However, this leads to a change in the sampling rate in the temporal domain among training images. We found that the rotation errors are often higher than average at these positions where the sampling rate changes. In order to study the effect of the sampling rate changes, for scene *Family* in Tanks and Temples [15], we sample every other image as test images, i.e. training on images with odd frame ids and testing on images with even frame ids.

	Scenes	Type	Seq. length	Frame rate	Max. rotation (deg)
ScanNet	0079_00	indoor	90	30	54.4
	0418_00	indoor	80	30	27.5
	0301_00	indoor	100	30	43.7
	0431_00	indoor	100	30	45.8
Tanks and Temples	Church	indoor	400	30	37.3
	Barn	outdoor	150	10	47.5
	Museum	indoor	100	10	76.2
	Family	outdoor	200	30	35.4
	Horse	outdoor	120	20	39.0
	Ballroom	indoor	150	20	30.3
	Francis	outdoor	150	10	47.5
	Ignatius	outdoor	120	20	26.0

Table 7. **Details of selected sequences.** We downsample several videos to a lower frame rate. FPS denote frame per second. *Max rotation* denotes the maximum relative rotation angle between any two frames in a sequence. We show our method can handle dramatic camera motion (large maximum rotation angle) whereas previous methods can only handle forward-facing scenes.

A.2. Training Details

During training, we sample 1024 pixels/rays for an image and we sample 128 points along each ray for our approaches and all baselines. For all approaches, we use the same pre-defined sampling range (i.e., near and far) and sample uniformly between this range. During scheduling, the learning rate of NeRF model decays every 10 epochs with 0.9954, and the learning rate for the camera poses decays every 100 epochs with 0.9. As the scene scales can be arbitrary, the optimised scale parameter of the depth map during training is also arbitrary. To avoid scale collapsing (all scales reduced to 0.0) during training, we manually set the scale of the depth map for the last frame to 1.0. We also use the normalised point clouds when computing the inter-frame point cloud loss.

A.3. Test-time Optimisation

During the evaluation for novel view synthesis, following our baselines NeRFmm [47], BARF [18] and SC-NeRF [12], we run a test-time optimisation to align the camera poses of the test set by minimising the photometric error on the synthesised images, while keeping the trained NeRF model froze. Although all these baseline methods have their own way to align camera poses (discussed below), all of them fail to align camera poses in complex camera trajectories in ScanNet and Tanks and Temples.

To fairly evaluate all methods in challenging camera trajectories, we propose to align test camera poses by first initialising from learned poses of adjacent training images, followed by a test-time optimisation. We shorthand this alignment as **Neighbour + opt**. In practice, we find this initialisation is robust and provides the best alignment for all approaches. All results in our main paper are evaluated in this way.

The following paragraphs outline previous alignment methods, and we show a comparison for all method with a ScanNet scene in Tab. 8.

Identity + opt. BARF [18] uses test-time optimisation to identify poses for the test frames, where all poses are initialised with identity matrices. This initialisation works well for simple forward-facing scenes, but not for complex trajectories. The optimisation is sensitive to the learning rate, and can easily fall into local minima when the target pose is far from the identity initialisation.

Sim(3) + opt. In NeRFmm [47], the poses are first initialised using Sim(3) alignment with an ATE toolbox [58]. Then, an additional test-time optimisation is used to further adjust the test poses. This initialisation works well when the learned poses can be aligned precisely to COLMAP poses (Ours in Tab. 8). However, incorrect pose estimations can affect the Sim(3) alignment.

Sim(3) + no opt. In SC-NeRF [12], the test poses are identified using a Sim(3) alignment between COLMAP poses and the learned poses. And no test-time optimisation is used. However, the results are biased toward COLMAP estimations, and misalignment can affect the view synthesis quality significantly.

A.4. Evaluation Metrics

Novel View Synthesis. We use Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) [46] and Learned Perceptual Image Patch Similarity (LPIPS) [57] to measure the novel view synthesis quality. For LPIPS, we use a VGG architecture [36].

	Sim(3) + no opt.			Identity + opt.			Sim(3) + opt.			(4) Neighbour + opt		
	PSNR↑	SSIM ↑	LPIPS ↓	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Ours	17.24	0.62	0.58	13.38	0.39	0.70	32.47	0.84	0.41	32.47	0.84	0.41
BARF	14.68	0.55	0.66	19.56	0.65	0.57	17.82	0.60	0.61	32.31	0.83	0.43
NeRFmm	11.28	0.40	0.80	30.59	0.81	0.49	12.46	0.43	0.80	30.59	0.81	0.49
SC-NeRF	10.68	0.38	0.80	22.39	0.71	0.55	11.25	0.40	0.80	31.33	0.82	0.46

Table 8. **Comparison of various pose alignment methods during test-time optimisation (ScanNet 0079_00).**

scenes	Ours			NeRFmm		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR	SSIM	LPIPS
Fern	**23.01**	**0.71**	**0.38**	20.58	0.59	0.50
Flower	**29.39**	**0.86**	**0.19**	27.02	0.76	0.32
Fortress	**29.38**	**0.80**	**0.28**	24.94	0.57	0.57
Horns	**25.24**	**0.73**	**0.37**	23.67	0.66	0.48
Leaves	**19.85**	**0.60**	**0.40**	19.46	0.55	0.46
Orchids	**19.51**	**0.56**	**0.43**	16.77	0.40	0.55
Room	**28.54**	**0.89**	**0.28**	26.14	0.84	0.39
Trex	**25.82**	**0.84**	**0.29**	24.13	0.77	0.39
mean	**25.09**	**0.75**	**0.33**	22.84	0.64	0.46

Table 9. **Novel view synthesis results on LLFF-NeRF dataset.**

scenes	Ours			NeRFmm		
	RPE_t ↓	RPE_r ↓	ATE↓	RPE_t	RPE_r	ATE
Fern	**0.252**	**0.993**	**0.003**	0.706	1.816	0.007
Flower	**0.035**	**0.096**	**0.001**	0.086	0.418	**0.001**
Fortress	**0.081**	**0.296**	**0.001**	0.233	0.739	0.004
Horns	**0.217**	**0.452**	**0.004**	0.321	0.850	0.008
Leaves	0.218	0.143	0.002	**0.138**	**0.051**	**0.001**
Orchids	**0.203**	**0.383**	**0.003**	0.686	2.030	0.010
Room	**0.244**	**0.936**	**0.004**	0.670	1.664	0.011
Trex	**0.219**	**0.319**	**0.004**	0.542	0.775	0.009
mean	**0.184**	**0.452**	**0.003**	0.423	1.043	0.006

Table 10. **Pose accuracy on LLFF-NeRF dataset.**

Depth. The error metrics we use for depth evaluation include Abs Rel, Sq Rel, RMSE, RMSE log, δ_1, δ_2 and δ_3. The definitions are as follows:

- Abs Rel: $\frac{1}{|\mathcal{V}|}\sum_{d\in\mathcal{V}} \|d - d_{gt}\|/d_{gt}$;
- Sq Rel: $\frac{1}{|\mathcal{V}|}\sum_{d\in\mathcal{V}} \|d - d_{gt}\|_2^2/d_{gt}$;
- RMSE: $\sqrt{\frac{1}{|\mathcal{V}|}\sum_{d\in\mathcal{V}} \|d - d_{gt}\|_2^2}$;
- RMSE log: $\sqrt{\frac{1}{|\mathcal{V}|}\sum_{d\in\mathcal{V}} \|\log d - \log d_{gt}\|_2^2}$;
- δ_i: % of y s.t. $max(\frac{d}{d_{gt}}, \frac{d_{gt}}{d}) = \delta < i$;

where d is the estimated depth, d_{gt} is the ground truth depth, and $\mathcal{V}$ is the collection of all valid pixels on a depth map.

Appendix B. Additional Results

LLFF-NeRF Dataset. We compare our approach against NeRFmm on the LLFF-NeRF dataset [23] in terms of novel view synthesis quality (Tab. 9) and pose accuracy (Tab. 10). We show better performances than NeRFmm in both pose accuracy and synthesis quality. We use the normalized device coordinate (NDC) for both approaches.

Depth Estimation. We show detailed depth evaluation results for ScanNet scenes in Tabs. 11 to 14. Our depth estimation accuracy outperforms other baselines by a large margin.

Pose Estimation. We visualise additional results for pose estimation on Tanks and Temples (Fig. 9) and ScanNet (Fig. 10).

0079_00	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
Ours	**0.099**	**0.047**	**0.335**	**0.128**	**0.904**	**0.995**	**1.000**
BARF	0.208	0.165	0.588	0.263	0.639	0.896	0.983
NeRFmm	0.494	1.049	1.419	0.534	0.378	0.567	0.765
SC-NeRF	0.360	0.450	0.902	0.396	0.407	0.730	0.908
DPT	0.149	0.095	0.456	0.173	0.818	0.978	0.999

Table 11. **Depth map evaluation on ScanNet 0079_00.**

0418_00	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
Ours	**0.152**	**0.137**	**0.645**	**0.185**	**0.738**	**0.988**	**0.997**
BARF	0.718	1.715	1.563	0.630	0.205	0.569	0.769
NeRFmm	0.907	3.650	2.176	0.769	0.240	0.456	0.621
SC-NeRF	0.319	0.441	0.898	0.377	0.456	0.792	0.930
DPT	0.190	0.187	0.745	0.211	0.719	0.965	**0.997**

Table 12. **Depth map evaluation on ScanNet 0418_00.**

0301_00	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
Ours	0.185	0.252	0.711	**0.233**	**0.792**	**0.918**	**0.958**
BARF	**0.179**	**0.146**	**0.502**	0.268	0.736	0.883	0.938
NeRFmm	0.444	0.830	1.239	0.481	0.397	0.680	0.845
SC-NeRF	0.383	0.378	0.810	0.452	0.360	0.663	0.846
DPT	0.317	0.568	1.133	0.350	0.597	0.821	0.914

Table 13. **Depth map evaluation on ScanNet 0301_00.**

0431_00	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
Ours	**0.127**	**0.111**	**0.579**	**0.160**	**0.877**	**0.978**	**0.994**
BARF	0.398	0.710	1.307	0.444	0.381	0.655	0.847
NeRFmm	0.514	1.354	1.855	0.562	0.250	0.539	0.742
SC-NeRF	0.608	1.300	1.706	0.677	0.225	0.446	0.645
DPT	0.132	0.135	0.670	0.171	0.855	0.973	0.991

Table 14. **Depth map evaluation on ScanNet 0431_00.**

More Visualisations. We present additional qualitative results for novel view synthesis and depth estimation on Tanks and Temples (Fig. 7) and ScanNet (Fig. 8).



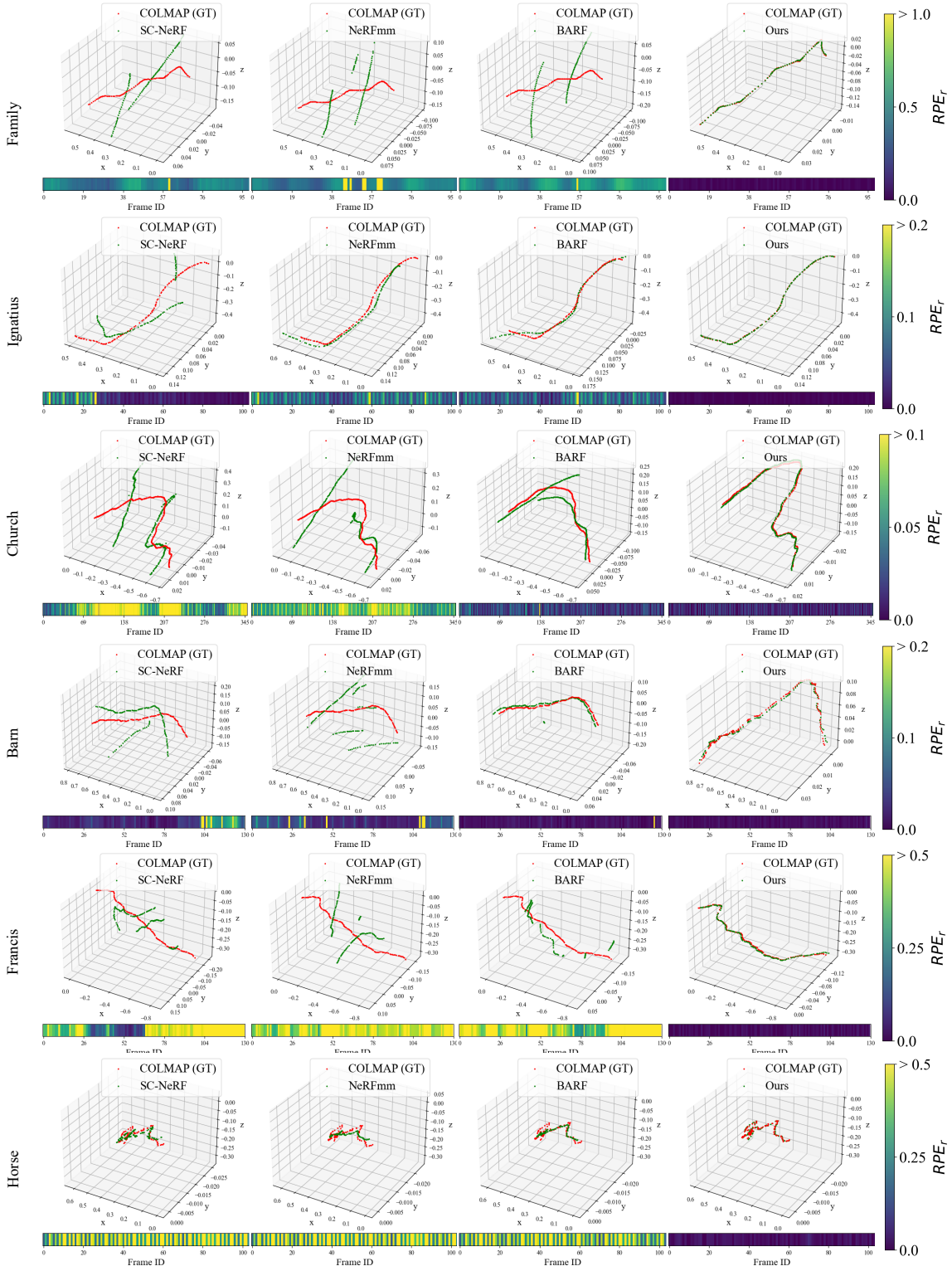
(a) BARF (b) NeRFmm (c) SC-NeRF (d) Ours (e) Ground Truth

Figure 7. **Qualitative results of novel view synthesis and depth prediction on Tanks and Temples.** We visualise the synthesised images and the rendered depth maps (top left of each image) for all methods. NoPe-NeRF is able to recover details for both colour and geometry.



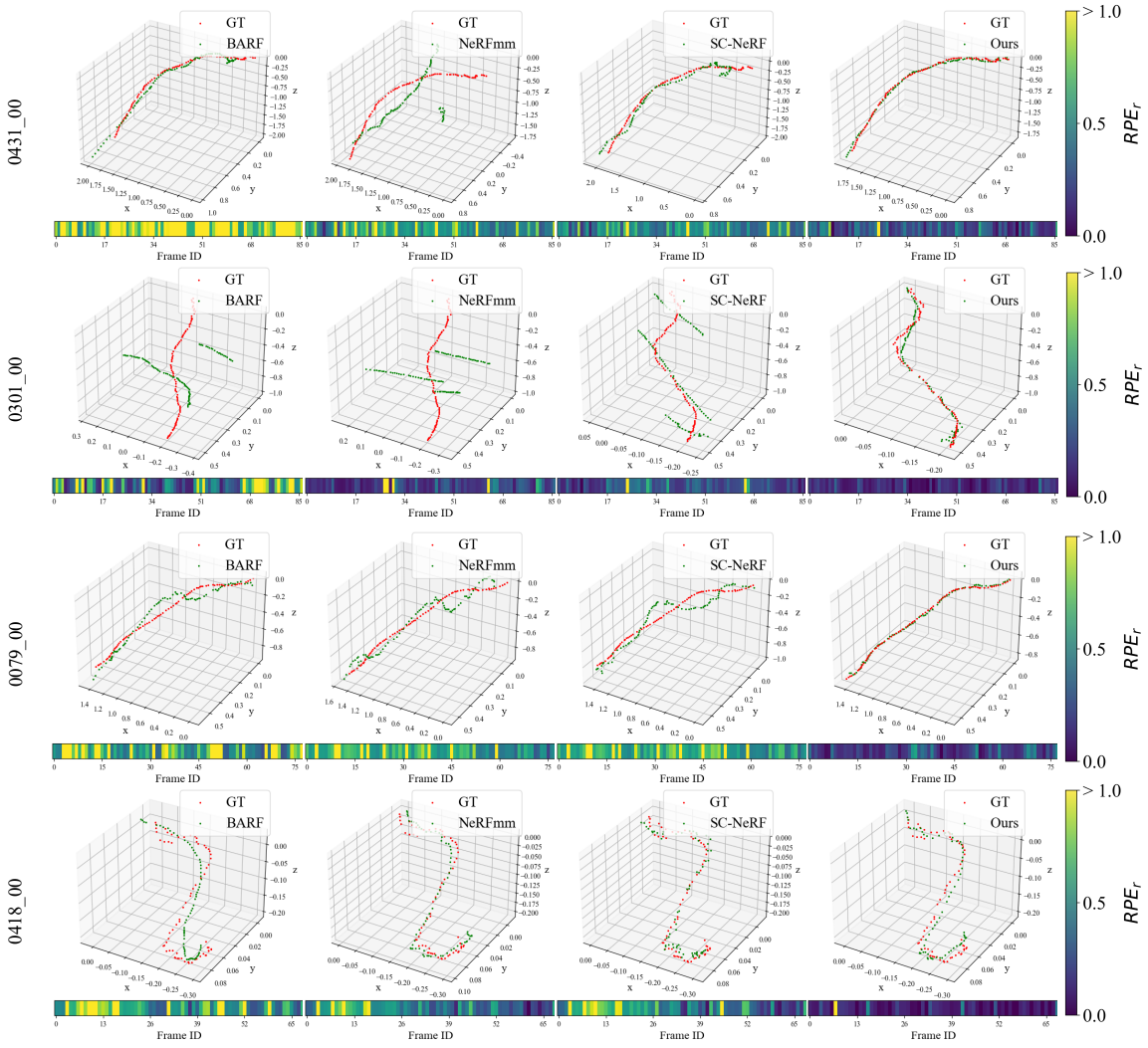
(a) BARF (b) NeRFmm (c) SC-NeRF (d) Ours (e) Ground Truth

Figure 8. **Qualitative results of novel view synthesis and depth prediction on ScanNet.** We visualise the synthesised images and the rendered depth maps (top left of each image) for all methods. NoPe-NeRF is able to recover details for both colour and geometry.



(a) BARF (b) NeRFmm (c) SC-NeRF (d) Ours

Figure 9. **Pose Estimation Comparison on Tanks and Temples.** We visualise the trajectory (3D plot) and relative rotation errors RPE_r (bottom colour bar) of each method on *Ballroom* and *Museum*. The colour bar on the right shows the relative scaling of colour.



(a) BARF (b) NeRFmm (c) SC-NeRF (d) Ours

Figure 10. **Pose Estimation Comparison on ScanNet.** We visualise the trajectory (3D plot) and relative rotation errors RPE_r (bottom colour bar) of each method on *Ballroom* and *Museum*. The colour bar on the right shows the relative scaling of colour.

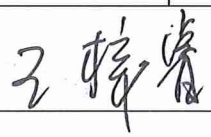
Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	NoPe-NeRF: Optimising Neural Radiance Field with No Pose Prior
Publication Status	☑ Published ☐ Accepted for Publication ☐ Submitted for Publication ☐ Unpublished and unsubmitted work written in a manuscript style
Publication Details	Wenjing Bian, **Zirui Wang**, Kejie Li, Jia-Wang Bian, Victor Adrian Prisacariu. In Conference on Computer Vision and Pattern Recognition (CVPR), 2023.

Student Confirmation

Student Name:	Zirui Wang		
Contribution to the Paper	Second-author contribution: • Conception of research ideas. • Design models. • Writing and presentation of the paper.		
Signature		Date	20/Sep/2023

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	Professor Victor Adrian Prisacariu		
Supervisor comments:	The description is accurate.		
Signature		Date	20/Sep/2023

This completed form should be included in the thesis, at the end of the relevant chapter.

Chapter 5

2D – 2D Registration with Interest Point Discovery

This chapter addresses the issue of image registration, specifically the alignment of two images via affine transformation estimation. As outlined in Section 2.4, numerous methods exist for this purpose; however, our focus lies on the field of interest point detection, a crucial element in estimating fundamental matrices and homography transformation.

The ability to handle significant viewpoint changes is a key property desired in interest point detection. Existing methods [54, 55, 56, 57, 58, 59, 60] have largely been confined to detecting points within a single image. Although this single-view approach allows efficient pipelines, it imposes bottlenecks when dealing with extensive viewpoint alterations. This has led to higher-level interest point development, particularly in human and hand imagery [61, 62, 63, 64], but broadening this to more general scenes is challenging.

To address these limitations, we introduces a novel interest point discovery network designed to handle input of multi-view images. This multi-view framework enables the network to distinguish local image appearance from factors such as texture and 3D shapes.

Specifically, our contributions are twofold: *First*, we introduce the concept of Abstract Interest Point (AIP) in this chapter, representing a high-level keypoint consistent across various conditions and applicable to diverse scenes. *Second*, we present a self-supervised approach to train a network for this task. Our training approach involves a geometric-based self-supervised strategy with a contrastive and triangulation term. This method focuses on overall scene structures rather than local surface textures, similar to human keypoint discovery but without specific scene content assumptions.

Consequently, our multi-view interest point discovery method significantly enhances the repeatability of identified interest points, which is beneficial to the theme of 2D – 2D registration.

AIP: Abstract Interest Point Discovery from Multi-view Self-supervision

Zirui Wang*
University of Oxford
zirui.wang@eng.ox.ac.uk

Yipu Zhao*
Meta Reality Lab
yipuz@meta.com

Tanner Schmidt
University of Oxford
tanner.schmidt@meta.com

Tomasz Malisiewicz
Meta Reality Lab
tomasz.malisiewisz@meta.com

Jakob Engel
Meta Reality Lab
jakob.engel@meta.com

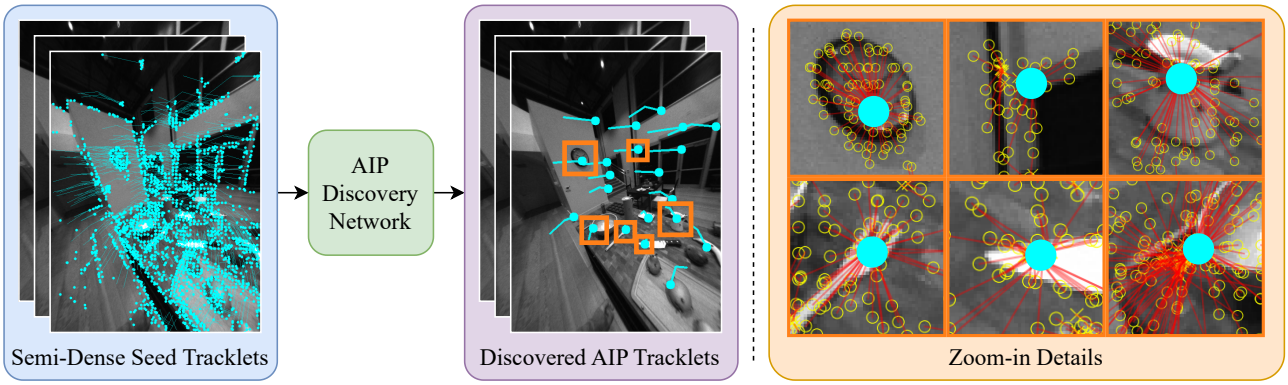


Figure 1. **Abstract interest point discovery.** Our method learns to discover highly repeatable abstract interest points from multi-view images. **Left:** the input of our method, including 1) multi-view images, and 2) semi-dense seed tracklets obtained from tracking low-level image features using point tracking such as optical flow or particle video methods [30, 47, 61]. Seed tracklets are usually sampled at high-texture image area. **Center:** the output of our method, abstract interest points (AIP), discovered by searching at local solution space from each seed tracklet. The AIP is learned by the proposed geometric-only self-supervision, without any appearance hint. **Right:** Detailed zoom-in patches. Starting from each initial tracklet (yellow circle), our method discovers the optimal local interest point (cyan point). Note that the high consensus of per-tracklet independent discoveries, leading to a set of sparse and condensed solutions. For simplicity, we only show a single view out of multi-view input and output.

Abstract

We re-visit the fundamental challenge of interest point discovery in computer vision. Existing local keypoint methods are closely tied to low-level image features that are mostly latched on physical surface, therefore can only be detected within limited range of view points or illumination conditions. Inspired by the abstract interest points that are adapted for specific categories such as human or hands, we aim to discover abstract interest points (AIP) that are 1) highly repeatable towards view point changes and occlusion, and 2) remain generalizable to diverse scenarios. To that end, we proposed a novel self-supervised training method that only relies on geometric consistency, eliminating the tendency of over-indexing on low-level image features. Experimental results demonstrate that the discovered AIPs are distinct from existing local keypoints derived from low-level image features, providing extra robustness under challenging conditions while remaining sparsely distributed, and can operate on a wide range of indoor and outdoor scenarios.

*Equal contribution. Author order may alternate for equal visibility.

1. Introduction

Interest points provide a sparse set of anchors upon which to establish reliable visual correspondence, and are therefore essential elements in computer vision. A core property of good interest points is their repeatability: points should be consistently identified at the same position under a variety of view points, lighting conditions, and partial occlusions. The most common interest points are local keypoints, i.e., based on low-level image features such as blobs and corners, and detected with either hand-written algorithms [14, 29, 50] or neural networks [6, 12, 37, 60].

One major limitation of depending on low-level image features for interest point detection is the fact that these features are mostly at physical surfaces and are therefore only reliably detectable within a limited range of viewing angles, viewing distances, and illumination conditions. This motivates the pursuit of interest point at a higher level, i.e., abstracted beyond low-level image features, and thus more repeatable to detect. For specific categories such as human and hands, abstract interest points have been widely adapted, in either human annotated or self-supervised manner [18, 48, 49, 58]. Generalizing beyond designated categories, however, remains a challenge for existing abstract interest point detection methods.

In this paper, we aim to identify *abstract interest points* (AIPs) that are highly repeatable across a wide range of viewing conditions yet remain generalizable to diverse scenes. It is difficult to directly define an algorithm for detecting good abstract interest points, and direct supervision is non-trivial even when using neural networks to learn the detection function. Instead, we identify the desired properties of interest points and design losses to encourage the discovery of interest points with these properties.

Specifically, we want points to be 1) *geometrically consistent*, in the sense that multiple detections of the same interest point triangulate to a single 3D point, and 2) *repeatable*, in the sense that multiple observations of the same scene from disparate viewing angles and / or under different illumination should lead to the detection of the same interest points.

The former property is encouraged by detecting AIPs in a multi-view image series and penalizing geometric inconsistency using differentiable triangulation [35]. The latter property is encouraged by a loss that induces clustering of keypoint detections in 3D. It is this cluster-inducing loss that makes the detected interest points abstract – in order to reliably detect the same points in many views of the same scene, the network must avoid over-indexing on local surface texture and instead identify scene structure that is more broadly observable. This approach echoes similar ideas in the human keypoint discovery literature [18, 48, 49, 58], but in contrast to this prior work we do not make any assumptions on scene content.

In summary, we present two major contributions. **First**, we introduce a novel concept of *abstract interest points* (AIPs) that are highly repeatable across various viewing conditions and generalizable to different scenes, moving beyond low-level image feature-based keypoints. **Second**, we propose a novel self-supervised strategy to identify AIPs effectively. This strategy comprises two terms: a contrastive term that ensures AIPs to be repeatable and sparse, and a triangulation term that encourages AIPs to be multi-view consistent. As a result, we show AIPs offer superior keypoint identification performance in terms of repeatability and related pose estimation, as well as more accurate point tracking in comparison with modern 2D trackers.

The rest of this paper is structured as follows: We first review related work in Sec. 2, then delve into our methodology in Sec. 3, followed by a discussion of our results and limitations in Sec. 4, and conclude in Sec. 5.

2. Related Work

Three groups of prior work are reviewed: 1) *generic* interest point identification, which are broadly applicable across various scenarios; 2) *categorial* interest point identification, which are tailored for recognizing interest patterns in specific categories; and 3) *point tracking*, which are designed to track 2D query points across a series of frames.

Generic Interest Point Identification. A long history of works exist on detecting interest point with low-level image features using hand-crafted rules [14, 25, 29] or neural network [6, 9, 52, 54]. More recently, progress has been made towards discovering interest points with self-supervision, further reducing the level of human input to the learning process. To construct the self-supervision, existing works assume the availability of pixel-level correspondence, which is established either with homography (therefore limit to planar scene) [2, 24, 41, 56, 57] or other low-level feature based algorithm such as optical flow or structure from motion [12, 33, 37, 54, 59, 60]. In other word, the target interest points that these self-supervised discovery methods are trying to find, are either by-definition on physical surface, or derived from low-level image features that are mostly at physical surface. Over-indexing on local surface textures is not ideal, as it places an inherent limit on the range of viewing angle / distance / illumination condition that the interest points can be detected from.

Categorial Interest Point Identification. There has been long interest in identifying interest point for specific categories, *e.g.*, human [4, 11], face [5, 36] or hand [3, 55]. To avoid laborious manual labeling and to facilitate utilizing of large amount of unlabeled in-the-wild data, various self-supervision attempts have been made using single im-

age input [18, 19, 49, 58] or multiple-view [46]. Overall, this line of work heavily relies on category priors to limit the solution space. In contrast, we aim to discover interest points that are more generic and can be applied to a variety of scenarios.

Point Tracking. Establishing correspondence is a fundamental challenge in vision tasks. While explicit feature matching methods [27, 39, 45] are viable for tracking query points, our method aligns more closely to flow-like point tracking approaches. For example, optical flow [8, 15, 16, 30, 44, 47] and correspondence flow [17, 21, 22, 28, 53] methods estimate displacements between adjacent frames. Particle video methods [7, 13, 20, 38, 61] track query points in a window of consecutive images. While point tracking methods produce outputs in similar forms to ours, they primarily address the challenge of effectively tracking a query point, without particular consideration for repeatability and distinctiveness. In contrast, our method emphasizes not only tracking but also driving query input to more repeatable and distinctive points.

3. Method

Given a set of input images and a seed tracklet tracking low-level image features within these images, our goal is to discover a new tracklet within some proximity of the seed, whose 2D points are 1) projections of the same 3D point, i.e., **geometrically consistent**, and 2) can be reliably rediscovered under viewpoint and occlusion variations, i.e., **repeatable**. We refer the discovered new tracklet or these points as Abstract Interest Points (AIPs).

Formally, let's define the multi-view images input $\mathcal{I}_w = \{\mathbf{I}_i | i = 1 \dots N_{\text{img}}\}$, where N_{img} denotes the number of images in a window and w the window index. From $\mathcal{I}_w$ and common point tracking methods [7, 20, 30, 38, 47, 61], we get a seed tracklet $\mathcal{P} = \{\mathbf{p}_i \in \mathbb{R}^2 | i = 1 \dots N_{\text{img}}\}$. Our goal is to find a function $g(\cdot)$ that predicts a AIP tracklet $\hat{\mathcal{P}}$ from a seed tracklet $\mathcal{P}$, conditioning on multi-view observations $\mathcal{I}_w$:

$$\hat{\mathcal{P}} = g(\mathcal{P} | \mathcal{I}_w). \tag{1}$$

In practice, to avoid numerical issues, we predict a set of offset vectors that shift a seed tracklet to a nearby AIP tracklet:

$$\mathcal{O} = g(\mathcal{P} | \mathcal{I}_w), \tag{2}$$

where $\mathcal{O} = \{\mathbf{o}_i \in \mathbb{R}^2 | i = 1 \dots N_{\text{img}}\}$, and $\mathbf{o}_i$ shifts an initial 2D point $\mathbf{p}_i$ to a new 2D point $\hat{\mathbf{p}}_i$ in view i:

$$\hat{\mathbf{p}}_i = \mathbf{p}_i + \mathbf{o}_i. \tag{3}$$

We approximate function g with a neural network (Sec. 3.2).

The key challenge is how to supervise g to predict a set of $\mathbf{o}_i$ (and $\hat{\mathbf{p}}_i$ thereafter), such that the resulting AIP tracklet is geometrically consistent and repeatable. We cover this in the next section.

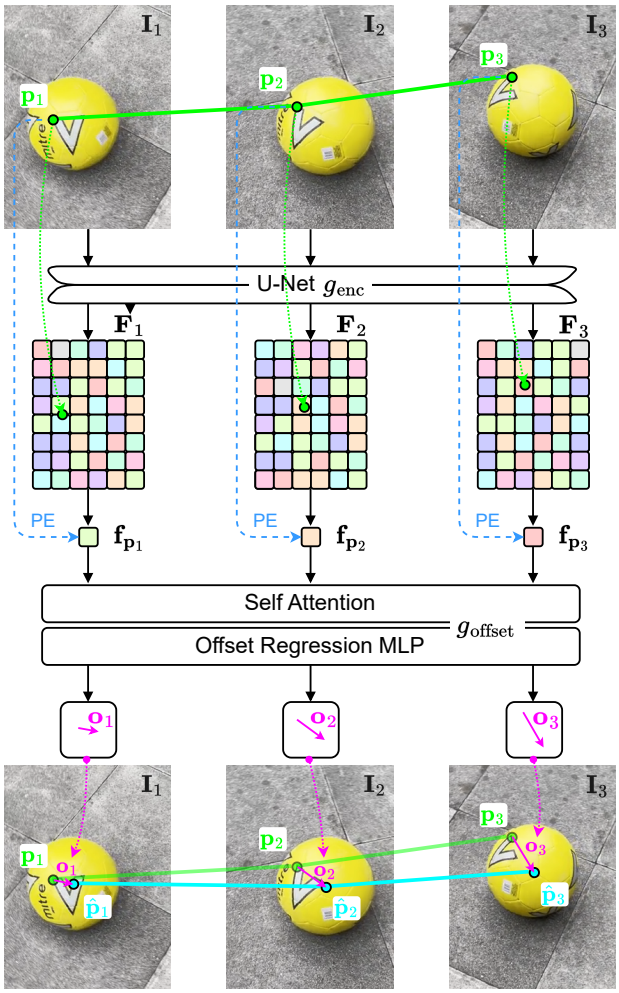


Figure 2. **AIP discovery at inference time.** Starting from an input multi-image window $\mathcal{I} = \{\mathbf{I}_i\}_1^3$, our network first encodes images to feature maps $\{\mathbf{F}_i\}_1^3$. Next, feature vectors $\mathcal{F} = \{\mathbf{f}_{\mathbf{p}_i}\}_1^3$ are extracted for a seed tracklet $\mathcal{P} = \{\mathbf{p}_i\}_1^3$ and get positional encoded (PE). Subsequently, the network predicts offset vectors $\mathcal{O} = \{\mathbf{o}_i\}_1^3$ pointing to a new tracklet that tracks a nearby AIP $\hat{\mathcal{P}} = \{\hat{\mathbf{p}}_i\}_1^3$. In this toy example, the seed tracklet $\mathcal{P}$ is initialized by tracking a corner on a football's surface, which is prone to self-occlusion. In contrast, the AIP tracklet $\{\hat{\mathcal{P}}\}$ consistently locates at the football's geometric center, effectively adapting to variations in viewpoint and occlusion.

3.1. Self-Supervised Training

Instead of setting up self-supervision based on a single tracklet, which is derived from low-level image features therefore provides limited repeatability, the key idea of our self-supervision is to utilize a pair of tracklets whose corresponding 3D points are spatial neighbors. The intuition

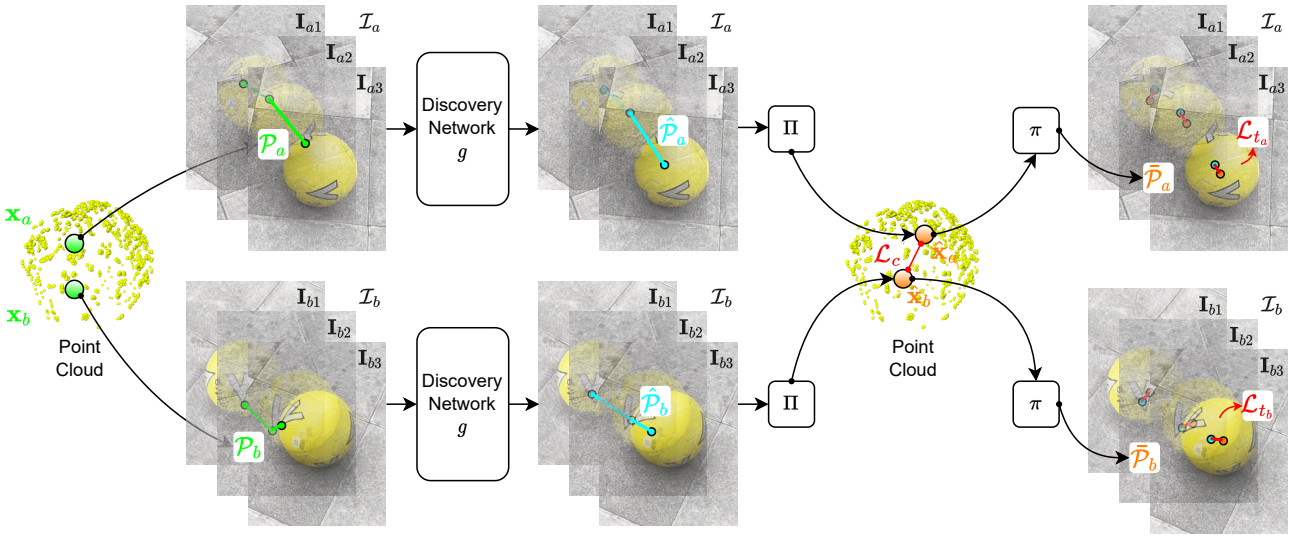


Figure 3. **Self-supervised training with a pair of tracklets.** Our network utilizes two types of self-supervisory signals for training: triangulation losses $\mathcal{L}_{t_{a/b}}$ and contrastive loss $\mathcal{L}_c$. Specifically, the process begins with two input seed tracklets $\mathcal{P}_{a/b}$. From these, the network does a forward pass and predicts two corresponding tracklets $\hat{\mathcal{P}}_{a/b}$. These allow us to triangulate (Π) two 3D points $\hat{\mathbf{x}}_{a/b}$. At backward pass, we first collect the triangulation losses $\mathcal{L}_{t_{a/b}}$ by re-projecting (π) these 3D points to each image window $\mathcal{I}_{a/b}$. Then with these 3D points we apply a contrastive loss $\mathcal{L}_c$ based on their distanced_{ab}. The gradients derived from $\mathcal{L}_{t_{a/b}}$ forces $\hat{\mathcal{P}}_{a/b}$ to be consistent across views, while the gradients from $\mathcal{L}_c$ either draw closer or repel the points. For simplicity, we use a/b to denote independent terms for tracklets $\mathcal{P}_a$ and $\mathcal{P}_b$. In essence, these losses are the key unlocking the capability of discovering AIPs.

is that, if the network can predict the same tracklet from two neighboring seed tracklets that are observed from diverse view points, the prediction must be highly repeatable, *i.e.*, is an AIP. Furthermore, to stop the network from over-indexing on local surface texture, it is crucial to drop the dependency on low-level image features entirely. To that end, we propose propose a self-supervision strategy that is built upon a pair of tracklets, leveraging two types of geometry-only constraints.

Considering a pair of seed tracklets $\mathcal{P}_a$ and $\mathcal{P}_b$ whose corresponding 3D points are close in 3D space but are observed from image windows $\mathcal{I}_a$ and $\mathcal{I}_b$ with potentially distinct appearance, our network predicts two sets of offsets independently:

$$\mathcal{O}_a = g(\mathcal{P}_a|\mathcal{I}_a), \tag{4}$$
$$\mathcal{O}_b = g(\mathcal{P}_b|\mathcal{I}_b), \tag{5}$$

from which two new 3D points $\hat{\mathbf{x}}_a$ and $\hat{\mathbf{x}}_b$ can be estimated with differentiable triangulation:

$$\hat{\mathbf{x}}_a = \Pi(\mathcal{P}_a + \mathcal{O}_a, \mathcal{T}_a), \tag{6}$$
$$\hat{\mathbf{x}}_b = \Pi(\mathcal{P}_b + \mathcal{O}_b, \mathcal{T}_b), \tag{7}$$

where $\mathcal{T}_a$ and $\mathcal{T}_b$ denote camera parameters required by triangulation, π the 3D-to-2D projection, and Π the 2D-to-3D multi-view triangulation.

With $\hat{\mathbf{x}}_a$ and $\hat{\mathbf{x}}_b$, we design a constrastive loss $\mathcal{L}_c$ to pull two neighboring 3D points together, while avoiding the degenerated solution of all points being pushed to one location. Mathematically, the 3D contrastive loss is defined as:

$$\mathcal{L}_c = \begin{cases} d_{ab}^2 & \text{if } d_{ab} \leq r_g, \\ c(d_{ab}^2 - r_e^2) & \text{if } r_e > d_{ab} > r_g, \\ 0 & \text{if } d_{ab} \geq r_e, \end{cases} \tag{8}$$

where d_{ab} is the euclidean distance between 3D points $\hat{\mathbf{x}}_a$ and $\hat{\mathbf{x}}_b$, effective radii r_g and r_e control the loss behavior between pulling and repelling two points, and $c = r_g^2/(r_g^2 - r_e^2)$ is a constant to limit repelling strength.

To encourage geometrical consistency for discovered tracklets, we penalize a per-tracklet triangulation loss $\mathcal{L}_t$ that summarizes re-projection error on each view:

$$\mathcal{L}_t = \frac{1}{N_{\text{img}}} \sum_{i=1}^{N_{\text{img}}} |\hat{\mathbf{p}}_i - \bar{\mathbf{p}}_i|_2, \tag{9}$$

where $\hat{\mathbf{p}}_i$ is a discovered position and $\bar{\mathbf{p}}_i = \pi(\hat{\mathbf{x}}, \mathbf{T}_i)$ is a re-projected position on view i.

The full self-supervision loss is a weighted sum of both losses $\mathcal{L} = \lambda_t \mathcal{L}_t + \lambda_c \mathcal{L}_c$, where λ_t and λ_t are weighting terms. Note that $\mathcal{L}$ does not require the identified AIPs to adhere to any physical surface. Note both $\mathcal{L}_t$ and $\mathcal{L}_c$ can be extended to more than a pair of tracklets.

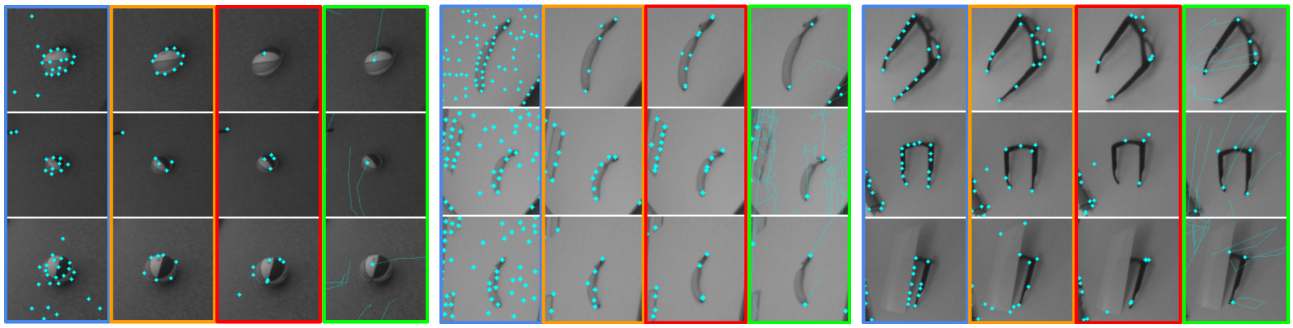


Figure 4. **Example interest points identified by various methods**. In each example, from left to right: blob detector (SIFT), corner detector (SuperPoint), single-view discovery (ALIKED), and multi-view discovery (AIP). Top to bottom: zoom-in crops of multiple example images captured at diverse view points or with occlusions. All methods are configured to output points without any budget limits. Since our network predicts tracklets (a 2D position on each view), we show 2D positions at the current view with predictions from previous views appended as tails.

At training time, we assume both reconstructed point clouds and all camera parameters (poses and intrinsics) are available to support tracklets seeding and triangulation. At inference time, the network no longer needs to access camera parameters, and can operate as a video-based point detector / tracker.

In summary, with the geometric-only self-supervision design, the network is forced to search for AIP that are beyond low-level image features.

3.2. Network Structure

The network for function $g(\cdot)$ consists of two components: 1) a U-Net encoder g_{enc}, which encodes input images to feature maps and 2) an attention-based [51] offset predictor g_{offset}, which regresses a set of offset for a seed tracklet. Specifically, we encode each input view $\mathbf{I}_i$ to a feature map:

$$\mathbf{F}_i = g_{\text{enc}}(\mathbf{I}_i), \tag{10}$$

from which we extract a feature vector $\mathbf{f}_{\mathbf{p}_i}$ on point coordinate $\mathbf{p}_i$ with a bi-linear interpolation operation $\langle\cdot\rangle$:

$$\mathbf{f}_{\mathbf{p}_i} = \mathbf{F}_i\langle\mathbf{p}_i\rangle. \tag{11}$$

Additionally two more positional embedding are concatenated to feature vector $\mathbf{f}_{\mathbf{p}_i}$: one encoding frame order, the other encoding relative pixel coordinates as in PIPs [13].

Assembling all extracted features from tracklet $\mathcal{P}$ forms a set of feature vectors $\mathcal{F} = \{\mathbf{f}_{\mathbf{p}_i} | i = 1 \dots N_{\text{img}}\}$, which contain appearance and geometric information of the initial tracklet. From the feature set $\mathcal{F}$, the offset regressor g_{offset} predicts a set of offsets that adjust the seed tracklet $\mathcal{P}$:

$$\mathcal{O} = g_{\text{offset}}(\mathcal{F}). \tag{12}$$

In practice, we implement the offset regressor g_{offset} with self-attention layers and an MLP head. This offset prediction process can be applied in an iterative manner, further refining discovered tracklets and extending discovery scope. The overall network design, as well as a toy example of our method running at inference time, is illustrated in Fig. 2.

4. Experiments

We begin this section by briefly outlining implementation details, then move to Sec. 4.1, presenting visual comparisons to illustrate the differences between AIP and various baseline methods. In Sec. 4.2, we first conduct a comprehensive point discovery evaluation under challenging scenarios. Subsequently, in Sec. 4.3, we assess the point tracking performance of AIP against well-established point tracking baselines. The experiments focus on egocentric datasets, for three reasons: their larger image collection, the variety of viewpoints they offer, and their growing relevance in virtual/augmented reality and contextual AI research.

Implementation Details. Our dataset combines 27 hours of 640×480 gray-scale video, mixing internal data with segments of the Aria Pilot dataset [31]. These are processed using a visual-inertial SLAM algorithm [10] for accurate camera parameters (poses and intrinsics) and 3D point clouds. The network is trained on 192 V100 GPUs for 5 days, employing a one-cycle learning rate schedule with starting rates of 1e-4 for g_{enc} and 5e-4 for g_{offset}, and a consistent weight decay of 1e-4. We select a window size of 9 for extracting temporal windows from videos. The number of steps for iterative prediction is set to 3. For contrastive loss $\mathcal{L}_c$, we set the radii r_g and r_e to 0.2m and 0.3m, respectively. We apply standard Non-Maximum Suppression with a 5-pixel radius, using point scores from local point voting. Seed tracklets are initialized via a motion-assisted KLT tracker. Further details are available in the supplementary material.

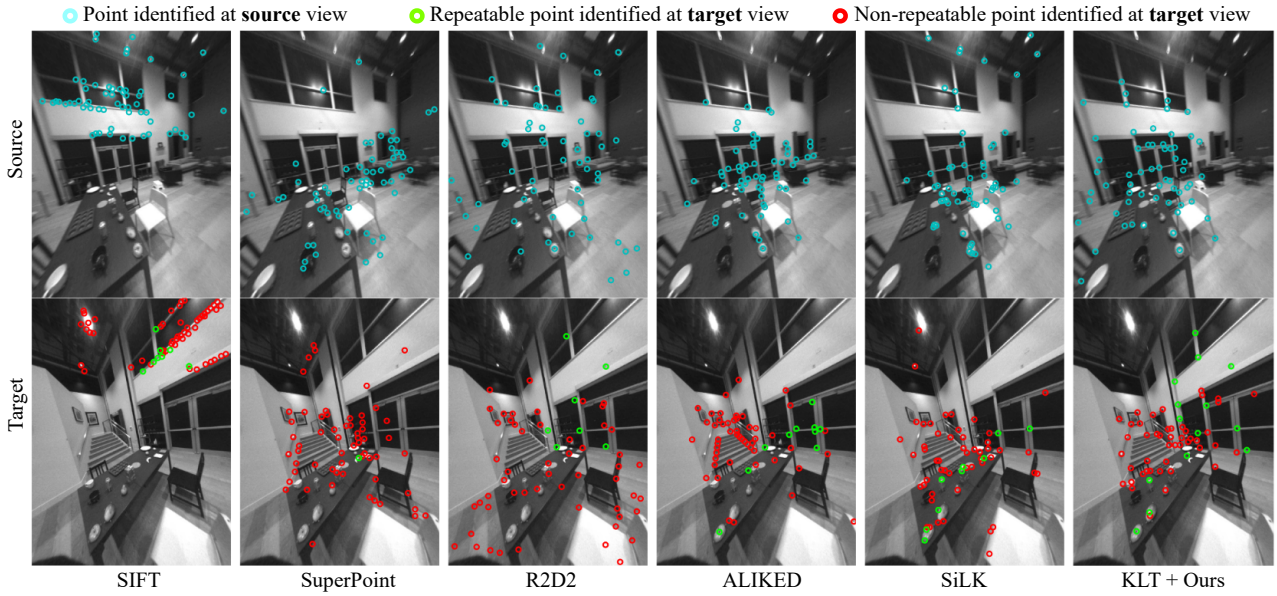


Figure 5. **Repeatability under dramatic view angle changes (70-deg)**. All methods are configured to identity top 60 points in both source and target views. AIPs demonstrate higher repeatability and larger scene coverage, outperforming baseline keypoints.

4.1. Qualitative Evaluation

To demonstrate the limitations of representative existing local keypoint methods, such as SIFT [28], SuperPoint [6], and ALIKED [60], and to highlight the effectiveness of our approach in predicting AIP, we customize a set of controlled scenarios that cover three attributes: symmetrical, low-texture, and occluded, as shown in Fig. 4.

Symmetrical Scenario. We select a basketball as an example of symmetric cases. Our network distinguishes itself in handling symmetrical objects, discovering a stable AIP at the object center. This contrasts with other baselines that heavily focus on unreliable boundaries and surface textures.

Low-texture Scenario. The second case features a banana with notably low-texture patterns. Conventional methods often overemphasize local textural details such as local curvature, picking up inconsistent keypoints during viewpoint changes. In contrast, our network reliably discover two AIPs at both ends of the banana.

Occluded Scenario. The third sequence contains a pair of glasses that is partially covered at the end of recording. This experiment showcases that although our method utilizes information aggregated from a larger scale, it remains robust against information bleeding caused by changes in the scene. As a result all observable AIPs remain detected before and after partial occlusion.

4.2. Quantitative Evaluation: Point Discovery

Preparation. To effectively assess a multi-view point discovery method, an ideal benchmark should offer 1) dense multi-view coverage, 2) precise camera parameters, and 3) accurate depth maps. These elements are crucial for reliable occlusion estimation, point projection, and back-projection. In light of these requirements, we selected the *Aria Digital Twin* (ADT) dataset [34], an egocentric video dataset known for its precise camera parameters and millimeter-level depth accuracy per pixel. Other benchmarks were not considered, as they typically contain only sparse image sets [1, 26, 42] or provide depth maps with inadequate accuracy [40, 42, 43].

In specific, two locations in ADT (*Apartment* & *Minimall*) that contain diverse camera motion and complex scene structure are used. Out of all video frames, 9100 image pairs are sampled with two criteria: 1) having sufficient scene overlap so repeatable points exist in theory, and 2) temporally distant enough to avoid our multi-view network taking both images as input, therefore getting an unfair advantage (i.e., tracking rather than detecting). Example image pairs from *Apartment* are illustrated in Fig. 5.

The availability of accurate per-pixel depth map is significant here, since it disentangles the repeatability of detected interest point from the impact of descriptor matchability: we can directly examine whether a point detected at frame i is repeatedly detected at another frame j by first back-project from i to 3D, then project to frame j. To support non-surface AIP that our network could predict (e.g., center of sphere), we relaxed the depth used in back-projection to

+/-10 cm around physical surface.

Metrics. We report two well-recognized metrics in the field of local keypoint detection. First, *repeatability* [32] assesses the ability to re-identify keypoints across image pairs. Second, we utilize related pose estimation recall rate, briefly referred to *pose recall*[1], as a measure of the keypoint identification performance. Pose recall quantifies the percentage of correctly recovered relative poses within predefined error thresholds, in our case, 10-degree for rotation and 1-meter for translation. Both metrics are reported across various keypoints budgets, following common convention in [6, 12, 37, 60]

Baselines. Five representative local keypoint detection methods serve as our baselines: SIFT [29], SuperPoint [6], R2D2 [37], ALIKED [60], and SiLK [12].

Performance Under Limited Budget. AIP is designed to discover a sparse set of interest points that are highly repeatable and robust to view point changes, which are beneficial to tasks such as re-localization under tight budget constraint or with coarse-to-fine search strategy. In this experiment, we report repeatability and pose recall under various point budgets K.

In specific, Fig. 5 illustrates a comparison of repeatability across all baselines. This comparison specifically focuses on a pair of images that have a 70-degree difference in viewing angles, where our method discovers a greater number of repeatable points covering the entire field-of-view, meanwhile other baselines are either biased towards specific local texture patterns, or failed to detect repeatable points.

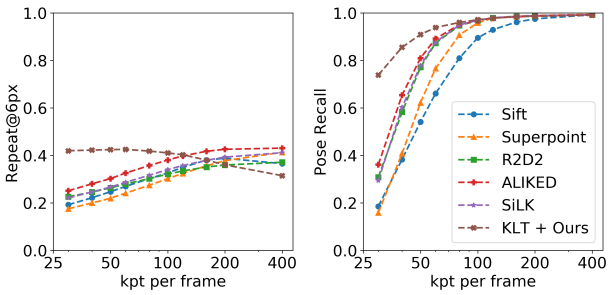


Figure 6. **Repeatability and pose recall under various budgets.** Our method's predicted AIPs significantly outperform all baselines in repeatability and pose recall given limited point budgets ($K \leq 100$), especially notable at $K \leq 60$. This result is attributed to the deliberate sparsity of AIP. The focus on low-budget setups also explains the decreased repeatability when $K > 100$, as the network cannot predict more AIPs with high consensus while keeping sparsity.

[1] Image Matching Challenge 2022.

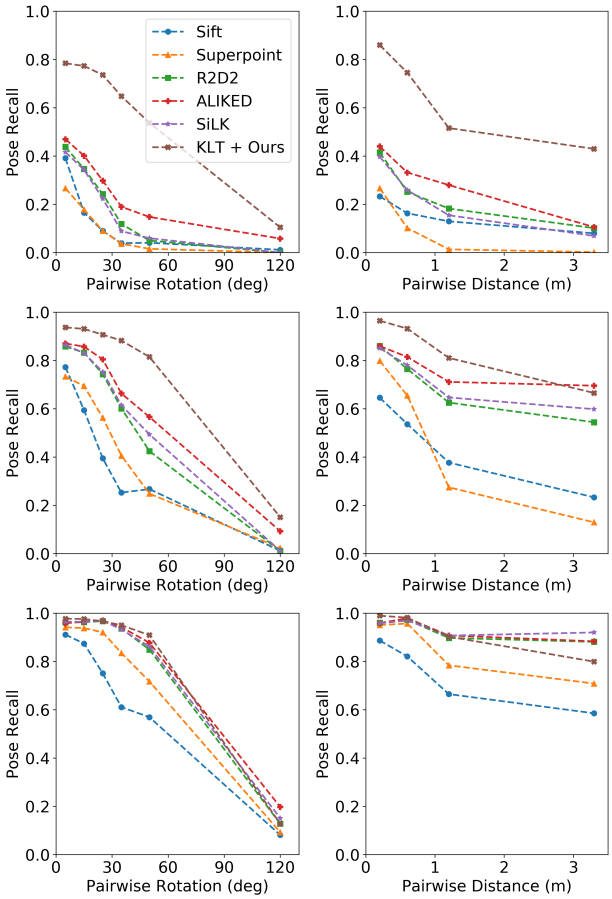


Figure 7. **Pose recall under view point changes.** Left-right: pose recall with regard to relative rotation and translation. Top-bottom: various points budget $K = 30/50/80$. Our method outperforms all baseline under low point budget $K = 30/50$, and is on-par with state-of-the-art baselines [12, 37, 60] when $K = 80$.

For each budget K of a designed range, we report the repeatability and pose recall averaged on all 9100 image pairs. The results are summarized in Fig. 6. We demonstrate that the AIPs predicted by our method significantly outperforms all baselines in terms of both repeatability and pose recall when limited point budgets available $K \leq 100$. The advantage is particularly noticeable for ($K \leq 60$). This is inherently connected to the sparsity natural of discovered AIP, as determined by both the shift towards utilizing high-level image features, and the designated 3D contrastive radii r_g and r_e. In other words, the working space of AIP is exactly aiming at the low budget setups. This also explains the drop of repeatability in our method when $K > 100$, as the network cannot predict more AIPs than it can discover with high consensus, therefore start making random low confident predictions to try to meet the budget.

To understand how viewpoint change affects pose recall,

Tracker	Query	σ	1exRMSE ↓	
			Seed	AIP from seed
PIPs++	ALIKED	-	4.39	**4.30**
	SuperPoint	-	4.58	**4.48**
	HG	-	4.75	**3.11**
KLT	HG	9	3.53	**2.82**
	HG	6	2.69	**1.53**
	HG	3	1.52	**0.94**
	HG	-	**0.001**	0.88

Table 1. **Point tracking accuracy (in pixel) on AP static scenes.** HG denotes high-gradient area and σ denotes i.i.d. uniform noise (in pixel) added to all 2d points in a tracklet. We show our method refines all seed tracklets for better accuracy, whereas PIPs++ is less effective in egocentric videos with significant camera motion. Note KLT tracking with raw HG query presents reprojection error close to 0. This is due to KLT and pseudo ground truth camera parameters share a same SLAM system [10].

we further break view point changes down to rotational and translational components. Fig. 7 illustrates that AIP offers superior pose recall than baseline methods in both rotational and translational camera motion.

4.3. Quantitative Evaluation: Point Tracking

As a multi-view interest point discovery network, an output AIP is not a set of unrelated interest points on each image, but a tracklet that spans input multiple images. Our triangulation-based self-supervisory signal enables accurate point tracking performance within input window. Hence, we make a point tracking comparison between our network and a state-of-the-art point tracker PIPs++ [61].

Test Data. We choose 14 videos from Aria Pilot (AP) [31] as the testing data. These videos, captured in real-world indoor settings, feature significant camera motion and have pseudo ground truth camera parameters provided. To ensure a more accurate evaluation, dynamic segments within these videos have been excluded.

Metric. With full camera parameters available, we can assess point tracking accuracy by triangulating 3D points from tracklets and measuring their reprojection error, which is quantified in pixels and reported as root mean square error (RMSE). Given that metrics such as Trajectory Error and Average Jaccard, common in 2D point tracking literature [7, 13, 20, 61], offer limited understanding in 3D tasks, we find RMSE to be a more suitable metric for evaluating point tracking performance in 3D contexts.

Setup. Both PIPs++ and our network are configured to track query points for 9 frames. No point budget is applied in this experiment. We consider three types of query points for PIPs++: ALIKED [60], SuperPoint [6], and points from high-gradient image area. Among these, query points from high-gradient area present the most significant challenge, as keypoint identification methods often act as an extra layer of saliency check. Our network processes seed tracklets that are tracked using IMU-assisted KLT or PIPs++.

Note that raw KLT reprojection error is very low due to self-consistency, since both the KLT and pseudo ground truth camera parameters are from the same SLAM system [10]. To examine the network performance on KLT seed tracklets, we furthermore added i.i.d. noise to each view of KLT seed before sending them to our network.

Results. We report the re-projection RMSE for AP [31] testing holdout videos in Table 1. Our findings are three fold: 1) PIPs++ demonstrates limited effectiveness in egocentric videos, especially when significant camera motion present, indicating the open challenge in point tracker research; 2) Our method predict more accurate tracklets with seed tracklets initiated from all types of query points, achieving sub-pixel accuracy when seed tracklets are less noisy; 3) The network does require relatively good seed tracklet to get significant tracking accuracy improvement, suggesting a convergence basin of semi-local nature.

5. Conclusion

In conclusion, this paper presents a novel method for detecting abstract interest points (AIPs) on general scenes. The self-supervised training approach that only focus on geometric consistency successfully circumvents the limitations of traditional local keypoint methods that rely heavily on low-level image features. The discovered AIPs demonstrate remarkable repeatability across varying viewpoints and conditions, maintaining generalizability across diverse scenarios.

Acknowledgment. The authors appreciate insightful discussions with Nan Yang, Jing Dong, Henry Howard-Jenkins, Vasileios Balntas and Daniel Detone.

n

A. Additional Implementation Details

We provide further implementation details of our method. Appendix A.1 details the structure of our network. Appendix A.2 discusses the specifics of our training process, and Appendix A.3 elaborates on the inference details. We will release our code in the future.

A.1. Network Structure

Our network g consists of two primary components, g_{enc} for image encoding, and g_{offset} for offset regression. To enhance efficiency and ensure faster runtime, we build a compact network with a total of 15.2M parameters.

Image Encoder g_{enc} utilizes a U-Net structure with a VGG-11 backbone to extract two feature maps for input images in resolution $H \times W$. Specifically, feature maps at the last two U-Net layers, $\mathbf{F}_{\text{layer}=-1} \in \mathbb{R}^{H \times W \times 32}$ and $\mathbf{F}_{\text{layer}=-2} \in \mathbb{R}^{H/2 \times W/2 \times 64}$, are bi-linearly interpolated for per-point feature vector $\mathbf{f}$. In addition, two Fourier positional embeddings, a frame order embedding with 16 dimensions and a relative pixel coordinate embedding with 128 dimensions, are appended to $\mathbf{f}$, leading to a final per-point feature vector $\mathbf{f}$ with dimension 240.

Offset Regressor g_{offset} includes two parts, a self-attention module, which reasons relationships among features $\mathbf{f}_{\mathbf{p}_i}$ from input views, and a MLP-based regression head, which makes the final offset predictions. The self-attention module is implemented with xFormer [23], and is stacked with 6 encoder layers. Each layer has 8 attention heads and 480 embedding channels. A 4-layer MLP then projects the per-point features transformed by the attention module to 2 channels in a normalized range $(-1, +1)$, which is further converted to pixel space with a scale factor 12. This leads to a max offset range of ± 12 pixel around 2D points on a seed tracklet in one prediction step. With multiple iterative steps, 3 in our experiments, our method is able to discover AIPs that are significantly distant from seed tracklets.

A.2. Training

Data. We curated a comprehensive dataset for training, integrating both internal data and a segment of the publicly available Aria Pilot dataset (AP) [31]. This combination contains 27 hours of egocentric video, including a diverse range of both indoor and outdoor environments. These videos are recorded using Aria devices[2], which are equipped with high accuracy Inertial Measurement Units (IMUs) and two global shutter cameras with fisheye lenses. For generalization purposes, we rectify fisheye videos with a pin-hole camera model, and crop the rectified videos to 640×480 resolution. To run the model on video collected using different lenses, we can apply a similar pin-hole rectification and crop it to the same 640×480 resolution.

Seed Tracklets Generation. The training data was processed with a visual-inertial SLAM algorithm[3] to obtain accurate camera parameters including both poses and intrinsics. The estimated camera parameters further serve as a prior to guide the tracking of seed tracklets[4]. The seed tracklets are first initialized at image regions with high-gradient, e.g., by checking Sobel filter response. Then a motion-assisted KLT tracks these initial points across frames. Comparing to visual-only KLT, motion-assisted KLT has prior knowledge of relative camera motion between frames, therefore is able to 1) conduct a 1D search along epipolar line/curve rather than a 2D area, and 2) terminate a tracklet once the geometric consistency is violated. As a result, seed tracklets from motion-assisted KLT capture scene structures accurately in a semi-dense fashion, which is crucial to the proposed geometric-based self-supervised training.

Seed Tracklets Sampling. In Sec. 3.1 we present our self-supervised training with a pair of tracklets. During training, we adopt a sliding window strategy to enable batched processing. In specific, each training step starts by sampling a pair of sliding windows, each consisting of N_{img} frames that partially cover the same 3D space. We then take all seed tracklets that have a minimum observation threshold m out of window size N_{img} from each window. Next, the tracklets from both windows are paired by checking the 3D distance between 3D points of corresponding tracklets, and only those within the 3D contrastive radius $r_e = 0.3m$ are kept as a part of the training batch. We found $m = 5$ and $N_{\text{img}} = 9$ providing sufficient tracklets pairs that can be triangulated reliably.

Training Schedule. Our model was trained with 192 V100 GPUs for 5 days (0.5 day for bootstrap and 4.5 days for actual training) on the 27 hours of processed video. To speed up convergence, a 15-epoch bootstrap stage is conducted first, where the pulling component in Eq. (8) is applied to seed tracklet pairs that are sampled from predefined 3D tracklet clusters. The 3D clusters were obtained using off-the-shelf 3D clustering algorithm HDBSCAN[5]. This bootstrap stage ensures the pulling behavior between two tracklets emerges early. However, the cluster-fixed training has clear limitation as well, due to the strong bias introduced by the initial cluster assignment for seed tracklets. Therefore, after 15 epochs of bootstrapping, we switch to cluster-free self-supervision, where seed tracklet pairs are sampled based on the 3D contrastive radius r_e. The cluster-free self-supervision either enhances or corrects bootstrapped predictions. In total, our model is trained for 500 epochs.

Differentiable Triangulation. Our differentiable triangulation is implemented with Theseus [35], which offers effi-

[2] Project Aria.
[3] Machine Perception Services.
[4] Semi-dense Motion-assisted KLT Tracker.
[5] HDBSCAN.

cient back-propagation for non-linear least squares (NLS) operations using implicit theorem and sparse operations. We find a 5-iteration NLS operation is sufficient to triangulate 3D points from tracklets, while avoiding introducing too much overhead due to the iterative nature of NLS. To penalize predictions that violate geometric consistency, we employ L_2 reprojection error in the triangulation objective without robust kernels. To stabilize the training process, tracklet predictions with an average reprojection error higher than 6 pixels are discarded from back propagation.

A.3. Inference

Inference Time Input. During the inference phase, our model processes the input video with a sliding window approach. Our method discovers AIPs by taking input of a window of consecutive images, along with a set of seed tracklets associated with this window. We apply the same window size $N_{\text{img}} = 9$ (as training) during inference.

Non-Maximum Suppression We adopt a common non-maximum suppression (NMS) module[6] that takes input points with associated scores. For each AIP, we derive a score from the local density of all AIPs. Specifically, an AIP is scored by counting the number of nearby AIPs at the last image of current sliding window, where the nearby range for counting is identical to NMS range. We set it to 5-pixel for all experiments. Those that form condensed clusters with high consensus will survive from NMS, resulting in a sparse set of highly scored AIPs. Meanwhile those that latch to arbitrary places with low consensus will be scored lower, therefore either be suppressed during NMS or get ranked too low to pass top-K budgeting.

[6] NMS implementation.

References

[1] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *CVPR*, 2017. 6

[2] Axel Barroso-Laguna, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Key. net: Keypoint detection by handcrafted and learned cnn filters. In *ICCV*, 2019. 2

[3] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 3d hand shape and pose from images in the wild. In *CVPR*, 2019. 2

[4] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *TPAMI*, 2018. 2

[5] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019. 2

[6] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, 2018. 2, 6, 7, 8

[7] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. *arXiv preprint arXiv:2306.08637*, 2023. 3, 8

[8] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 3

[9] 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 *CVPR*, 2019. 2

[10] Jakob Engel, Kiran Somasundaram, Michael Goesele, Albert Sun, Alexander Gamino, Andrew Turner, Arjang Talattof, Arnie Yuan, Bilal Souti, Brighid Meredith, Cheng Peng, Chris Sweeney, Cole Wilson, Dan Barnes, Daniel DeTone, David Caruso, Derek Valleroy, Dinesh Ginjupalli, Duncan Frost, Edward Miller, Elias Mueggler, Evgeniy Oleinik, Fan Zhang, Guruprasad Somasundaram, Gustavo Solaira, Harry Lanaras, Henry Howard-Jenkins, Huixuan Tang, Hyo Jin Kim, Jaime Rivera, Ji Luo, Jing Dong, Julian Straub, Kevin Bailey, Kevin Eckenhoff, Lingni Ma, Luis Pesqueira, Mark Schwesinger, Maurizio Monge, Nan Yang, Nick Charron, Nikhil Raina, Omkar Parkhi, Peter Borschowa, Pierre Moulon, Prince Gupta, Raul Mur-Artal, Robbie Pennington, Sachin Kulkarni, Sagar Miglani, Santosh Gondi, Saransh Solanki, Sean Diener, Shangyi Cheng, Simon Green, Steve Saarinen, Suvam Patra, Tassos Mourikis, Thomas Whelan, Tripti Singh, Vasileios Balntas, Vijay Baiyya, Wilson Dreewes, Xiaqing Pan, Yang Lou, Yipu Zhao, Yusuf Mansour, Yuyang Zou, Zhaoyang Lv, Zijian Wang, Mingfei Yan, Carl Ren, Renzo De Nardi, and Richard Newcombe. Project aria: A new tool for egocentric multi-modal ai research. *arXiv preprint arXiv:2308.13561*, 2023. 5, 8

[11] Hao-Shu Fang, Jiefeng Li, Hongyang Tang, Chao Xu, Haoyi Zhu, Yuliang Xiu, Yong-Lu Li, and Cewu Lu. Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time. *TPAMI*, 2022. 2

[12] Pierre Gleize, Weiyao Wang, and Matt Feiszli. Silk–simple learned keypoints. In *ICCV*, 2023. 2, 7

[13] Adam W Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *ECCV*, 2022. 3, 5, 8

[14] Chris Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey Vision Conference*, 1988. 2

[15] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 1981. 3

[16] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017. 3

[17] Allan Jabri, Andrew Owens, and Alexei Efros. Space-time correspondence as a contrastive random walk. In *NeurIPS*, 2020. 3

[18] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea

Vedaldi. Unsupervised learning of object landmarks through conditional image generation. In *NeurIPS*, 2018. 2, 3

[19] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Self-supervised learning of interpretable keypoints from unlabelled videos. In *CVPR*, 2020. 3

[20] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. *arXiv preprint arXiv:2307.07635*, 2023. 3, 8

[21] Zihang Lai and Weidi Xie. Self-supervised learning for video correspondence flow. In *BMVC*, 2019. 3

[22] Zihang Lai, Erika Lu, and Weidi Xie. Mast: A memory-augmented self-supervised tracker. In *CVPR*, 2020. 3

[23] Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, and Daniel Haziza. xformers: A modular and hackable transformer modelling library. https://github.com/facebookresearch/xformers, 2022. 9

[24] Karel Lenc and Andrea Vedaldi. Learning covariant feature detectors. In *ECCVW*, 2016. 2

[25] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. Brisk: Binary robust invariant scalable keypoints. In *ICCV*, 2011. 2

[26] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 6

[27] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *ICCV*, 2023. 3

[28] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *TPAMI*, 2010. 3, 6

[29] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 2, 7

[30] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, 1981. 1, 3

[31] Zhaoyang Lv, Edward Miller, Jeff Meissner, Luis Pesqueira, Chris Sweeney, Jing Dong, Lingni Ma, Pratik Patel, Pierre Moulon, Kiran Somasundaram, Omkar Parkhi, Yuyang Zou, Nikhil Raina, Steve Saarinen, Yusuf M Mansour, Po-Kang Huang, Zijian Wang, Anton Troynikov, Raul Mur Artal, Daniel DeTone, Daniel Barnes, Elizabeth Argall, Andrey Lobanovskiy, David Jaeyun Kim, Philippe Bouttefroy, Julian Straub, Jakob Julian Engel, Prince Gupta, Mingfei Yan, Renzo De Nardi, and Richard Newcombe. Aria pilot dataset. https://about.facebook.com/realitylabs/projectaria/datasets, 2022. 5, 8, 9

[32] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *IJCV*, 2004. 7

[33] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. Lf-net: Learning local features from images. In *NeurIPS*, 2018. 2

[34] Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar Parkhi, Richard Newcombe, et al. Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. *arXiv preprint arXiv:2306.06362*, 2023. 6

[35] Luis Pineda, Taosha Fan, Maurizio Monge, Shobha Venkataraman, Paloma Sodhi, Ricky TQ Chen, Joseph Ortiz, Daniel DeTone, Austin Wang, Stuart Anderson, et al. Theseus: A library for differentiable nonlinear optimization. In *NeurIPS*, 2022. 2, 9

[36] Rajeev Ranjan, Vishal M Patel, and Rama Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *TPAMI*, 2017. 2

[37] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. In *NeurIPS*, 2019. 2, 7

[38] Peter Sand and Seth Teller. Particle video: Long-range motion estimation using point trajectories. *IJCV*, 2008. 3

[39] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 3

[40] Paul-Edouard Sarlin, Mihai Dusmanu, Johannes L. Schönberger, Pablo Speciale, Lukas Gruber, Viktor Larsson, Ondrej Miksik, and Marc Pollefeys. Lamar: Benchmarking localization and mapping for augmented reality. In *ECCV*, 2022. 6

[41] Nikolay Savinov, Akihito Seki, Lubor Ladicky, Torsten Sattler, and Marc Pollefeys. Quad-networks: unsupervised learning to rank for interest point detection. In *CVPR*, 2017. 2

[42] Thomas Schops, Johannes L. Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, 2017. 6

[43] Thomas Schops, Torsten Sattler, and Marc Pollefeys. Bad slam: Bundle adjusted direct rgb-d slam. In *CVPR*, 2019. 6

[44] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 3

[45] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, 2021. 3

[46] Jennifer J Sun, Lili Karashchuk, Amil Dravid, Serim Ryou, Sonia Fereidooni, John C Tuthill, Aggelos Katsaggelos, Bingni W Brunton, Georgia Gkioxari, Ann Kennedy, et al. Bkind-3d: Self-supervised 3d keypoint discovery from multi-view videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9001–9010, 2023. 3

[47] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 1, 3

[48] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks by factorized spatial embeddings. In *ICCV*, 2017. 2

[49] James Thewlis, Samuel Albanie, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of landmarks by descriptor vector exchange. In *ICCV*, 2019. 2, 3

[50] Miroslav Trajković and Mark Hedley. Fast corner detection. *Image and Vision Computing*, 1998. 2

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 5

[52] Yannick Verdie, Kwang Yi, Pascal Fua, and Vincent Lepetit. Tilde: A temporally invariant learned detector. In *CVPR*, 2015. 2

[53] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019. 3

[54] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *ECCV*, 2016. 2

[55] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*, 2020. 2

[56] Linguang Zhang and Szymon Rusinkiewicz. Learning to detect features in texture images. In *CVPR*, 2018. 2

[57] Xu Zhang, Felix X Yu, Svebor Karaman, and Shih-Fu Chang. Learning discriminative and transformation covariant local feature detectors. In *CVPR*, 2017. 2

[58] Yuting Zhang, Yijie Guo, Yixin Jin, Yijun Luo, Zhiyuan He, and Honglak Lee. Unsupervised discovery of object landmarks as structural representations. In *CVPR*, 2018. 2, 3

[59] Xiaoming Zhao, Xingming Wu, Jinyu Miao, Weihai Chen, Peter C. Y. Chen, and Zhengguo Li. Alike: Accurate and lightweight keypoint detection and descriptor extraction. *IEEE Transactions on Multimedia*, 2021. 2

[60] 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*, 2023. 2, 6, 7, 8

[61] Yang Zheng, Adam W Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *ICCV*, 2023. 1, 3, 8

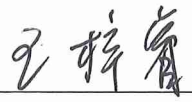
Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	AIP: Abstract Interest Point Discovery from Multi-view Self-supervision
Publication Status	□Published □ Accepted for Publication ☑Submitted for Publication □ Unpublished and unsubmitted work written in a manuscript style
Publication Details	**Zirui Wang***, Yipu Zhao*, Tanner Schmidt, Tomasz Malisiewicz, Jakob Engel. Tech Report, 2023. *Equal contribution.

Student Confirmation

Student Name:	Zirui Wang		
Contribution to the Paper	Co-first-author contribution: • Conception of research ideas. • Design and implementation of models. • Running of large-scale experiments. • Writing of the paper.		
Signature		Date	22/Nov/2023

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	Professor Victor Adrian Prisacariu		
Supervisor comments:	The description is accurate.		
Signature		Date	22/Nov/2023

This completed form should be included in the thesis, at the end of the relevant chapter.

Chapter 6

Conclusion

This thesis explored various avenues in both 3D and 2D registration. Over the course of this DPhil study, there has been a notable shift in the research landscape, transitioning from classical representations like point clouds and level sets, to neural representations such as occupancy networks and Neural Radiance Fields (NeRF).

Initially, our work focused on developing techniques for point cloud registration in both rigid and non-rigid contexts, as discussed in Sections 3.1 and 3.2. We demonstrated advancements in 3D rigid and non-rigid registration through point cloud scene flow estimation and point cloud normal estimation methods.

Subsequently, we turned our attention to 3D – 2D registration challenges, particularly in Sections 4.1 and 4.2. Here, we introduced a joint optimisation pipeline for Neural Radiance Fields and camera parameters. Our work began with NeRF−− for forward-facing scenes and was extended in NoPe-NeRF to accommodate larger camera movements and additional mono-depth estimations.

Lastly, we revisited a low-level task of interest point detection in Chapter 5. Aimed at addressing long-term localisation and mapping, we designed a multi-view, self-supervised interest point discovery pipeline that identifies highly repeat-

able and multi-view consistent interest points.

Overall, this thesis contributes to the evolving field of image and shape registration, offering new methods and insights across a range of scenarios and representations.

6.1 Achievement

We are grateful to see that our research contributions have received positive recognition within their relevant research communities.

In Section 3.1, we enhanced a scene flow estimation network and introduced the concept of using scene flow estimation as an initial step for complex non-rigid registration, improving dynamic reconstruction outcomes. Published in 2020, this work has earned over 90 citations and has influenced subsequent studies in dynamic reconstruction [171] and scene flow estimation networks [172, 173].

Our research, presented in Section 3.2, was among the first to introduce the concept of point soft-weighting in normal estimation tasks. This work was honoured with an oral presentation at BMVC 2020 and has spawned follow-up research such as [174, 175, 176].

The work in Section 4.1 was the first effort in the area of joint NeRF and camera parameter optimisation. Our subsequent work in Section 4.2 received a highlight presentation at CVPR 2023 and expands upon Section 4.1 by introducing a novel way of integrating monocular depth, addressing challenges from larger camera movements. This line of research has inspired various directions such as joint camera and NeRF optimisation [177, 107, 141, 145, 178, 108], lens modelling [179, 105, 180], as well as became a default camera refinement module in dynamic NeRF [140, 181] and city-scale NeRF [182]. Moreover, the GitHub repos-

itories[1][2] for Sections 4.1 and 4.2 have collectively received over 750 stars and have been cited more than 280 times within two years of release, establishing a solid reference in this evolving field.

Lastly, Chapter 5 outlined our ongoing work in the field of interest point discovery, where we propose a novel multi-view setup to address this challenge. This work is currently under peer review. We expect that this work will be particularly valuable for long-term mapping and localisation research.

6.2 Future Work

Current registration methods predominantly depend on geometry and appearance correspondences, which are vulnerable to challenges such as motion blur, sensor noise, and varying lighting conditions. A promising direction for future research is to incorporate high-level semantic from recent advancements in representation learning, including MAE [183], DINO [184, 185], and large vision and language models [186, 187]. Merging semantic information with low-level appearance details could result in more robust registration performance in challenging conditions.

Additionally, in this thesis, we demonstrated that integrating classical multi-view geometry principles can significantly improve learning-based registration algorithms. This integration leads to benefits such as accelerated training processes, enhanced accuracy, and improved consistency across multiple views. Hence, similar idea can be employed to augment large-scale vision and language models such as CLIP [186], StableDiffusion [187], and Zero-1-to-3 [112], by serving as an extra self-supervision mechanism that encourages more consistent results in multi-view setups.

[1] NeRF−−: https://github.com/ActiveVisionLab/nerfmm.
[2] NoPe-NeRF: https://github.com/ActiveVisionLab/nope-nerf/.

Bibliography

[1] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *TPAMI*, 2007.

[2] Georg Klein and David Murray. Parallel tracking and mapping on a camera phone. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2009.

[3] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011.

[4] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *ICRA*, 2014.

[5] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *ECCV*, 2014.

[6] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 2015.

[7] Victor Adrian Prisacariu, Olaf Kähler, Stuart Golodetz, Michael Sapienza, Tommaso Cavallari, Philip HS Torr, and David W Murray. Infinitam v3: A framework for large-scale 3d reconstruction with loop closure. *arXiv preprint arXiv:1708.00783*, 2017.

[8] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M Seitz, and Richard Szeliski. Building rome in a day. In *ICCV*, 2009.

[9] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

[10] Changchang Wu. Towards linear-time incremental structure from motion. In *3DV*, 2013.

[11] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *TPAMI*, 2009.

[12] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016.

[13] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *ECCV*, 2018.

[14] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[15] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (TOG)*, 2022.

[16] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 2023.

[17] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*, 2021.

[18] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, et al. Fusion4d: Real-time performance capture of challenging scenes. *ACM Transactions on Graphics (TOG)*, 2016.

[19] Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *CVPR*, 2015.

[20] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *ICCV*, 2019.

[21] Christoph Bregler, Aaron Hertzmann, and Henning Biermann. Recovering non-rigid 3d shape from image streams. In *CVPR*, 2000.

[22] Yuchao Dai, Hongdong Li, and Mingyi He. A simple prior-free method for non-rigid structure-from-motion factorization. *IJCV*, 2014.

[23] Chen Kong and Simon Lucey. Deep non-rigid structure from motion. In *ICCV*, 2019.

[24] Paul J Besl and Neil D McKay. A method for registration of 3-d shapes. *TPAMI*, 1992.

[25] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 1992.

[26] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *Robotics: science and systems (RSS)*, 2009.

[27] Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Colored point cloud registration revisited. In *CVPR*, 2017.

[28] Miroslava Slavcheva, Wadim Kehl, Nassir Navab, and Slobodan Ilic. Sdf-2-sdf: Highly accurate 3d object reconstruction. In *ECCV*, 2016.

[29] Miroslava Slavcheva, Maximilian Baust, Daniel Cremers, and Slobodan Ilic. Killingfusion: Non-rigid 3d reconstruction without correspondences. In *CVPR*, 2017.

[30] Miroslava Slavcheva, Maximilian Baust, and Slobodan Ilic. Sobolevfusion: 3d reconstruction of scenes undergoing free non-rigid motion. In *CVPR*, 2018.

[31] Long Quan and Zhongdan Lan. Linear n-point camera pose determination. *TPAMI*, 1999.

[32] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *TPAMI*, 2003.

[33] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem. *IJCV*, 2009.

[34] KGA Gilhuijs, PJH Van De Ven, and M Van Herk. Automatic three-dimensional inspection of patient setup in radiation therapy using portal images, simulator images, and computed tomography data. *Medical Physics*, 1996.

[35] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *ICCV*, 2011.

[36] Olivier D Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *ECCV*, 1992.

[37] Richard I Hartley. Estimation of relative camera positions for uncalibrated cameras. In *ECCV*, 1992.

[38] Quan-Tuan Luong and Olivier D Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. *IJCV*, 1996.

[39] Philip HS Torr and David William Murray. The development and comparison of robust methods for estimating the fundamental matrix. *IJCV*, 1997.

[40] René Ranftl and Vladlen Koltun. Deep fundamental matrix estimation. In *ECCV*, 2018.

[41] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial Intelligence*, 1981.

[42] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, 1981.

[43] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. Flownet3d: Learning scene flow in 3d point clouds. In *CVPR*, 2019.

[44] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. Pointnetlk: Robust & efficient point cloud registration using pointnet. In *CVPR*, 2019.

[45] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *ICCV*, 2019.

[46] Weixin Lu, Guowei Wan, Yao Zhou, Xiangyu Fu, Pengfei Yuan, and Shiyu Song. Deepvcp: An end-to-end deep neural network for point cloud registration. In *CVPR*, 2019.

[47] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *ICCV*, 2015.

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[49] Brian Amberg, Sami Romdhani, and Thomas Vetter. Optimal step nonrigid icp algorithms for surface registration. In *CVPR*, 2007.

[50] Hao Li, Robert W Sumner, and Mark Pauly. Global correspondence optimization for non-rigid registration of depth scans. In *Computer Graphics Forum*, 2008.

[51] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics*, 1992.

[52] Yizhak Ben-Shabat, Michael Lindenbaum, and Anath Fischer. Nesti-net: Normal estimation for unstructured 3d point clouds using convolutional neural networks. In *CVPR*, 2019.

[53] Paul Guerrero, Yanir Kleiman, Maks Ovsjanikov, and Niloy J Mitra. Pcpnet: learning local shape properties from raw point clouds. In *Computer Graphics Forum*, 2018.

[54] Chris Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey Vision Conference*, 1988.

[55] Miroslav Trajković and Mark Hedley. Fast corner detection. *Image and Vision Computing*, 1998.

[56] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[57] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, 2018.

[58] Karel Lenc and Andrea Vedaldi. Learning covariant feature detectors. In *ECCVW*, 2016.

[59] Nikolay Savinov, Akihito Seki, Lubor Ladicky, Torsten Sattler, and Marc Pollefeys. Quad-networks: unsupervised learning to rank for interest point detection. In *CVPR*, 2017.

[60] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. In *NeurIPS*, 2019.

[61] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks by factorized spatial embeddings. In *ICCV*, 2017.

[62] James Thewlis, Samuel Albanie, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of landmarks by descriptor vector exchange. In *ICCV*, 2019.

[63] Yuting Zhang, Yijie Guo, Yixin Jin, Yijun Luo, Zhiyuan He, and Honglak Lee. Unsupervised discovery of object landmarks as structural representations. In *CVPR*, 2018.

[64] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks through conditional image generation. In *NeurIPS*, 2018.

[65] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 2019.

[66] Ivan E Sutherland. Sketchpad: A man-machine graphical communication system. In *Proceedings of Spring Joint Computer Conference*, 1963.

[67] Herbert Voelcker and Aristides Requicha. Constructive solid geometry. *Technical Report*, 1977.

[68] AS Besicovitch. Parametric surfaces. *Bulletin of the American Mathematical Society*, 1950.

[69] Marc Levoy and Turner Whitted. The use of points as a display primitive. *Technical Report*, 1985.

[70] Bruce Guenther Baumgart. *Geometric modeling for computer vision.* Stanford University, 1974.

[71] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 1988.

[72] Hanspeter Pfister, Matthias Zwicker, Jeroen Van Baar, and Markus Gross. Surfels: Surface elements as rendering primitives. In *Proceedings of the conference on Computer Graphics and Interactive Techniques*, 2000.

[73] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM Transactions on Graphics (TOG)*, 2018.

[74] Natasha Gelfand, Niloy J Mitra, Leonidas J Guibas, and Helmut Pottmann. Robust global registration. In *Symposium on Geometry Processing*, 2005.

[75] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *ICRA*, 2009.

[76] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *ECCV*, 2016.

[77] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017.

[78] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017.

[79] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 2019.

[80] Silvio Giancola, Jesus Zarzar, and Bernard Ghanem. Leveraging shape completion for 3d siamese tracking. In *CVPR*, 2019.

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

[82] Zirui Wang, Shuda Li, Henry Howard-Jenkins, Victor Prisacariu, and Min Chen. Flownet3d++: Geometric losses for deep scene flow estimation. In *WACV*, 2020.

[83] Himangi Mittal, Brian Okorn, and David Held. Just go with the flow: Self-supervised scene flow estimation. In *CVPR*, 2020.

[84] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *TPAMI*, 2020.

[85] David Adalsteinsson and James A Sethian. A fast level set method for propagating interfaces. *Journal of Computational Physics*, 1995.

[86] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (TOG)*, 2013.

[87] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, pages 165–174, 2019.

[88] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *NeurIPS*, 2021.

[89] Frederic Gibou, Ronald Fedkiw, and Stanley Osher. A review of level-set methods and some recent applications. *Journal of Computational Physics*, 2018.

[90] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *ICCV*, 2021.

[91] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022.

[92] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *ECCV*, 2022.

[93] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *CVPR*, 2023.

[94] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021.

[95] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021.

[96] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *ICCV*, 2021.

[97] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Generalizable patch-based neural rendering. In *ECCV*, 2022.

[98] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *NeurIPS*, 2021.

[99] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *CVPR*, 2021.

[100] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *CVPR*, 2021.

[101] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, 2021.

[102] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *CVPR*, 2023.

[103] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf−−: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021.

[104] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *ICCV*, 2021.

[105] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Anima Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. In *ICCV*, 2021.

[106] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. In *CVPR*, 2023.

[107] Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari. Sparf: Neural radiance fields from sparse and noisy poses. In *CVPR*, 2023.

[108] Zezhou Cheng, Carlos Esteves, Varun Jampani, Abhishek Kar, Subhransu Maji, and Ameesh Makadia. Lu-nerf: Scene and pose estimation by synchronizing local unposed nerfs. In *ICCV*, 2023.

[109] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *ICCV*, 2021.

[110] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *CVPR*, pages 867–876, 2022.

[111] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *ICLR*, 2022.

[112] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *ICCV*, 2023.

[113] Kyle Gao, Yina Gao, Hongjie He, Denning Lu, Linlin Xu, and Jonathan Li. Nerf: Neural radiance field in 3d vision, a comprehensive review. *arXiv preprint arXiv:2210.00379*, 2022.

[114] Takeshi Masuda. Registration and integration of multiple range images by matching signed distance fields for object shape modeling. *Computer Vision and Image Understanding*, 2002.

[115] Peter Claes, Dirk Vandermeulen, Luc Van Gool, and Paul Suetens. Robust and accurate partial surface registration based on variational implicit surfaces for automatic 3d model building. In *International Conference on 3D Digital Imaging and Modeling (3DIM)*, 2005.

[116] Erik Bylow, Jürgen Sturm, Christian Kerl, Fredrik Kahl, and Daniel Cremers. Real-time camera tracking and 3d reconstruction using signed distance functions. In *Robotics: Science and Systems (RSS)*, 2013.

[117] Daniel R Canelhas, Todor Stoyanov, and Achim J Lilienthal. Sdf tracker: A parallel algorithm for on-line pose estimation and scene reconstruction from depth images. In *International Conference on Intelligent Robots and Systems*, 2013.

[118] Yu Chen and Gim Hee Lee. Dreg-nerf: Deep registration for neural radiance fields. *arXiv preprint arXiv:2308.09386*, 2023.

[119] Lily Goli, Daniel Rebain, Sara Sabour, Animesh Garg, and Andrea Tagliasacchi. nerf2nerf: Pairwise registration of neural radiance fields. In *ICRA*, 2023.

[120] HG Barrow, JM Tenenbaum, RC Bolles, and HC Wolf. Parametric correspondence and chamfer matching: two new techniques for image matching. In *IJCAI*, 1977.

[121] Heng Yang, Jingnan Shi, and Luca Carlone. Teaser: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 2020.

[122] Bailin Deng, Yuxin Yao, Roberto M Dyke, and Juyong Zhang. A survey of non-rigid 3d registration. In *Computer Graphics Forum*, 2022.

[123] Sundar Vedula, Simon Baker, Peter Rander, Robert Collins, and Takeo Kanade. Three-dimensional scene flow. In *ICCV*, 1999.

[124] Brett Allen, Brian Curless, and Zoran Popović. The space of human body shapes: reconstruction and parameterization from range scans. *ACM transactions on graphics (TOG)*, 2003.

[125] Robert W Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. *ACM Transactions on Graphics (TOG)*, 2007.

[126] Xiuye Gu, Yijie Wang, Chongruo Wu, Yong Jae Lee, and Panqu Wang. Hplflownet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds. In *CVPR*, 2019.

[127] Wenxuan Wu, Zhi Yuan Wang, Zhuwen Li, Wei Liu, and Li Fuxin. Pointpwc-net: Cost volume on point clouds for (self-) supervised scene flow estimation. In *ECCV*, 2020.

[128] Zachary Teed and Jia Deng. Raft-3d: Scene flow using rigid-motion embeddings. In *CVPR*, 2021.

[129] Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishman, David Levin, and Claudio T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 2003.

[130] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the Eurographics Symposium on Geometry Processing*, 2006.

[131] James T Kajiya. The rendering equation. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques*, 1986.

[132] Frédéric Cazals and Marc Pouget. Estimating differential quantities using polynomial fitting of osculating jets. *Computer Aided Geometric Design*, 2005.

[133] John B Hampshire II and Alex Waibel. The meta-pi network: Building distributed knowledge representations for robust multisource pattern recognition. *TPAMI*, 1992.

[134] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural Computation*, 1991.

[135] Jiaxin Li and Gim Hee Lee. Deepi2p: Image-to-point cloud registration via deep classification. In *CVPR*, 2021.

[136] John R Adler Jr, Martin J Murphy, Steven D Chang, and Steven L Hancock. Image-guided robotic radiosurgery. *Neurosurgery*, 1999.

[137] Roch M Comeau, Abbas F Sadikot, Aaron Fenster, and Terry M Peters. Intraoperative ultrasound for guidance and tissue shift correction in image-guided neurosurgery. *Medical Physics*, 2000.

[138] Primoz Markelj, Dejan Tomaževič, Bostjan Likar, and Franjo Pernuš. A review of 3d/2d registration methods for image-guided interventions. *Medical Image Analysis*, 2012.

[139] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *TPAMI*, 2017.

[140] Yu-Lun Liu, Chen Gao, Andreas Meuleman, Hung-Yu Tseng, Ayush Saraf, Changil Kim, Yung-Yu Chuang, Johannes Kopf, and Jia-Bin Huang. Robust dynamic radiance fields. In *CVPR*, 2023.

[141] Samarth Sinha, Jason Y Zhang, Andrea Tagliasacchi, Igor Gilitschenski, and David B Lindell. Sparsepose: Sparse-view camera pose regression and refinement. In *CVPR*, 2023.

[142] Peng Wang, Hao Tan, Sai Bi, Yinghao Xu, Fujun Luan, Kalyan Sunkavalli, Wenping Wang, Zexiang Xu, and Kai Zhang. Pf-lrm: Pose-free large reconstruction model for joint pose and shape prediction. *arXiv preprint arXiv:2311.12024*, 2023.

[143] Hanwen Jiang, Zhenyu Jiang, Kristen Grauman, and Yuke Zhu. Few-view object reconstruction with unknown categories and camera poses. *arXiv preprint arXiv:2212.04492*, 2022.

[144] Zhenpei Yang, Zhile Ren, Miguel Angel Bautista, Zaiwei Zhang, Qi Shan, and Qixing Huang. Fvor: Robust joint shape and pose optimization for few-view object reconstruction. In *CVPR*, 2022.

[145] Cameron Smith, Yilun Du, Ayush Tewari, and Vincent Sitzmann. Flowcam: Training generalizable 3d radiance fields without camera poses via pixel-aligned scene flow. In *NeurIPS*, 2023.

[146] Amy Lin, Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose++: Recovering 6d poses from sparse-view observations. *arXiv preprint arXiv:2305.04926*, 2023.

[147] Hanwen Jiang, Zhenyu Jiang, Yue Zhao, and Qixing Huang. Leap: Liberate sparse-view 3d modeling from camera poses. *arXiv preprint arXiv:2310.01410*, 2023.

[148] Shuai Chen, Zirui Wang, and Victor Prisacariu. Direct-posenet: Absolute pose regression with photometric consistency. In *3DV*. IEEE, 2021.

[149] Shuai Chen, Xinghui Li, Zirui Wang, and Victor A Prisacariu. Dfnet: Enhance absolute pose regression with direct feature matching. In *ECCV*, 2022.

[150] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *CVPR*, 2013.

[151] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In *ECCV*, 2014.

[152] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *CVPR*, 2017.

[153] Meng Xu, Youchen Wang, Bin Xu, Jun Zhang, Jian Ren, Stefan Poslad, and Pengfei Xu. A critical analysis of image-based camera pose estimation techniques. *arXiv preprint arXiv:2201.05816*, 2022.

[154] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015.

[155] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017.

[156] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020.

[157] J-P Thirion. Image matching as a diffusion process: an analogy with maxwell's demons. *Medical Image Analysis*, 1998.

[158] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 1995.

[159] Baba C Vemuri, J Ye, Yunmei Chen, and Christiana Morison Leonard. Image registration via level-set motion: Applications to atlas-based segmentation. *Medical Image Analysis*, 2003.

[160] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *TPAMI*, 2010.

[161] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019.

[162] Zihang Lai and Weidi Xie. Self-supervised learning for video correspondence flow. In *BMVC*, 2019.

[163] Zihang Lai, Erika Lu, and Weidi Xie. Mast: A memory-augmented self-supervised tracker. In *CVPR*, 2020.

[164] Allan Jabri, Andrew Owens, and Alexei Efros. Space-time correspondence as a contrastive random walk. In *NeurIPS*, 2020.

[165] Junhwa Hur and Stefan Roth. Optical flow estimation in the deep learning age. *Modelling Human Motion: From Human Perception to Robot Design*, 2020.

[166] Andréa Macario Barros, Maugan Michel, Yoann Moline, Gwenolé Corre, and Frédérick Carrel. A comprehensive survey of visual slam algorithms. *Robotics*, 2022.

[167] Frédéric Huguet and Frédéric Devernay. A variational method for scene flow estimation from stereo sequences. In *ICCV*, 2007.

[168] Mariano Jaimez, Mohamed Souiai, Javier Gonzalez-Jimenez, and Daniel Cremers. A primal-dual framework for real-time dense rgb-d scene flow. In *ICRA*, 2015.

[169] 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 *CVPR*, 2016.

[170] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020.

[171] Aljaz Bozic, Pablo Palafox, Michael Zollhöfer, Angela Dai, Justus Thies, and Matthias Nießner. Neural non-rigid tracking. In *NeurIPS*, 2020.

[172] Zan Gojcic, Or Litany, Andreas Wieser, Leonidas J Guibas, and Tolga Birdal. Weakly supervised learning of rigid 3d scene flow. In *CVPR*, 2021.

[173] Ivan Tishchenko, Sandro Lombardi, Martin R Oswald, and Marc Pollefeys. Self-supervised learning of non-rigid residual flow and ego-motion. In *3DV*, 2020.

[174] Emon Kumar Dey, Fayez Tarsha Kurdi, Mohammad Awrangjeb, and Bela Stantic. Effective selection of variable point neighbourhood for feature point extraction from aerial building point cloud data. *Remote Sensing*, 2021.

[175] Weijia Wang, Xuequan Lu, Dasith De Silva Edirimuni, Xiao Liu, and Antonio Robles-Kelly. Deep point cloud normal estimation via triplet learning. In *IEEE International Conference on Multimedia and Expo (ICME)*, 2022.

[176] Shujuan Li, Junsheng Zhou, Baorui Ma, Yu-Shen Liu, and Zhizhong Han. Neaf: Learning neural angle fields for point normal estimation. In *AAAI*, 2023.

[177] Jiahui Zhang, Fangneng Zhan, Rongliang Wu, Yingchen Yu, Wenqing Zhang, Bai Song, Xiaoqin Zhang, and Shijian Lu. Vmrf: View matching neural radiance fields. In *ACM International Conference on Multimedia*, 2022.

[178] Keunhong Park, Philipp Henzler, Ben Mildenhall, Jonathan T Barron, and Ricardo Martin-Brualla. Camp: Camera preconditioning for neural radiance fields. In *SIGGRAPH Asia*, 2023.

[179] Wenqi Xian, Aljaž Božič, Noah Snavely, and Christoph Lassner. Neural lens modeling. In *CVPR*, 2023.

[180] Kai Gu, Thomas Maugey, Sebastian Knorr, and Christine Guillemot. Omni-nerf: neural radiance field from 360 image captures. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, 2022.

[181] Mert Asim Karaoglu, Hannah Schieber, Nicolas Schischka, Melih Görgülü, Florian Grötzner, Alexander Ladikos, Daniel Roth, Nassir Navab, and Benjamin Busam. Dynamon: Motion-aware fast and robust camera localization for dynamic nerf. *arXiv preprint arXiv:2309.08927*, 2023.

[182] Jianfei Guo, Nianchen Deng, Xinyang Li, Yeqi Bai, Botian Shi, Chiyu Wang, Chenjing Ding, Dongliang Wang, and Yikang Li. Streetsurf: Extending multi-view implicit surface reconstruction to street views. *arXiv preprint arXiv:2306.04988*, 2023.

[183] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

[184] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

[185] 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. *Transactions on Machine Learning Research (TMLR)*, 2023.

[186] 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 *ICML*, 2021.

[187] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.