Bournemouth University

Real-time Topology-Aware Augmented Reality

QINGHONG GAO

Supervisor: Professor Wen Tang

Department of Science and Technology
Bournemouth University

A thesis submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

March 2024

Abstract

Augmented Reality (AR) technology fuses virtual information with the real-world environment to enhance the way people interact with digital information in their physical world. This thesis is concerned with topology-aware AR systems designed to be aware of the topology changes in the surroundings and explore the topological features of scenes. Topological structures, such as graphs, can provide information on the relationship between point clouds to improve the quality of point cloud-based real-world 3D map reconstructions for topology-aware AR systems. The reconstructed 3D maps provide information to improve the registration accuracy between virtual objects and the physical environment. Furthermore, 3D maps also help to reduce registration failures caused by complex and dynamic scenes, such as object occlusions, object motion, and object deformation.

This thesis explores algorithms, computational methods, and frameworks for dense 3D surface reconstructions based on monocular videos and images for augmented reality applications. The main contributions of this PhD work are: 1) Proposed a graph deep learning-based framework for monocular depth estimation, which learns non-Euclidean features and improves the accuracy of depth estimations. Mathematical background on group equivariance, including translation equivariance and permutation equivariance, is also introduced to provide theoretical support for the proposed network; 2) Conducted two use cases to demonstrate the capabilities of the proposed methods in improving fine details of depth estimation for complex and unstructured environments with free camera motions; 3) A further improved the framework to address low-illumination endoscopy videos; 4) Proposed a statistical method to handle the non-rigid point cloud registration with special topology changes. Within which, a clustering and refinement scheme is proposed to deal with distribution irregularities of point sets; 5) Developed a framework to demonstrate the functionality of the proposed method in AR.

Under challenging scenes such as endoscopy and unmanned aerial vehicle videos, the proposed methods outperform the state-of-the-art algorithms with robustness and accuracy. For example, the proposed depth estimation method improves the 3D data acquisition, the *Break and Splice* framework improves the 3D dynamic reconstruction, and the proposed AR framework provides a solution in dynamic scenes for medical applications.

Table of contents

List of figures

List of tables

Acknowledgments

My passion for this research subject is a testament to the enduring support and guidance I have received throughout my life and this transformative PhD journey.

Foremost, I extend my deepest appreciation to my supervisor, Professor Wen Tang, whose consistent encouragement and wisdom have been my guiding lights. Beyond the academic realm, her genuine concern for my wellbeing has made this scholarly pursuit not only intellectually enriching but also personally fulfilling. I am indebted to Dr. Tao Ruan Wan, whose mentorship during my MSc project laid the foundation for the doctoral research undertaken at Bournemouth University in the UK. Their belief in my potential has been a driving force, without which my achievement would be inconceivable..

During my time at Bournemouth University, I am fortunate to be surrounded by an exceptional community of researchers who have become cherished friends. Drs Yan Zhao and Long Xi stand out for their invaluable assistance in completing my first paper and their continuous insightful discussions throughout my research expedition. I am equally grateful to Drs WeiLai Xu and Ge Zheng for their unwavering care and assistance.

My research secondments on the H2020-iGame project in Budapest and Malaga have introduced me to Mr. Attila Biro, who has greatly enriched my experience and provided insights into Hungary's local culture, and to Professor Fermin Mayoral Clerise and Professor Cuesta Vargas Antonio at the University of Malaga.

Driven by shared goals and fueled by shared experiences, the camaraderie of friends turned family made my stay in England both enjoyable and comforting. My heartfelt thanks extend to Ying Chi, Yili Sun, Lingting Qiao, and Zhiqi Li. Amid the challenges posed by the COVID-19 pandemic, their companionship and shared meals provided solace and a sense of belonging.

The unyielding support of my parents, Mr. Luoyang Gao and Mrs. Lixin Li, has been a constant in my life, fostering humility and courage in the face of adversities. Their resilience and unwavering belief in me have inspired me, both in my native China and on foreign shores. My gratitude extends to my older sister, Dandan Gao, whose devoted care towards our parents allowed me to focus on my academic pursuits with a lighter heart. To my family, I cherish and love you deeply always.

As I conclude this remarkable PhD journey, I stand poised to embrace the challenges that the future may unfold with a newfound excitement and gratitude for the incredible support that has shaped my academic odyssey.

Declaration

I, Qinghong Gao, hereby declare that this thesis represents my own work which has been done after registration for the degree of P.h.D at Bournemouth University and has not been submitted to any other institute for any award or qualifications. I also confirm that this work fully acknowledges the contributions from the work of others. Unless otherwise stated, all photographs were taken by or under the direction of the author.

QINGHONG GAO
March 2024

Nomenclature

Acronyms / Abbreviations

3D Three Dimensions

AR Augmented Reality

AS Angular Similarity

BCPD Bayesian Coherent Point Drift

CLAHE Contrast Limited Adaptive Histogram Equalization

CNNs Convolutional Neural Networks

CPD Coherent Point Drift

CRF Conditional Random Field

CT Computed Tomography

DPGMM Dirichlet Process Gaussian Mixture Model

GAT Graph Attention Network

GCN Graph Convolution Network

GNNs Graph Neural Networks

GPS Global Positioning Systems

GPS Global Positioning Systems

KLT Kanade-Lukas-Tomashi

LiDAR Light Detection and Ranging

MIS Minimally Invasive Surgery

MRF Markov Random Field

MRI Magnetic Resonance Imaging

MR Mixed Reality

PnP Perspective-n-Point

RAMIS Robot-Assisted Minimally Invasive Surgery

RGB-D RGB-Depth

RMSE Root Mean Square Error

SfM Structure from Motion

SfS Shape from Shading

TPS Thin Plate Spline

UAVs Unmanned Aerial Vehicles

VO Visual Odometry

VR Virtual Reality

VSLAM Visual Simultaneous Localization And Mapping

I would like to dedicate this thesis to my parents, Luoyang Gao, and Lixin Li.

Chapter 1

Introduction

1.1 Background

Augmented reality (AR) technology utilises computer-generated virtual information to enhance the real world, thereby improving people's ability to perceive and interact with their physical environment. AR technology is an interdisciplinary field encompassing computer vision, computer graphics, machine learning and deep learning, and human-computer interactions. AR has been used in a variety of industries and domains, such as education [12], entertainment [13], medicine [14], and military [15].

Milgram and Kishino [16] first provided the concept of AR in 1994. They proposed a virtuality continuum diagram to represent AR, Virtual Reality (VR) and Mixed Reality (MR). Compared with VR, which is represented by a fully virtual environment at the rightmost end, AR remains closer to the experiences comprising a real environment and provides information that goes beyond what humans can perceive with their senses alone. Figure 1.1 shows milestones in the history of AR. AR technology includes 3D registration that fuses virtual information, including 3D information or models, with real-world elements such as images and videos [17]. The solutions for 3D registration can be classified into two main categories: image-based methods using features detected from images and 3D map-based methods using point features. The former category achieves the virtual object registration in the real world based on Perspective-n-Point (PnP) [18]. PnP utilises templates, such as makers [19] or reference images [20], to match videos, which is then used to calculate the camera pose for virtual object registration. The registration stability of these methods can cause distracting visual artifacts (i.e. flashing visual effects due to virtual objects being unstable) [21]. 3D map-based methods utilise 3D map information to register virtual objects. The reconstructed 3D map not only provides positional information but also helps overcome the challenge of virtual object registration failure caused by complex and dynamic scenes, such as object occlusions, object motion, and object deformation. [22], [23]. There are two types of 3D maps: sparse and dense maps. Sparse 3D maps are representations of a space with a low point density and are generally used to compute only feature points for depth estimation. Dense 3D maps, on the other hand, have a high point density and usually have the same number of 3D points

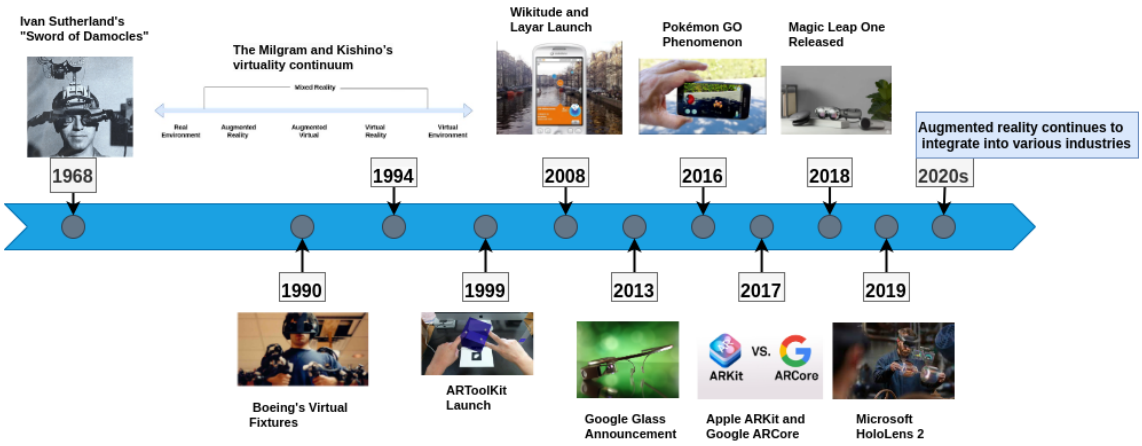


Fig. 1.1 Milestones in the history of AR.

with image pixels. Dense 3D maps can provide detailed and accurate representations of 3D point clouds and improve the registration between virtual objects and the environment for AR applications.

This research primarily focuses on data acquisition and non-rigid registration. Data acquisition is to capture a point cloud of multiple observations or frames of the object or a scene using various sensing technologies such as RGB cameras (monocular and stereo cameras), RGB-D sensors, Light Detection and Ranging (LiDAR), or other 3D scanning techniques. The choice of sensors depends on the specific environment of an application. For example, in Minimally Invasive Surgery (MIS), the visual sensor, called an endoscope, is usually a monocular camera due to its small size and lightweight, making it easy to integrate into the surgical setup [24]. RGB-D sensors, such as the Kinect, are commonly employed in indoor environments [25]. LiDAR sensors have been used in autonomous vehicles [26] or cultural heritage preservation [27] to capture high-precision point clouds. RGB cameras have certain advantages, such as being cost-effective, lightweight and richer information about the environment compared to other sensors. This project investigates data acquisition using monocular RGB cameras for 3D surface reconstruction of generating 3D point clouds or meshes for dynamic scenes or environments that are changing over time. This research work focuses on monocular depth estimation for dense depth maps. On the other hand, non-rigid registration is to find a warp field W to be applied to a source point cloud S such that the warped source point cloud best explains the target point cloud T: $W(S) = T$. This research primarily focuses on non-rigid registration with special topology changes, e.g. connections and separations (see Fig. 7.1 in Chapter 7).

More specifically, traditional monocular depth estimation involves Structure from Motion (SfM)-based methods and handcrafted feature-based methods. SfM-based methods [28], [29], [30] aim to find a set of corresponding pixels on a series of images of a scene to compute depth values. Obtained depth maps by these methods are usually sparse. On the other hand, handcrafted feature-based methods [31], [32] utilise the features extracted from images to estimate dense depth by optimising a probabilistic model such as a Markov Random Field (MRF) [33] or a Conditional Random Field (CRF) [34]. More

recently, deep-learning-based methods [35], [36], [37] have made significant progress in recovering the depth of information from a single image. However, many learning-based methods fail to learn fine details. This is because point clouds as depth maps are out-of-order and include many non-Euclidean features, such as curvature, geodesic distance, and hyperbolic angle [38]. Furthermore, monocular depth estimation for endoscopy scenes remains a significant challenge because of the complex surgical environment coupled with textureless surface features and low-illumination conditions. Another challenging case is that images captured by Unmanned Aerial Vehicles (UAVs) in unstructured environments lack well-defined or predictable structures, which may have some degrees of complexity, randomness, or variability.

Depending on applications, point clouds can be aligned by rigid methods [39] or non-rigid methods [40]. Rigid methods are suitable for the registration of static scenes or objects, which can be aligned with a single translation and rotation. Non-rigid methods often involve a series of transformation matrices and different parts of the object that undergo various deformations. The computation of deformations is a fundamental problem in the acquisition and analysis of non-rigidly deformable objects [41]. Non-rigid registration methods based on point correspondence are to estimate affine transformations between the source and the target point sets [42], [43]. Statistical methods, such as the motion coherence theory, have been proposed to estimate maximum-likelihood solutions for non-rigid registration [44], [44]. However, non-rigid registration methods have failed to effectively address the topology changes of objects, such as connection and separation issues between objects. Topology changes in a scene can lead to misregistration and inaccurate reconstructions. In addition, large inter-frame motions can also cause significant deformations and changes in topology, which poses a challenge for non-rigid registration.

1.2 Motivation

This project aims to address two challenges in 3D surface dynamic reconstruction in order to improve the registration accuracy between virtual objects and their physical environments and to improve the perception of depth in challenge applications with a monocular camera with free motions, such as Minimally Invasive Surgery(MIS) and Unmanned Aerial Vehicles(UAVs). More specifically, this work will firstly address challenges in depth estimation to learn complex topology information contained in scenes captured by monocular devices. Secondly, this research will investigate non-rigid registration by dealing with special topology changes, which is an essential part of traditional 3D dynamic reconstruction.

1.3 Hypothesis and Research Questions

This thesis hypothesizes that an accurate surface 3D reconstruction method, algorithm, or framework can improve AR applications. The second hypothesis is that exploring

topological structures can improve surface 3D reconstruction. Research questions for this thesis are:

I. What factors influence the self-supervised monocular depth estimation accuracy in state-of-the-art algorithms, and how to achieve more accurate depth estimations for 3D surfaces?

II. What factors affect the performance of depth estimation in low-illumination environments?

III. How to solve non-rigid registration with special topology changes, like connections and separation?

IV. How to apply depth estimation and dynamic 3D surface reconstruction algorithms to AR systems?

1.4 Research Contributions

This PhD research develops a self-supervised graph deep learning-based framework for monocular depth estimation. Two use cases are presented: one is for minimally invasive surgery using endoscopy videos; another is the depth estimation from videos captured by Unmanned Aerial Vehicles (UAVs). A statistical method for non-rigid point cloud registration is proposed to solve topological changes. AR applications are finally developed based on the proposed monocular depth estimation framework to achieve accurate surgery instrument tracking and deal with missing depth caused by organs underneath the surgery instruments.

The main contributions of this work are:

- **The development of a new self-supervised graph deep learning-based framework for monocular depth estimation.**(*Chapter 3, 4*)

  A novel coarse-fine encoder framework is proposed that utilises a graph attention network (GAT) [45] to learn non-Euclidean features and refine the depth geometry feature. The graph structure that explores four-connectivity (two elements are considered connected if they share a common edge, specifically the top, bottom, left, or right edge) can keep the original neighbour point information so that avoiding the loss of local feature information of the point cloud. In this thesis, mathematical foundations are provided to support the validity of the coarse-to-fine encoder model.

- **Two use case studies of unstructured scenes are conducted to evaluate the performance and efficacy of the proposed method.**(*Chapter 5, 6*)

  The use of case studies demonstrates the depth estimation effectiveness of the proposed method for unstructured scenes. In the endoscopy case study, three endoscopy datasets are used for quantitative evaluations with five state-of-the-art methods. In addition, the qualitative results show that the proposed monocular depth estimation

framework can correctly recover the depth of surgical instruments and have well-distributed point clouds along the edges. In the UAVs case study, three different weather datasets are used to compare with state-of-the-art methods for metrics evaluations, and results of two real-world datasets, urban settings and natural wilderness, show that the proposed framework can achieve fine details of depth compared with other methods.

- **Improving the depth of low-illumination endoscopy video.**(*Chapter 5*)

  To address the issue of low-illumination environments in specific endoscopy scenarios, the Contrast Limited Adaptive Histogram Equalization (CLAHE) method [46] is employed to ensure brightness consistency and enhance the details of endoscopic images. This application of CLAHE improves depth estimation in challenging low-illumination conditions.

- **A novel statistical non-rigid point registration.**(*Chapter 7*)

  A statistical algorithm is proposed for non-rigid point cloud registration, addressing the challenge of handling topology changes without the need to estimate the correspondence points of two point clouds. A novel *Break and Splice* framework is developed to cluster a pair of point sets and assign labels to the source point cloud and target point cloud. The point clouds are registered with the same labels using the Bayesian Coherent Point Drift (BCPD) method [47]. The proposed approach is evaluated on three public datasets and two of our datasets using various qualitative and quantitative metrics. The results show that the *Break and Splice* framework outperforms the state-of-the-art methods and achieves error reduction $\sim 60\%$ and a registration time reduction $\sim 57.8\%$.

- **Two AR applications based on the proposed methods.**(*Chapter 8*)

  The accurate localisation of instruments is useful in minimally invasive surgery. AR applications based on the proposed depth estimation framework are developed to provide valuable augmented information for feedback, such as the relative distance of instruments, for robotic surgery. In addition, AR applications using dynamic 3D organ reconstruction can overcome missing depth if organs are beneath the instruments.

1.5 Thesis Outline

- **Chapter 1:** Introduction of research background, aim and main contributions.

- **Chapter 2:** Literature review on depth recovery from video, and introductions of traditional algorithms, supervised and unsupervised learning-based methods. Furthermore, the review encompasses many non-rigid point cloud registration methods, including general transformation and Gaussian Mixture Models. 3D dynamic surface reconstruction methods are also mentioned.

- **Chapter 3:** Theoretical foundation is provided upon which this thesis is built. In particular, the mathematical background is used to support the proposed depth estimation framework, including the group equivariance deep learning and projective geometry. The RGB-D imaging process is also introduced, which will be used to conduct non-rigid registration experiments.
- **Chapter 4:** A novel group equivariance deep learning framework is proposed for monocular depth estimation based on the mathematical foundations described in Chapter 3 to support the validity of the coarse-to-fine encoder model.
- **Chapter 5:** A use case on endoscopy videos is used to evaluate the proposed framework described in Chapter 4, and a new loss function is designed to improve the endoscopy video in low-illumination conditions.
- **Chapter 6:** Another use case on videos captured by UAVs in unstructured environments, which are similar to endoscopy environments with free-motion cameras and textureless regions, is used to evaluate the proposed framework in Chapter 4.
- **Chapter 7:** A statistical method for non-rigid point cloud Registration. This chapter utilises *Break and Splice* framework to handle point clouds undergoing topology changes and large inter-frame motions.
- **Chapter 8:** AR applications of organ 3D dynamic reconstruction and instruments tracking based on the depth map.
- **Chapter 9:** Conclusion and future work.

1.6 List of Publications

Accepted Paper

- **QingHong Gao**, Yan Zhao, Wen Tang, TaoRuan Wan, Long Xi. Break and Splice: A Statistical Method for Non-rigid Point Cloud Registration. *Computer Graphics Forum (2023).*
- Long Xi,Yan Zhao, Long Chen, **QingHong Gao**, Wen Tang, TaoRuan Wan, Tao Xue. Recovering Dense 3D Point Clouds from Single Endoscopic Image. *Computer Methods and Programs in Biomedicine (2021).*

Paper Under Review

- **QingHong Gao**, Tao Ruan Wan, Wen Tang: G-depth: A Self-supervised Deep Learning Model based on Group Equivariance for Monocular Depth Estimation of Endoscopy. *IEEE Transactions on Medical Imaging.*
- **QingHong Gao**, Tao Ruan Wan, Wen Tang: UAV-depth: A Self-supervised Monocular Depth Estimation for Unstructured Environments. *IEEE Transactions on Geoscience and Remote Sensing.*

Chapter 2

Literature Review

This chapter reviews the state-of-the-art approaches, methodologies, and computational models for 3D surface reconstructions and non-rigid point cloud registrations. Since this PhD work mainly focuses on monocular depth estimation for generating dense depth maps for 3D surface reconstructions, a literature review on Structure from Motion (SfM) based and feature-based methods is first introduced. Secondly, supervised and unsupervised deep learning-based methods are reviewed for monocular depth estimation. For non-rigid point cloud registration, general transformation models and Gaussian mixture models are described, and the registration problem is analysed in terms of its applications on 3D dynamic surface reconstruction.

2.1 Depth Estimation from Videos

Depth estimation refers to the set of techniques and algorithms used to obtain a representation of the spatial structure of a scene, aiming to measure the distance of each point in the observed scene. Many depth estimation methods [48], [49], [50] rely on stereo matching to estimate depth maps. Stereo depth estimation methods are known for their accuracy since the disparity between corresponding points in the left and the right images is obtained to calculate the depth. However, stereo-vision-based methods come with intrinsic limitations. Collecting stereo images requires precise alignment and calibration procedures, which can be complex and time-consuming. Additionally, the baseline distance between the two cameras used in stereo setups can limit the effective range for accurate depth estimation. The farther the objects are from the camera pair, the more inaccuracy in the depth estimation might occur. Stereo cameras are still too large to be widely used in practice, such as in endoscopy surgery or micro drones, compared with monocular cameras.

2.1.1 Traditional Methods

Monocular depth estimation is considered an ill-posed problem because a single monocular image can be captured from different 3D scenes. Therefore, traditional algorithms exploit monocular cues, such as texture, occlusion, known object size, and lighting and shading,

to recover depth. These methods can often be classified into two categories: SfM-based and handcrafted feature-based methods [51].

SfM is a process of predicting and reconstructing 3D structures from a series of images taken from different viewpoints [52]. This process commonly starts with a feature extraction step from a sequence of input images. SfM, then, finds the correspondence between different images based on texture features and removes the incorrectly matched points. Finally, once matched between images, features are tracked from one image to another to estimate the camera motion between the images. By triangulating the matched feature points from multiple camera viewpoints, a 3D point cloud representation of the scene is constructed.

Wedel *et al.* [28] utilized SfM to determine the scaling factor of supervised image regions and estimated the scene depth. Let $\mathbf{I}(t) = (X(t), Y(t), Z(t))^\top$ be a point the value of 3D position at time t and its corresponding image point as $I(t)$. The point at time $t+\tau$ can be defined by $\mathbf{I}(t+\tau) = \mathbf{I}(t) + \mathbf{T}(t+\tau)$, where $\mathbf{T}(t+\tau)$ is camera translation between time t and $t+\tau$. Scene depth can be directly calculated through given vehicle translation$T_Z(t,\tau)$ and displacement of image points:

$$d = \frac{s(t,\tau)}{1-s(t,\tau)} T_Z(t,\tau) \tag{2.1}$$

where the $s(t,\tau)$ is directly obtained by region tracking [53].

Prakash *et al.* [29] utilized a multiscale fast feature point detector to detect key points in the image, and these corresponding 2D points were used to calculate 3D points through two-view geometry and triangulation. In this method, the scene depth values of feature points are computed through a metric transformation. Hyowon *et al.* [54] proposed a depth acquisition pipeline from a small camera motion, which utilized the Harris corner detector [55] in a reference image to extract features and found the corresponding points through the Kanade-Lukas-Tomashi (KLT) algorithm [56]. The dense depth map was reconstructed through a so-called plane sweeping [29]. Javidnia *et al.* [30] used ORB features [57] to improve computing efficiency.

The accuracy of depth maps generated by SfM-based methods depends on the effectiveness of feature detection algorithms and the feature matching accuracy between the image pair. Feature detection can be challenging, especially in texture-less or low-contrast scenes that can result in a small number of key feature points. As a result, many existing SfM-based methods tend to produce sparse depth maps. For many applications, sparse depth maps are not sufficient in terms of measuring the detailed information required for high-precision operations. For example, for endoscopy surgery or UAV navigations in complex environments, dense 3D depth maps are required to obtain accurate positional information for the surgical instruments or the control of UAVs.

Methods using handcrafted features often utilized superpixels as inputs to compute features. For each superpixel, these features and depth cues are used to estimate the depth. An MRF (Markov Random Field) model was applied to combine the superpixel-based depth estimation with the relationship and context between different superpixels to ensure

the coherence and smoothness of the depth map. A well-known classic method is called Shape from Shading (SfS) [58], which relies on the gradual changes in shading as a cue to estimate shape and depth. SfS assumes certain lighting conditions and the surface to be Lambertian (with diffuse reflection). It is not suitable for real-world scenarios. Torralba *et al.* [31] proposed a method to learn the relationship between the structures of the image and the mean depth of the scene. Using a set of features obtained from Fourier and wavelet transforms, and the mean depth of the scene, the absolute scene depth of monocular images can be inferred. Jung *et al.* [32] proposed a monocular depth estimation method by considering object types in a single-view image. This method defined four different object types and six attributes to describe object units. Each object was classified using the Bayesian classifier based on the training data, and depth values were then allocated differently based on the respective object types. Saxena *et al.* [59] and Liu *et al.* [33] used MRF incorporated with multiscale image features to learn monocular cues and estimated depth in a supervised manner. Pre-designed features were used to extract specific chosen characteristics. However, the need for pre-processing or post-processing imposes a computational burden, rendering these methods unsuitable for real-time applications such as endoscopy surgery and UAVs.

2.1.2 Deep Learning-based Methods

Deep learning-based methods have emerged as a promising approach for predicting depth maps from monocular videos. The methods can be broadly classified into supervised and unsupervised depending on the need for ground truth. Supervised deep learning methods often incorporate an individual image and its corresponding depth map ground truth to train a model and learn scene structural features for estimating a depth map. Unsupervised or Self-supervised models can be considered as an alternative when ground truth data is absent, as they can be trained by using a comparison between a target image and its reconstructed image as a supervisory signal.

Supervised Deep Learning Methods

Eigen *et al.* [35] proposed a multiscale convolutional neural network and a scale-invariant loss function to estimate depth from a single image. The real scale of depth is recovered without any post-processing. Compared with the commonly used method of uniform discretization, Fu *et al.* [60] proposed a spacing-increasing discretization strategy to discretize depth maps. This method can overcome over-strengthened loss for the large depth values, particularly in the KITTI dataset [61]. Guo *et al.* [62] used a stereo network to train with synthetic data as a pre-trained model. The real data is used to refine the depth model under supervised or unsupervised settings, which reduces the domain gaps between synthetic and real data. Recently, Chen *et al.* [63] proposed an attention-based context aggregation network for capturing the continuous context information and improving the depth estimation. These methods performed well on KITTI [61] and Make3D [64] datasets. The KITTI dataset was captured by driving around the mid-size city of Karlsruhe, in

rural areas and on highways in a structured environment. The Make3D dataset includes monocular images and corresponding depth maps, but monocular or stereo sequences are unavailable. However, depth estimation in endoscopic scenes remains challenging due to the difficulty of obtaining ground truth depth data, making it challenging to develop supervised learning methods for this task. To address this problem, most methods [65], [66] utilized synthetic endoscopy images to train depth maps. However, they tend to perform poorly since synthetic depth data may not represent the full range of diversity in real-world scenarios, including different lighting conditions, camera positions, and object orientations.

Unsupervised Deep Learning Methods

Garg *et al.* [67] proposed a self-supervised network to learn depth by a stereo photometric reprojection warping loss. Godard *et al.* [68] used a left-right consistency loss to improve predicted depths from the stereo images. Godard *et al.* [4] proposed a monocular self-supervised depth estimation network based on a per-pixel minimum reprojection loss. Johnston *et al.* [5] used self-attention and discrete disparity volume to increase the accuracy of depth estimation. Similarly, for endoscopy data, Turan *et al.* [69] proposed an unsupervised framework for real-time odometry and depth estimation by using monocular endoscopic video. Liu *et al.* [70] used sparse self-supervisory signals derived from SfM to establish supervision. Ozyoruk *et al.* [71] combined residual networks with a spatial attention module and a brightness-aware photometric loss to improve the robustness of depth estimation. Recently, Shao *et al.* [72] took advantage of the appearance flow to address the brightness inconsistency problem in depth estimation. These methods often fail to obtain fine details of depth, such as boundary objects, since they cannot learn intrinsic features based on Euclidean space. Therefore, Graph Neural Networks (GNNs) have been proposed to handle non-Euclidean data as a solution to the constraint of Convolutional Neural Networks (CNNs). Works for monocular depth estimation based on GNNs include Fu *et al.*, which used two steps to reconstruct depth maps. Masoumian *et al.* [6] embedded the graph convolution network into a decoder to improve the accuracy of depth maps. However, they utilized a random graph structure to learn features, which not only increased the training time but also lost the local feature information of point clouds.

2.2 Non-rigid Point Cloud Registration

Point cloud registration has many applications in computer vision, including 3D reconstruction, pose estimation, augmented reality, object matching and recognition [73], [14], [74], [75]. Accurate registration of multiple point clouds obtained from different views or time instants is necessary for building a complete and consistent 3D model of the scene. In addition, point cloud registration enables the estimation of the relative pose and motion of objects, the recognition and matching of objects in different scenes, and the creation of virtual and augmented reality experiences [76], [77].

While many registration methods work well on rigid objects [78], [79], they often perform poorly on dynamic scenes or deformed objects. This is because objects with non-rigid deformations and motions cannot be modelled by rigid transformations. In addition, non-rigid objects may undergo topology changes, such as separation, which is the act of creating a visible gap or distance between objects or individuals, and connection, which is regarded as a reverse process of separation. The topology changes pose additional challenges for registration methods that rely on correspondences between the source point sets and the target point sets [80], [81]. Therefore, developing registration methods that can handle non-rigid objects and dynamic scenes is an active research area in computer vision. In this chapter, non-rigid point cloud registration methods are reviewed, which involve general transformation models and Gaussian mixture models. 3D dynamic surface reconstruction methods based on non-rigid point cloud registration are also introduced.

Chui *et al.* [42] used a thin plate spline (TPS) to define a general transformation model that consisted of an affine transformation and a TPS smoothness kernel. Yang *et al.* [43] used a global and local mixture distance to estimate the correspondence between the source point set and the target point set and update the rigid and non-rigid transformations and minimized the mixture distance using a TPS. Huang *et al.* [80] utilized a rigid local transformation for each point to obtain a global non-rigid registration. Meanwhile, the local affine transformation [82], [83] is frequently applied in non-rigid registration because the surface representation allows surface details to be captured precisely due to its more freedom. However, general transformation methods depend on correspondence estimation based on the features of the source point and target point sets, and the result of correspondence estimation directly affects registration accuracy and efficiency.

Myronenko *et al.* [84], [85] proposed a Coherent Point Drift (CPD) algorithm for Gaussian mixture models. They formulated the registration as a maximum likelihood estimation problem, where one point set moves coherently to align with the other set under motion coherence constraint over the velocity field.

Based on the motion coherence theory [86], two adjacent points tend to move coherently, and this motion coherence is an important feature that influences the smoothness of the transformation. Golyaniet *et al.* [87] extended the CPD registration algorithm using correspondence priors and a coarse-to-fine optimization strategy to achieve robust non-rigid point registration with an improved speed of the registration process. Bai *et al.* [88] proposed a statistical framework by aligning two point sets represented by Gaussian mixture models. Hirose [89] proposed the Bayesian Coherent Point Drift (BCPD) method, which formulates CPD in a Bayesian setting to improve registration accuracy and efficiency. These methods achieved good results for the connection but failed to register the separation of two objects. The main reason is that the CPD-based framework requires all points to be transformed coherently as a whole (e.g. a single point set) whose displacement must meet the coherent point drift condition. When there are separations and connections during the object topology change, the point set will separate into two sets. Thus, CPD-based methods would fail to address the non-rigid registration challenges. Recently, Zampogiannis *et*

al. [90] proposed a framework to address the issues of separation and connection, but this method did not work well on large inter-frame motions.

Non-rigid registration methods have been applied to dynamic reconstructions. Newcombe *et al.* [40] proposed a DynamicFusion system that fused live frame depth maps into the canonical space via the estimated warp field to achieve high-quality 3D models. The experiment in DynamicFusion does not deal with large inter-frame motions with object topology change issues of separations and connections. The Fusion4D [91] and Kaiwen [92] used RGBD camera inputs to reconstruct dynamic scenes simultaneously. Although the method of Kaiwen tackled the object connection issue, object separation remains unsolved. [93] and [94] proposed new methods to tackle this issue by incorporating volumetric data. However, the detailed 3D information of volume-based methods is lower than that of other point-based registration methods.

2.3 Summary

In summary, the depth estimation from video and non-rigid point cloud registration methods have both been well-developed in recent years. Depth estimation with a dense 3D map or surface is a crucial technology for AR applications in endoscopy surgery and UAVs obstacle avoidance and control, as it plays a significant role in enhancing precision and stability. However, existing self-supervised depth estimation algorithms and methodologies struggle to capture the intricate topological details inherent in scenes captured by monocular devices. In addition, depth estimation networks based on GNNs often showcase their works primarily through experimental results, lacking a robust mathematical background. This research proposed a self-supervised GNNs-based coarse-to-fine encoder to achieve non-Euclidean features and improve depth maps of endoscopy and UAV scenes. The next chapter also provides mathematical details about the translation equivariance in CNNs and permutation equivariance in GNNs. These details serve to elucidate how these networks possess the capability to learn diverse and meaningful features for various objectives. Non-rigid point cloud registration is essential to the 3D dynamic surface acquisition, and 3D dynamic scene reconstruction is vital to the SLAM-based AR system. Most existing point cloud non-rigid registration methods have limitations in handling topology changes and large inter-frame motions. This research proposes a *Break and Splice* no-rigid point cloud registration framework, which integrates the Dirichlet Process Gaussian Mixture Model (DPGMM) and BCPD to achieve non-rigid registration with topology changes, to overcome the aforementioned challenges. In addition, a non-rigid point cloud registration example is conducted with endoscopy datasets provided by proposed monocular depth estimation methods.

Chapter 3

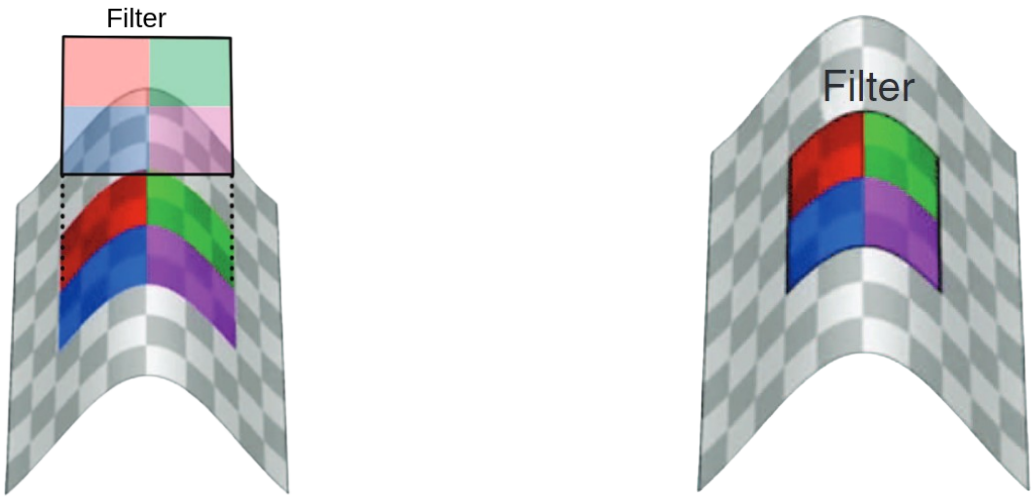
Mathematical Fundamentals

This chapter gives an overview of the mathematical fundamentals and background used in the development of depth estimation and AR applications. We first introduce Euclidean and non-Euclidean features. Then, group equivariance deep learning is introduced to support the proposed novel self-supervised framework developed in this PhD research for depth estimation. In particular, translation equivariance and permutation equivariance are used in this research. The Graph Convolution Network(GCN) and GAT network (Graph Attention network) will be introduced for the permutation equivariance. Finally, projective geometry is used to explain the relation between RGB images and 3D point clouds, which is also applied to AR applications.

3.1 Euclidean and Non-Euclidean Features

The deep learning on depth estimation for images should not only learn Euclidean features, but also learn non-Euclidean features. The Euclidean features are usually based on Euclidean geometry, which assumes that space is flat and parallel lines never intersect [3]. Particularly, the images can be regarded as a 2D grid of points which follows the rules of Euclidean geometry. The CNN-based methods [95], [96], [97] are designed to process data on regular grids for object recognition, segmentation, and classification. These tasks are usually effective for obtaining 2D results, e.g. pixels. The principal limitation of these approaches often stems from their treatment of geometric data as Euclidean structures. Firstly, for intricate 3D objects, Euclidean structures like depth images or voxels may result in the loss of significant parts of the object, including fine details and topological structure. Secondly, Euclidean structure lacks intrinsic properties and occurs variation with changes in pose or object deformation. Attaining invariance to shape deformations demands complex models due to the considerable degrees of freedom involved in describing non-rigid deformations, as shown in 3.1.

The depth estimation of images is to recover 3D point clouds. Scenes such as endoscopic scenes are usually complex since the surface of the objects in the scene is often textureless and occluded. In addition, they include not only the motions of instruments, but also the deformation of the objects. Therefore, learning only the grid geometry features is



(a) Filter of Euclidean (b) Filter of Non-Euclidean

Fig. 3.1 An illustration of the difference between filter(colored window) of Euclidean and Non-Euclidean on a 3D shape [1]. (a) A classical CNN applied to a mesh considered as a Euclidean object. (b) a geometry network filter applied intrinsically on the surface, and the convolutional filter is deformation invariant by construction.

insufficient. In addition to Euclidean features, the 3D surface of the deformable objects also often exhibits non-Euclidean features, such as curvature, geodesic distance and hyperbolic angles. To learn non-Euclidean features of 3D surfaces, a graph structure similar to point clouds, including irregular and out-of-order properties, is proposed. The graph edges can capture relationships between neighbouring pixels in the image, while the nodes can explore the depth information in terms of 3D coordinates of point clouds.

3.2 Group Equivariance Deep Learning

Neural networks are designed for specific data types, and their operations are based on built-in assumptions about the data, which lead to the emergence of symmetries [1]. For example, CNNs are designed for grid data such as images in the translation symmetry group [98], geodesic CNN [99] is designed for manifold data such as meshes in the gauge symmetry group, and GNNs is used in sociology [100] or the prediction of chemical reactivity [101], which usually utilized graphs as models of analysis and a graph system is invariant for permutation symmetry group.

The symmetries can provide insights into the underlying structure of a system or an object. In this research, the properties of the symmetry group and its equivariance are used to design the proposed network framework for depth estimation. A symmetry group is a group of transformations that leave an object or a system invariant in its structure. In other words, if a system has a symmetry group G, applying any transformation in G to the system will not change its structure. For example, a rigid object has two symmetry groups: the translation group and the rotation group [102]. Equivariance is the property of a transformation that preserves the symmetry of an object or system. If a function is

equivariant with respect to a symmetry group G, applying any G transformation to the input will result in the same transformation being applied to the output. For example, GNNs are permutation equivariant networks that operate on graph-structured data [103], [104]. Therefore, a GNNs-based structure is used to learn non-Euclidean features and achieve more accurate depth images or point clouds than others. In addition, a mathematical framework is used to demonstrate this hypothesis based on the geometry properties of symmetry.

3.2.1 Symmetry Group

A symmetry of an object is a transformation that leaves a certain property of an object invariant, and a set of transformations with certain properties (Associativity, Identity, Inverse, Closure) is called a symmetry group [102]. For example, a set of 2D integers $I(u,v) = \{(u,v) \mid (u,v) \in \mathbb{Z}^2\}$ is a symmetry group of translations. For associativity, $((u,v)+(m,n))+(p,q) = (u+m+p, v+n+q) = (u,v)+((m,n)+(p,q))$, where $(u,v),(m,n),(p,q) \in \mathbb{Z}^2$; Existence of identity, the $I(u,v)$ of translation operators by vector $\mathbf{0}$ is the identity operator; Every element of $I(u,v)$ has an inverse: $I(-u,-v) = \{(-u,-v) \mid (u,v) \in \mathbb{Z}^2\}$, and $(-u,-v)+(u,v) = (0,0) = \mathbf{0}$; At last, for closure $(u,v)+(m,n) = (u+m, v+n)$, it means that the sum of two translations is again a translation. Therefore, a set of 2D integers can be verified to satisfy translation group properties. The symmetry group is meaningful since different convolutions can be explained by exploring the invariant for different transformation groups [1], such as CNNs for the translation group and GNNs for the permutation group. In addition, the properties of the group are used to explain the proposed monocular depth estimation framework.

The basic concepts that are helpful in understanding the proposed model need to be introduced. In the context of data (such as a grid or graph), a symmetry group G represents a set of geometric transformations. This research is mostly interested in how symmetry groups act on data, called group action. Let X be a set and a group G, a left action of G on X is a map $\rho : G \times X \to X$ that is compatible with the group properties, such as $g_1 \circ (g_2 \circ x) = (g_1 \circ g_2) \circ x$ for all $g_1, g_2 \in G$ and $x \in X$, where $\circ$ is the operation of juxtaposition. Group action plays an important role in defining the invariance of data. Formally, given $x \in X$, $g \in G$ and a group representation $\rho_X(g)x$ of G on X, the definition of $G-$**equivariant** is given by group actions, as follows:

Definition 1 *The encoding function $f : X \to Y$ is $G-$**equivariant** if $f(\rho_X(g)x) = \rho_X(g)f(x)$ $\forall x \in X, \forall g \in G$. $f : X \to Y$ is an autoencoder framework with encoding function, mapping between the data domain X, and latent domain Y.*

Definition 1 means that there is the same effect for group action on input and output. This is an important property to analyse the proposed method. In the following sections, the mathematical derivations of the translation-equivariant on CNNs for the grid geometry structure and the permutation-equivariant on GNNs for the graph geometry structure will be introduced.

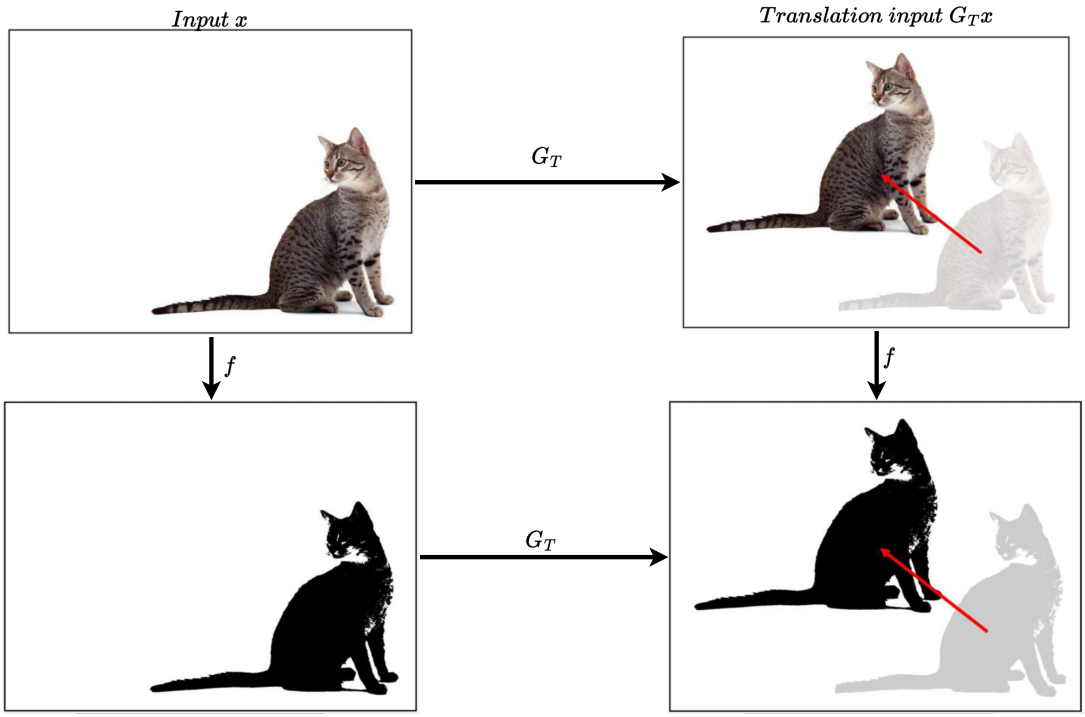


Fig. 3.2 Translation-equivariant of CNNs [2], f is a segmentation network of cat and G_T is translation operator.

3.2.2 Translation-equivariant of CNNs

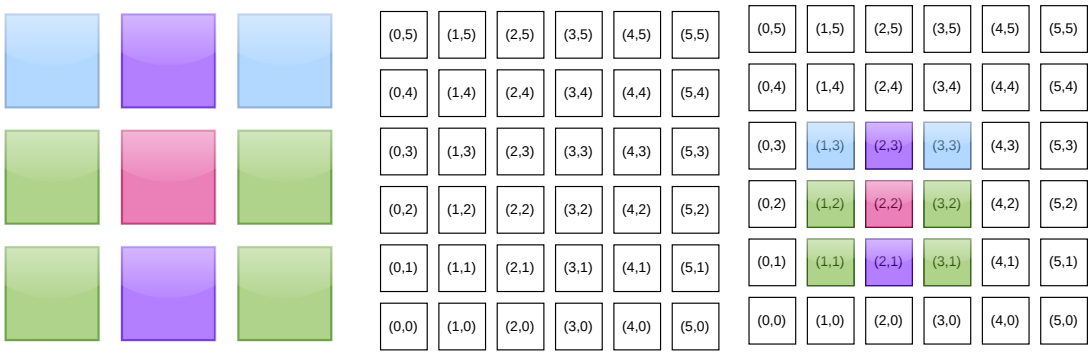
In order to show that CNNs are translation-equivariant, both visualization explanation and explicit derivation are introduced. Fig. 3.2 shows an example of cat segmentation in CNNs. In this case, the output results of two processes are the same, i.e. cat being shifted first and then segmented has the same results as the cat being segmented and then shifted. This process can be written: $G_T f(x) = f(G_T x)$, which means that the cat segmentation network is translation-equivariant. Although it is certainly easy to see through visualization, an explicit derivation can be useful to understand that CNNs are equivariant to the translation group. Therefore, we first recall the definition of convolution used in CNNs.

For each CNNs layer l, feature maps can be stacked $I : \mathbb{Z}^2 \longrightarrow \mathbb{R}^{M^l}$ and convolved with a shared weight $K^l : \mathbb{Z}^2 \longrightarrow \mathbb{R}^{M^l}$:

$$(I * K^l)[i,j] = \sum_{a\in\mathbb{Z}} \sum_{b\in\mathbb{Z}} \sum_{c=1}^{M^l} I_c[a,b] K_c^l[i-a, j-b] \tag{3.1}$$

where $i \in \mathbb{Z}$ and $j \in \mathbb{Z}$ are the value of coordinate after convolution, $a \in \mathbb{Z}$ and $b \in \mathbb{Z}$ are value of coordinate in a pixel, M^l is the number of input channels, and let $M^l = 1$ for brevity, c is the channel number. Then, if CNNs are translation-equivariant, for any translation $g_t = (t_1, t_2) \in G_T$, where G_T is the group of all translations of $\mathbb{Z}^2$, they should satisfy the equation as follow:

$$((g_t I) * K)[i,j] = (g_t (I * K))[i,j] \tag{3.2}$$



(a) Feature map (b) Pixel map (c) Original relationship

Fig. 3.3 An image data can be regarded as a combination of a feature map and a pixel map. (a) is part of a feature map under a 3x3 convolution kernel, (b) is part of a pixel, and (c) is the initial position relationship between the feature map and the pixel map.

where the left side indicates applying a translation transformation action on the feature map, followed by convolution. The right side signifies performing convolution first and then applying a translation transformation action on its outcome. The same notation is also mentioned in [102], in which a translation transformation action on the feature map (the values of RGB or grayscale): $g_t I$ and action on the point (the value of a pixel position): $g_t^{-1}[a,b]$ are the same operations. In other words, $g_t I$ is to move the feature map and fix the pixel map, but $g_t^{-1}[a,b]$ is to move the pixel map and fix the feature map, where g_t^{-1} is the inverse of g_t. Visualization is used to illustrate the $g_t I = g_t^{-1}[a,b]$, from Fig. 3.3 to Fig. 3.5. Fig. 3.3 shows the initial position relationship between the feature and pixel maps. Fig. 3.4 shows the process of moving the feature map and fixing the pixel map. Fig. 3.5 shows the process of moving the feature map and fixing the pixel map. It can seen that $g_t I$ and $g_t^{-1}[a,b]$ have the same results.

In addition, $g_t^{-1}[a,b]$ means that a pixel $[a,b]$ is shifted by $g_t^{-1} = (-t_1,-t_2)$ and can be written $g_t^{-1}[a,b] = [a-t_1, b-t_2]$. $(g_t I)$ can be expressed as the Eq.3.3:

$$(g_t I)[a,b] = I(g_t^{-1}[a,b]) = I[a-t_1, b-t_2] \tag{3.3}$$

For Eq.3.2, we use the substitution $a \longrightarrow a+t_1$ and $b \longrightarrow b+t_2$, and it can be expressed as Eq.3.4:

$$\begin{aligned}
((g_t I) * K)[i,j] &= (I[a-t_1, b-t_2] * K)[i,j] \\
&= \sum_a \sum_b I[a-t_1, b-t_2] K[i-a, j-b] \\
&= \sum_a \sum_b I[a,b] K[i-t_1-a, j-t_2-b] \\
&= \sum_a \sum_b I[a,b] K[(i-t_1)-a, (j-t_2)-b] \\
&= (I * K)[i-t_1, j-t_2] \\
&= (g_t (I * K))[i,j]
\end{aligned} \tag{3.4}$$

(a) Moving feature map (b) Result

Fig. 3.4 This process is about moving the feature map and fixing the pixel map:(a) shows the colour squares move to the position of (4,4) centre. (b) shows the result of movement $g_t I$.

(a) Moving pixel map (b) Result

Fig. 3.5 This process is about moving the pixel map and fixing the feature map: (a) shows the white square of (4,4) pixel position move to the position of pink. (b) shows the result of movement $g_t^{-1}[a,b]$.

Therefore, CNNs are equivariant to the translation group so that they can learn the same feature from different images. However, CNNs are not equivariant to the permutation group since CNNs will fail to get the same feature when the order of the grid nodes, which a 2D image can be as a grid, is changed. For depth estimation or point cloud, they are out of order. So, we apply an additional graph neural network to learn the intrinsic features of a point cloud.

3.2.3 Permutation-equivariant of GCN

In this section, the definition of the layer propagation rule on GCN [105] is recalled. We use the graph convolution with the following layer propagation rule:

$$F(X,A) = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{\frac{1}{2}}XW) \tag{3.5}$$

where X is the node features of input, $\tilde{A} = A + I_N$ is the adjacency matrix with added self-connections (I_N is the identity matrix), $\tilde{D} = diag(\sum_{i \neq j} \tilde{a}_{ij})$ is the degree matrix, W is a trainable weight matrix, and $\sigma(\cdot)$ denotes an activation function. Therefore, applying a permutation matrix Π ($\Pi^\top \Pi = \Pi\,\Pi^\top = I$) to the node features X automatically implies applying it to rows and columns of the adjacency matrix A, which can be written as $\Pi A \Pi^\top$. The $F(X,A)$ is permutation-equivariant for any permutation matrix Π if:

$$F(\Pi X, \Pi A \Pi^\top) = \Pi F(X,A) \tag{3.6}$$

According to Eq.3.5, and the left of Eq.3.6 can be written as :

$$\begin{aligned} F(\Pi X, \Pi A \Pi^\top) &= \sigma(\Pi\tilde{D}^{-\frac{1}{2}}\Pi^\top \Pi\tilde{A}\Pi^\top \Pi\tilde{D}^{\frac{1}{2}}\Pi^\top \Pi A W) \\ &= \sigma(\Pi\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{\frac{1}{2}} A W) \end{aligned} \tag{3.7}$$

Because the nonlinear activation function of a graph network is an element-wise operation, which is permutation equivariant [106], the Eq.3.7 can be written as :

$$\begin{aligned} \sigma(\Pi\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{\frac{1}{2}} A W) &= \Pi\sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{\frac{1}{2}} A W) \\ &= \Pi F(X,A) \end{aligned} \tag{3.8}$$

where the left of Eq.3.7 is equal to the right of Eq.3.7. Therefore, GCN is equivariant to the permutation group.

3.2.4 Permutation-equivariant of GAT

Similar to GCN, the definition of the layer propagation rule on GAT [107], [45] is also recalled. Let $G_a = (N,E,A)$ be an undirected weighted graph with the nodes N and the edges E, represented by the adjacency matrix $A = (a_{ij})$, where $a_{ij} = a_{ji}, a_{ij} = 0$ if $(i,j) \notin E$

and $a_{ij} > 0$ if $(i,j) \notin E$. At the layer l, the graph convolution operation is defined as :

$$h_v^{(l)} = \sigma(\sum_{u \in N(u) \cup v} \alpha_{vu}^{(k)} W^{(l)} h_u^{(l-1)}) \tag{3.9}$$

where $\sigma(\cdot)$ denotes an activation function, $N(u) \cup v$ denotes that add a self-loop for all nodes, the attention weight $\alpha_{vu}^{(k)}$ measures the importance between the node v and its neighbour u, and the definition of α in [45] is used. For a node feature x_v and the set of its neighbourhood S_{A_v}, we use $f(x_v, S_{A_v})$ to define a node operation function, where f is same to Eq.3.9 and h_v only includes the feature of a node. Then, all nodes can be defined as:

$$F(X,A) = [f(x_1, S_{A_1}), f(x_2, S_{A_2}), \ldots, f(x_n, S_{A_n})] \tag{3.10}$$

where $F(X,A)$ means that the function f applies independently to neighbourhood of every node. Therefore, similar to Eq. 3.6 if :

$$F(\Pi X, \Pi A \Pi^\top) = \Pi F(X,A) \tag{3.11}$$

where $\Pi X = [x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(n)}]$, and we use the substitution $\Pi A \Pi^\top \longrightarrow B$, where B is adjacency matrix of $G_b = (N,E,B)$ (where G_a and G_b are isomorphic). The left of Eq.3.11:

$$\begin{aligned} F(\Pi X, \Pi A \Pi^\top) &= F(\Pi X, B) \\ &= \left[f(x_{\pi(1)}, S_{B_1}), \ldots, f(x_{\pi(n)}, S_{B_n})\right] \end{aligned} \tag{3.12}$$

For the right of Eq.3.11:

$$\Pi F(X,A) = \left[f_{\pi(1)}, f_{\pi(2)}, \ldots, f_{\pi(n)}\right] \tag{3.13}$$

where $f_{\pi(n)} = f(x_{\pi(n)}, S_{A_{\pi(n)}})$. $S_{A_{\pi(n)}}$ and $S_{B_{\pi(n)}}$ have same elements for nodes feature $x_{\pi(n)}$. Since summation does not depend on the order of the set, $f_{\pi(n)} = f(x_{\pi(n)}, S_{B_n})$ and the left of Eq.3.11 is equal to the right of Eq.3.11. Therefore, GAT is equivariant to the permutation group. Because the translation group is the subgroup of the permutation group and the geometry properties of the point cloud are similar to the permutation group, graph-based networks can learn more information than translation equivariant and fine details for depth images. Although GCN and GAT have the same symmetry group, GAT is suitable for different rankings of neighbours and can assign different importance to nodes of the same neighbourhood [45]. Then, the details of the implementation based on G-equivariance will be introduced in Chapter 4.

3.3 Projective Geometry

The development of methods relies on the utilization of projective geometry, such as a pinhole camera model. In particular, it can explain the reprojection error mentioned in

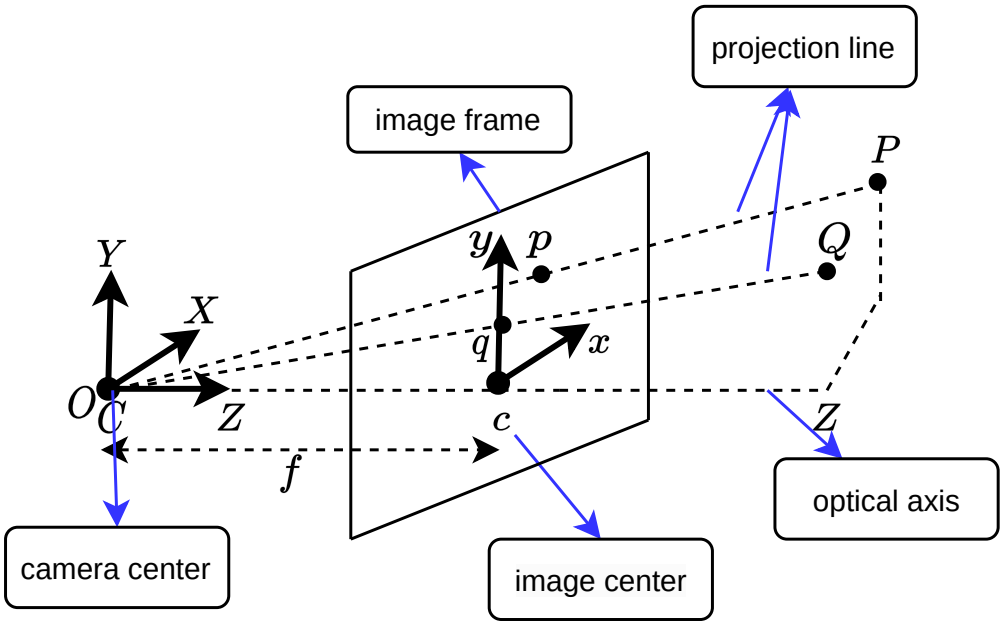


Fig. 3.6 Pinhole camera model [3].

monocular depth estimation, the transformation between RGB-D images and point cloud that are used to experiment in Chapter 7 and the registration of AR applications between virtual objects and the real world.

The camera model can describe the projection of a real-world 3D scene onto an image plane. The pinhole camera model is used in this research. This model is a simplified representation of a camera that assumes it to be a basic optical device without lenses, where light enters through a pinhole and projects an image onto the camera's image plane. Figure 3.6 illustrates the process of projection. This model is an idealized approximation that overlooks distortions, leading to higher accuracy near the optical image centre and reduced precision towards the edges. However, these limitations can be addressed through suitable calibration methods [108], [109]. Therefore, this model is often used in computer graphics and computer vision to simplify the mathematical description of camera projection and scene rendering.

As shown in Figure 3.6, the distance between the camera centre and the image centre is the focal length f of the camera. The symbols f_x and f_y represent the scaling factors for the focal lengths in the x and y directions, respectively. The image centre or principal point (c_x, c_y) is the intersection of the optical axis and the image plane. The intrinsic camera matrix can be written as follows:

$$\mathbf{K} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{3.14}$$

The projection operator is that projects a 3D point $P = (X,Y,Z)^\top$ onto a pixel (u,v) in the image plane as follow:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{K}(\mathbf{I}_{3\times 3}\ \mathbf{0}_{3\times 1}) \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{3.15}$$

where $\mathbf{I}_{3\times 3}$ is identity matrix and assumes the camera is static.

For the monocular depth estimation introduced in Chapter 4, the Eq. 3.15 need to add a rotation $\mathbf{R}$ and a translation $\mathbf{t}$ due to the camera moved, as follow:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{K}(\mathbf{R}\ \mathbf{t}) \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{3.16}$$

where $\mathbf{R}$ and $\mathbf{t}$ can be obtained by an unsupervised network, and according to the depth map, the pixel coordinates can be calculated. The reprojection means that different viewpoints for the same static object have the same 3D point values. Therefore, the depth created by a network can get the pixel coordinates, which can be used to compare with the original input images.

For the RGB-D images used in Chapter 7, this type of data is usually captured by RGB-D camera, such as Kinect [25] or RealSense [110]. They can acquire a 2D depth image, which stores the distance to the surfaces observed from the camera centre. Therefore, the point cloud can be computed by Eq. 3.15 as follows:

$$\begin{aligned} X &= \tfrac{(u-c_x)\cdot Z}{f_x} \\ Y &= \tfrac{(u-c_y)\cdot Z}{f_y} \\ Z &= D \end{aligned} \tag{3.17}$$

where D is the depth value from a depth image.

For the registration between virtual objects and the real world, Eq. 3.16 can be directly used, where (u,v) is the video screen coordinates, and $P = (X,Y,Z)^\top$ is the virtual object 3D information. In addition, the $\mathbf{R}$ and $\mathbf{t}$ can be obtained based on Perspective-n-Point [111].

3.4 Summary

In this chapter, the concepts of Euclidean and Non-Euclidean Features are proposed for learning image geometry features according to different networks. Specifically, CNNs are shown to excel in learning 2D grid Euclidean features, while GNNs are adept at learning graph non-Euclidean features The concept of group equivariance deep learning is used to provide visualization and mathematical proofs about translation-equivariant

in CNNs and permutation-equivariant in GCN. will present a novel network grounded in the principles of group equivariance. In addition, we leverage projective geometry to explain the relationship between images and 3D points, setting the groundwork for subsequent chapters. This framework will be applied in Chapter 4 for self-supervised training, Chapter 7 for generating point clouds from depth images and RGB images, and Chapter 8 for coordinate transformations.

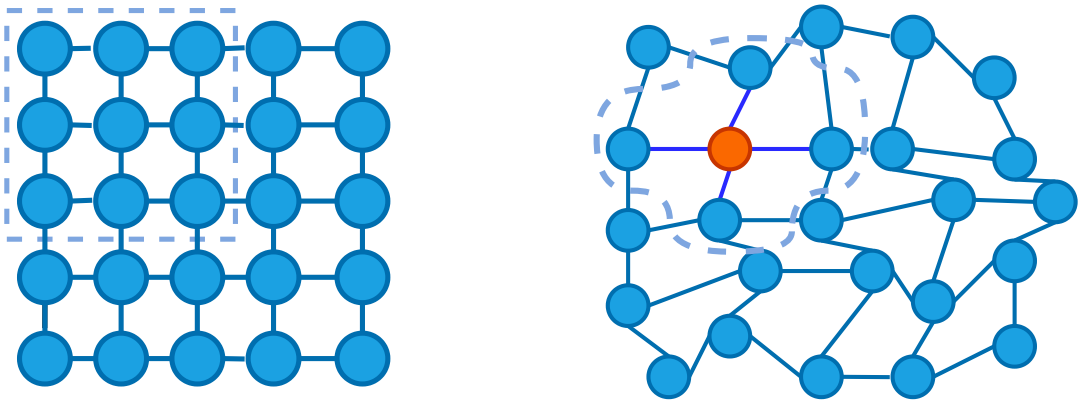
Chapter 4

Group Equivariance Deep Learning Framework

4.1 Introduction

In this chapter, a group equivariance deep learning framework is proposed for monocular depth estimation. Previous unsupervised monocular depth estimation methods mentioned in Chapter 2 rely on convolutional neural networks (CNNs) as the main learning framework. CNNs can learn hierarchical features from raw image data for depth estimation. CNNs are designed to process data on regular grids and capture spatial correlations and patterns in Euclidean space, as shown in Fig. 4.1a. However, most data is usually not ordered or regular, such as point clouds, which need to extend deep neural networks to non-Euclidean domains [67], where the geometry structure of the graph is similar to that of the data, such as the point clouds, with out-of-order and irregular shapes. There are some works based on graph neural networks (GNNs), as shown in Fig. 4.1b, which handle points cloud tasks, such as segmentation and classification [112], [113]. For self-supervised monocular depth estimation, there are relatively few works based on GNNs. Fu *et al.* [114] use two steps to reconstruct depth maps, which will lead to an increase in the complexity of the proposed model. Masoumian *et al.* [6] embedded the graph convolution network into a decoder to improve the accuracy of depth maps. However, they did not explain the reasons why their models are effective in terms of mathematical principles. [6] method utilized a random graph structure to learn features, which increased the training time and the loss of the local/neighbouring feature information of the image.

The main contribution of this novel deep-learning architecture is that a new coarse-to-fine encoder framework is proposed for depth estimation, which utilizes a GAT [45] to learn non-Euclidean features and refine the depth geometry features. Moreover, the graph structure based on four-connectivity can keep the original neighbour point information and the mathematical foundations described in Chapter 3 are used to support the validity of the coarse-to-fine encoder model.



(a) CNN on a grid data (b) GNN on a graph data

Fig. 4.1 Comparison of CNN and GNN on Euclidean and non-Euclidean domains, respectively. (a) CNN: The region of dashed is ordered and has a fixed size. (b) GNN: The region of dashed is out-of-order and irregular.

4.2 Group Equivariance Deep Learning

This section describes a novel self-supervised depth estimation framework based on the permutation-equivariant and translation-equivariant as described in Chapter 3. The proposed framework consists of three main components: a coarse-to-fine encoder, a discrete volume decoder, and a pose estimation network. To estimate the depth from monocular images, the CNN network based on U-Net [115] is used to learn coarse depth features, while the graph attention network is used to learn fine depth features. The decision-making process for depth estimation is further refined using a discrete volume decoder [5]. An essential part of the proposed framework is the pose estimation network, which calculates the relative transformation matrix by comparing the differences of neighbouring images. The entire framework of the proposed model is illustrated in Fig. 4.2, and the endoscopy use case is used as an example.

4.2.1 Coarse-to-Fine Encoder

For the coarse part of the encoder, the input is an image, which can be regarded as the grid and regular geometry data. CNNs can explore the global structure of image data by convolution weight sharing, which is owed to data distribution with approximately invariant to translations. Therefore, in our model, CNNs are suitable for extracting global coarse depth information from an input image. The coarse encoder includes five layers. The first layer is a fast convolutional layer, which includes three Conv3×3s (3×3 convolution, batch normalization, and ReLU activate function) and a max-pooling operation. The last four layers are ResNet-101 [97]. The network details are shown in Table 4.1.

For the fine part of the encoder, as mentioned in group equivariance in Chapter 3, the graph attention network can explore feature details based on permutation equivariant.

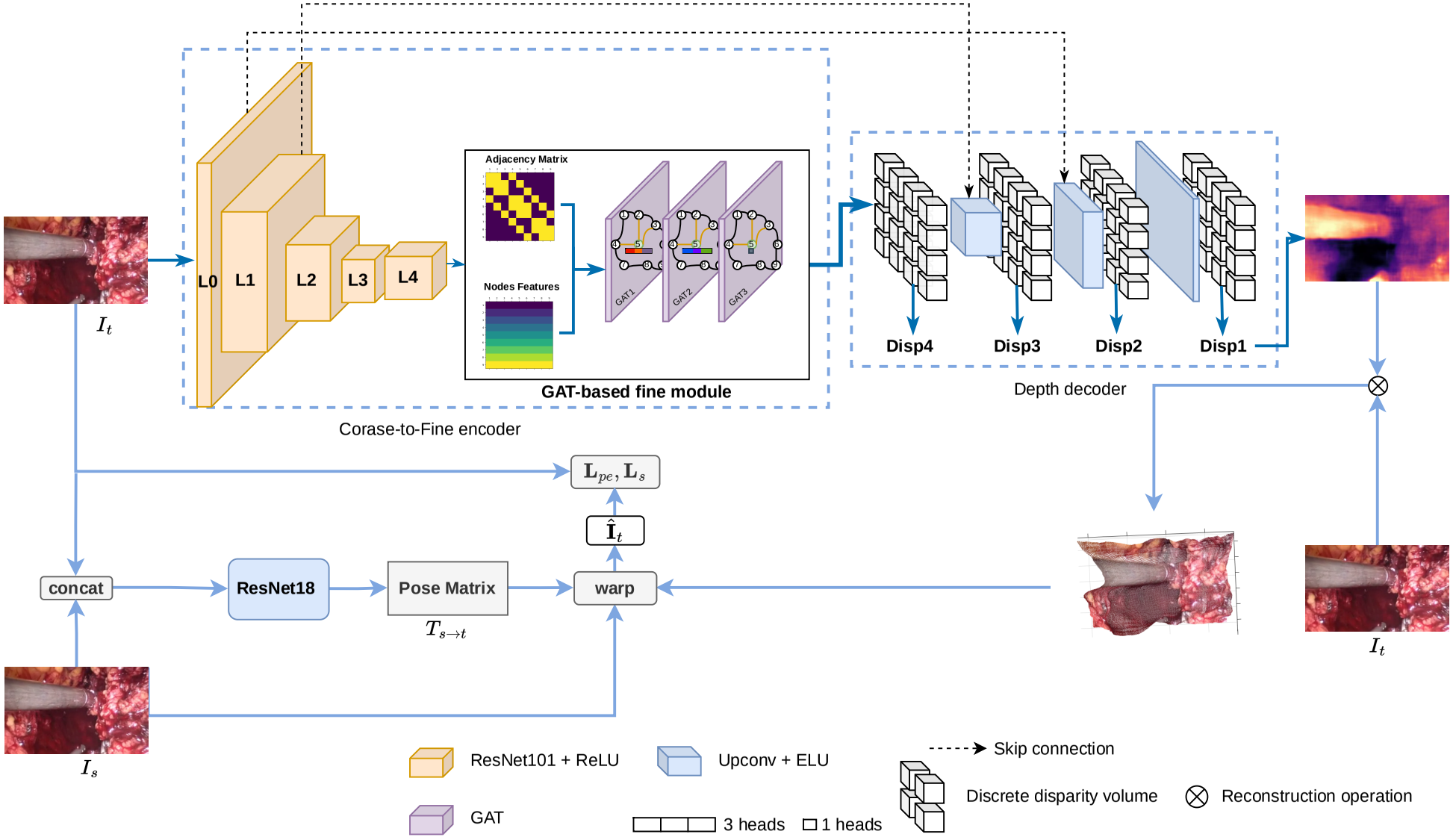


Fig. 4.2 Overview of the self-supervision depth estimation pipeline. The depth network inputs the I_t and outputs depth maps and a point cloud. The coarse-to-fine encoder includes Resnet and GAT-based fine module; Details are shown in Table 4.1 and Table 4.2. The depth decoder is a multi-scale decoder [4], [5]. The pose net based on Resnet-18 takes $[I_t, I_s]$ as input and outputs a relative camera pose $T_{s\to t}$. The outputs of the depth decoder and a pose net are used to warp the I_s to reconstruct the target image$\hat{I}_t$, and the L_{pe} and L_s loss are used to train the depth network.

Table 4.1 The network architecture of the Coarse encoder: K is the number of block repetitions, S is the stride, Chn is the number of output channels, input corresponds to the input channel of each layer, and "-" is without activation function

Layer	K	S	Ch	Input	Activation
Conv1 3×3 L0	1	1	64	image(320×192×3)	ReLU
Conv2 3×3 L0	1	1	64	Conv1(320×192×64)	ReLU
Conv3 3×3 L0	1	1	128	Conv2(160×96×64)	ReLU
. Maxpooling L0	1	2	128	Conv3(160×96×128)	ReLU
ResNet-101 L1	3	1	256	Maxpooling(160×96×128)	—
ResNet-101 L2	4	2	512	L1(80×48×256)	ReLU
ResNet-101 L3	23	1	1024	L2(40×24×512)	ReLU
ResNet-101 L4	3	1	2048	L3(40×24×1024)	ReLU

Therefore, the fine encoder usually includes two main parts: generating graphs and generating graph neural networks. In the former, pixel connectivity is used to generate the adjacency matrix and the features generated by the coarse part as node features. Compared with the random adjacency matrix [6], the adjacency matrix based on pixel connectivity can retain image position information. This is because there is a high correlation and continuity between the depth of a pixel and its neighbours. Algorithm 1 shows details about how to generate the adjacency matrix, and a four connectivity is used for our graph network. For the latter, the proposed model is based on the graph attention network [45] that is suitable for different rankings of neighbours and can assign different importance to nodes of the same neighbourhood. Therefore, the graph attention at the l layer can be defined as follows:

Algorithm 1: four_connectivity(rows, cols)

Data: rows, cols
Result: AdjacencyMatrix

$num\_nodes \leftarrow rows \times cols$;
$AdjacencyMatrix \leftarrow \text{zeros}(num\_nodes, num\_nodes)$;
for $i \leftarrow 0$ **to** $rows - 1$ **do**
  for $j \leftarrow 0$ **to** $cols - 1$ **do**
    $nodeIdx \leftarrow i \times cols + j$;
    if $j + 1 < cols$ **then**
      $rightIdx \leftarrow nodeIdx + 1$;
      $AdjacencyMatrix[nodeIdx][rightIdx] \leftarrow 1$;
      $AdjacencyMatrix[rightIdx][nodeIdx] \leftarrow 1$;
    if $i + 1 < rows$ **then**
      $bottomIdx \leftarrow nodeIdx + cols$;
      $AdjacencyMatrix[nodeIdx][bottomIdx] \leftarrow 1$;
      $AdjacencyMatrix[bottomIdx][nodeIdx] \leftarrow 1$;
$AdjacencyMatrix \leftarrow add\_selfloop(AdjacencyMatrix)$;
return *AdjacencyMatrix*;

Table 4.2 The network architecture of fine encoder part. K is the number of block repetitions, S is the stride, H is the number of heads, Chn is the number of output channels, input corresponds to the input channel of each layer, and "-" is without activation function

Layer	K	S	H	Ch	Input	Activation
Conv1 1×1	1	1	-	128	L4(40×24×1024)	–
GAT1	-	-	3	64	Conv1(40×24×128)	ReLU
GAT2	-	-	3	32	GAT1(40×24×64)	ReLU
GAT3	-	-	1	32	GAT2(40×24×32)	ReLU
Conv2 1×1	1	1	-	128	GAT3(40×24×32)	–

.

$$h_v^{(l)} = \sigma(\frac{1}{K}\sum_{K=1}^{K}\sum_{u\in N(u)\cup v}\alpha_{vu}^{(k)}W^{(l)}h_u^{(l-1)}) \tag{4.1}$$

where $h_u, \in \mathbb{R}^{F_{in}}$ and $h_v, \in \mathbb{R}^{F_{out}}$ are the input and output node features(F is the number of nodes features). $W \in \mathbb{R}^{KF_{out}}$ is the weight matrix. K is the number of heads ($K = 3$ for all layers). The equation in Eq. 4.1 remains permutation-equivariant because the additional operation is a summation. $\sigma(\cdot)$ denotes the non-linear activation function, which in this case is the ReLU. At last, α_{vu} is an attention score defined as:

$$\alpha_{vu} = softmax_j(e(h_v,h_u)) = \frac{exp(e(h_v,h_u))}{\sum_{u'\in N(u')\cup v} exp(e(h_v,h_{u'}))} \tag{4.2}$$

where $e(h_v,h_u)$ is an edge scoring function. These scores are normalized across all neighbours by the softmax function, and the edge scoring function is defined as:

$$e(h_v,h_u) = a^\top \text{LeakyReLU}(W\cdot[h_v \parallel h_u]) \tag{4.3}$$

where a and W are the weight matrix, and $\parallel$ is vector concatenation.

Note that F_{in} is latent features extracted from the coarse part in the first layer. The parameters of each layer used in our fine encoder part are described in Table 4.2. It includes two 1×1 convolution layers, two graph hidden layers and an output graph projection layer.

4.2.2 Depth Decoder and Pose Estimation

In the depth decoder, a discrete disparity volume is used to help extract depth information, which can improve depth estimation robustness and sharpness [60]. Specifically, the discrete disparity volume at r resolution can be written as follows:

$$\delta(D_r) = \sum_{c=1}^{Chn-1} softmax(D_r)\times tensor(\alpha + \frac{(\beta-\alpha)\times c}{Chn-1}) \tag{4.4}$$

where D_r is disparity values at different resolutions, Chn is the number of channels, and the function $tensor(\cdot)$ generates a tensor with the same value at all positions. α is 0.01, and β is 1 in our experiment. At the start of the process, the r is 40×24, and the Chn

Table 4.3 The network architecture of fine encoder part. K is the number of block repetitions, S is the stride, Chn is the number of output channels, and input corresponds to the input channel of each layer, Upconv consists of a 3×3 convolution and a nearest-neighbour upsampling that factor is 2, Outconv consists of Batch normalization and a 3×3 convolution, Disp is the disparity of output that is obtained by Eq. 4.4, and "-" is without activation function

Layer	K	S	Ch	Input	Activation
Disp4	1	1	1	GAT3(40×24×128)	Softmax
Upconv1 3×3	1	1	64	GAT3(40×24×128)	ELU
Conv1 3×3	1	1	64	Upconv1(80×48×64),ResNet-101 L1	ELU
Outconv1 3×3	1	1	128	Conv1(80×48×64)	–
Disp3	1	-	1	Outconv1(80×48×128)	Softmax
Upconv2 3×3	1	1	64	Conv1(80×48×64)	ELU
Conv2 3×3	1	1	64	Upconv2(160×96×64),Conv3 L0	ELU
Outconv2 3×3	1	1	128	Conv2(160×96×64)	–
Disp2	1	1	1	Outconv2(160×96×128)	Softmax
Upconv3 3×3	1	1	32	Conv2(160×96×64)	ELU
Conv3 3×3	1	1	32	Upconv3(320×192×32)	ELU
Outconv3 3×3	1	1	128	Conv3(320×192×32)	–
Disp1	1	1	1	Outconv3(320×192×128)	Softmax

is 128. Similar to [5], [4], the 40×24 resolution is up-sampled to multi-resolutions by the nearest neighbour method, and the decoder network details are shown in Table 4.3. The computational process, based on group equivariance, is achieved by generating a topological depth graph at multiple scales, allowing the network to capture both local and global features of the scene.

The pose estimation network is an essential component of our model, as it provides accurate estimates of the relative transformation between two images recorded at different time steps as follows:

$$T_{t\to s} = \Phi_p(I_t, I_s) \tag{4.5}$$

where Φ_p is pose network that receives a pair of images, I_t and I_s, the output of Φ_p is a rigid transformation matrix $T_{t\to s}$, which include a rotation matrix and translation vector. In our model, the standard ResNet-18 blocks [97] are used to pose encoder, and the pose decoder is the same to [4].

4.3 Self-supervised Monocular Depth Training

The adoption of a self-supervised depth estimation approach in Minimally Invasive Surgery(MIS) and Unmanned Aerial Vehicle(UAV) videos is driven by its ability to capitalize on large amounts of unlabeled data, enabling the model to autonomously learn depth information without the need for explicit depth annotations. This approach proves particularly advantageous in the medical domain, where obtaining precisely labelled depth data for training can be challenging due to the intricate nature of surgical procedures and

the associated ethical considerations. In addition, obtaining accurate depth ground truth data for UAVs can be challenging due to limited sensors and payload constraints. However, practical applications need other sensor data to fine-tune their predictions accurately, particularly for MIS.

Self-supervised monocular depth estimation that uses a single colour input I_t and relative transformation matrix $T_{t\to s}$ to reconstruct a depth map D_t. This transformation can be described in Eq. 3.16 in Chapter 3. A per-pixel correspondence can be established between any point p_t in the target image I_t and a corresponding point p_s in the source image I_s by

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

where K denotes the camera intrinsic matrix. Then, the $\hat{I}_t$ can be reconstructed from I_s through the differentiable bi-linear sampling operation $\hat{I}_t = s(I_s, p_s)$ [116]. Similarly to [4], L1-norm and SSIM [117] are applied as photo-metric error defined:

$$pe(I_t, \hat{I}_t) = \frac{\alpha}{2}(1 - SSIM(I_t, \hat{I}_t)) + (1-\alpha)\left\|I_t - \hat{I}_t\right\|_1 \tag{4.7}$$

where α is set to 0.85 in all experiments. The SSIM term is not particularly sensitive to uniform biases [118], which can lead to changes in brightness or shifts of colours. The L1-norm term can preserve colours and luminance. This error is weighted equally regardless of the local structure and does not produce quite the same contrast as SSIM. In this research, we combine both error functions to capture the best characteristics.

To address the depth of ambiguity, an edge-aware smoothness term [68] is used to enforce smoothness in depths,

$$L_s = \left|\partial_x D_t\right| e^{-\left|\partial_x I_t\right|} + \left|\partial_y D_t\right| e^{-\left|\partial_y I_t\right|} \tag{4.8}$$

where ∂_x and ∂_y are image gradients along horizontal and vertical axes. In self-supervised monocular depth estimation, assumptions of a moving camera and a static scene are unmet, where the camera motion may be small, and the scene may be dynamically changing in real-time. This can result in the prediction of inaccurate depth maps. Therefore, the auto-masking of stationary points [4] is utilized, which masks out areas of the image where the camera motion is small, and the scene is relatively static. As a result, this method can prevent the model from being influenced by these static areas without camera motions and improve the accuracy of the predicted depth maps. This mask can be defined as:

$$\mu = \left[\min_s(pe(I_t, \hat{I}_t)) < \min_s(pe(I_t, I_s))\right] \tag{4.9}$$

where $[\cdot]$ is the Iverson bracket. The photo-metric error combined with auto-masking can be rewritten as

$$L_{pe} = \frac{1}{N_R}\sum_{r\in R} \min_s(\mu_r * pe(I_t^r, \hat{I}_t^r)) \tag{4.10}$$

where N_R is the number of multi-resolutions, and its value is 4 in our experiment, such as $R = [(320,192),(160,96),(80,48),(40,24)]$. In summary, the final loss L is combined with photo-metric error in Eq. 4.10 and an edge-aware smoothness loss in Eq. 4.8:

$$L = L_{pe} + \lambda L_s \tag{4.11}$$

where λ is the weight parameter, and its value is 0.0015 similar to [4], we empirically found that this weighting parameter offers a desired balance between sharpness and overall structure correctness of the depth prediction.

4.4 Discussion

The proposed novel deep-learning architecture based on group equivariance proposed in this chapter is that the graph structure based on the four-connectivity of pixels in images can keep the original neighbour point information. Compared with the random adjacency matrix [6], the adjacency matrix based on pixel connectivity can retain image position information. Since every graph has the same structure and one-step network, the proposed can reduce the training time compared with other GNN-based methods [6] [114]. In addition, the mathematical background of group equivariance deep learning in Chapter 3 is introduced to support the proposed novel self-supervised depth estimation framework, which can learn intrinsic features of point clouds. Thus, the proposed group equivariance graph network can improve the depth details recovered from monocular images and videos [4] [5], such as endoscopy videos [72] and videos captured by UAVs [8].

4.5 Summary

This chapter presents a novel self-supervised monocular depth estimation framework based on group equivariance deep learning described in Chapter 3. The coarse-to-fine encoder architecture can learn non-Euclidean information and refine the depth geometry feature. In addition, the graph structure based on four-connectivity can keep the original neighbour point information. In the following chapters, two use cases will be used to demonstrate the proposed depth estimation framework.

Chapter 5

Topology-aware Depth Estimation from Endoscopy Videos

5.1 Motivation

This chapter presents a case study of topology-aware depth estimation from endoscopy videos. Based on the proposed novel self-supervised depth estimation framework, this case study aims to demonstrate the effectiveness of the proposed method and the group equivariance designed to learn non-Euclidean information and intrinsic features of point clouds. *DaVinci* dataset, which includes many instruments and complex depth, is used for qualitative evaluation on point clouds. Furthermore, several public endoscopy datasets are used to compare with other state-of-art methods, including SCARED, SERV-CT and Hamlyn heart datasets.

Depth estimation from monocular endoscopy plays a crucial role in the context of Minimally Invasive Surgery (MIS), enabling 3D reconstruction, surgical navigation, and AR patient-specific data visualization. However, there remain some challenges. The complex nature of the MIS surgical environment, coupled with the featureless surface representations, makes it challenging to estimate depth accurately, particularly for the task of localization of instruments, which is crucial for surgical navigation. Furthermore, the ground truth is difficult to obtain for supervised deep learning methods.

5.2 Introduction

Depth estimation is a common task of robotic vision and visual computing, involving the prediction of the depth or distance of objects in a scene from a 2D image or a video. This can be done either using a single image (monocular depth estimation) or multiple images (stereo or multi-view depth estimation). The output of depth estimation can be used for a variety of applications, including 3D reconstruction, object detection, scene understanding, and AR [119]. Depth estimation also finds its use in medical imaging and surgery, such as robotic surgery. In surgical settings, depth estimation can assist navigation and improve

the understanding of surface anatomy, particularly in MIS, where 2D endoscope images may compromise depth perception [120].

Various methods have been proposed for reconstructing 3D structures or estimating depth from endoscopic images [121], [122], [123]. Shape from Shading (SfS) [124] is one such method that estimates 3D surface geometries from observed image shading patterns caused by surface normals. Feature-point matching techniques, such as structure from motion (SfM) and visual odometry (VO), have also been applied to endoscopic images for depth estimation [125], [126]. However, these approaches may not perform well in estimating depths from real endoscopic images. The reason is that endoscopic images can capture a wide range of organ surface textures, and organs depicted in these images often exhibit non-rigid deformations. As a result, the accuracy of feature point matching in the images may decrease, failing in estimating depths.

In recent years, deep learning methods have emerged as a promising approach for predicting depth maps from monocular videos. These methods can be broadly classified into two categories: supervised learning [35], [36], [127], and unsupervised learning [72], [4], [71]. The former requires ground-truth depth maps to be available during the model training process. It is often difficult to collect a large-scale and accurate ground-truth depth of endoscopic scenes due to several challenges, such as sensor noise, limited field of view, and varying lighting conditions [72]. This chapter proposes an unsupervised framework without relying on real-depth maps as the ground truth for model training and learning. The proposed self-supervised deep-learning architecture based on GNN in Chapter 4, for the first time, is applied to estimate the depth of endoscopy videos. In this chapter, three public datasets are evaluated to demonstrate the ability of the proposed framework to obtain fine details in endoscopy depth estimation, including surgical instruments, which are also measured with greater accuracy compared to five state-of-the-art methods. In addition, to improve the performance of the proposed framework in low-illumination endoscopy videos, a new loss function based on the CLAHE algorithm is introduced.

5.3 Evaluation on Endoscopy Video

Several experiments are conducted to evaluate the different aspects of the proposed approach. In particular, the proposed method is evaluated on *DaVinci* datasets [128] and three public datasets, SCARED [129], SERV-CT [130] and Hamlyn[1]. The ground truth from SCARED datasets is used for quantitative and qualitative analysis. Ablation studies based on SCARED datasets are discussed to give a more detailed analysis of the proposed method.

5.3.1 Experiment Setup

The proposed depth estimation network is based on the public deep-learning platform PyTorch framework [131]. All methods are trained on a single NVIDIA GeForce RTX

[1]http://hamlyn.doc.ic.ac.uk/vision/

2080 GPU for a batch size of 5 and 20 epochs. The Adam [132] is used to optimize both the pose and depth net, and the learning rate is $1e^{-4}$ at the beginning. This rate will be dropped to $1e^{-5}$ after 15 epochs. The pre-trained ResNet-18 is used for the PoseNet, but the coarse encoder (CNN) part is trained without a pre-trained model, which can achieve more accurate results for endoscopy images. Since the raw resolution of *DaVinci* datasets is 384×192, the resolution of images is set to 320×192 for all datasets.

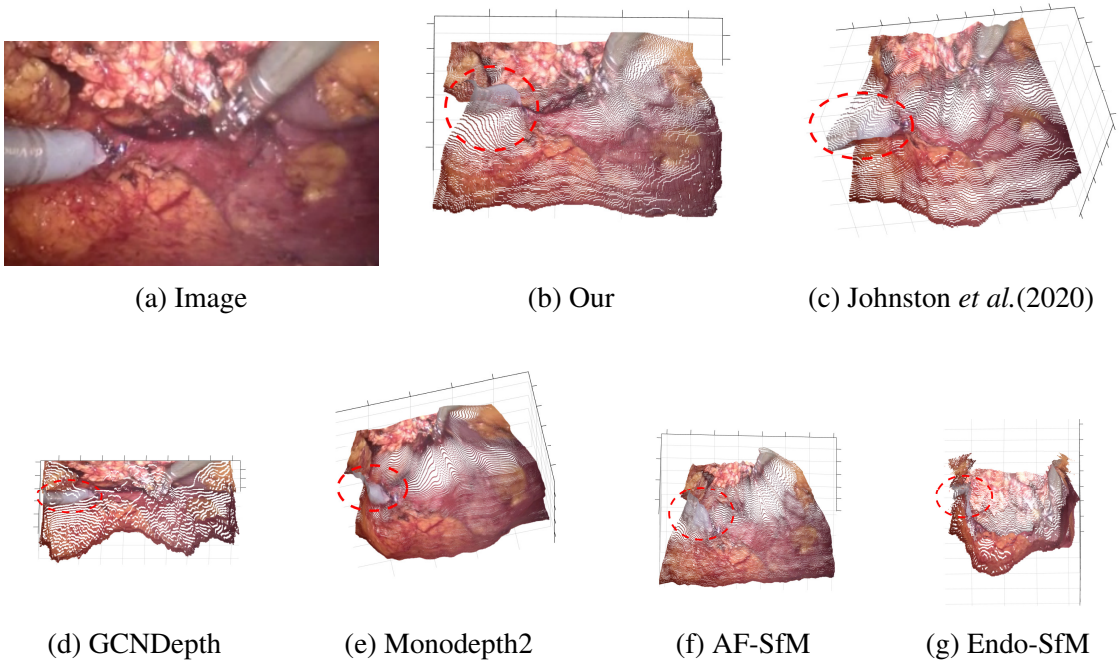
5.3.2 *DaVinci* Datasets Results

The *DaVinci* datasets contain recordings of minimally invasive surgeries, which include detailed information on surgical tools. The *DaVinci* datasets were released by the Imperial College of London into the public domain and are freely available for research communities. There are no ethical concerns for this PhD work to use these datasets. These datasets are well-suited for evaluating the proposed method, as they provide a realistic and challenging environment for testing the performance of the proposed approach. Additionally, the accurate localization of surgical instruments is useful for various applications, such as augmented reality (AR) in surgical navigation systems. In this experiment, the left images were used as the training data, $70\% - 15\% - 15\%$, consisting of 34240 images for training, 7191 images for validation, and 7191 for testing. The depth estimation performance of the proposed framework is evaluated against five state-of-the-art self-supervised methods, including AF-SfM [72], GCNDepth [6], Monodepth2 [4], Johnston *et al.*(2020) [5] and Endo-SfM [71]. Since an endoscope (the camera) is usually close to an organ and neighbouring objects have a similar depth or small depth difference, the point cloud is used to show results. Moreover, the point cloud results are highly versatile and can be easily applied in various applications.

As an illustration, Fig. 5.1 demonstrates the aforementioned results, indicating that our model performs better on surgery instruments and exhibits a more regular distribution. The results obtained by AF-SfM, GCNDepth, Monodepth2, and Endo-SfM suggest that surgical instruments tend to have a planar shape with similar depth values as the background. Meanwhile, when applying the method proposed by Johnston *et al.* [5]to the *DaVinci* dataset, the resulting depth values on the surgical instruments are non-planar, and the orientation of the depth values is incorrect. However, our method can correctly recognise the depth of surgical instruments, and the distribution of the top-point cloud closely resembles the actual depth. Fig. 5.3 shows more examples of various methods, and our method can handle the details of the surgical instruments.

5.3.3 Quantitative Evaluation

Since the *DaVinci* dataset is a real surgery video without ground truth, the SCARED, SERV-CT and heart of Hamlyn datasets are used to evaluate the proposed framework. The SCARED dataset contains nine stereo videos, and the ground truth was captured using a structured light camera. The 22,922 left images are collected from the first seven videos with ground truth data. These images are divided into 16,046 for training,



(a) Image (b) Our (c) Johnston *et al.*(2020)

(d) GCNDepth (e) Monodepth2 (f) AF-SfM (g) Endo-SfM

Fig. 5.1 Comparison of point cloud results on *DaVinci* dataset. The proposed framework perform better on surgical instruments, and the resulting distribution is more regular.

6,062 for validation, and 814 for testing purposes, where 70%, 25% and 5% are used for training, validation and testing, respectively. The model trained on the SCARED dataset is utilized for the Hamlyn and SERV-CT datasets due to their analogous data distribution and characteristics. This means that the common feature representations learned by the network, like organ surface. Another reason is the similar environmental context under human organs.

Regarding the quantitative evaluation, standard depth evaluation metrics are used, such as Absolute and Relative Error (Abs Rel), Squared Relative Error (Sq Rel), Root Mean Squared Error (RMSE), and Root Mean Squared Log Error (RMSE log). Additionally, $\delta < 1.25$ is used to determine the accuracy of the estimated depth using a threshold proposed in [133] and the details shown in Eq. 5.1. The predicted depth is multiplied with median scaling ($\hat{s} = median(D_{gt})/median(D_{pred})$) before the evaluation, which is introduced in [37].

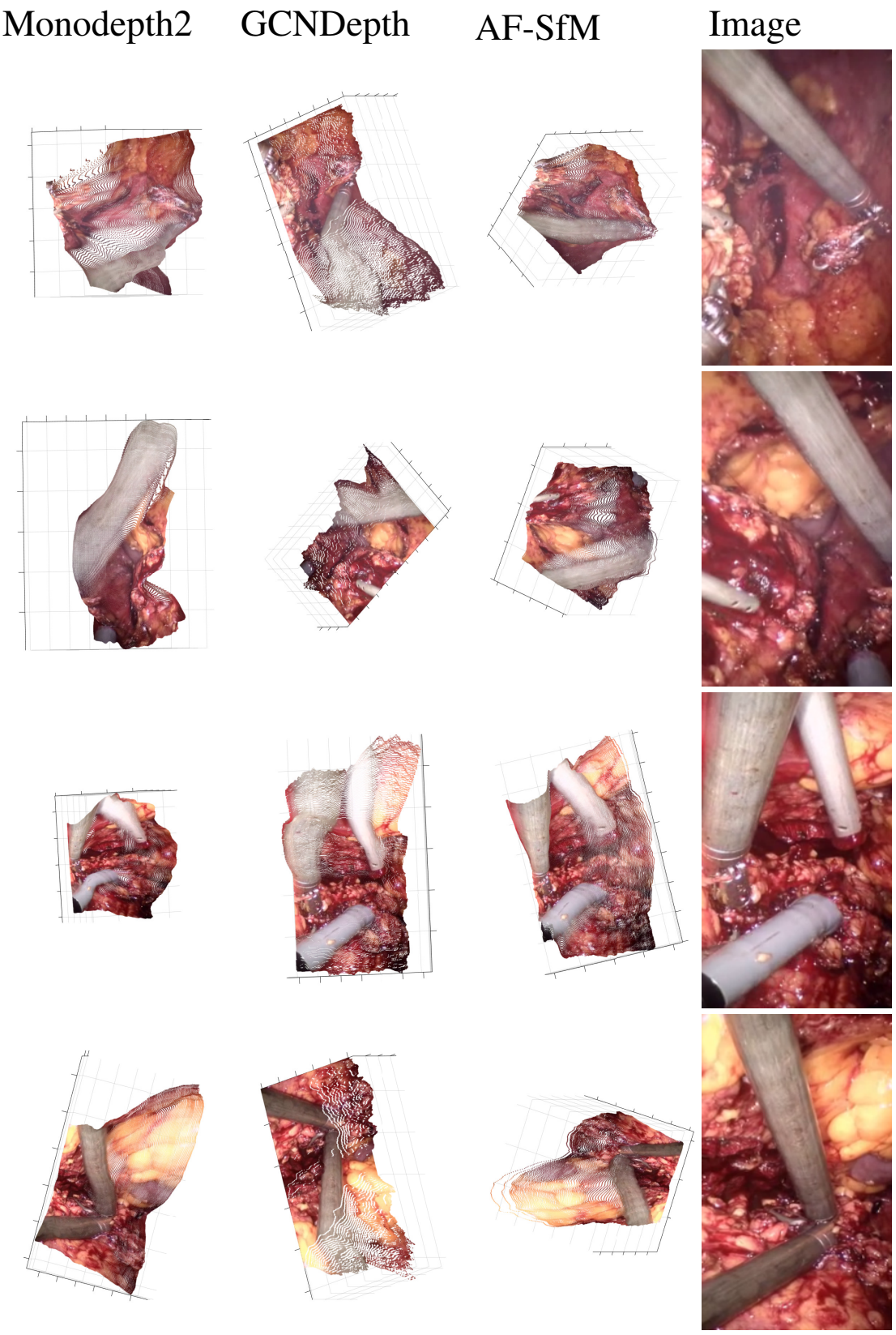

Monodepth2
GCNDepth
AF-SfM
Image

Fig. 5.3 Comparison of point cloud results on *DaVinci* dataset. Our models perform better on surgical instruments, and the resulting distribution is more regular.

$$
\begin{aligned}
\text{Abs Rel} &= \frac{1}{D}\sum_{d^*\in D} |d^* - d| / d^*, \\
\text{Sq Rel} &= \frac{1}{D}\sum_{d^*\in D} \|d^* - d\|^2 / d^*, \\
\text{RMSE} &= \sqrt{\frac{1}{D}\sum_{d^*\in D} \|d^* - d\|^2}, \\
\text{RMSE log} &= \sqrt{\frac{1}{D}\sum_{d^*\in D} \|log(d^*)\ log(d)\|^2}, \\
\delta &= \frac{1}{D}\left|\left\{d^* \in D | max(\frac{d^*}{d}, \frac{d}{d^*}) < 1.25\right\}\right|.
\end{aligned}
\tag{5.1}
$$

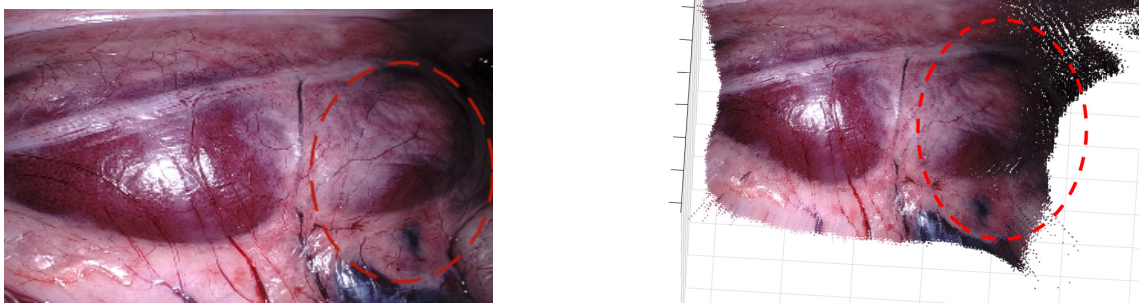
where d^* and d denote ground truth and the predicted depth maps, D indicates a set of valid ground truth depth values in one image, and $|.|$ returns the number of elements in the input set.

All methods are evaluated on SCARED [129], Hamlyn and SERV-CT [130] datasets. The results are shown in Table 5.1, and the Abs Rel metric is used for subsequent analysis. In general, the proposed method significantly outperforms other state-of-the-art methods and results show remarkable improvements in each evaluation metric. For SCARED and Hamlyn datasets, the proposed method achieves the highest performance. The Abs Rel metric is improved by $\sim 30.1\%$ and $\sim 7.14\%$ compared with the other five methods. While the GCNDepth model achieved better results than our model on the SERV-CT dataset, the proposed framework outperformed GCNDepth on other datasets, and the metrics show similar results between the proposed framework and GCNDepth.

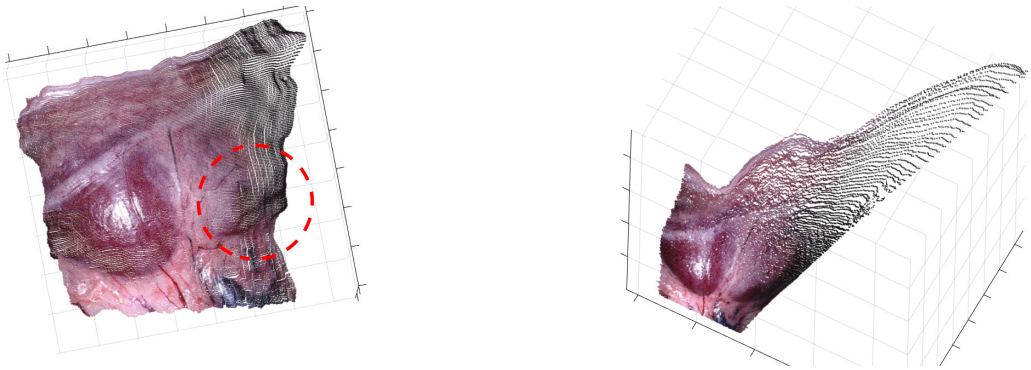
The qualitative results of the SCARED dataset are shown in Fig. 5.4 and Fig. 5.5. The proposed method is compared with five approaches, AF-SfM [72], GCNDepth [6], Monodepth2 [4], Johnston *et al.*(2020) [5] and Endo-SfM [71]. In general, The proposed method can produce fine detailed depth maps. In the first example (Fig. 5.4(a) is the image, and Fig. 5.4(b) is the point cloud ground truth). The result of AF-SfM, shown in Fig. 5.4(d), has an irregular point distribution on the edge. Although Fig. 5.4(e) shows details on the red circle region, it fails to reconstruct the top areas accurately. The results of Monodepth2 (shown in Fig. 5.4(f)) and Johnston *et al.*(2020) (shown in Fig. 5.4(g)) tend to flatten the compared ground truth. The Endo-SfM method cannot achieve the correct point cloud distribution. However, our method can accurately detect different depths in the red region, as shown in Fig. 5.4(c). In the second example, illustrated from Fig. 5.5(a) to Fig. 5.5(h), most of the methods can get the correct depth, but the GCNDepth fails to recover the depth in the red region, and the AF-SfM cannot capture the top areas depth. Overall, Fig. 5.4 and Fig. 5.5 demonstrate the effectiveness of the proposed framework in accurately detecting detailed point clouds.

Table 5.1 Quantitative results. Comparison of our method to existing methods on the SCARED dataset, Hamlyn datasets and SERV-CT. The best results in each category are in **bold**.

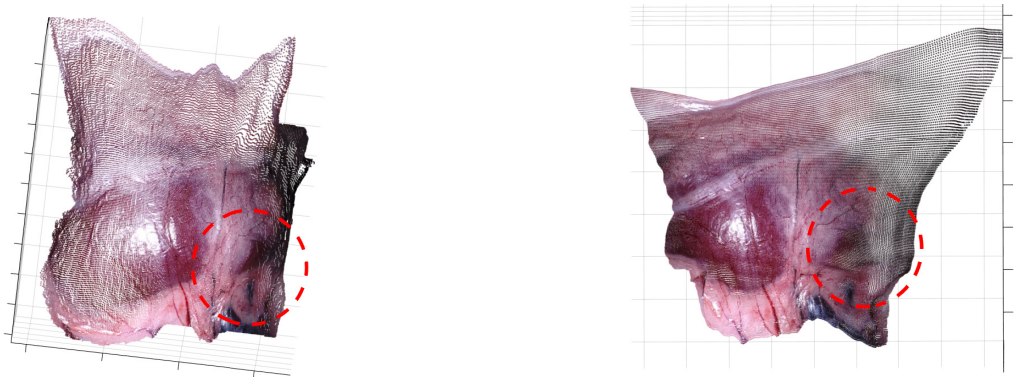
	Lower Better				Higher Better
Method	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$
SCARED dataset					
AF-SfM	0.093	0.995	6.315	0.116	0.913
GCNDepth	0.151	2.225	9.738	0.183	0.781
Monodepth2	0.060	0.412	4.202	0.081	0.967
Johnston *et al.*(2020)	0.062	0.471	4.432	0.085	0.965
Endo-SfM	0.145	1.989	9.917	0.193	0.774
Our	**0.058**	**0.364**	**4.063**	**0.077**	**0.971**
Hamlyn heart datasets					
AF-SfM	0.286	8.219	6.612	0.286	0.864
GCNDepth	0.271	**7.505**	6.237	0.276	0.888
Monodepth2	0.286	8.577	6.276	0.281	0.886
Johnston *et al.*(2020)	0.289	8.534	6.461	0.287	0.867
Endo-SfM	0.302	9.206	7.136	0.293	0.859
Our	**0.266**	7.550	**5.929**	**0.271**	**0.909**
SERV-CT dataset					
AF-SfM	0.091	0.901	7.129	0.112	0.931
GCNDepth	**0.068**	**0.510**	**5.351**	**0.083**	0.974
Monodepth2	0.096	0.947	7.116	0.116	0.910
Johnston *et al.*(2020)	0.082	0.723	6.361	0.103	0.946
Endo-SfM	0.172	2.943	13.232	0.230	0.680
Our	**0.068**	0.521	5.538	0.085	**0.976**



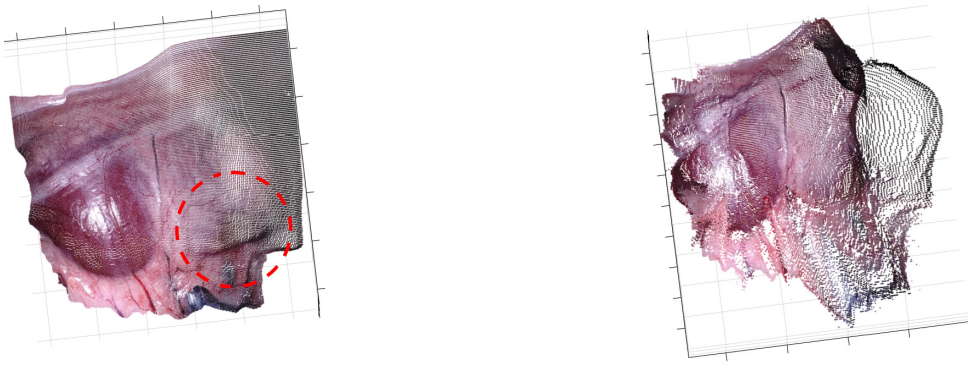
(a) Image (b) GT



(c) Our (d) AF-SfM

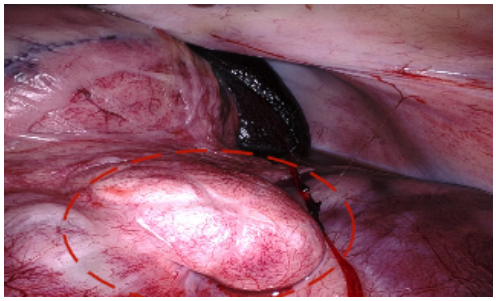


(e) GCNDepth (f) Monodepth2

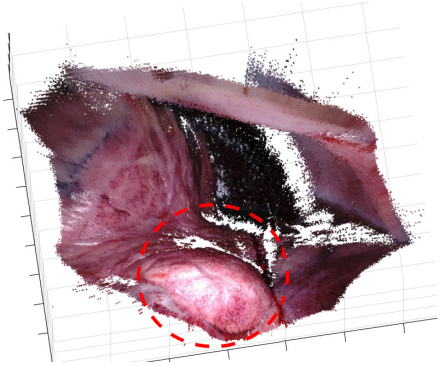


(g) Johnston *et al.*(2020) (h) Endo-SfM

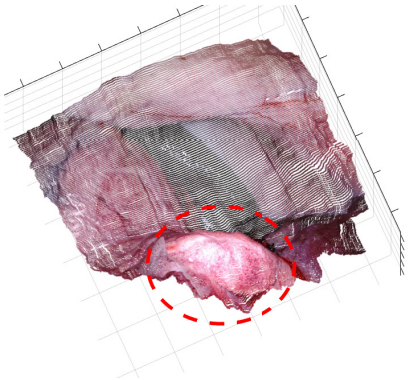
Fig. 5.4 Comparison of a point cloud result on the SCARED dataset. The proposed framework performs better on point cloud details, and the resulting distribution is more regular than others, especially for the red dash areas.



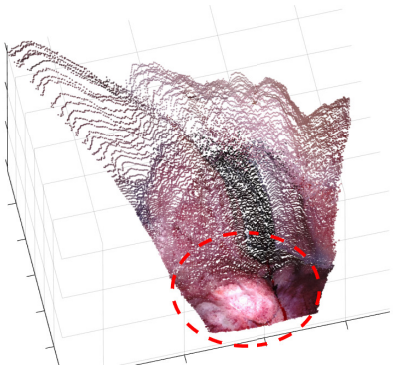
(a) Image



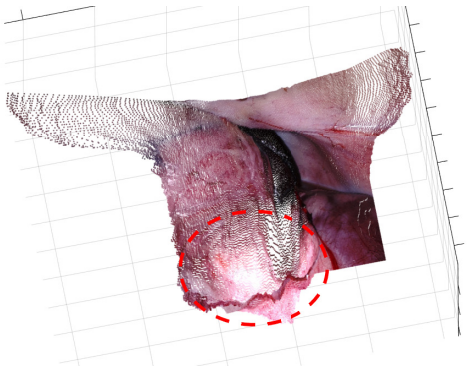
(b) GT



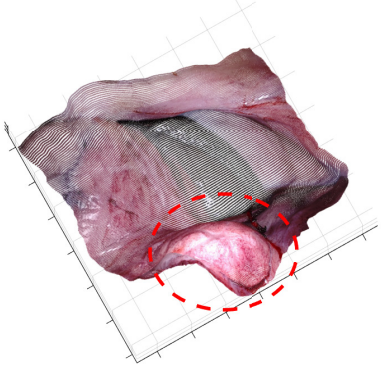
(c) Our



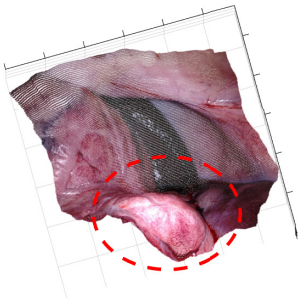
(d) AF-SfM



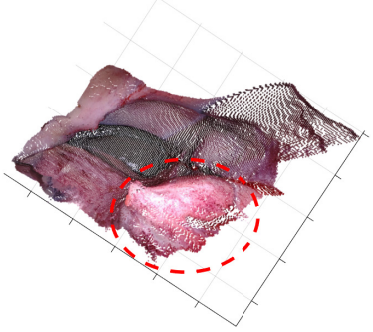
(e) GCNDepth



(f) Monodepth2



(g) Johnston *et al.*(2020)

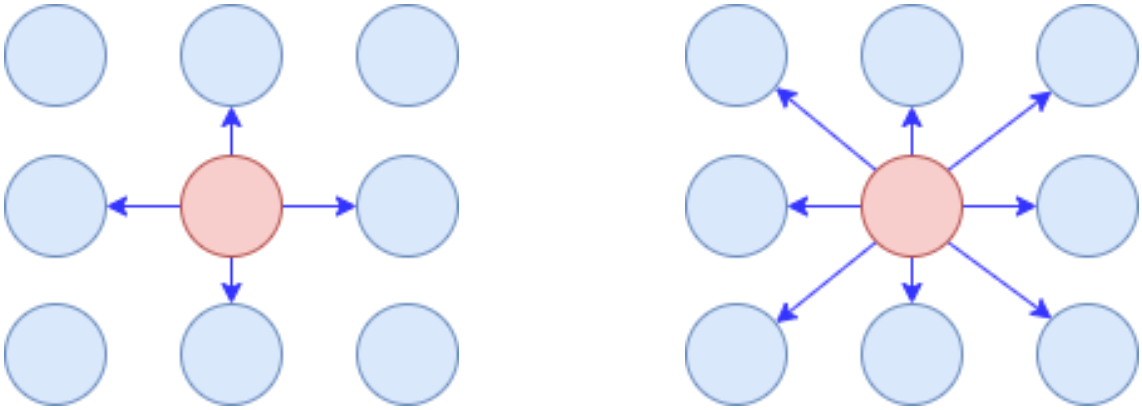


(h) Endo-SfM

Fig. 5.5 Comparison of another point cloud result on the SCARED dataset. The proposed framework performs better on point cloud details, and the resulting distribution is more regular than others, especially for the red dash areas.

Table 5.2 Ablation results for different components. **w/o GAT** represents without GAT network. The best results in each category are in **bold**.

Methods	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$
Baseline(Resnet-18 w/o GAT)	0.102	1.083	6.737	0.130	0.894
Ours-Resnet-18 w/ GAT	0.061	0.414	4.276	0.081	0.965
Ours-Resnet-50 w/o GAT	0.061	0.393	4.236	0.081	**0.971**
Ours-Resnet-50 w/ GAT	0.060	0.386	4.183	0.080	0.968
Ours-Resnet-101 w/o GAT	0.073	0.862	6.844	0.103	0.942
Ours-Resnet-101 w/ GAT	**0.058**	**0.364**	**4.063**	**0.077**	**0.971**



(a) Four-connectivity (b) Eight-connectivity

Fig. 5.6 (a) is a four-connectivity structure for one pixel, and (b) is an eight-connectivity structure for one pixel.

5.3.4 Ablation Study

To better understand the contribution of each component in the proposed framework to the overall performance during training, an ablation study is conducted by modifying different parts of the proposed framework, as shown in Table 5.2. The Resnet-18 only includes a discrete disparity volume encoder without a GAT network. Using ResNet-101 instead of ResNet-18 and ResNet-50 has improved the results and accuracy slightly. The baseline model, without any of the proposed contributions, performs the worst. However, when all components are combined, the proposed method observes a significant improvement in performance.

In addition, Table 5.3 shows different network framework results. The higher resolution 320×256 obtains higher errors than 320×192 under the same connectivity, and the four-connectivity method is more suitable for the proposed depth estimation network than eight-connectivity. The 4&8 means that there are two types of graphs, including four-connectivity and eight-connectivity, as shown in Fig. 5.6, whose leaned features are concatenated in the last layer of Table 4.2. The idea of random connectivity stems from GCNDepth, and the adjacent matrix of the graph is randomly generated. This method gets the highest Absolute and Relative Error than others. Since GAT does not incorporate edge features into the model, a graph attention layer that handles edge features from the

Rossmann-Toolbox [134]. The difference part compared with the GAT is in how the attention scores $e(h_v, h_u)$ are obtained:

$$e(h_v, h_u) = F(f'_{vu}) \tag{5.2}$$

$$f'_{vu} = \text{LeakyReLU}(A[h_v \parallel f_{vu} \parallel h_u]) \tag{5.3}$$

where f'_{vu} and f_{vu} are edge features, F is weight vector and A is weight matrix. Then, the node features h_v are updated, the same as the GAT. The initial value of f_{vu} is determined by computing the absolute value of the difference colour vector. The use of the absolute operation indicates that the graph is non-directional or bidirectional. In general, the results of the GAT with learned edge features tend to have higher errors compared with the proposed method without learned edge features. The results of 4-pixel connections are better than 8-pixel connections. One reason is that 4-pixel connections focus on a more localized context compared to 8-pixel connections. This can be advantageous in scenarios where capturing fine-grained details or features within a limited spatial range is crucial for accurate predictions. Another is that 4-pixel connections tend to exhibit lower sensitivity to noise or irrelevant information in the surrounding pixels. In summary, these results demonstrate that more pixel connectivity, more complex networks or more information may not work on the proposed monocular depth estimation network framework.

Table 5.3 Compared with other network frameworks. **Ours-256-4** represents the results of 320×256 resolution with four-connectivity, **Ours-192-8** represents the results of 320×192 resolution with eight-connectivity, **random** represents that the connectivity of edges is randomly generated, and **EGAT** represents that the GAT includes the feature of edges. The best results in each category are in **bold**.

Methods	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$
Ours-256-4	0.076	0.628	5.341	0.101	0.939
Ours-192-8	0.130	1.731	8.953	0.173	0.810
Ours-192-4&8	0.203	3.798	12.516	0.237	0.659
Ours-192-random	0.279	7.525	17.057	0.417	0.570
Ours-192-EGAT-4	0.114	1.322	7.675	0.148	0.853
Ours-192-EGAT-8	0.112	1.241	7.494	0.146	0.859
Ours-192-4	**0.058**	**0.364**	**4.063**	**0.077**	**0.971**

5.3.5 Discussion

A novel self-supervised framework is proposed for endoscopy monocular depth estimation. The coarse-to-fine encoder architecture can learn non-Euclidean information and refine the depth geometry feature. Moreover, the graph structure based on four-connectivity can keep the original neighbour point information. Qualitative and quantitative analysis of different datasets can demonstrate that the proposed framework can obtain fine details and improve accuracy. However, this method may fail to estimate the depth of images with low lighting. Low lighting conditions typically result in darker images with reduced visibility of details.

This can lead to the loss of important information and fine-grained textures in the image. This self-supervised framework is trained to learn patterns and features from input data. If an image has textureless areas due to low lighting, the network might find it hard to extract meaningful features and patterns from those regions. This can lead to poor performance on monocular depth estimation. Therefore, next, a self-supervised framework based on image enhancement is proposed for low-illumination endoscopy video.

5.4 Low-illumination Endoscopy Video

5.4.1 Challenges

The depth estimation on low-illumination Endoscopy video remains a challenging problem. Most existing works usually utilized thermal images to recover depth [135], [136]. Spencer *et al.* [137] proposed a DeFeat-Net to simultaneously learn a cross-domain dense feature representation and depth estimation to acquire more robust results. However, this method fails to tackle the low visibility. Vankadari *et al.* [138] considers this problem as a domain adaptation problem, where they trained a network on daytime data to work for nighttime images, called ADFA. Different from ADFA, Wang [139] utilized depth distribution from daytime data as regularization and directly learned depth information from nighttime scenes. However, both methods need corresponding daytime data or good-visible images, which are usually unavailable under MIS. Image enhancement aims to improve details in low-visibility regions. Retinex model-based methods allow images to be decomposed into illumination and reflectance [140], [141]. Recently, deep learning Retinex-based methods combined CNNs and Retinex to pursue better accuracy and robustness [142], [143]. Although these methods are effective, it is not suitable for depth estimation, which usually assumes training data with the same brightness among frames [4]. Therefore, like [139], the Contrast Limited Histogram Equalization (CLHE) algorithm [46] is used to enhance the visibility of endoscopy images and keep brightness consistency simultaneously. In this section, an novel loss function is proposed to improve depth estimation on low-illumination Endoscopy video based on the framework in Chapter 4.

5.4.2 Method

The framework is the same with Fig.4.2 in Chapter 4, and it is different on the SSIM and L1-norm loss for self-supervised training. It is noticed that the CLHE is only used to compute the photo-metric loss in Equ.4.7, and the SSIM loss can be rewritten as follows: $\hat{I}_t = s(I_s, p_s)$

$$pe(I'_t, \hat{I}_t{}') = \frac{\alpha}{2}(1 - SSIM(I'_t, \hat{I}'_t)) + (1-\alpha)\left\|I'_t - \hat{I}'_t\right\|_1 \tag{5.4}$$

where α is also set to 0.85 in all experiments, $I'_t = m(I_t), \hat{I}_t{}' = m(\hat{I}_t) = s(m(I_s), p_s)$, m is brightness mapping function. The main parts of computing m are as follows [139]: Assuming we have the frequency distribution $f_a = h(a)$ for the input image, where f_a

represents the frequency of brightness level b. Then, the frequencies exceeding the parameter $\sigma = 0.003$ are clipped to prevent noise signal amplification, and the clipped frequencies are evenly distributed across each brightness level. Finally, m can be obtained by:

$$m(a) = \frac{cdf(a) - cdf_{min}}{cdf_{max} - cdf_{min}}(L-1) \tag{5.5}$$

where cdf is the cumulative distribution, cdf_{min} and cdf_{max} are the minimum and maximum of cdf. L is the number of brightness levels, and its values usually are 256 for colour images.

5.4.3 Evaluation

Apart from the SSIM loss, other modules are the same as mentioned before. Since the CLHE is only used to enhance the image when computing the SSIM loss and does not alter the input of the networks, the image can be input into the trained model without the enhancement. Since there are no available low-illumination endoscopy datasets with ground truth, only qualitative analysis is conducted. In addition, there are no frameworks, which focus on depth estimation with a low-illumination condition, or consider all images with a low-illumination condition [72]. Therefore, the five state-of-the-art self-supervised methods are used to evaluate low-illumination endoscopy videos, including AF-SfM [72], GCNDepth [6], Monodepth2 [4], Johnston *et al.*(2020) [5] and Endo-SfM [71]. This experiment uses several videos with low-illumination conditions in the Hamlyn dataset. There are 15294 images for training with the proportion$70\% - 25\% - 5\%$, 6350 images for validation, and 400 images for testing.

The qualitative results of the low-light endoscopy dataset are shown in Fig. 5.8, Fig. 5.9 and Fig. 5.10. In Fig. 5.8, a big instrument is used under a low-illumination scene. The results of Johnston *et al.*(2020)(Fig. 5.8(d)), Our-woIE(Fig. 5.8(c)) and AF-SfM(Fig. 5.8(g)) show that the instrument tends to be blended together with organs. The results of GCNDepth(Fig. 5.8(e)) and Endo-SfM (Fig. 5.8(h))show a big deformation at the edges. In addition, Fig. 5.8(f) also show the failure to recover depth on the instrument. However, Fig. 5.8(b) with the new SSIM loss function achieve better results than others, and there is a convex surface in the position of the instrument. In Fig. 5.9, there are without any instrument. The region of the red circle should have different heights, from high to low. The results of GCNDepth(Fig. 5.9(e)) and Endo-SfM (Fig. 5.9(h))show there is a big deformation at the edges. Fig. 5.9(c) and Fig. 5.9(f) show the wrong depth on the right side of the image. Fig. 5.9(g) tends to show the same depth on both sides. Lastly, Although Fig. 5.9(b) and Fig. 5.9(d) have correct depth, the former has a good distribution. In the third experiment, shown in Fig. 5.10, there is a small instrument in the middle. Similar to before, GCNDepth and Endo-SfM have bed results. AF-SfM and Johnston *et al.*(2020) show that the instrument tends to be together with the organ and have the same depth values. The result of Our-wIE is better on the instrument compared with Fig. 5.10(c) and Fig. 5.10(f). In general, The proposed method with the new SSIM loss can produce fine

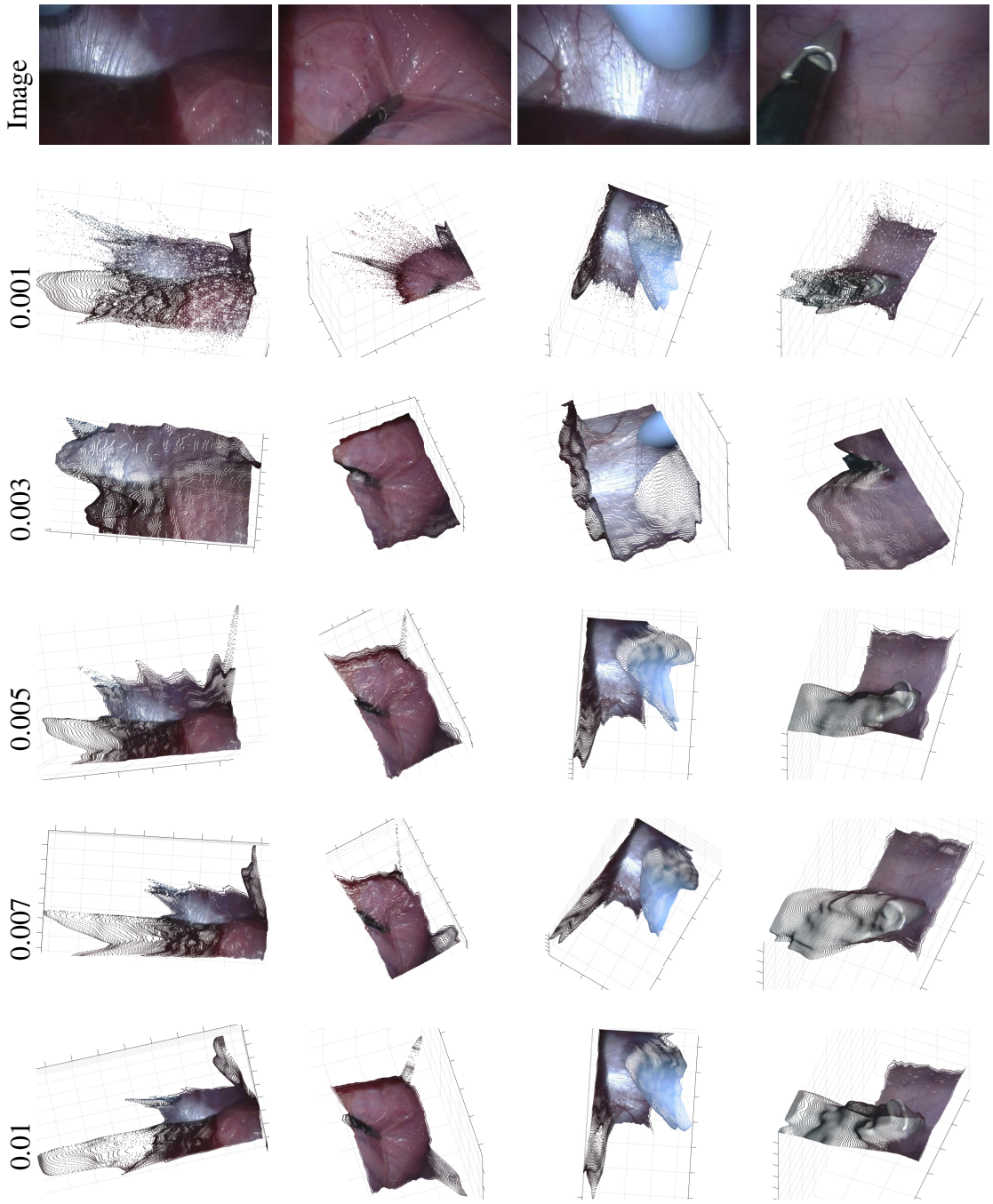
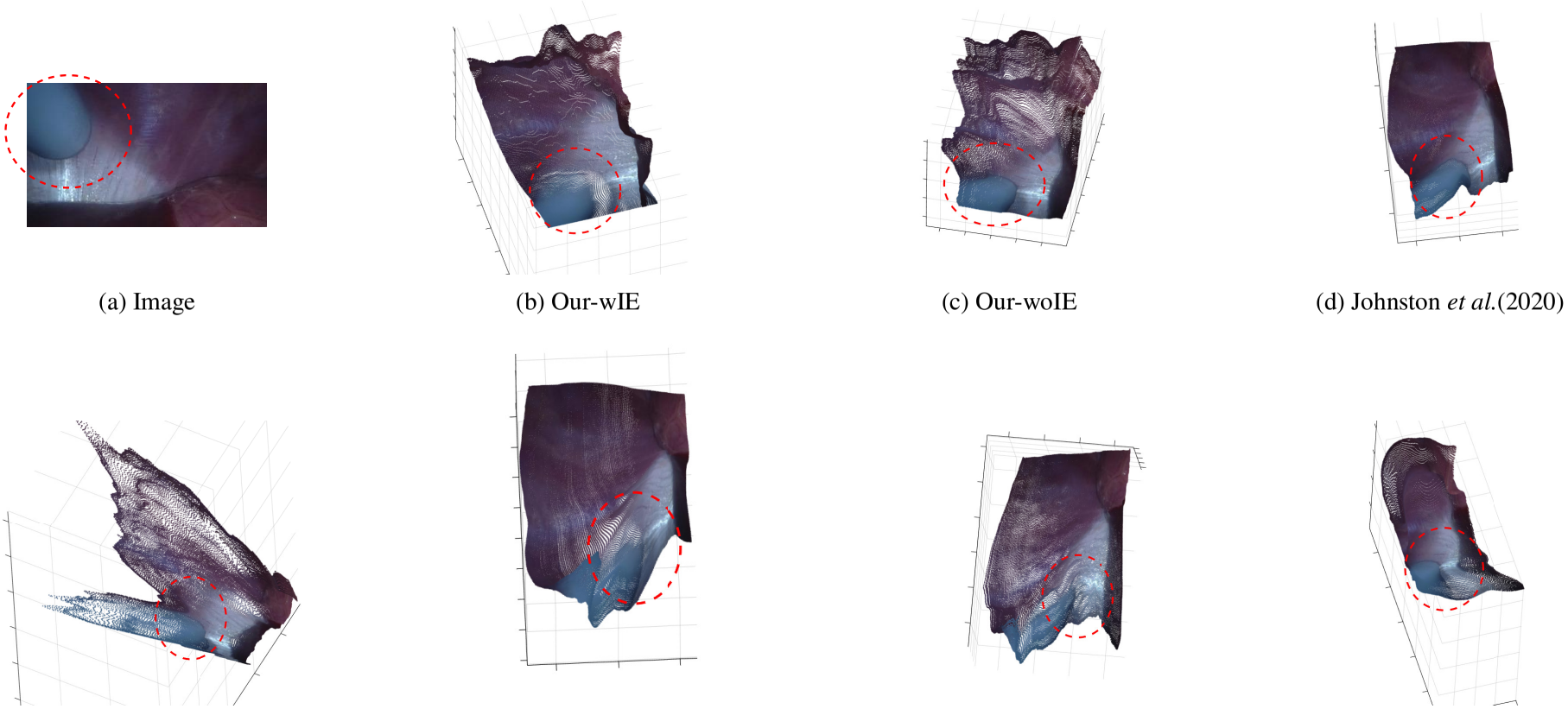


Fig. 5.7 Comparison of different σ results on low-illumination endoscopy video.

detail depth maps. However, there are some disadvantages. For example, the edges of too dark remain fail to estimate the depth. In addition, the value of σ will have a significant influence on the Low-illumination image depth estimation. Therefore, Fig. 5.7 shows different values of σ, and it can obtain better results in $\sigma = 0.003$. A lower value of σ (less than 0.003) tends to introduce more noise, as illustrated in the second row of Fig. 5.7. A higher value of σ shows irregular edges and increases the scale of instruments.

5.5 Conclusion

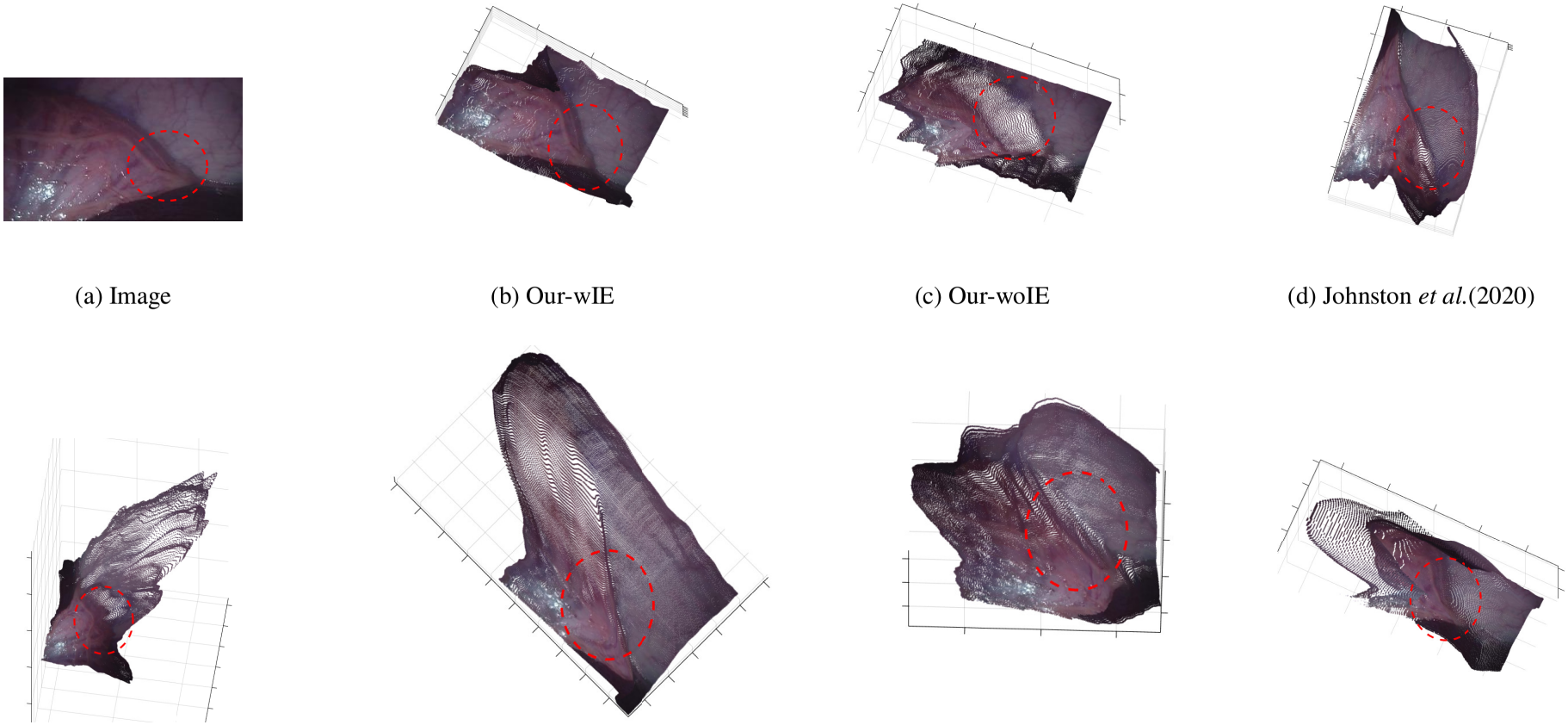
This chapter uses the endoscopy video to evaluate the proposed depth estimation framework. In the quantitative experiment, the proposed framework achieves better results in the SCARED and Hamlyn heart datasets than the five state-of-the-art methods and has comparable results in SERV-CT compared with GCNDepth. In the qualitative experiment, the proposed framework can sense the depth of surgical instruments accurately and achieve well-distribution around the edge of point clouds. Finally, the new loss function for handling low-illumination endoscopy images can improve the performance of depth estimation under poor illumination conditions.



(a) Image (b) Our-wIE (c) Our-woIE (d) Johnston *et al.*(2020)

(e) GCNDepth (f) Monodepth2 (g) AF-SfM (h) Endo-SfM

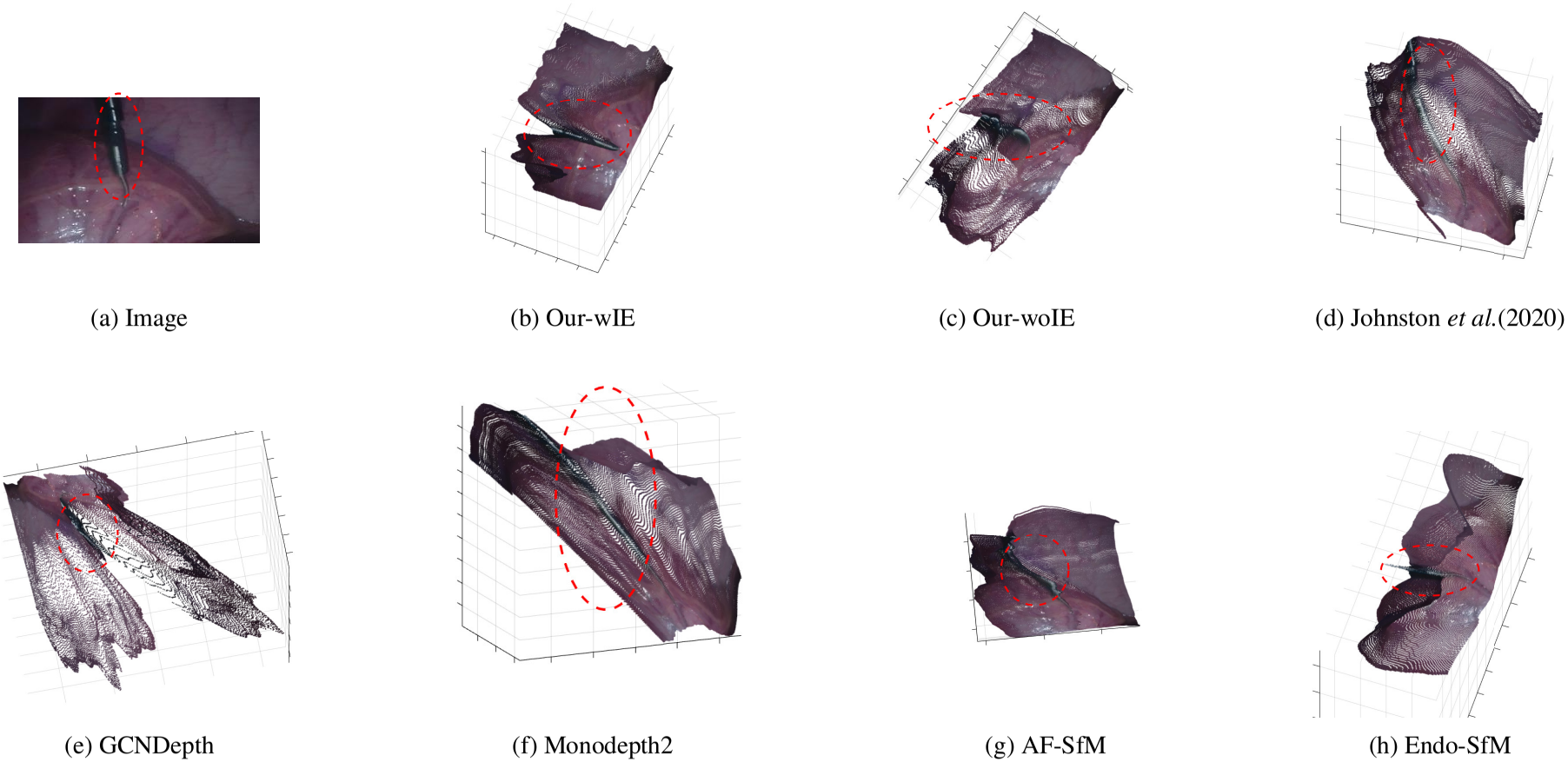
Fig. 5.8 Comparison of point cloud results with a big instrument on the low-illumination dataset, where Our-wIE is the result with the new SSIM loss Eq. 5.4 and Our-woIE is without the new SSIM loss. The proposed framework with CLHE performs better on surgical instruments, and the resulting distribution is more regular.



(a) Image (b) Our-wIE (c) Our-woIE (d) Johnston *et al.*(2020)

(e) GCNDepth (f) Monodepth2 (g) AF-SfM (h) Endo-SfM

Fig. 5.9 Comparison of point cloud results without an instrument on the low-illumination dataset, where Our-wIE is the result with the new SSIM loss Eq. 5.4 and Our-woIE is without the new SSIM loss. The proposed framework with CLHE performs better on surgical instruments, and the resulting distribution is more regular.



(a) Image (b) Our-wIE (c) Our-woIE (d) Johnston *et al.*(2020)

(e) GCNDepth (f) Monodepth2 (g) AF-SfM (h) Endo-SfM

Fig. 5.10 Comparison of point cloud results with a small instrument on the low-illumination dataset, where Our-wIE is the result with the new SSIM loss Eq. 5.4 and Our-woIE is without the new SSIM loss. The proposed framework with CLHE performs better on surgical instruments, and the resulting distribution is more regular.

Chapter 6

Topology-aware Depth Estimation from Unmanned Aerial Vehicle Videos

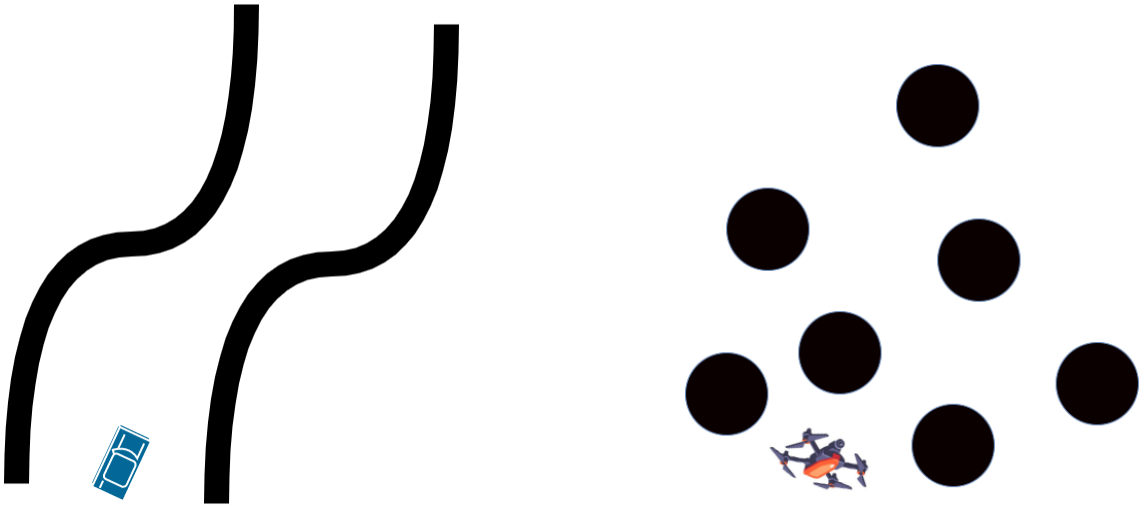
6.1 Motivation

This chapter presents another case study of topology-aware depth estimation from videos captured by Unmanned Aerial Vehicles (UVAs). Based on the proposed novel self-supervised depth estimation framework, this case study aims to demonstrate the effectiveness of the proposed method in unstructured environments with free camera motions.

Most of the works mentioned in Chapter. 2 usually recovered the depth of images captured on road or highway scenes for Autonomous Driving Vehicles. These scenes often have fixed features of or structured environments, as shown in Fig. 6.1(a). For example, the mid-regions of the image are the sky, and both sides are buildings. These environments with fixed features can be considered structured environments, and the fixed features could be learned by CNNs under grid geometry structures. However, similar to endoscopy environments with free camera motions, environments captured by UAVs are often unstructured, as shown in Fig. 6.1(b). Therefore, an implementation of group equivariance deep learning, which utilizes the depth estimation framework proposed in Chapter 4.2, is used for the videos captured by UAVs. In addition, three UAV datasets and six state-of-the-art methods are used for quantitative and qualitative evaluations.

6.2 Introduction

UAVs, often called drones, have emerged as widely adopted platforms for photogrammetric measurements and reconstructions. Their popularity can be attributed to their accessibility, affordability, and exceptional versatility in capturing images. In particular, quadcopters [144] have positioned them as excellent choices for various tasks, including aerial photography, surveying, and delivery services [144]. However, effectively controlling these vehicles remotely demands specialized skills that are difficult and expensive to acquire. This situation has led to a growing demand for automated or assisted flight solutions that can alleviate the need for highly skilled operators [145].



(a) Structured environment (b) Unstructured environment

Fig. 6.1 structured environment and unstructured environment. (a) is a car driving on the road, (b) is a UAV flying in an unstructured environment, and the black circles can be regarded as trees.

In order to avoid undesired collisions in an uncontrolled and unstructured environment, it is necessary to design a motion path or trajectories for UAVs. However, the specialized sensors commonly employed in robotics for distance estimation pose practical challenges when considering their adaptation to small UAVs due to their large size, weight, or power constraints. In practice, UAVs for military applications are usually equipped with powerful cameras, wireless communication devices, highly accurate Global Positioning Systems (GPS), and specialized collision avoidance sensors [146], which can obtain exact positions, but with significant complexities and costs. Therefore, depth estimation based on images, which involves calculating the distance between a camera and an object within the surroundings, has become an attractive solution. In contrast to 2D remote sensing monitoring, which has limitations for the detection of self-occluded vegetation areas, and the assessment of the canopy structure, 3D information can extract the height and volume of plants and develop a more accurate analysis of the plant's condition considering geometric, spatial, and multi-temporal features [147]. 3D information in environmental, agriculture and forestry applications improves the recognition of trees, the study of spatial colonization by dominant species in natural environments, forest inventory, and harvest forecasting [148].

As described in Chapter 1, this research will also focus on monocular depth estimation methods but with UAV videos. Similarly, previous methods can be classified into three categories: SfM-based, supervised and self-supervised depth estimation. SfM-based methods often include image feature extraction and matching, sparse reconstruction and bundle adjustment to reduce accumulated error. Vallet *et al.* [149] improved the quality of a digital terrain model by analysing the internal geometry of the camera. Westoby *et al.* [150] designed a framework to generate a fully rendered 3D model by using photographic data. Nesbit *et al.* [151] improved spatial accuracy and precision in point clouds by fixing

oblique camera angles. These methods often suffer from some challenges, such as the requirement of multiple images, time-consuming bundle adjustment, and low-quality depth in regions with a narrow view between images.

Supervised depth estimation can provide powerful feature extractions that local relations and global cues can be learnt. Miclea *et al.* [152] proposed a new loss function for aerial images, which combines an ordinal regression and a regular classification. The former can improve the results of smooth areas, and the latter can improve the quality of depth in isolated objects. Similarly, due to the limitation in acquiring ground truth for training, Hermann *et al.* [153] utilized three frames to train their model, including stereo frames and a reference frame, for pose estimation and estimate the depth of the reference frame. Madhuanand *et al.* [8] improved the results of depth on complex scenes for UAV images. However, these methods tend to lose fine details based on CNNs. In this chapter, our proposed GNNs-based methods can improve the fine details of depth and achieve better results than the aforementioned works.

6.2.1 Evaluation on UVAs Videos

The proposed framework is compared with prior approaches both quantitatively and qualitatively on the Mid-Air dataset [154], UAVid [155] and Wilduav [156]. The Mid-Air dataset with ground truth is used to conduct both quantitative and qualitative analysis. An ablation study on the Mid-Air dataset is also used to study the impact of GNNs and various Resnets based on the availability of the ground truth.

Since the UAVid and Wilduav datasets have no ground truth, created by SfM-based methods, these datasets with real-world data are used to conduct qualitative analysis only. Similarly, the details of experiments are the same as in Chapter. 5.3, and the resolution of images is set to 320×192 for all datasets.

6.2.2 Quantitative Evaluation

The Mid-Air dataset is a synthetic dataset for unstructured environments, which includes a flying quadcopter equipped with navigation and vision sensors. This dataset includes the ground truth of depth and different climates. In this experiment, three types of data are quantitatively evaluated: sunny weather, and spring and winter seasons, which have significantly different features. Sunny weather is the clear sky at midday. The features of the spring season are trees with green leaves and luxuriant ground vegetation. In addition, it contains different times of the day and foggy weather. On the contrary, features of the winter season include trees without leaves and an environment covered with snow. The left images are used for all experiments, and the details are as shown in Table 6.1. The depth estimation performance of the proposed framework is evaluated against six state-of-the-art self-supervised methods: GCNDepth [6], Monodepth2 [4], Johnston *et al.*(2020) [5], Lite-mono [7], MonoFormer [9], Madhuanand *et al.*(2021) [8]. Apart from GCNDepth, Monodepth2 and Johnston *et al.*(2020), the latest works on monocular depth estimation, called Lite-mono and MonoFormer, are used to compare with the proposed

Table 6.1 The training details in different datasets.

Datasets	Training images	Validation images	Testing images
Sunny Weather	46360	17869	2000
Spring Season	37044	14376	1500
Winter Season	37044	14376	1500

method. Madhuanand *et al.*(2021)focused on the depth estimation of UAV images, which is also introduced in this chapter. Since GCNDepth has no convergence on all UAV datasets, the model provided by GCNDepth is used for evaluations. In addition, the experiments are conducted for Madhuanand *et al.*(2021) according to the part code provided and the article. All models are trained with the same conditions.

Similar to Chapter 5, several standard depth evaluation metrics are used, such as Absolute and Relative Error (Abs Rel), Squared Relative Error (Sq Rel), Root Mean Squared Error (RMSE) and Root Mean Squared Log Error (RMSE log). The details are shown in Eq. 5.1, the results are shown in Table 6.2, and the Abs Rel metric is used for subsequent analysis. In general, the proposed method for UAV images outperforms other state-of-the-art methods, and the results of our method are better in all evaluation metrics compared with those of others. The proposed method achieves similar results on Sunny Weather and Spring Season datasets since there are many similar scenes in the two datasets. The Sq Rel metric improves by $\sim 66.67\%$ on the Sunny Weather dataset and $\sim 55.16\%$ on the Spring Season dataset compared with others. There is a significant improvement in the Winter Season dataset. This is because there are different situations in the Sunny Weather dataset, including illumination or fog, and the Winter Season dataset is almost white. The results are better than the former. The following section details qualitative analysis.

6.2.3 Qualitative Evaluation

The results of three datasets will be compared for qualitative evaluation: Mid-Air, UAVid and Wilduav. In contrast to endoscopy datasets, these datasets usually have a big range of distances different from endoscopy. Thus, disparity images rather than point clouds will be used for analysis. The results of the Mid-Air dataset are shown in Fig. 6.2 and Fig. 6.3. The former shows the results of the Sunny dataset. The results of the proposed method are a closer approximation to ground truth (GT). In particular, the proposed framework can reconstruct trees, as shown in the green circles of the second column. As can be seen, the results of Johnston *et al.*(2020), Monodepth2, Lite-mono and MonoFormer cannot estimate the depth of trees, and models of Madhuanand *et al.* and GCNDepth produce bad results in the Sunny dataset. The latter shows the results in the Spring and Winter datasets. The first and third columns have the same scenes with different climates. The models of Lite-mono, MonoFormer and the proposed method achieve similar results in the green regions. However, the results of Johnston *et al.*(2020) and Monodepth2 in the green region tend to be together between leaves. In addition, the proposed method and Monodepth2 achieve better results in the fourth column, which has a clear depth in the tree trunk.

Table 6.2 Quantitative results. Comparison of our method to existing methods on the Sunny Weather, Spring Season and Winter Season. The best results in each category are in **bold**.

	Lower Better				Higher Better
Method	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$
Sunny Weather					
MonoFormer	0.057	0.957	5.078	0.111	0.945
GCNDepth	0.082	0.898	7.610	0.116	0.903
Madhuanand *et al.*(2021)	0.066	0.771	6.329	0.107	0.900
Monodepth2	0.060	1.461	5.575	0.132	0.894
Johnston *et al.*(2020)	0.076	1.840	6.710	0.159	0.856
Lite-mono	0.079	1.919	7.111	0.166	0.865
Our	**0.042**	**0.473**	**3.628**	**0.071**	**0.953**
Spring Season					
MonoFormer	0.050	0.438	4.277	0.081	0.943
GCNDepth	0.082	0.870	7.334	0.112	0.920
Madhuanand *et al.*(2021)	0.069	0.839	6.824	0.116	0.900
Monodepth2	0.092	1.602	8.166	0.169	0.813
Johnston *et al.*(2020)	**0.042**	0.302	3.606	0.066	0.964
Lite-mono	0.053	0.426	4.350	0.081	0.964
Our	**0.042**	**0.277**	**3.449**	**0.063**	**0.976**
Winter Season					
MonoFormer	0.124	4.457	10.214	0.296	0.841
GCNDepth	0.080	0.801	7.139	0.107	0.934
Madhuanand *et al.*(2021)	0.061	0.668	5.812	0.098	0.922
Monodepth2	0.058	0.870	4.909	0.101	0.889
Johnston *et al.*(2020)	0.194	8.796	17.473	0.707	0.758
Lite-mono	0.118	3.750	9.715	0.257	0.842
Our	**0.025**	**0.095**	**1.982**	**0.035**	**0.984**

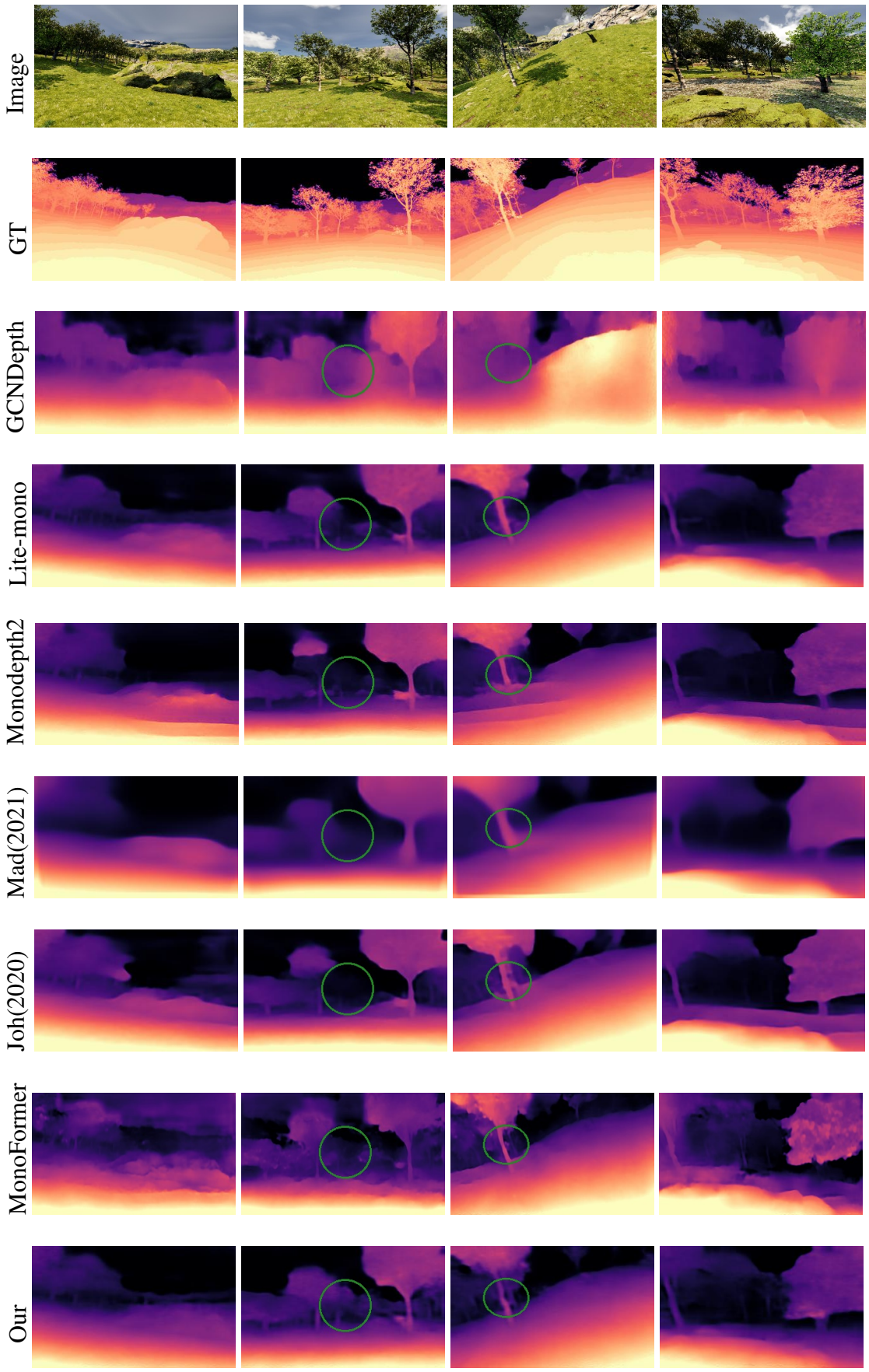


Fig. 6.2 Comparison of monocular depth estimation results on Sunny dataset of Mid-Air. The first row is test images, and the next is ground-truth depth. From top to bottom, models are GCNDepth [6], Lite-mono [7], Monodepth2 [4], Madhuanand *et al.*(2021) [8], Johnston *et al.*(2020) [5], MonoFormer [9], the proposed method(Our).

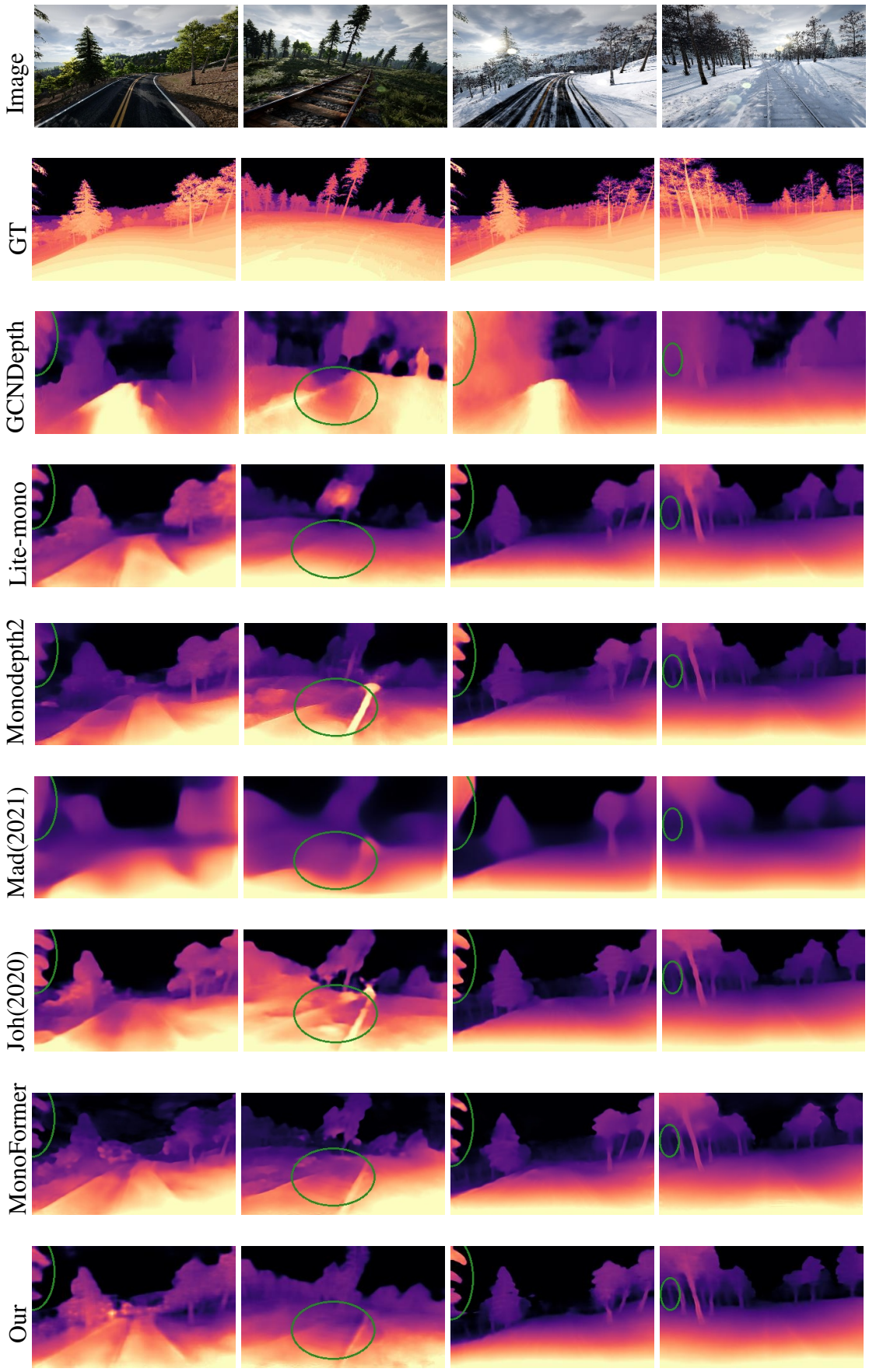


Fig. 6.3 Comparison of monocular depth estimation results on Spring and Winter dataset of Mid-Air. The first row is test images, and the next is ground-truth depth. From top to bottom, models are GCNDepth [6], Lite-mono [7], Monodepth2 [4], Madhuanand *et al.*(2021) [8], Johnston *et al.*(2020) [5], MonoFormer [9], the proposed method(Our).

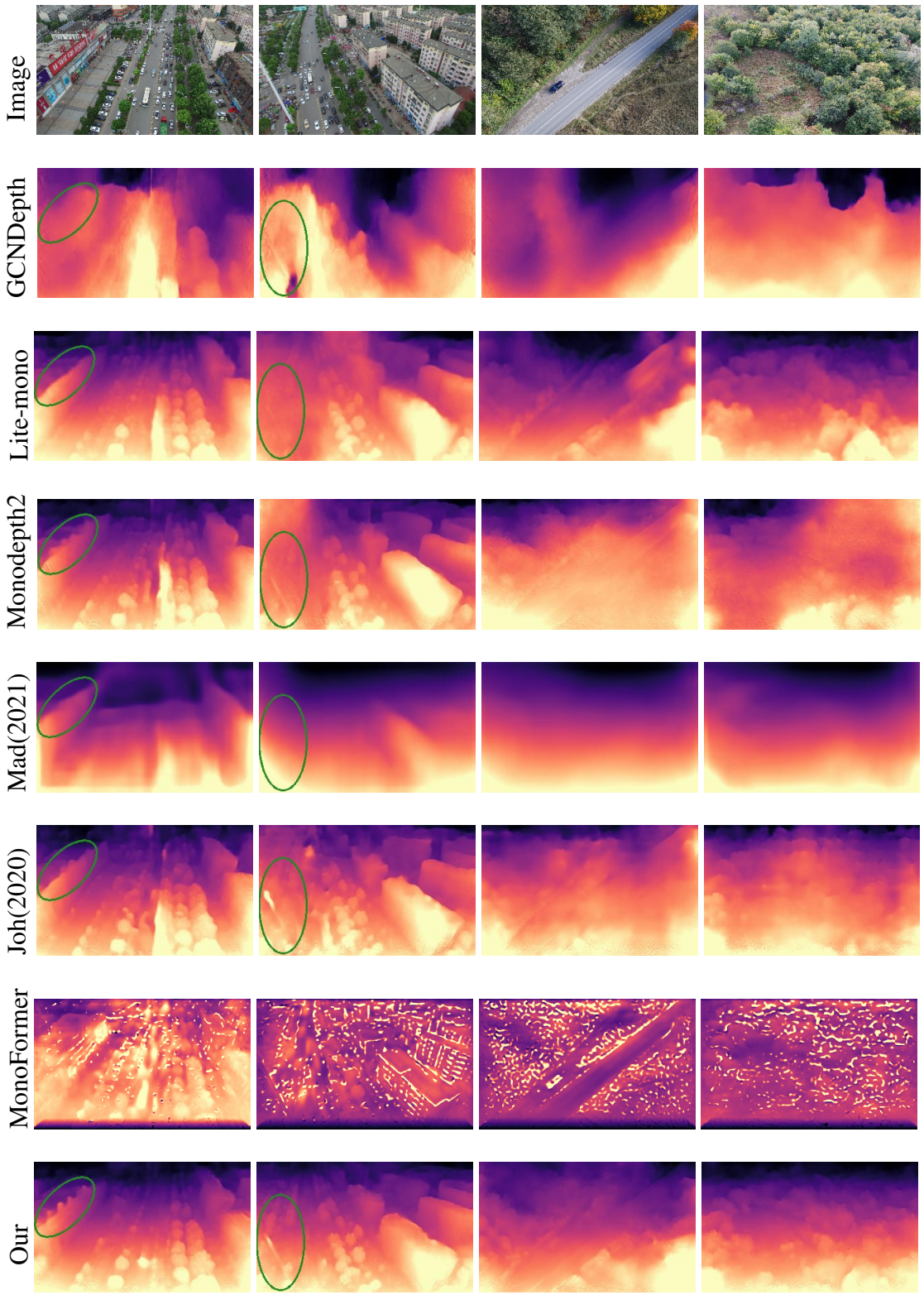


Fig. 6.4 Comparison of monocular depth estimation results on China of UAVid and Wilduav. The first row is test images. From top to bottom, models are GCNDepth [6], Lite-mono [7], Monodepth2 [4], Madhuanand *et al.*(2021) [8], Johnston *et al.*(2020) [5], MonoFormer [9], the proposed method(Our).

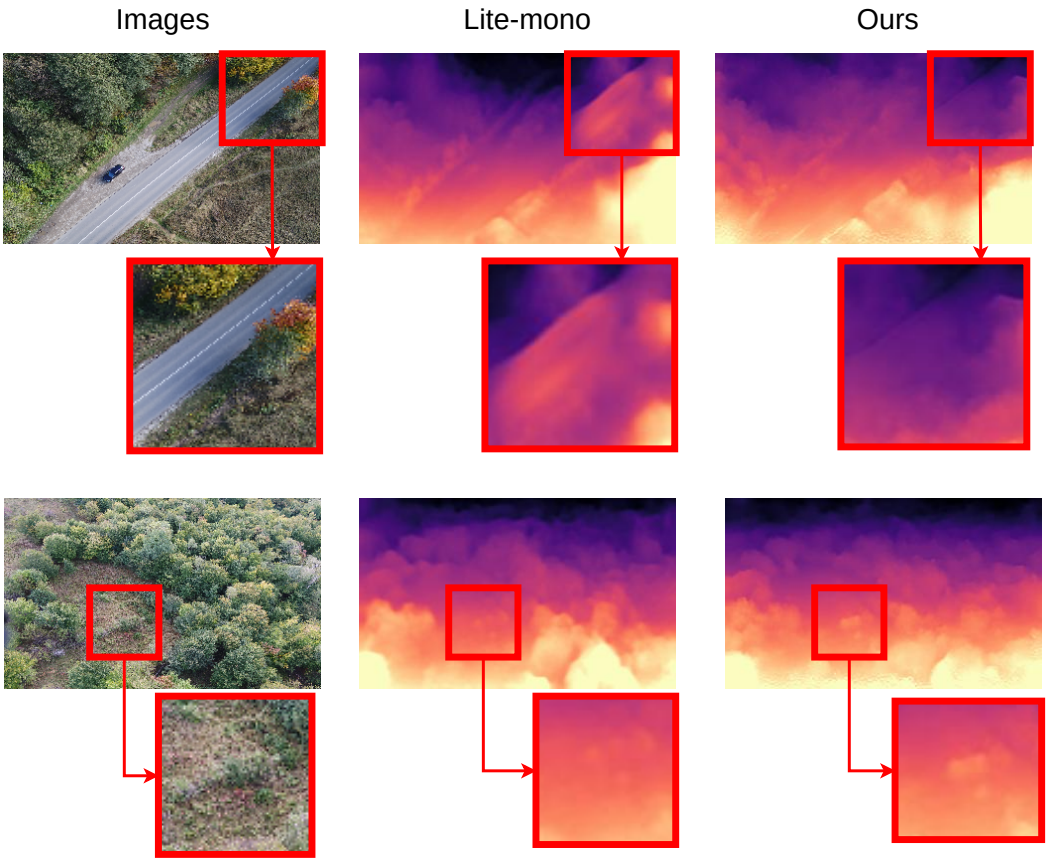


Fig. 6.5 Comparison of the detailed results on Wilduav. The first column is images, the second column is the results of Lite-mono, and the last one is the proposed framework(Ours)

Table 6.3 Ablation results for different components. **w/o GAT** represents without GAT network. The best results in each category are in **bold**.

Methods	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$
Baseline(Resnet-18 w/o GAT)	0.064	0.711	7.811	0.092	0.949
Ours-Resnet-18 w/ GAT	0.056	0.494	6.366	0.078	0.978
Ours-Resnet-50 w/o GAT	0.066	1.677	6.203	0.150	0.894
Ours-Resnet-50 w/ GAT	0.051	0.780	4.473	0.093	0.913
Ours-Resnet-101 w/o GAT	0.049	0.871	4.292	0.091	0.930
Ours-Resnet-101 w/ GAT	**0.042**	**0.473**	**3.628**	**0.071**	**0.953**

The UAVid and Wilduav datasets are the real world but with different scenes. The UAVid dataset is captured in different country cities: China and Germany. The Wilduav dataset is captured at low altitudes in unstructured forest and shrubgrass vegetation areas with varying terrain profiles. The experiment on real-world datasets can demonstrate that the proposed method can estimate the depth of images captured by UAVs in the real world. The UAVid and Wilduav datasets differ from the Mid-Air dataset, where the UAV is located in a low attitude. Therefore, the models need to be trained. The China dataset of the UAVid is only trained, and the trained models will be directly used for the Wilduav dataset without any finetuning. Similar to Madhuanand *et al.*(2021), the frame rate needs to be changed since the small parallax error may lead to various noises, and it is set to 20 for all experiments. There are 24800 images used for training and 2752 images for validation. The results are shown in Fig. 6.4. The results of GCNDepth and MonoFormer cannot estimate the right values of depth, and Madhuanand *et al.*(2021) and Monodepth2 fail to recover the depth on the Wilduav dataset. For the China dataset, the proposed method can accurately capture the depth of the billboard on the roof of the building, as shown in the green area in the first column of Fig. 6.4. Although there are similar results in the Wilduav dataset, our method can achieve fine details for the depth, as shown in Fig. 6.5. The proposed method can estimate the depth of the road and small bushes.

6.2.4 Ablation Study

Similar to the evaluation of endoscopy datasets, an ablation study is conducted to understand the contribution of each component in the proposed framework to the overall performance during training. The results are shown in Table 6.3. The Resnet-18 only include a discrete disparity volume encoder without a GAT network. Using ResNet-101 instead of ResNet-18 and ResNet-50 improved the results and accuracy slightly. The baseline model, without any of the proposed contributions, performs the worst. However, when all components are combined, the proposed method observes a significant improvement in performance. In addition, Table 6.4 shows different network framework results based on GAT. There are lower Abs Rel on the resolution of 192×320 with four-connectivity and eight-connectivity than 256×320 with four-connectivity. The networks with mixed connectivity (four and eight) and the networks with random edges yield similar results. Finally, the network with learned edge features tends to achieve the highest Abs Rel.

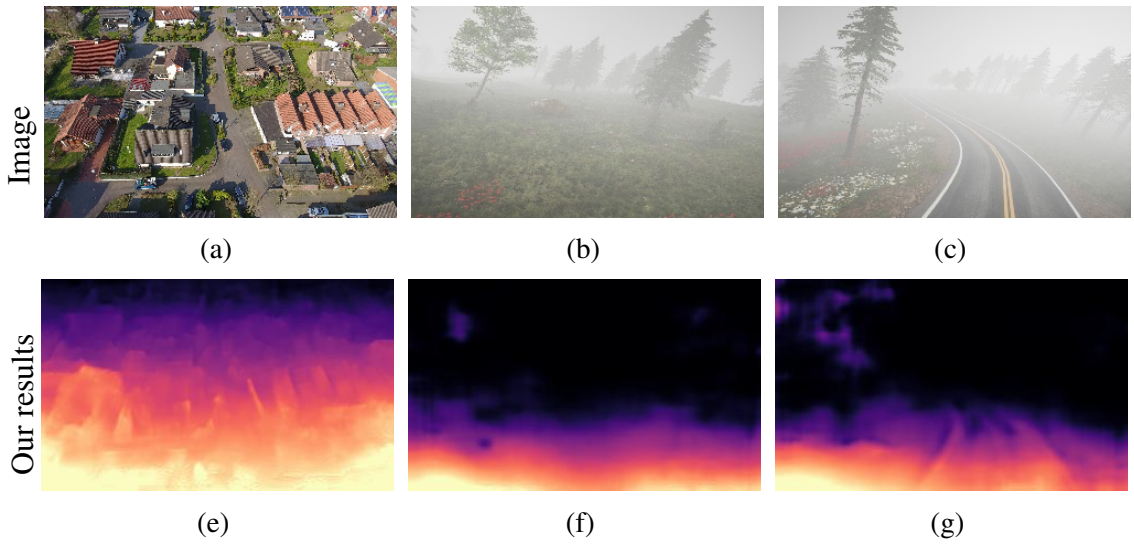


Fig. 6.6 Some failed examples in different datasets: (a) is an image in the Germany dataset of UAVid, and (e) is its result; (b) and (c) are images in the Spring dataset with fog, and their results of depth estimation are (f) and (g).

Different results emerge with endoscopy data, where **Ours-192-**4&8 and **random** exhibit the lowest performance.

Table 6.4 Compared with other network frameworks. **Ours-256-4** represents the results of 320×256 resolution with four-connectivity, **Ours-192-8** represents the results of 320×192 resolution with eight-connectivity, **random** represents that the connectivity of edges is randomly generated, and **EGAT** represents that the GAT includes the feature of edges. The best results in each category are in **bold**.

Methods	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$
Ours-256-4	0.051	0.753	4.389	0.090	0.931
Ours-192-8	0.045	0.535	3.898	0.078	0.938
Ours-192-4&8	0.050	1.186	4.564	0.107	0.933
Ours-192-random	0.051	1.118	4.528	0.106	0.942
Ours-192-EGAT-4	0.056	0.971	4.846	0.104	0.920
Ours-192-EGAT-8	0.060	0.634	5.680	0.095	0.926
Ours-192-4	**0.042**	**0.473**	**3.628**	**0.071**	**0.953**

6.2.5 Discussion

The novel self-supervised monocular depth estimation framework is evaluated in unstructured environments. The coarse-to-fine encoder architecture can achieve fine-detailed results. In addition, qualitative and quantitative analysis has demonstrated that the proposed framework is effective in improving the depth details of the UAV dataset. However, some challenges need to be addressed, as shown in Fig. 6.6. The proposed method obtains unsatisfactory results when applied to the Germany dataset, particularly demonstrating limitations in accurately estimating the depth of trees under foggy weather conditions. Another challenge is the dataset in UAVs, which should include the ground truth in the real world rather than the depth generated by SfM-based methods.

6.3 Conclusion

This chapter describes another use case that is using videos captured by UAVs to evaluate the proposed depth estimation framework in Chapter 4. Mid-Air dataset with different climates is used for quantitative evaluation, and the proposed framework achieves lower errors compared with six state-of-the-art methods. Our method can recover fine details of depth information in the Mid-Air dataset for qualitative evaluation. On the UAVid and Wilduav datasets captured in the real world, the edge region and small bushes of depth can be recovered by the proposed depth estimation framework. Finally, the results of the ablation study show the effectiveness of the proposed method. However, depth estimation in foggy weather has shown poor performance, and the depth, including numerous distant regular shapes, also requires further investigation.

Chapter 7

Non-rigid Point Cloud Registration with Topology Changes[1]

7.1 Motivation

This chapter proposed a novel non-rigid point cloud registration framework, which can handle topology changes. 3D object matching and registration on point clouds are widely used in computer vision. However, most existing point cloud registration methods have limitations in handling non-rigid point sets or topology changes (e.g., connections and separations). As a result, critical characteristics such as large inter-frame motions of the point clouds may not be accurately captured. This chapter proposes a statistical algorithm for non-rigid point set registration, addressing the challenge of handling topology changes without the need to estimate correspondence. The algorithm uses a novel *Break and Splice* framework to treat the non-rigid registration challenges as a reproduction process and a Dirichlet Process Gaussian Mixture Model (DPGMM) to cluster a pair of point sets. Labels are assigned to the source point set with an iterative classification procedure, and the source is registered to the target with the same labels using the Bayesian Coherent Point Drift (BCPD) method. The results demonstrate that the proposed approach achieves lower registration errors and efficiently registers point sets undergoing topology changes and large inter-frame motions. The proposed approach is evaluated on several data sets using various qualitative and quantitative metrics. In addition, an application of endoscopy image, which can reconstruct point clouds by the method in Chapter 4, is conducted by the proposed non-rigid registration framework.

7.2 Introduction

Point cloud registration is a crucial step in 3D acquisition and has many applications in computer vision, including 3D reconstruction, pose estimation, augmented reality, object matching, and recognition [73], [14], [74], [75]. Accurate registration of multiple point

[1]Published: QingHong Gao, Yan Zhao, Wen Tang, TaoRuan Wan, Long Xi. Break and Splice: A Statistical Method for Non-rigid Point Cloud Registration. Computer Graphics Forum (2023).

clouds obtained from different views or time instants is necessary for building a complete and consistent 3D model of the scene or object of interest. In addition, point cloud registration enables us to estimate the relative pose and motion of objects, recognize and match objects in different scenes, and create virtual and AR experiences [76], [77].

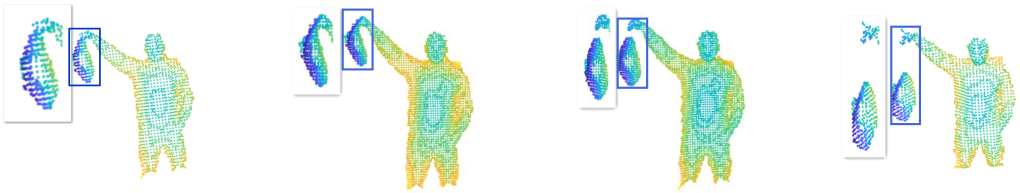
While many registration methods work well on rigid objects [78], [79], they often perform poorly on dynamic scenes or deformed objects. This is because these objects have non-rigid deformations and motions that cannot be modelled by rigid transformations. In addition, non-rigid objects may undergo topology changes such as connections and separations, which pose additional challenges for registration methods that rely on correspondences between the source point sets and the target point sets point [80], [81]. Therefore, developing registration methods that can handle non-rigid objects and dynamic scenes is an active research area in computer vision.

Topological changes are common in dynamic scenes. Many previous works on non-rigid registration have failed to effectively address the connection and separation issues, and large inter-frame motions, which can lead to misregistration and inaccurate reconstructions. In addition, large inter-frame motions can cause significant deformations and changes in topology. Chui *et al.* [42] and Yang *et al.* [43] proposed non-rigid registration methods based on point correspondence to estimate affine transformations between the source and the target point sets. However, these methods may not be robust to changes in the topology of the point sets, such as connections and separations. The accuracy of these methods is highly dependent on the accuracy of the correspondence estimation, which can be challenging in the presence of large deformations or changes in topology.

To address these issues, recent approaches have focused on developing statistical methods that do not rely on explicit point correspondence or feature extraction, but instead model the probabilistic relationship between the point sets. Myronenko *et al.* [84] and Hirose [89] have utilized statistical methods, such as the motion coherence theory, to estimate maximum-likelihood solutions for non-rigid registration. However, some of these methods, such as Coherent Point Drift (CPD), can suffer from local minima and slow convergence. More recently, Zampogiannis *et al.* [90] have explored a bidirectional estimation method, where pairs of point clouds are aligned from the source to the target and from the target to the source. However, this method still struggles to handle large inter-frame motions, where the source and target point clouds may undergo significant deformation between frames.

The challenge of handling objects' *separation and connection* can be best illustrated through an example shown in Fig. 7.1. In this example, from frame 69 to frame 89, where the hat is separated from the hand, large inter-frame motions are manifested. The difficulty of non-rigid registration stems from the effectiveness of the method in dealing with a single point set when it is rapidly separated into two point sets (the hat and the hand). The connection between the two point sets is regarded as a reverse process of separation.

In summary, this chapter proposes a novel statistical approach that can handle changes in topology and large inter-frame motions. The proposed framework utilizes a statistical approach based on the DPGMM to handle non-rigid point sets without the need for explicit



(a) frame 69 (b) frame 72 (c) frame 86 (d) frame 89

Fig. 7.1 Non-rigid registration challenges. The (a), (b), (c), and (d) show that the hat is separated from the hand from frame 69 to 89 in the public data set, including two object separations [10].

correspondence estimation, leading to improved registration performance compared to previous methods. The proposed method regards non-rigid registration as a reproduction process with a four-step registration scheme, as shown in Fig. 7.2, which generates a model as close as possible to the target model. Another contribution of this chapter is that a *Cluster and Refine* scheme is designed to handle the distribution irregularities of point sets, making the topology of source points the same as that of target points, which results in a great improvement of the accuracy and efficiency of the proposed statistic-based methods. The proposed method is evaluated on five datasets using a variety of qualitative and quantitative metrics. It is important to note that the experimental datasets only contain single object separations and connections against a simple background. In the following parts, a detailed *Break and Splice* registration framework and methodology will be introduced.

7.3 Method

Given two 3D point sets **X** and **Y**, the proposed framework aims to handle registrations between two point sets **X** and **Y** that exhibit *Connection* and *Separation* topology changes. The framework *Break and Splice* is designed to reproduce the states of *Connection* and *Separation* between point sets. For brevity, a *Separation* example is used to introduce this framework, where **X** is the target point set, and **Y** is the source point set, as shown in Fig. 7.2.

The *Break and Splice* framework can be divided into three modules: a) *Assigning labels* to aim to determine the partition template (point sets with more boundaries, such as **X**) to be allocated different labels; b) *Break and Splice* assigns labels to **Y** based on the labels of **X**; c) *Registration* utilizes the Bayesian Coherent Point Drift (BCPD) algorithm to achieve registration between point sets with same labels.

The proposed non-rigid registration method between the point sets **X** and **Y** can be divided into four steps. *Step 1*, the partition template is determined by using Delaunay triangulation to extract boundaries of the target and source point sets and assign labels for each point of partition template **X** based on the boundaries in the *Assigning labels module* (Section 7.3.1). *Step 2*, the source and target point sets are mixed into one to reduce the cluster difference that will occur in the *Break and Splice module* (Section 7.3.2). *Step 3*, in

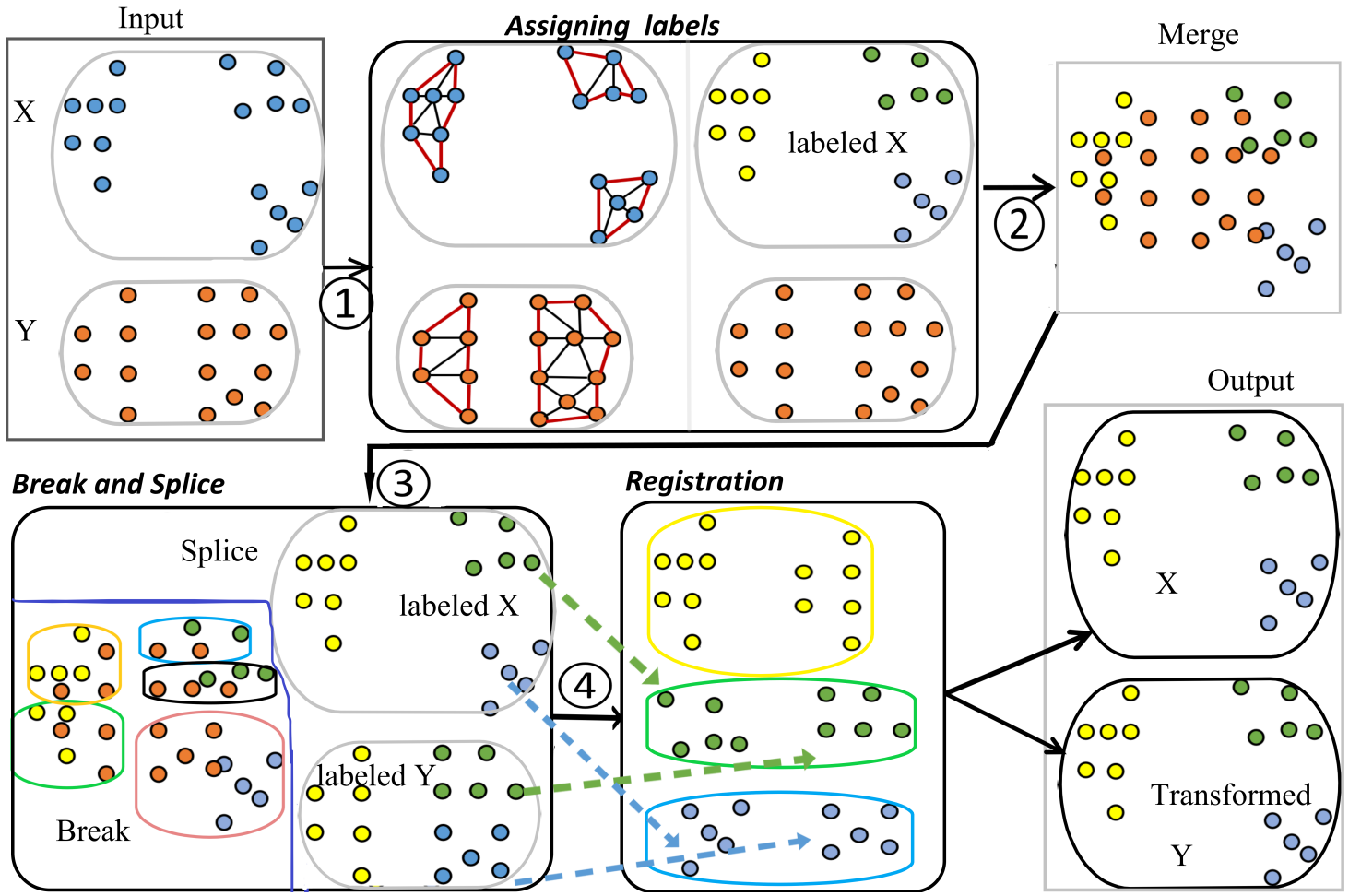


Fig. 7.2 Overview of *Break and Splice* registration framework (The different colours mean different labels, and the black arrow indicates the process of our non-rigid registration.): *Step 1*, extracting boundaries of point sets to determine the partition template and allocating labels to the partition template **X**. In *Step 2*, the labelled point set **X** and the unlabeled point set **Y** are merged into one. *Step 3*, assigning labels to point set **Y** according to the labels of partition template **X**. Specifically, clustering the merged point sets into different groups and assigning labels to unlabeled points in each cluster. **X** and **Y** will be reassembled by *Splice* together the points in different clusters, respectively. *Step 4*, point sets with the same labels are registered to obtain the transformed source point set **Y**(The dashed arrows indicate registration with the same labels.).

the *Break and Splice module*, the DPGMM is used to cluster point sets and assign labels to **Y** according to the labels of **X** in each cluster. Then, the partitions of points with the same labels are spliced together to generate point subsets that need to be registered. *Step 4*, in the *Registration* module (Section 7.3.3), the Bayesian Coherent Point Drift (BCPD) method is used to register the source point set groups and the target groups which have the same labels.

7.3.1 Assigning Labels

The boundary extracting approach [157] is used to identify the boundaries of point sets **X** and **Y**. The point sets with more boundaries are selected as the partition template, which will be allocated to different labels. The target point set **X** is used as the example for boundary identification and assigning labels, for brevity's sake, as shown in Fig. 7.3.

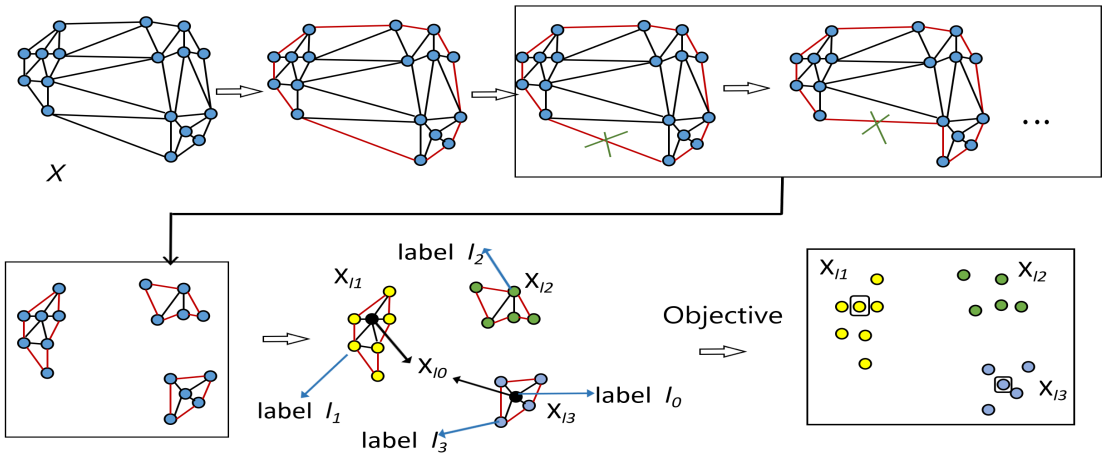


Fig. 7.3 Assigning labels for **X**: The top shows the process of extracting boundaries using triangulation, and the bottom illustrates the allocating labels l_1, l_2 and l_3 based on extracted boundaries. The red is the extracted boundaries. Black circles are the inner points $\mathbf{X}_{l_0}$. Our objective is to allocate labels to those inner points.

Boundary Identification

The point set **X** is projected to a 2D plane, and the Delaunay triangulation is formed on plane points (shown in Fig. 7.3). One of the triangle sides along the periphery of **X** is associated with only one triangle. All such edges on that side form the initial boundary. Let d_{max} indicate the maximum distance between the nearest neighbouring points in **X**. Any initial boundary edges whose length exceeds $2d_{max}$ will be removed (d_{max} is achieved by K-Nearest-Neighbor). The removal of long boundary edges continues iteratively until every edge along the boundary is at most d_{max} in length.

Assigning Labels for the Target Point Set

Once the boundary of **X** is identified, different connected components are assigned different labels. An adjacency matrix can be obtained after using Delaunay triangulation to extract

the boundaries of **X**. The elements in the adjacency matrix represent the number of triangles that share the same edges connected by nodes. Then, the breadth-first search (BFS) method is used to search for connected components. Assuming there are three different connected components by boundary identification from **X**, the nodes/points in the same connection component will be assigned the same label. The points in **X** will be allocated the labels $L_x = \{l_1, l_2, l_3\}$.

7.3.2 Break and Splice

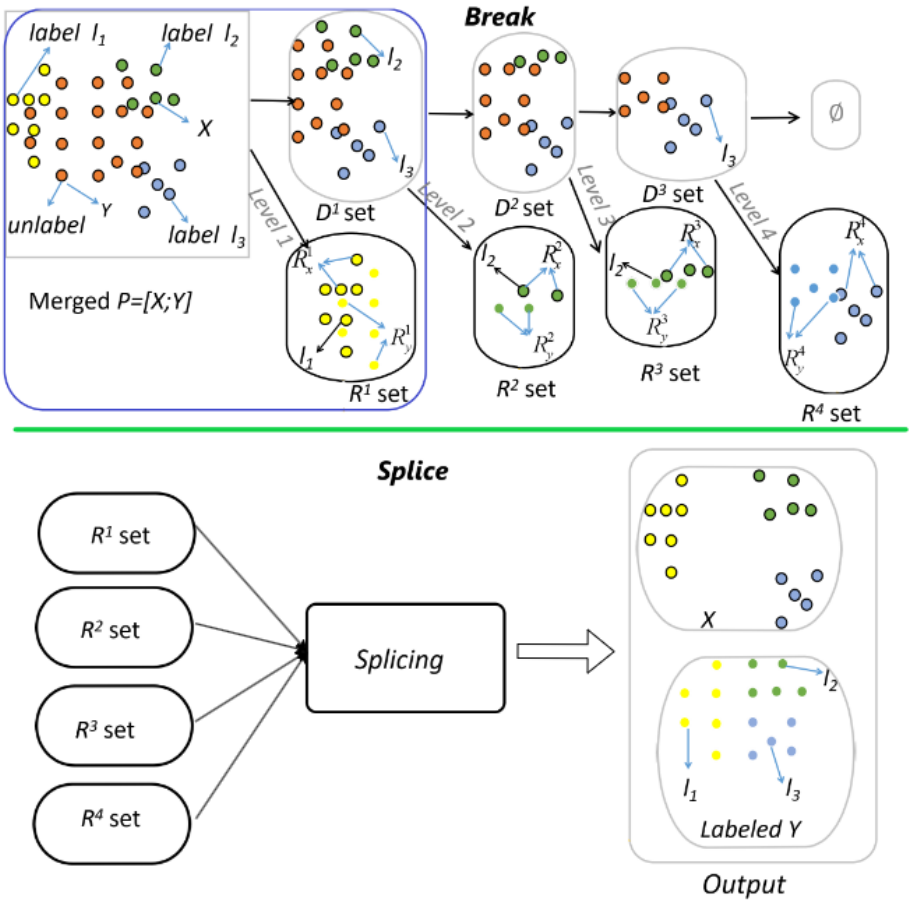


Fig. 7.4 The structure of the *Break and Splice* module for assigning labels to source points set **Y**: The blue box shows *Break*, which involves the partitions and labels allocation of point sets. *Splice* indicates the stitching of the partitions based on the labels. The *Break* is a binary tree with merged point sets as a root node, and the leaf node $R^j set$ keeps the points that will not be re-partitioned at the $j+1$ level, and the branch node $D^j set$ contains the points to be divided. R_x^j and R_y^j represent the target points, and source points in the $R^j set$, $\varnothing$ demonstrates that there is no point left to be partitioned, which marks the end of the partition. *Splice* reassembles the points (R_x^j and R_y^j) in different $R^j set$ to recover the labeled source point set **Y** and target point set **X**.

The Break and Splice framework is similar to that of the bisection method. The Break process will not end until the category of the labels of target points in each cluster is unique. Just like the bisection method, the process of splitting the interval will not end until the solution of a continuous function is found. This framework aims to assign the

labels of target points to the source points. The reason to set the process of assigning labels to source points as a binary tree structure is that there is more than one way to assign labels to source points in one cluster after only one partition. To make the label of source points in one cluster unique, the cluster with different labels of the target points needs to be re-partitioned. This process will be repeated until the category of the labels of target points is unique in one cluster. However, the method of the cluster may cause irregularity of the initial partitions, confusing label allocation and increasing the error of registration. In Section 7.3.2, a refinement algorithm is proposed to handle this gap.

After attaining the labels L_x for the points set $\mathbf{X}$, the goal then is to allocate labels L_x to the points set $\mathbf{Y}$. Assigning L_x to $\mathbf{Y}$ can be regarded as the process of *Break* and *Splice*. *Break* involves the partition of the merged point sets $\mathbf{P} = [\mathbf{X}; \mathbf{Y}]$ and the label allocation in each partition, which is the key, to handling the topology changes between point sets for registration. *Splice* is to splice the partitions of $\mathbf{Y}$ together based on the allocated labels. Fig. 7.4 shows how the *Break and Splice* process is carried out . *Break* can be regarded as a binary tree generation process. The root node of the binary tree is the merged points set $\mathbf{P} = [\mathbf{X}; \mathbf{Y}]$, and the leaf nodes $R^j set$ include the target point sub-sets R_x^j and the source point sub-sets R_y^j. The branch node $D^j set$ keeps the merged points sub-set that will be divided at the $j+1$ level. The binary tree grows until one of the leaf nodes is empty $\varnothing$. It is worth noting that the points in each leaf node $R^j set$ have the same labels, and the points in branch node $D^j set$ have different labels. Namely, the number of category labels in nodes determines whether the node is a leaf node or a branch node. *Splice* recovers the labelled target points set $\mathbf{X}$ and labelled source points set $\mathbf{Y}$ by reassembling the R_x^j and R_y^j, respectively.

The process of *Break* can be divided into *Cluster* and *Refine*. *Cluster* is to utilize DPGMM [158] to attain the initial partitions C_k, $k \in \{1, \cdots, K\}$, where K is the number of partitions. The advantage of DPGMM is that it automatically discovers the number of clusters and is likely to converge to the data's actual clusters [159]. To guarantee the consistency of the partitions for target points set X and source points set Y, the *Refine* is used to overcome the significant irregularity of the initial partitions. Based on the pruned partitions, the labels of target points C_x^k are allocated to the source points C_y^k. Fig.7.5 illustrates the structure of *Break* at *Level 1* of the binary tree. Especially, it is different from *Step 1*. In *Step 1*, the KNN is used to search the K points close to the initial point, and then the K points will be assigned the label of the initial point. DPGMM, as a clustering method, clusters the merged points according to the coordinates of points without defining the number of clusters (such as the K in the K-means clustering method).

Cluster

Suppose $\mathbf{P}$ is a mixture of K Gaussian distributions (K is unknown). For simplicity, we note $\mathbf{p}_i$ as the i_{th} point in $\mathbf{P}$. $\mathbf{c} = \{c_1, \cdots c_{M+N}\}$ $(c_i \in \{1, \cdots, K\})$ is the indicator variables, and $c_i = k$ indicates that point p_i is generated from the k_{th} Gaussian distribution. π_k is defined

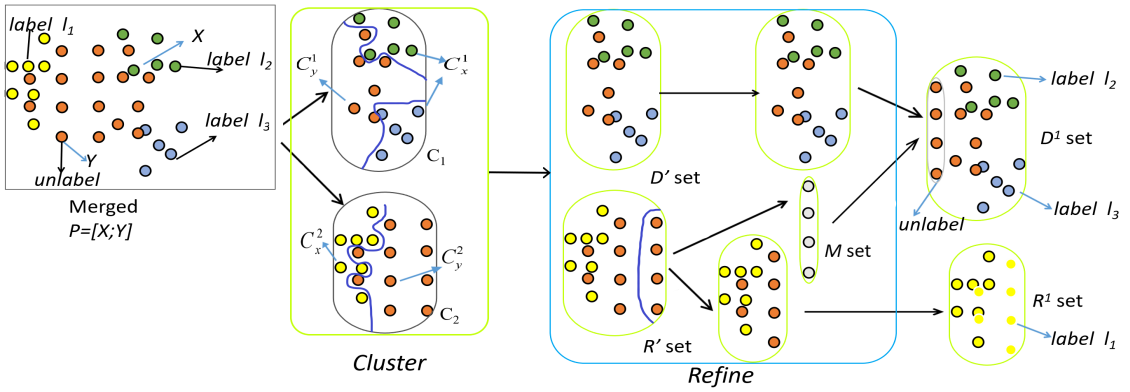


Fig. 7.5 The structure of *Break* at *Level 1* of Fig. 7.4 (The black arrow indicates the process of the *Break* at *Level 1*. *Cluster* involves the initial partitions (C_1 and C_2) of merged points using the DPGMM clustering method. The target points sub-set C_x^1 in C_1 have different labels and C_2 contains the target points sub-set C_x^2 with the same labels. Besides, the cluster C_1 and cluster C_2 also include the source points sub-set C_y^1 and C_y^2. The clusters with single labels form the $R^{'}$*set*. $D^{'}$*set* includes those clusters with different labels. *Refine* aims to overcome the significant irregularity of $R^{'}$*set*. The irregular point sets are selected as *M set* to be mixed with $D^{'}$*set* to generate the brunch node D^1 *set*, which will be divided at *Level 2*. The $R^{'}$*set*'s remaining source points will then be allocated the labels of the target points in the same clusters. The target points and the source points with the same labels form the R^1 *set*, which is the leaf node in Fig. 7.4

.

to represent the weight of the k_{th} Gaussian component, where $\pi_k \geq 0, k = \{1, \cdots K\}$, and $\sum_{k=1}^{K} \pi_k = 1$.

The Gaussian mixture model with K components may be written as:

$$p\left(\mathbf{p}_i|\theta_1, \cdots, \theta_K\right) = \sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{p}_i|\boldsymbol{\mu}_k, \mathbf{S}_k\right) \tag{7.1}$$

where $\theta_k = \{\boldsymbol{\mu}_k, \mathbf{S}_k, \pi_k\}$ is the set of parameters for component k. $\boldsymbol{\mu}_k$ is the mean vector for k_{th} Gaussian component, and $\mathbf{S}_k$ is its precision (inverse covariance matrix). We set the joint prior distribution on the component parameters as *Normal-Wishart* distribution.

The conditional posterior class probabilities derived by the Dirichlet Process Gaussian Mixture Model (DPGMM) are

for the k_{th} component with $n_{-i,k} > 0$:

$$\begin{aligned} & p\left(c_i = k|\mathbf{c}_{-i}, \boldsymbol{\mu}_k, \mathbf{S}_k, \alpha\right) \\ & \propto \frac{n_{-i,k}}{M+N-1+\alpha} \mathcal{N}\left(\mathbf{p}_i|\boldsymbol{\mu}_k, \mathbf{S}_k\right) \end{aligned} \tag{7.2}$$

for new Gaussian component:

$$
\begin{aligned}
& p\left(c_i \neq c_{i'} for\ all\ i \neq i' | \mathbf{c}_{-i}, \boldsymbol{\xi}, \rho, \beta, \mathbf{W}, \alpha\right) \\
& \propto \frac{\alpha}{M+N-1+\alpha} \\
& \times \int p\left(\mathbf{p}_i | \boldsymbol{\mu}, \mathbf{S}\right) p\left(\boldsymbol{\mu}, \mathbf{S} | \boldsymbol{\xi}, \rho, \beta, \mathbf{W}\right) \mathrm{d}\boldsymbol{\mu}\, \mathrm{d}\mathbf{S} \\
& \propto \frac{\alpha}{M+N-1+\alpha} t_{\beta_n - D+1}\left(\boldsymbol{\xi}_*, \frac{\mathbf{W}_*\left(\rho_n+1\right)}{\rho_n\left(\beta_n - D+1\right)}\right)
\end{aligned} \tag{7.3}
$$

where α is the concentration parameter of the Dirichlet Process, $\alpha > 0$. $\boldsymbol{\xi} \in \mathbb{R}^D, \rho, \beta$, and $\mathbf{W} \in \mathbb{R}^{D \times D}$ are hyperparameters common to all mixture components. n_k is the occupation number, which indicates the number of points assigned to the k_{th} Gaussian component. $-i$ indicates all indices except for i, and $n_{-i,k}$ is the number of points, excluding $\mathbf{p}_i$, that are associated with the k_{th} Gaussian component. t is the *Student's t-distribution* with $\beta_n - D + 1$ degrees of freedom.

$$
\begin{aligned}
\beta_n &= \beta + n_k \\
\rho_n &= \rho + n_k \\
\boldsymbol{\xi}_* &= \frac{\rho\boldsymbol{\xi} + \sum_{i:c_i=k} \mathbf{p}_i}{\rho + n_k} \\
\mathbf{W}_* &= \mathbf{W} + \rho\boldsymbol{\xi}\boldsymbol{\xi}^T + \sum_{i:c_i=k} \mathbf{p}_i\mathbf{p}_i^T - \left(\rho + n_k\right)\boldsymbol{\xi}_*\boldsymbol{\xi}_*^T
\end{aligned}
$$

Collapsed Gibbs Sampling method is used for the inference on the model above. For a detailed sampling process, please refer to [160]. The DPGMM model can be expressed as follows. The maximum conditional posterior class probability determines the clustering to which each point belongs.

$$
\begin{aligned}
\mathbf{p}_i | c_i &\sim \mathcal{N}\left(\boldsymbol{\mu}_{c_i}, \mathbf{S}_{c_i}\right) && (7.4) \\
c_i | \boldsymbol{\pi} &\sim Please) && (7.5) \\
\boldsymbol{\pi} | \alpha &\sim Dir\left(\frac{\alpha}{K}, \cdots, \frac{\alpha}{K}\right) && (7.6) \\
\left(\boldsymbol{\mu}_k | \mathbf{S}_k, \boldsymbol{\xi}, \rho\right) &\sim \mathcal{N}\left(\boldsymbol{\xi}, \left(\rho\mathbf{S}_k\right)^{-1}\right) && (7.7) \\
\left(\mathbf{S}_k | \beta, \mathbf{W}\right) &\sim \mathcal{W}\left(\beta, \mathbf{W}\right) && (7.8) \\
\left(\boldsymbol{\mu}_k, \mathbf{S}_k\right) &\sim \mathcal{NW}\left(\boldsymbol{\xi}, \rho, \beta, \mathbf{W}\right) && (7.9) \\
\left(n_1, \cdots n_K\right) &\sim Multi\left(\pi_1, \cdots, \pi_K\right) && (7.10)
\end{aligned}
$$

where α is the concentration parameter of the Dirichlet Process, $\alpha > 0$. $\boldsymbol{\xi} \in \mathbb{R}^D, \rho, \beta$, and $\mathbf{W} \in \mathbb{R}^{D \times D}$ are hyperparameters common to all mixture components. n_k is the occupation number, which indicates the number of points assigned to the k_{th} Gaussian component. *Cat* is the *Categorical* distribution. *Dir* and *Multi* represent *Dirichlet* distribution and

Multinomial distribution respectively.

Assume the mixed point set $\mathbf{P}$ has been divided into K clusters by DPGMM. $\mathbf{P} = \{\mathbf{C}_1, \cdots, \mathbf{C}_K\}$ and $\mathbf{C}_k = \left\{\mathbf{C}_x^k, \mathbf{C}_y^k\right\}$, $k \in \{1, \cdots, K\}$. $\mathbf{C}_x^k$ and $\mathbf{C}_y^k$ are the target points sets and source points sets in cluster $\mathbf{C}_k$ respectively. The number of target points in cluster $\mathbf{C}_k$ is N_k. And the label of target points $\mathbf{x}_i$ in cluster $\mathbf{C}_k$ is denoted as $L_{\mathbf{x}_i}^k$. The K clusters are divided into $R^{'}set$ ($R^{'}$set includes target points with a single label and source points after cluster) and $D^{'}set$ based on the label category of target points $\mathbf{C}_x^k$. Especially, even if no cluster forms an $R^{'}set$ during the initial iteration, the original mixed point cloud will be divided into several clusters, and the distribution of the target point and the source point in each cluster is similar. Therefore, with the refinement and clustering of the mixed point subsets, there will always be an $R^{'}set$ where the labels of target points are unique.

$$\tau_k = \prod_{i=1}^{N_k} \delta \left(L_{\mathbf{x}_i}^k - \frac{\sum_{i=1}^{N_k} L_{\mathbf{x}_i}^k}{N_k} \right) \tag{7.11}$$

where δ is the *Dirac function*. If $\tau_k = 1$, then $\mathbf{C}_k$ belongs to $R^{'}set$, otherwise $\mathbf{C}_k$ belongs to $D^{'}set$. Eq. 7.11 indicates that the $R^{'}set$ includes those clusters where the label categories of $\mathbf{C}_y^k$ are unique. $D^{'}set$ consists of those clusters where the label categories of $\mathbf{C}_y^k$ are various.

Refine

The irregularity of the initial partitions achieved by DPGMM will affect the subsequent partition results, confusing label allocation. Furthermore, the confusion label will increase the error of registration. Thus, the *Refine* is used to prune the clusters in $R^{'}set$. The regularized clusters compose the *R set*, which will be spliced together to recover the labelled source points set $\mathbf{Y}$. The refining path and direction are designed according to the characteristics of point distribution. Then the label L_x will be assigned to the source points $\mathbf{y}_j$ in the regularized clusters.

Assume $\mathbf{C}_{\hat{k}} = [\mathbf{C}_x^{\hat{k}}, \mathbf{C}_y^{\hat{k}}]$ is a cluster in $R^{'}set$, and $\mathbf{C}_x^{\hat{k}}, \mathbf{C}_y^{\hat{k}}$ are target points and source points in $\mathbf{C}_{\hat{k}}$ respectively. The degree of dispersion of the $\mathbf{C}_x^{\hat{k}}, \mathbf{C}_y^{\hat{k}}$ in each dimension determines the path for refining points.

$$Std_{\mathbf{x}}^d = \sqrt{\frac{\sum_{i=1}^{N_k} \left(\left[\mathbf{C}_x^{\hat{k}}\right]_{i,d} - \overline{\left[\mathbf{C}_x^{\hat{k}}\right]}_{:,d} \right)}{N_k - 1}} \tag{7.12}$$

$$Std_{\mathbf{y}}^d = \sqrt{\frac{\sum_{i=1}^{n_k - N_k} \left(\left[\mathbf{C}_y^{\hat{k}}\right]_{i,d} - \overline{\left[\mathbf{C}_y^{\hat{k}}\right]}_{:,d} \right)}{N_k - 1}} \tag{7.13}$$

where $\left[\mathbf{C}^{\hat{k}}\right]_{i,d}$ represents the coordinate of the i_{th} point in the d_{th} dimension. $\overline{\left[\mathbf{C}^{\hat{k}}\right]}_{:,d}$ indicates the mean value of the coordinates of all points in the d_{th} dimension. $Std_{\mathbf{x}}^d$ and

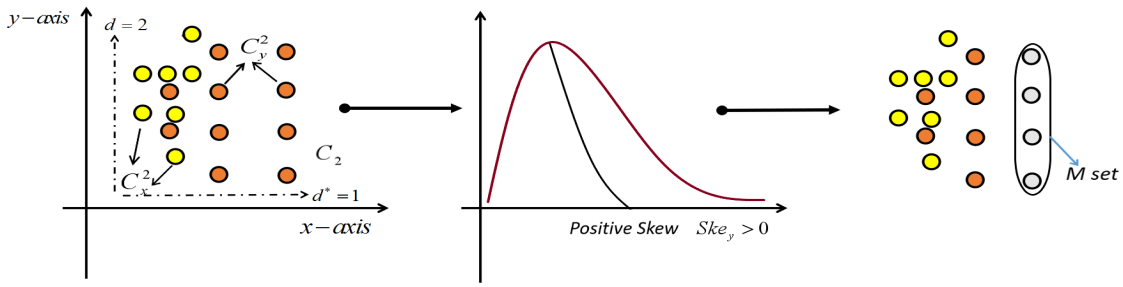


Fig. 7.6 Illustration for the impact of *skewness* on *Refine*. Due to the positive skew of the $\mathbf{C}_y^2$ on the $x-axis$, the points in the $\mathbf{C}_y^2$ with the largest x-coordinate will be transferred to the *M set*.

$Std_{\mathbf{y}}^d$ show the standard deviation of $\mathbf{C}_x^{\hat{k}}, \mathbf{C}_y^{\hat{k}}$ in d_{th} dimension, $d \in \{1,2,3\}$.

$$d^* = \underset{d\in\{1,2,3\}}{\arg\max} \left| Std_{\mathbf{x}}^d - Std_{\mathbf{y}}^d \right| \tag{7.14}$$

d^* involves the refining path, in which the object has the largest deformation. For example, the stretching of an object in one dimension will lead to its squashing in another dimension. For example, when $d^* = 1$, we will refine points along with the X-axis. If $\left| Std_{\mathbf{x}}^{d^*} - Std_{\mathbf{y}}^{d^*} \right| < \gamma$, there is no need to refine the $\mathbf{C}_{\hat{k}}$. Otherwise, the refining direction must also determined based on the refining path.

$$Ske_{\mathbf{y}} = \frac{\frac{1}{n_k - N_k} \sum_{j=1}^{n_k - N_k} \left(\left(\left[\mathbf{C}_y^{\hat{k}}\right]_{j,d^*} - \overline{\left[\mathbf{C}_y^{\hat{k}}\right]_{:,d^*}} \right) \right)^3}{\left[\frac{1}{n_k - N_k - 1} \sum_{j=1}^{n_k - N_k} \left(\left[\mathbf{C}_y^{\hat{k}}\right]_{i,d^*} - \overline{\left[\mathbf{C}_y^{\hat{k}}\right]_{:,d^*}} \right)^2 \right]^{\frac{2}{3}}} \tag{7.15}$$

$Ske_{\mathbf{y}}$ shows the skewness of the source points $\mathbf{C}_y^{\hat{k}}$ in the d_{th}^* dimension. We denote the index of the minimal $\left[\mathbf{C}_y^{\hat{k}}\right]_{j,d^*}$ as J_{min} and the index of the maximal $\left[\mathbf{C}_y^{\hat{k}}\right]_{j,d^*}$ as J_{max}. If $Ske_{\mathbf{y}} < 0$, the $\left[\mathbf{C}_y^{\hat{k}}\right]_{J_{min},:}$ will be separated from $\mathbf{C}_y^{\hat{k}}$ and moved it into a temporary set *M set*. If $Ske_{\mathbf{y}} > 0$, the $\left[\mathbf{C}_y^{\hat{k}}\right]_{J_{max},:}$ will be moved into *M set*. $\left[\mathbf{C}_y^{\hat{k}}\right]_{j,:}$ represents the j_{th} point in $\mathbf{C}_y^{\hat{k}}$. It is worth noting that the initial *M set* is an empty set.

Fig. 7.6 takes the C_2 in Fig. 7.5 as an example to illustrate the relationship between $Ske_{\mathbf{y}}$ and *M set*. Because of the positive skewness of the source points $\mathbf{C}_y^2$ on the $x-axis$, the source points with the maximal x-coordinate are transferred to the *M set*.

Repeat calculating the Eq. 7.12, Eq. 7.13, Eq. 7.14, and Eq. 7.15 to determine the refining path and direction until $\left| Std_{\mathbf{x}}^{d^*} - Std_{\mathbf{y}}^{d^*} \right| < \gamma$. The pseudocode for refining the $\mathbf{C}_y^{\hat{k}}$ in $R^{'}set$ is shown in Alg. 2.

After being separated from $R^{'}set$, *M set* will be merged with $D^{'}set$ to generate the branch node *D set* to be divided again. Also, the labels of $\mathbf{C}_x^{\hat{k}}$ will be assigned to $\mathbf{C}_y^{\hat{k}}$. The process for clustering and refining will be repeated until there are no points in *D set* to be divided.

Algorithm 2: Refine $\mathbf{C}_y^{\hat{k}}$ in *R set*

Input: $\mathbf{C}_x^{\hat{k}}, \mathbf{C}_y^{\hat{k}}, \gamma$
Output: $\mathbf{C}_y^{\hat{k}}$, *M set*
Initialize $M = \varnothing$
repeat
  Calculate Eq. 7.12, Eq. 7.13 and Eq. 7.14 to get $Std_{\mathbf{x}}^{d}, Std_{\mathbf{y}}^{d}, d^{*}$.
  if $\left|Std_{\mathbf{x}}^{d^*} - Std_{\mathbf{y}}^{d^*}\right| > \gamma$ **then**
    Calculate Eq. 7.15 to get $Ske_{\mathbf{y}}$ and J_{min}, J_{max}; **if** $Ske_{\mathbf{y}} < 0$ **then**
      $M\ set = \left[M\ set; \left[\mathbf{C}_y^{\hat{k}}\right]_{J_{min},:}\right]; \left[\mathbf{C}_y^{\hat{k}}\right]_{J_{min},:} = \varnothing;$
    else
      $M\ set = \left[M\ set; \left[\mathbf{C}_y^{\hat{k}}\right]_{J_{max},:}\right]; \left[\mathbf{C}_y^{\hat{k}}\right]_{J_{max},:} = \varnothing;$
until $\left|Std_{\mathbf{x}}^{d^*} - Std_{\mathbf{y}}^{d^*}\right| < \gamma;$

As for *Splice*, it is a process to reassemble the labelled source points set $\mathbf{Y}$ and the target points set $\mathbf{X}$. The source group with label l can be denoted as $\mathbf{G}_{\mathbf{y}}^{l}$ and the target group with label l can be denoted as $\mathbf{G}_{\mathbf{x}}^{l}$. Splicing the *R set* together can reduce the number of partitions and achieve less computational cost for registration to improve registration accuracy.

7.3.3 Registration

Bayesian coherent point drift (BCPD) algorithm [89] is used to register the source group $\mathbf{G}_{\mathbf{y}}^{l}$ and target group $\mathbf{G}_{\mathbf{x}}^{l}$. BCPD is the Bayesian formulation of the Coherent Point Drift (CPD) [85]. The key difference between BCPD and CPD is that BCPD defines motion coherence using a prior distribution instead of the regularization term in CPD. Besides, the transformation model in BCPD is a combination of non-rigid and similarity transformations, which enables it to handle the registration task with large deformation. As for the computational time, BCPD uses the Nyström method [161] and the KD tree search [162] to accelerate registration without losing registration accuracy.
The key models in the BCPD algorithm can be generalized as follows:

Transformation model:

$$\boldsymbol{\tau}(\mathbf{y}_i) = T(\mathbf{y}_i + \mathbf{v}_i) = s\mathbf{R}(\mathbf{y}_i + \mathbf{v}_i) + \mathbf{t} \tag{7.16}$$

where $s \in \mathbb{R}$ is a scale factor, $\mathbf{R} \in \mathbb{R}^{D\times D}$ is a rotation matrix, $\mathbf{t} \in \mathbb{R}^{D}$ is a translation vector, and $\mathbf{v}_i \in \mathbb{R}^{D}$ is a displacement vector that characterizes a non-rigid transformation.

Prior distribution:

$$p(\mathbf{v}|\mathbf{y}) = \phi\left(\mathbf{v}; \mathbf{0}, \lambda^{-1}\mathbf{G} \otimes \mathbf{I}_D\right) \tag{7.17}$$

where λ is a positive constant and $\otimes$ denotes the Kronecker product. $\mathbf{G} = \left(g_{mm'}\right) \in \mathbb{R}^{M\times M}$ with $g_{mm'} = \mathscr{K}\left(\mathbf{y}_m, \mathbf{y}_{m'}\right)$ is a positive definite matrix, where $\mathscr{K}(,)$ is a positive-definite kernel. $\phi\left(\mathbf{v}; \mathbf{0}, \lambda^{-1}\mathbf{G} \otimes \mathbf{I}_D\right)$ is the *multivariate normal distribution* of $\mathbf{v}$ with a mean vector $\mathbf{0}$ and a covariance matrix $\lambda^{-1}\mathbf{G} \otimes \mathbf{I}_D$.

7.4 Experiment

The proposed approach is evaluated by performing experiments on five data sets, three public data sets and two of our own data sets. The experiments are implemented on Intel Xeon CPU E5-1603 @ 2.80GHz and 32G RAM.

There are three groups of parameters in the proposed framework: the maximum distance between neighbouring points d_{max} for extracting boundaries in *Assigning labels* module; the hyperparameters $\{\alpha, \rho, \beta, D\}$ for *Cluster* and γ for *Refine* in *Break and Splice* module; and λ for registration. These three groups of parameters empirically are set as follows in experiments: $d_{max} = 0.03$; $\alpha = 1, \rho = 1, \beta = 3, D = 3$; $\gamma = 10^{-3}$ and $\lambda = 10$. In addition, these data sets are filtered by down-sampling to compare the BCPD method and tend to find the distribution and change of every point after registration. Its value is 0.002. At last, these data sets are without background since the proposed non-rigid registration focuses on the deformed objects, and the background does not include the deformation. On the other hand, it is to compare FB-Warp without any background.

In this research, two frames are separated by 20 frames at least are regarded as large inter-frame motions, and the gap between objects usually is large on [163], [10], and RGB-D data sets created by ourselves. The proposed method is tested against the BCPD method [89] and the FB-Warp [90] for separation and connection problems with large inter-frame motion. The proposed algorithm and the FB-Warp are compared with quantitative evaluation. In terms of performance metrics, the accuracy of registration is measured by Root Mean Square Error (RMSE), Angular Similarity (AS) [164], Structural Similarity using the colour-based feature (SSIM) [165], and the computation time measures the efficiency of the algorithms. In these experiments, only the point sets are used to register.

7.4.1 Non-rigid Registration

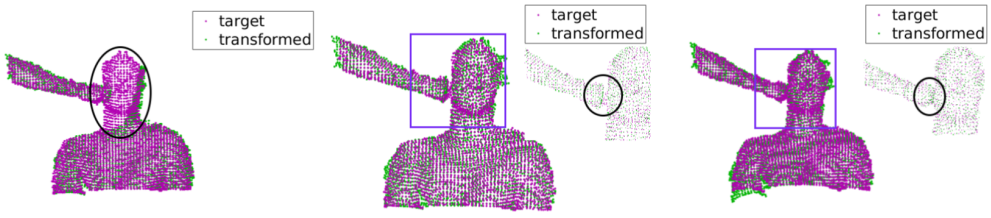
The data sets with dynamic scene topology changes are used to evaluate the proposed algorithm on large inter-frame motions. A boxing sequence in [163] as shown in Fig. 7.7 and sequences of Hat and Alex in [10] as shown in Fig. 7.9 and Fig. 7.11 are selected because these public data sets include both separations of two objects and deformations. In addition, RGB-D data sets are created by Kinect v2 as shown in Fig. 7.13 and Fig. 7.15 to demonstrate a rigid bunny separated from a table surface and scenes that consists of the separation of a deformed soft pillow from a table surface. Finally, an experiment with the connection (shown in Fig. 7.16) is conducted on the pillow data set.

Fig. 7.7 shows the results of boxing, and source points (frame 95) and target points (frame 130) are obtained by down-sampling without any background. Significantly, body

(a) Source image (frame 98) (b) Source points (c) Target image (frame 130) (d) Target points

(e) FB-Warp (f) BCPD (g) Ours

Fig. 7.7 The results of Boxing data: (a) and (c) are colour images. (b) and (d) are their corresponding point sets. The second row shows the result of registration from source points to target points, and the blue areas show the main differences: (e) uses the method of FB-Warp, (f) uses the method of BCPD without *Break and Splice*, and (g) is our algorithm.



(a) FB-Warp (b) BCPD (c) Ours

Fig. 7.8 The results of connection registration on Boxing data and the source point set and target point set in Fig. 7.7 are exchanged (Fig. 7.7 (d) as the source point set): (a) use the method of FB-Warp, (b) uses the method of BCPD without *Break and Splice*, and (c) is our algorithm.

deformation also occurs in addition to the separation. The second row of Fig. 7.7 shows the comparison of registration results. It can be seen that BCPD and FB-Warp are unable to handle registration with the separation, as shown in the final results containing points between the fist and the face (green points). However, the proposed method can effectively register the source and the target points. Similarly, the results of Alex sequences (Fig. 7.9) and Hat data sets (Fig. 7.11) show that the proposed *Break and Splice* framework can achieve a better result than others.

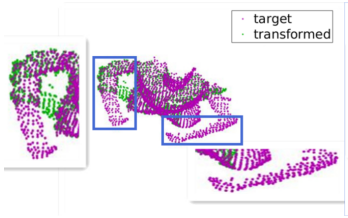
In addition, to experiment with the connection, the source point set and target point set are exchanged, and the results are shown in Fig. 7.8(Boxing data), Fig. 7.10 (Alex data), Fig. 7.12 (Hat data). The green points in Fig. 7.8 (a) on the face are only located on the edge. Although the result of BCPD shows good registration, between the fist and the face only show the target points (the black circle in Fig. 7.8 (b)) compared with the proposed. For the Alex and Hat data results, there are a few points between the topology changes, but the distribution is uniform in the results of the proposed method.

(a) Source image (frame 246)

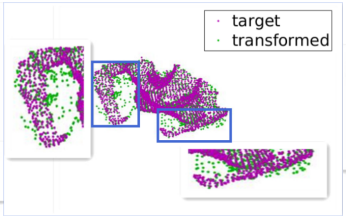
(b) Source points

(c) Target image (frame 273)

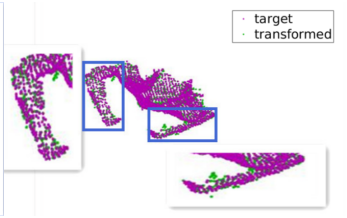
(d) Target points



(e) FB-Warp



(f) BCPD



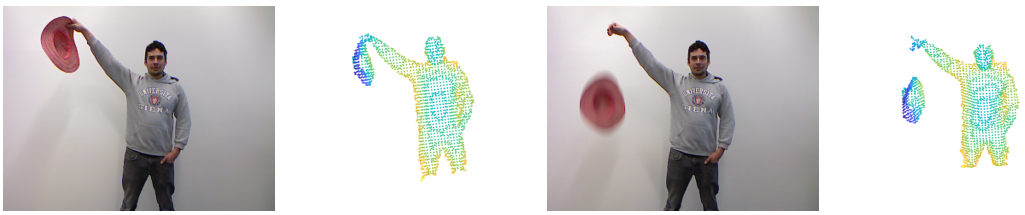
(g) Ours

Fig. 7.9 The results of Alex data: (a) and (c) are colour images. (b) and (d) are their corresponding point sets. The second row shows the results of registration from source points to target points, and the blue areas show the main differences: (e) uses the method of FB-Warp, (f) uses the method of BCPD without *Break and Splice*, and (g) is our algorithm.

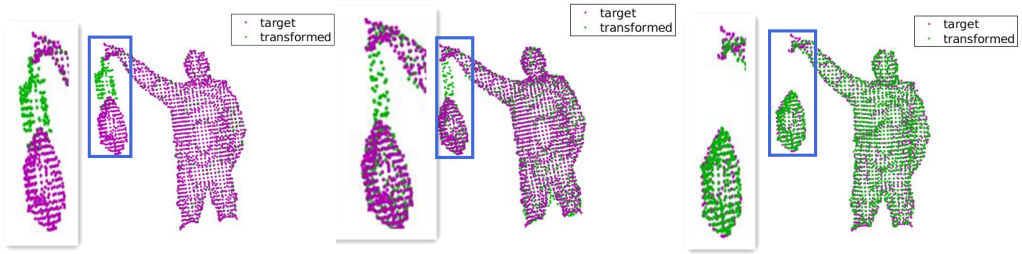
In order to demonstrate the effectiveness of our *Break and Splice* framework in dealing with different scenes, an RGB-D camera is used to create additional point cloud data sets and compare the proposed method with BCPD and FB-Warp on these data sets. A rigid object (bunny, Fig. 7.13) and a non-rigid object (pillow, Fig. 7.15) are used to conduct the experiment. Fig. 7.13 shows that when the bunny is placed far away from the table, this final state is regarded as target points and its initial state as source points. Fig. 7.13 (e) shows that although the result of FB-Warp contains no points between the bunny and the table, many green points are distributed on the boundary of the transformed bunny points. The BCPD have many green points between the bunny and the table, as shown in Fig. 7.13 (f). The result of the proposed method shown in Fig. 7.13 (g) has well-distributed transformed points without any points between the object and the table.

(a) FB-Warp (b) BCPD (c) Ours

Fig. 7.10 The results of connection registration on Alex data and the source point set and target point set in Fig. 7.9 are exchanged(Fig. 7.9 (d) as the source point set): (a) uses the method of FB-Warp, (b) uses the method of BCPD without *Break and Splice*, and (c) is our algorithm.

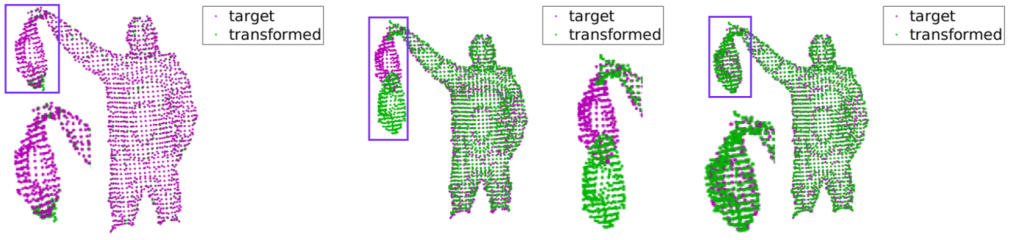


(a) Source image (frame 69) (b) Source points (c) Target image (frame 89) (d) Target points



(e) FB-Warp (f) BCPD (g) Ours

Fig. 7.11 The results of Hat data: (a) and (c) are colour images. (b) and (d) are their corresponding point sets. The second row shows the results of registration from source points to target points, and the blue areas show the main differences: (e) uses the method of FB-Warp, (f) uses the method of BCPD without *Break and Splice*, and (g) is our algorithm.



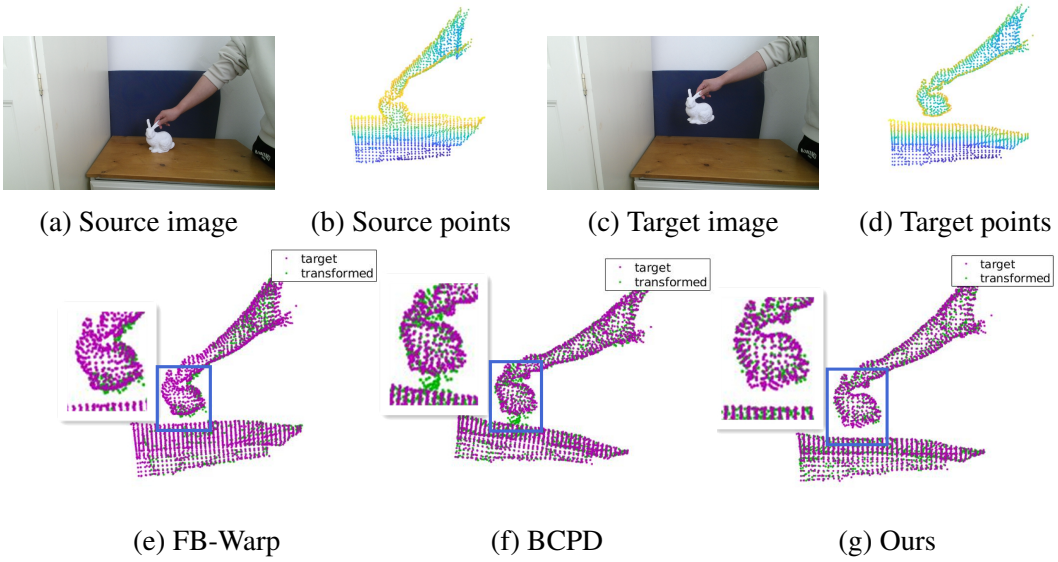
(a) FB-Warp (b) BCPD (c) Ours

Fig. 7.12 The results of connection registration on Hat data and the source point set and target point set in Fig. 7.11 are exchanged(Fig. 7.11 (d) as the source point set): (a) uses the method of FB-Warp, (b) uses the method of BCPD without *Break and Splice*, and (c) is our algorithm.

For the connection, Fig. 7.13 (a) is used as the target point set to register. Our method can achieve a suitable result, especially for the bunny. The result of FB-Warp shows most of the green points of the bunny distribute the head of the bunny, and the result of BCPD fails to register the bunny(in Fig. 7.14 (b))

In the Pillow experiment (Fig. 7.15), the data set simultaneously includes significant deformation and separation. The proposed method produces a better result than that of FB-Warp and BCPD. In the FB-Warp, the transformed points appear at the bottom of the pillow, and some points remain between the pillow and the table. In Fig. 7.15 (f), although the transformed points are well-distributed for a pillow, it fails to handle the separation between the pillow and the table. Whereas in Fig. 7.15 (g), our *Break and Splice* can effectively deal with combined deformation and separation events and yield a significantly improved result.

In addition, the source point set and target point set are exchanged to experiment with connection and the results are shown in Fig. 7.16. It can be seen that the BCPD and our method can achieve good registration with well-distributed green points, but the FB-Warp has a few green points (transformed point set) on the bottom half of the pillow.



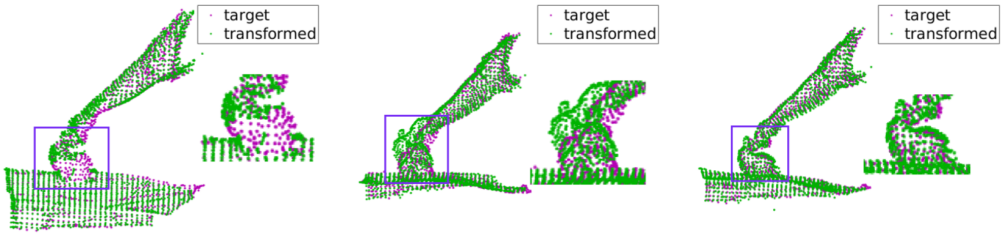
(a) Source image (b) Source points (c) Target image (d) Target points

(e) FB-Warp (f) BCPD (g) Ours

Fig. 7.13 The results of our data set with Bunny: (a) and (c) are colour images. (b) and (d) are their corresponding point sets. The second row shows the results of registration from source points to target points, and the blue areas show the main differences: (e) uses the method of FB-Warp, (f) uses the method of BCPD without *Break and Splice*, and (g) is our algorithm.

7.4.2 Quantitative Evaluation

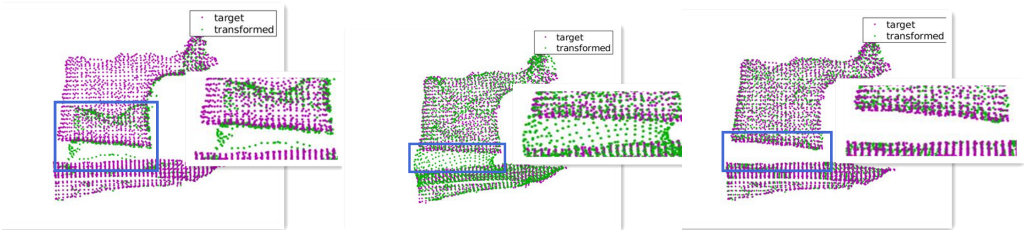
The accuracy and the cost of computation of our non-rigid registration framework are evaluated using RMSE (Eq. 7.18), AS (Eq. 7.19, Eq. 7.20 and Eq. 7.21) and SSIM (Eq. 7.22) as measurement metrics tested on two consecutive frames (Table 7.2) and large inter-frame motions (Table 7.3) with topology changes. RMSE efficiently measures the registration error. Since the ground truth of non-rigid registration is the target point



(a) FB-Warp (b) BCPD (c) Ours

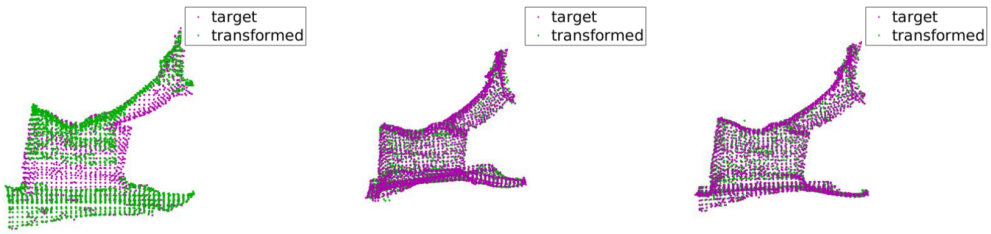
Fig. 7.14 The results of connection registration on Bunny data and the source point set and target point set in Fig. 7.13 are exchanged(Fig. 7.13 (d) as the source point set): (a) uses the method of FB-Warp, (b) uses the method of BCPD without *Break and Splice*, and (c) is our algorithm.

(a) Source image (b) Source points (c) Target image (d) Target points



(e) FB-Warp (f) BCPD (g) Ours

Fig. 7.15 The results of our data set with Pillow: (a) and (c) are colour images. (b) and (d) are their corresponding point sets. The second row shows the results of registration from source points to target points, and the blue areas show the main differences: (e) uses the method of FB-Warp, (f) uses the method of BCPD without *Break and Splice*, and (g) is our algorithm.



(a) FB-Warp (b) BCPD (c) Ours

Fig. 7.16 The results of connection registration on pillow data and the source point set and target point set in Fig. 7.15 are exchanged(Fig. 7.15 (d) as the source point set): (a) uses the method of FB-Warp, (b) uses the method of BCPD without *Break and Splice*, and (c) is our algorithm.

Table 7.1 Number of Point Sets

	Alex	Boxing	Hat	Bunny	Pillow
Source point set	1264	2813	1683	2411	2244
Target point set	1294	2786	1731	2145	2504

set, The AS can be used to compare the similarity between the target point set and the transformed point set. If the value of AS is closer to 1, the transformed point sets are the more similar target point sets. The similarity of colour can also be an important metric because the colour is not influenced by registration, and it is easy to find the difference between transformed point sets and target point sets. Table 7.1 shows the number of point sets during the experiment.

$$RMSD = \sqrt{\frac{\sum_{i=1}^{N} (y_i - x_i)^2}{N}} \tag{7.18}$$

where y_i is transformed point sets, x_i is a target point sets, N is the number of target point sets.

$$s_{X,Y} = \sqrt{\frac{\sum_{i=1}^{M} \left(1 - \frac{2}{\pi} * arccos(\left| \frac{\vec{n}_i^y \cdot \vec{n}_i^x}{\|\vec{n}_i^y\| \|\vec{n}_i^x\|} \right|)\right)^2}{M}} \tag{7.19}$$

$$s_{Y,X} = \sqrt{\frac{\sum_{j=1}^{N} \left(1 - \frac{2}{\pi} * arccos(\left| \frac{\vec{n}_j^y \cdot \vec{n}_j^x}{\|\vec{n}_j^y\| \|\vec{n}_j^x\|} \right|)\right)^2}{N}} \tag{7.20}$$

$$AS = min\{s_{X,Y}, s_{Y,X}\} \tag{7.21}$$

where Y is the transformed point sets, X is a target point sets, $s_{X,Y}$ is the score of angular similarity with Y as the reference point set, and $s_{Y,X}$ is the score of angular similarity with X as the reference point set. $vecn^x$ and $vecn^y$ are the normal of X and Y point sets, N and M are the number of Y and X point sets.

$$SSIM_{pointcolor} = \frac{1}{N} \sum_{i=1}^{N} \left(1 - \frac{|F_X(q) - F_Y(p)|}{max\{|F_X(q)|, |F_Y(p)|\} + \varepsilon}\right) \tag{7.22}$$

where Y is the transformed point sets, X is a target point sets, F is the feature based on color [165], each neighborhood of Y is associated with a neighborhood of X, by identifying for every point p of Y its nearest point q in X. ε equals the machine rounding error for floating point numbers, and N is the number of Y point sets.

As shown in Table 7.2, our method is more accurate than FB-Warp and BCPD, and the average error is lower by about 60% than the FB-Warp. The results of the AS and the SSIM show that the average values of our method are higher by about 3.2% and 8.5%, respectively. FB-Warp can achieve better results for the Boxing data since the deformation

Table 7.2 Registration Error(consecutive frames)

	Alex	Boxing	Hat	Bunny	Pillow	Overall
RMSE(FB-Warp)	0.0170	0.0037	0.0208	0.0089	0.0526	0.0206
RMSE(BCPD)	0.0131	0.0037	0.0135	0.0181	0.0214	0.0140
RMSE(Ours)	**0.0110**	**0.0029**	**0.0092**	**0.0074**	**0.0083**	**0.0078**
AS(FB-Warp)	0.9508	**0.9923**	0.9113	0.9836	0.9577	0.9591
AS(BCPD)	0.8924	0.9105	0.8250	0.8680	0.6965	0.8385
AS(Ours)	**0.9820**	0.9839	**0.9940**	**0.9910**	**0.9964**	**0.9895**
SSIM(FB-Warp)	0.6246	**0.6711**	0.5859	0.5922	0.5120	0.5971
SSIM(BCPD)	0.6091	0.5905	0.5626	0.5740	0.4991	0.5671
SSIM(Ours)	**0.6557**	0.6571	**0.6605**	**0.6314**	**0.6360**	**0.6481**

Table 7.3 Registration Error(large inter-frame motions)

	Alex	Boxing	Hat	Bunny	Pillow	Overall
RMSE(FB-Warp)	0.7694	0.0626	0.8681	0.3337	0.3341	0.4736
RMSE(BCPD)	0.4151	0.1375	0.6034	0.1647	0.5427	0.3727
RMSE(Ours)	**0.1506**	**0.0486**	**0.1752**	**0.0995**	**0.1464**	**0.1241**
AS(FB-Warp)	0.7303	0.7711	0.8426	0.7932	0.7215	0.77174
AS(BCPD)	0.8142	0.8462	0.8374	0.7932	0.7605	0.8103
AS(Ours)	**0.8377**	**0.8632**	**0.8611**	**0.8407**	**0.8443**	**0.8494**
SSIM(FB-Warp)	0.4776	0.6277	0.6531	0.6148	0.5412	0.5829
SSIM(BCPD)	0.6071	0.6412	0.6210	0.6392	0.5910	0.6199
SSIM(Ours)	**0.6136**	**0.6458**	**0.6689**	**0.6698**	**0.6249**	**0.6446**

is slight between the adjacent frames. The proposed method can achieve the best results on all data for the large inter-frame motions (Table 7.3). At the same time, registration time is lower than FB-Warp, as shown in Table 7.4 (C means consecutive frames, and L means large inter-frame motions). In this table, the Partition times are the *Break and Splice*, and the registration times are the total times of two parts registration by BCPD.

7.4.3 Evaluation with Gaussian Noise

In order to evaluate the robustness of the proposed method against noise, several experiments are conducted with Gaussian noise in the source points and target points, respectively.

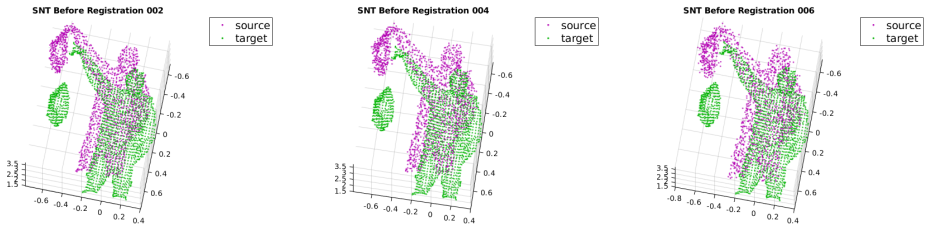
Table 7.4 Registration Time(s)

	Alex	Boxing	Hat	Bunny	Pillow	Overall
FB-Warp-C	50.2430	**15.8930**	53.7950	36.7800	76.4530	46.6328
FB-Warp-L	45.6756	**12.8875**	42.8859	25.8088	53.2297	36.0975
Ours-C	**13.6605**	18.3537	**21.6762**	**14.1161**	**28.3915**	**19.2496**
Ours-L	**10.5121**	15.0353	**12.2921**	**12.1661**	**9.8807**	**11.9773**
Partition-C	1.5958	4.8619	1.1519	0.0761	0.8829	1.7137
Partition-L	0.1990	4.0456	1.1071	1.1031	1.0238	1.4957
registration-C	12.0647	13.4918	20.5243	14.0400	27.5086	17.5259
registration-L	10.3131	10.9897	11.0590	11.1890	8.8569	10.4815

Table 7.5 Registration error with Gaussian noise

	SNT			STN			SNTN		
	FB-Warp	BCPD	Ours	FB-Warp	BCPD	Ours	FB-Warp	BCPD	Ours
0.002	0.1543	0.1358	**0.0267**	0.1546	0.1229	**0.1344**	0.1296	0.1261	**0.0274**
0.004	0.1395	0.1224	**0.0325**	0.1470	0.1268	**0.1325**	0.1334	0.1241	**0.0339**
0.006	0.1660	0.1292	**0.0448**	0.1494	0.1228	**0.1202**	0.1703	0.1269	**0.0484**

Fig. 7.11 (b) is source points, and Fig. 7.11 (d) is target points. For the first test, we sample noise from Gaussian distribution for each point in the source point cloud, where the mean and the standard deviation are $mean(sourcepoints)$ and $\alpha * std(sourcepoints)$ ($\alpha \in \{0.002, 0.004, 0.006\}$). During testing, FB-Warp and BCPD with noise in source data are compared with each algorithm. Fig. 7.17 shows the results before registration, and Fig. 7.18 shows the results of registration. The proposed method results better than FB-Warp and BCPD when α is 0.002 and 0.004. FB-Warp performs the worst, with many inaccurate points on the hat. However, when the α is 0.006, there are many wrong points between hand and hat for all methods except for 7.20 (c) STN result (source point set is original and target point set is with Gaussian noise).



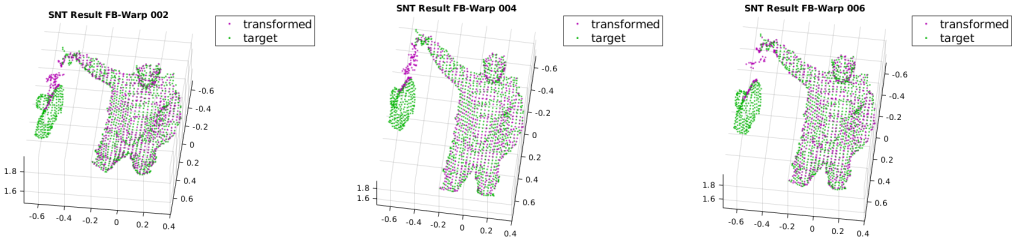
Before registration

Fig. 7.17 The source point set with noise and target point set without noise(SNT) are for different α, and figures from left to right are 0.002, 0.004 and 0.006.

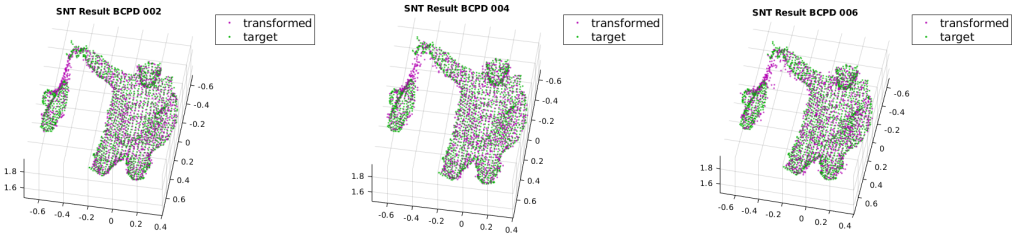
In addition, the noise of the target point cloud is sampled (Fig. 7.19), where the mean and the standard deviation are $mean(targetpoints)$ and $\alpha * std(targetpoints)$ ($\alpha \in \{0.002, 0.004, 0.006\}$). Fig. 7.20 shows the results of registration. At last, the source and target points are sampled Gaussian noise (Fig. 7.18). Fig. 7.22 shows the results of registration. The proposed method gets similar registration results, and the proposed method based on Break and Splice is robust to Gaussian noise to some degree. At the same time, the RMSE is used to evaluate different methods, and the results are shown in Table 7.5. The proposed method can lower errors in different situations.

7.4.4 Discussion

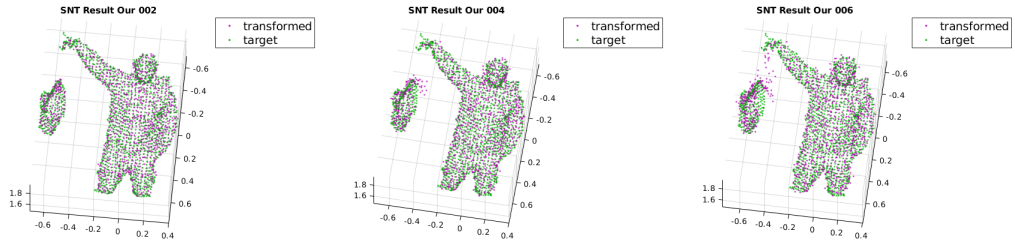
The proposed method works well, as expected, in dealing with separations and connections in dynamic scenes for point set registrations. Especially in the separation event, the proposed method achieves excellent results. Although the BCPD handles the connection event (Fig. 7.16 (b)), it fails to register the target point set (Fig. 7.15 (f)). In addition, the proposed *Break and Splice* framework achieves lower errors and fast computing time.



(a) FB-Warp

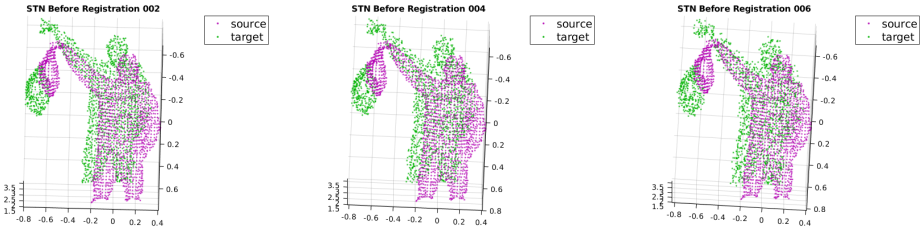


(b) BCPD



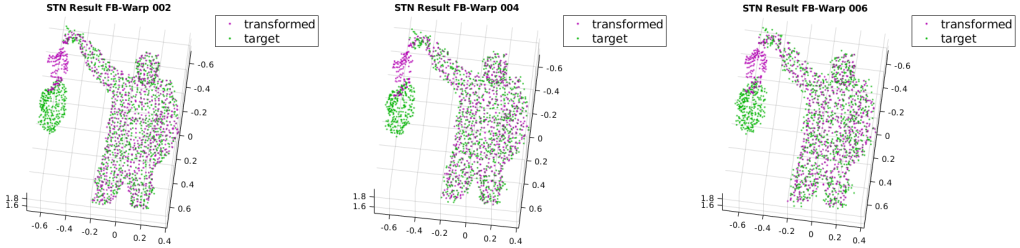
(c) Ours

Fig. 7.18 The results of registration on Hat data with noise: (a) uses the method of FB-Warp, (b) uses the method of BCPD without *Break and Splice*, and (c) is our algorithm.

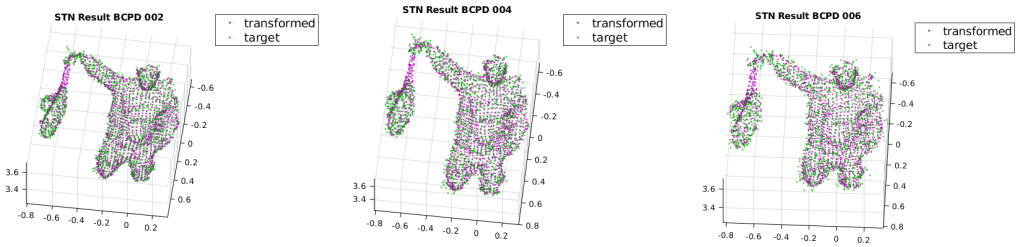


Before registration

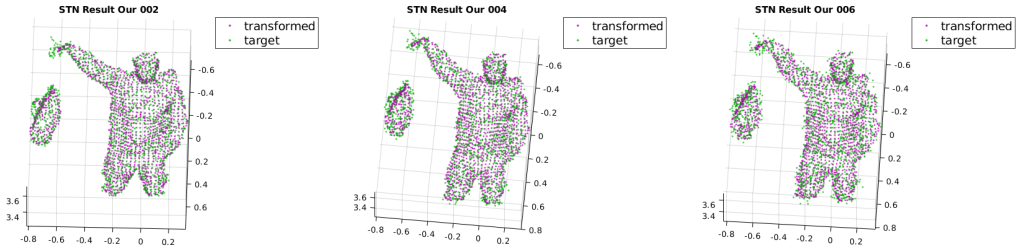
Fig. 7.19 The source point set without noise and target point set with noise (STN) are for different α, and figures from left to right are 0.002, 0.004 and 0.006.



(a) FB-Warp

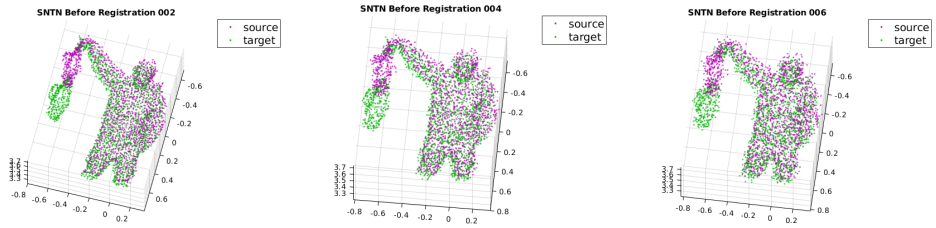


(b) BCPD



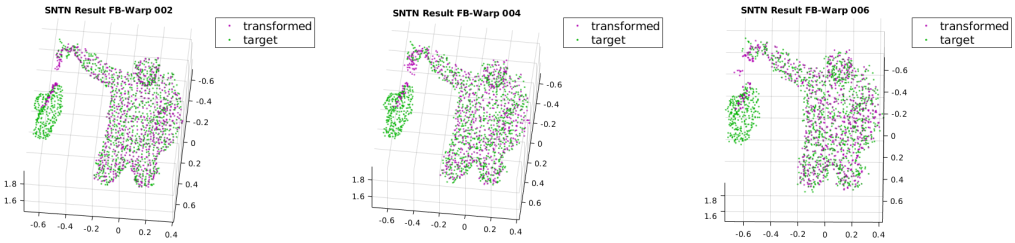
(c) Ours

Fig. 7.20 The results of registration on Hat data with noise: (a) uses the method of FB-Warp, (b) uses the method of BCPD without *Break and Splice*, and (c) is our algorithm.

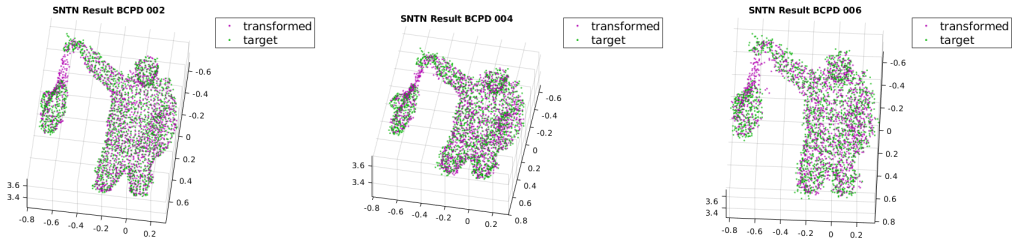


Before registration

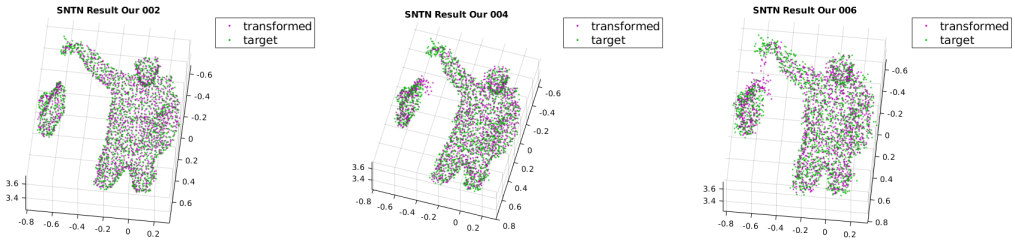
Fig. 7.21 The source point set with noise and target point set with noise (SNTN) are for different α, and figures from left to right are 0.002, 0.004 and 0.006.



(a) FB-Warp



(b) BCPD

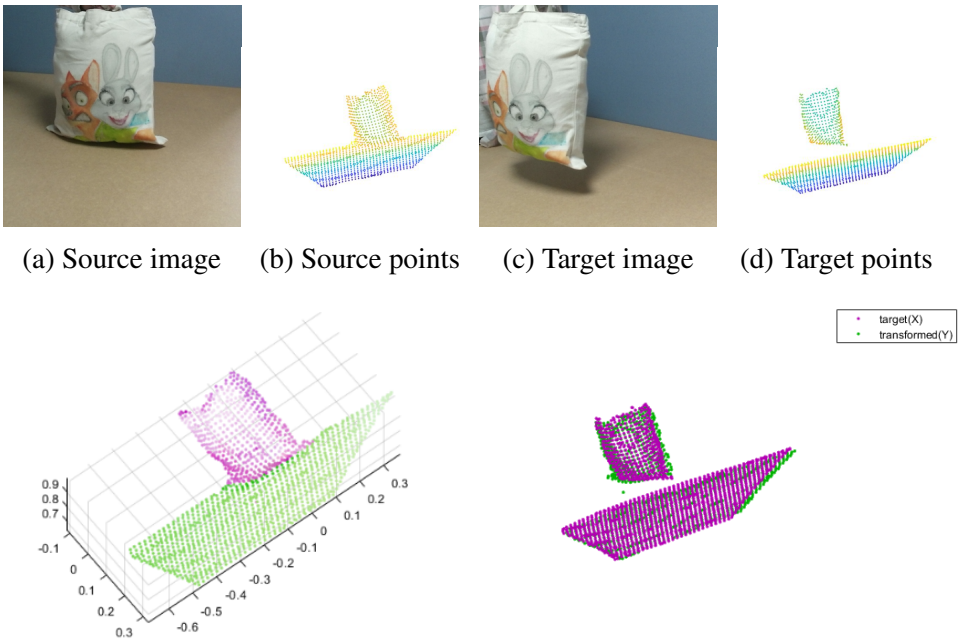


(c) Ours

Fig. 7.22 The results of registration on Hat data with noise: (a) uses the method of FB-Warp, (b) uses the method of BCPD without *Break and Splice*, and (c) is our algorithm.

In addition, there are no available public data sets with various viewpoints and special topological changes. Thus, we experiment with significant view changes. As shown in the following (Fig. 7.23 (a) and (c) are images, Fig. 7.23 (b) and (d) are point sets), the two-point clouds are acquired by moving the camera about 45 degrees from left to right as a case of significant view change. The proposed method is applied to this case, as shown in the result of labelled source points in Fig. 7.23 (e). The proposed method still works for large view changes. This is because we always find a part of the source and target points set under a common coordinate system after merging the two-point clouds by *Cluster*. Based on this part, one of the labels can be found by *Refine*, and the rest of the point cloud repeats this process (Cluster and Refine) until there is only one label in the merging point cloud. Therefore, the proposed method will not be a failure without the initial alignment.

Meanwhile, the refinement is sensitive to the parameter γ. If the value of the parameter γ is not desirable, the source point will not be refined, which will cause fewer source points to be matched with the target points at the final registration stage. However, BCPD, as an advanced non-rigid registration method, can handle the registration with an inconsistent number of point clouds. The unsatisfied result is that the dense and sparse distribution of the transformed point cloud is different from that of the target point cloud.



(a) Source image (b) Source points (c) Target image (d) Target points

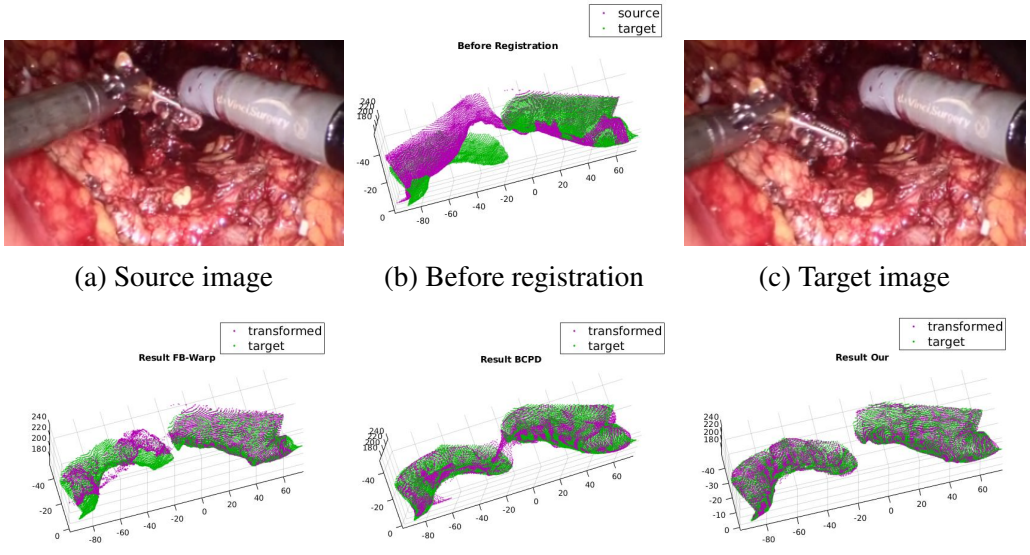
(e) labeled source points (f) our method

Fig. 7.23 The result of registration about the large view changes: (a) and (c) are colour images. (b) and (d) are their corresponding point sets. The second row shows the results of assigning labels in source points (e) and registration from source points to target points (f).

7.5 Registration of Medical Instruments

The data sets for conducting point cloud non-rigid registration experiments in the previous are captured by Kinect, which can directly obtain 3D points from the sensors. However,

it is unsuitable for applications under the MIS background. In the context of applying the proposed non-rigid registration method to MIS, one of the initial steps would involve obtaining a point cloud representation of the surgical scene. Therefore, the point clouds are estimated by the method in Chapter 4, which can accurately recover the depth of instruments. Since the proposed method needs to be done without background, a segmentation method [166] based on the image is used to compute the position of instruments. Then, the point clouds of instruments without a background are through combining the results of segmentation and point clouds from images, as shown in Fig. 7.24(b). Similar to before, FB-Warp and BCPD are used to compare with the proposed method. It can be seen that Fig. 7.24(d) and Fig. 7.24(e) fail to register with the topological changes, and Fig. 7.24(f) achieves a better result than others.



(a) Source image (b) Before registration (c) Target image

(d) FB-Warp (e) BCPD (f) Ours

Fig. 7.24 The results of the MIS data set with instruments: (a) and (c) are colour images. (b) is point sets before registration. The second row shows the results of registration from source points to target points; (d) uses the method of FB-Warp, (e) uses the method of BCPD without *Break and Splice*, and (f) is our algorithm.

7.6 Conclusion

In this chapter, a novel non-rigid point cloud registration framework is presented that handles separation and connection topology changes. The *Break and Splice* framework allows clustering and refinement of point sets to overcome distribution irregularities of the point sets, which can improve the accuracy of non-rigid registration efficiently. Experiment results have shown that the proposed method aligns two-point sets with these topology changes more effectively than the state-of-the-art approaches.

Currently, the framework does not take into account RGB information. Thus, no texture information is included in the results. Another issue is that the proposed approach requires parts of the two-point clouds to be found in a common coordinate system, which

(a) Source points (b) Target points

Fig. 7.25 An example for the cluster. (a) is a source point set, and (b) is a target point set that a bunny is separated from the table.

must include the parts of the source point set and target point set simultaneously. If this condition is not met, for example, in the case of large deformation, some large camera movements may cause a failure on the label onto Y in Fig. 7.2. Especially, the proposed method cannot handle more clusters or self-occlusion data due to getting inaccurate labels, which are based on 2D boundary extraction. For example, Fig. 7.25 shows a dragon behind the bunny. When the bunny is separated from the table, it is difficult to find the boundary of the bunny since the bunny and the leg of the dragon will be regarded as an object.

At last, if the proposed non-rigid registration framework is applied to AR applications in the next chapter, there are several problems to be handled. Firstly, the registration time is still long and cannot reach real-time. Secondly, how to calculate the camera pose(used to ensure stable registration of the virtual objects), but the proposed method only provides the change of points. Finally, multi-frame non-rigid fusion or registration based on the proposed method for reconstructing the map, which is used to provide an accurate 3D position for a virtual object, remains a challenge. Therefore, the next chapter only uses the proposed monocular depth estimation for AR applications.

Chapter 8

Topology-aware AR Applications under Minimally Invasive Surgery

8.1 Background and Motivation

This chapter will introduce two AR applications based on the proposed depth estimation framework on endoscopy videos, which is often applied in Minimally Invasive Surgery (MIS). MIS involves performing surgical procedures through small incisions, resulting in minimal tissue damage and faster recovery compared to traditional surgery. During MIS, a laparoscope—a thin, tube-like instrument with a light and a lens—is inserted through one incision to provide visual guidance. In contrast, tiny surgical instruments are inserted through other openings for the actual surgical procedure. Challenges associated with MIS can lead to procedure failures. These challenges include limited field of view [167], and MIS procedures often rely on endoscopic cameras that provide only 2D information, lacking depth perception. Therefore, obtaining accurate positional information on organs or instruments during MIS is an active research topic. In addition, telesurgical robotics, which is a technical solution for Robot-Assisted Minimally Invasive Surgery (RAMIS), has the potential to achieve widespread global clinical adoption [168]. Robotic surgery may be physically separated from the patient, and the surgical instruments are under direct guidance and remotely controlled by human operators. RAMIS usually depends on real-time display, using an endoscopic camera, which provides a video stream as the primary sensory feedback from the surgical site [168]. A key challenge of image-guided RAMIS is that clinicians need to imagine the real distance between organs through a video stream. Therefore, accurate positioning may assist surgical operations and contribute to the development of automated robotic surgery. Augmented Reality (AR) technology can provide a technical solution to address these challenges. In this chapter, a topology-aware AR application in MIS is presented. The proposed GNN-based monocular depth estimation framework, which can accurately reconstruct the depth of instruments, is applied in the context of topology-aware AR applications. An AR framework will be first introduced. Secondly, experiments on the distance tracking between instruments and between the instrument and an organ are described. Finally, an AR system framework is implemented

to demonstrate a practical use case of the computational models of this PhD work in the context of AR technology.

8.2 Method

The flowchart in Fig. 8.1 shows an AR application in MIS. The proposed monocular depth estimation algorithm generates an unorganized and dense point cloud. The 3D mesh is built based on the point cloud by Poisson surface reconstruction [169]. The mesh surface is projected into the input image plane via a camera matrix and space transformation. In addition, the input image is segmented by SiamMask [166] and calculates the centre of instruments. The relative distance between the two instruments can be obtained by combining the point cloud and the instrument's centre. Finally, the virtual augmentation information can be displayed on the reconstructed mesh surface.

8.2.1 Coordinates Transformation

AR aims to register virtual objects in the real world. To achieve this, coordinate transformations are necessary to ensure that virtual objects are aligned correctly with the physical environment. In AR systems based on 3D maps, there are three Coordinates: camera coordinates, world coordinates and model coordinates. Their relations are shown in Fig. 8.2. Assuming P_m is a 3D point in model coordinates, then the 3D point can be transformed into image plane point p_c under the camera coordinates by the following equation:

$$p_c = \mathbf{K}\mathbf{T}P_m \tag{8.1}$$

where $\mathbf{K}$ is the camera intrinsic matrix and $\mathbf{T} = T_{WC}T_{WM}$. In the following experiments, camera coordinates and world coordinates are under the same coordinates $T_{WC} = I$ due to the fixed camera view. Therefore, T_{WM} can be written to T_{CM}, and Eq. 8.1 can be expressed as:

$$p_c = \mathbf{K}T_{CM}P_m \tag{8.2}$$

where T_{CM} can be obtained by solving the Perspective-n-Point (PnP). It is similar to Eq. **??** mentioned in Chapter 3.

8.2.2 Segmentation

The purpose of segmentation is to obtain the 3D position of instruments in a MIS scene. The SiamMask provided a solution using a semi-supervised method, including 2D object tracking and segmentation. After a single bounding box initialisation, SiamMask can produce an accurate object segmentation mask, as shown in Fig. 8.3. According to the masks, the centre 2D positions can be computed. Since the estimation point cloud is unorganized, and it's impossible to find the corresponding 3D points based on the 2D

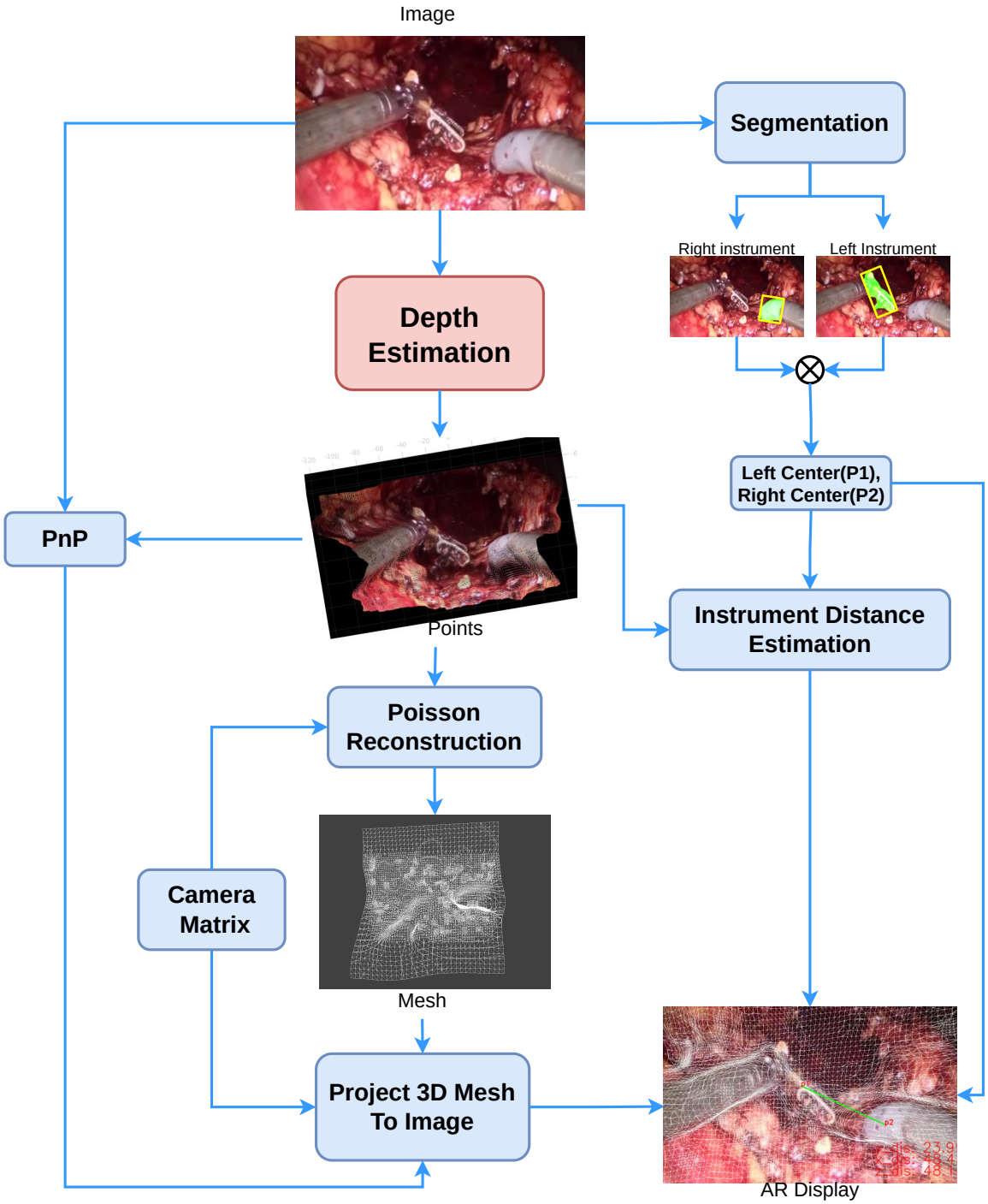


Fig. 8.1 The flowchart of the proposed AR framework.

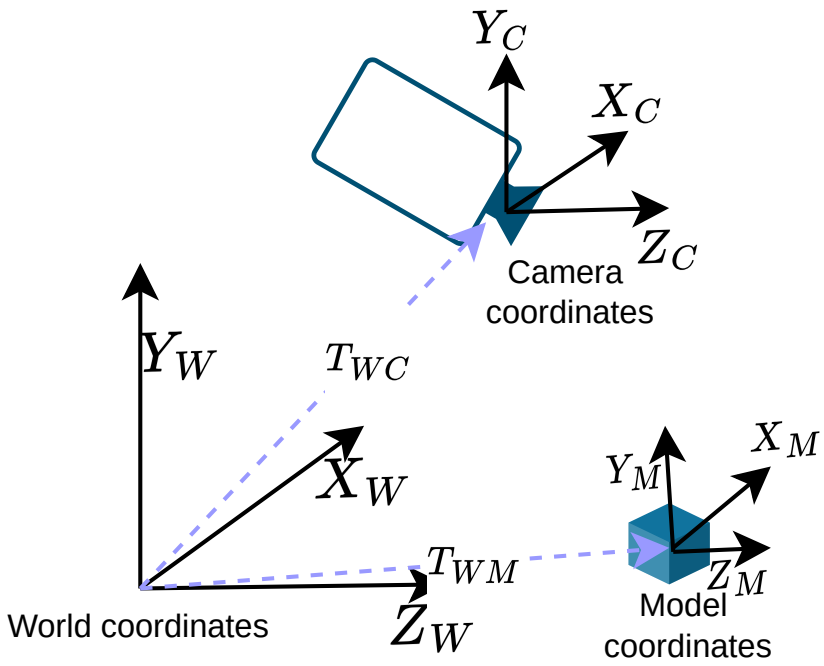
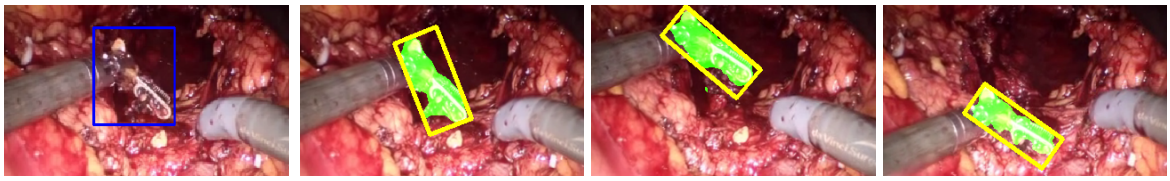


Fig. 8.2 Coordinates Transformation.



Init Estimates

Fig. 8.3 The results of SiamMask. The initial image is obtained by the user, and others are the results of estimation.

image positions, we can obtain an organized point cloud using the 'reshape' function [170]. Organized point clouds typically have the same height and width as the images, which can find the 3D points via corresponding 2D positions of images. In the following experiments, the average value of the mask region is used as the centre of instruments.

8.2.3 Poisson Surface Reconstruction

The Poisson surface reconstruction method is an implicit surface representation. This implicit representation allows for smooth and continuous surfaces without the need for explicit meshing. This method also is inherently less sensitive to noise and outliers. The Poisson surface reconstruction defines an implicit function that computes an indicator function χ, where χ equals one if the points are inside the model and equals 0 at the outside points. Therefore, the indicator function can be reduced to find the χ whose gradient best approximates a vector field $\vec{V} \in \mathbb{R}^3$:

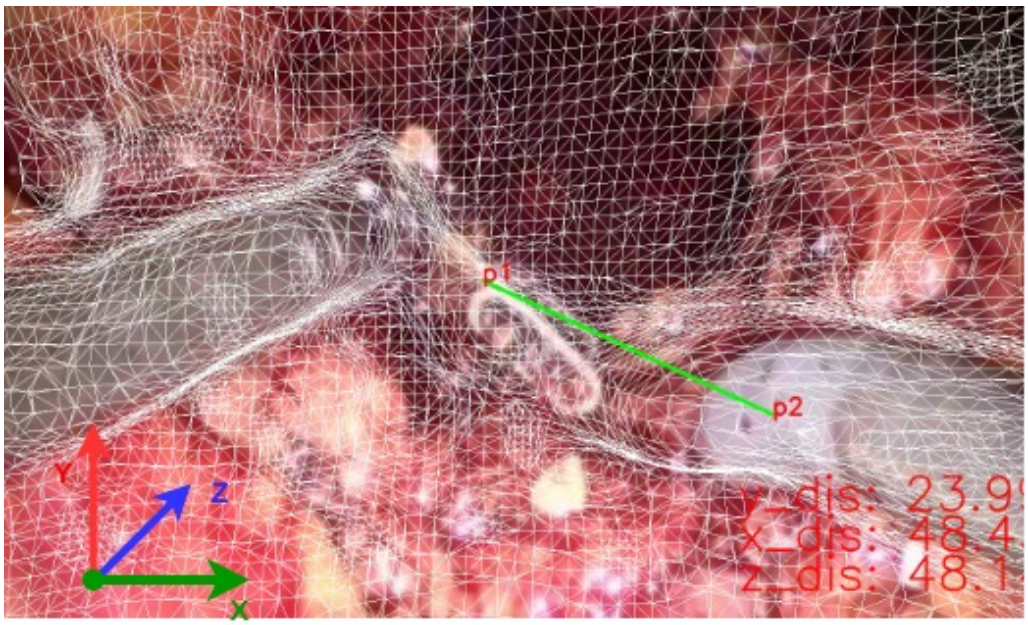


Fig. 8.4 The result of AR on *DaVinci* datasets.

$$min_{\chi} \left\| \nabla \chi - \vec{V} \right\| \tag{8.3}$$

where $min \|\cdot\|$ is to solve for the scalar function $\chi : \mathbb{R}^3 \rightarrow \mathbb{R}$ minimizing Then, it can be transformed into a Poisson problem by applying the divergence operator, computing the scalar function $\triangle \chi$ where *bigtriangleup* is the Laplace operator:

$$\triangle \chi = \nabla \cdot \nabla \chi = \nabla \cdot \vec{V} \tag{8.4}$$

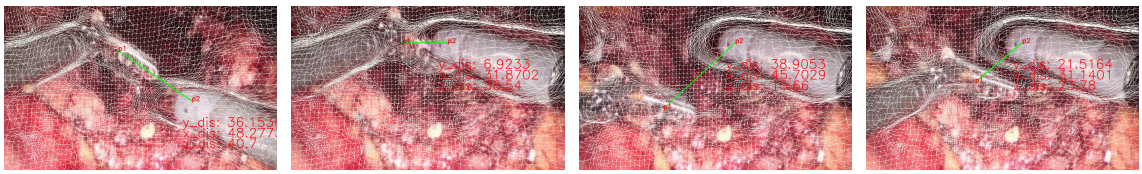
Finally, the 3D surface can be obtained by extracting an isosurface of the resulting indicator function [171]. Since Poisson reconstruction is a global solution that takes into account all the data simultaneously, it produces smooth surfaces that robustly approximate noisy data.

8.3 Experiments

The proposed AR framework is applied to two examples. The one is a real video acquired from *DaVinci* datasets [128], including the movement of two instruments. Another is that two robotic surgery videos with instruments are used to conduct experiments [11]. It offers examples of how 3D dynamic reconstruction can be applied in AR. The AR framework is implemented in an Ubuntu 18.04 environment. The size of image sequences is 1280×768 pixels for all experiments.

8.3.1 Instruments Tracking Based on Depth Map

Endoscopic navigation systems are crucial in achieving precision medicine and enhancing surgical safety, and the accurate localization of instruments will be helpful for diagnosing



(a) Frame 36 (b) Frame 48 (c) Frame 60 (d) Frame 72

Fig. 8.5 The results of AR on *DaVinci* datasets from Frame 36 to Frame 72.

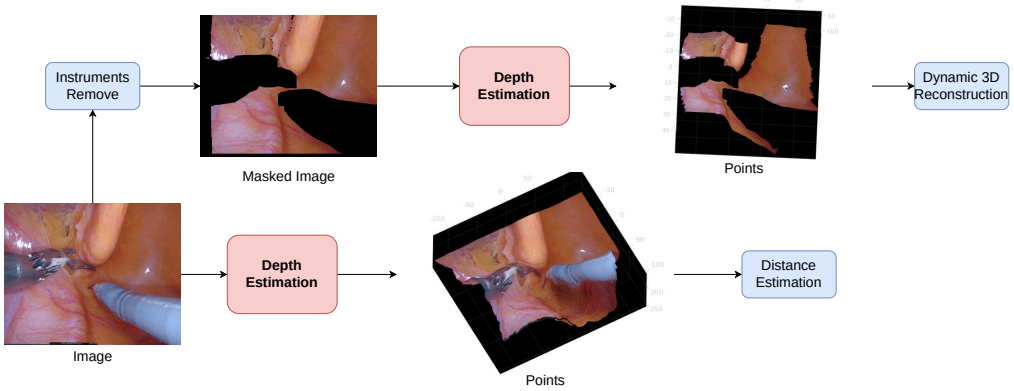
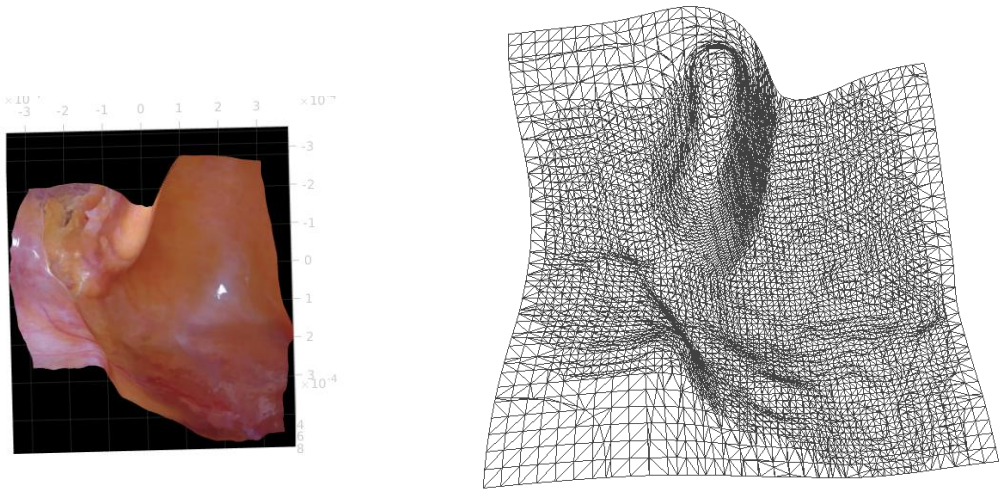


Fig. 8.6 The flowchart of AR on Pull dataset [11].

and treating during MIS. Therefore, a video capture from *DaVinci* datasets is used for the AR application based on the proposed AR framework, as shown in Fig. 8.4. The coordinates are shown in the corner of the left of Fig. 8.4. The distance for different directions can be computed based on the depth estimation. Since the depth obtained by the proposed depth estimation network is scale ambiguity, Fig. 8.4 only shows the distance without the scale factor. It can be seen that the value of distance on Y's axe is small, which is similar to the real value. It provides valuable feedback information for RAMIS and clinicians. The mesh (white line shown in Fig. 8.4) can also provide 3D information for the registration of virtual information with patient-specific data. Fig. 8.5 shows more results for different positions of instruments from Frame 36 to Frame 72.

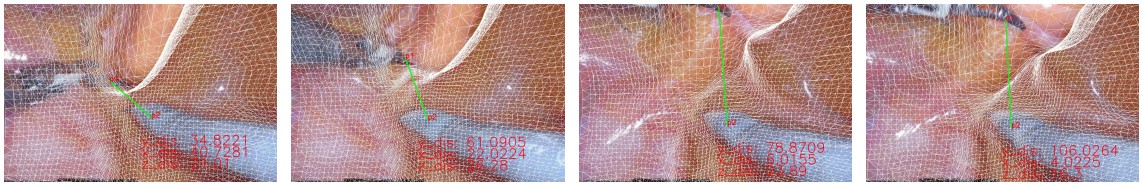
8.3.2 Organ 3D Dynamic Reconstruction for AR

In some scenarios, the distance between instruments and organs is also useful for robotic surgery automation. However, since the surgical instruments always occlude part of the soft tissue, they may fail to obtain the distance when the organs are below the instruments. This issue can be solved by reconstructing the whole scene without any instruments. The [11] dataset includes two cases: one contains significant tissue pushing and pulling, and another contains tissue cutting. There are some different parts compared with the aforementioned AR framework. Firstly, all instruments of input images need to be masked based on SiamMask, and the masked images are used for depth estimation. Then, these point clouds without instruments are used to reconstruct the scene. In addition, the depth with instruments is also estimated for distance estimation. The details can be shown in Fig. 8.6. Fig. 8.7 shows the results of dynamic 3D reconstruction based on [11] and its



(a) Points (b) Mesh

Fig. 8.7 The results of dynamic 3D reconstruction without instruments on Pull dataset [11]: (a) is the point cloud, and (b) is mesh.

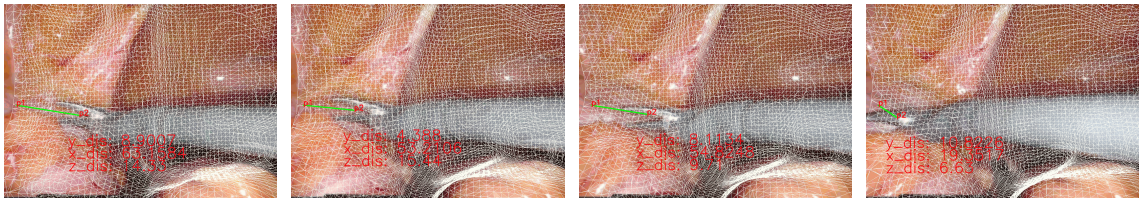


(a) Frame 15 (b) Frame 30 (c) Frame 45 (d) Frame 58

Fig. 8.8 The results of AR on Pull dataset [11] from Frame 15 to Frame 58.

Poisson surface reconstruction. It can be seen in Fig. 8.7(a) that the absent parts of the scene can also be reconstructed.

Two examples are used to conduct AR experiments, as shown in Fig. 8.8 and Fig. 8.9. Fig. 8.8 shows that an instrument is pulling the organ in which there are significant topology changes in the mesh for the organ, and the mesh of the instruments is unchanged. Moreover, another example is that an instrument is cutting the organs, as shown in Fig. 8.9, and the results are similar to that of example one.



(a) Frame 36 (b) Frame 50 (c) Frame 100 (d) Frame 150

Fig. 8.9 The results of AR on Cut dataset [11] from Frame 36 to Frame 150.

8.4 Conclusions

In this Chapter, an AR framework is proposed, which shows how the proposed depth estimation network combined with 3D dynamic reconstruction can build a real-time topology-aware AR application in a practical environment. Recognizing this, a graph topology structure is used to learn non-Euclidean features on images because the graph is similar to point clouds, including irregular and out-of-order properties. Therefore, the proposed GNN-based method can achieve accurate depth, which can provide important feedback signals for RAMIS, particularly for instruments. In addition, AR examples of the endoscopy dataset also show the function of 3D dynamic reconstruction in MIS. However, the method proposed in [11] needs multiple point clouds as the input to achieve a reconstructed scene, which is difficult to apply in the real world. Another problem is that there are too many preprocessing steps involved in conducting AR experiments, including segmentation, 3D dynamic reconstruction and Poisson surface reconstruction. The results of this chapter have highlighted both the huge potential and challenges of AR technology for complex environment/scene applications.

Chapter 9

Conclusions and Future Works

9.1 Conclusions

This thesis demonstrates that a graph-based self-supervised monocular depth estimation framework in Chapter 4, exploring topological structures, achieves more accurate depth estimation and surface 3D reconstruction. Two use cases are presented in Chapter 5 for endoscopy datasets and Chapter 6 for videos captured by UAVs under unstructured environments. In Chapter 7, a *Break and Splice* framework is used to handle the non-rigid point cloud registration with special topology changes. Apart from RGB-D datasets, an example based on the result of Chapter 5 is conducted in the endoscopy scene. While the proposed *Break and Splice* framework still faces challenges in the context of AR applications, it holds promise as a potential solution for advancing traditional 3D dynamic reconstruction methodologies. Finally, for 8, combining the point cloud results of Chapter 5 with associated methods, like 3D dynamic reconstruction and segmentation, can improve AR applications in MIS. To conclude:

- A mathematical background about group equivariance deep learning and projective geometry is introduced in Chapter 3.
- A novel self-supervised depth estimation framework based on group equivariance deep learning is proposed to improve the fine details of depth in Chapter 4.
- A use case study on endoscopy datasets aims to demonstrate the effectiveness of the proposed self-supervised depth estimation framework, and the improved SSIM loss function for low-illumination datasets is introduced in Chapter 5.
- Another use case study on videos captured by UAVs is in Chapter 6, which has a similar structure to endoscopy datasets, including unstructured scenes and cameras with free motion.
- A statistical algorithm for non-rigid point set registration, which is the essential part of 3D dynamic reconstruction, can address the challenge of topology changes without estimating correspondence in Chapter 7.

- Two topology-aware AR Applications under Minimally Invasive Surgery(MIS) are introduced in Chapter 8, One is the distance of two instruments based on depth maps shown in real-time. Another application utilized depth to dynamically reconstruct a background without instruments, which is to deal with miss depth when the organs are below the instruments.

9.2 Future Works

Although the proposed depth estimation and non-rigid point set registration approaches have demonstrated better performance than state-of-the-art methods and shown promising applications, some challenges remain. These challenges provide potential research avenues for future work.

- *Improvement of depth estimation on UAV video.* The results of depth estimation in Chapter 6 show that the proposed method cannot handle depth with fog weather. How to estimate the depth in the case of fog weather will be a research direction. Recently, some works [172], [173] utilized CNN-based methods to remove the weather, including fog, rain and snow. These methods may be helpful in improving depth estimation. In addition, due to the limitation of hardware, all experiment is under low-resolution. It is effective for endoscopy images, but UVA videos usually include long-distance objects, which tend to lose detailed information under low resolution and achieve a bad depth map. A potential method is super-pixel, which may reduce the dependence on hardware.

- *Improvement of non-rigid point cloud registration.* Dealing with the mentioned issues in Chapter 7 will be a potential direction in the future. Firstly, the development of non-rigid registration between the part point cloud and the whole point cloud may be helpful for large camera movements and multi-frame fusion. Then, combining the point cloud and the segmentation based on images may solve inaccurate labels due to clusters or self-occlusion of objects. Finally, the patch-based non-rigid 3D reconstruction[174] may be a potential solution for addressing time consumption issues.

- *Improvement of AR application.* Robustness for future AR development in medical practices remains a challenge. For example, placing AR content in the incorrect position could mislead the surgeon, potentially leading to critical medical errors. As mentioned in this thesis, the depth of the monocular camera is obtained by a complex deep-learning framework. However, it is easy to obtain an accurate and reliable point cloud through a depth camera, such as Kinect or RealSense [175]. Therefore, the development of depth cameras based on depth estimation algorithms in MIS may solve some limitations on AR applications.

References

[1] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[2] Bernhard Kainz. Deep learning-equivariance and invarince. https://www.doc.ic.ac.uk/~bkainz/teaching/DL/notes/equivariance.pdf, 2020-12-11. Accessed: 2023–10-16.

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

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

[5] Adrian Johnston and Gustavo Carneiro. Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume. In *Proceedings of the ieee/cvf Conference on Computer Vision and Pattern Recognition*, pages 4756–4765, 2020.

[6] Armin Masoumian, Hatem A Rashwan, Saddam Abdulwahab, Julián Cristiano, M Salman Asif, and Domenec Puig. Gcndepth: Self-supervised monocular depth estimation based on graph convolutional network. *Neurocomputing*, 517:81–92, 2023.

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

[8] Logambal Madhuanand, Francesco Nex, and Michael Ying Yang. Self-supervised monocular depth estimation from oblique uav videos. *ISPRS journal of photogrammetry and remote sensing*, 176:1–14, 2021.

[9] Jinwoo Bae, Sungho Moon, and Sunghoon Im. Deep digging into the generalization of self-supervised monocular depth estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 187–196, 2023.

[10] Miroslava Slavcheva, Maximilian Baust, Daniel Cremers, and Slobodan Ilic. Killingfusion: Non-rigid 3d reconstruction without correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1386–1395, 2017.

[11] Yuehao Wang, Yonghao Long, Siu Hin Fan, and Qi Dou. Neural rendering for stereo 3d reconstruction of deformable tissues in robotic surgery. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 431–441. Springer, 2022.

[12] Sandeep N Kundu, Nawaz Muhammad, and Fraha Sattar. Using the augmented reality sandbox for advanced learning in geoscience education. In *2017 IEEE 6th international conference on teaching, assessment, and learning for engineering (TALE)*, pages 13–17. IEEE, 2017.

[13] Janne Paavilainen, Hannu Korhonen, Kati Alha, Jaakko Stenros, Elina Koskinen, and Frans Mayra. The pokémon go experience: A location-based augmented reality mobile game goes mainstream. In *Proceedings of the 2017 CHI conference on human factors in computing systems*, pages 2493–2498, 2017.

[14] Long Chen, Wen Tang, Nigel W John, Tao Ruan Wan, and Jian Jun Zhang. Slam-based dense surface reconstruction in monocular minimally invasive surgery and its application to augmented reality. *Computer methods and programs in biomedicine*, 158:135–146, 2018.

[15] Fabricio Amaguaña, Brayan Collaguazo, Jonathan Tituaña, and Wilbert G Aguilar. Simulation system based on augmented reality for optimization of training tactics on military operations. In *Augmented Reality, Virtual Reality, and Computer Graphics: 5th International Conference, AVR 2018, Otranto, Italy, June 24–27, 2018, Proceedings, Part I 5*, pages 394–403. Springer, 2018.

[16] Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329, 1994.

[17] Ronald T Azuma. A survey of augmented reality. *Presence: teleoperators & virtual environments*, 6(4):355–385, 1997.

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

[19] Mark Fiala. Artag, a fiducial marker system using digital techniques. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 590–596. IEEE, 2005.

[20] Gabriel Takacs, Vijay Chandrasekhar, Natasha Gelfand, Yingen Xiong, Wei-Chao Chen, Thanos Bismpigiannis, Radek Grzeszczuk, Kari Pulli, and Bernd Girod. Outdoors augmented reality on mobile phone using loxel-based visual feature organization. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, pages 427–434, 2008.

[21] Qing Hong Gao, Tao Ruan Wan, Wen Tang, and Long Chen. A stable and accurate marker-less augmented reality registration method. In *2017 International Conference on Cyberworlds (CW)*, pages 41–47. IEEE, 2017.

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

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

[24] Feng Gong, Paul Swain, and Timothy Mills. Wireless endoscopy. *Gastrointestinal endoscopy*, 51(6):725–729, 2000.

[25] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012.

[26] Santiago Royo and Maria Ballesta-Garcia. An overview of lidar imaging systems for autonomous vehicles. *Applied sciences*, 9(19):4093, 2019.

[27] Jesús Balado, Ernesto Frías, Silvia M González-Collazo, and Lucía Díaz-Vilariño. New trends in laser scanning for cultural heritage. In *New Technologies in Building and Construction: Towards Sustainable Development*, pages 167–186. Springer, 2022.

[28] Andreas Wedel, Uwe Franke, Jens Klappstein, Thomas Brox, and Daniel Cremers. Realtime depth estimation and obstacle detection from monocular video. In *Joint Pattern Recognition Symposium*, pages 475–484. Springer, 2006.

[29] Charan D Prakash, Jinjin Li, Farshad Akhbari, and Lina J Karam. Sparse depth calculation using real-time key-point detection and structure from motion for advanced driver assist systems. In *International Symposium on Visual Computing*, pages 740–751. Springer, 2014.

[30] Hossein Javidnia and Peter Corcoran. Accurate depth map estimation from small motions. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2453–2461, 2017.

[31] Antonio Torralba and Aude Oliva. Depth estimation from image structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1226–1238, 2002.

[32] Jae-Il Jung and Yo-Sung Ho. Depth map estimation from single-view image using object classification based on bayesian learning. In *2010 3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video*, pages 1–4. IEEE, 2010.

[33] Beyang Liu, Stephen Gould, and Daphne Koller. Single image depth estimation from predicted semantic labels. In *2010 IEEE computer society Conference on Computer Vision and Pattern Recognition*, pages 1253–1260. IEEE, 2010.

[34] Miaomiao Liu, Mathieu Salzmann, and Xuming He. Discrete-continuous depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 716–723, 2014.

[35] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014.

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

[37] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, 2017.

[38] Christopher Weber, Stefanie Hahmann, and Hans Hagen. Sharp feature detection in point clouds. In *2010 shape modeling international conference*, pages 175–186. IEEE, 2010.

[39] 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 *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. Ieee, 2011.

[40] Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 343–352, 2015.

[41] Bailin Deng, Yuxin Yao, Roberto M Dyke, and Juyong Zhang. A survey of non-rigid 3d registration. In *Computer Graphics Forum*, volume 41, pages 559–589. Wiley Online Library, 2022.

[42] Haili Chui and Anand Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2-3):114–141, 2003.

[43] Yang Yang, Sim Heng Ong, and Kelvin Weng Chiong Foong. A robust global and local mixture distance based non-rigid point set registration. *Pattern Recognition*, 48(1):156–173, 2015.

[44] Andriy Myronenko, Xubo Song, and Miguel Carreira-Perpinan. Non-rigid point set registration: Coherent point drift. *Advances in neural information processing systems*, 19, 2006.

[45] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations*, 2022.

[46] Stephen M Pizer, E Philip Amburn, John D Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart ter Haar Romeny, John B Zimmerman, and Karel Zuiderveld. Adaptive histogram equalization and its variations. *Computer vision, graphics, and image processing*, 39(3):355–368, 1987.

[47] Osamu Hirose. A bayesian formulation of coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(7):2269–2286, 2020.

[48] Liang Wang and Ruigang Yang. Global stereo matching leveraged by sparse ground control points. In *CVPR 2011*, pages 3033–3040. IEEE, 2011.

[49] Mircea Paul Muresan, Mihai Negru, and Sergiu Nedevschi. Improving local stereo algorithms using binary shifted windows, fusion and smoothness constraint. In *2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 179–185. IEEE, 2015.

[50] Robert Spangenberg, Tobias Langner, Sven Adfeldt, and Raúl Rojas. Large scale semi-global matching on the cpu. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 195–201. IEEE, 2014.

[51] Xingshuai Dong, Matthew A Garratt, Sreenatha G Anavatti, and Hussein A Abbass. Towards real-time monocular depth estimation for robotics: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):16940–16961, 2022.

[52] Shimon Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 203(1153):405–426, 1979.

[53] Gregory D Hager and Peter N Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, 1998.

[54] Hyowon Ha, Sunghoon Im, Jaesik Park, Hae-Gon Jeon, and In So Kweon. High-quality depth from uncalibrated small motion clip. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5413–5421, 2016.

[55] Chris Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988.

[56] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI'81: 7th international joint conference on Artificial intelligence*, volume 2, pages 674–679, 1981.

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

[58] Berthold KP Horn. Shape from shading: A method for obtaining the shape of a smooth opaque object from one view. 1970.

[59] Ashutosh Saxena, Sung Chung, and Andrew Ng. Learning depth from single monocular images. *Advances in neural information processing systems*, 18, 2005.

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

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

[62] Xiaoyang Guo, Hongsheng Li, Shuai Yi, Jimmy Ren, and Xiaogang Wang. Learning monocular depth by distilling cross-domain stereo networks. In *European Conference on Computer Vision*, pages 484–500, 2018.

[63] Yuru Chen, Haitao Zhao, Zhengwei Hu, and Jingchao Peng. Attention-based context aggregation network for monocular depth estimation. *International Journal of Machine Learning and Cybernetics*, 12:1583–1596, 2021.

[64] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):824–840, 2008.

[65] Richard J Chen, Taylor L Bobrow, Thomas Athey, Faisal Mahmood, and Nicholas J Durr. Slam endoscopy enhanced by adversarial depth prediction. *arXiv preprint arXiv:1907.00283*, 2019.

[66] Faisal Mahmood, Richard Chen, and Nicholas J Durr. Unsupervised reverse domain adaptation for synthetic medical images via adversarial training. *IEEE transactions on medical imaging*, 37(12):2572–2581, 2018.

[67] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 740–756. Springer, 2016.

[68] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017.

[69] Mehmet Turan, Evin Pinar Ornek, Nail Ibrahimli, Can Giracoglu, Yasin Almalioglu, Mehmet Fatih Yanik, and Metin Sitti. Unsupervised odometry and depth learning for endoscopic capsule robots. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1801–1807. IEEE, 2018.

[70] Xingtong Liu, Ayushi Sinha, Masaru Ishii, Gregory D Hager, Austin Reiter, Russell H Taylor, and Mathias Unberath. Dense depth estimation in monocular endoscopy with self-supervised learning methods. *IEEE transactions on medical imaging*, 39(5):1438–1447, 2019.

[71] Kutsev Bengisu Ozyoruk, Guliz Irem Gokceler, Taylor L Bobrow, Gulfize Coskun, Kagan Incetan, Yasin Almalioglu, Faisal Mahmood, Eva Curto, Luis Perdigoto, Marina Oliveira, et al. Endoslam dataset and an unsupervised monocular visual odometry and depth estimation approach for endoscopic videos. *Medical Image Analysis*, 71:102058, 2021.

[72] Shuwei Shao, Zhongcai Pei, Weihai Chen, Wentao Zhu, Xingming Wu, Dianmin Sun, and Baochang Zhang. Self-supervised monocular depth and ego-motion estimation in endoscopy: Appearance flow to the rescue. *Medical Image Analysis*, 77:102338, 2022.

[73] Gary KL Tam, Zhi-Quan Cheng, Yu-Kun Lai, Frank C Langbein, Yonghuai Liu, David Marshall, Ralph R Martin, Xian-Fang Sun, and Paul L Rosin. Registration of 3d point clouds and meshes: a survey from rigid to nonrigid. *IEEE transactions on visualization and computer graphics*, 19(7):1199–1217, 2012.

[74] Thomas Whelan, Renato F Salas-Moreno, Ben Glocker, Andrew J Davison, and Stefan Leutenegger. Elasticfusion: Real-time dense slam and light source estimation. *The International Journal of Robotics Research*, 35(14):1697–1716, 2016.

[75] Syed Afaq Ali Shah, Mohammed Bennamoun, and Farid Boussaid. Keypoints-based surface representation for 3d modeling and 3d object recognition. *Pattern Recognition*, 64:29–38, 2017.

[76] Michael Zollhöfer, Patrick Stotko, Andreas Görlitz, Christian Theobalt, Matthias Nießner, Reinhard Klein, and Andreas Kolb. State of the art on 3d reconstruction with rgb-d cameras. In *Computer Graphics Forum*, volume 37, pages 625–652. Wiley Online Library, 2018.

[77] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.

[78] K Somani Arun, Thomas S Huang, and Steven D Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (5):698–700, 1987.

[79] Dmitry Chetverikov, Dmitry Stepanov, and Pavel Krsek. Robust euclidean alignment of 3d point sets: the trimmed iterative closest point algorithm. *Image and vision computing*, 23(3):299–309, 2005.

[80] Qi-Xing Huang, Bart Adams, Martin Wicke, and Leonidas J Guibas. Non-rigid registration under isometric deformations. In *Computer Graphics Forum*, volume 27, pages 1449–1457. Wiley Online Library, 2008.

[81] Jiayi Ma, Weichao Qiu, Ji Zhao, Yong Ma, Alan L Yuille, and Zhuowen Tu. Robust $l_{2}e$ estimation of transformation for non-rigid registration. *IEEE Transactions on Signal Processing*, 63(5):1115–1129, 2015.

[82] Brett Allen, Brian Curless, and Zoran Popović. The space of human body shapes: reconstruction and parameterization from range scans. *ACM Transactions on Graphics (ToG)*, 22(3):587–594, 2003.

[83] Brian Amberg, Sami Romdhani, and Thomas Vetter. Optimal step nonrigid icp algorithms for surface registration. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[84] Andriy Myronenko, Xubo Song, Miguel A Carreira-Perpinán, et al. Non-rigid point set registration: Coherent point drift. *Advances in neural information processing systems*, 19:1009, 2007.

[85] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, 2010.

[86] Alan L Yuille and Norberto M Grzywacz. A mathematical analysis of the motion coherence theory. *International Journal of Computer Vision*, 3(2):155–175, 1989.

[87] Vladislav Golyanik, Bertram Taetz, Gerd Reis, and Didier Stricker. Extended coherent point drift algorithm with correspondence priors and optimal subsampling. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016.

[88] Lifei Bai, Xianqiang Yang, and Huijun Gao. Nonrigid point set registration by preserving local connectivity. *IEEE transactions on cybernetics*, 48(3):826–835, 2017.

[89] Osamu Hirose. A bayesian formulation of coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(7):2269–2286, 2021.

[90] Konstantinos Zampogiannis, Cornelia Fermüller, and Yiannis Aloimonos. Topology-aware non-rigid point cloud registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(3):1056–1069, 2021.

[91] 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)*, 35(4):1–13, 2016.

[92] Kaiwen Guo, Feng Xu, Tao Yu, Xiaoyang Liu, Qionghai Dai, and Yebin Liu. Real-time geometry, albedo, and motion reconstruction using a single rgb-d camera. *ACM Transactions on Graphics (ToG)*, 36(4):1, 2017.

[93] Miroslava Slavcheva, Maximilian Baust, and Slobodan Ilic. Sobolevfusion: 3d reconstruction of scenes undergoing free non-rigid motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2646–2655, 2018.

[94] Chao Li and Xiaohu Guo. Topology-change-aware volumetric fusion for dynamic scene reconstruction. In *European Conference on Computer Vision*, pages 258–274. Springer, 2020.

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

[96] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[97] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[98] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980.

[99] Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 37–45, 2015.

[100] Vassilis N Ioannidis, Antonio G Marques, and Georgios B Giannakis. A recurrent graph neural network for multi-relational data. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8157–8161. IEEE, 2019.

[101] Connor W Coley, Wengong Jin, Luke Rogers, Timothy F Jamison, Tommi S Jaakkola, William H Green, Regina Barzilay, and Klavs F Jensen. A graph-convolutional neural network model for the prediction of chemical reactivity. *Chemical science*, 10(2):370–377, 2019.

[102] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016.

[103] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

[104] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[105] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

[106] Horace Pan and Risi Kondor. Permutation equivariant layers for higher order interactions. In *International Conference on Artificial Intelligence and Statistics*, pages 5987–6001. PMLR, 2022.

[107] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. *6th International Conference on Learning Representations*, 2017.

[108] Roger Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal on Robotics and Automation*, 3(4):323–344, 1987.

[109] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

[110] Leonid Keselman, John Iselin Woodfill, Anders Grunnet-Jepsen, and Achintya Bhowmik. Intel realsense stereoscopic depth cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops(CVPRW)*, pages 1–10, 2017.

[111] Shiqi Li, Chi Xu, and Ming Xie. A robust o (n) solution to the perspective-n-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1444–1450, 2012.

[112] 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)*, 38(5):1–12, 2019.

[113] Yingxue Zhang and Michael Rabbat. A graph-cnn for 3d point cloud classification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6279–6283. IEEE, 2018.

[114] Junwei Fu, Jun Liang, and Ziyang Wang. Monocular depth estimation based on multi-scale graph convolution networks. *IEEE Access*, 8:997–1009, 2019.

[115] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[116] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015.

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

[118] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on computational imaging*, 3(1):47–57, 2016.

[119] Matteo Poggi, Fabio Tosi, Konstantinos Batsos, Philippos Mordohai, and Stefano Mattoccia. On the synergies between machine learning and binocular stereo for depth estimation from images: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5314–5334, 2021.

[120] Aleks Attanasio, Bruno Scaglioni, Matteo Leonetti, Alejandro F Frangi, William Cross, Chandra Shekhar Biyani, and Pietro Valdastri. Autonomous tissue retraction in robotic assisted minimally invasive surgery–a feasibility study. *IEEE Robotics and Automation Letters*, 5(4):6528–6535, 2020.

[121] Yuichiro Hayashi, Kazunari Misawa, Masahiro Oda, David J Hawkes, and Kensaku Mori. Clinical application of a surgical navigation system based on virtual laparoscopy in laparoscopic gastrectomy for gastric cancer. *International journal of computer assisted radiology and surgery*, 11:827–836, 2016.

[122] Xiongbiao Luo, Ying Wan, Xiangjian He, and Kensaku Mori. Observation-driven adaptive differential evolution and its application to accurate and smooth bronchoscope three-dimensional motion tracking. *Medical Image Analysis*, 24(1):282–296, 2015.

[123] Andru P Twinanda, Sherif Shehata, Didier Mutter, Jacques Marescaux, Michel De Mathelin, and Nicolas Padoy. Endonet: a deep architecture for recognition tasks on laparoscopic videos. *IEEE transactions on medical imaging*, 36(1):86–97, 2016.

[124] Ariel Tankus, Nir Sochen, and Yehezkel Yeshurun. Shape-from-shading under perspective projection. *International Journal of Computer Vision*, 63:21–43, 2005.

[125] Aji Resindra Widya, Yusuke Monno, Kosuke Imahori, Masatoshi Okutomi, Sho Suzuki, Takuji Gotoda, and Kenji Miki. 3d reconstruction of whole stomach from endoscope video using structure-from-motion. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3900–3904. IEEE, 2019.

[126] Miguel Lourenço, Danail Stoyanov, and Joao P Barreto. Visual odometry in stereo endoscopy by using pearl to handle partial scene deformation. In *Augmented Environments for Computer-Assisted Interventions: 9th International Workshop, AECAI 2014, Held in Conjunction with MICCAI 2014, Boston, MA, USA, September 14, 2014. Proceedings 9*, pages 33–40. Springer, 2014.

[127] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016.

[128] M Ye, E Johns, A Handa, L Zhang, P Pratt, and G-Z Yang. Self-supervised siamese learning on stereo image pairs for depth estimation in robotic surgery. In *The Hamlyn Symposium on Medical Robotics*, page 27, 2017.

[129] Max Allan, Jonathan Mcleod, Congcong Wang, Jean Claude Rosenthal, Zhenglei Hu, Niklas Gard, Peter Eisert, Ke Xue Fu, Trevor Zeffiro, Wenyao Xia, et al. Stereo correspondence and reconstruction of endoscopic data challenge. *arXiv preprint arXiv:2101.01133*, 2021.

[130] PJ Eddie Edwards, Dimitris Psychogyios, Stefanie Speidel, Lena Maier-Hein, and Danail Stoyanov. Serv-ct: A disparity dataset from cone-beam ct for validation of endoscopic 3d reconstruction. *Medical Image Analysis*, 76:102302, 2022.

[131] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

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

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

[134] Kamil Kamiński, Jan Ludwiczak, Maciej Jasiński, Adriana Bukala, Rafal Madaj, Krzysztof Szczepaniak, and Stanisław Dunin-Horkawicz. Rossmann-toolbox: a deep learning-based protocol for the prediction and design of cofactor specificity in rossmann fold proteins. *Briefings in bioinformatics*, 23(1):bbab371, 2022.

[135] Namil Kim, Yukyung Choi, Soonmin Hwang, and In So Kweon. Multispectral transfer network: Unsupervised depth estimation for all-day vision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[136] Yawen Lu and Guoyu Lu. An alternative of lidar in nighttime: Unsupervised depth estimation based on single thermal image. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3833–3843, 2021.

[137] Jaime Spencer, Richard Bowden, and Simon Hadfield. Defeat-net: General monocular depth via simultaneous unsupervised representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14402–14413, 2020.

[138] Madhu Vankadari, Sourav Garg, Anima Majumder, Swagat Kumar, and Ardhendu Behera. Unsupervised monocular depth estimation for night-time images using adversarial domain feature adaptation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*, pages 443–459. Springer, 2020.

[139] Kun Wang, Zhenyu Zhang, Zhiqiang Yan, Xiang Li, Baobei Xu, Jun Li, and Jian Yang. Regularizing nighttime weirdness: Efficient self-supervised monocular depth estimation in the dark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16055–16064, 2021.

[140] Edwin H Land. The retinex theory of color vision. *Scientific american*, 237(6):108–129, 1977.

[141] Xiaojie Guo, Yu Li, and Haibin Ling. Lime: Low-light image enhancement via illumination map estimation. *IEEE Transactions on image processing*, 26(2):982–993, 2016.

[142] Risheng Liu, Long Ma, Jiaao Zhang, Xin Fan, and Zhongxuan Luo. Retinex-inspired unrolling with cooperative prior architecture search for low-light image enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10561–10570, 2021.

[143] Yonghua Zhang, Xiaojie Guo, Jiayi Ma, Wei Liu, and Jiawan Zhang. Beyond brightening low-light images. *International Journal of Computer Vision*, 129:1013–1037, 2021.

[144] Mostafa Hassanalian and Abdessattar Abdelkefi. Classifications, applications, and design challenges of drones: A review. *Progress in Aerospace Sciences*, 91:99–131, 2017.

[145] Praveen Kumar Reddy Maddikunta, Saqib Hakak, Mamoun Alazab, Sweta Bhattacharya, Thippa Reddy Gadekallu, Wazir Zada Khan, and Quoc-Viet Pham. Unmanned aerial vehicles in smart agriculture: Applications, requirements, and challenges. *IEEE Sensors Journal*, 21(16):17608–17619, 2021.

[146] Roberto Sadao Yokoyama, Bruno Yuji Lino Kimura, and Edson dos Santos Moreira. An architecture for secure positioning in a uav swarm using rssi-based distance estimation. *ACM SIGAPP Applied Computing Review*, 14(2):36–44, 2014.

[147] Juan Manuel Jurado, L Ortega, Juan José Cubillas, and FR Feito. Multispectral mapping on 3d models and multi-temporal monitoring for individual characterization of olive trees. *Remote Sensing*, 12(7):1106, 2020.

[148] Juan M Jurado, Alfonso López, Luís Pádua, and Joaquim J Sousa. Remote sensing image fusion on 3d scenarios: A review of applications for agriculture and forestry. *International Journal of Applied Earth Observation and Geoinformation*, 112:102856, 2022.

[149] Julien Vallet, Flory Panissod, Christoph Strecha, and M Tracol. Photogrammetric performance of an ultra light weight swinglet uav. Technical report, 2011.

[150] Matthew J Westoby, James Brasington, Niel F Glasser, Michael J Hambrey, and Jennifer M Reynolds. 'structure-from-motion'photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, 179:300–314, 2012.

[151] Paul Ryan Nesbit and Christopher H Hugenholtz. Enhancing uav–sfm 3d model accuracy in high-relief landscapes by incorporating oblique images. *Remote Sensing*, 11(3):239, 2019.

[152] Vlad-Cristian Miclea and Sergiu Nedevschi. Monocular depth estimation with improved long-range accuracy for uav environment perception. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–15, 2021.

[153] M Hermann, B Ruf, M Weinmann, and S Hinz. Self-supervised learning for monocular depth estimation from aerial imagery. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 5:357–364, 2020.

[154] Michael Fonder and Marc Van Droogenbroeck. Mid-air: A multi-modal dataset for extremely low altitude drone flights. In *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, June 2019.

[155] Ye Lyu, George Vosselman, Gui-Song Xia, Alper Yilmaz, and Michael Ying Yang. Uavid: A semantic segmentation dataset for uav imagery. *ISPRS journal of photogrammetry and remote sensing*, 165:108–119, 2020.

[156] Horaţiu Florea, Vlad-Cristian Miclea, and Sergiu Nedevschi. Wilduav: Monocular uav dataset for depth estimation tasks. In *2021 IEEE 17th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 291–298. IEEE, 2021.

[157] M Awrangjeb. Using point cloud data to identify, trace, and regularize the outlines of buildings. *International Journal of Remote Sensing*, 37(3):551–579, 2016.

[158] Thomas S. Ferguson. A bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230, 1973.

[159] Khadidja Meguelati, Bénédicte Fontez, Nadine Hilgert, and Florent Masseglia. Dirichlet process mixture models made scalable and effective by means of massive distribution. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 502–509, 2019.

[160] Radford M Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of computational and graphical statistics*, 9(2):249–265, 2000.

[161] Christopher Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *Proceedings of the 14th annual conference on neural information processing systems*, number CONF, pages 682–688, 2001.

[162] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[163] Matthias Innmann, Michael Zollhöfer, Matthias Nießner, Christian Theobalt, and Marc Stamminger. Volumedeform: Real-time volumetric non-rigid reconstruction. In *European Conference on Computer Vision*, pages 362–379. Springer, 2016.

[164] Evangelos Alexiou and Touradj Ebrahimi. Point cloud quality assessment metric based on angular similarity. In *2018 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2018.

[165] Evangelos Alexiou and Touradj Ebrahimi. Towards a point cloud structural similarity metric. In *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 1–6. IEEE, 2020.

[166] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1328–1338, 2019.

[167] Suren Kumar, Pankaj Singhal, and Venkat N Krovi. Computer-vision-based decision support in surgical robotics. *IEEE Design & Test*, 32(5):89–97, 2015.

[168] Tamas Haidegger, Stefanie Speidel, Danail Stoyanov, and Richard M Satava. Robot-assisted minimally invasive surgery—surgical robotics in the data age. *Proceedings of the IEEE*, 110(7):835–846, 2022.

[169] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, page 0, 2006.

[170] Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.

[171] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013.

[172] Ruoteng Li, Robby T Tan, and Loong-Fah Cheong. All in one bad weather removal using architectural search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3175–3185, 2020.

[173] Jeya Maria Jose Valanarasu, Rajeev Yasarla, and Vishal M Patel. Transweather: Transformer-based restoration of images degraded by adverse weather conditions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2353–2363, 2022.

[174] Carmel Kozlov, Miroslava Slavcheva, and Slobodan Ilic. Patch-based non-rigid 3d reconstruction from a single depth stream. In *2018 International Conference on 3D Vision (3DV)*, pages 42–51. IEEE, 2018.

[175] Sanjay Rijal, Suruchi Pokhrel, Madhav Om, and Vaghawan Prasad Ojha. Comparing depth estimation of azure kinect and realsense d435i cameras. *Available at SSRN 4597442*, 2023.