# Enhancing the Reliability of LiDAR Point Cloud Sampling: A Colorization and Super-Resolution Approach Based on LiDAR-Generated Images

University of Turku
Department of Computing
Master's Thesis
Turku Intelligent Embedded and
Robotic Systems (TIERS) Lab
September 2024
Honghao Du

Supervisors:
Dr. Xianjia Yu
Prof. Tomi Westerlund

UNIVERSITY OF TURKU
Department of Computing

HONGHAO DU: Enhancing the Reliability of LiDAR Point Cloud Sampling: A Colorization and Super-Resolution Approach Based on LiDAR-Generated Images

Master's Thesis, 45 p.
Turku Intelligent Embedded and
Robotic Systems (TIERS) Lab
September 2024

---

In recent years, Light Detection and Ranging (LiDAR) technology, a critical sensor in robotics and autonomous systems, has seen significant advancements. These improvements include enhanced resolution of point clouds and the capability to provide 360° low-resolution images. These images encode various data such as depth, reflectivity, and near-infrared light within the pixels. However, an excessive density of points and conventional point cloud sampling can be counterproductive, particularly in applications such as LiDAR odometry, where misleading points may induce drift errors and geometry information is degraded. Currently, extensive research efforts are being directed towards leveraging LiDAR-generated images to improve situational awareness.

This paper presents a comprehensive review of current deep learning (DL) techniques, including colorization and super-resolution, which are traditionally utilized in conventional computer vision tasks. These techniques are applied to LiDAR-generated images and are analyzed qualitatively. BBuilding on this analysis, we have developed a novel approach that selectively integrates the most effective methods with LiDAR imagery to sample reliable points from the LiDAR point cloud. This approach aims to not only improve the accuracy of point cloud registration but avoid mismatching caused by lacking of geometry information, thereby augmenting the utility and precision of LiDAR systems in practical applications. In our evaluation, the proposed approach demonstrates superior performance compared to our previous work, achieving lower translation and rotation errors with a reduced number of points.

Keywords: LiDAR, Odometry, Super-Resolution, Colorization, LiDAR-as-a-camera, Point Cloud Sampling

# Contents

# List of Figures

# List of Tables

# List of acronyms

**ALIKE** Accurate and Lightweight Keypoint Detection and Descriptor Extraction

**APES** Attention-based Point Cloud Edge Sampling

**BRIEF** Binary Robust Independent Elementary Features

**CNN** Convolution Neural Network

**DGCNN** Dynamic Graph Convolution Neural Network

**DL** Deep Learning

**DoG** Difference-of-Gaussian

**FAST** Features from Accelerated Segment Tes

**FPS** Farthest Point Sampling

**GAN** Generative Adversarial Network

**GIS** Geographic Information Systems

**HR** High Resolution

**ICP** Iterative Closest Point

**LiDAR** Light Detection and Ranging

**LO** LiDAR Odometry

**LOMA** Lidar Odometry and Mapping

**LR** Low Resolution

**MNN** Mutual Nearest Neighbor Matching

**NDT** Normal Distribution Transformation

**NIN** Networks-in-Network

**ORB** Oriented FAST and Rotated BRIEF

**ROS** Robot Operating System

**SIFT** Scale-Invariant Feature Transform

**SLAM** Simultaneous Localization and Mapping

**SR** Super Resolution

**SURF** Speeded-Up Robust Features

# 1 Introduction

LiDAR (Light Detection and Ranging) sensors have become increasingly significant in various domains of robotics and autonomous systems, particularly in navigation, perception, LiDAR Odometry (LO), and Simultaneous Localization and Mapping (SLAM) [1], [2]. Key factors that facilitate LiDARs' utility are the progressively increasing precision and density of point cloud data, which offers extensive geometric information about the surroundings. LiDAR technology is able to provide systems with accurate 3D spatial information from point cloud data, enabling robots to understand and respond to changes in their surroundings in real time. As LiDAR sensor accuracy and point cloud density continue to increase, the accuracy of navigation and localization tasks has improved dramatically. High-precision point cloud data not only provides rich geometric detail, but also improves the resolution of environment modeling, thus enhancing the system's ability to perceive complex environments.

However, despite the increasing quality and resolution of point cloud data, there are still challenges in processing LiDAR data. During calculating accurate LO or SLAM, the dense point cloud and conventional sampling approaches may introduce more inaccuracies, leading to error drift, which in turn affects the accuracy of localization and map construction. This issue becomes particularly evident in environments where geometric information is degraded, such as tunnels and corridors. Consequently, the process of extracting relevant points from the LiDAR point

cloud is of notable importance for effective point cloud registration. In these environments, the geometric features in LiDAR data are often insufficient to provide accurate spatial location information, and the system has difficulty distinguishing between different locations, leading to increased localization errors.

Nowadays, continuously more LiDAR manufacturers enable the capability of LiDARs to capture low-resolution panoramic images by encoding the depth, reflectivity, and near-infrared light information within the pixels. This technological breakthrough in multidimensional data fusion allows LiDAR to go beyond traditional geometric modeling tasks and generate richer environmental information. And this development holds significant potential for the application of deep learning (DL) techniques from the traditional computer vision field to LiDAR-generated images [3]. Such applications are particularly advantageous given that DL models utilized directly to point cloud data tend to be computationally expensive.

In addition, LiDAR-generated images open up new opportunities for computer vision tasks such as semantic segmentation, target detection, and object recognition. Combining reflectance information with depth images, the system can recognize and classify objects more accurately with existing DL models. Meanwhile, the use of near-infrared light data enhances the system's performance in low-light and complex environments, providing higher robustness and reliability for autonomous driving, robot navigation, and other fields.

In the context of enhancing the accuracy and robustness of LO, existing research has primarily focused on traditional methods utilizing intensity information for the displacement calculation between image frames [4]. In our previous work, we assessed the performance of keypoint detectors and extractors on LiDAR-generated images, exploring the potential of these tools to enhance the robustness of LO [5]. However, there remains significant potential for other DL approaches to improve system robustness further. For instance, Figure 1.2 illustrates our evaluation of

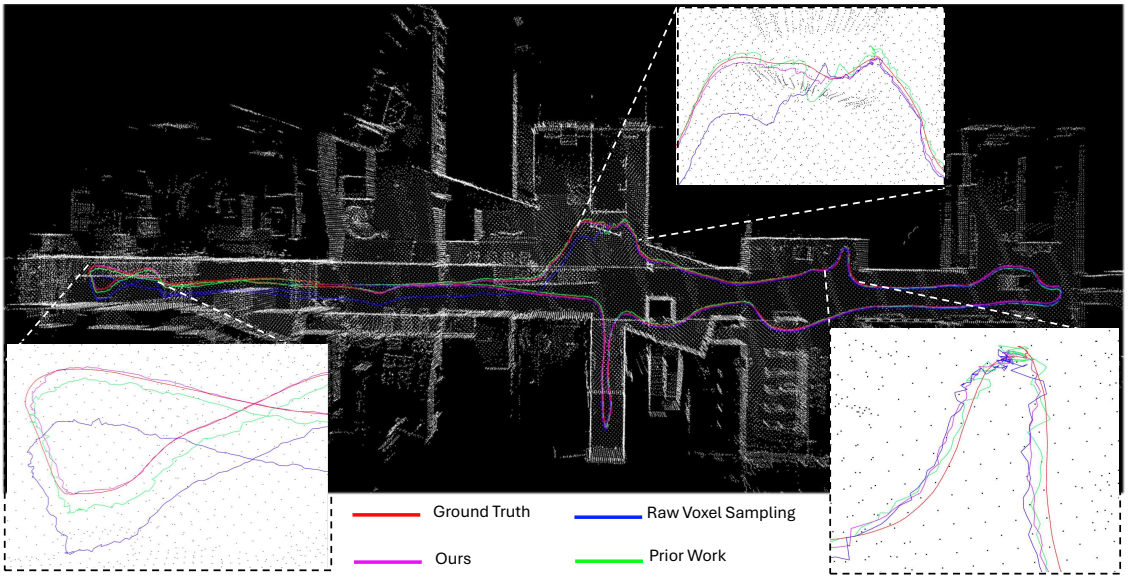



Figure 1.1: Example of potential drift and its correctness of our proposed approach.

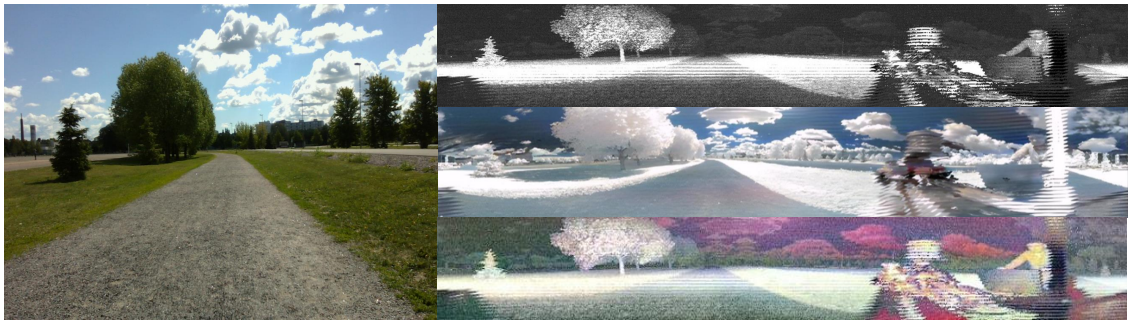


Figure 1.2: Results of DL-based super-resolution and colorization on a LiDAR generated image. The RGB image is shown on the left side, the images on the right, from top to bottom, represent the LiDAR-generated signal image, the colorized near-infrared (near-IR) image, and the colorized signal image, respectively

one of the DL-based super-resolution and colorization approaches applied to LiDAR images. Given that LiDAR-generated images are low-resolution, these techniques demonstrate significant potential for enhancement. In addition to this, unlike in our previous work, we employed key point extractors across all color channels (R, G, and B) of the image, rather than restricting the extraction to a single channel.

## 1.1 Problem Statement

Currently, LO exhibits some lack of robustness when facing scenes with degraded geometric information. These degraded scenarios may include low-texture, planar surfaces, long-range targets, etc., which makes it difficult for the system to accurately capture the geometric features of the surrounding environment. In addition, despite some progress in the work in [6] using LiDAR-generated images, the performance of the existing methods can still be improved when facing complex degraded scenes. Meanwhile, with the development of LiDAR technology, the potential of LiDAR data to generate images has not been fully investigated and utilized. By better mining and fusing LiDAR-generated image information, it may be possible to improve the performance of the system in complex environments. Therefore, how to improve the robustness of LiDAR Odometry to scenes with degraded geometric information and fully utilize the potential of LiDAR-generated images has become an important issue that needs to be addressed.

Despite the rapid adoption of LiDAR in autonomous driving, LiDAR-generated images have a low presence in the visual domain. Previous experiments using signal image keypoints to estimate LiDAR's odometry in a new approach have shown dramatic improvements in drift in LO. However, in terms of rotation offset, perhaps due to the lack of a reference with only a single image. The new idea we propose brings down the rotation error.

## 1.2 Significance and Motivation

In the field of deep learning, Colorization and Super-Resolution techniques have made significant progress. Colorization technology can convert grayscale images into color images to enrich visual information, while Super-Resolution technology can improve the resolution of images and enhance the detail performance. These

two techniques have been widely used in natural image restoration, enhancement and medical image processing, and have demonstrated powerful processing capabilities. However, although they show great potential in visual data processing, their application in LiDAR-generated images has not been fully explored.

LiDAR-generated images differ significantly from traditional natural images due to characteristics such as low resolution, sparse data, and non-visual spectra. This makes it challenging to effectively apply colorization and super-resolution techniques on LiDAR images. However, if these techniques can be successfully applied to LiDAR image processing, they will provide important opportunities to improve the overall performance of LiDAR systems. For example, the colorization technique can introduce more dimensional information into the otherwise monotonous LiDAR images to enhance their visual expression, while the super-resolution technique can make up for the shortcomings of low resolution of LiDAR data and improve the capture and understanding of environmental details.

Combining the recent advances of these techniques in the field of deep learning and exploring their applications in LiDAR image processing not only helps to improve the quality of LiDAR-generated images, but also may provide new solutions and ideas for existing work, which in turn will promote the development of LiDAR Odometry as well as other related fields. This is the core motivation of this study.

Currently, there is a lack of systematic benchmarking on the processing effectiveness of LiDAR-generated images in different environments, especially the performance of colorization and super-resolution techniques applied. This has led to a bottleneck in the further development of LiDAR-based simultaneous localization and map building (SLAM) techniques in diverse environments. To fill this research gap, we propose a benchmark test comparing the effect of colorization and super-resolution techniques on LiDAR image processing in different environments (e.g., open roads, forests, indoor offices, halls). This benchmark test provides the pos-

sibility of evaluating the performance of LiDAR-based SLAM systems in diverse scenarios and helps to assess the robustness of the system to real-world scenarios.

We hope that the research results in this paper can provide useful references and insights for other scholars with similar interests in this field.

## 1.3 Contribution

To address these issues, this paper gives the most comprehensive review of existing DL-based super-resolution and colorization methods, offering a detailed comparison of their key features. Subsequently, we provide a qualitative analysis of their performance on low-resolution panoramic LiDAR-generated images, aiming to enrich the photometric information available. Building on this analysis, we applied these DL approaches to keypoint extraction, testing various combinations of LiDAR-generated images enhanced through super-resolution and colorization techniques. Our evaluation of LO accuracy revealed that these methods resulted in improved accuracy and robustness compared to our previous work.

## 1.4 Structure of the Thesis

This paper is structured into six chapters, each providing a coherent flow of the research process.

1. Chapter 2 offers an overview of the related work, discussing current state-of-the-art studies, as well as the foundational settings, software, and hardware systems utilized in the development of this research.

2. Chapter 3 delivers an in-depth survey of existing deep learning-based models for colorization and super-resolution.

3. Chapter 4, the research methodology is thoroughly detailed, including the experimental design, technical discussion, and a description of the algorithms employed.

4. Chapter 5 presents the experimental results, offering insights into the findings.

5. Chapter 6 concludes the thesis, summarizing the contributions and outlining directions for future work.

# 2 Related Work

This section first introduce the LiDARs, then presents the methods of point cloud processing with sampling and matching. This is followed by a brief analysis of the LiDAR Odometry with point cloud and the pipeline of robotic operating system. Finally, summarize and explain the LiDAR application in robotics and some keypoint extractor methods.

## 2.1 LiDARs

LiDAR is an advanced sensor technology that uses lasers to scan the surface of a target object in order to construct detailed three-dimensional information. The technology works by emitting high-frequency laser pulses, which are reflected when they encounter an object, and the sensor measures the round-trip time delay of the light by recording the time it takes for the light to be reflected back to the sensor, which in turn accurately calculates the distance between the object and the sensor. This process is capable of generating 3D point cloud data with very high accuracy, accurately reflecting the spatial structure of the surrounding environment.

The images generated by the OS0-64 shown in Figure 2.1 include a signal image, a range image, and a point cloud with expansive $360^\circ \times 90^\circ$ field of view. Range image is a 2D image representation of the distance between the LiDAR sensor and objects in the environment. It is created from the 3D point cloud data collected by the LiDAR sensor and shows the distance from each point in the sensor's field of

view to the sensor itself. And signal image represents the strength of the reflected laser signal. Signal intensity is a measure of how much of the emitted laser energy is reflected back to the sensor, and is affected by material, surface roughness and reflection angle.

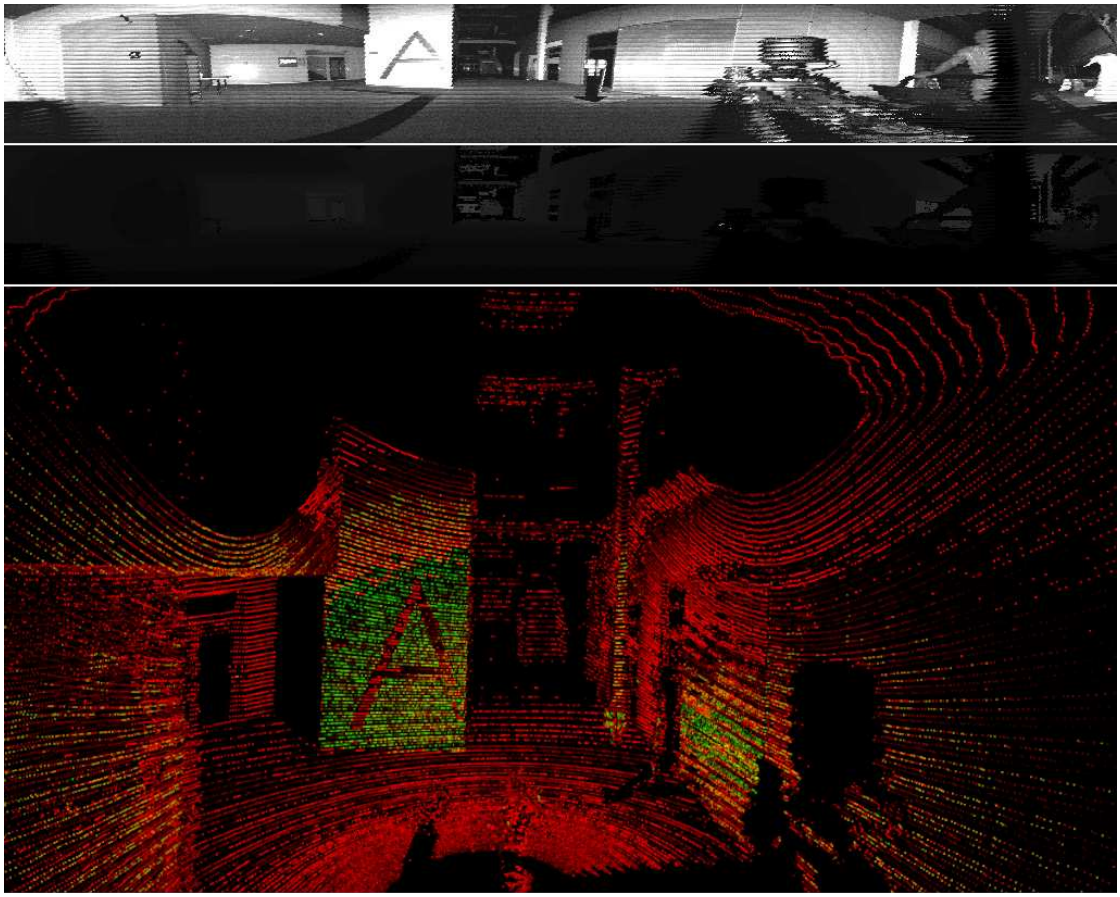


Figure 2.1: Samples of LiDAR-generated images, from top to bottom, are signal image, range image, and point cloud

The advantage of LiDAR technology is its ability to provide high-resolution data under different environmental conditions, especially when the field of view is limited or the ambient light is insufficient. Compared to traditional imaging methods, such as photogrammetry or satellite imagery, LiDAR technology has greater penetration capabilities and can effectively cope with adverse weather conditions such as light rain and light fog. In addition, because LiDAR does not rely on natural light sources, it continues to provide accurate 3D environmental data at night or in low

light conditions. This gives LiDAR an irreplaceable advantage in applications that require real-time, accurate sensing, especially in self-driving cars and unmanned aircraft systems that need to deal with complex environments.

In the field of robotics and autonomous driving, LiDAR, as one of the core sensors, can help the system perceive the surrounding environment in real time, and realize the functions of path planning and obstacle avoidance, etc. LiDAR's high-precision data helps the system quickly identify and classify obstacles on the road, ensuring that the vehicle or robot can perform its tasks safely and reliably. The three-dimensional point cloud data acquired by LiDAR enables self-driving cars to build accurate maps of the environment so that they can make effective driving decisions.

In addition, LiDAR also plays an important role in the fields of geographic information systems (GIS) [7], surveying and mapping, archaeology [8], and urban modeling [9]. LiDAR systems can generate high-resolution topographic maps and three-dimensional models, providing accurate spatial data that can help with geographic analysis and spatial research. For example, in terrain mapping, LiDAR technology can obtain fine terrain information that cannot be captured by traditional methods, and is especially suitable for accurate modeling of complex landforms.

Compared with traditional RGB cameras, LiDAR sensors not only have a wider range of distance detection, but also have higher ranging accuracy and can capture finer depth information than cameras. Therefore, in applications that require high accuracy and high reliability, LiDAR technology shows strong competitiveness by virtue of its robustness and robustness, and is especially suitable for data collection and analysis in complex and dynamic environments.

In summary, this section not only introduces the basic principles of LiDAR technology, but also outlines its applications and importance in several key areas, emphasizing its important role in modern research and technological development, es-

pecially in the wide range of applications in data collection and analysis.

## 2.2 Point Cloud Processing

Point cloud processing has become an important and active research area in computer vision and robotics in recent years, as it plays a crucial role in a variety of application scenarios, such as 3D computer vision, robot navigation, autonomous driving, and augmented reality. A point cloud is composed of a set of discrete 3D points that are usually captured by LiDAR or stereo vision system. These 3D points represent the surface geometry and spatial layout of an object or environment, and can provide rich geometric and structural information to computers. Therefore, point clouds are widely used in tasks such as 3D reconstruction, object detection, scene understanding, surface modeling, and map construction.

### 2.2.1 Point Cloud Sampling

Point cloud sampling is a critical step in 3D data processing, particularly for efficiently handling large datasets generated by modern 3D scanning technologies. Non-learning-based methods like voxel downsampling and Farthest Point Sampling (FPS) are widely used. Voxel downsampling is common in applications such as LiDAR odometry [10]–[12] and SLAM [13], [14], where it reduces point cloud size by replacing points within each voxel with a single representative point, though it may sacrifice fine details. FPS [15], [16], often employed in deep learning applications, iteratively selects well-distributed points across the cloud, balancing uniform coverage with feature preservation [17]. However, FPS is computationally intensive and may overlook small details.

Recent advancements in deep learning have introduced methods like S-NET [18] and PST-NET [19], which optimize sampling for specific tasks. S-NET learns task-

specific sampling strategies, outperforming traditional methods like FPS by tailoring the selection of points to the needs of applications such as classification and retrieval. PST-NET further innovates by leveraging a point-based transformer to consider geometric relationships among points, incorporating features like self-attention and local feature extraction to generate an optimal resampling distribution. Inspired by Canny edge detection and attention mechanisms, APES [20] is proposed as a non-generative method for capturing salient points in point cloud outlines.

Despite these advancements, research in point cloud processing has predominantly focused on tasks like classification [21], [22], segmentation [23], [24], and object detection [25], [26], with less attention given to sampling strategies. This underscores the need for continued research into efficient and effective sampling techniques, particularly within learning-based frameworks, to enhance the performance of point cloud applications.

### 2.2.2 Point Cloud Matching

Point cloud matching is a key task in 3D computer vision that involves the precise alignment of two or more point clouds. This task has important applications in 3D object recognition, spatial alignment, and motion tracking, especially in scenarios such as autonomous driving, robot navigation, and AR/VR. In order to achieve high-precision point cloud matching, various types of algorithms have emerged.

The iterative closest point (ICP) algorithm [27], proposed by Besl and McKay in 1992, is one of the earliest and most widely used matching algorithms.ICP continuously optimizes the alignment between two sets of point clouds by iteratively minimizing the Euclidean distance between the corresponding points of the two sets of point clouds. However, ICP has a slow convergence rate and may exhibit limitations when dealing with large-scale data or noisy scenes.

In this experiment, we use KISS-ICP [28] as the main point cloud matching

method.KISS-ICP is a simplified variant of ICP, which aims to improve efficiency and robustness, and is particularly suitable for real-time application scenarios such as autonomous navigation. Compared to traditional ICP methods, KISS-ICP reduces the algorithm complexity by removing redundant computational steps while maintaining good performance in dynamic and noisy environments. Its main advantage is that it provides real-time matching capabilities without sacrificing accuracy, which makes it particularly suitable for time-sensitive applications such as SLAM.

In addition to ICP and its variants, there are a number of non-ICP matching methods. The Normal Distribution Transformation (NDT) [29] is a probabilistic model-based approach. Instead of relying on point-to-point distance minimization, NDT divides the point cloud into voxels and models normal distributions in each voxel. By maximizing the fit of the point cloud to these distributions, NDT achieves faster convergence and is especially superior in noisy environments. Compared to ICP, NDT is better at handling rough initial alignments and is often used for real-time localization and map construction in autonomous driving.

Another non-ICP algorithm is Super4PCS [30], which is a global point cloud matching method based on geometric features. Super4PCS achieves fast global matching without initial alignment by using 4-point Congruent Sets as the matching datum. The algorithm achieves fast global matching without initial alignment. The algorithm shows high efficiency on large-scale point cloud data and is robust to large, noisy or partially occluded scenes.

In addition, Fast Point Feature Histogram (FPFH) [31] is an accelerated local geometric feature description method. FPFH generates compact and representative global descriptors by calculating the local geometric features of each point in the point cloud, which greatly improves the computational efficiency. Compared with the standard PFH, FPFH significantly reduces the computation time while preserving the matching accuracy. This approach performs well in dealing with noise,

partial occlusion, and cases with low initial alignment requirements.

Dealing with partial occlusions, noise and dynamic environments in point cloud matching remains a challenge for the algorithms, and KISS-ICP provides a practical solution by balancing simplicity and effectiveness, especially in real-time systems.

## 2.3 Lidar Odometry with Point Cloud

In the context of LiDAR odometry, 3D point cloud matching plays a key role in determining the pose of a robot by aligning successive point clouds to estimate the motion between them. By accurately matching the point cloud data collected at different points in time, the system can infer the motion trajectory of the device and ensure that it can realize real-time positioning in complex 3D environments.

LiDAR odometry methods typically involve two key components: feature extraction and matching. The goal of feature extraction is to extract key points with significant geometric features, such as edges, planes, etc., from the point cloud data, while feature matching is to find the correspondence between different point clouds through these key points, and then derive the bitmap transformation between them. Traditional feature extraction methods rely on geometric information, such as geometric feature extraction based on corner points and planes. With the development of deep learning techniques, the efficiency and robustness of the feature extraction and matching process have been greatly improved. For example, 3DFeatNet [32] utilizes a convolutional neural network to extract global and local features from point clouds, which significantly improves the accuracy of point cloud matching. Similar work has been done with PointNet [33] and DGCNN [34], which enable a more robust feature extraction and matching process in dynamic scenes by learning the geometric structure inside the point cloud. Advances in the field utilize deep learning techniques to enhance the feature extraction and matching process, enabling more accurate and robust odometer estimation even in challenging environments

with dynamic objects [35].

In the field of autonomous robotics, LiDAR SLAM technology is one of the keys to achieve autonomous navigation.SLAM systems require the robot to track its position while constructing a map in an unknown environment. Compared to traditional vision-based SLAM methods such as ORB-SLAM [36], LiDAR SLAM has obvious advantages in outdoor scenes due to its insensitivity to ambient lighting conditions and its ability to penetrate environmental disturbances such as light fog and light rain. Therefore, LiDAR SLAM has become a mainstream solution for applications such as autonomous driving and UAV navigation.

A typical LiDAR SLAM system is LOAM (Lidar Odometry and Mapping) [37]. LOAM uses LiDAR data for real-time odometry estimation, and provides environment awareness and path planning with a continuously updated map. To further improve the real-time performance and efficiency of SLAM, Shan and Englot proposed LeGO-LOAM [38] in 2018, which is an algorithm designed for ground-based robots, and greatly improves the computational efficiency and accuracy of the system by partitioning the point cloud into ground and non-ground points.The innovations of LeGO-LOAM are utilizing ground points for pose estimation, which in turn reduces the computational burden and allows the SLAM system to be more robust and efficient in real-time applications.

In summary, LiDAR-generated 3D point cloud data and its matching techniques have become a core tool for advancing robot perception, navigation, and environment understanding. With the introduction of deep learning methods, LiDAR odometry and SLAM systems are increasingly capable of handling dynamic environments. These technological advances have driven the rapid development of autonomous systems, especially in the field of unmanned vehicles and robotics, enabling them to perform more reliably and accurately in diverse and complex environments.

## 2.4 Robotic Operating System

The Robot Operating System (ROS) is an open-source framework for robotics that helps users and developers control devices with code and pass information between robotics applications. At the core of ROS's modular design and communication capabilities are the nodes, each of which is responsible for a module, such as controlling a motor or publishing data from a sensor. A complete ROS system consists of many cooperating nodes, each of which can send and receive data to and from other nodes by subscribing to or publishing ros topics. Figure 2.2 presents the working principle of ROS.

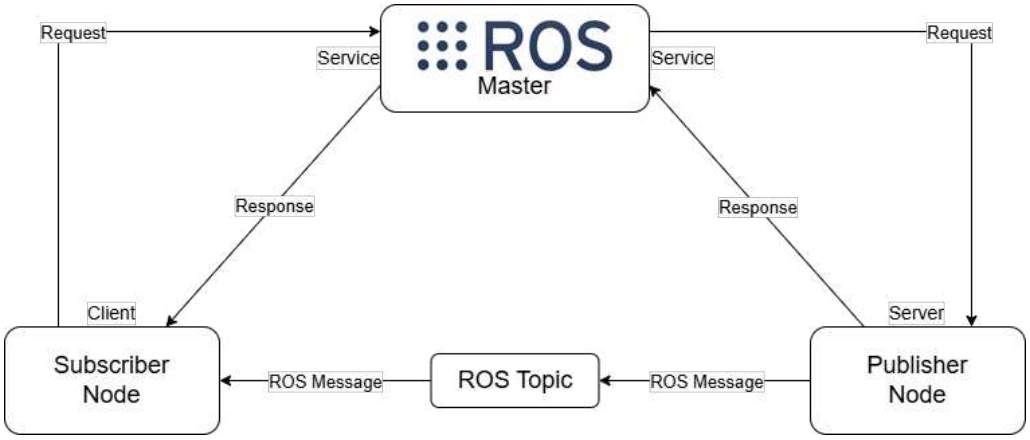



Figure 2.2: Introduction to the working principle of ROS

One of the greatest strengths of ROS is its open source nature, which fosters collaboration and innovation among a global community of developers and researchers. This allows for a variety of libraries, drivers, and algorithms, which can be easily integrated into new projects. In terms of visualization and tools, ROS includes a variety of tools such as Rviz (a 3D visualization tool) and Gazebo (a simulation environment) that enable developers to visualize sensor data, test algorithms, and simulate robot motion without the need for physical hardware. Figure2.3 shows the data publishing flow of the ROS driver and sensor.

ROS is widely used in the development of autonomous systems such as self-


Figure 2.3: ROS interfaces for the LiDAR sensors integrated

driving cars and drones. Its modular architecture and real-time data processing capabilities make it suitable for handling tasks such as sensor fusion, path planning and navigation. In manufacturing and automation, ROS supports the development of flexible robotic systems that can be easily reconfigured for different tasks, such as assembly lines or material handling.

Essentially, ROS revolutionized the way robotic software is developed by providing a unified open source platform to simplify the development, testing, and deployment of robotic systems. With the transition to ROS 2, the framework continues to evolve to meet the growing demand for real-time, secure, and scalable robotic applications. Extensive community support and a wide range of available packages make ROS an essential tool for research, education, and industry, shaping the future of robotics.

In this thesis, the datasets used are collected by various sensors controlled by ROS. The collected spatial 3D coordinates (X, Y, Z) are exchanged through the nodes, allowing us to integrate and process the information collected by the various sensors, greatly reducing the time required to tune the parameters of individual sensors.

## 2.5 LiDAR-Generated Images in Robotics

In the field of robotics, computer vision is an important research direction because it is closely related to the robot's ability to perceive, understand and interact with the environment. Computer vision acquires images or point cloud data of the surrounding environment through sensors such as cameras and LiDAR, and processes this data to enable object detection, recognition, localization, navigation, and other functions. These capabilities are critical to the proper functioning of systems such as self-driving cars, drones, and service robots.

The article [39] evaluates the effectiveness of LiDAR imagery as an input to autonomous driving and also conducts driving tests at night and in winter. It is shown that LiDAR can perform many basic autonomous driving subtasks, such as real-vehicle road tracking tasks. This result demonstrates the feasibility and utility of LiDAR data for autonomous driving applications, especially in poor lighting conditions, where LiDAR sensors have a greater advantage over traditional cameras.

To further enhance the processing capability of LiDAR data, from 2017 to 2018, researchers began to introduce deep learning techniques to enhance the analysis and interpretation of LiDAR data. For example, Chen et al. in their work [40] utilized deep learning methods to improve the target detection algorithm in LiDAR data. By combining LiDAR data with image data and with the support of convolution neural network (CNN), this method effectively improves the detection accuracy of self-driving cars in complex scenarios, especially in the multi-target detection task, which shows significant advantages. Yang et al. [41] also applied deep learning techniques in classification and recognition tasks on LiDAR data. Deep learning models are utilized to enhance the performance of self-driving cars in dynamic scenarios, especially in urban environments with complex traffic. By introducing 3D convolutional networks, this enables better processing of 3D point cloud data from LiDAR sensors.

Overall, the combination of LiDAR and deep learning has opened up new opportunities for autonomous driving technology. the high precision spatial information provided by LiDAR sensors, coupled with the power of deep learning in large-scale data processing, enables self-driving cars to sense and understand their surroundings more accurately in complex environments.

## 2.6 Keypoint Extractor

Keypoint extraction is a fundamental step in many computer vision projects, it aims to identify salient points in an image that are invariant to transformations such as rotation, scaling, and illumination changes. Over the years, several keypoint extraction methods have been proposed.

Scale-Invariant Feature Transform (SIFT) [42], [43] uses a Difference-of-Gaussian (DoG) method to detect keypoints and compute invariant descriptors to image scaling and rotation. It is highly accurate but computationally expensive. Speeded-Up Robust Features (SURF) [44] is a faster alternative to SIFT, uses box filters to approximate DoGs, thus enabling faster computation while maintaining robust performance. Features from Accelerated Segment Test (FAST) [45] is an efficient keypoint detector. It identifies keypoints by comparing the intensity of a pixel to the intensity of pixels in a circular neighborhood. FAST is designed for speed and is well suited for applications that require real-time processing. However, it does not provide orientation information, making it less robust to rotation. Binary Robust Independent Elementary Features (BRIEF) [46] constructs binary strings by comparing the intensities of random pixel pairs within smooth image blocks. Although BRIEF is extremely fast and memory efficient, it is not inherently rotation-invariant, which limits its robustness under rotation. Oriented FAST and Rotated BRIEF (ORB) [47] enhances the FAST detector by adding orientation information and coupling it to the BRIEF descriptor.

SuperPoint [48] is an end-to-end self-supervised neural network designed for feature detection and description in computer vision. It addresses the challenge of detecting and describing keypoints in images by learning from data through self-supervision. In many cases, SuperPoint outperforms traditional methods like SIFT and ORB. With its learned feature points and descriptors, the model excels in various computer vision tasks.

Accurate and Lightweight Keypoint Detection and Descriptor Extraction (ALIKE) stands out from the crowd of methods because of its focus on both accuracy and computational efficiency [49], [50]. ALIKE implements a hybrid approach that utilizes classical computer vision techniques augmented with modern machine learning methods. This enables accurate and reliable keypoint detection in a wide range of image conditions. The descriptors in ALIKE are designed to be both unique and compact to be less susceptible to image variations such as noise, lighting changes, or occlusion. This improves the high robustness against image transformations.

In summary, ALIKE represents a significant advancement in the field of keypoint extraction and descriptor extraction, providing a practical solution without compromising performance. Its combination of accuracy and efficiency makes it a valuable tool for a variety of computer vision applications.

# 3 DL-based Super-Resolution and Colorization

In this section, we endeavor to conduct the most comprehensive review possible at this time on the DL-based super-resolution and colorization approaches, paving the way for our following research attempts.

## 3.1 DL-based Colorization

In the realm of DL-based Colorization, it is commonly used in image restoration, digital medial and thermal infrared image colorization. We evaluate various coloring models and their application environments for LiDAR-generated ranging and signal images, and describe in Table 3.1 the effect of the models on LiDAR-generated images.

DeOldify [51] model is a DL-based tool that is used to colorize and restore old black and white images and videos. It utilizes the techniques of NoGAN [1] to achieve. The generator network learns to add colour to the greyscale image, while the discriminator network learns to differentiate between the real colour image and the generator-generated image. Compared to the traditional GAN model, this model prioritizes natural colors and performs well and consistently on landscapes.

[1]No paper for this GAN, it is a new type of GAN developed to solve problems in the DeOldify process. It provides GAN training advantages and minimizes the time spent on direct GAN training.

Table 3.1: Details of Deep Learning-based Colorization Models

| Model | Indoor | Outdoor | Code | Description | Language compatible |
|---|---|---|---|---|---|
| *DeOldify [51]* | √ | √ | √ | Effectiveness in restoring and colorizing old black-and-white photos | Python3 |
| *DDColor [52]* | √ | √ | √ | Utilizes diffusion models for high-quality image colorization | Python3 |
| *BigColor [53]* | √ | √ | √ | Designed for large-scale, high-resolution image colorization | Python3 |
| *BigGAN [54]* | | √ | | Generating high-resolution, high-fidelity natural images | |
| *PearlGAN [55]* | | √ | √ | Focuses on colorizing thermal and infrared images using GANs, effective in outdoor environments | Python3 |
| *I2V-GAN [56]* | | √ | √ | GAN-based model for colorizing infrared to visible spectrum images | Python3 |
| *Colorful Image Colorization [57], [58]* | √ | √ | √ | One of the first CNN-based colorization models, introducing probabilistic color assignment | Python3 |
| *Let there be Color [59]* | √ | √ | √ | A fully automatic image colorization method that works on both grayscale and natural images using deep learning | Lua |
| *DISCO [60]* | √ | √ | √ | Uses deep neural networks to provide automatic colorization with a focus on preserving detail and texture | Python3 |
| *Deep Koalarization [61]* | √ | √ | √ | Combines a pre-trained VGG network with a colorization model for realistic colorization | Python3 |
| *Palette [62]* | √ | √ | √ | A versatile colorization model that leverages palette-based techniques to achieve high-quality results across various image types | Python3 |
| *ChromaGAN [63]* | √ | √ | √ | GAN-based model that emphasizes perceptual loss for natural colorization results | Python3 |
| *InstColorization [64]* | √ | √ | √ | An instance-aware colorization model that handles complex scenes by focusing on individual object instance | Python3 |
| *SCGAN [65]* | √ | √ | √ | A self-consistent GAN-based colorization approach that ensures color consistency across different regions of an image | Python3 |
| *pix2pix [66]* | √ | √ | √ | A general-purpose image-to-image translation model | Python3 |

The core concept of the DDColor [52] model is the use of two distinct decoders for image colorization. The Pixel Decoder focuses on recovering spatial details and image structure, while the Colour Decoder learns semantically aware color representations from visual features at multiple scales. By combining the outputs of both decoders, DDColor aims to produce more natural and vivid colors, especially in complex scenes with multiple objects.

BigColor [53] model serves as a precursor to the DDColor model, which builds upon and refines the foundations established by BigColor. This model introduces

a novel approach based on BigGAN [67], expanding the representation space and offering a range of coloring results.

BigGAN is a colorization model for generating high-resolution, high-fidelity natural images based on Large Scale GAN Training for High Fidelity Natural Image Synthesis [68] training. The framework consists of a generator and a discriminator. The generator creates images, the discriminator evaluates them against real images, and then iteratively trains to generate high quality images.

PearlGAN model [55] is tailored for converting nighttime thermal infrared (NTIR) images into daytime color (DC) images. It excels at producing high-quality colorization for NTIR images of open roads. However, its application is restricted to outdoor scenes, and its colorization capabilities are primarily focused on roads and trees, particularly in LiDAR-generated images.

I2V-GAN model [56] is designed for unpaired infrared-to-visible video translation. Like PearlGAN, it is specialized for outdoor scenes and works with infrared images. The model facilitates the conversion of infrared video data into visible spectrum equivalents, although it shares similar limitations in its applicability to specific scene types and environments.

Let There Be Color [59] is a Convolutional Neural Network (CNN) model trained using paired grayscale and color images. The model learns to colorize grayscale images by leveraging the correlations between the grayscale inputs and their corresponding color images. It employs a deep architecture to capture complex patterns and color distributions, resulting in visually appealing colorization of grayscale images. The approach effectively handles a variety of image types, providing robust colorization across different scenes.

DISCO model [60] provides a new method of colouring images by separating colour representation from spatial information. This method helps to produce images with vibrant and realistic colors.

Deep Koalarization [61] combines CNNs with the advanced Inception-ResNet-v2 architecture, allowing it to capture complex details and nuances in images. The model leverages Inception modules for efficiently handling multi-scale features, while ResNet's residual connections enable deeper networks to be trained without encountering gradient vanishing issues. However, the model's complexity and high computational demands require significant processing power and memory. Paltte [62] is similar to Deep Koalarization, as it reproduces the full model.

ChromaGAN [63] by utilizing an adversarial framework, ChromaGAN's network of generators and discriminators work in tandem to produce high-quality color images. Where the generator network learns to generate color images from grayscale inputs, the discriminator network evaluates the veracity of these colorizations. However, like many GAN-based models, ChromaGAN requires significant computational resources and is a complex and time-consuming process.

Instance-aware Image Colorization [64] Enhance coloring accuracy and fidelity by incorporating instance-level perception into the process. The model distinguishes between different instances of the same object class in an image, enabling more accurate and consistent application of color. However, the complexity of the model and instance-level segmentation increases the computing time.

Saliency Map-Guided Colorization With Generative Adversarial Network (SCGAN) [65] is the use of saliency maps to inform the generator network within the GAN framework to ensure that the generated colors are visually appropriate, which is particularly beneficial for images with complex compositions or where certain objects should stand out.

Image-to-Image Translation with Conditional Adversarial Networks (pix2pix) [66] enables the model to perform transformations that are both visually compelling and contextually relevant by introducing a versatile framework that uses conditional generative adversarial networks (cGANs) to transform one type of image into an-

Table 3.2: Details of Deep Learning based Super Resolution Models

| Model | Code | Architecture | Training Data | Strengths | Weaknesses | Language compatible |
|---|---|---|---|---|---|---|
| *SRCNN [69]* | | 3-layer CNN | ImageNet | Simple, Effective | Limited long-range dependency capture | |
| *VDSR [70]* | | 20-layer CNN | ImageNet | Deep architecture, Residual learning | High computational cost | |
| *SRGAN [71]* | | CNN with GAN | ImageNet, DIV2K | High perceptual quality | Potential artifacts | |
| *DRRN [72]* | | Recursive Residual Network | ImageNet, DIV2K | Parameter efficient, Deep architecture | High computational cost | |
| *CARN [73]* | √ | Cascading Residual Network | DIV2K | Lightweight, Fast inference | Slightly lower accuracy | Python3 |
| *SwinIR [74]* | √ | Transformer-based | DIV2K, Flicker2K | High performance, Handles large scale factors | High computational cost | Python3 |
| *SCUNet [75]* | √ | CNN with Spatially Consistent Normalization | DIV2K | Maintains spatial consistency, Effective edge preservation | High computational resources, Training complexity | Python3 |
| *ESRGAN [76]* | √ | GAN with Residual-in-Residual Dense Blocks | DIV2K, Flicker2K | Realistic textures, Superior perceptual quality | High computational cost, Can produce artifacts | Python3 |
| *DCSCN [77]* | √ | CNN and Sparse Coding | CIFAR-10, ImageNet | Efficient high frequency detail capture | Struggles with complex images | Python3 |
| *RGT [78]* | √ | Recursive Generation Transformer | DIV2K, Flickr2K | Excellent handling of recursive structures | High computational demands | Python3 |
| *CAT [79]* | √ | Cross-Attention Transformer | DIV2K, Flickr2K | Strong cross-modal learning, High fidelity | Computationally expensive | Python3 |
| *Perceptual Losses [80]* | | CNN with Perceptual Loss | MS-COCO, ImageNet | High-quality image generation, Real-time performance | Dependent on pre-trained networks, Generalization limits | |
| *SinGAN [81]* | √ | GAN with Single Image Learning | Single Image | Effective for artistic enhancement, Versatile | Training on each image required, Limited scalability | Python3 |
| *DnCNN [82], [83]* | √ | Deep CNN with Batch Normalization | ImageNet, BSD400 | Effective denoising, Simple architecture | Limited to small scale factors | Python3 |
| *SR3 [84]* | √ | Diffusion Model for SR | CelebA-HQ, FFHQ | Generates realistic and high-quality images | High computational cost, Slow inference | Python3 |
| *EDSR [85]* | | Enhanced Deep Residual Network | DIV2K | High accuracy, No batch normalization | Requires extensive training, Large memory footprint | |

other. However, the model is more computationally demanding and may encounter problems such as pattern collapse.

To sum up, although there are many DL based colorization models, they are mainly designed for black and white images and thermal infrared images, and no colorization models specifically for LiDAR-generated images. DeOldify [51] performed better than the other models in all aspects, therefore, this model was used as the colorization method for this research.

## 3.2 DL-based Super Resolution

DL has greatly advanced the development of image super-resolution (SR), enabling the models to generate high-quality magnified images from low-resolution inputs. This part provides a comparative analysis of various DL super-resolution models, describing their architecture, performance, programming language compatibility, ad-

vantages, and disadvantages, as shown in Table 3.2.

Super-Resolution Convolutional Neural Network (SRCNN) [69] is one of the pioneering deep learning models for image super-resolution. Its three-layer CNN is designed for feature extraction, non-linear mapping, and reconstruction. Trained on ImageNet with mean square error (MSE) as the loss function, SRCNN aims to minimize the difference between the output and the ground truth image. It achieves strong results on datasets like Set5, Set14, and BSD. However, due to its shallow architecture, SRCNN struggles to capture long-range dependencies in complex images, limiting its overall performance.

Very Deep Super Resolution (VDSR) [70] model builds on the SRCNN framework by employing a much deeper CNN with 20 layers. This increased depth enables VDSR to learn more complex mappings from low- to high-resolution images, significantly improving performance. To mitigate the vanishing gradient problem, the model incorporates residual learning. VDSR consistently outperforms SRCNN on benchmark datasets, showcasing the benefits of its deeper architecture for super-resolution tasks. However, this added depth also results in higher computational costs.

Super-Resolution Generative Adversarial Network (SRGAN) [71] was the first to apply generative adversarial networks (GANs) to super-resolution tasks. SRGAN comprises two neural networks: a generator that creates realistic high-resolution (HR) images from low-resolution (LR) inputs, and a discriminator that distinguishes between generated images and real HR photos. This adversarial framework pushes the generator to produce increasingly convincing results. However, the adversarial training can introduce artifacts, and achieving a balance between the generator and discriminator remains challenging.

Deep Recursive Residual Network (DRRN) [72] leverages deep recursive learning with recursive residual blocks to effectively capture dependencies across different

scales. The model employs a recursive structure with shared parameters, allowing it to increase network depth without significantly inflating the number of parameters. However, the recursive design complicates the training process and can be computationally intensive.

Cascading Residual Network (CARN) [73] is a ResNet-based image super-resolution model designed for real-time applications. By using cascading connections, CARN reduces computational cost while retaining high performance, enabling it to merge features across layers and capture intricate image details more effectively. Trained on large-scale super-resolution datasets, CARN strikes a balance between speed and accuracy, though its efficiency-focused design can lead to slight compromises in image quality compared to more complex models.

Swin Transformer for Image Restoration (SwinIR) [74] leverages the Swin Transformer architecture for tasks like super-resolution. By dividing images into smaller patches and analyzing them at multiple scales, SwinIR captures both local details and long-range dependencies. This method enhances the model's understanding of the image's global structure, although it increases computational complexity.

Swin Conv UNet (SCUNet) [75] introduces a novel method for enhancing image quality by removing noise without prior knowledge of its properties. This is accomplished through the integration of the Swin Conv UNet model with a strategic data synthesis process. SCUNet outperforms traditional methods and other state-of-the-art models in denoising tasks, effectively eliminating noise while preserving image details and textures. However, its high computational requirements and reliance on high-quality synthetic data are key considerations for practical deployment.

Enhanced Super-Resolution Generative Adversarial Network (ESRGAN) [76] improves upon the SRGAN model by introducing Residual Dense Blocks (RRDB) and Perceptual Loss Functions to enhance image realism. ESRGAN trains generators and discriminators simultaneously, resulting in superior image quality. However,

the model is computationally demanding, requiring substantial processing power, memory, and numerous iterations for optimal results. Additionally, ESRGAN can produce artifacts, particularly in images with complex textures.

Deep CNN with Skip Connection and Network (DCSCN) [77] leverages deep Convolutional Neural Networks (CNNs), skip connections, and Networks-in-Networks (NINs) architectures to enhance the quality of super-resolution images while maintaining computational efficiency. This approach enables the generation of high-quality images with precise details and textures. However, despite its efficiency-focused design, the deep architecture and NIN layers still demand considerable processing power and memory. Additionally, if not properly normalized, the model's powerful feature extraction capabilities may lead to overfitting, impacting performance on unseen data.

Recursive Generalization Transformer for Image Super-Resolution (RGT) [78] uses a recursive structure within the Transformer framework that allows it to refine image features over multiple iterations recursively. This recursive approach helps to incrementally improve image quality by refining learned features and capturing more complex patterns, resulting in superior super-resolution results. However, the recursive structure of the model is computationally intensive and requires a lot of processing power and memory.

Cross Aggregation Transformer for Image Restoration (CAT) [79] utilizes a cross-attention mechanism to aggregate features from various parts of the image, allowing it to recover finer details and textures that are often lost in traditional methods, and also ensuring that the model can more efficiently handle a variety of restoration tasks including denoising, deblurring, and SR. However, the complexity and computational demands of the transformer architecture can be challenging and require significant processing power.

Perceptual Losses [80] utilizes a perceptual loss function to enhance the visual

quality of generated images, using high-level feature differences from a pre-trained CNN, rather than relying on traditional pixel-by-pixel comparisons. Despite its advantages, the model's reliance on pre-trained networks for perceptual loss calculations may limit its adaptability across different datasets, as performance may depend on the characteristics of the specific network used.

SinGAN [81] generates images by training a generative adversarial network (GAN) on a single image, enabling the model to generate a variety of outputs that maintain the visual features of the original image, including texture, color, and structure. While SinGAN's ability to process a single image is an advantage, the model is computationally intensive and the results can sometimes lack diversity compared to models trained on large datasets. In addition, the quality of the generated images depends heavily on the characteristics of the original image.

Deep CNN with Skip Connection and Network (DnCNN) [82], [83] employs a deep CNN combined with NIN architecture and skip connections. These connections help integrate NIN layers, enhancing the model's ability to capture fine details and represent high-frequency components, improving texture and edge preservation in super-resolution images. However, without adequate regularization, the model's strong feature extraction capabilities can lead to overfitting, potentially affecting performance on unseen data.

SR3 [84] enhances the image resolution in multiple stages, with each iteration refining the output of the previous stage. This progressively improves the image quality, resulting in finer details and more accurate textures in the final high-resolution image. However, the iteration of the process can lead to an increase in computational requirements, as multiple passes are required to obtain the final result. This can limit its use in real-time applications or environments with limited computational resources.

Based on the traditional deep residual network (ResNet), Enhanced Deep Resid-

ual Networks for Single Image Super-Resolution (EDSR) [85] has optimized its architecture for super-resolution tasks by removing unnecessary modules and introducing a deeper and wider network structure that enables it to capture more complex patterns and finer details in images. However, the increased depth and complexity of the EDSR model come with the cost of higher computational requirements, making it more demanding in terms of processing power and memory.

The comparison of these models include evolution from simple CNNs to complex transformer-based architectures. The summary in Table 3.2 provides a clear visual comparison.

# 4 Methodology

In this section, we illustrate the methodology of this study. Initially, we explain the qualitative evaluation of the DL-Based Super Resolution and Colorization on LiDAR-generated images. Subsequently, we introduce our proposed algorithm, which integrates these approaches into the point cloud registration within the context of LO.

## 4.1 Dataset for Evaluation

We carry out all evaluations in the experiment with the published open-source multi-modal LiDAR datasets [2], [86]. However, for the purposes of this study, we specifically utilize data from the Ouster LiDAR, the detailed specifications of which are provided in Table 4.1. Ouster LiDAR generates an incredibly dense point cloud along with various types of images. These images include range images, signal images, and ambient images, each encoded with specific information: depth data, infrared intensity, and ambient light intensity, respectively. We particularly use range and signal images as they have been proven effective enough in the key point extraction [5].

Table 4.1: Specifications of Ouster OS0-128.

| | **IMU** | **Type** | **Channels** | **Image Resolution** |
|---|---|---|---|---|
| **Ouster OS0-64** | ICM-20948 | spinning | 128 | $1024 \times 128$ |
| **FoV** | **Angular Resolution** | **Range** | **Freq** | **Points** |
| $360° \times 90°$ | $V : 0.7°, H : 0.18°$ | 50 m | 10 Hz | 2,621,440 pts/s |

The data sequences used for evaluation include indoor and outdoor environments. The outdoor environment is from the normal road and a forest, denoted as *Open road* and *Forest*, respectively. The indoor data includes a hall in a building and two rooms, denoted as *Hall (large)*, *Lab space (hard)*, and *Lab space (easy)*, respectively. The *Forest* dataset we recorded was collected within a forested area, with a limited traversal distance of approximately 12 meters due to the constraints of our motion capture system. Thus, although the dataset was recorded outdoors, the environment is relatively confined.

## 4.2 Analysis of Colorization Approaches

In Section 3.1, we conduct a comprehensive survey on the DL-based colorization Approaches. In this part, we qualitatively analyze the performance of the approaches on LiDAR-generated images (signal images), utilizing publicly available code repositories on GitHub. The selection of evaluation methods is based not only on the availability of their implementations but also on the simplicity of their deployment for a standard laptop. More specifically, the approaches examined include *BigColor, Colorful Image Colorization, DDColor, DeOldify, DISCO, InstColorization, Let there be color, PearlGAN*, as detailed in shown in Table 3.1.

## 4.3 Analysis of Super-Resolution Approaches

The review of the DL-based super-resolution approaches is in Section 3.2. Among these approaches, we select the methods based on the same principle mentioned above: the availability of implementation and the simplicity of deployment for a standard laptop. The approaches included in the part are *CARN, SwinIR, DCSCN, ESRGAN, SCUNET*, as detailed in Table 3.2. We qualitatively evaluate the super-resolution performance of these approaches on signal images.

## 4.4 Evaluation Scheme of the Proposed Algorithm

### 4.4.1 Overall Pipline

The overall pipeline of the proposed algorithm is illustrated in Figure 4.1. Compared to our previous work [5], the proposed approach integrates DL-based super-resolution and colorization models, along with key point extractors utilizing the RGB three channels from colorized images within the processing pipeline.

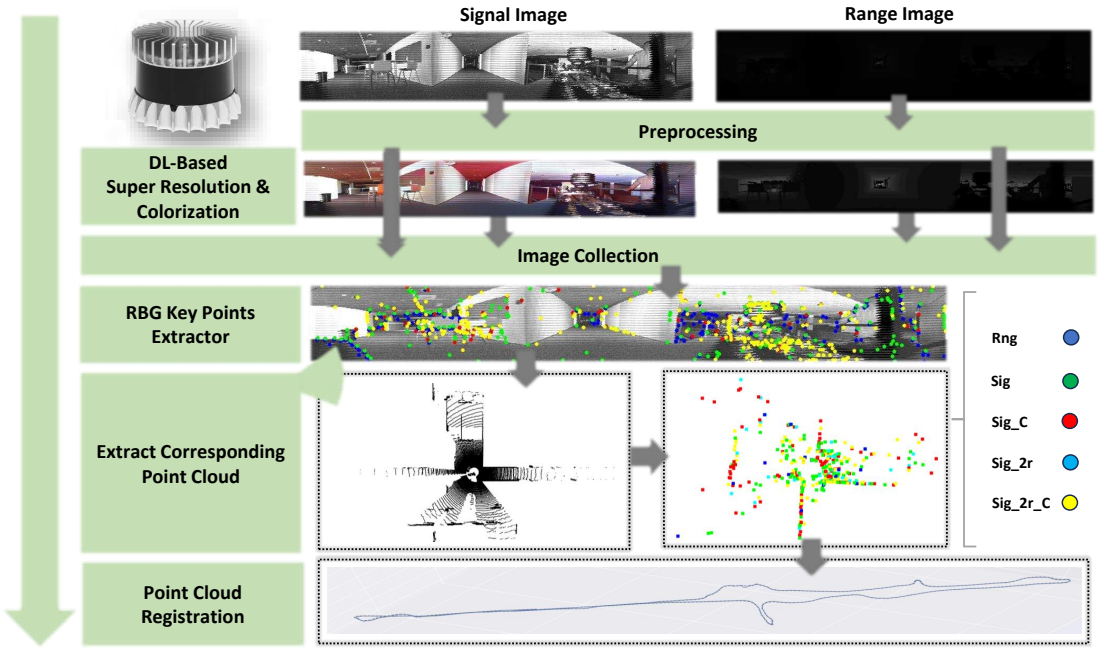



Figure 4.1: The system overview of the proposed approach

In the evaluation, we applied DL-based colorization approaches to the signal images, as our findings indicated that applying the models to the range images did not significantly improve in keypoint extraction.

### 4.4.2 Combination of Super Resolution and Colorization

Given that super-resolution results in varying image sizes within the system, we conducted the evaluation using different combinations of images produced through super-resolution and colorization techniques. Table 4.2 shows the exact combination

we applied in the experiment. It is worth noting that our findings indicate that the resolution size did not significantly affect the results of the effective key point extraction if it is above 2.

| Name | Combinations |
|---|---|
| $comb\_0$ | $[rng \;\; sig \;\; sig_{2r}]$ |
| $comb\_1$ | $[rng \;\; sig \;\; sig_c]$ |
| $comb\_2$ | $[rng \;\; sig \;\; sig^c_{2r}]$ |
| $comb\_3$ | $[rng \;\; sig \;\; sig^c \;\; sig_{2r} \;\; sig^c_{2r}]$ |
| $comb\_4$ | $[rng \;\; rng_{2r} \;\; sig \;\; sig^c \;\; sig_{2r} \;\; sig^c_{2r}]$ |
| $comb\_5$ | $[rng \;\; rng_{2r} \;\; sig_{2r} \;\; sig^c_{2r}]$ |
| $comb\_6$ | $[rng \;\; rng_{2r} \;\; sig \;\; sig^c_{2r}]$ |

Table 4.2: The different combinations of images from DL-based super-resolution and colorization are denoted as follows: $rng$ represents range images, $sig$ represents signal images, $_{2r}$ indicates a resolution size increased by a factor of two, and $^c$ denotes the application of colorization.

### Selection of DL-based Methods

The super-resolution and colorization models utilized in this study are *CARN* and *DeOldify*, respectively. This selection is based on our analysis of both result quality and inference speed. For keypoint extraction, we employed the *Alike* model.

### Point Cloud Registration

In this experiment, KISS-ICP [1] is employed as the method for calculating LO. Specifically, we disable the sampling functionality within the KISS-ICP code, as our approach focuses on sampling the point cloud using key point extractors.

### Evaluation Metrics

We are particularly interested in understanding how the DL-based super-resolution and colorization techniques contribute to mitigating drift error and improving the

[1] https://github.com/PRBonn/kiss-icp.git

accuracy of the LO system. To evaluate the accuracy of the LO, we calculated the translation and rotation errors using the tool evo [2]. Additionally, we quantified the number of extracted point clouds to assess the effectiveness of these techniques.

**Evaluation Scheme**

The detailed evaluation scheme is outlined in Algorithm 1. In this scheme, we systematically iterate through all image combinations with the corresponding DL-based approaches applied, listed in Table 4.2. We execute the full sequence of processes for each combination, including preprocessing, super-resolution, colorization, keypoint extraction, and LO calculation, followed by the computation of translation and rotation errors.

Given that LiDAR-generated images typically appear dark, a key objective during the preprocessing stage is to apply gamma compensation to enhance image brightness (Lines 6 - 13). Unlike range images, signal images exhibit highly uneven exposure across different regions. To address this issue, we first retain pixels with pixel values exceeding a predefined threshold, denoted as $p_{thresh} = 240$ . For pixel values below this threshold, adaptive histogram equalization (CLAHE) is applied to enhance details in the darker regions of the signal image. Following the preprocessing stage, super-resolution and colorization techniques were applied to the images as needed, depending on the specific combination of methods. Additionally, the key point extraction process was integrated into these enhanced images for further analysis.

To ensure greater robustness and consistency in key points across different image frames, we employed the Mutual Nearest Neighbor Matching(MNN) algorithm to match the key points extracted between the current and previous frames. We retained only the matched key points for subsequent processing.

[2]https://github.com/MichaelGrupp/evo

After obtaining the robust key points, we apply the aforementioned LO approach and calculate the translation error and rotation error using the EVO tool.

### Hardware Information

The evaluation was conducted using a Razer Blade 15 laptop by Ubuntu 22.04.4 LTS equipped with an Intel Core i7-12800H-20 processor, 16 GB of RAM with a frequency of 4800 MHz, and a GeForce RTX 3070 Ti GPU with 8 GB of memory.

**Listing 1** The evaluation scheme

**Input:**

Range image: $rng$
Signal image: $sig$
Point cloud: $pc$
Combinations in Table 4.2: $combs = [comb\_i]$, $i \in (0 \sim 6)$

**Output:**

Translation error: $trans\_err$ $(mean/rmse, unit : m)$
Rotation error: $rot\_err$ $(mean/rmse, unit : °)$

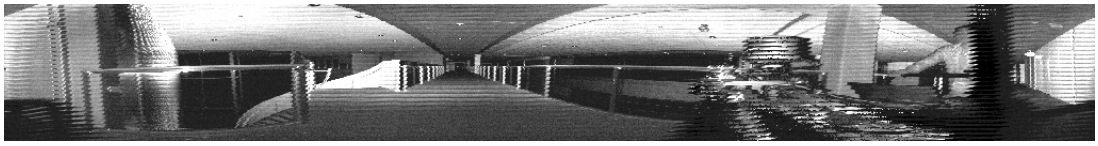
```
/* Variable Declarations:  */
```

$kps_t$ : Current Key points $mkpts_t$ : matched Key points $pc_{kp}$: Point cloud corresponded to key points $LOC$ : LiDAR odometry calculation $GT$ : Ground truth

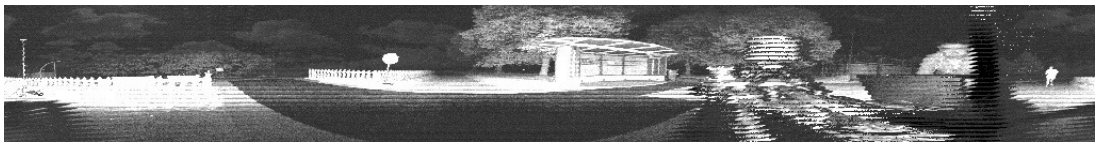
```
/* Image preprocess for brighter images */
def img_preprocess(img):
    if img == rng then
        img_prc ← gamma (img)
    else
        if pixel_value < p_thresh then
            img_hist ← hist_equalizer (img)  img_prc ← gamma (img_hist)
    return img_prc;

/* Key point detect & track */
def kp_tracker(kps_t, kps_{t-1}):
    matches ← MnnMatcher(kps_{t-1}, kps_t)  mkps ← kps_t[matches]  return mkps
rng = img_preprocess(rng)  rng_{2r} ← super_resolution(rng)
  sig = img_preprocess(sig)  sig^c ← colorization(sig)
  sig_{2r} ← super_resolution(sig)
  sig^c_{2r} ← colorization(sig_{2r})
  /* Arrange and combine sig, rng, rng_{2r}, sig_{2r},sig^c_{2r},sig^c to be comb in table 4.2 */
foreach comb_i in combs do
    foreach img in comb_i do
        kps_t ← keypoint_detect (img)  mkpts_t ← kp_tracker(kps_t, kps_{t-1})
        pc^i_kp ← pc[index[mkpts_t]]
    pc_kp ← Combine(pc^i_kp)
    Odom ← LOC(pc_kp)
    trans_err, rot_err ← EVO(Odom,GT)
```

# 5 Experimental Results

In this section, we present the qualitative evaluation results of DL-based super-resolution and colorization models. Following this analysis, we select the model that, based on our experimental findings, is more suitable in terms of both processing speed and output quality. These selected models are then applied to our point cloud sampling approach, where we further analyze its performance in terms of accuracy and the number of points extracted.

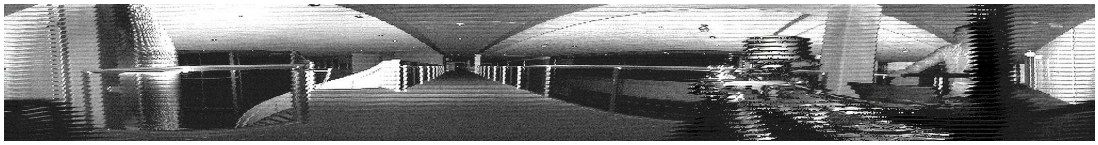


(a) Raw image indoor



(b) Raw image outdoor

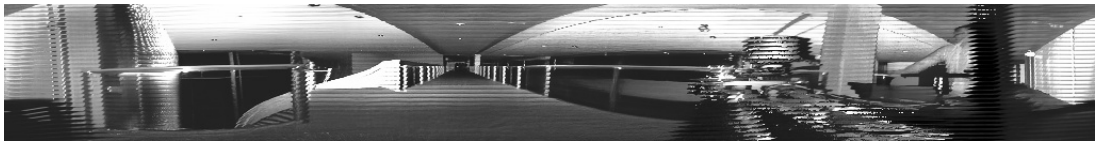
Figure 5.1: Raw images of indoor and outdoor environment



(a) Example of CARN super-resolution indoor image

(b) Example of CARN super-resolution outdoor image

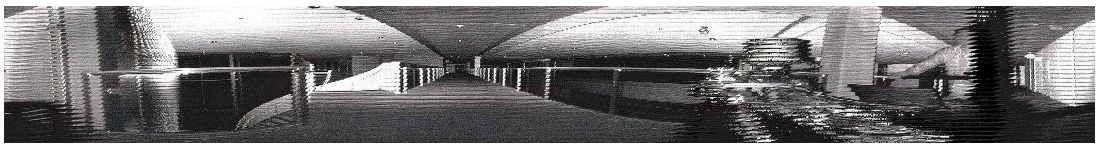
Figure 5.2: Example of CARN super-resolution



(a) Example of SCUNET super-resolution indoor image

(b) Example of SCUNET super-resolution outdoor image

Figure 5.3: Example of SCUNET super-resolution



(a) Example of ESRGAN super-resolution indoor image

(b) Example of ESRGAN super-resolution outdoor image

Figure 5.4: Example of ESRGAN super-resolution

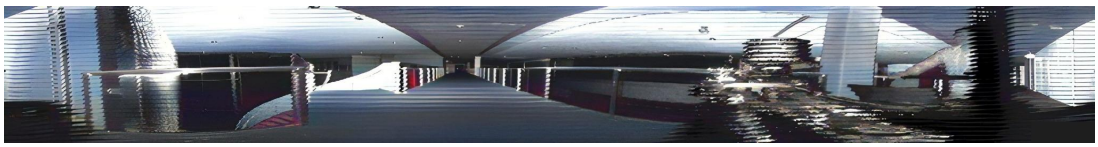
## 5.1 Super-Resolution of LiDAR-generated Images

The example processed images in Figure 5.2 and Figure 5.3, like most super-resolution models, the final rendering for the signal image has a higher degree of sharpening

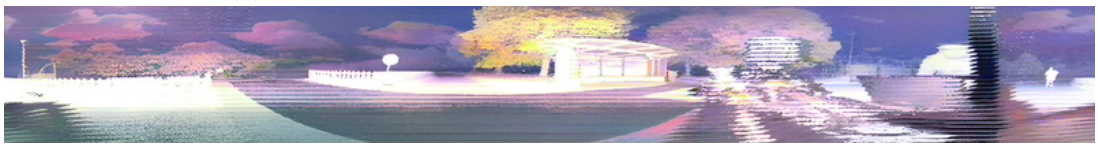
or an overall smoother image than the raw image in Figure 5.1. Table 5.2 shows the inference speeds of the multiple popular DL-based super-resolution models with input and output sizes illustrated. The DL model *CARN* demonstrates relatively high result quality, as illustrated in Figure 5.2, while also exhibiting fast processing speed, as shown in Table 5.2.

| **Model** | **Input size** | **Output size** | **Running speed (secs/image)** |
|---|---|---|---|
| CARN | (1024, 128) | (2048, 256) | 0.005 |
| SwinIR | (1024, 128) | (4096, 512) | 2.217 |
| DCSCN | (1024, 128) | (2048, 256) | 0.238 |
| ESRGAN | (1024, 128) | (4096, 512) | 2.667 |
| SCUNET | (1024, 128) | (1024, 128) | 1.706 |

Table 5.1: Comparison of running speed of each super-resolution model based on local environment



(a) Example of DeOldify colorization indoor image



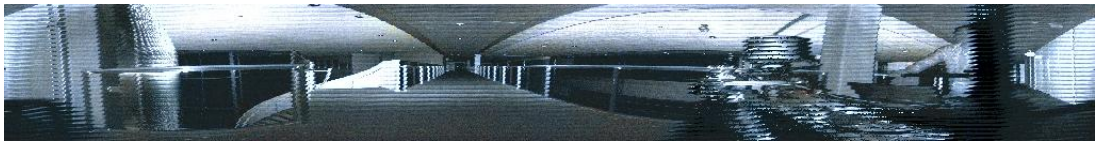
(b) Example of DeOldify colorization outdoor image

Figure 5.5: Example of DeOldify colorization

(a) Example of DDColor colorization indoor image

(b) Example of DDColor colorization outdoor image

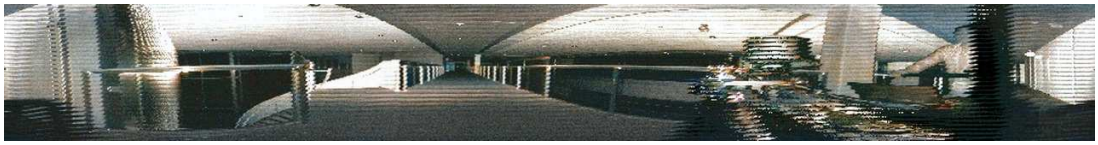
Figure 5.6: Example of DDColor colorization



(a) Example of BigColor colorization indoor image

(b) Example of BigColor colorization outdoor image

Figure 5.7: Example of BigColor colorization



(a) Example of Paltte colorization indoor image

(b) Example of Paltte colorization outdoor image

Figure 5.8: Example of Paltte colorization

## 5.2 Colorization of LiDAR-generated Images

Figure 5.5 and Figure 5.6 show the result of the two colorization models after coloring, which present different results due to the differences in the training datasets. Table 5.2 shows the inference speed of the multiple popular DL-based colorization models with input and output sizes illustrated. The DL model *DeOldify* produces good colorization results, as shown in Figure 1.2 and Figure 5.5, while demonstrating relatively faster processing speed, as indicated in Table 5.2.

| Model | Input size | Output size | Running speed (secs/image) |
|---|---|---|---|
| BigColor | (1024, 128) | (1024, 128) | 0.54 |
| Colorful Image Colorization | (1024, 128) | (1024, 128) | 0.27 |
| DDColor | (1024, 128) | (1024, 128) | 0.37 |
| DeOldify | (1024, 128) | (1024, 128) | 0.23 |
| DISCO | (1024, 128) | (1024, 128) | 3.27 |
| InstColorization | (1024, 128) | (256, 256) | 0.05 |
| PearlGAN | (1024, 128) | (1024, 128) | 0.28 |

Table 5.2: Comparison of running speed of each colorization model based on local environment

## 5.3 Our Proposed Point Cloud Sampling Approach

To assess the efficacy of our proposed point cloud sampling approach, as outlined by various combinations in Table 4.2, we conducted an evaluation using ICP-based LO to calculate both its translation and rotation errors. The resulting errors, along with comparative data from previous studies under different scenarios, are presented in Table 5.3. The data indicate that our sampling method performs particularly well in more expansive environments, such as on *Open road* and an *Hall(large)* datasets. Furthermore, the rotation errors associated with our method are generally lower than those reported in prior studies. However, in more confined spaces, such as *Forest*, *Lab space(hard)*, and *Lab space (easy)*, our method exhibits slightly higher translation errors compared to existing approaches.

It is important to note that our current methodology does not incorporate neighboring points around the key points. This exclusion results in a significantly lower point density, as evidenced in Table 5.4, while still maintaining relatively competitive accuracy, as shown in Table 5.3.

| Combination | | Open road | Forest | Lab space (hard) | Lab space (easy) | Hall (large) |
|---|---|---|---|---|---|---|
| | | (Translation error (mean/rmse), rotation error (mean)) | | | | |
| comb_0 | | (1.055/1.222, 1.782) | (0.086/0.102, 1.666) | (0.094/0.107, 1.185) | (0.083/0.095, 1.143) | (0.457/0.500, **0.903**) |
| comb_1 | | (0.605/0.731, 1.855) | (0.087/0.103, 1.663) | (0.120/0.136, 1.398) | (0.098/0.115, 1.216) | (0.485/0.532, 0.955) |
| comb_2 | | (**0.357**/**0.409**, 1.808) | (0.087/0.102, 1.668) | (0.126/0.149, 1.473) | (0.097/0.110, 1.186) | (0.456/0.500, 0.903) |
| comb_3 | | (5.726/6.738, 1.957) | (0.086/**0.102**, 1.666) | (0.045/0.050, **0.721**) | (0.032/0.036, **0.657**) | (**0.438**/**0.476**, 0.920) |
| comb_4 | | (1.464/1.641, 1.806) | (0.086/0.102, **1.662**) | (0.092/0.104, 1.235) | (0.080/0.095, 1.066) | (1.175/1.326, 0.929) |
| comb_5 | | (1.052/1.307, 1.807) | (0.088/0.103, 1.672) | (0.101/0.114, 1.213) | (0.089/0.105, 1.118) | (0.456/0.503, 0.922) |
| comb_6 | | (0.492/0.595, 1.806) | (0.086/0.103, 1.668) | (0.109/0.123, 1.319) | (0.093/0.107, 1.156) | (0.452/0.498, 0.929) |
| Prior work [5] (*rng_sig*) | 4_7 | (0.817/0.952, 2.33) | (0.082/0.102, 7.88) | (0.039/0.046, 1.46) | (0.027/0.033, 0.98) | (0.583/0.660, 2.88) |
| | 5_5 | (2.176/2.410, **1.76**) | (0.108/0.203, 6.96) | (0.037/**0.043**, 1.35) | (0.027/0.032, 0.97) | (0.707/0.801, 2.66) |
| | 7_5 | (1.784/2.006, 2.30) | **0.080**/0.102, 7.22) | (**0.033**/0.047, 1.59) | (**0.025**/**0.028**, 0.97) | (0.698/0.803, 3.11) |

Table 5.3: Comparison of translation and rotation errors across various combinations of DL-based super-resolution and colorization methods shown in Table 4.2, benchmarked against the results reported in prior work [5]. In the table, *sig* and *rng* represent the size of neighboring point areas for the signal and range images, respectively, denoted as *sig_ rng*.

| Combination | | Open road | Forest | Lab space (hard) | Lab space (easy) | Hall (large) |
|---|---|---|---|---|---|---|
| | | (Number of Points (pts)) | | | | |
| comb_0 | | 1149 | 1131 | 1613 | 1589 | 1492 |
| comb_1 | | 628 | 826 | 999 | 970 | 875 |
| comb_2 | | 787 | 820 | 1264 | 1229 | 1093 |
| comb_3 | | 1310 | 1167 | 1796 | 1270 | 1731 |
| comb_4 | | 1317 | 1170 | 2053 | 2031 | 1737 |
| comb_5 | | 1028 | 732 | 1587 | 1571 | 1322 |
| comb_6 | | 793 | 823 | 1396 | 1370 | 1099 |
| Prior work [5] (*rng_sig*) | 4_7 | 4784 | 11447 | 9518 | 9392 | 7094 |
| | 5_5 | 3183 | 7568 | 6446 | 6292 | 4783 |
| | 7_5 | 4756 | 11627 | 9469 | 9378 | 7078 |

Table 5.4: Comparison of the number of points across various combinations of DL-based super-resolution and colorization methods shown in Table 4.2, benchmarked against the results reported in prior work [5].In the table, *sig* and *rng* represent the size of neighboring point areas for the signal and range images, respectively, denoted as *sig_ rng*.

# 6 Conclusion and Future Work

## 6.1 Conclusion

This paper introduces a novel approach to point cloud sampling, specifically designed to mitigate drift error during the point cloud registration phase of LO. In contrast to our previous work, this study employs a DL-based super-resolution and colorization technique to enhance the key point extraction process for LiDAR-generated images. This approach is supported by a more comprehensive review of existing DL-based super-resolution and colorization methods compared to other literature. Our review demonstrates that these DL-based techniques perform effectively on LiDAR-generated images. Furthermore, the proposed sampling method surpasses our previous work in terms of rotation error across most datasets and translation error in more open environments. However, it exhibits reduced accuracy in translation errors within more confined spaces.

The most computationally expensive and time-consuming phase is DL-based colorization, as it operates at the pixel level and is applied in parallel to multiple images in this study. The system currently achieves a processing frequency of approximately 4 Hz. However, this performance could be further improved through distributed computing, integration with other methods such as nearest points searching, and advancements in lightweight colorization models.

## 6.2 Future Works

Our findings suggest a promising strategy for reducing drift. In future work, this approach could be integrated into the entire LO process, such as by combining it with existing LO and SLAM methods, like Faster-LIO. Additionally, during our evaluation of the colorization and super-resolution models on LiDAR images, we observed that the existing models are primarily designed and trained on camera images rather than LiDAR images. While these models have shown significant improvements in the quality of LiDAR images, we believe their potential has yet to be fully realized. Therefore, future research could focus on developing and training colorization and super-resolution models specifically tailored for LiDAR images, which could further enhance the accuracy and performance of these techniques in the context of LO and SLAM systems.

# References


[1] S. Jiang, S. Wang, Z. Yi, M. Zhang, and X. Lv, "Autonomous navigation system of greenhouse mobile robot based on 3d lidar and 2d lidar slam", *Frontiers in plant science*, vol. 13, p. 815 218, 2022.

[2] H. Sier, Q. Li, X. Yu, J. Peña Queralta, Z. Zou, and T. Westerlund, "A benchmark for multi-modal lidar slam with ground truth in gnss-denied environments", *Remote Sensing*, vol. 15, no. 13, p. 3314, 2023.

[3] X. Yu, S. Salimpour, J. P. Queralta, and T. Westerlund, "General-purpose deep learning detection and segmentation models for images from a lidar-based camera sensor", *Sensors*, vol. 23, no. 6, p. 2936, 2023.

[4] P. Pfreundschuh, H. Oleynikova, C. Cadena, R. Siegwart, and O. Andersson, "Coin-lio: Complementary intensity-augmented lidar inertial odometry", in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2024, pp. 1730–1737.

[5] H. Zhang, X. Yu, S. Ha, and T. Westerlund, "Lidar-generated images derived keypoints assisted point cloud registration scheme in odometry estimation", *Remote Sensing*, vol. 15, no. 20, p. 5074, 2023.

[6] H. Zhang, X. Yu, S. Ha, and T. Westerlund, "Lidar-generated images derived keypoints assisted point cloud registration scheme in odometry estimation", *Remote Sensing*, vol. 15, no. 20, 2023, ISSN: 2072-4292. DOI: 10.3390/

rs15205074. [Online]. Available: https://www.mdpi.com/2072-4292/15/20/5074.

[7] J. Shan and C. K. Toth, *Topographic laser ranging and scanning: principles and processing*. CRC press, 2018.

[8] R. H. Bewley, S. P. Crutchley, and C. A. Shell, "New light on an ancient landscape: Lidar survey in the stonehenge world heritage site", *Antiquity*, vol. 79, no. 305, pp. 636–647, 2005.

[9] C. Brenner, "Building reconstruction from images and laser scanning", *International Journal of Applied Earth Observation and Geoinformation*, vol. 6, no. 3-4, pp. 187–198, 2005.

[10] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, "Kiss-icp: In defense of point-to-point icp–simple, accurate, and robust registration if done the right way", *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 1029–1036, 2023.

[11] W. Xu and F. Zhang, "Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter", *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3317–3324, 2021.

[12] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry", *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

[13] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3d lidar inertial odometry and mapping", in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 3144–3150.

[14] H. Wang, C. Wang, C.-L. Chen, and L. Xie, "F-loam: Fast lidar odometry and mapping", in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2021, pp. 4390–4396.

[15] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi, "The farthest point strategy for progressive image sampling", *IEEE transactions on image processing*, vol. 6, no. 9, pp. 1305–1315, 1997.

[16] C. Moenning and N. A. Dodgson, "Fast marching farthest point sampling", University of Cambridge, Computer Laboratory, Tech. Rep., 2003.

[17] Q. C. Ruizhongtai, Y. Li, S. Hao, and G. L. J. PointNet+, "Deep hierarchical feature learning on point sets in a metric space", *Advances in Neural Information Processing Systems*, vol. 30, 2023.

[18] O. Dovrat, I. Lang, and S. Avidan, "Learning to sample", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2760–2769.

[19] X. Wang, Y. Jin, Y. Cen, C. Lang, and Y. Li, "Pst-net: Point cloud sampling via point-based transformer", in *Image and Graphics: 11th International Conference, ICIG 2021, Haikou, China, August 6–8, 2021, Proceedings, Part III 11*, Springer, 2021, pp. 57–69.

[20] C. Wu, J. Zheng, J. Pfrommer, and J. Beyerer, "Attention-based point cloud edge sampling", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5333–5343.

[21] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[22] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space", *Advances in neural information processing systems*, vol. 30, 2017.

[23] J. Li, B. M. Chen, and G. H. Lee, "So-net: Self-organizing network for point cloud analysis", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9397–9406.

[24] Y. Liu, B. Fan, S. Xiang, and C. Pan, "Relation-shape convolutional neural network for point cloud analysis", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8895–8904.

[25] C. R. Qi, O. Litany, K. He, and L. J. Guibas, "Deep hough voting for 3d object detection in point clouds", in *proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9277–9286.

[26] X. Pan, Z. Xia, S. Song, L. E. Li, and G. Huang, "3d object detection with pointformer", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 7463–7472.

[27] P. Besl and N. D. McKay, "A method for registration of 3-d shapes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992. DOI: 10.1109/34.121791.

[28] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, "KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way", *IEEE Robotics and Automation Letters (RA-L)*, vol. 8, no. 2, pp. 1029–1036, 2023. DOI: 10.1109/LRA.2023.3236571.

[29] P. Biber and W. Strasser, "The normal distributions transform: A new approach to laser scan matching", in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, vol. 3, 2003, 2743–2748 vol.3. DOI: 10.1109/IROS.2003.1249285.

[30] N. Mellado, D. Aiger, and N. J. Mitra, "Super 4pcs fast global pointcloud registration via smart indexing", *Computer Graphics Forum*, vol. 33, no. 5,

pp. 205–215, 2014, ISSN: 1467-8659. DOI: 10.1111/cgf.12446. [Online]. Available: http://dx.doi.org/10.1111/cgf.12446.

[31] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration", in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 3212–3217. DOI: 10.1109/ROBOT.2009.5152473.

[32] Z. J. Yew and G. H. Lee, "3dfeat-net: Weakly supervised local 3d features for point cloud registration", in *Computer Vision – ECCV 2018*. Springer International Publishing, 2018, pp. 630–646, ISBN: 9783030012670. DOI: 10.1007/978-3-030-01267-0_37. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-01267-0_37.

[33] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation", in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 77–85. DOI: 10.1109/CVPR.2017.16.

[34] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, *Dynamic graph cnn for learning on point clouds*, 2019. arXiv: 1801.07829 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1801.07829.

[35] Y. Li, L. Ma, Z. Zhong, *et al.*, "Deep learning for lidar point clouds in autonomous driving: A review", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 8, pp. 3412–3432, 2021. DOI: 10.1109/TNNLS.2020.3015992.

[36] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: A versatile and accurate monocular slam system", *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015, ISSN: 1941-0468. DOI: 10.1109/tro.2015.2463671. [Online]. Available: http://dx.doi.org/10.1109/TRO.2015.2463671.

[37] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time", in *Proceedings of Robotics: Science and Systems*, Berkeley, USA, Jul. 2014. DOI: 10.15607/RSS.2014.X.007.

[38] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 4758–4765.

[39] A. Tampuu, R. Aidla, J. A. van Gent, and T. Matiisen, "Lidar-as-camera for end-to-end driving", *Sensors*, vol. 23, no. 5, 2023, ISSN: 1424-8220. DOI: 10.3390/s23052845. [Online]. Available: https://www.mdpi.com/1424-8220/23/5/2845.

[40] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving", *CoRR*, vol. abs/1611.07759, 2016. arXiv: 1611.07759. [Online]. Available: http://arxiv.org/abs/1611.07759.

[41] B. Yang, W. Luo, and R. Urtasun, "PIXOR: real-time 3d object detection from point clouds", *CoRR*, vol. abs/1902.06326, 2019. arXiv: 1902.06326. [Online]. Available: http://arxiv.org/abs/1902.06326.

[42] D. G. Lowe, "Object recognition from local scale-invariant features", in *Proceedings of the seventh IEEE international conference on computer vision*, Ieee, vol. 2, 1999, pp. 1150–1157.

[43] D. G. Lowe, "Distinctive image features from scale-invariant keypoints", *International journal of computer vision*, vol. 60, pp. 91–110, 2004.

[44] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features", in *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*, Springer, 2006, pp. 404–417.

[45] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection", in *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*, Springer, 2006, pp. 430–443.

[46] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features", in *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, Springer, 2010, pp. 778–792.

[47] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf", in *2011 International conference on computer vision*, Ieee, 2011, pp. 2564–2571.

[48] D. DeTone, T. Malisiewicz, and A. Rabinovich, *Superpoint: Self-supervised interest point detection and description*, 2018. arXiv: 1712.07629 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1712.07629.

[49] X. Zhao, X. Wu, W. Chen, P. C. Y. Chen, Q. Xu, and Z. Li, "Aliked: A lighter keypoint and descriptor extraction network via deformable transformation", *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–16, 2023. DOI: 10.1109/TIM.2023.3271000. [Online]. Available: https://arxiv.org/pdf/2304.03608.pdf.

[50] X. Zhao, X. Wu, J. Miao, W. Chen, P. C. Y. Chen, and Z. Li, "Alike: Accurate and lightweight keypoint detection and descriptor extraction", *IEEE Transactions on Multimedia*, 2022. DOI: 10.1109/TMM.2022.3155927. [Online]. Available: http://arxiv.org/abs/2112.02906.

[51] J. Antic, *Deoldify–a deep learning based project for colorizing and restoring old images (and video!)*, 2019.

[52] X. Kang, T. Yang, W. Ouyang, P. Ren, L. Li, and X. Xie, "Ddcolor: Towards photo-realistic image colorization via dual decoders", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 328–338.

[53] G. Kim, K. Kang, S. Kim, *et al.*, "Bigcolor: Colorization using a generative color prior for natural images", in *European Conference on Computer Vision*, Springer, 2022, pp. 350–366.

[54] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis", in *International Conference on Learning Representations*, 2019. [Online]. Available: `https://openreview.net/forum?id=B1xsqj09Fm`.

[55] F. Luo, Y. Li, G. Zeng, P. Peng, G. Wang, and Y. Li, "Thermal infrared image colorization for nighttime driving scenes with top-down guided attention", *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[56] S. Li, B. Han, Z. Yu, C. H. Liu, K. Chen, and S. Wang, "I2v-gan: Unpaired infrared-to-visible video translation", in *ACMMM*, 2021.

[57] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization", in *ECCV*, 2016.

[58] R. Zhang, J.-Y. Zhu, P. Isola, *et al.*, "Real-time user-guided image colorization with learned deep priors", *ACM Transactions on Graphics (TOG)*, vol. 9, no. 4, 2017.

[59] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification", *ACM Transactions on Graphics (Proc. of SIGGRAPH 2016)*, vol. 35, no. 4, 2016.

[60] M. Xia, W. Hu, T.-T. Wong, and J. Wang, "Disentangled image colorization via global anchors", *ACM Transactions on Graphics (TOG)*, vol. 41, no. 6, 204:1–204:13, 2022.

[61] F. Baldassarre, D. G. Morín, and L. Rodés-Guirao, "Deep koalarization: Image colorization using cnns and inception-resnet-v2", *CoRR*, vol. abs/1712.03400, 2017. arXiv: 1712.03400. [Online]. Available: http://arxiv.org/abs/1712.03400.

[62] E. Wallner. "Coloring-greyscale-images". (), [Online]. Available: https://github.com/emilwallner/Coloring-greyscale-images?tab=readme-ov-file.

[63] P. Vitoria, L. Raad, and C. Ballester, "Chromagan: Adversarial picture colorization with semantic class distribution", in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 2445–2454.

[64] J.-W. Su, H.-K. Chu, and J.-B. Huang, "Instance-aware image colorization", in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[65] Y. Zhao, L.-M. Po, K.-W. Cheung, W.-Y. Yu, and Y. Abbas Ur Rehman, "Scgan: Saliency map-guided colorization with generative adversarial network", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 8, pp. 3062–3077, 2020.

[66] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks", *CVPR*, 2017.

[67] A. Brock, J. Donahue, and K. Simonyan, *Large scale gan training for high fidelity natural image synthesis*, 2019. arXiv: 1809.11096 [cs.LG].

[68] A. Brock, J. Donahue, and K. Simonyan, *Large scale gan training for high fidelity natural image synthesis*, 2019. arXiv: 1809.11096 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1809.11096.

[69] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks", *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2015.

[70] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1646–1654.

[71] C. Ledig, L. Theis, F. Huszár, *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.

[72] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[73] N. Ahn, B. Kang, and K.-A. Sohn, "Fast, accurate, and lightweight super-resolution with cascading residual network", *arXiv preprint arXiv:1803.08664*, 2018.

[74] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "Swinir: Image restoration using swin transformer", *arXiv preprint arXiv:2108.10257*, 2021.

[75] K. Zhang, Y. Li, J. Liang, *et al.*, "Practical blind image denoising via swin-conv-unet and data synthesis", *Machine Intelligence Research*, vol. 20, no. 6, pp. 822–836, 2023. DOI: 10.1007/s11633-023-1466-0. [Online]. Available: https://doi.org/10.1007/s11633-023-1466-0.

[76] X. Wang, K. Yu, S. Wu, *et al.*, "Esrgan: Enhanced super-resolution generative adversarial networks", in *The European Conference on Computer Vision Workshops (ECCVW)*, Sep. 2018.

[77] J. Yamanaka, S. Kuwashima, and T. Kurita, *Fast and accurate image super resolution by deep cnn with skip connection and network in network*, 2020. arXiv: `1707.05425 [cs.CV]`. [Online]. Available: `https://arxiv.org/abs/1707.05425`.

[78] Z. Chen, Y. Zhang, J. Gu, L. Kong, and X. Yang, "Recursive generalization transformer for image super-resolution", in *ICLR*, 2024.

[79] Z. Chen, Y. Zhang, J. Gu, Y. Zhang, L. Kong, and X. Yuan, "Cross aggregation transformer for image restoration", in *NeurIPS*, 2022.

[80] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution", in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, Springer, 2016, pp. 694–711.

[81] T. Rott Shaham, T. Dekel, and T. Michaeli, "Singan: Learning a generative model from a single natural image", in *Computer Vision (ICCV), IEEE International Conference on*, 2019.

[82] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising", *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[83] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte, "Plug-and-play image restoration with deep denoiser prior", *arXiv preprint*, 2020.

[84] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi, "Image super-resolution via iterative refinement", *IEEE Transactions on Pattern*

*Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4713–4726, 2023. DOI: 10.1109/TPAMI.2022.3204461.

[85] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jul. 2017.

[86] L. Qingqing, Y. Xianjia, J. P. Queralta, and T. Westerlund, "Multi-modal lidar dataset for benchmarking general-purpose localization and mapping algorithms", in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2022, pp. 3837–3844.