



Article

LiDAR-Generated Images Derived Keypoints Assisted Point Cloud Registration Scheme in Odometry Estimation

Haizhou Zhang ^{1,2,†} , Xianjia Yu ^{2,*,†} , Sier Ha ² and Tomi Westerlund ²

¹ School of Information Science and Technology, Fudan University, Shanghai 200433, China; 21210720289@m.fudan.edu.cn

² Turku Intelligent Embedded and Robotic Systems Laboratory, Faculty of Technology, University of Turku, 20014 Turku, Finland; sierha@utu.fi (S.H.); tovewe@utu.fi (T.W.)

* Correspondence: xianjia.yu@utu.fi

† These authors contributed equally to this work.

Abstract: Keypoint detection and description play a pivotal role in various robotics and autonomous applications, including Visual Odometry (VO), visual navigation, and Simultaneous Localization And Mapping (SLAM). While a myriad of keypoint detectors and descriptors have been extensively studied in conventional camera images, the effectiveness of these techniques in the context of LiDAR-generated images, i.e., reflectivity and ranges images, has not been assessed. These images have gained attention due to their resilience in adverse conditions, such as rain or fog. Additionally, they contain significant textural information that supplements the geometric information provided by LiDAR point clouds in the point cloud registration phase, especially when reliant solely on LiDAR sensors. This addresses the challenge of drift encountered in LiDAR Odometry (LO) within geometrically identical scenarios or where not all the raw point cloud is informative and may even be misleading. This paper aims to analyze the applicability of conventional image keypoint extractors and descriptors on LiDAR-generated images via a comprehensive quantitative investigation. Moreover, we propose a novel approach to enhance the robustness and reliability of LO. After extracting keypoints, we proceed to downsample the point cloud, subsequently integrating it into the point cloud registration phase for the purpose of odometry estimation. Our experiment demonstrates that the proposed approach has comparable accuracy but reduced computational overhead, higher odometry publishing rate, and even superior performance in scenarios prone to drift by using the raw point cloud. This, in turn, lays a foundation for subsequent investigations into the integration of LiDAR-generated images with LO.

Keywords: LiDAR; LiDAR-generated images; keypoint detector and descriptor; point cloud registration; LiDAR odometry



Citation: Zhang, H.; Yu, X.; Ha, S.; Westerlund, T. LiDAR-Generated Images Derived Keypoints Assisted Point Cloud Registration Scheme in Odometry Estimation. *Remote Sens.* **2023**, *15*, 5074. <https://doi.org/10.3390/rs15205074>

Academic Editor: Filiberto Chiabrando

Received: 19 September 2023

Revised: 16 October 2023

Accepted: 18 October 2023

Published: 23 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

LiDAR technology has become a primary sensor for facilitating advanced situational awareness in the domains of robotics and autonomous systems ranging from LiDAR Odometry (LO), Simultaneous Localization And Mapping (SLAM), object detection and tracking, and navigation. Among these applications, LO, as a fundamental component in robotics, has significantly drawn our attention. Extensive research efforts have focused on the integration of diverse sensors, including Inertial Measurement Units (IMUs), to bolster LO performance. However, in scenarios where LiDAR data lacks geometric distinctness or even contains misleading information, the process of point cloud registration continues to present challenges in achieving precise estimations and even causing drift in certain cases (Figure 1a).

Notably, recent years have witnessed substantial progress in LiDAR technology, marked by the emergence of numerous high-resolution spinning and solid-state LiDAR

devices offering various modalities of sensor data [1,2]. The increased density of the point cloud brings a challenge for point cloud registration with a significant computation overhead, especially for devices with limited computational resources.

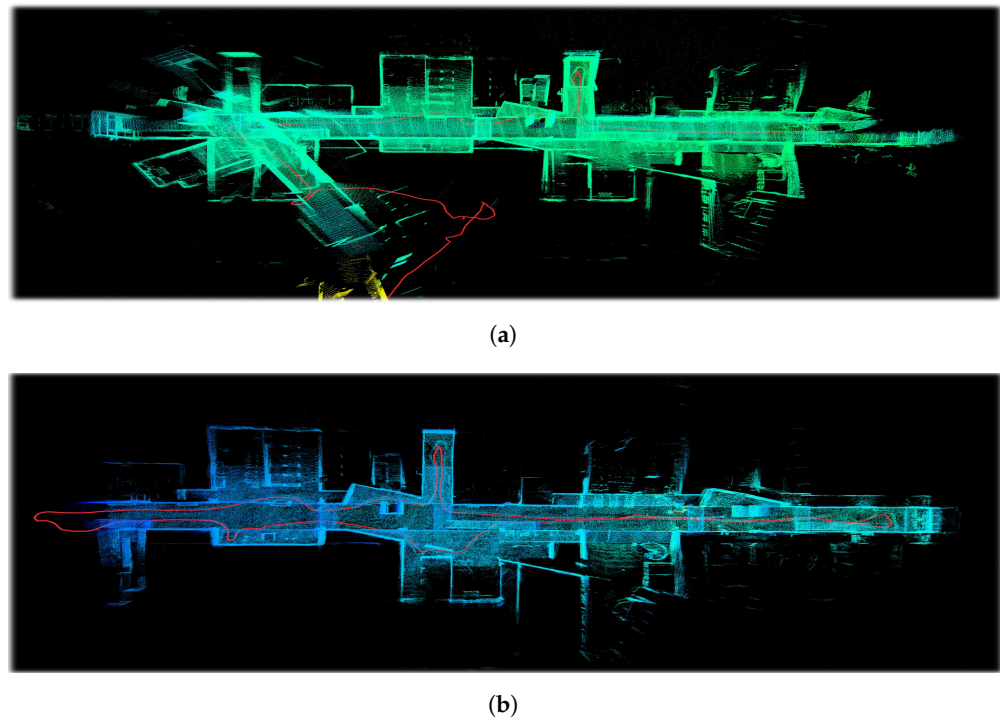


Figure 1. Samples of LiDAR odometry results run in our experiment. (a) Raw point cloud with point cloud matching approach (KISS-ICP), a drift happened after a certain period. (b) Our proposed LiDAR-generated keypoint extraction-based approach.

Within the aforementioned modalities, LiDAR-generated images, including reflectivity images, range images, and near-infrared images, have introduced the potential to apply conventional camera image processing techniques to LiDAR-generated images. These images are low-resolution but possibly panoramic and exhibit heightened resilience and robustness in challenging environments, such as those characterized by fog and rain, compared to conventional camera images. Additionally, these images can potentially provide crucial information for point cloud registration when there is a deficiency of geometric data, or the raw point cloud lacks useful information so as to avoid drift (Figure 1b).

Keypoint detectors and descriptors have found extensive utility across diverse domains within visual tasks such as place recognition, scene reconstruction, Visual Odometry (VO), Visual Simultaneous Localization And Mapping (VSLAM), and Visual Inertial Odometry (VIO). Nevertheless, there remains a lack of investigation into the performance of extant keypoint detectors and descriptors when applied to LiDAR-generated imagery.

Contemporary methodologies for Visual Odometry (VO) or Visual Inertial Odometry (VIO) rely significantly on the operability of visual sensors, necessitating knowledge of camera intrinsics to facilitate Structure from Motion (SfM)—a requisite not met by LiDAR-generated images. This poses the difficulty of extracting keypoints from LiDAR-generated images in a certain way to further apply them in the odometry estimation.

In summary, the extant LiDAR-based point cloud registration paradigm confronts notable challenges, principally arising from the presence of drifts or misalignments engendered by the inherent density of the point cloud, coupled with the substantial computational overhead it imposes. Given the escalating ubiquity of LiDAR-derived imagery in contemporary contexts, there exists a propitious potential to leverage these data modalities for the amelioration of these prevailing challenges.

Therefore, to address the above issues, in this study:

- (i) We investigate the efficacy of the existing keypoint detectors and descriptors on LiDAR-generated images with multiple specialized metrics providing a quantitative evaluation.
- (ii) We conduct an extensive study of the optimal resolution and interpolation approaches for enhancing the low-resolution LiDAR-generated data to extract keypoints more effectively.
- (iii) We propose a novel approach that leverages the detected keypoints and their neighbors to extract a reliable point cloud (downsampling) for the purpose of point cloud registration with reduced computational overhead and fewer deficiencies in valuable point acquisition.

The structure of this paper is as follows. In Section 2, we survey the recent progress on keypoint detectors and descriptors, including approaches and metrics, point cloud matching, and the status of LO for the support of the selected methods utilized in the following sections. Section 3 provides an overview of the quantitative evaluation of the existing keypoint detectors and descriptors. Based on the above analysis, we propose our LiDAR-generated image keypoint-assisted point cloud registration and describe the details of the approach in this section. Section 4 demonstrates the experimental results in detail. In the end, we conclude the work and sketch out some future research directions in Section 5.

2. Related Work

In this section, we commence by presenting a comprehensive review of the prevailing detector and descriptor algorithms documented in the literature. Subsequently, a brief summary of the current advancements in the domain of LiDAR-imaged techniques is offered. We conclude with a concise analysis of the leading algorithms for point cloud registration in LO for the purpose of justifying the selected methods.

2.1. Keypoint Detector and Descriptor

In recent years, there have been multiple widely applied detectors and descriptors in the field of computer vision. As illustrated in Table 1, we have captured the essential characteristics of different detectors and descriptors. And Table 2 includes more about the explanation of the pros and cons of the different detectors and descriptors.

Harris detector [3] can be seen as an enhanced version of Moravec's corner detector [4,5]. It is used to identify corners in an image, which are the regions with large intensity variations in multiple directions. The Shi-Tomasi Corner Detector [6] is an improvement upon the Harris Detector with a slight modification in the corner response function that makes it more robust and reliable in certain scenarios. The Features from Accelerated Segment Test (FAST) [7] algorithm operates by examining a circle of pixels surrounding a candidate pixel and testing for a contiguous segment of pixels that are either significantly brighter or darker than the central pixel.

For descriptor-only algorithms, Binary Robust Independent Elementary Features (BRIEF) [8] utilizes a set of binary tests on pairs of pixels within a patch surrounding one keypoint. Fast Retina Keypoint (FREAK) [9] is inspired by the human visual system, which constructs a retinal sampling pattern that is more densely sampled towards the center and sparser towards the periphery. Then, it compares pairs of pixels within this pattern to generate a robust binary descriptor.

With respect to the combined detector–descriptor algorithms, the Scale-Invariant Feature Transform (SIFT) [10,11] detects keypoints by identifying local extrema in the difference of Gaussian scale–space pyramid, then computes a gradient-based descriptor for each keypoint. Speeded-Up Robust Features (SURF) [12] is designed to address the computational complexity of SIFT while maintaining robustness to various transformations. Binary Robust Invariant Scalable Keypoints (BRISK) [13] uses a scale–space FAST [7] detector to identify keypoints and computes binary descriptors based on a sampling pattern of concentric circles. Oriented FAST and Rotated BRIEF (ORB) [14] extends the

FAST detector with a multi-scale pyramid and computes a rotation-invariant version of the BRIEF [8] descriptor, aiming to provide a fast and robust alternative to SIFT and SURF. Accelerated-KAZE (AKAZE) [15] employs a fast explicit diffusion scheme to accelerate the detection process and computes a Modified Local Difference Binary (M-LDB) descriptor [16] for robust matching.

Table 1. Keypoint detectors and descriptors.

Method	Detector	Descriptor	Description
Harris	✓		Corner detection method focusing on local image variations.
Shi-Tomasi	✓		Variation of Harris with modification in the response function to be more robust.
FAST	✓		Efficient corner detection for real-time applications.
FREAK		✓	Robust to transformations, based on human retina's structure.
BRIEF		✓	Efficient short binary descriptor for keypoints.
SIFT	✓	✓	Invariant to scale, orientation, and partial illumination changes.
SURF	✓	✓	Addresses the computational complexity of SIFT while maintaining robustness.
BRISK	✓	✓	Faster binary descriptor method, efficient compared to SIFT/SURF.
ORB	✓	✓	Combines FAST detection and BRIEF descriptor, commonly used now.
AKAZE	✓	✓	Builds on KAZE but faster, good for wide baseline stereo correspondence.
Superpoint	✓	✓	A state-of-the-art AI approach that exhibits superior performance when applied to traditional camera images.

Table 2. Pros and cons of keypoint detectors and descriptors.

Method	Pros	Cons
Harris	Simple, effective for corner detection.	Not invariant to scale or rotation.
Shi-Tomasi	Improvement over Harris, more robust.	Still not invariant to scale or rotation.
FAST	Very efficient, suitable for real-time applications.	Not robust to scale and rotation changes.
FREAK	Robust to transformations, inspired by human vision.	Only a descriptor, requires a separate detector.
BRIEF	Efficient, compact binary descriptor.	Not rotation or scale invariant.
SIFT	Scale and rotation invariant, robust to illumination changes.	Computationally expensive.

Table 2. Cont.

Method	Pros	Cons
SURF	Faster than SIFT, also scale and rotation invariant.	Still more computationally intensive than binary descriptors.
BRISK	Efficient binary descriptor, faster than SIFT/SURF.	Not as robust to transformations as SIFT/SURF.
ORB	Combines the efficiency of FAST and BRIEF, rotation invariant.	Not scale invariant.
AKAZE	Faster than KAZE, good for stereo correspondence.	Less popular, not as extensively studied as others.
Superpoint	Superior performance, state-of-the-art with AI benefits.	Requires deep learning infrastructure, more computational overhead.

The emergence of deep learning (DL) techniques, particularly convolutional neural networks (CNN) [17,18], has revolutionized computer vision over the last decade. SuperPoint [19] detector employs a full CNN to predict a set of keypoint heatmaps, where each heatmap corresponds to an interest point's probability at a given pixel location. Then, the descriptor part generates a dense descriptor map for the input image by predicting a descriptor vector at each pixel location.

To sum up, while numerous detector and descriptor algorithms have gained popularity, it is imperative to note that they have primarily been designed for traditional camera images, not LiDAR-based images. Consequently, it is of paramount importance for this study to identify the algorithms that maintain efficacy for LiDAR-based images.

2.2. LiDAR-Generated Images in Robotics

Within the realm of robotics, some studies over the years have delved into the utilization of LiDAR-based images. But before exploring specific applications, it is vital to know the process by which range images and signal images are generated from the point cloud, as detailed in [20,21]. And it is also essential to understand the effectiveness of LiDAR-based images, through an extensive evaluation in the article [22], showing that LiDAR-based images have remarkable resilience to seasonal and environmental variations.

Perception emerges as the indisputable first step in the use of LiDAR within robotics. In [23], Ouster introduced their work to explain the possibility of using LiDAR as a camera. They demonstrate the effectiveness of car and road segmentation by putting the LiDAR-based image into a pre-trained DL model. In the work [24], Tsiourva et al. proposed a saliency detection model based on LiDAR-generated images. In the model, the attributes of reflectivity, intensity, range, and ambient images are carefully contrasted and analyzed. After several advanced image processing steps, multiple conspicuity maps are created. These maps help make a unified saliency map, which identifies and emphasizes the most distinct objects in the image. In [25], Sier et al. explored using LiDAR-as-a-camera sensors to track Unmanned Aerial Vehicles (UAVs) in GNSS-denied environments, fusing LiDAR-generated images and point clouds for real-time accuracy. In the study conducted by Lacopo et al. [26], images were synthesized utilizing reflectivity and depth data derived from solid-state LiDAR, serving as an initialization procedure for UAV tracking tasks. The work [27] explores the potential of general-purpose deep learning perception algorithms, specifically detection and segmentation neural networks, based on LiDAR-generated images. The study provides both a qualitative and quantitative analysis of the performance of a variety of neural network architectures, proving that the DL models built for visual camera images also offer significant advantages when applied to LiDAR-generated images.

Delving deeper into subsequent applications, for example, localization, the research in [28] explores the problem of localizing mobile robots and autonomous vehicles within a large-scale outdoor environment map, by leveraging range images produced by 3D LiDAR.

2.3. Evaluation Metrics for Keypoint Detectors and Descriptors

The efficacy of detector and descriptor algorithms is typically assessed through some specific evaluation metrics. As illustrated in Table 3, the first three metrics, number of keypoints, computational efficiency, and robustness of detector are straightforward to comprehend and implement, and also widely adopted in numerous studies [12,29,30]. For instance, the robustness of the detector [31] is implemented by contrasting keypoints before and after the transformations like *scaling*, *rotation*, and *Gaussian noise interference*.

Table 3. Metrics for evaluating keypoint detectors and descriptors.

Metrics	Description
Number of keypoints	A high number of keypoints can always lead to more detailed image analysis and better performance in subsequent tasks like object recognition.
Computational efficiency	Computational efficiency remains paramount in any computer vision algorithms. We gauge this efficiency by timing the complete detection, description, and matching process.
Robustness of detector	An efficacious detector should recognize identical keypoints under varying conditions such as scale, rotation, and Gaussian noise interference.
Match ratio	The ratio of successfully matched points to the total number of detected points offers insights into the algorithm's capability in identifying and relating unique keypoints.
Match Score	A homography matrix is estimated from two point sets, to distinguish spurious matches, then the algorithm precision is quantified by the inlier ratio.
Distinctiveness	Distinctiveness entails that the keypoints isolated by a detection algorithm should exhibit sufficient uniqueness for differentiation among various keypoints.

When assessing the precision of the entire algorithmic procedure, which is prioritized by the majority of tasks, the prevalent metrics often necessitate benchmark datasets, such as KITTI [32] or HPatches [33]. These datasets either provide the transformation matrix between images or directly contain the keypoint ground truth. For example, in Mukherjee et al.'s study [34], one crucial metric, "Precision", is defined as *correct matches/all detected matches*, where correct matches are ascertained through the geometric verification based on a known camera position provided by dataset [35]. Similarly, in another recent work [36], the evaluation tasks, including "keypoint verification", "image matching", and "keypoint retrieval", all rely on the homography matrix between images in the benchmark dataset [33].

Nevertheless, given that research predicated on LiDAR images is at a nascent stage, no benchmark dataset exists in the field of LiDAR-based images. And the effort required for data labeling [37,38] to produce such a dataset is considerable and challenging. To bridge this gap, we select multiple key evaluation metrics: match ratio, match score, and distinctiveness, as shown in Table 3 from previous studies. Match ratio [34] is quantitatively defined as the *number of matches/number of keypoints*. A high match ratio can suggest that the algorithm is adept at identifying and correlating distinct features; while the exact homography matrix between images remains unknown when lacking benchmark datasets, it can be approximated using mathematical methodologies from two point sets. This computed homography can subsequently be utilized to find correct matches. The *number*

of estimated correct matches/number of matches is denoted as match score in our work. And distinctiveness is computed as follows: For every image, the k-nearest neighbors algorithm, with $k = 2$, is employed to identify the two best matches [10]. If the descriptor distance of the primary match is notably lower than that of the secondary match, it demonstrates the algorithm's competence in recognizing and describing highly distinctive keypoints. Consequently, this defines the metric distinctiveness.

2.4. 3D Point Cloud Downsampling

Point cloud downsampling is crucial in operating LO or SLAM within a computation-constrained device. Nowadays, there is a substantial body of work focusing on the employment of DL networks, for example, a lightweight transformer [39]. Other approaches utilized various filters in order to achieve not only point cloud downsampling but also denoising [40].

2.5. 3D Point Cloud Matching in LO

LO has been widely studied, yet is challenging due to the complexity of the environment in the robotic field. Contemporary research endeavors have witnessed a notable surge in efforts integrating supplementary sensors, such as Inertial Measurement Units (IMUs), aimed at augmenting the precision and resilience of LO. However, as we focus on the point cloud registration phase of LO, this is out of the scope of the related work of this part. We primarily discuss the solely LiDAR-based LO. Among these solely LiDAR-based approaches, LOAM [41], as a popular matching-based SLAM and LO approach, has encouraged a significant amount of other LO approaches, including Lego-LOAM [42] and F-LOAM [43].

Point cloud matching or registration constitutes the key component in LO. Since its inception approximately three decades ago, the Iterative Closest Point (ICP) algorithm, introduced by Besl and McKay [44], has spawned numerous variants. These include notable adaptations such as Voxelized Generalized ICP (GICP) [45], CT-ICP [46], and KISS-ICP [47]. Among these ICP iterations, KISS-ICP, denoting “keep it small and simple”, distinguishes itself by providing a point-to-point ICP approach characterized by robustness and accuracy in pose estimation. Furthermore, the Normal Distributions Transform (NDT) [48] represents another prominent point cloud registration technique frequently employed in LO research. As the latest ICP approach, KISS-ICP is the designated methodology for the point cloud registration we adopted in this study.

3. Methodology

In this section, we first introduce the dataset we used in the paper, including the detailed specifications of the sensors, data modality, and the data sequences. Then, we describe our experimental procedure in detail including the optimal pre-processing configuration for the keypoint detectors and the workflow of our proposed LiDAR-generated image keypoints assisted the point cloud registration approach.

3.1. Dataset

For the evaluation of keypoint detectors and descriptors and our proposed approach, we utilized the published open-source dataset for multi-modal LiDAR sensing [1]. The dataset is available and accessible via the Github repos ((<https://github.com/TIERS/tiers-lidars-dataset-enhanced>, accessed on 18 October 2023); <https://github.com/TIERS/tiers-lidars-dataset>, accessed on 18 October 2023). The dataset consists of various LiDARs and among them, Ouster LiDAR provides not only point cloud but also its generated images. The Ouster LiDAR applied in the dataset is OS0-128 with its detailed specifications shown in Table 4.

Table 4. Specifications of Ouster OS0-128.

	IMU	Type	Channels	Image Resolution	FoV	Angular Resolution	Range	Freq	Points
Ouster OS0-128	ICM-20948	spinning	128	2048 × 128	360° × 90°	V : 0.7°, H : 0.18°	50 m	10 Hz	2,621,440 pts/s

The images generated by OS0-128 shown in Figure 2 include signal images, reflectivity images, near-infrared images, and range images with its expansive $360^\circ \times 90^\circ$ field of view. Signal images are representations of the signal strength of the light returned to the sensor for a given point, which depends on various factors, such as the angle of incidence, the distance from the sensor, and the material properties of the object. In near-infrared images, each pixel's intensity is represented by the amount of detected photons that are not emitted by the sensor's own laser pulse but may come from sources such as sunlight or moonlight. And every pixel in a reflectivity image represents the calculated calibrated reflectivity. Then, range images demonstrate the distance from the sensor to objects in the environment.

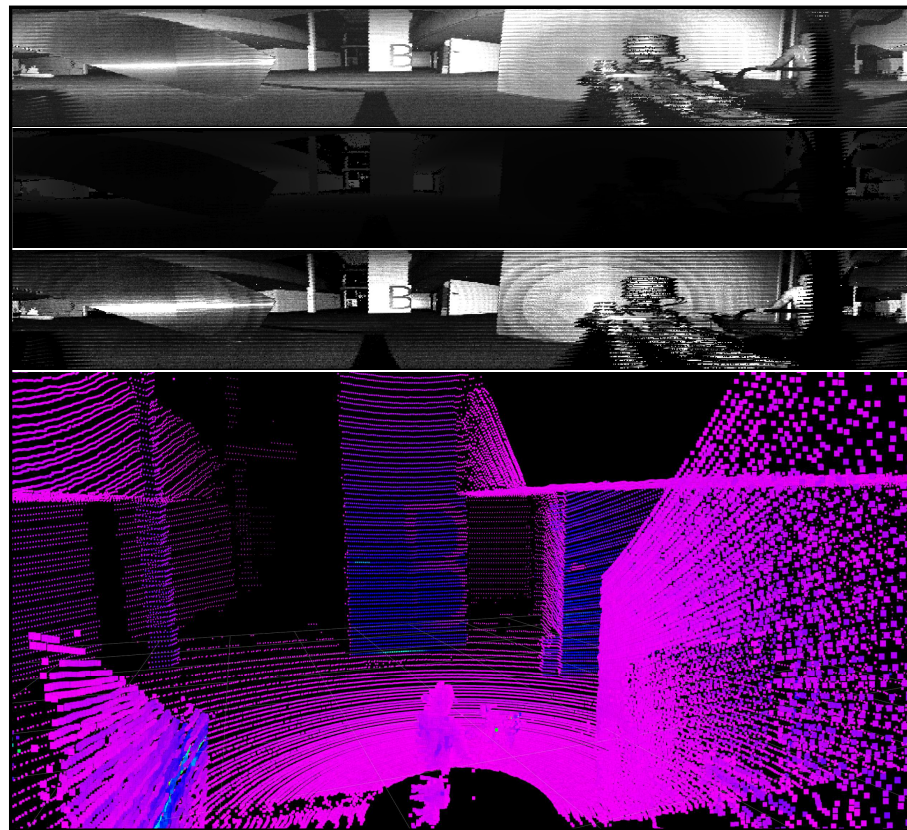


Figure 2. Samples of LiDAR-generated images, from above to bottom, are signal image, range image, reflectivity image, and point cloud.

As indicated by the findings of our previous research, signal images have exhibited superior performance in the execution of conventional DL tasks within the domain of computer vision [27]. In light of this, for the first two parts of our experiment, we opted to employ signal images from the “indoor_01_square” scene provided by the dataset, which is a scene that spans 114 s and comprises 1146 image messages.

3.2. Optimal Pre-Processing Configuration Searching for LiDAR-Generated Images

LiDAR-generated images at hand are typically panoramic but low-resolution. Moreover, these images often exhibit a substantial degree of noise. This prompts a concern of utilizing the original images for facilitating the functionality evaluation of the keypoint

detector and descriptor algorithms. And our preliminary experiments have evinced unsatisfactory performance across an array of detectors and descriptors when employing the unaltered original LiDAR-generated images. To identify the optimal resolution and interpolation methodology for augmenting image resolution, an extensive comparative experiment was conducted.

In this part, we implement an array of interpolation techniques on the original images, employing an extensive spectrum of image resolution combinations. The interpolation methodologies encompass bicubic interpolation (CUBIC), Lanczos interpolation over 8×8 neighborhood (LANCZOS4), resampling using pixel area relation (AREA), nearest neighbor interpolation (NEAREST), and bilinear interpolation (LINEAR). The primary procedure of the pre-processing is elucidated in Algorithm 1.

Algorithm 1: Preprocessing configuration evaluation

Input:

N number of signal images: $\{S_i\}, i \sim N$;

Interpolation methods: $IA = \{CUBIC, LANCZOS4, AREA, NEAREST, LINEAR\}$;

Targeted Width: $TW = \{min : 512; max : 4096; step : 128\}$;

Targeted Height: $TH = \{min : 32; max : 256; step : 32\}$;

Detectors and descriptors:

$DET = \{SURF, SIFT, SHITOMASI, HARRIS, BRISK, FAST, AKAZE, ORB\}$;

$DES = \{FREAK, SIFT, BRISK, SURF, BRIEF, AKAZE, ORB\}$;

Output: Metrics

foreach *interplation approach in IA* **do**

foreach *width in TW* **do**

foreach *height in TH* **do**

foreach *det in DET and des in DES* **do**

foreach S_i **do**

 Calculate the value of aforementioned metrics;

 Save the calculated value;

Analyze the metric values.

More specifically, we iterate a range of image dimensions and interpolation methods in conjunction with the suite of detector and descriptor algorithms designated for evaluation. Each iteration involves a rigorous evaluation of a comprehensive metrics set detailed in Table 3. Following a quantitative analysis, we compute mean values for these metrics. This extensive assessment aims to identify the optimal pre-processing configuration that offers balanced performance for different keypoint detectors and descriptors.

3.3. Keypoint Detectors and Descriptors for LiDAR-Generated Images

The evaluation workflow of detector–descriptor algorithms typically comprises three stages, including feature extraction, keypoint description, and keypoint matching between successive image frames. In this section, the specific procedures for executing these stages in our experimental setup will be elaborated upon.

3.3.1. Designated Keypoint Detector and Descriptor

An extensive array of keypoint detectors and descriptors, as detailed in Table 1 from Section 2.1, were investigated. The employed keypoint detectors include *SHITOMASI*, *HARRIS*, *FAST*, *BRISK*, *SIFT*, *SURF*, *AKAZE*, and *ORB*. Additionally, we integrated Superpoint, a DL-based keypoint detector, into our methodology. The keypoint descriptors implemented in our experiment are *BRISK*, *SIFT*, *SURF*, *BRIEF*, *FREAK*, *AKAZE*, and *ORB*.

3.3.2. Keypoints Matching between Images

Keypoint matching, the final stage of the detector–descriptor workflow, focuses on correlating keypoints between two images, which is essential for establishing spatial

relationships and forming a coherent scene understanding. The smaller the distance of the descriptors between two points, the more likely it is that they are the same point or object between two images. In our implementation, we employ a technique termed “brute-force match with cross-check”, which means for a given descriptor \mathcal{D}_A in image A and another descriptor \mathcal{D}_B in image B , a valid correspondence requires that both descriptors recognize each other as their closest descriptors.

3.3.3. Selected Evaluation Metrics

As explained in Section 2.3, we have opted not to rely on ground truth-based evaluation methodologies due to the lack of benchmark datasets and the substantial labor involved in data labeling. Instead, we combined some specially designed metrics that are independent of ground truth, together with several intuitive metrics, to form the complete indicators listed in Table 3. To our best understanding, this represents the most extensive set of evaluation metrics currently available in the absence of a benchmark dataset.

3.3.4. Evaluation Process

The flowchart shown in Algorithm 2 below provides an outline of the steps carried out by the program. Two nested loops are employed to iterate over different detector–descriptor pairs. For each image, the algorithm detects and describes its keypoints. If more than one image has been processed, keypoints from the current image are matched to the previous one. And metrics are placed in corresponding positions to assess the algorithm’s performance.

Algorithm 2: Overall evaluation pipeline of keypoint detectors and descriptors

Input:
 N number of signal images: $\{S^i\}, i \sim N$;
 $DET = \{SURF, SIFT, SHITOMASI, HARRIS, BRISK, FAST, AKAZE, ORB\}$;
 $DES = \{FREAK, SIFT, BRISK, SURF, BRIEF, AKAZE, ORB\}$;
Output: Metrics: Number of keypoints, Robustness of Detector, Computational Efficiency, Match Ratio, Match Score, Distinctiveness

```

foreach  $Detector \in DET$  do
  foreach  $Descriptor \in DES$  do
    foreach  $S^i$  do
       $S^i \leftarrow Preprocess(S^i)$ ;
       $\mathcal{KP}^i, \mathcal{D}^i \leftarrow detect\_and\_compute(S^i, Detector, Descriptor)$ ;
      Running time of detection and descriptor :  $\mathcal{T}_1^i$ ;
      Number of Keypoints :  $\mathcal{N}_{kp}^i$ ;
      Apply different transformations to image  $S_i$ , then calculate robustness:  $\mathcal{R}_{rot}^i, \mathcal{R}_{scale}^i, \mathcal{R}_{blur}^i$ ;
      if  $i > 1$  then
         $\mathcal{V}_{match}^i \leftarrow Match(\mathcal{D}^i, \mathcal{D}^{i-1})$ ;
        Running time of matching :  $\mathcal{T}_2^i$ ;
        Computational Efficiency :  $\mathcal{T}_1^i + \mathcal{T}_2^i$ ;
        if  $\mathcal{N}_{match}^i > 0$  then
          Match Ratio =  $\mathcal{N}_{match}^i / \mathcal{N}_{kp}^i$ ;
           $\mathcal{KP}_{Ref}^i \leftarrow find\_matched\_points(\mathcal{KP}^i, \mathcal{V}_{match}^i)$ ;
           $\mathcal{KP}_{Source}^i \leftarrow find\_matched\_points(\mathcal{KP}^{i-1}, \mathcal{V}_{match}^i)$ ;
           $\mathcal{H}_{Estimated\_homo}^i \leftarrow estimate\_homo\_matrix(\mathcal{KP}_{Source}^i, \mathcal{KP}_{Ref}^i)$ ;
          foreach  $\mathcal{P}_{Ref} \in \mathcal{KP}_{Ref}^i, \mathcal{P}_{Source} \in \mathcal{KP}_{Source}^i$  do
            if  $\|\mathcal{P}_{Source} - \mathcal{P}_{Ref} * \mathcal{H}_{Estimated\_homo}^i\| < 3$  then
               $\mathcal{N}_{good\_match}^i = +1$ ;
          Match Score =  $\mathcal{N}_{good\_match}^i / \mathcal{N}_{match}^i$ ;
        else
          Match Ratio = 0;
          Match Score = 0;
         $\mathcal{V}_{Primary\_match}^i, \mathcal{V}_{Secondary\_match}^i \leftarrow KnnMatch(\mathcal{D}^i, \mathcal{D}^{i-1}, k = 2)$ ;
        foreach  $Match1 \in \mathcal{V}_{Primary\_match}^i, Match2 \in \mathcal{V}_{Secondary\_match}^i$  do
          if  $Distance_{match1} / Distance_{match2} < 0.8$  then
             $\mathcal{N}_{Distinctive\_match}^i = +1$ ;
        Distinctiveness =  $\mathcal{N}_{Distinctive\_match}^i / \mathcal{N}_{kp}^i$ ;

```

Analyze the metric values.

3.4. LiDAR-Generated Image Keypoints Assisted Point Cloud Registration

3.4.1. Selected Data

The selected data for the evaluation from the dataset mentioned in Section 3.1 includes indoor and outdoor environments. The outdoor environment is from the normal road, denoted as “Open road”, and a forest, denoted as “Forest”. The indoor data include a hall in a building, denoted as “Hall (large)”, and two rooms, denoted as “Lab space (hard)”, and “Lab space (easy)”.

3.4.2. Point Cloud Matching Approach

In this part, we applied KISS-ICP (<https://github.com/PRBonn/kiss-icp.git>, accessed on 18 October 2023) as our point cloud matching approach. It also provides the odometry information, affording us the means to assess the efficacy of our point cloud downsampling approach through an examination of a positioning error, namely translation error and rotation error. To generalize our proposed approach, we tested an NDT-based simple SLAM program (https://github.com/Kin-Zhang/simple_ndt_slam.git, accessed on 18 October 2023) as well.

3.4.3. Proposed Method for Point Cloud Downsampling

Following the pre-processing of LiDAR-generated images outlined in Section 3.3, we derive optimal configurations for the keypoint detectors and descriptors. Utilizing these configurations as a foundation, we establish the workflow of our proposed methodology, illustrated in Figure 3. Within this process, we conduct distinct pre-processing procedures for both the range and signal images, employing them individually for keypoint detection and descriptor extraction. Subsequently, we combine the keypoints obtained from both images and search the K nearest points to each of these keypoints. We systematically varied K within the range of 3 to 7, adhering to a maximum threshold of 7 to align with our primary objective of downsampling the point cloud. Consequently, we find the corresponding point cloud of the keypoints and their neighbors within the raw point cloud, thereby constituting the downsampled point cloud. And then, we feed the downsampled point cloud into the point cloud registration for the odometry-generating purpose.

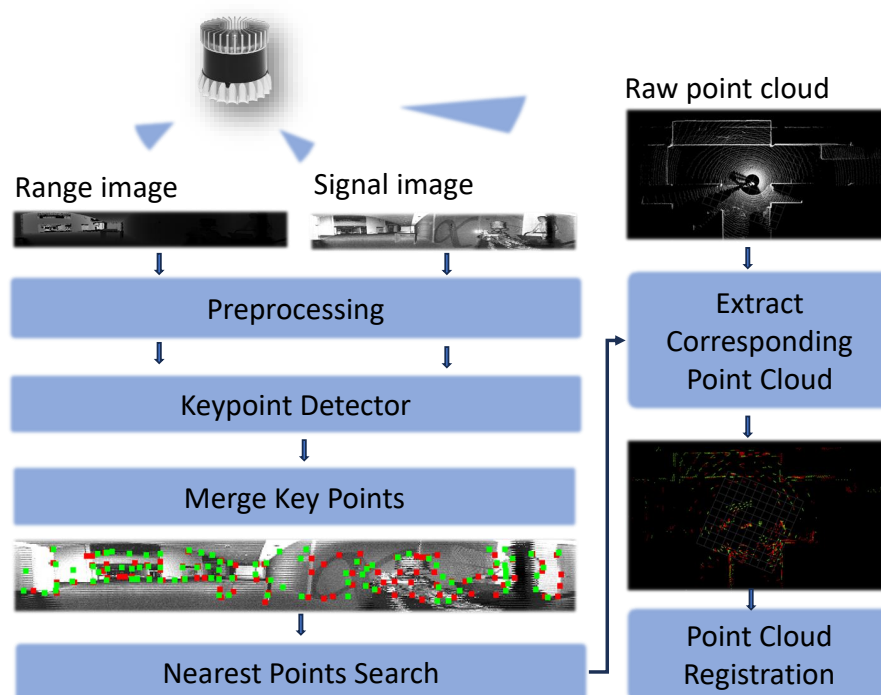


Figure 3. The process of the proposed LiDAR-generated images assisting point cloud registration.

In our analysis, we examined not only the positional error but also the rotational error, computational resource utilization, downsampling-induced alterations in point cloud density, and the publishing rate of LO.

3.5. Hardware and Software Information

Our experiments are run on the ROS Noetic on the Ubuntu 20.04 system. The platform is equipped with an i7 8-core 1.6 GHz CPU and an Nvidia GeForce MX150 graphics card. Primarily, we used libraries like OpenCV and PCL. Note that we have used some non-free copyright-protected algorithms from OpenCV, such as SURF, just for research.

The assessment of keypoint-based point cloud downsampling was conducted on a Lenovo Legion notebook equipped with the following specifications: 16 GB RAM, a 6-core Intel i5-9300H processor (2.40 GHz), and an Nvidia GTX 1660Ti graphics card (boasting 1536 CUDA cores and 6 GB VRAM). Within this study, our primary focus was on the evaluation of the two open-source algorithms delineated in Section 3.4.2, namely, KISS-ICP and Simple-NDT-SLAM. It is imperative to highlight that a consistent voxel size of 0.2 m was employed for both algorithms. Our project is primarily written in C++ (including the DL approach, Superpoint), publicly available in GitHub (<https://github.com/TIERS/ws-lidar-as-camera-odom>, accessed on 18 October 2023).

4. Experiment Result

Through this section, we first cover the final results of our exploration of the preprocessing workflow of LiDAR-based images. Subsequently, an in-depth analysis of keypoint detectors and descriptors for LiDAR-based images is conducted. Then, a detailed quantitative assessment of the performance of LO facilitated by LiDAR-generated image keypoints is presented.

4.1. Results of Preprocessing Methods for LiDAR-Generated Image

As elucidated in Section 2.3, distinctiveness and match score are considered as paramount measures for the overall accuracy of the entire algorithm pipeline. Consequently, in scenarios where different sizes and interpolation methods show peak performance on different metrics, these two metrics are our primary concern. Based on such criteria, the size 1024×64 demonstrated better performance across all detector and descriptor methods. Then, in Table 5, our evaluation also revealed that the **linear** interpolation method yielded the most optimal results among the various interpolation techniques.

Table 5. Evaluation metrics under different interpolation approaches.

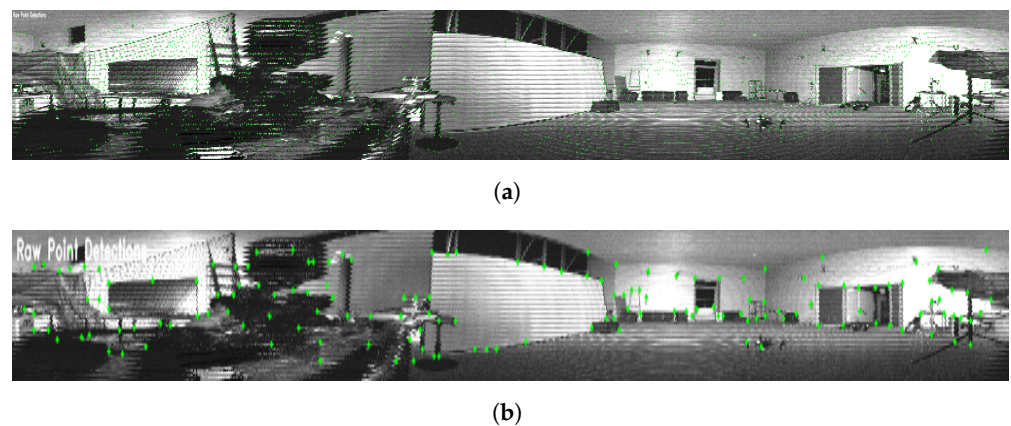
Interpolation	Robustness of (Rotation, Scaling, Noise)	Distinctiveness	Matching Score
AREA	(0.81, 0.106, 0.574)	0.309	0.415
CUBIC	(0.82 , 0.121, 0.569)	0.292	0.408
LANCZOS4	(0.819, 0.127 , 0.559)	0.286	0.405
NEAREST	(0.818, 0.128, 0.573)	0.275	0.401
LINEAR	(0.815, 0.1, 0.583)	0.314	0.415

The findings in Table 6 also suggest that there is a clear advantage in properly reducing the size of an image as opposed to enlarging it. Additionally, in the process of image downscaling, one pixel often corresponds to several pixels in the original image. So overly downscaled images might lead to substantial deviations in the detected keypoints when re-projected to their original positions, suggesting that extreme image size reductions should be avoided.

Here is a more intuitive result to show how reducing the size of an image is far better than enlarging it. In Figure 4a,b, Superpoint detectors identify keypoints as green dots. The enlarged image Figure 4a displays many disorganized points. Conversely, the downscaled image Figure 4a reveals distinct keypoints, such as room corners and the points where various planes of objects meet. Note that we resized the two images for paper readability: originally, their sizes varied.

Table 6. Evaluation metrics under different resized resolutions.

Size	Robustness of (Rotation, Scaling, Noise)	Distinctiveness	Matching Score
512 × 32	(0.856, 0.157, 0.551)	0.267	0.366
896 × 128	(0.827, 0.156, 0.591)	0.37	0.515
896 × 256	(0.843, 0.146, 0.663)	0.309	0.486
1024 × 64	(0.809, 0.124, 0.54)	0.427	0.53
1024 × 128	(0.832, 0.147, 0.584)	0.372	0.504
1024 × 256	(0.851, 0.134, 0.659)	0.32	0.483
1280 × 64	(0.798, 0.116, 0.527)	0.41	0.5
1280 × 128	(0.823, 0.138, 0.575)	0.353	0.479
1280 × 256	(0.849, 0.127, 0.652)	0.301	0.464
1920 × 128	(0.808, 0.124, 0.553)	0.321	0.436
1920 × 256	(0.844, 0.114, 0.644)	0.274	0.43
2048 × 128	(0.799, 0.129, 0.544)	0.309	0.425
2048 × 256	(0.837, 0.119, 0.633)	0.263	0.421
2560 × 128	(0.802, 0.111, 0.547)	0.294	0.4
2560 × 256	(0.842, 0.106, 0.644)	0.249	0.401
4096 × 128	(0.799, 0.087, 0.557)	0.257	0.339

**Figure 4.** Keypoint detected in the resized signal images. (a) Detected keypoints in an enlarged image. (b) Detected keypoints in a downscaled image.

4.2. Results of Keypoint Detectors and Descriptors for LiDAR Image

As presented in Table 3, different metrics offer a comprehensive evaluation of the detector-descriptor pipeline from various perspectives.

The metric “Number of key points” is shown in Figure 5, which is only related to detectors. FAST and BRISK algorithms detected the highest number of keypoints, but there were significant fluctuations in the counts. Comparatively, AKAZE, ORB, and Superpoint identified a reduced number of keypoints, but the consistency was notable. It is important to note that for this metric, a high number of keypoints could still contain numerous false detections. Therefore, this metric should be considered in conjunction with other accuracy indicators, especially match score, match ratio, and distinctiveness, to be more convincing.

Figure 6 depicts the computational efficiency, where the majority of the algorithms operate in less than 50 ms. After CUDA was enabled, SuperPoint runs significantly faster with minimal variance. Among all algorithms, BRISK is the most time-consuming one, and using BRISK solely as a descriptor with other detectors will hinder the overall efficiency. And descriptors utilizing BRIEF exhibit enhanced performance speed.

Figure 6 shows the robustness of the detector. Superpoint consistently demonstrates robust performance across various transformations. Among conventional detectors, AKAZE has proven effective, especially in handling rotated transformations and noise interference. Most detectors exhibit marked poor performance under scale invariance. The horizontal textures inherent in LiDAR-based images might explain such weakness: when the images are enlarged, these textures can be erroneously detected as keypoints. Another thing we may notice is that the ORB detector does not show a good robust performance for LiDAR-based images, especially considering that ORB-SLAM is one of the most famous algorithms of the last few decades.

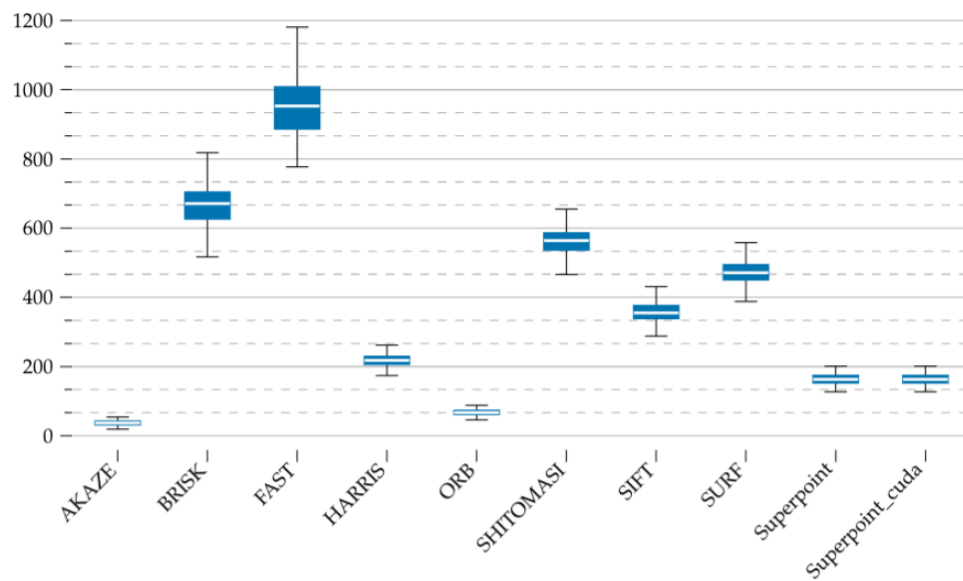


Figure 5. Number of keypoints.

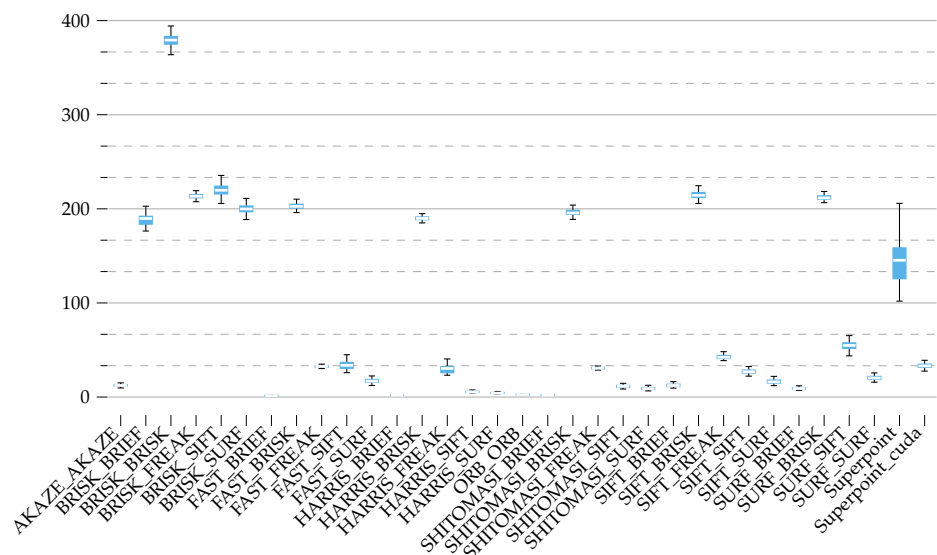


Figure 6. Computational efficiency.

As emphasized in Section 2.3, a multitude of keypoint detections and rapid matches could be useless if their accuracy is not guaranteed. Therefore, match ratio, match Score, and distinctiveness, which pertain to algorithmic accuracy, can be regarded as the most pivotal indicators across various application perspectives. Figures 7–10 present the results of these three metrics, indicating that Superpoint, when augmented with CUDA, is the most effective solution. Moreover, among traditional algorithms, AKAZE demonstrates top-tier performance across the majority of evaluated metrics, making it a commendable choice. Other methods such as AKAZE, FAST_BRIEF, HARRIS_BRIEF, have also shown good accuracy performance. It is discerned that some fusion algorithms like FAST_BRIEF combination really offer robust accuracy performance metrics. Conversely, other combinations, such as integrating the SURF descriptor with alternative detectors, yield less accurate results when compared to the complete SURF detector and descriptor algorithm.

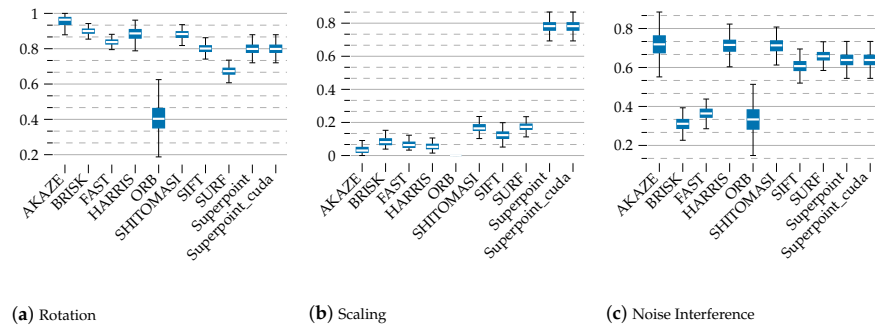


Figure 7. Robustness of the detector.

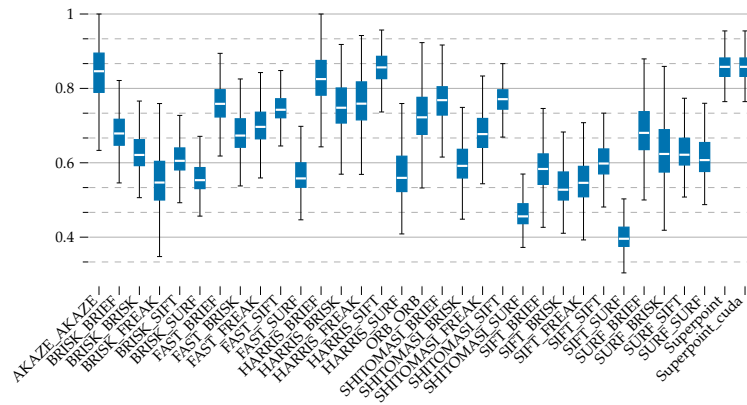


Figure 8. Match ratio.

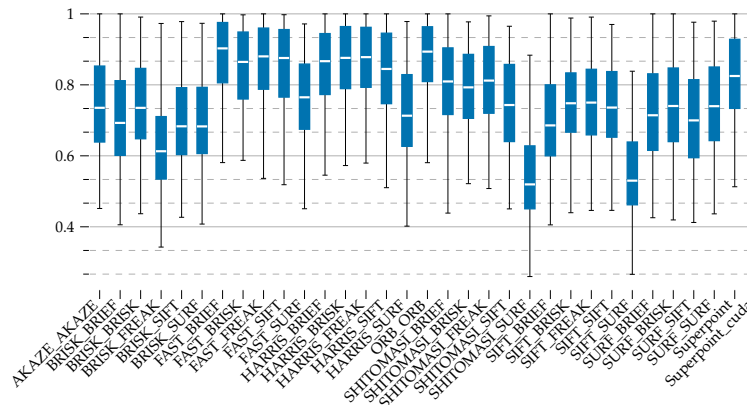


Figure 9. Match score.

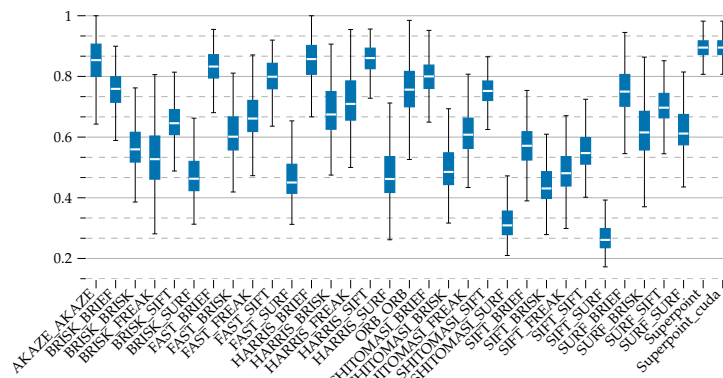
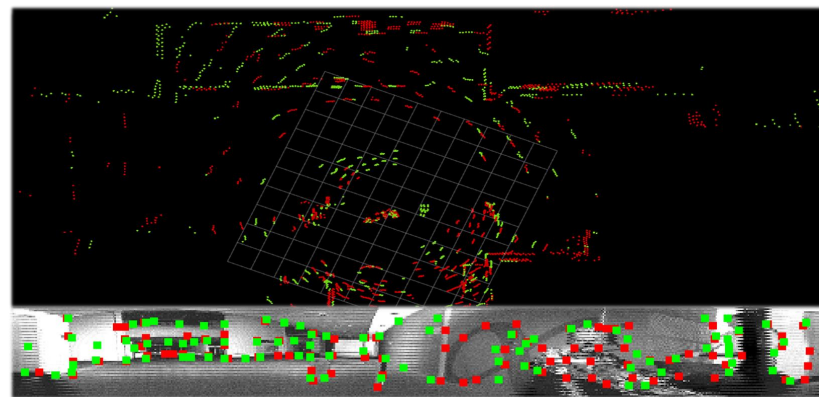


Figure 10. Distinctiveness.

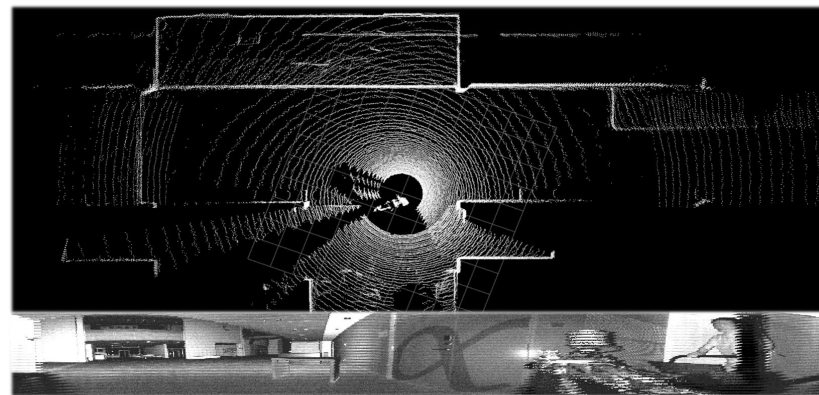
4.3. Results of LiDAR-Generated Image Keypoints Assisted LO

4.3.1. Downsampled Point Cloud

In Figure 11, we demonstrate the sample result of the downsampled point cloud in Figure 11a compared with the raw point cloud in Figure 11b. Notably, in the downsampled point cloud in Figure 11a, the red points are extracted based on signal images and the green ones are from range images. We draw the keypoints from both images to the signal image shown in the lower part of Figure 11a. The disparity between the points extracted based on these two types of images shows the significance of different LiDAR-generated images. Additionally, in the preliminary evaluation of LO, we found the accuracy of LO is lower if we only integrated the signal images instead of both modalities. This encourages us to utilize both signal and range images in the latter part.



(a)



(b)

Figure 11. Samples of point cloud data. (a) The upper part presents the downsampled point cloud from our LiDAR-based method, while the bottom illustrates keypoint distribution in the signal image: red from the signal image and green from the distance image. (b) Raw point cloud and signal image.

4.3.2. LO-Based Evaluation

In our experiment, various numbers of neighbor points are utilized, ranging from three to seven for each type of LiDAR-generated image. We selected part of them to show the result here based on the principle that it is more accurate but with fewer points. As we found in the previous section, the Superpoint has reliable keypoints detected, so we utilize this DL method to extract keypoints in our proposed approach while KISS-ICP is the point cloud registration and LO method. Table 7 shows the performance of LO based on different sizes of neighbor point sizes in both indoor (Lab space, Hall) and outdoor (Open road and Forest) environments.

As shown in Table 7, in the scenarios of Open road, Lab space (hard), and Hall (Large), the LO from KISS-ICP applying raw point cloud cannot work properly with large drift, which the error can not be calculated. Meanwhile, our proposed approach works all the

time. Additionally, even when applying raw point cloud to KISS-ICP works, our approach can achieve comparable translation state estimation while being more robust in the rotation state estimation across most of the situations.

Table 7. Performance evaluation of LO (KISS-ICP) with raw point cloud and our downsampled point cloud, ‘Sig’ and ‘Rng’ represent the size of neighboring point areas for the signal and range images, respectively, denoted as Sig_Rng.

Neighbor Size (Sig_Rng)	Open Road	Forest (Translation Error (Mean/Rmse) (m), Rotation Error (Deg))	Lab Space (Hard) (Translation Error (Mean/Rmse) (m), Rotation Error (Deg))	Lab Space (Easy) (Translation Error (Mean/Rmse) (m), Rotation Error (Deg))	Hall (Large)
4_4	N/A	(0.079/0.090, 6.58)	(0.052/0.062, 1.44)	(0.027/0.031, 0.99)	(1.111/1.274, 3.37)
4_5	N/A	(0.086/0.096, 7.22)	(0.043/0.051, 1.51)	(0.031/0.035, 1.05)	(0.724/0.819, 2.95)
4_7	(0.817/0.952, 2.33)	(0.082/0.102, 7.78)	(0.039/0.046, 1.46)	(0.028/0.033, 0.98)	(0.583/0.660, 2.88)
5_4	(1.724/2.038, 2.10)	(0.085/0.100, 6.81)	(0.059/0.070, 1.71)	(0.025/0.028, 0.98)	(1.065/1.242, 2.73)
5_5	(2.176/2.410, 1.76)	(0.108/0.203, 6.96)	(0.037/0.043, 1.35)	(0.028/0.032, 0.97)	(0.707/0.801, 2.66)
5_7	(1.298/1.443, 2.71)	(0.076/0.084, 6.11)	(0.064/0.075, 1.54)	(0.025/0.028, 0.94)	(0.676/0.746, 3.67)
7_4	(1.696/1.888, 2.31)	(0.082/0.094, 6.98)	(0.074/0.085, 1.64)	(0.027/0.032, 0.99)	(0.806/0.917, 3.68)
7_5	(1.784/2.006, 2.30)	(0.080/0.102, 7.72)	(0.033/0.047, 1.59)	(0.025/0.028, 0.97)	(0.698/0.803, 3.11)
Raw PC	N/A	(0.057/0.073, 8.91)	N/A	(0.020/0.022, 0.62)	N/A

In outdoor settings, a neighbor size 4_7 (4×4 for signal images and 7×7 for range images) exhibits notable efficacy in both translation and rotation state estimation. Conversely, in indoor environments, a neighbor size 5_5 (5×5 for signal images and 5×5 for range images) demonstrates commendable performance in the estimation of translation and rotation states, in addition to exhibiting efficient downsampling capabilities, as delineated in Table 8.

Table 8. The number of points left after downsampling with varied neighbor size: ‘Sig’ and ‘Rng’ represent the size of neighboring point areas for the signal and range images, respectively, denoted as Sig_Rng.

Neighbor Size (Sig_Rng)	Open Road	Forest	Lab Space (Hard) Number of Points (pts)	Lab Space (EASY) Number of Points (pts)	Hall (Large)
4_4	2650	7787	6435	6360	4742
4_5	3206	7812	6416	6302	4792
4_7	4784	11,447	9518	9392	7094
5_4	3182	7843	6409	6333	4776
5_5	3183	7568	6446	6292	4783
5_7	4763	11,519	9513	9386	7066
7_4	4760	11,631	9445	9356	7070
7_5	4756	11,627	9469	9378	7078
Raw PC	131,072	131,072	131,072	131,072	131,072

Based on the above result, we apply the neighbor size 4_7 for outdoor settings and the neighbor size 5_5 for indoor settings to further extend the performance evaluation by including the conventional keypoint detector approach and another point cloud matching approach, NDT. It is worth noting that the purpose of applying NDT here is not to compare it with KISS-ICP but to show the generalization of our proposed approach among other point cloud registration methods.

The results in Tables 9 and 10 prove that the conventional keypoint extractor can achieve comparable LO translation estimation and more accurate rotation estimation than Superpoint. This performance is obtained with much less CPU and memory utilization and fewer cloud points, but higher odometry publishing rates. Similar results are achieved by the NDT-based approach, which validates the above result in a certain way. Notably, the memory consumption using KISS-ICP with raw point cloud in Table 9 is lower than others. As our observation indicates, the primary reason behind this is the drift, resulting in few points for the point cloud registration.

Table 9. Evaluation of LO based on conventional and DL keypoint detectors with KISS-ICP.

Evaluation Indicators	Approaches		KISS-ICP			
			Outdoor		Indoor	
	AKAZE	Superpoint	Raw PC	AKAZE	Superpoint	Raw PC
Translation Error (<i>m</i>)	0.096/0.107	0.082/0.102	N/A	0.092/0.099	0.037/0.043	N/A
Rotation error (<i>deg</i>)	4.27	7.78	N/A	1.22	1.71	N/A
CPU (%)	263.16	457.59	544.61	82.57	425.93	572.50
Mem (<i>MB</i>)	247.62	308.67	165.73	198.27	232.85	84.37
Avg Pts	4849	11,447	131,072	1176	6446	131,072
Odom Rate (<i>Hz</i>)	10.0	4.0	2.95	10.0	7.6	2.82

Table 10. Evaluation of LO based on conventional and DL keypoint detectors with NDT.

Evaluation Indicators	Approaches		NDT			
			Outdoor		Indoor	
	AKAZE	Superpoint	Raw PC	AKAZE	Superpoint	Raw PC
Translation Error (<i>m</i>)	0.115/0.126	0.090/0.098	N/A	0.102/0.114	0.054/0.071	N/A
Rotation error (<i>deg</i>)	4.84	5.66	N/A	1.15	1.31	N/A
CPU (%)	100.39	325.69	571.12	82.20	338.54	581.30
Mem (<i>MB</i>)	285.06	548.16	705.43	253.95	290.34	645.21
Avg Pts	4849	11,447	131,072	1176	6446	131,072
Odom Rate (<i>Hz</i>)	10.0	4.6	3.34	10	8.21	1.20

5. Conclusions and Future Work

To mitigate computational overhead while ensuring the retention of a sufficient number of dependable keypoints for point cloud registration in LO, this study introduces a novel approach that incorporates LiDAR-generated images. A comprehensive analysis of keypoint detection and descriptors, originally designed for conventional images, is conducted on the LiDAR-generated image. This not only informs subsequent sections of this paper but also sets the stage for future research endeavors aimed at enhancing the robustness and resilience of LO and SLAM technology. Building upon the insights gleaned from this analysis, we propose a methodology for down-sampling the raw point cloud while preserving the integrity of salient points. Our experiments demonstrate that our proposed approach exhibits comparable performance to utilizing the complete raw point cloud and, notably, surpasses it in scenarios where the full raw point cloud proves ineffective, such as in cases of drift. Additionally, our approach exhibits commendable robustness in the face of rotational transformations. The computation overhead of our approach is lower than the LO utilizing raw point cloud but with a higher odometry publishing rate.

Building on the methodologies and insights of this study, there exists a clear trajectory for enhancing the LiDAR-generated image keypoint extraction process within the comprehensive framework of the SLAM system. Subsequent research could focus on the synergistic integration of features extracted from LiDAR-generated images with those from point cloud data, with an objective to capitalize on the unique advantages of each modality to achieve superior mapping and localization accuracy. Additionally, conceptualizing and implementing a lightweight yet robust SLAM system, augmented by sensory data from devices such as an IMU, can further advance the domain, addressing inherent challenges like drift and swift orientation alterations in real-world navigation scenarios. Beyond traditional methodologies, the incorporation of deep learning techniques into the SLAM pipeline offers potential avenues for refinement, especially in the domains of keypoint extraction, feature matching, and adaptability to varied environmental conditions.

Author Contributions: Conceptualization, X.Y. and H.Z.; methodology, X.Y. and H.Z.; software, X.Y., H.Z. and S.H.; validation, X.Y., H.Z. and S.H.; data analysis, H.Z. and S.H.; writing—original draft preparation, X.Y. and H.Z.; writing—review and editing, X.Y., H.Z. and S.H.; visualization, X.Y., H.Z. and S.H.; supervision, T.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Research Council of Finland’s AeroPolis project (Grant No. 348480) and RoboMesh project (Grant No. 336061).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, Q.; Yu, X.; Queralta, J.P.; Westerlund, T. Multi-Modal Lidar Dataset for Benchmarking General-Purpose Localization and Mapping Algorithms. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; pp. 3837–3844. [CrossRef]
2. Sier, H.; Li, Q.; Yu, X.; Peña Queralta, J.; Zou, Z.; Westerlund, T. A benchmark for multi-modal lidar slam with ground truth in gnss-denied environments. *Remote Sens.* **2023**, *15*, 3314. [CrossRef]
3. Harris, C.; Stephens, M. A combined corner and edge detector. In Proceedings of the Alvey Vision Conference, Manchester, UK, 31 August–2 September 1988; Volume 15, pp. 10–5244.
4. Moravec, H. *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*; Technical Report; Carnegie Mellon University: Pittsburgh, PA, USA, 1980.
5. Dey, N.; Nandi, P.; Barman, N.; Das, D.; Chakraborty, S. A comparative study between Moravec and Harris corner detection of noisy images using adaptive wavelet thresholding technique. *arXiv* **2012**, arXiv:1209.1558.
6. Shi, J.; Tomasi. Good features to track. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 21–23 June 1994; pp. 593–600. [CrossRef]
7. Rosten, E.; Drummond, T. Machine learning for high-speed corner detection. In Proceedings of the Computer Vision—ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 430–443.
8. Calonder, M.; Lepetit, V.; Strecha, C.; Fua, P. BRIEF: Binary robust independent elementary features. In *Proceedings of the Computer Vision—ECCV 2010*; Daniilidis, K., Maragos, P., Paragios, N., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 778–792.
9. Alahi, A.; Ortiz, R.; Vanderghyest, P. FREAK: Fast Retina Keypoint. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 510–517. [CrossRef]
10. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
11. Lowe, D. Object recognition from local scale-invariant features. In Proceedings of the 7th IEEE International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999; Volume 2, pp. 1150–1157. [CrossRef]
12. Bay, H.; Tuytelaars, T.; Van Gool, L. SURF: Speeded up robust features. In *Proceedings of the Computer Vision—ECCV 2006*; Leonardis, A., Bischof, H., Pinz, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 404–417.
13. Leutenegger, S.; Chli, M.; Siegwart, R.Y. BRISK: Binary Robust invariant scalable keypoints. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2548–2555. [CrossRef]
14. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2564–2571.
15. Alcantarilla, P.F.; Nuevo, J.; Bartoli, A. Fast explicit diffusion for accelerated features in nonlinear scale Spaces. In Proceedings of the British Machine Vision Conference, Bristol, UK, 9–13 September 2013.
16. Yang, X.; Cheng, K.T.T. Local Difference Binary for Ultrafast and Distinctive Feature Description. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 188–194. [CrossRef] [PubMed]
17. Hassaballah, M.; Awad, A.I. *Deep Learning in Computer Vision: Principles and Applications*; CRC Press: Boca Raton, FL, USA, 2020.
18. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
19. DeTone, D.; Malisiewicz, T.; Rabinovich, A. Superpoint: Self-supervised interest point detection and description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–23 June 2018; pp. 224–236.
20. Wu, T.; Fu, H.; Liu, B.; Xue, H.; Ren, R.; Tu, Z. Detailed analysis on generating the range image for lidar point cloud processing. *Electronics* **2021**, *10*, 1224. [CrossRef]
21. Point Cloud Library. How to Create a Range Image from a Point Cloud. Available online: https://pcl.readthedocs.io/projects/tutorials/en/latest/range_image_creation.html (accessed on 13 September 2023).
22. Tampuu, A.; Aidla, R.; van Gent, J.A.; Matiisen, T. LiDAR-as-Camera for End-to-End Driving. *Sensors* **2023**, *23*, 2845. [CrossRef] [PubMed]
23. Pacala, A. Lidar as a Camera—Digital Lidar’s Implications for Computer Vision. Ouster Blog. 2018. Available online: <https://ouster.com/blog/the-camera-is-in-the-lidar/> (accessed on 13 September 2023).
24. Tsiourva, M.; Papachristos, C. LiDAR Imaging-based attentive perception. In Proceedings of the 2020 International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, 1–4 September 2020; pp. 622–626. [CrossRef]
25. Sier, H.; Yu, X.; Catalano, I.; Queralta, J.P.; Zou, Z.; Westerlund, T. UAV Tracking with Lidar as a Camera Sensor in GNSS-Denied Environments. In Proceedings of the 2023 International Conference on Localization and GNSS (ICL-GNSS), Castellon, Spain, 6–8 June 2023; pp. 1–7.

26. Catalano, I.; Sier, H.; Yu, X.; Queralta, J.P.; Westerlund, T. UAV Tracking with Solid-State Lidars: Dynamic Multi-Frequency Scan Integration. *arXiv* **2023**, arXiv:2304.12125.
27. Yu, X.; Salimpour, S.; Queralta, J.P.; Westerlund, T. General-Purpose Deep Learning Detection and Segmentation Models for Images from a Lidar-Based Camera Sensor. *Sensors* **2023**, *23*, 2936. [[CrossRef](#)] [[PubMed](#)]
28. Chen, X.; Vizzo, I.; Läbe, T.; Behley, J.; Stachniss, C. Range Image-based LiDAR Localization for Autonomous Vehicles. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 5802–5808. [[CrossRef](#)]
29. Heinly, J.; Dunn, E.; Frahm, J.M. Comparative evaluation of binary features. In Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October 2012.
30. Mikolajczyk, K.; Tuytelaars, T.; Schmid, C.; Zisserman, A.; Matas, J.; Schaffalitzky, F.; Kadir, T.; Van Gool, L. A Comparison of affine region detectors. *Int. J. Comput. Vis.* **2005**, *65*, 43–72. [[CrossRef](#)]
31. Tareen, S.A.K.; Saleem, Z. A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK. In Proceedings of the 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 3–4 March 2018; pp. 1–10. [[CrossRef](#)]
32. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets Robotics: The KITTI Dataset. *Int. J. Robot. Res. (IJRR)* **2013**, *32*, 1231–1237. [[CrossRef](#)]
33. Balntas, V.; Lenc, K.; Vedaldi, A.; Mikolajczyk, K. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 26–17 July 2017.
34. Mukherjee, D.; Wu, Q.M.J.; Wang, G. A comparative experimental study of image feature detectors and descriptors. *Mach. Vis. Appl.* **2015**, *26*, 443–466. [[CrossRef](#)]
35. Strecha, C.; von Hansen, W.; Van Gool, L.; Fua, P.; Thoennessen, U. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8. [[CrossRef](#)]
36. Bojanić, D.; Bartol, K.; Pribranić, T.; Petković, T.; Diez, Y.; Mas, J. On the Comparison of Classic and Deep Keypoint Detector and Descriptor Methods. *arXiv* **2007**, arXiv:2007.10000.
37. Sager, C.; Janiesch, C.; Zschech, P. A survey of image labelling for computer vision applications. *J. Bus. Anal.* **2021**, *4*, 91–110. [[CrossRef](#)]
38. Rapson, C.J.; Seet, B.C.; Naeem, M.A.; Lee, J.E.; Al-Sarayreh, M.; Klette, R. Reducing the pain: A novel tool for efficient ground-truth labelling in images. In Proceedings of the 2018 International Conference on Image and Vision Computing New Zealand (IVCNZ), Auckland, New Zealand, 19–21 November 2018; pp. 1–9. [[CrossRef](#)]
39. Wang, X.; Jin, Y.; Cen, Y.; Wang, T.; Tang, B.; Li, Y. Lightn: Light-weight transformer network for performance-overhead tradeoff in point cloud downsampling. *arXiv* **2022**, arXiv:2202.06263.
40. Zou, B.; Qiu, H.; Lu, Y. Point cloud reduction and denoising based on optimized downsampling and bilateral filtering. *IEEE Access* **2020**, *8*, 136316–136326. [[CrossRef](#)]
41. Zhang, J.; Singh, S. LOAM: Lidar odometry and mapping in real-time. In Proceedings of the Robotics: Science and Systems, Berkeley, CA, USA, 12–16 July 2014; Volume 2, pp. 1–9.
42. Shan, T.; Englot, B. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 4758–4765.
43. Wang, H.; Wang, C.; Chen, C.L.; Xie, L. F-loam: Fast lidar odometry and mapping. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October–24 January 2021; pp. 4390–4396.
44. Besl, P.J.; McKay, N.D. Method for registration of 3-D shapes. In Proceedings of the Sensor Fusion IV: Control Paradigms and Data structures, Spie, Boston, MA, USA, 12–15 November 1992; Volume 1611, pp. 586–606.
45. Koide, K.; Yokozuka, M.; Oishi, S.; Banno, A. Voxelized GICP for fast and accurate 3D point cloud registration. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 11054–11059.
46. Dellenbach, P.; Deschaud, J.E.; Jacquet, B.; Goulette, F. CT-ICP: Real-time elastic LiDAR odometry with loop closure. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 5580–5586.
47. Vizzo, I.; Guadagnino, T.; Mersch, B.; Wiesmann, L.; Behley, J.; Stachniss, C. Kiss-icp: In defense of point-to-point icp—simple, accurate, and robust registration if done the right way. *IEEE Robot. Autom. Lett.* **2023**, *8*, 1029–1036. [[CrossRef](#)]
48. Biber, P.; Straßer, W. The normal distributions transform: A new approach to laser scan matching. In Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No. 03CH37453), Las Vegas, NV, USA, 27–31 October 2003; Volume 3, pp. 2743–2748.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.