Building Better Models: A Benchmark on Feature Extractors and Matchers for Structure from Motion in Construction Sites

Master of Science in Technology Thesis
Department of Computing, Faculty of Technology
Robotics and Autonomous Systems
University of Turku

Author:
Carlos Roberto Cueto Zumaya

Supervisors:
Msc. Iacopo Catalano
Docent Jorge Peña Queralta
Antti Kolu

June 2024

The originality of this thesis has been checked in accordance with the University of Turku quality assurance system using the Turnitin OriginalityCheck service.

UNIVERSITY OF TURKU
Department of Computing, Faculty of Technology

CARLOS ROBERTO CUETO ZUMAYA: Building Better Models: A Benchmark on Feature Extractors and Matchers for Structure from Motion in Construction Sites

Master of Science in Technology Thesis, 122 p.
Robotics and Autonomous Systems
June 2024

The increased popularity of Structure from Motion (SfM) techniques has revolutionized 3D reconstruction in various fields, including construction site mapping. SfM enables the generation of detailed and accurate 3D models from real-world scenes captured in 2D images, facilitating better project monitoring, analysis, and decision-making. Key to reconstruction using SfM, is the feature extraction and matching process, which identifies and matches corresponding points in different images to reconstruct the scene accurately. Benchmarks have been conducted to evaluate the performance of traditional and learning-based feature extraction and matching. However, these studies have not focused on construction site mapping and predominantly evaluate single components of the SfM pipeline. This thesis aims to provide a comprehensive evaluation of traditional and learning-based feature extraction and matching methods within the SfM pipeline, focusing on the reconstruction quality and their effectiveness in construction site mapping. Traditional feature extraction methods like SIFT, AKAZE, and ORB, are compared against advanced learning-based methods like SuperPoint, D2-Net, DISK, SOS-Net, and R2D2. Similarly, matching techniques like SuperGlue and LightGlue are compared against traditional ones such as brute-force and nearest neighbor. Results indicate that learning-based methods generally outperform traditional approaches in terms of robustness and accuracy, particularly in scenarios with complex lighting and limited visual overlap. Deep learning methods also demonstrated superior feature extraction and matching capabilities, producing more detailed and accurate reconstructions with fewer missing areas, as well as, faster processing times thanks to their GPU-accelerated implementations. The findings underscore the importance of selecting appropriate techniques based on specific scene characteristics and desired outcomes when performing construction site mapping using SfM, and the potential of learning-based methods to enhance the quality and efficiency of 3D reconstruction in this domain. Likewise, the results highlight the need for further research on refining these methods to handle more complex real-world scenarios effectively, improving their robustness and computational efficiency for broader practical adoption.

Keywords: SfM, Structure from Motion, Benchmark, Dataset, Local Features, Matching Features, 3D Reconstruction, Construction Sites

Contents

List of Figures

List of Tables

List Of Acronyms

AKAZE	Accelerated-KAZE
ALS	Airborne Laser Scanning
BRIEF	Binary Robust Independent Elementary Features
CNN	Convolutional Neural Network
FAST	Features from Accelerated Segment Test
FED	Fast Explicit Diffusion
LBD	Local Binary Descriptor
MVS	Multi-View Stereo
NN	Nearest Neighbour
ORB	Oriented FAST and Rotated BRIEF
PnP	Perspective-n-Point
RANSAC	Random Sample Consensus
ROS	Robot Operating System
SIFT	Scale-Invariant Feature Transform
SfM	Structure from Motion
SLAM	Simultaneous Localization and Mapping
SURF	Speeded-Up Robust Features
TLS	Terrestrial Laser Scanning

1 Introduction

The use of Structure from Motion (SfM) techniques in 3D map reconstruction has grown due to the increased availability of high-quality and cost-effective cameras, as well as advancements in computer vision algorithms. Its usage is widespread in various fields such as robotics [1]–[3], geosciences [4], and urban planning [5].

In construction, efficient and precise capture of spatial geometry is crucial for applications such as asset visualization [6], progress monitoring [7], quality control [8], and safety analysis [9]. This innovation presents an alternative to costly methods like Terrestrial Laser Scanning (TLS), making spatial reconstruction more accessible [10]. Yet, the performance of SfM can be constrained by factors such as the quality of the input images and the effectiveness of the feature extraction and matching processes, particularly under challenging conditions like low light, occlusions, and repetitive patterns [11].

Traditionally, hand-crafted feature extractors like SIFT, AKAZE and ORB have been used for feature extraction in SfM; as for feature matching, techniques such as brute-force or nearest neighbor are commonly employed, each offering its trade-offs in terms of accuracy, efficiency, and robustness [12], [13]. Overall, these approaches have proven effective in many situations, but their limitations become apparent as the complexity of the data (images) increases, resulting in scenarios with high outlier ratios up to 99% [11].

In response, the research community has explored learned features and matching techniques. These methods have shown promising results across various applications, often outperforming traditional techniques in robustness and accuracy [14]–[17]. By combin-

ing a learned feature extractor with a tailored matching technique like SuperPoint [18] with SuperGlue [16], one can expect to achieve state-of-the-art performance. This thesis aims to evaluate whether learned-based methods consistently outperform traditional ones, particularly in complex dynamic environments like construction sites, and evaluate the trade-offs between the two approaches.

1.1 Related works

This section explores the evolving landscape of 3D map reconstruction. Additionally, an overview of benchmarks will provide context for comparing performance, trade-offs, and limitations of traditional and learning-based feature extraction and matching methods. Finally, the application of SfM and MVS techniques in construction site mapping will be discussed, emphasizing the transformative impact of these technologies on construction site analysis, monitoring, and visualization.

1.1.1 Benchmarking SfM Methods

Beginning with [19], we see an emphasis on establishing a robust benchmark for assessing the performance of 3D reconstruction technologies. The authors introduced the *"Tanks and Templates"* dataset and benchmark, which address significant gaps in the field of image-based 3D reconstruction, particularly the need for realistic, challenging, and diverse datasets that reflect the complexities of real-world environments. Precision, recall, and F-scores are used as key metrics for assessing the performance, providing a quantitative basis to compare the fidelity and completeness of the reconstructed models against the ground-truth data.

Other studies, such as those by [20] and [21], have investigated the accuracy, efficiency, and application suitability of various SfM pipelines and software tools in complex urban settings and emergency response scenarios. These studies highlight the importance

of detailed evaluations across different operational environments. In contrast, [22] compares multiple 3D reconstructions generated by commercial software and open-source pipelines, revealing significant differences in performance based on the tool's ability to handle large image datasets and the environmental complexity. Furthermore, [23] emphasizes the importance of high-density point clouds for detailed analysis in construction sites, indicating that the choice of tools can significantly influence the accuracy and efficiency of the 3D reconstruction process.

The studies above suggest that future research should focus on incorporating real-world operational conditions to make sure that the evaluation metrics accurately reflect the complexity of practical applications.

1.1.2 Benchmarking Feature Extractors and Matching Methods

Recent advancements in feature extraction and matching methods have increased the interest in SfM due to their crucial role in enhancing the accuracy and robustness of the process in complex environments. However, evaluating performance remains challenging due to the complex interactions between different components within the SfM pipeline. [24] presented a comprehensive comparative analysis of both handcrafted and learning-based feature extractors. Their findings suggest that while binary feature extractors are more computationally efficient, float-type descriptors (e.g., SIFT or L2Net) provide superior accuracy and robustness. Furthermore, they highlight that generalizability and computational efficiency continue to be challenges for descriptors such as those derived from convolutional neural networks (CNNs) like L2-Net [25]. Similarly, [13] conducted a detailed comparison of several handcrafted extraction methods, including SIFT, SURF, KAZE, AKAZE, ORB, and BRISK. Their findings indicated that SIFT demonstrated superior performance in ensuring scale and rotation invariance across various scenes, although ORB remains preferred for its efficiency in real-time applications.

In further discussion, [26] expanded the research by comparing a wider selection and

combination of feature extractors and descriptors. The study indicates that handcrafted methods can still compete with learning-based methods in terms of repeatability and robustness. This finding is significant as it challenges the notion that deeper, more complex networks result in better performance. However, the results also acknowledge that deep learning methods can outperform traditional algorithms in terms of efficiency, especially when they are GPU-accelerated (e, g. SuperPoint), bringing into question the trade-offs between performance and computational efficiency, especially in the context of hardware acceleration.

Similarly, [27] provided a comparative analysis of local features, considering the entire pipeline, including 3D reconstruction, and their effectiveness in photogrammetry applications typical of civil and cultural heritage projects. The study demonstrated that learning-based methods can achieve comparable accuracy under optimal image conditions, with methods like R2D2 exhibiting the best keypoints in terms of reliability and repeatability despite the high RMSE in some challenging 3D models.

In evaluating the practical impact of descriptors, [28] conducted a comprehensive analysis comparing handcrafted and learned local features. Although learned descriptors have shown superior discriminative power in traditional matching scenarios, they do not always translate into better results, particularly in practical applications and in image reconstruction under challenging conditions. To address this evaluation gap, the authors extended their evaluation assessment to include reconstruction metrics relevant to practical applications, such as the number of registered images, observations per image, reprojection error, and track length. This approach provided a more thorough, evaluation of the descriptors' impact on the SfM pipeline and the resulting 3D models.

Unlike the aforementioned studies, which primarily evaluate single components either in isolation or within specific applications, this thesis conducts a comparative analysis of both traditional and learning-based feature extraction and matching methods within the Structure-from-Motion (SfM) pipeline. By integrating both feature extractors and

feature matchers into a single evaluation, the research aims to provide insights into the effectiveness, efficiency, and applicability of each combination in real-world construction settings.

1.1.3 Application of SfM in Construction Sites

Advancements in software tools and SfM techniques have enabled more efficient and accurate 3D reconstruction of construction sites, offering transformative solutions for enhancing construction site analysis, monitoring, and visualization.

ConstructAide, as described in [29], employs a technique that integrates user-guided registration with model-assisted SfM. This approach facilitates tasks such as architectural visualization, performance monitoring, and 3D navigation through construction progress, allowing for immersive and photorealistic exploration of construction sites. Additionally, the study demonstrates that the user-assisted SfM method outperforms traditional techniques in both real-world and synthetic data tests, highlighting its potential impact on construction management and architectural visualization. However, it also acknowledges limitations in handling large datasets and complex scenes when high-quality BIM models are unavailable.

Similarly, [6] explores the use of mobile iOS devices for documenting field sites, focusing on evaluating the accuracy of consumer-grade 3D model reconstruction platforms. The study involved mapping a recently excavated trench using three SfM-MVS apps and two iOS LiDAR-based mobile apps, comparing the results to assess model accuracy, scaling errors, and orientation. The findings indicate that while iOS LiDAR apps can produce accurately scaled models, only SfM could generate correctly oriented models. However, limitations such as vertical and horizontal scale errors or rational errors were identified. The study aimed to identify the most suitable technology for on-the-go field documentation, offering insights into the potential of mobile devices for this purpose.

Another practical application is [8] which evaluated the practical application of un-

manned aerial vehicles (UAVs) for inspecting a river trail bridge. Their study demonstrates the integration of UAVs into routine structural assessment practices by creating a high-density, multiscale photorealistic 3D model of the bridge using hierarchical SfM. The research highlights the efficacy of UAV inspections in providing detailed and accurate 3D representations, presenting a more cost-effective and efficient alternative to traditional laser scanning for bridge inspections. Despite challenges such as image noise and environmental factors affecting data quality, the UAV-based method provided a competent alternative to traditional laser scanning in terms of model completeness and detail resolution.

Together, the aforementioned studies highlight the transformative potential of SfM and the shift towards integrating digital technologies into construction site mapping, facilitating better decision-making, enhanced analysis, and monitoring of construction projects. However, the limitations in handling large datasets, complex scenes, and the dependency on high-quality BIM models or optimal environmental conditions for data collection indicate the need for further refinement of these technologies. Overcoming these challenges, enhancing computational efficiencies, and improving the robustness of SfM techniques under diverse operational conditions will be crucial for their broader adoption and for realizing their full potential in practical scenarios.

1.2 Structure

The structure of this thesis is organized as follows. Chapter 2 outlines the research background, discussing the theoretical foundations of feature extraction, matching techniques, and Structure from Motion (SfM). Additionally, various traditional and deep learning-based feature extraction and matching methods (which will be evaluated) are described. The Incremental SfM pipeline is also introduced as the foundation for the 3D reconstruction process employed in this thesis.

Chapter 3 provides a design overview, detailing the datasets utilized, and the methodology adopted to benchmark the feature extraction and matching methods. It further discusses the criteria for selecting methods, datasets, and evaluation metrics, as well as the reconstruction setup, including the software libraries employed, the combinations of feature extraction and matching methods.

The results are presented in Chapter 4, where the outcomes of the 3D reconstruction process are examined. The evaluation is divided into two main sections: Indoor and Outdoor scenes, where the datasets, reconstructions, and performance metrics are thoroughly analyzed.

Finally, Chapter 5 presents the conclusion, summarizing the research findings and suggesting potential directions for future research in this area. An analysis of the results is provided, highlighting the strengths and weaknesses of the different feature extraction and matching methods, as well as the implications of these findings for both practice and research.

2 Background

2.1 Overview of Structure from Motion (SfM)

Structure from Motion (SfM) is a photogrammetry technique used to generate three-dimensional structures from a set of two-dimensional image sequences that may be taken from multiple viewpoints, enabling applications ranging from autonomous navigation and augmented reality to historical preservation and virtual tourism [30]. Several approaches to SfM have been developed, each with its strengths and weaknesses, such as Incremental [31], Global [32], Graph-Based [33], and Hierarchical [34]. Although SfM is a well-established and extensively researched field, there is no universal solution, and the choice of method is often determined by the specific requirements of the application.

Incremental SfM is particularly notable for its application to unordered image collections. The subsequent section, 2.2, provides an overview of the Incremental SfM pipeline, highlighting its key components and stages. Given the scope and objectives of this thesis, the feature extraction and matching stages are of particular interest. The section 2.3 and 2.4 will therefore explore each process independently and the traditional and learning-based methods that will be evaluated in Chapter 4.

2.2 Incremental SfM

Incremental SfM has traditionally been predominant in the field due to its straightforward implementation and computational efficiency. Over time, it has been refined and

optimized to handle large-scale datasets, making it a popular choice for many SfM applications. In COLMAP [31], [35], Incremental SfM incorporates enhancements to address common limitations related to robustness, accuracy, and scalability. These improvements include a more robust geometric verification strategy, a next best view selection algorithm, and a more efficient outlier management system. The pipeline is divided into two main stages: correspondence search and incremental reconstruction. Figure 2.1 provides a visual representation of the pipeline.

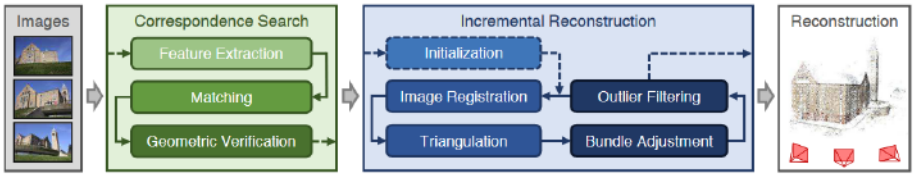


Figure 2.1: Incremental SfM pipeline. Image Source [31]

Correspondence Search

The initial step in Incremental SfM involves identifying overlapping images and establishing correspondences between image pairs, which is critical for successful reconstruction. It includes feature extraction, feature matching, and geometric verification. The process begins with the extraction of local features and descriptors (keypoints) from the images, which are then matched across pairs of images to identify potential overlaps. After identifying matches, a geometric verification process eliminates incorrect matches. This involves estimating geometric transformations, such as homographies or fundamental matrices, and employing techniques such as RANSAC [36] to robustly filter out outliers.

Incremental Reconstruction

Once reliable correspondences have been established, the reconstruction process starts with the selection of an initial pair of images that is used to configure the geometry of the reconstruction. New images are incrementally integrated into the reconstruction through

the solution of the Perspective-n-Point (PnP) problem [37], which estimates the camera poses by aligning 2D-3D correspondences between the image and the 3D points in the reconstruction. Each addition of a new image involves re-triangulating points and refining the reconstruction through bundle adjustment [38] to minimize the overall reprojection error. This process is repeated until all images have been added to the reconstruction.

2.3 Feature Extraction

For each image I_i in the dataset, a set of keypoints and descriptors $\{K_i, D_i\}$ are extracted. The keypoints are points of interest in the image, and the descriptors are the feature vectors that describe the local appearance of them, together they form the features of the image. To obtain a set of reliable points, they should be invariant to changes, such as rotation, scale, and illumination, and should be distinctive enough to be matched accurately [39].

Historically, feature extraction in SfM relied on handcrafted algorithms such as SIFT (Scale-Invariant Feature Transform), AKAZE, and ORB (Oriented FAST and Rotated BRIEF). These methods are known for their robustness in detecting and describing local features. The evolution from traditional methods to deep learning-based approaches is due to traditional methods falling short in complex real-world scenarios. In contrast, learning-based methods leverage the power of Convolutional Neural Networks (CNN) to learn feature descriptors from large datasets, offering improved adaptability and robustness across more varied and challenging conditions. This transition reflects advancements in architecture design and training strategies and aligns with increasing computational capabilities available today, enabling more complex models to be deployed in real-time applications. The extraction methods to be evaluated in this thesis are detailed below.

2.3.1 Traditional Methods

SIFT. Introduced by Lowe [40] is one of the most renowned feature detection and description algorithms. It identifies and describes keypoints based on the Difference of Gaussian (DoG), and operates in a multi-stage pipeline that begins with the detection of potential keypoints in the image by identifying local extrema in the DoG scale-space, which is formulated as

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \tag{2.1}$$

where G is the Gaussian Blur function, I is the image, and σ is the scale parameter. These potential points are refined by filtering out those with low contrast or poorly defined edges, enhancing the algorithm's robustness to noise and illumination changes. Next, is the orientation assignment stage, where each keypoint is assigned a dominant orientation based on the local gradient directions to ensure rotation invariance, figure 2.2 shows the magnitude and orientation of a SIFT keypoint.

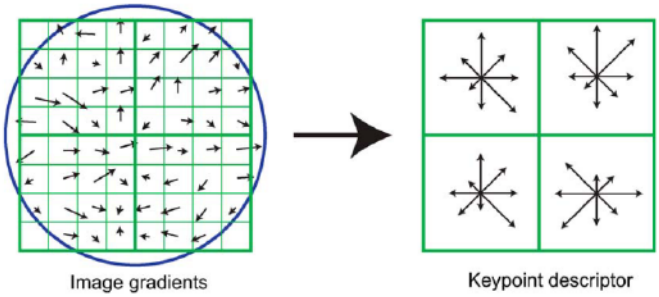


Figure 2.2: SIFT Gradient Magnitude and Orientation. Image Source [40]

Finally, the descriptor is computed as a histogram of gradient orientations within a local neighborhood around the keypoint, resulting in a 128-dimensional vector that encapsulates the point's appearance. This high-dimensional vector representation allows for precise feature matching across different images, providing a basis for the robustness of SIFT in handling various image transformations such as rotation, scale, and affine distortion [39]. SIFT's patent expired in 2020, making it free to use for commercial and

non-commercial purposes [41]. Figure 2.3 shows an example of a single SIFT keypoint and its corresponding descriptor in an image section from the CONSLAM dataset [42]. Note, the descriptor has been reshaped from 128-dimensions to 8×16 for visualization purposes.

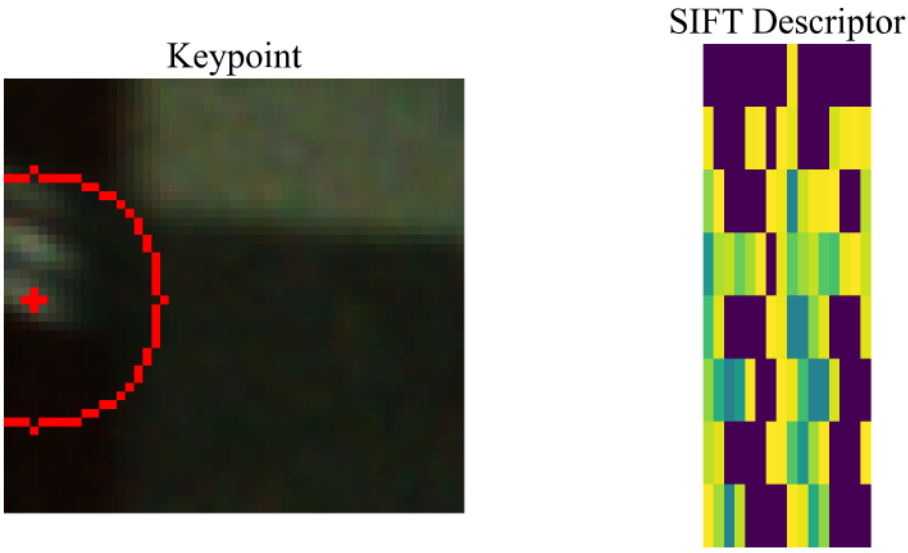


Figure 2.3: SIFT Keypoint and Descriptor

ORB. Introduced by Rublee, Rabaud, Konolige, *et al.* [43] as a fast alternative to SIFT and SURF while maintaining good accuracy, it is based on FAST detector [44] and BRIEF descriptor [45]. First, FAST detects corners in the image by using a pixel intensity comparison test. A pixel p is considered a corner (or keypoint) if at least three of its neighbors are significantly brighter or darker than its neighbors. Then Harris corner detection [46], which is a score based on the local gradient changes around the pixel, is applied to find the top K corners. ORB introduces rotation invariance by computing the intensity centroid of the pixels around the selected keypoint with a radius r based on the moments of the circle patch. Figure 2.4 shows the FAST circle in action.

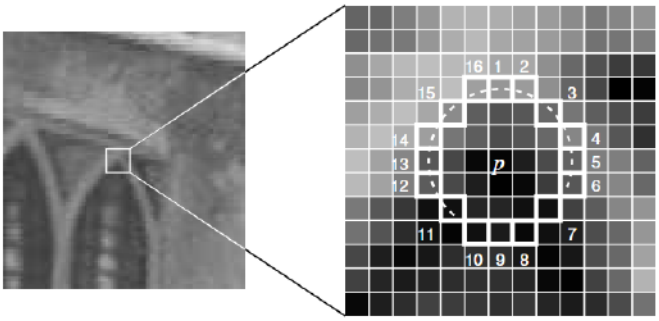


Figure 2.4: FAST Intensity Circle. Image Source [44]

The rotation-aware BRIEF descriptor is then computed for each keypoint in the image, where a 32-dimensional binary string is generated by comparing the intensity of the pixels in the patch surrounding the keypoint. The descriptor is determined by comparing the intensities of two pixels x and y, which are rotated by the orientation θ to align it with the keypoint's orientation. The binary nature of the BRIEF descriptor allows for fast matching by using the Hamming distance, which is computationally less expensive than the Euclidean distance. Figure 2.5 shows a single ORB keypoint and its corresponding descriptor over an image section from the CONSLAM dataset [42].

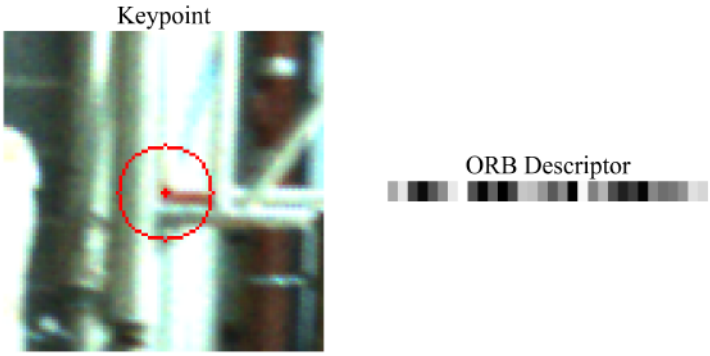


Figure 2.5: ORB Keypoint and Descriptor

The synergy between FAST and BRIEF makes ORB a fast and efficient algorithm for feature detection and description, making it suitable for real-time applications where com-

putational resources are limited while providing a free alternative to patented algorithms like SIFT and SURF.

AKAZE. Introduced by Alcantarilla and Solutions [47], it leverages the advantages of non-linear scale spaces for feature detection and description, and unlike KAZE [48] it employs a faster framework for the computation of features using Fast Explicit Diffusion (FED) schemes, which are used to construct the nonlinear space. FED schemes use a sequence of n explicit diffusion steps with varying time steps to build the scale space efficiently. To detect keypoints, AKAZE computes the response of the scale-normalized determinant of the Hessian matrix at each level of the scale space, and is defined as

$$\det(H_L) = \sigma_{i,\text{norm}}^2 (L_{i,xx} L_{i,yy} - L_{i,xy}^2) \tag{2.2}$$

where $L_{i,xx}$, $L_{i,yy}$, $L_{i,xy}$ are the second-order horizontal, vertical and cross derivative respectively, of the scale space L at level i, and $\sigma_{i,\text{norm}}$ is the normalization factor [47]. The determinant of the Hessian matrix is used to identify regions with significant changes in intensity that are indicative of potential keypoints. Similar to SIFT, AKAZE finds the dominant orientation in a circular area of radius r to ensure rotation invariance.

For the descriptor, AKAZE uses a modified version of the Local Difference Binary (m-LDB) descriptor. The descriptor for each keypoint is calculated by rotating the image based on the previously computed orientation, dividing the patch surrounding the point into a grid, and sampling the intensity values of the pixels. Figure 2.6 shows the grid used to sample the intensity values.

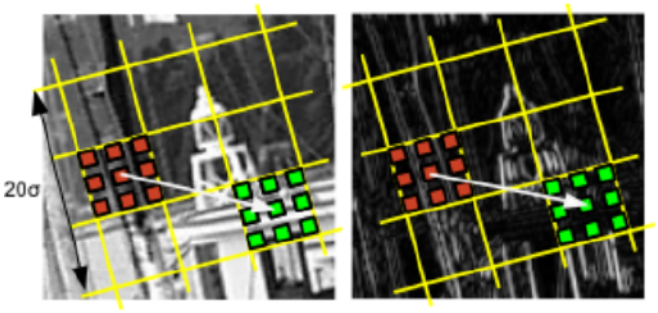


Figure 2.6: AKAZE Binary Descriptor. Image Source [47]

This comparison results in a binary descriptor, that can be efficiently matched using the Hamming distance, reducing computational costs [47]. Figure 2.7 shows an example of a single AKAZE keypoint and its corresponding descriptor in an image section from the CONSLAM dataset [42].

Figure 2.7: AKAZE Keypoint and Descriptor

2.3.2 Learning-based Methods

D2-Net. Introduced by Dusmanu, Rocco, Pajdla, *et al.* [14], D2-Net represents a paradigm shift in feature extraction with its detect-and-describe approach. Typically, traditional approaches perform keypoint detection before the description of surrounding regions, a process that often struggles under conditions of significant appearance changes.

In contrast, D2-Net postpones keypoint detection until after the formation of a dense image representation has been completed, leading to more stable keypoints based on higher-level information. The network architecture, shown in Figure 2.8, is divided into two main components: the feature extraction network and the keypoint detection network. The feature extraction network is responsible for generating a dense representation of the input image, while the keypoint detection network identifies the most salient points in the image. These two networks are trained jointly to optimize the performance of the final descriptor.

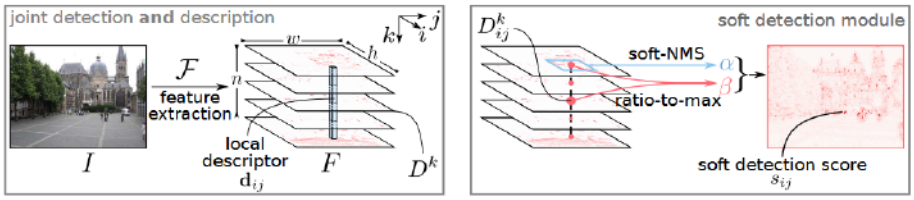


Figure 2.8: D2-Net detect-and-describe CNN network. Image source [14].

To improve the robustness against scale changes, D2Net employs an image pyramid to extract features at different scales, which are then concatenated to form the final descriptors. The network prioritizes the detection of local maxima of a response map, effectively identifying the most salient points in the image.

Training is done over the MegaDepth [49] dataset without the need for manual annotations by using pixel correspondences extracted from large-scale SfM reconstructions from COLMAP [31], and its loss function is a triplet margin ranking loss that encourages the repeatability of keypoints and distinctiveness of descriptors. Figure 2.9 shows an example of a single D2Net keypoint and its corresponding descriptor in an image section from the CONSLAM dataset. Note, the descriptor has been reshaped from 512-dimensions to 32×16 for visualization purposes.

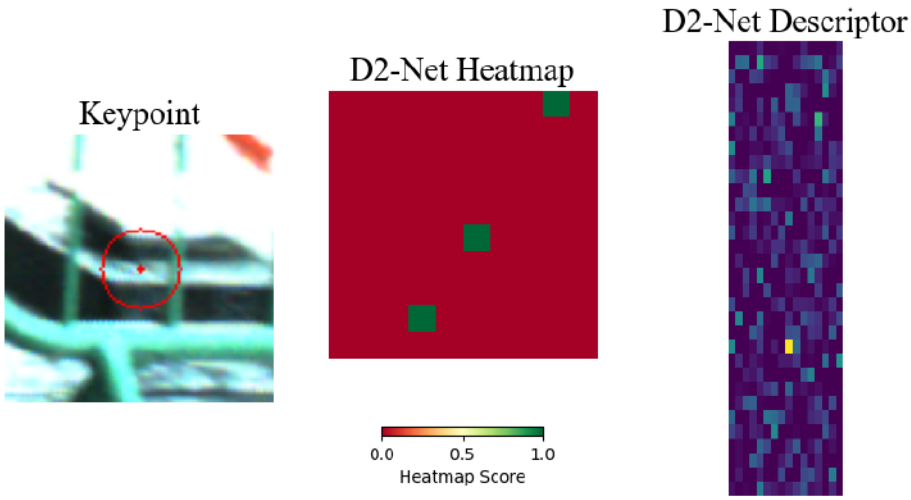


Figure 2.9: D2-Net keypoint, heatmap and descriptor

DISK. Introduced by Tyszkiewicz, Fua, and Trulls [17], DISK marks a significant shift in learning local features by employing reinforcement learning (RL) principles. This approach addresses the challenges associated with the inherent discreteness in keypoint selection and matching in an end-to-end fashion. The DISK network utilizes a modified U-Net architecture [50], designed with distinct output channels for keypoint heatmaps and dense descriptors, enabling the simultaneous extraction of both elements in a single forward pass. From these heatmaps, keypoints are probabilistically sampled, where the spatial distribution of potential keypoints is represented as probabilities over pixel locations.

Once keypoints are sampled, descriptors at those pixel locations are utilized to generate a distribution of potential matches based on the L2 distance between descriptors. These matches are then assessed using geometric ground truths, with each potential match receiving a reward based on its geometric consistency. DISK employs a policy gradient approach to directly optimize a reward function that evaluates the accuracy of feature matches. This contrasts with prior attempts that typically utilize gradient descent methods constrained by the differentiability of the matching process.

Figure 2.10 illustrates the DISK's heatmap and the 128-dimension descriptor from a

keypoint of an image section from the CONSLAM dataset [42]. Note, the descriptor has been reshaped from 128 to 8×16 for visualization purposes.

By consolidating the extraction and matching of features into a single probabilistic model, DISK facilitates a more direct and effective optimization process. This is accomplished by creating a probabilistic relaxation of the matching process, allowing for the gradient of the expected reward to be approximated. Through this method, DISK bridges the gap between training and inference while maintaining sufficient convergence properties for reliable training from scratch.

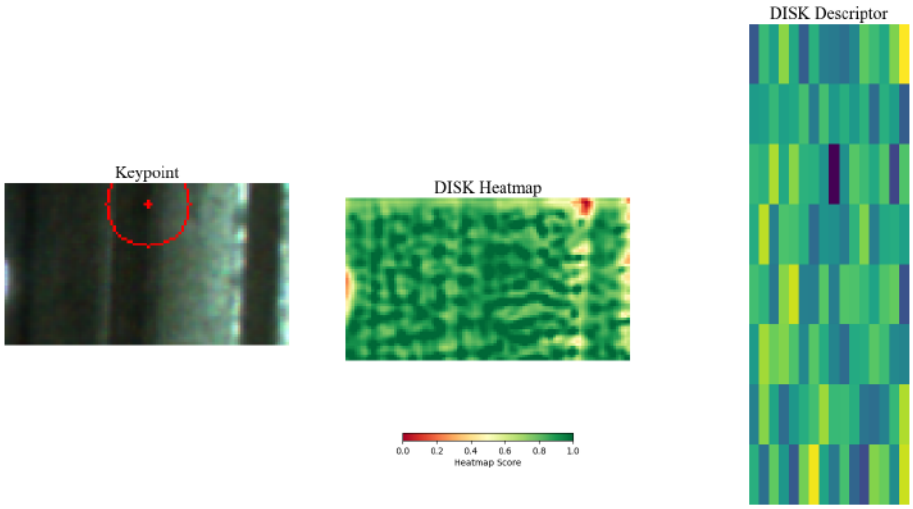


Figure 2.10: DISK Keypoint, heatmap and descriptor

R2D2. Introduced by Revaud, De Souza, Humenberger, *et al.* [15], R2D2 proposes a method where keypoint detection and description are jointly learned along with the assessment of the local descriptors' uniqueness. This method is based on the insight that image regions characterized by high repeatability do not inherently possess discriminative features conducive to reliable matching. The network's architecture, based on the L2-Net [25] backbone, generates dense local descriptors for each pixel and confidence maps that evaluate the repeatability and reliability of each detected point. The repeatability loss L_{rep} is defined to ensure that keypoints are detected consistently across different views, calculated using cosine similarity between transformed repeatability maps, and a

peakiness function to ensure local maxima are distinct. The reliability loss $L_{AP,R}$ is used to further refine the learning process, ensuring that the network focuses on regions of the image that are not only repeatable but also reliable for matching. This approach ensures that the descriptors are optimized for areas that are most likely to contribute positively to the matching performance. For the descriptor, a listwise ranking loss based on a differentiable Average Precision metric is used, which is a recent advancement in metric learning [51], [52]. This method allows for the simultaneous optimization of the repeatability and reliability of keypoints, significantly outperforming state-of-the-art methods on benchmarks such as the HPatches [53] and Aachen Day-Night datasets [11], [54]. In essence, R2D2 enhances conventional feature detection by incorporating this novel AP-based loss function, which is key in enabling the network to identify and describe points in a manner that also considers their reliability for subsequent matching tasks.

Figure 2.11 illustrates R2D2 heatmaps and the 128-dimension descriptor from a keypoint of an image section from the CONSLAM dataset [42]. Note, the descriptor has been reshaped from 128 to 8×16 for visualization purposes.

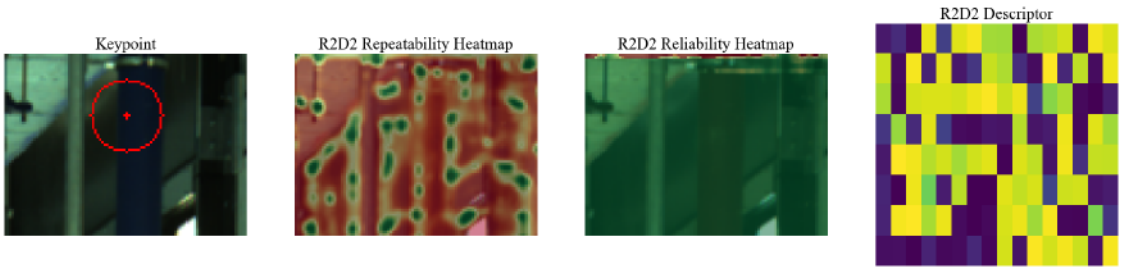


Figure 2.11: R2D2 Keypoint, Repeatability and Reliability Heatmap, and Descriptor

SuperPoint. Introduced by DeTone, Malisiewicz, and Rabinovich [18], it proposes a novel network aimed at enhancing interest point detection and description through a self-supervised approach using a fully convolutional network to predict both simultaneously. A key innovation is the introduction of Homographic Adaptation, a technique that leverages multiscale multi-homography warping to augment training data and improve interest

point detection, repeatability and facilitate cross-domain adaptation (transitioning from synthetic to real-world data). The network's architecture features a shared VGG-based encoder [55] that processes the input image, reducing its dimensionality. Following the encoding process, the network splits into two decoder heads used to predict the interest points and descriptors. The interest point head predicts the probability of a pixel being an interest point, while the descriptor head predicts a semi-dense grid of L2-normalized N-dimensional descriptors. This dual-head design enables task-specific weight adjustments, figure 2.12 illustrates the network's architecture.

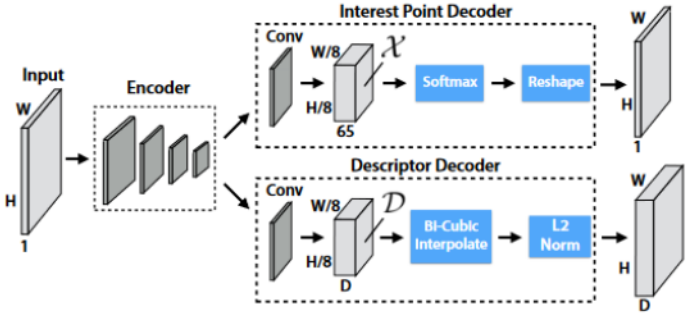


Figure 2.12: SuperPoint Architecture. Image Source [18]

The network's training process consists of three stages: pre-training, self-labeling, and joint-training. Initially, the interest point detector (MagicPoint) is pre-trained using a synthetic dataset. This dataset is specially designed for this purpose, and consists of simple geometric shapes that define potential interest points, reducing ambiguity and facilitating learning without labeled data. Following pre-training, the network uses Homographic Adaptation where MagicPoint is applied to unlabeled real-world images to generate pseudo-ground truth labels for the interest points by applying multi-homography warping. In the final stage, the network is fine-tuned using the pseudo-ground truth data, both MagicPoint and the descriptor network are trained jointly. This training approach optimizes the network to perform both tasks simultaneously, improving the network's performance. Figure 2.13 shows the network's training process.

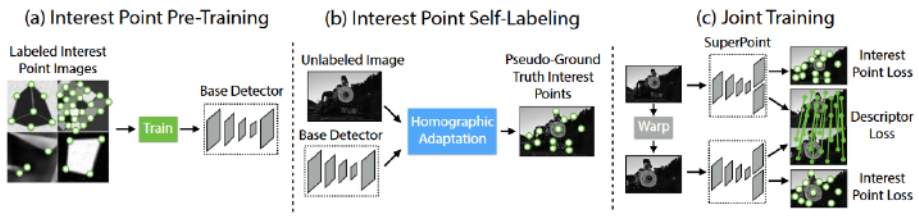


Figure 2.13: SuperPoint Training Process. Image Source [18]

Despite its superior performance in numerous scenarios, especially in homography estimation tasks (e.g., HPatch dataset), SuperPoint can be matched or even surpassed by traditional methods like SIFT under conditions requiring extremely high sub-pixel precision. Additionally, the reliance on synthetic pre-training and self-supervision may introduce biases or limitations in detecting interest points not represented in the synthetic training phase or that do not benefit from the Homographic Adaptation process [18]. Figure 2.14 shows an example of a single SuperPoint keypoint and its corresponding descriptor on an image section from the CONSLAM dataset [42]. Note, the descriptor has been reshaped from 256-dimensions to 16×16 for visualization purposes.

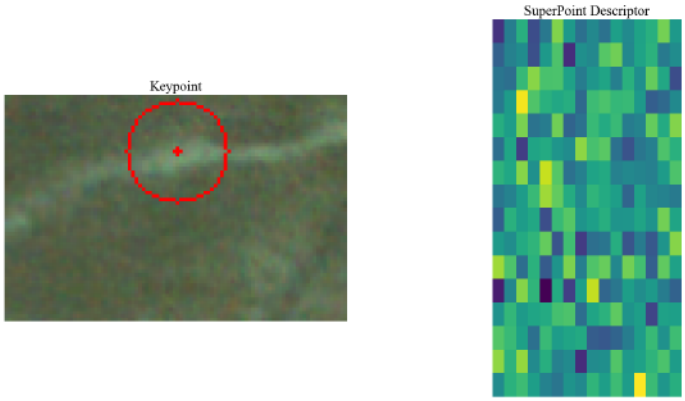


Figure 2.14: SuperPoint Keypoint and Descriptor

SOSNet. Introduced by Tian, Yu, Fan, *et al.* [56], SOSNet is a network that uses Second Order Similarity Regularization (SOSR) along with First Order Similarity (FOS) to create robust local descriptors and enforce the principle that a positive pair of matching points should exhibit similar distances regarding other points in the embedding space. Typically,

SOS has been used in clustering tasks due to its ability to capture the second-order statistics of the data, capturing more structural information while being robust to deformations and distortions. The network is based on a modified version of L2-Net [25], it employs a CNN to transform the input image into an N-dimensional descriptor, each normalized to unit vectors to maintain consistent scale across features. The training combines SOSR, FOS, and a Quadratic Hinge Triplet (QHT) loss, significantly enhancing the learning of matching and non-matching descriptors through the formula

$$LT = LFOS + \lambda \cdot RSOS \tag{2.3}$$

where LT is the total loss, $LFOS$ is the First Order Similarity loss, $RSOS$ is the Second Order Similarity loss, and λ is a balancing parameter [56]. This approach ensures not only the closeness of matching descriptors but also a consistent structural relationship across the entire dataset (SOS). See figure 2.15 for a visual representation of the embedding space learned by SOSNet. Additionally, SOSNet incorporates a von Mises-Fischer distribution-based evaluation method to assess the quality and characteristics of the learned descriptors post-training, focusing on the structural and distributional properties of the descriptor space shaped by the training process, including the effects of both FOS and SOS.

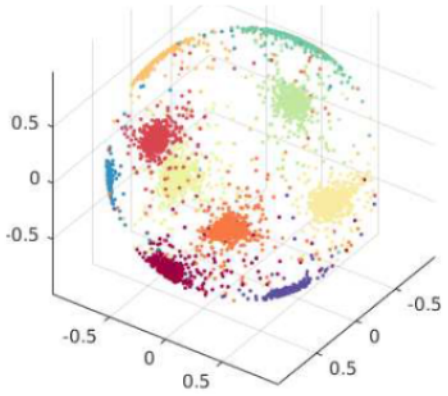


Figure 2.15: SOSNet Embedding Space. Image source [56]

Figure 2.16 shows an example of a single SOSNet keypoint and its corresponding descriptor of an image section from the CONSLAM dataset [42]. Note, the descriptor has been reshaped from 128-dimensions to 8×16 for visualization purposes.

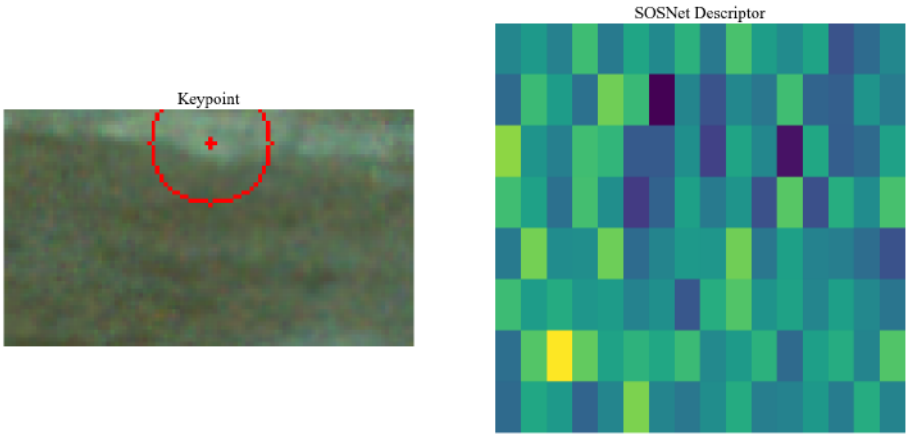


Figure 2.16: SOSNet Keypoint and Descriptor

2.4 Feature Matching

For every pair of images in the dataset (I_i, I_j), the goal is to find the set of correspondences $\{C_{ij}\}$, where each correspondence C_{ij} is a pair of matching features (k_i, k_j), with $k_i \in K_i$ and $k_j \in K_j$ [31], [57].

The naive approach, often referred to as the brute-force method, involves exhaustively comparing every possible pair of features between two images to find matches, defined by the equation

$$\frac{n(n-1)}{2} \tag{2.4}$$

where n is the total number of images in the datasets [57]. The computational complexity of this approach is $O(N_I^2 N_{F_i}^2)$, where N_I is the total number of images and N_{F_i} is the number of features in image I_i [31]. This approach is referred as 'naive' due to its simplicity and lack of efficiency, particularly in large-scale datasets where the number of potential comparisons grows quadratically with the number of images and features, leading to prohibitive computational costs. Therefore, alternatives have been proposed

to reduce the number of correspondences to be found. For instance, [58] leverages the use of NetVLAD [59] to narrow the number of comparisons by first performing a global search using the NetVLAD descriptors, and then a local search using the descriptors of the keypoints found in the global search. This approach significantly reduces the number of correspondences to be found, making it more computationally efficient, and will be used in this research to evaluate the performance of feature extractors and matching techniques.

In parallel to the evolution of feature extraction, feature matching has undergone significant advancements too, particularly with the introduction of techniques that enhance the accuracy and efficiency of finding correspondences between images, and dynamically adjusting keypoint matching based on both geometric and appearance data, significantly outperforming traditional methods in complex scenes.

2.4.1 Traditional Methods

Nearest Neighbor (NN). One of the simplest yet most effective classification techniques, characterized by its long-standing history and robust performance across various applications. The premise of NN is that similar points are typically found close to each other in the feature space, and it operates by identifying the closest data points to a query point within a dataset. For feature matching, the process can be executed in two primary modes: exact and approximate search [60]. In exact NN, the goal is to determine the point that has the minimum distance from the query point. On the other hand, approximate NN aims to find points within a certain proximity to the closest point, sacrificing some accuracy for increased computational speed. It is predominantly affected by the choice of distance metric and the structure of the dataset. Common distance metrics used include Euclidean (used to match SIFT features), Manhattan, and Hamming (used to match AKAZE and ORB features), each suitable for different types of data characteristics and dimensionalities [13]. The effectiveness of the NN search is highly dependent on these

metrics, as they define how 'closeness' is quantified between points. To enhance the performance and accuracy of NN, two main threshold approaches are utilized: *ratio-based* and *distance-based* [13].

A *ratio-based* threshold evaluates the validity of a match by examining the ratio of the distance between the closest and the second-closest match. A typical implementation of this method is Lowe's ratio test, used in the SIFT algorithm [40]. The method's reliance on relative differences rather than absolute distances lends robustness, especially in heterogeneously scaled spaces. See the figure 2.17 for an illustration of the NN-ratio method.

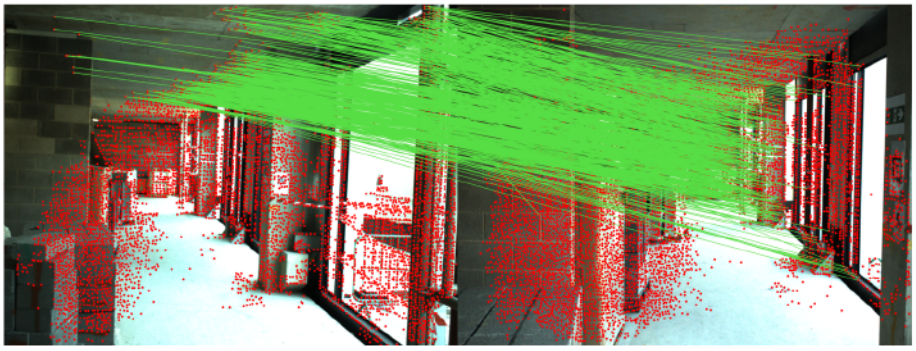


Figure 2.17: DISK + NN-ratio

A *distance-based* threshold uses a fixed distance threshold to determine match validity. It presumes consistent and meaningful distance measures across the dataset, which can be challenging if the metric distorts distances in various parts of the dataset. The choice between these two approaches is often dictated by the dataset's characteristics and the desired trade-off between accuracy and computational efficiency. See the figure 2.18 for an illustration of the NN-distance method.

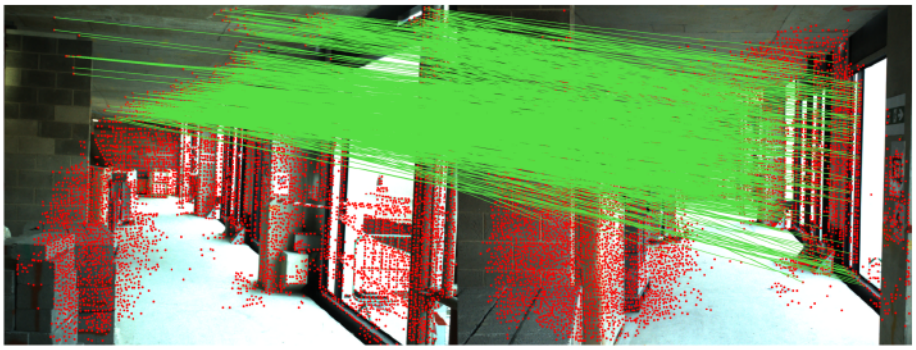


Figure 2.18: DISK + NN-distance

AdaLAM. Introduced by Cavalli, Larsson, Oswald, *et al.* [61], was designed as a lightweight alternative in outlier detection and rejection, designed to improve efficiency and effectiveness of matching features across images. AdaLAM is founded on three core assumptions: local planarity, locality, and adaptivity. Local planarity assumes that the keypoints lie on approximately planar surfaces, which simplifies the geometric modeling to affine transformations. The locality assumption ensures that the affine transformations are consistent within small neighborhoods, enhancing verification robustness. Lastly, adaptivity allows the method to adjust thresholds based on the local consistency and density of matches, maintaining robustness across different geometric and scene conditions. The process begins with the selection of confident matches (seed points) from a set of putative correspondences. For each seed point, neighboring correspondences are selected based on their consistency with the local affine model. The consistency is then evaluated and verified by using a highly parallel RANSAC algorithm with sample-adaptive inlier thresholds. See Figure 2.19 for a visual representation of the outlier rejection process. Despite its strengths, AdaLAM's limitations become apparent in scenarios involving significant non-planar features or extreme geometric transformations, where the affine model may fail to model the true transformation accurately. Additionally, the method's reliance on Difference of Gaussian (DoG) features (e.g., SOSNet, SIFT features), limits its compatibility with other feature types.

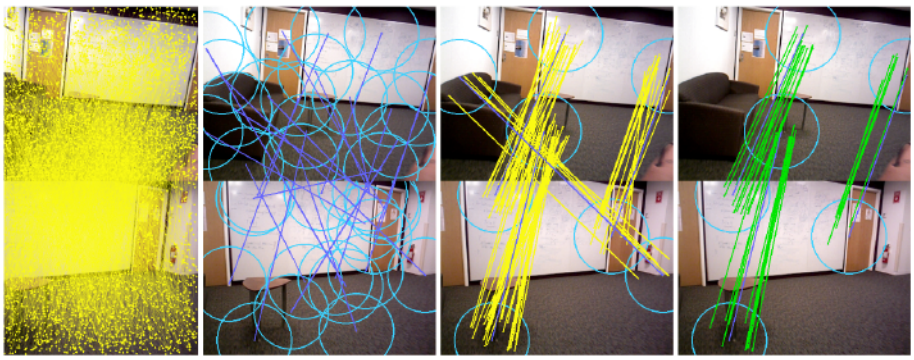


Figure 2.19: AdaLAM outlier rejection process

Points in yellow are a set of putative matches that are selected based on rough regions of interests (blue circles). For each region of interest, only consistent matches are selected (in yellow lines with blue circles) based on the affine transformation. The final set of matches is shown in green. Image Source [61]

Figure 2.20 shows a pair of images from the CONSLAM dataset [42] with their SOS-Net features being matched by AdaLAM.

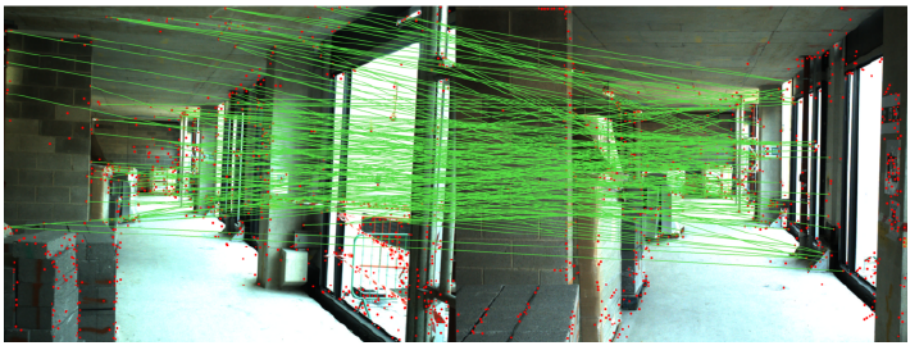


Figure 2.20: SOSNet + AdaLAM Matching Example

2.4.2 Learning-based Methods

SuperGlue. Introduced by Sarlin, DeTone, Malisiewicz, *et al.* [16], SuperGlue is a method that shifted the paradigm of feature matching by using Attentional Graph Neural Networks (AGNN). Figure 2.21 illustrates the SuperGlue architecture. The GNN predicts the costs of matching each keypoint to every other keypoint, handling the unmatchable

points more effectively, based on the idea that a keypoint can have at most a single correspondence. First, the network encodes the inputs $\{K_i, D_i\}$ for each image to produce a high-dimensional representation that includes both the original descriptor and the positional information of the keypoint, this enriched representation is crucial for enabling the attention mechanisms to simultaneously consider appearance and geometric information. In the graph, keypoints are treated as nodes with edges defined as *self-edges*, which connect keypoints within the same image to capture local context, and *cross-edges*, which connect keypoints between two images to establish potential matches. Then, the attention mechanism refines the features iteratively, enhancing the robustness of the matching process. Figure 2.22 shows attentional aggregation over a dynamic graph between keypoints.

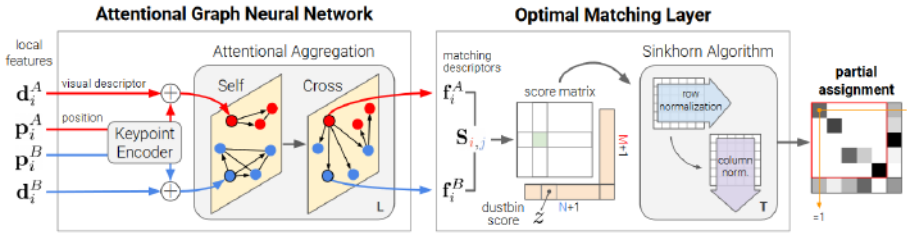


Figure 2.21: SuperGlue Architecture. Image Source [16]

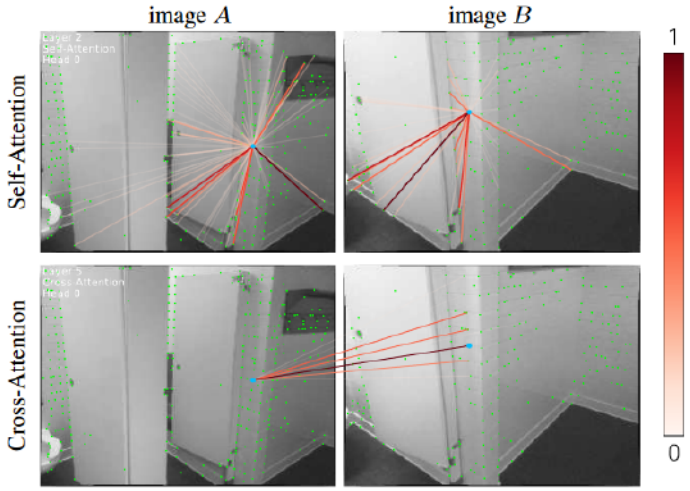


Figure 2.22: SuperGlue Attention Visualization. Image Source [16]

Each layer updates the feature vectors of keypoints by aggregating information from

connected nodes through a message-passing framework, allowing the model to adaptively focus on the most relevant for both images. The final stage involves an optimal matching layer that computes a score matrix for all potential matches, which the Sinkhorn [62]–[64] algorithm then processes to produce a probabilistic assignment matrix. This approach frames the feature matching problem as a differentiable optimal transport problem, avoiding the need for handcrafted heuristics and allowing end-to-end training. Training is performed using image pairs with known correspondences, allowing the network to learn geometric and appearance-based priors. SuperGlue is characterized by its ability to dynamically adjust the influence of each keypoint using attention mechanisms, which evaluate the contributions of each node based on the learned importance of its features. Figure 2.23 shows a pair of images from the CONSLAM dataset [42] with their SuperPoint features being matched by SuperGlue.

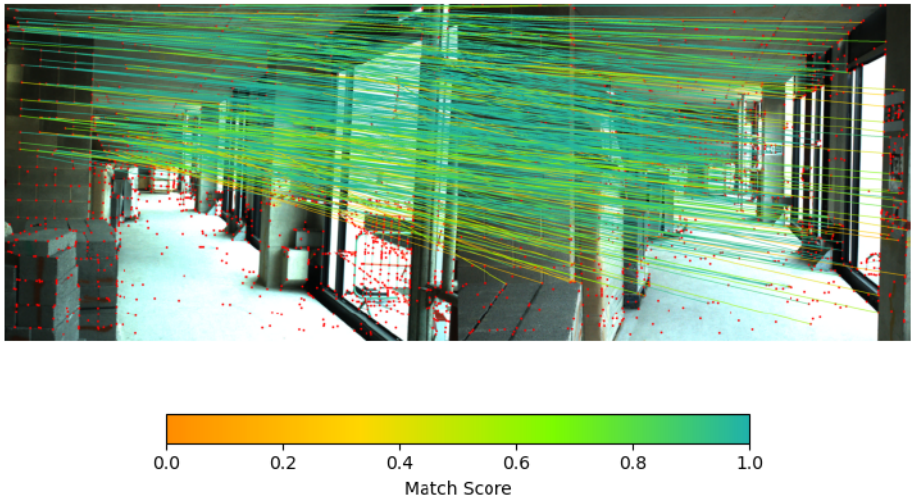


Figure 2.23: SuperPoint + SuperGlue Matching Example

LightGlue. Introduced by Lindenberger, Sarlin, and Pollefeys [65], builds upon his predecessor (SuperGlue), by optimizing various aspects of the model architecture and enhancing performance in terms of speed, accuracy, and training simplicity. The network's architecture is based on Transformers [66], it consists of a series of layers incorporating self- and cross-attention mechanisms with two sets of keypoint and descriptor embed-

dings, $\{K_i, D_i\}$, for each image. Figure 2.24 illustrates LightGlue's architecture.

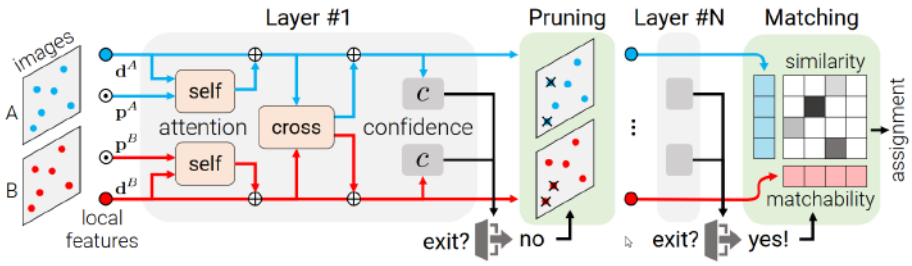


Figure 2.24: LightGlue Architecture. Image Source [65]

The attention units update each local feature's representation by aggregating information from keypoints within the same image (self-attention) and corresponding points from the other image (cross-attention). After updating, the network calculates a matching score for each feature pair across the two images, providing the likelihood that they correspond to the same point. This process results in a similarity matrix representing potential correspondences, which are thresholded to determine the final matches.

LightGlue is characterized by its ability to adapt based on the image complexity. This is achieved by making use of a confidence classifier, that determines when to halt the processing, ensuring efficiency without compromising accuracy. The classifier assesses the confidence level of the predictions at each layer, allowing the model to dynamically adjust its depth (number of layers) and width (number of points processed) as needed. See Figure 2.25 for a visual representation of this process. Notably, LightGlue's enhancements allow it to surpass SuperGlue not only in performance metrics but also in operational flexibility [65].

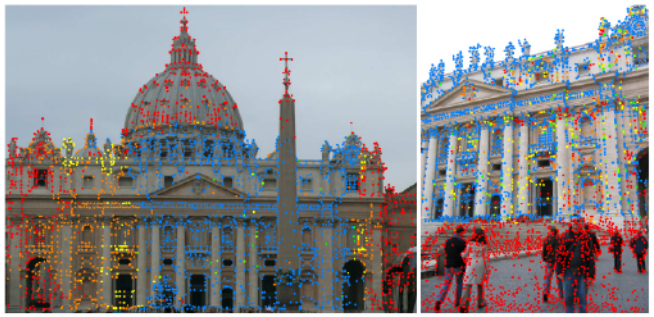


Figure 2.25: LightGlue Point Pruning

As context is aggregated, unmatchable (red) and non-repeatable (orange, yellow, green) points are discarded in progressively until only good matches (blue) are left. Image Source [65].

Figure 2.26 shows a pair of images from the CONSLAM dataset [42] with their SuperPoint features being matched by LightGlue.

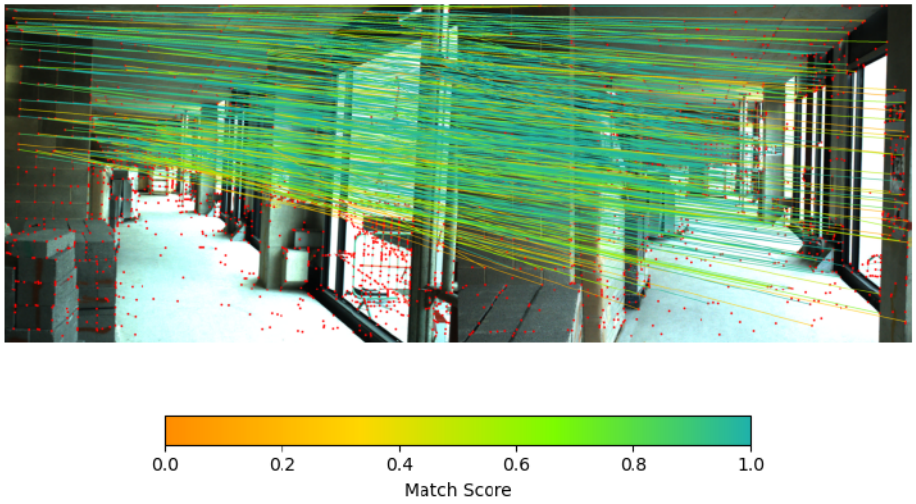


Figure 2.26: SuperPoint + LightGlue Matching Example

2.5 Summary

This chapter provided an in-depth exploration of Structure from Motion (SfM). Particularly, Incremental SfM workflow was discussed, which involves identifying correspon-

dences through feature extraction and matching, followed by incremental reconstruction. This method has been refined over time to enhance robustness, accuracy, and scalability, notable in systems like COLMAP. In the context of feature extraction and feature matching, traditional methods and learning-based methods were presented. Traditional methods like SIFT, ORB, and AKAZE have been widely used in the past, but have been displaced by learning-based methods like D2-Net, DISK, R2D2, SuperPoint, and SOS-Net, each offering unique advantages in terms of accuracy, efficiency, and robustness. Feature matching techniques have also evolved, with traditional methods like Nearest Neighbor (NN) and AdaLAM being replaced by learning-based methods like SuperGlue, LightGlue, which leverage attention mechanisms and graph neural networks to enhance the matching process. Overall, this section thoroughly examines the components and advancements in SfM, emphasizing the integration of machine learning techniques to overcome the limitations of traditional methods and adapt to the demands of modern applications. The next chapter will discuss the experiment setup and evaluation metrics used to assess the performance of feature extractors and feature matching presented in this section.

3 Design Overview

3.1 Datasets

This section introduces two publicly available datasets and a proprietary one that were used to evaluate SfM reconstruction in construction sites.

Hilti SLAM Challenge Dataset

In Helmberger, Morin, Berner, *et al.* [67], the authors emphasize the pivotal role of transitioning from academic and controlled environments to real-world applications and introduce the Hilti SLAM challenge dataset that is renewed annually. The sequences in this dataset include challenging featureless areas with diverse lighting conditions, from offices and laboratories to construction sites and parking lots.

To collect the data, the authors leverage the Robot Operating System (ROS) for processing sensor data of five cameras (Alphasense by Sevensense), two LiDARs (Ouster OS0-64 and Livox MID70), and three IMUs (Analog Devices ADIS16445, Bosch BMI085, and InvenSense ICM-20948) [67]. Figure 3.1 showcases the 3D model of the handheld device used to record the 2022 version of the dataset.

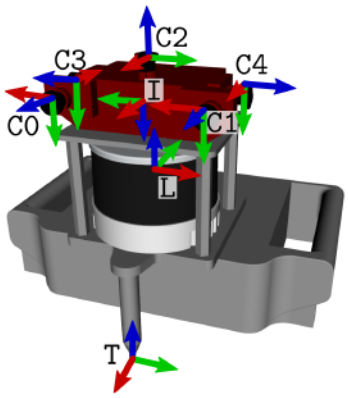


Figure 3.1: 2022 Hilti Device

The data collected includes a series of synchronized in time and spatially registered, images, point clouds, and inertial measurements, as well as a millimeter-accurate ground truth through a motion capture system and a total station [67]. The challenge and dataset can be publicly accessed[1]. The Hilti SLAM challenge dataset and its subsequent yearly iterations offer mixed indoor and outdoor construction sequences but lack periodic monitoring, limiting their ability to assess progress over time.

From this dataset, two sequences were chosen to evaluate the performance of the combination of feature extraction and matching methods. The sequences are named "Construction Site Outdoor 1" and "Construction Upper Level 1" from the years 2021 and 2022 respectively. As their name indicates, these contain images from an outdoor construction site and a level of a building under construction. The Figures 3.2 and 3.3 show a sample of the images from the sequences, they contain a total of 9,953 images and 25,148 images respectively.

Figure 3.2: Hilti Construction Site Outdoor 1

[1] https://hilti-challenge.com/index.html

Figure 3.3: Hilti Construction Upper Level 1

ConSLAM

In Trzeciak, Pluta, Fathy, *et al.* [42], the researchers identify a gap in periodic monitoring and assessment of construction sites, and introduce ConSLAM a collection of periodic scans over several months for construction progress monitoring.

To collect the data, the authors leverage ROS for processing sensor data of two cameras: one RGB (Alvium U-319c, 3.2 MP) and one Near-InfraRed (NIR) (Alvium 1800 U-501, 5.0 MP), a Lidar (Velodyne VLP-16), and an IMU (Xsens MTi-610) [42]. The data collected includes data synchronized in time and spatially registered, images, point clouds, and inertial measurements, as well as a millimeter-accurate ground truth obtained through a static Leica RTC 360 scanner [42]. The dataset can be publicly accessed [2]. Figure 3.4 showcases the devices used to record the dataset.

Figure 3.4: ConSLAM Devices. Image Source [42]

From this dataset, only the RGB images from Sequence 2 were chosen to evaluate the performance of the combination of feature extraction and matching methods. The Figure 3.5 shows a sample of the images from the sequence, it contains a total of 4,170 images

[2] https://github.com/mac137/ConSLAM

from an indoor construction site.

Figure 3.5: ConSLAM Sequence 2

Proprietary Dataset

To test a pure outdoor scenario, a local construction company provided a private dataset. This was collected on a real construction site by using an excavator machine equipped with a camera attached to the upper carriage and performing rotations and changing locations over the site. The dataset includes environmental changes in the construction site through actions such as excavation, relocation of objects within a designated area, and alteration or smoothing of gravel surfaces. Due to the proprietary nature of the dataset, the images are not publicly available. The Figure 3.6 shows a sample of the images from the dataset.

Figure 3.6: Proprietary Dataset

The table below, lists a summary of the overall datasets' characteristics used in this work

Table 3.1: Dataset Characteristics

Dataset	Map	Scenario	Challenge	# Images	# Cameras	Type
Con-SLAM	Sequence 2	Indoor	Overexposure, strong shadows, repetitive structures, feature scarcity, dynamic range issues, reflective surfaces	4170	1	RGB
Hilti	Construction Upper Level 1	Indoor	Overexposure, lighting contrast, repetitive patterns, limited features in foreground, blur and focus issues	25,148	4	Grayscale
	Construction Site Outdoor 1	Outdoor	Overexposure, dynamic objects, reflections, repetitive patterns, feature depletion in foreground (e.g. gravel), contrast issues	9,953	4	Grayscale
Private	Construction Site Outdoor	Outdoor	Dynamic background, limited natural features, lightning conditions, lens reflection, cloudy weather	477	1	RGB

3.2 Hardware and Implementation

This section describes the hardware and software used to generate the 3D reconstructions, as well as the design choices made to process the datasets. The software used in this work is the Hierarchical Localization toolbox[3] (HLOC) [58] and COLMAP[4] [31], an open-source software that provides a complete incremental SfM pipeline for 3D reconstruction from images.

3.2.1 Hardware

The hardware used to process the dataset consists of the components shown in Table 3.2. When using any deep-learning method that required the usage of a GPU, the script was restricted to using only one of the two GPUs available.

3.2.2 Software Libraries

COLMAP

COLMAP is used as the foundations of this project, since it is a modular general-purpose SfM and MVS pipeline that incorporates Incremental SfM explained in section 2.2. The

[3] https://github.com/crcz25/Hierarchical-Localization

[4] https://colmap.github.io/

Table 3.2: Hardware Components

Server Components	
Device	**Description**
CPU	AMD Ryzen Threadripper 3960X 24-core 48-threads processor
RAM	128 GB
Storage	4 TB NVME SSD
GPU	x2 NVIDIA GeForce RTX 3090
GPU Memory	x2 24GB
OS	Ubuntu 20.04.6 LTS

software also provides a set of evaluation metrics that can be used to assess the quality of the reconstruction that will be explored in section 3.3. To generate the reconstruction, only the SfM module was used, and the MVS module was ignored. When processing a 3D reconstruction, an SQLite database file with the structure of Figure 3.7 is generated, and contains the information for the 3D reconstruction process.

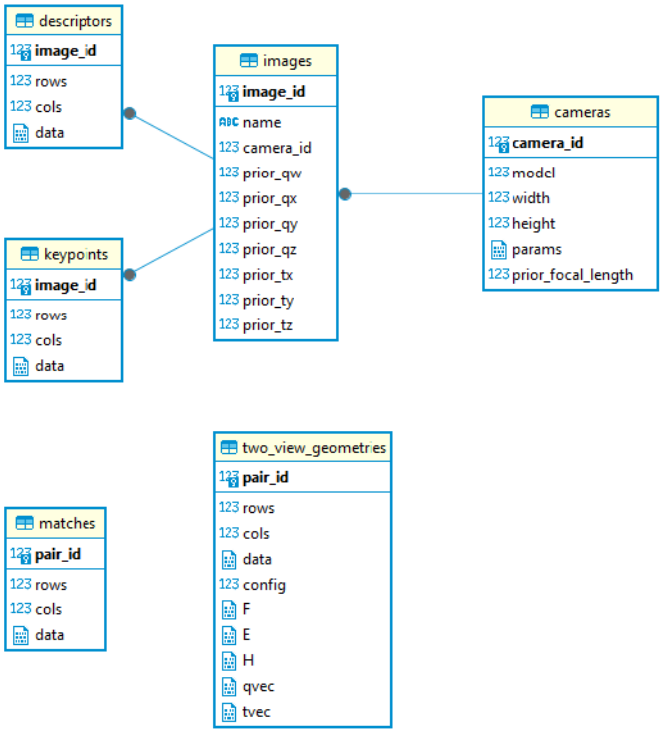


Figure 3.7: COLMAP Database Table Structure

Hierarchical Localization

As previously mentioned in Section 2.4, feature matching is a computationally expensive task in the 3D reconstruction pipeline. To address this issue, the Hierarchical Localization toolbox (HLOC) introduced by Sarlin, Cadena, Siegwart, *et al.* [58] was used to accelerate the reconstruction process. HLOC consists of a coarse-to-fine strategy that reduces the computational cost of comparing image pairs by limiting the search space to a set of prior frames that are likely to contain the same scene, it employs a global search (retrieval) and a local search (matching) to find correspondences between images.

The global search is used to generate a subset of k-candidate images (places) where the query image might belong; these images are likely to overlap spatially with the location of the query image. Once the candidate places are retrieved, the fine-localization is performed within the Incremental SfM pipeline, where local descriptors from any of the feature extraction and matching methods listed in sections 2.3 and 2.4, are used to find

correspondences between the query image and the candidate images.

Figure 3.8 provides a visual representation of how the HLOC and COLMAP pipelines were integrated in this work. The HLOC pipeline is used to generate the pairs of images to be matched, and the COLMAP pipeline is used to generate the 3D reconstruction from the matched pairs.

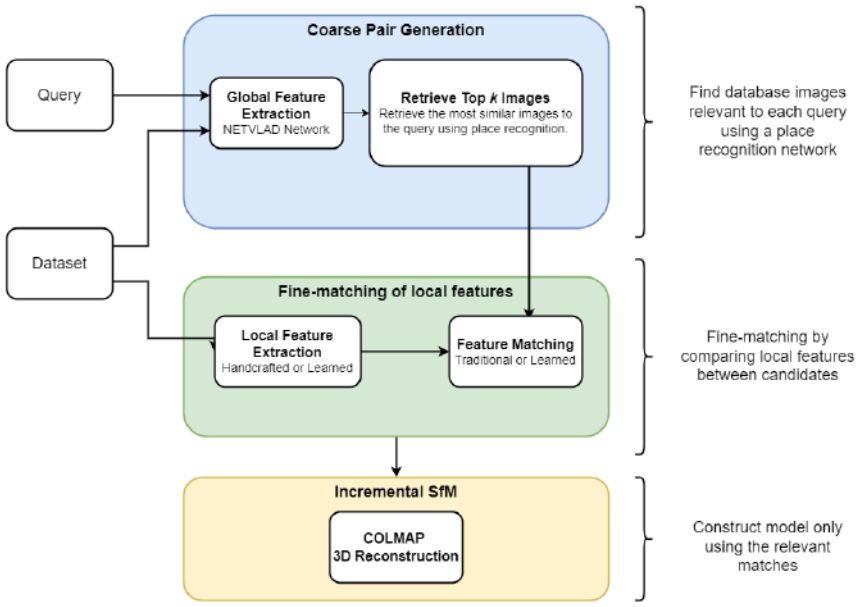


Figure 3.8: Reconstruction pipeline using HLOC and COLMAP.

3.2.3 Implementation

To reduce the size of the Hilti and ConSLAM datasets, the rosbags were downsampled to a rate of 2 Hz using the *topic_tools* package and the *throttle* node[5]. In the Hilti dataset, "Construction Site Outdoor 1" was reduced to 2,759 images, and "Construction Site Outdoor 2" was reduced to 1,220 images. Images from camera C2 were excluded, with only cameras C0, C3, C1, and C4 being used for the 3D reconstruction process. In the ConSLAM dataset, the sequence was reduced to 783 images.

After downsampling, all the images were processed using the calibration files pro-

[5] `https://wiki.ros.org/topic_tools/throttle`

vided by the dataset authors to rectify the images with the OpenCV library[6].

The implementation of the 3D reconstruction process was done using Python 3.10.14, HLOC, COLMAP and OpenCV. The process was divided into two main steps: feature extraction and matching, and 3D reconstruction. Algorithm 1 describes the feature extraction and matching process, while Algorithm 2 outlines the 3D reconstruction process.

Algorithm 1: Feature Extraction and Matching Process

Input: List of images captured from different angles
Output: Features extracted and matched between pairs of images
Step 1: Feature Detection (Local Features)
  Extract distinctive features (e.g., SIFT, ORB) in each image
  Write the features to an H5 file
Step 2: Pair Generation (Global Search)
  Retrieve k-candidate images using NETVLAD
  Generate pairs of images to be matched
  Write pairs to a .txt file
Step 3: Feature Matching (Local Search)
  Match features between pairs of images (e.g., SuperGlue, NN)
  Write the valid matches to an H5 file

Algorithm 2: 3D Reconstruction Process

Input: Image pairs and features extracted and matched
Output: 3D Reconstruction of the scene
Step 1: Database Creation
  Create a COLMAP database file
  Import images, features, and matches to the database
Step 2: Incremental SfM
  Perform geometric verification of the matches
  Run the Incremental SfM pipeline
Step 3: Point Cloud Generation
  Generate a PLY file with the 3D points from the reconstruction with the highest number of registered images

For feature extraction and matching, several methods were implemented. Handcrafted features such as AKAZE, ORB, and SIFT were used due to their availability in the OpenCV library and their performance in comparisons of feature extractors [12], [13], [26], [57]. Learned features including SuperPoint, R2D2, D2-Net, SOSNet, and DISK

[6] `https://docs.opencv.org/4.9.0/dc/dbb/tutorial_py_calibration.html`

were used due to their existing implementations in HLOC. For matching, NN-bruteforce, LightGlue, SuperGlue Fast, SuperGlue, NN-Distance, NN-Ratio, NN-Mutual, and AdaLAM were utilized. The first is a brute-force matching algorithm available in the OpenCV library, while the others are already implemented in HLOC. The combinations selected, as detailed in Table 3.3, were defined by the compatibility between the feature extractors and matching algorithms, as each extractor generates a specific type of feature descriptor requiring specific matching approaches.

Default settings were employed for both the feature extractors and matching algorithms. Configurations regarding the number of features and candidate images retrieved were adjusted for the reconstructions process, these are detailed in the next section 3.3.

For deep learning feature extractors paired with nearest neighbor matching algorithms, Mutual Check was enabled. A "mutual" match means that if a feature A from the first image is the best match for a feature B in the second image, then feature B from the second image should also be the best match for feature A in the first image. Additionally, similarity was computed using the dot product (cosine similarity) of the descriptors, and thresholds were set at 0.8 for ratio-based filtering and 0.7 for distance-based filtering. Further details on the default implementations can be found in the HLOC repository[7]. For the handcrafted feature extractors, brute-force matching was employed with the corresponding distance measurement to be used in the matching process, such as L2 distance for SIFT and Hamming distance for ORB and AKAZE, and cross-check was enabled. Further details on the default implementations can be found in the OpenCV documentation [8].

[7] https://github.com/crcz25/Hierarchical-Localization

[8] https://docs.opencv.org/4.9.0/dc/dc3/tutorial_py_matcher.html

Table 3.3: Feature Extraction and Matching Combinations

Combinations	
Feature Extractor	**Feature Matcher**
AKAZE	NN-bruteforce
ORB	NN-bruteforce
SIFT	NN-bruteforce
SuperPoint	LightGlue
SuperPoint	SuperGlue Fast
SuperPoint	SuperGlue
SuperPoint	NN-Distance
SuperPoint	NN-Ratio
SuperPoint	NN-Mutual
R2D2	NN-Distance
R2D2	NN-Ratio
R2D2	NN-Mutual
D2-Net	NN-Distance
D2-Net	NN-Ratio
D2-Net	NN-Mutual
SOSNet	NN-Distance
SOSNet	NN-Ratio
SOSNet	NN-Mutual
SOSNet	AdaLAM
DISK	LightGlue
DISK	NN-Distance
DISK	NN-Ratio
DISK	NN-Mutual

3.3 Assessment Criteria

SIFT was employed as a baseline due to its prevalent use and documented effectiveness in diverse scenarios [12], [13], [20], [22], [24], [26]. Guided by insights from [57], [58], several procedural decisions were made for processing and assessing the 3D reconstructions produced by the tested methods:

- Feature extraction and matching were conducted using combinations of methods as detailed in Table 3.3.

- The retrieval of candidate images through a global search (NETVLAD) was capped at 50, based on findings by Sarlin, Cadena, Siegwart, *et al.* [58] indicating a significant impact of this parameter on reconstruction quality.
- A maximum of 8000 features were extracted per image, based on Jin, Mishkin, Mishchuk, *et al.* [57] that highlights the influence of feature quantity on matching accuracy. It is noted that some algorithms, such as SuperPoint, have a feature extraction limit (e.g., 4096 features).
- Image resizing for deep learning methods was based on a maximum dimension of 1024 pixels to preserve the original aspect ratio.
- The model that registered the highest number of images was selected for detailed reconstruction quality evaluation.

The evaluations employed to assess the quality of the 3D reconstructions were divided into three areas: *Dataset Evaluation*, *Reconstruction Evaluation*, and *Performance Evaluation*. Dataset Evaluation, aims to analyze the quality of the dataset and its capacity for feature matching. Reconstruction Evaluation, assesses the quality of the 3D reconstructions produced. Performance Evaluation, evaluates the resource usage. The metrics used in each evaluation are detailed in the following sections.

Dataset Evaluation

The initial phase of the evaluation involved analyzing the quality of the dataset and its capacity for feature matching. Co-visibility, a metric that quantifies the extent of scene overlap between image pairs, was employed to gauge the matching complexity based on the prevalence of shared features [57]. For each image pair in the dataset, let $\{K_i, K_j\}$ represent the keypoints in images I_i and I_j, respectively. The set of co-visible points, K_{ij}, is defined as $K_i \cap K_j$. Subsequently, for these co-visible points, 2D bounding boxes, B_i and B_j, are computed in each image, with areas $|B_i|$ and $|B_j|$, respectively. The visibility

of co-visible points within each image is calculated as the ratio of the bounding box area to the total image area. The visibility ratio for each image is defined as,

$$V_i = \frac{|B_i|}{|I_i|}, V_j = \frac{|B_j|}{|I_j|} \tag{3.1}$$

Finally, the co-visibility metric between the two images is defined as the minimum of these two visibility ratios,

$$Co - visibilityRatio = \min(V_i, V_j) \tag{3.2}$$

The following figure illustrates the co-visibility ratio.

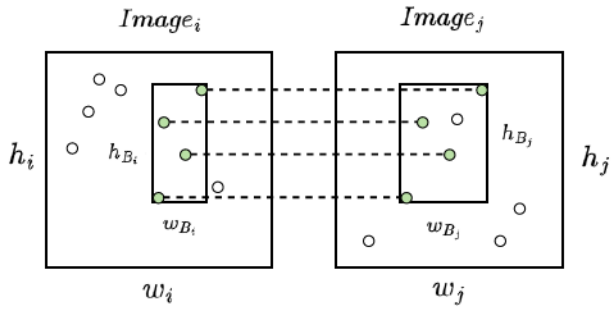


Figure 3.9: Co-visibility Ratio Example

To assess dataset quality, $\frac{n(n-1)}{2}$ images pairs were evaluated, and box plots of co-visibility ratios were generated. Scenes characterized by a higher proportion of pairs with low co-visibility scores are deemed more challenging due to the reduced number of overlapping features between images.

Reconstruction Evaluation

To assess the quality of the 3D reconstructions, a comparative analysis against SIFT' baseline was performed. The process involved aligning and registering the evaluated 3D point cloud with the baseline point cloud using the Iterative Closest Point (ICP) algorithm as implemented in Cloud Compare[9]. The metric, Cloud-to-Cloud error distance, was

[9] https://www.cloudcompare.org/doc/wiki/index.php/ICP

determined using the software Cloud Compare[10], too. Additional metrics for evaluating the 3D reconstructions were obtained from COLMAP[11]. The metrics employed in this evaluation are outlined as follows:

- **Cloud-to-Cloud error distance**. This metric quantifies the nearest neighbor distance, representing the shortest distance from any point in the source cloud to its nearest counterpart in the target cloud. In Cloud Compare, first a surface is approximated by a mathematical model (a Quadric model) on the local region of the target cloud. Then, the distance is calculated by finding the nearest point on the model (surface) to the source point. The Quadric Model is mathematically defined in Cloud Compare as follows with a default setting of $k = 6$ neighbors:

$$Z = aX^2 + bY^2 + cXY + dX + eY + f \tag{3.3}$$

  where Z is the distance between the source and target points, X and Y are the coordinates of the source points, and a, b, c, d, e, f are the coefficients of the model. After all distances have been calculated, the standard deviation and the mean value are calculated and used to report this metric.

- **Number of Registered Images**. The total number of images that were successfully registered in the reconstruction.

- **Number of Points**. The total number of 3D points in the reconstruction that are part of a 3D point track.

- **Number of Observations**. Total number of observations of 3D points across all registered images. It is calculated as,

$$N_{obs} = \sum_{i=1}^{n} NumPoints3D(I_i) \tag{3.4}$$

[10] https://www.cloudcompare.org/doc/wiki/index.php/Cloud-to-Cloud_Distance

[11] https://github.com/colmap/colmap/blob/main/src/colmap/scene/reconstruction.cc#L636

where $NumPoints3D(I_i)$ is the number of 3D points observed in the image I_i.

- **Mean Track Length**. The average number of images that observe each 3D point. It is calculated as,

$$MeanTrackLength = \frac{N_{obs}}{m} \tag{3.5}$$

where m is the number of unique 3D points in the reconstruction, and N_{obs} is the total number of observations of 3D points across all registered images.

- **Mean Observations per registered image**. The average number of 3D point observations per registered image. It is calculated as,

$$MeanObsPerImage = \frac{N_{obs}}{n} \tag{3.6}$$

where n is the number of registered images, and N_{obs} is the total number of observations of 3D points across all registered images.

- **Mean Reprojection Error**. The average reprojection error across all 3D points that have a recorded error in pixels. It is calculated as,

$$MeanReprojectionError = \frac{\sum_{i=1}^{m} Error(i)}{m} \tag{3.7}$$

where m is the number of 3D points with a recorded error, and $Error(i)$ is the reprojection error in pixels of the 3D point i.

- **Avg. Number of Keypoints**. The average number of keypoints extracted from the images.

- **Avg. Number of Matches**. The average number of valid matches found between pairs of images.

Performance Evaluation

When executing the 3D reconstruction process, the following metrics were collected during the execution, or at the end of it, to evaluate the resource usage. The metrics employed

in this evaluation are outlined as follows:

- **Elapsed Time**. The total time taken to process the images and generate the 3D reconstruction, calculated using the Linux command `time`.
- **Avg. Runtime Feature Extraction**. The average time taken to extract features per image.
- **Avg. Runtime Feature Matching**. The average time taken to match features per image pair.
- **Avg. Runtime Global Search**. The average time taken to retrieve the candidate images using NETVLAD.
- **CPU Usage**. The percentage of CPU usage during the process, calculated using the Linux command `time`. In a multicore system, the percentage can be higher than 100%. It is calculated as,

$$\frac{\text{Total CPU-seconds in user mode} + \text{Total CPU-seconds in kernel mode}}{\text{Elapsed real time}} \tag{3.8}$$

  For example, if the CPU is fully utilized, the usage on this system would be $48 * 100 = 4800\%$. To correct this, the CPU usage was reported by the command was divided by (48) the total number of threads in the system.
- **RAM Usage**. The amount of RAM used during the process, calculated using the Linux command `time`. It is measured as the maximum resident set size of the process, that is, the maximum amount of memory the process used during its execution.
- **GPU Usage**. The average percentage of GPU usage during the process, calculated using the Linux command `nvidia-smi`.
- **GPU Memory Usage**. The maximum amount of GPU memory used during the process, calculated using the Linux command `nvidia-smi`.

- **Disk Usage**. The amount of disk space used after the process has been completed, calculated using the python library `OS`.

3.4 Summary

This chapter provided a comprehensive overview of the methodology, datasets, implementation, and evaluation criteria used for the 3D reconstruction process. For datasets, three sources are utilized: two public (Hilti SLAM Challenge Dataset and ConSLAM) and one proprietary dataset. Each dataset's characteristics and challenges, such as lightning conditions and feature scarcity, are discussed, along with the data collection methods involved in each case. The hardware and software implementations are outlined, emphasizing the use of the Hierarchical Localization Toolbox (HLOC) and COLMAP for the reconstruction process. Different combinations of feature extraction and matching methods are listed, highlighting the flexibility and adaptability of the pipeline. The evaluation criteria are detailed, including metrics for dataset evaluation, reconstruction quality, and performance. The next chapter presents the results of the 3D reconstruction process using the datasets and evaluation criteria outlined in this chapter.

4 Results

In this chapter, the evaluation results are presented, structured into two primary sections: indoor scenes and outdoor scenes. Each section conducts an in-depth analysis of the datasets utilized, detailing the reconstruction outcomes and performance assessments for various combinations of feature extractors and matching algorithms. These results facilitate the analysis and interpretation of the effectiveness of different methods in varied environments, providing a clear understanding of their respective strengths and limitations. The term "ConSLAM" will be used to refer to the *ConSLAM — Sequence 2*, "Hilti" refers to *Hilti — Construction Upper Level 1* or *Hilti — Construction Site Outdoor* (depending on the section), and "Private" denotes *Private — Construction Site Outdoor*.

4.1 Indoor Scenes

4.1.1 Dataset Evaluation

Figures 4.2 and 4.1 illustrate the box plots of co-visibility ratios for different datasets, highlighting the variability and performance of various methods. Table 4.1 provides comprehensive descriptive statistics, including count, mean, standard deviation, and percentile values for the co-visibility ratios.

ConSLAM

High Co-visibility Performance (0.8-1.0). Methods demonstrating high performance are predominantly deep-learning approaches. SIFT, as the baseline method, achieved a mean co-visibility ratio of 0.88, underscoring the robustness of traditional feature extraction techniques. However, deep learning methods such as R2D2 and D2-Net using NN-Mutual and NN-Distance algorithms outperform SIFT, with mean co-visibility ratios exceeding 0.9. These top-performing methods deliver superior average performance and exhibit low standard deviations, indicating consistent scene visibility overlap across image pairs.

Moderate Co-visibility Performance (0.4-0.8). Methods in this performance range include a mix of traditional and deep-learning approaches. Among traditional methods, the best classified in this range is AKAZE with a mean co-visibility ratio of 0.79. In contrast, deep learning methods, DISK and SuperPoint both with NN-Distance, obtained mean ratios of 0.53 and 0.47.

Low Co-visibility Performance (0-0.4). The methods in this category struggled to maintain reliable scene overlap, resulting in lower mean co-visibility ratios. For example, some deep learning methods utilizing matching techniques like AdaLAM or tailored approaches like LightGlue, fall within this performance range. Moreover, methods with extreme outliers on the high side fall in this range with matching techniques like NN-Ratio and NN-Distance. Compared to the baseline, these methods are significantly less effective, underscoring the superiority of both SIFT and other higher-performing deep learning methods in achieving reliable reconstruction outcomes.

Outliers and Extremes. From the box plots of Figure 4.1, it is notable that certain methods exhibit unusually high maximum co-visibility ratios. For example, R2D2 (NN-Ratio) and D2-Net (NN-Ratio) achieve maximum ratios close to 0.99 and 0.97, respec-

tively, indicating their potential to achieve near-perfect overlap in optimal scenarios. On the other hand, some methods like DISK (NN-Distance) and SuperPoint (NN-Ratio) show higher variability and extreme values, with maximum ratios reaching up to 0.971 and 0.953, respectively, but also display higher standard deviations, suggesting inconsistent performance across different image pairs.

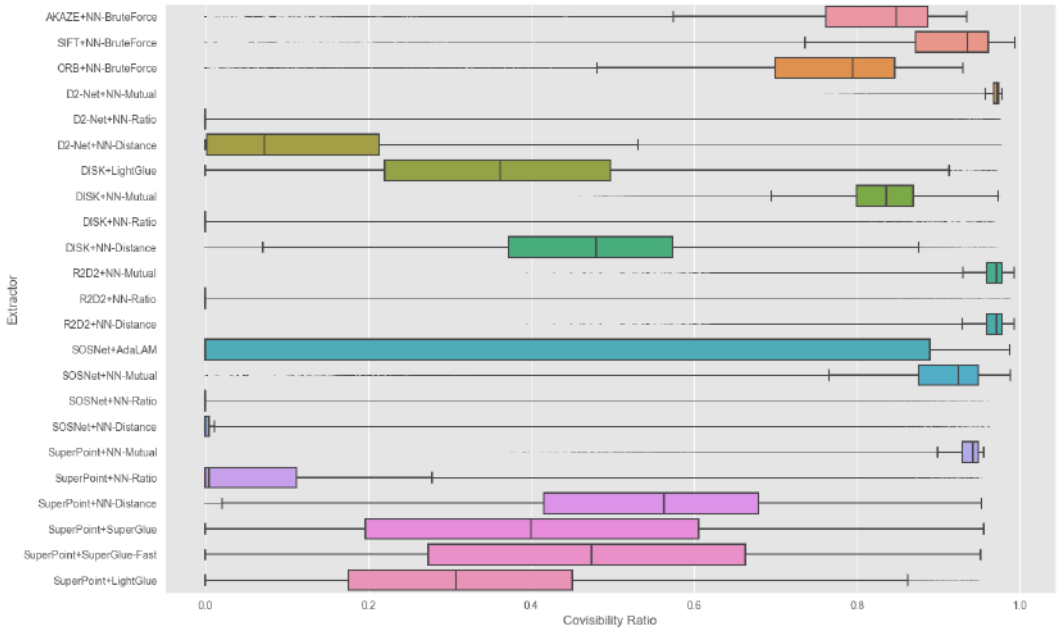


Figure 4.1: ConSLAM: Sequence 2 Co-visibility Ratios

Hilti

High Co-visibility Performance (0.8-1.0). Techniques such as R2D2 (both NN-Distance and NN-Mutual), D2-Net (NN-Mutual), and SuperPoint (NN-Mutual) significantly surpass the baseline. These approaches exhibit outstanding performance, with mean co-visibility ratios approaching or exceeding 0.9 and minimal variability. This indicates their superior efficacy in identifying and matching image pairs.

Moderate Co-visibility Performance (0.4-0.8). Within this range, SIFT and ORB demonstrate similar results, both achieving a mean co-visibility ratio of approximately 0.5 with

a very close variability. Despite this, the baseline (and traditional methods) still fall below the deep learning methods in this range. For example, DISK (NN-Mutual) surpasses traditional techniques, whereas others, such as SOSNet (AdaLAM), perform comparably or slightly worse.

Low Co-visibility Performance (0-0.4). Most deep learning methods, along with AKAZE, demonstrate mean co-visibility ratios significantly lower than the baseline. These findings underscore the difficulties these methods encounter in scenarios characterized by minimal overlap of visual features.

Outliers and Extremes. Similar to ConSLAM, certain approaches exhibit extreme behaviors or outliers in their performance metrics, as shown in the box plots of Figure 4.2. For example, R2D2 (NN-Distance and NN-Mutual) achieves cases with the highest mean ratios and does so with remarkably low standard deviations, even the combination NN-Ratio that has an average mean of close to zero, making it an outlier in terms of both high performance and consistency. In the case of SuperPoint, even though with the tailored matching methods show low performance, it has maximum ratios reaching up to 0.96, indicating a capability to match under difficult circumstances.

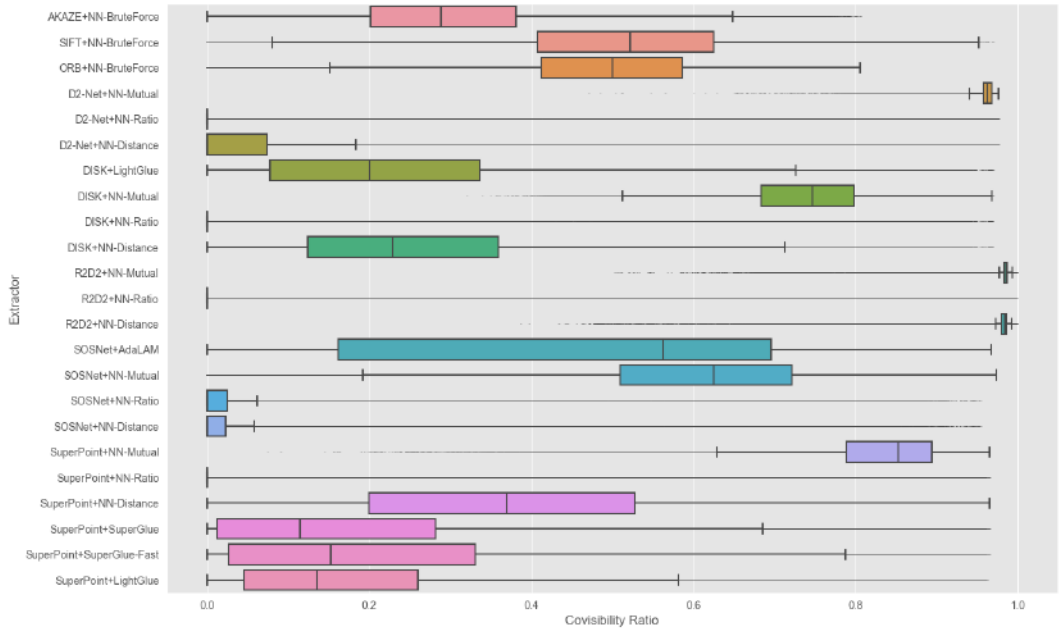


Figure 4.2: Hilti: Construction Upper Level 1 Co-visibility Ratios

Table 4.1: Descriptive Statistics of Co-visibility Ratios for Indoor Scenes

		ConSLAM - Sequence 2								Hilti - Construction Upper Level 1							
Extractor	**Matcher**	**count**	**mean**	**std**	**min**	**25%**	**50%**	**75%**	**max**	**count**	**mean**	**std**	**min**	**25%**	**50%**	**75%**	**max**
AKAZE	NN-BruteForce	304,590	0.79	0.16	0.00	0.76	0.85	0.89	0.93	456,490	0.29	0.14	0.00	0.20	0.29	0.38	0.81
SIFT	NN-BruteForce	305,371	0.88	0.15	0.00	0.87	0.94	0.96	0.99	474,825	0.50	0.19	0.00	0.41	0.52	0.63	0.97
ORB	NN-BruteForce	304,590	0.75	0.15	0.00	0.70	0.79	0.85	0.93	475,800	0.49	0.14	0.00	0.41	0.50	0.59	0.80
D2-Net	NN-Mutual	306,153	**0.97**	**0.01**	0.76	0.97	0.97	0.97	0.98	475,800	0.96	**0.01**	0.47	0.96	0.96	0.97	0.98
D2-Net	NN-Ratio	306,153	0.02	0.10	-	-	-	0.00	0.97	475,800	0.01	0.09	-	-	-	-	0.98
D2-Net	NN-Distance	306,153	0.15	0.21	-	0.00	0.07	0.21	0.98	475,800	0.10	0.21	-	-	0.00	0.07	0.98
DISK	LightGlue	306,153	0.36	0.18	-	0.22	0.36	0.50	0.97	475,800	0.22	0.17	-	0.08	0.20	0.34	0.97
DISK	NN-Mutual	306,153	0.83	0.06	0.46	0.80	0.84	0.87	0.98	475,800	0.74	0.08	0.32	0.68	0.75	0.80	0.97
DISK	NN-Ratio	306,153	0.02	0.08	-	-	-	-	0.97	475,800	0.01	0.08	-	-	-	-	0.97
DISK	NN-Distance	306,153	0.47	0.15	0.00	0.37	0.48	0.57	0.97	475,800	0.25	0.17	0.00	0.12	0.23	0.36	0.97
R2D2	NN-Mutual	306,153	0.96	0.03	0.39	0.96	0.97	0.98	0.99	475,800	**0.98**	0.02	0.50	0.98	0.99	0.99	1.00
R2D2	NN-Ratio	306,153	0.02	0.09	-	-	-	-	0.99	475,800	0.02	0.10	-	-	-	-	1.00
R2D2	NN-Distance	306,153	0.96	0.03	0.39	0.96	0.97	0.98	0.99	475,800	0.98	0.03	0.39	0.98	0.98	0.99	1.00
SOSNet	AdaLAM	306,153	0.35	0.42	-	-	-	0.89	0.99	475,800	0.47	0.29	-	0.16	0.56	0.70	0.97
SOSNet	NN-Mutual	306,153	0.87	0.14	0.00	0.88	0.92	0.95	0.99	474,825	0.59	0.18	0.00	0.51	0.63	0.72	0.97
SOSNet	NN-Ratio	306,153	0.03	0.11	-	-	-	0.00	0.96	83,626	0.07	0.18	0.00	0.00	0.00	0.02	0.95
SOSNet	NN-Distance	306,153	0.04	0.12	-	-	0.00	0.00	0.96	109,588	0.06	0.16	0.00	0.00	0.00	0.02	0.95
SuperPoint	NN-Mutual	306,153	0.93	0.04	0.37	0.93	0.94	0.95	0.96	475,800	0.83	0.09	0.08	0.79	0.85	0.89	0.96
SuperPoint	NN-Ratio	306,153	0.09	0.15	-	0.00	0.00	0.11	0.95	475,800	0.02	0.10	-	-	-	0.00	0.96
SuperPoint	NN-Distance	306,153	0.53	0.20	-	0.42	0.56	0.68	0.95	475,800	0.37	0.21	-	0.20	0.37	0.53	0.96
SuperPoint	SuperGlue	306,153	0.40	0.25	-	0.20	0.40	0.61	0.96	475,800	0.18	0.19	-	0.01	0.11	0.28	0.96
SuperPoint	SuperGlue-Fast	306,153	0.46	0.25	-	0.27	0.47	0.66	0.95	475,800	0.21	0.20	-	0.03	0.15	0.33	0.96
SuperPoint	LightGlue	306,153	0.32	0.18	-	0.18	0.31	0.45	0.95	475,800	0.17	0.16	-	0.05	0.14	0.26	0.96

4.1.2 Reconstruction Evaluation

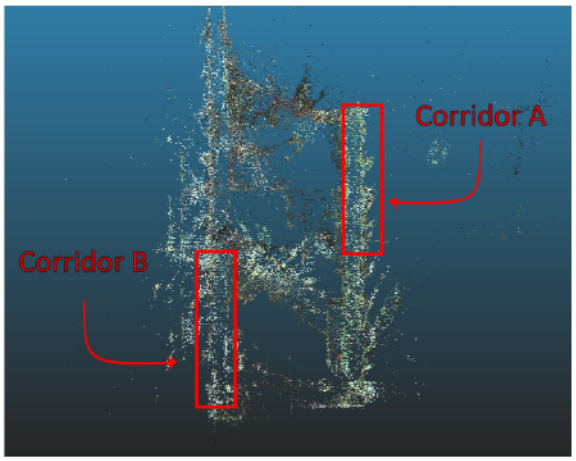
Table 4.2 presents the reconstruction results, detailing the number of registered images, points, observations, mean track length, mean observations per image, mean reprojection error, mean number of keypoints, and mean number of matches. A visual examination of the reconstructed point clouds offers insights into the quality and completeness of the reconstructions can be found in Figures 4.6 to 4.11 for ConSLAM and from 4.13 to 4.18 for Hilti, which display the generated point clouds from a front isometric view.

Table 4.2: Reconstruction Results for Indoor Scenes

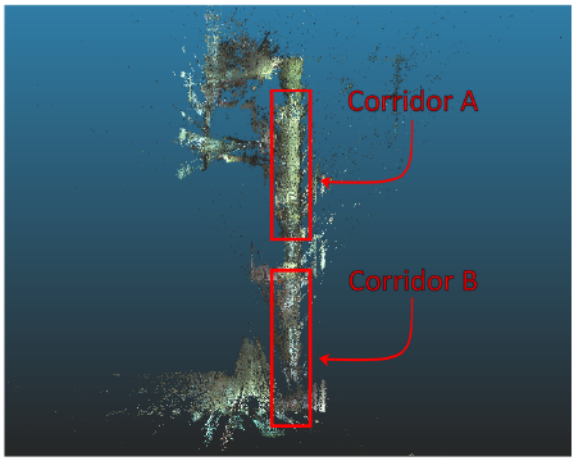
		ConSLAM - Sequence 2								Hilti - Construction Upper Level 1							
Extractor	**Matcher**	**# Reg. Img.**	**# Points**	**# Obs.**	**Mean Track Length**	**Mean Obs. Per Img**	**Mean Reproj. Error**	**Mean # Keypoints**	**Mean # Matches**	**# Reg. Img.**	**# Points**	**# Obs.**	**Mean Track Length**	**Mean Obs. Per Img**	**Mean Reproj. Error**	**Mean # Keypoints**	**Mean # Matches**
AKAZE	NN-BruteForce	598	162,116	1,278,721	7.89	2138.33	1.07	4694.11	1379.82	321	13,797	90,506	6.56	281.95	1.16	458.68	144.55
SIFT	NN-BruteForce	605	157,949	907,951	5.75	1500.75	**0.71**	4107.17	1023.93	452	24,355	130,712	5.37	289.19	0.80	580.85	162.72
ORB	NN-BruteForce	768	217,321	2,289,941	**10.54**	2981.69	1.12	7485.89	**1833.36**	953	87,726	749,104	8.54	786.05	1.10	2140.84	551.43
D2-Net	NN-Mutual	780	364,527	1,557,653	4.27	1996.99	1.50	4442.13	1054.82	957	216,417	1,308,981	6.05	1367.80	1.38	2312.98	579.80
D2-Net	NN-Ratio	336	46,033	226,822	4.93	675.07	1.46		90.70	43	3,871	48,141	12.44	1119.56	1.13		91.12
D2-Net	NN-Distance	780	261,884	1,201,174	4.59	1539.97	1.53		314.36	651	79,271	552,331	6.97	848.43	1.41		218.24
DISK	LightGlue	**783**	**536,211**	**3,780,568**	7.05	**4828.31**	1.35	6568.80	1122.41	920	288,775	3,005,325	10.41	3266.66	0.99	4128.47	775.60
DISK	NN-Mutual	**783**	469,707	3,378,905	7.19	4315.33	1.29		1029.01	967	**309,370**	2,799,201	9.05	2894.73	0.88		866.34
DISK	NN-Ratio	772	263,240	2,155,316	8.19	2791.86	1.16		373.46	704	136,753	1,421,581	10.40	2019.29	0.76		454.50
DISK	NN-Distance	772	344,770	2,661,758	7.72	3447.87	1.24		547.49	673	115,683	1,199,493	10.37	1782.31	0.64		404.79
R2D2	NN-Mutual	**783**	349,930	3,553,683	10.16	4538.55	1.44	**7995.16**	810.12	956	273,521	**3,364,291**	12.30	**3519.13**	0.91	**6618.23**	**878.93**
R2D2	NN-Ratio	485	122,945	1,104,995	8.99	2278.34	1.26		202.57	**974**	243,966	3,217,281	13.19	3303.16	0.90		767.43
R2D2	NN-Distance	**783**	347,882	3,539,670	10.17	4520.65	1.44		804.78	233	34,962	510,494	14.60	2190.96	0.67		349.81
SOSNet	Adalam	481	38,770	248,663	6.41	516.97	1.09	1028.13	148.30	576	21,161	118,795	5.61	206.24	0.92	290.41	97.47
SOSNet	NN-Mutual	477	43,176	254,733	5.90	534.03	1.07		341.13	121	2,652	18,944	7.14	156.56	0.76		25.98
SOSNet	NN-Ratio	474	29,302	201,975	6.89	426.11	1.02		78.60	66	2,555	10,310	4.04	156.21	0.66		65.81
SOSNet	NN-Distance	647	41,542	280,855	6.76	434.09	1.04		79.60	55	1,251	6,153	4.92	111.87	0.62		25.47
SuperPoint	NN-Mutual	713	102,468	665,417	6.49	933.26	1.34	1532.02	530.73	757	37,866	319,856	8.45	422.53	1.06	509.43	98.37
SuperPoint	NN-Ratio	648	56,062	454,708	8.11	701.71	1.30		115.75	763	38,158	310,980	8.15	407.58	1.05		95.37
SuperPoint	NN-Distance	780	93,849	679,938	7.25	871.72	1.36		203.65	689	37,469	260,311	6.95	377.81	0.99		162.16
SuperPoint	SuperGlue	782	124,437	830,524	6.67	1062.05	1.39		293.23	705	46,798	261,746	5.59	371.27	1.17		100.28
SuperPoint	SuperGlue-Fast	781	123,910	823,766	6.65	1054.76	1.39		305.16	260	10,022	90,903	9.07	349.63	0.97		81.25
SuperPoint	LightGlue	781	122,568	825,248	6.73	1056.66	1.39		286.78	43	698	12,696	**18.19**	295.26	**0.57**		46.49

ConSLAM

Corridor Misalignment We observed a reconstruction error in this dataset during the evaluation and manual inspection of the point clouds. Figure 4.3 illustrates the misalignment in the map using the colored point cloud generated by R2D2 with NN-Distance and SuperPoint and SuperGlue-Fast. The map has two distinct opposite corridors, Corridor A and Corridor B, which should not be interconnected as depicted in sub-figure 4.3a. However, both corridors were joined when reconstructing the map as a single corridor, as shown in sub-figure 4.3b. This error was observed consistently across all the methods except for the combination SuperPoint with SuperGlue-Fast, suggesting that the problem is likely related to incorrect feature matching and registration, resulting in an inaccurate map reconstruction. The error is attributed to the similarity of the images taken in those specific regions of the scene, leading to keypoints being mistakenly identified as valid matches between image pairs. Figure 4.4 demonstrates this with an image pair obtained from NETVLAD.



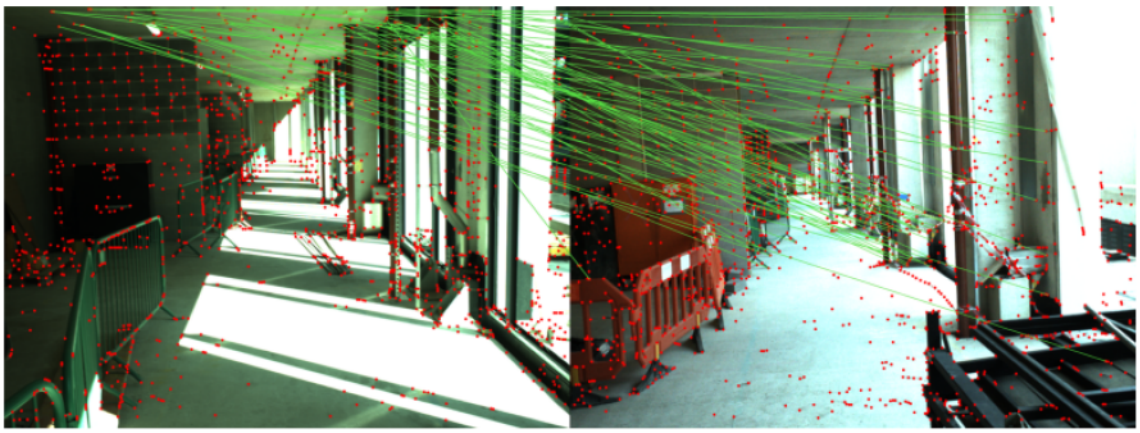
(a) Top view correct map reconstruction from SuperPoint+SuperGlue-Fast



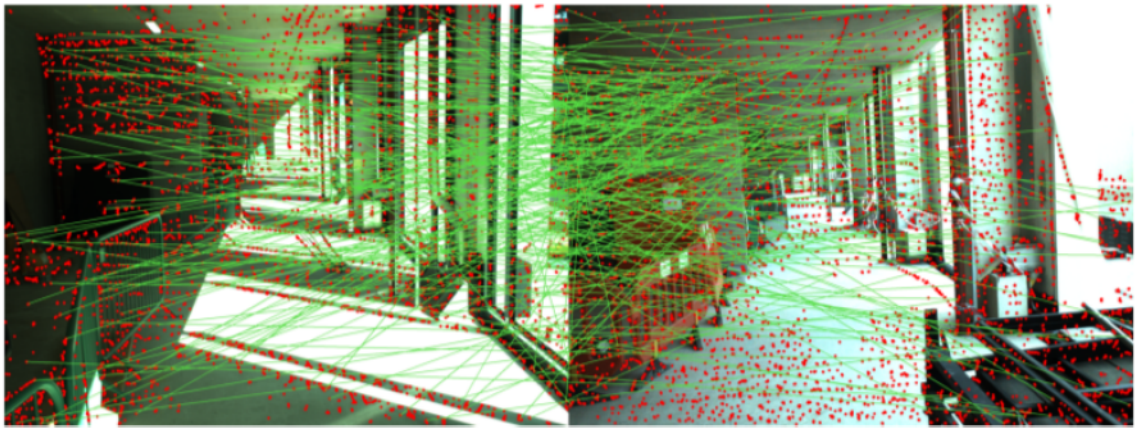
(b) Top view incorrect map reconstruction from R2D2+NN-Distance

(c) Images depicting the similarities between the Corridor A (left) and Corridor B (right).

Figure 4.3: Reconstruction Error in ConSLAM



(a) SuperPoint+SuperGlue-Fast — Matching



(b) R2D2+NN-Distance — Matching

Figure 4.4: Matching differences between the Corridors A (Left) and Corridor B (right)

Traditional Methods. SIFT, as the baseline method, successfully registered 605 images, generated 157,949 3D points, and accumulated a total of 907,951 observations. The average track length was 5.75, indicating a moderate frequency of image observations per 3D point. Furthermore, SIFT exhibited a low mean reprojection error, indicating high precision in the reconstructed 3D model. In contrast, AKAZE registered a slightly lower number of images but produced more 3D points and observations, resulting in a higher mean track length and a greater mean number of observations per image. However, the mean reprojection error increased, indicating a trade-off between the quantity of points

and the accuracy of the reconstruction. ORB, on the other hand, demonstrated superior performance among traditional methods, with the highest number of registered images, 3D points, and observations. This approach also achieved the highest mean track length among both traditional and deep learning-based methods. Despite excelling in most reconstruction metrics, ORB exhibited a higher mean reprojection error of 1.12 pixels, indicating a reduction in precision compared to SIFT.

Deep Learning-Based Methods. Deep Learning methods, present a similar range of outcomes in 3D reconstruction quality. Specifically, DISK (utilizing LightGlue and NN-Mutual) registers 783 images and captures an exceptionally high number of points (536,211 and 469,707 respectively). These also achieve a substantial number of observations (up to 3,780,568), underscoring their robustness in feature detection and matching. Nonetheless, despite these high figures, the reprojection errors for DISK (ranging from 1.29 to 1.35 pixels) indicate a slight reduction in precision relative to SIFT's. R2D2 (employing NN-Mutual and NN-Distance) also exhibits strong performance, registering the same number of images as DISK and exceeding 347,000 points. The mean reprojection errors are marginally higher than those of DISK, reflecting similar trends in precision and point density. Similarly, D2-Net, while effective in certain configurations, generally registers fewer images and points, particularly with the NN-Ratio matcher, where only 336 images are registered, suggesting limitations in robustness. SuperPoint with SuperGlue-Fast (as the only combination that reconstructed the map successfully), demonstrated a comparable number of registered images with fewer points and total observations, alongside a lower mean reprojection error. This highlights a trade-off between the volume of image registration and precision.

Figure 4.5 presents radar plots between the baseline and SuperPoint with SuperGlue-Fast. The values were scaled and translated into the range [0, 1] for improved visualization

using min-max scaling according to the following formula:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{4.1}$$

where x is the original value and x' is the scaled value. The minimum and maximum values were calculated separately for each column. The original values are provided in Table 4.2.

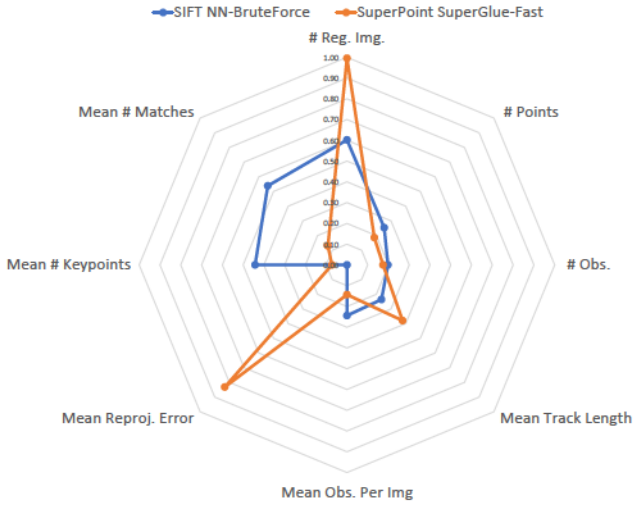
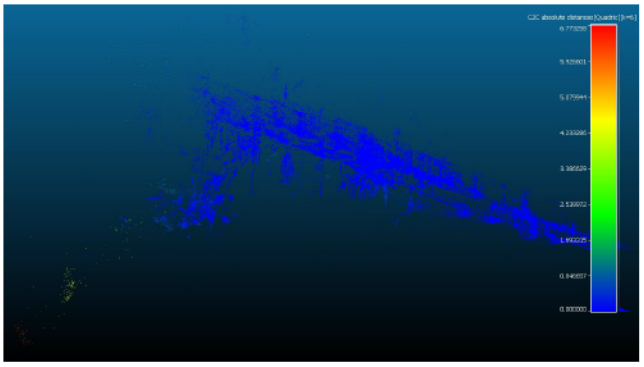
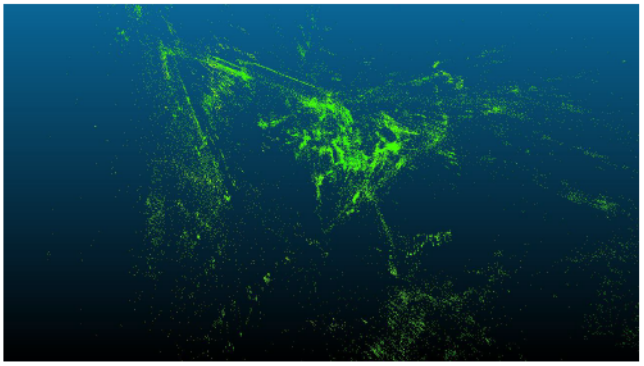


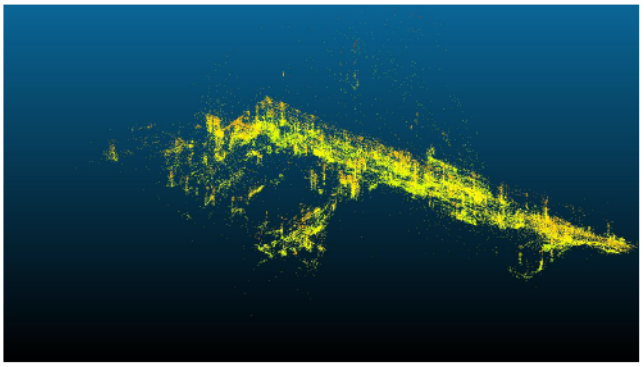
Figure 4.5: ConSLAM — Sequence 2: Reconstruction Comparison



(a) AKAZE+NN-BruteForce

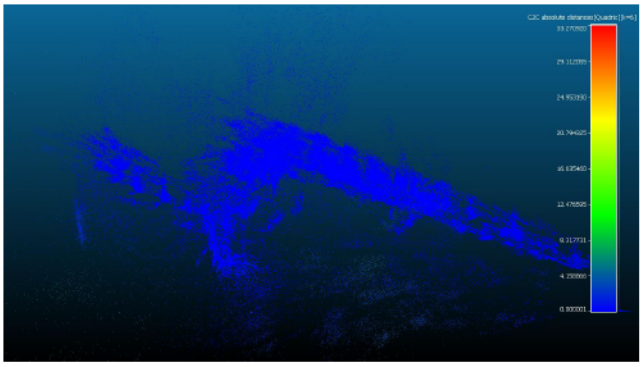


(b) ORB+NN-BruteForce

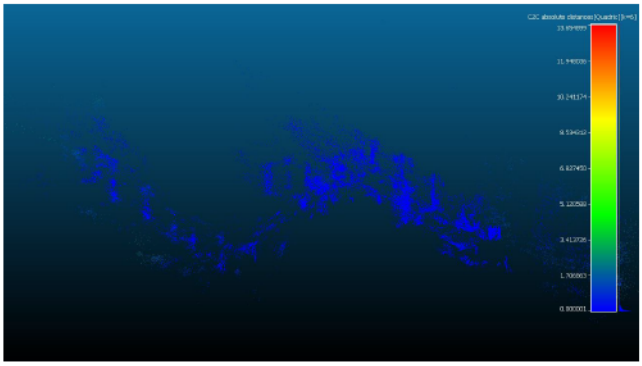


(c) SIFT+NN-BruteForce (Baseline)

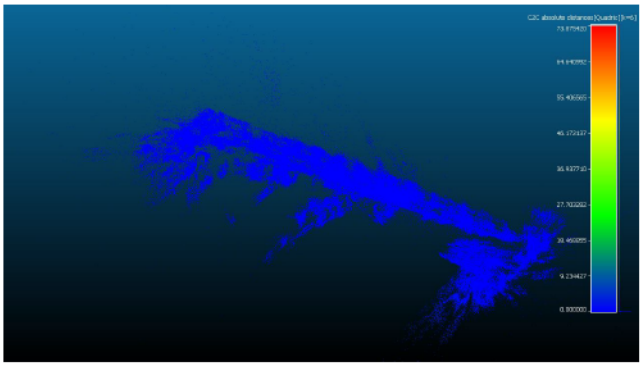
Figure 4.6: ConSLAM — Sequence 2: Traditional Reconstructions



(a) D2-Net+NN-Distance

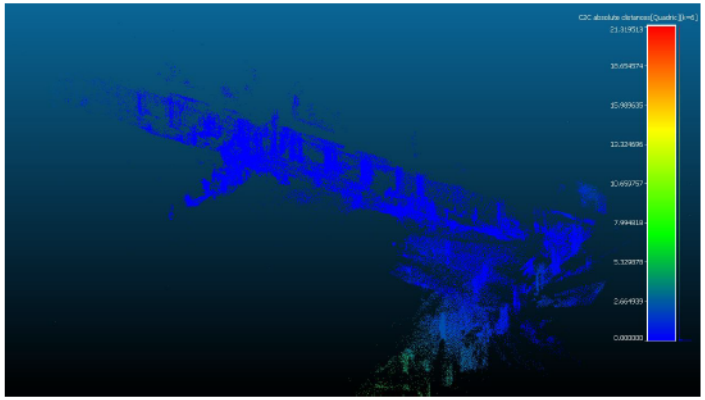


(b) D2-Net+NN-Ratio

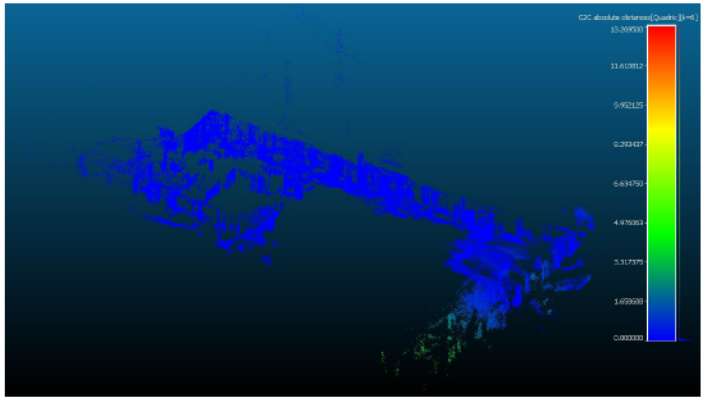


(c) D2-Net+NN-Mutual

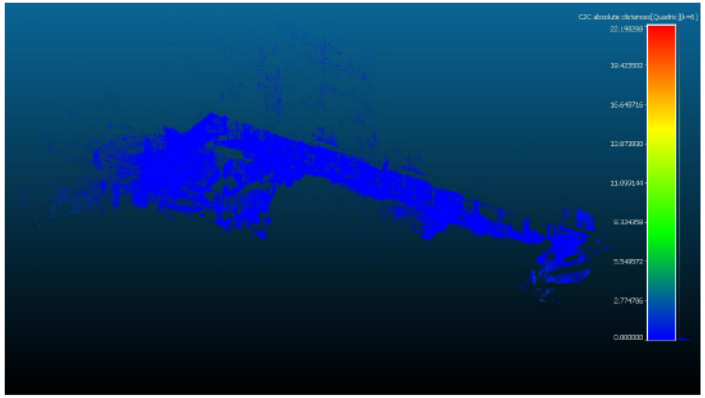
Figure 4.7: ConSLAM — Sequence 2: D2-Net Reconstructions



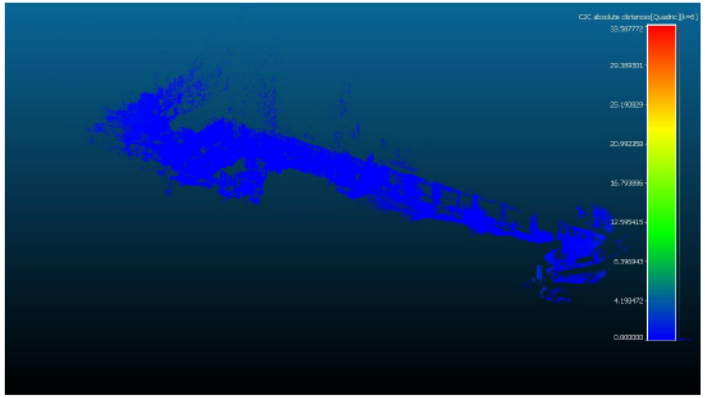
(a) DISK+NN-Distance



(b) DISK+NN-Ratio

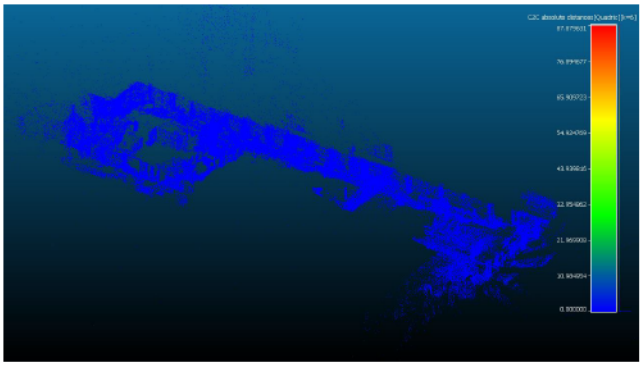


(c) DISK+NN-Mutual

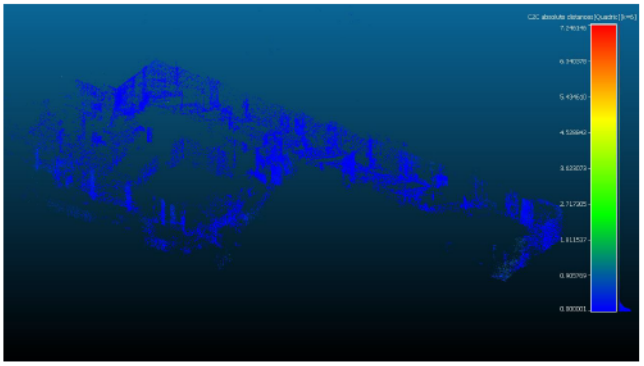


(d) DISK+LightGlue

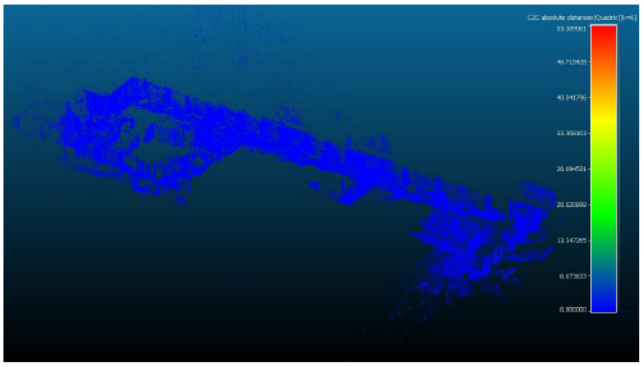
Figure 4.8: ConSLAM — Sequence 2: DISK Reconstructions



(a) R2D2+NN-Distance

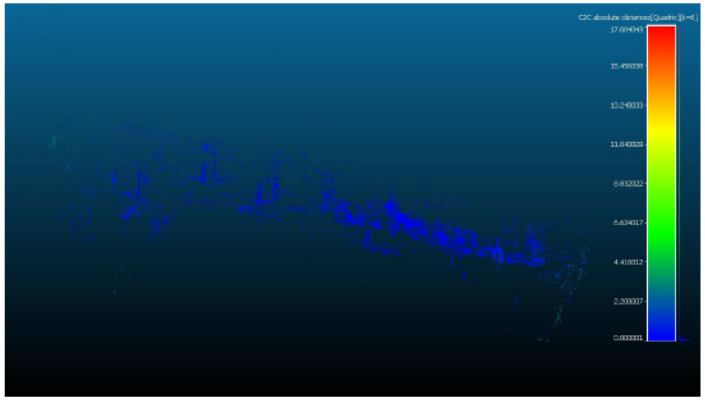


(b) R2D2+NN-Ratio

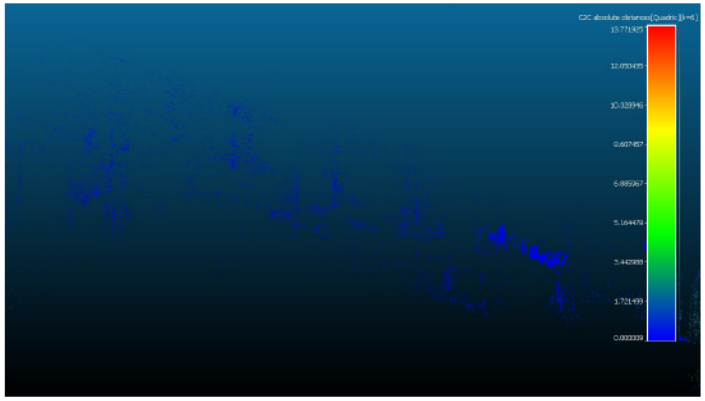


(c) R2D2+NN-Mutual

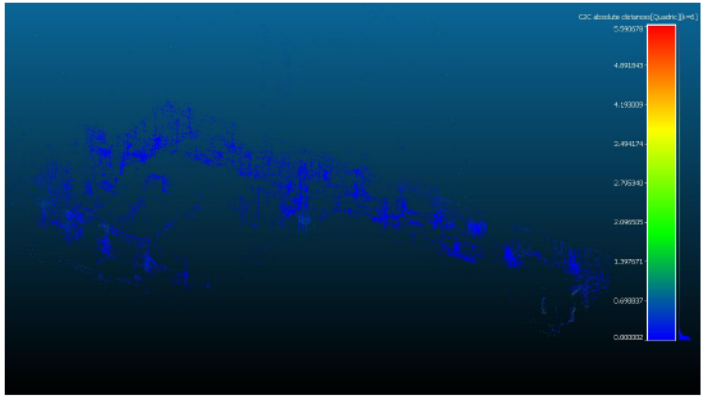
Figure 4.9: ConSLAM — Sequence 2: R2D2 Reconstructions



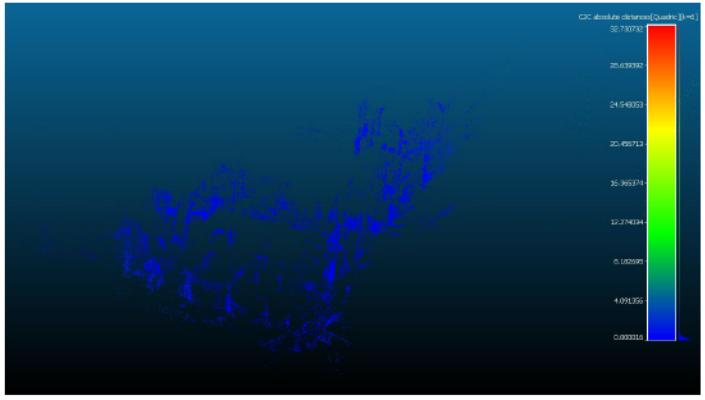
(a) SOSNet+NN-Distance



(b) SOSNet+NN-Ratio

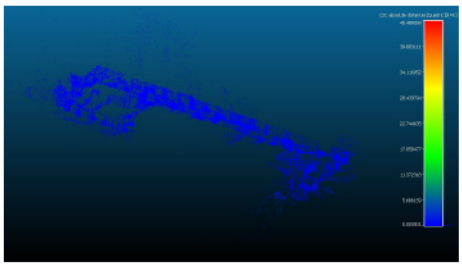


(c) SOSNet+NN-Mutual

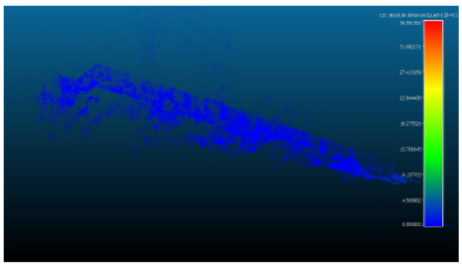


(d) SOSNet+AdaLAM

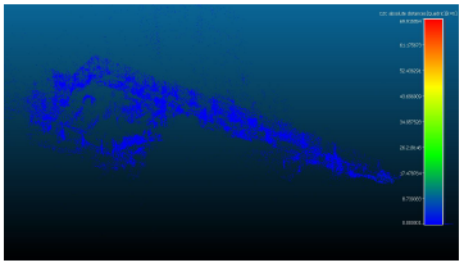
Figure 4.10: ConSLAM — Sequence 2: SOSNet Reconstructions



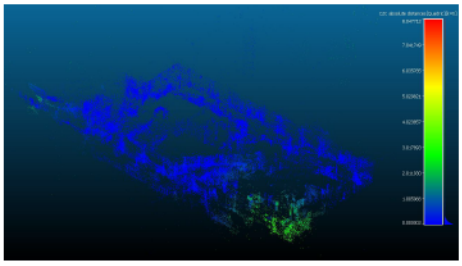
(a) SuperPoint+NN-Distance



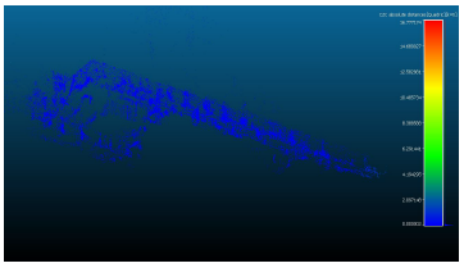
(b) SuperPoint+SuperGlue



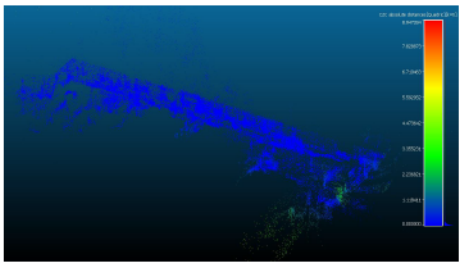
(c) SuperPoint+NN-Mutual



(d) SuperPoint+SuperGlue-Fast



(e) SuperPoint+NN-Ratio



(f) SuperPoint+LightGlue

Figure 4.11: ConSLAM — Sequence 2: SuperPoint Reconstructions

Hilti

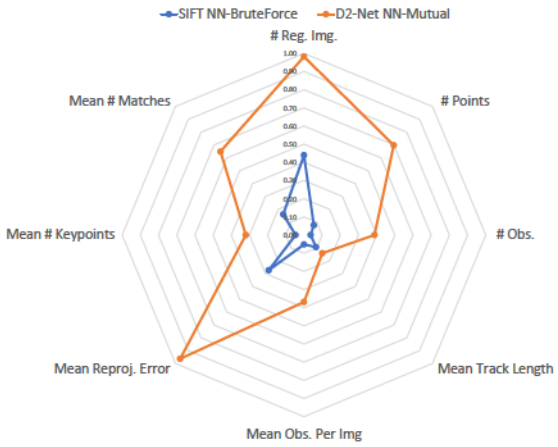
Traditional Methods. SIFT, as the baseline method, registers 452 images, generating a robust dataset with 24,355 points and 130,712 observations. It achieves a mean track length of 5.37, reflecting moderate feature persistence across images, and demonstrates a relatively low mean reprojection error, underscoring its precision in point localization. AKAZE, falls into the worst of the three traditional methods across all the metrics. In contrast, ORB significantly surpasses both in terms of registered images, number of points, and observations, indicating its efficiency in feature extraction and matching. However, its mean reprojection error is marginally higher at 1.10, suggesting a trade-off between quantity and precision. Regarding cloud-to-cloud error distances, AKAZE results in a mean distance of 0.33 with a standard deviation of 0.47, indicating a moderate level of precision, while no cloud-to-cloud data is available for ORB, leaving a gap in the comparative analysis.

Deep Learning-Based Methods. These approaches typically register more images and generate more 3D points and observations than baseline and traditional methods. For example, R2D2 with NN-Mutual, despite not registering the maximum number of images, achieves the highest number of observations, observations per image, keypoints, and matches among all methods, demonstrating superior feature extraction and tracking capabilities with a mean reprojection error of just 0.91. In terms of cloud-to-cloud error distances, deep learning methods display higher mean distances and standard deviations, indicating variability in precision. Notably, R2D2 with NN-Ratio achieves the lowest mean distance and standard deviation, reflecting the highest precision among all. These findings suggest that deep learning methods can achieve superior accuracy and precision in 3D reconstruction tasks compared to traditional techniques in such a complex dataset with occlusions and difficult light conditions where overexposure and underexposure are common.

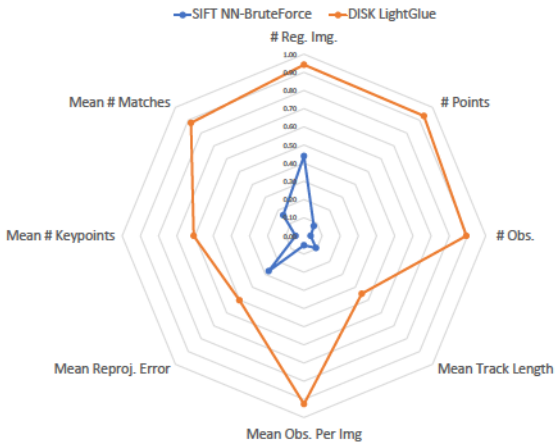
Figure 4.12 presents radar plots between the baseline and the top three deep learning methods (D2-Net, DISK, and R2D2), or in the case of ConSLAM only SuperPoint with SuperGlue-Fast, the reason will be explained in further sections. The values were scaled and translated into the range [0, 1] for improved visualization using min-max scaling according to the following formula:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{4.2}$$

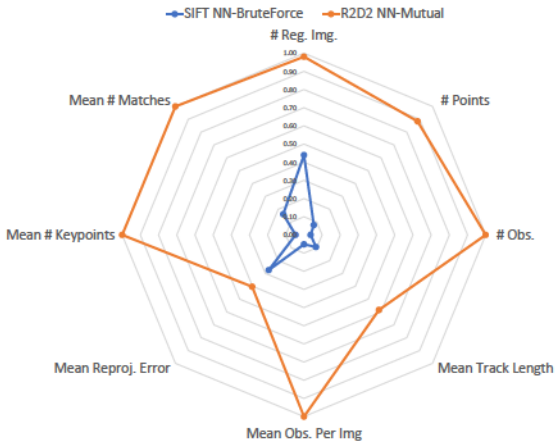
where x is the original value and x' is the scaled value. The minimum and maximum values were calculated separately for each column. The original values are provided in Table 4.2.



(a) D2-Net+NN-Mutual vs. Baseline

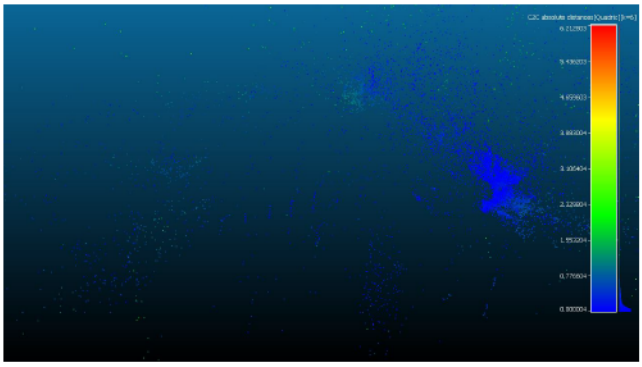


(b) DISK+LightGlue vs. Baseline

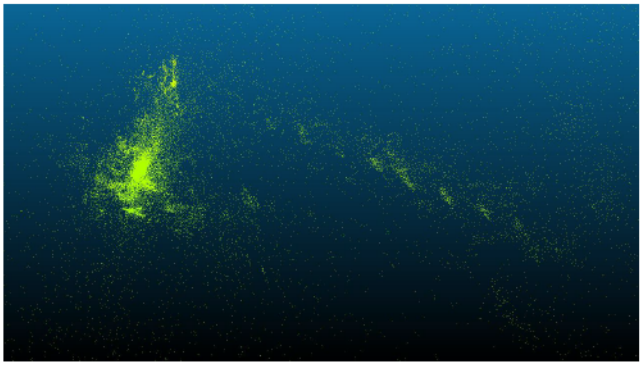


(c) R2D2+NN-Mutual vs. Baseline

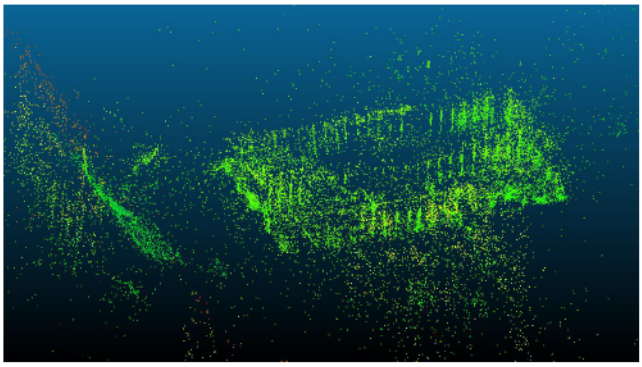
Figure 4.12: Hilti — Construction Upper Level 1: Reconstruction Comparison



(a) AKAZE+NN-BruteForce

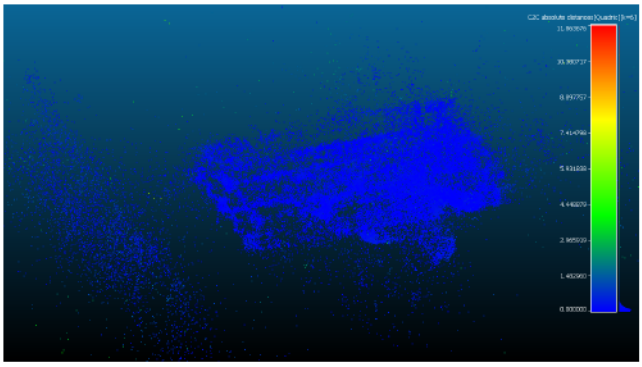


(b) ORB+NN-BruteForce

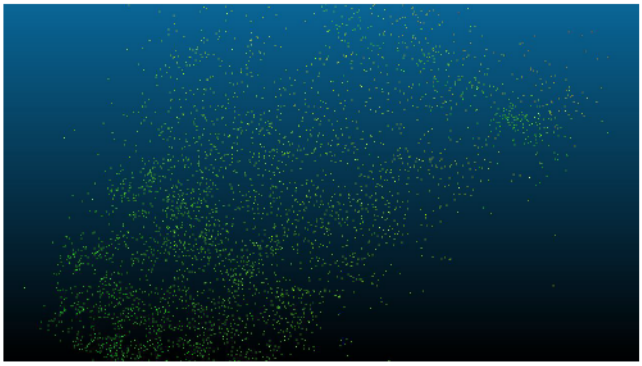


(c) SIFT+NN-BruteForce (Baseline)

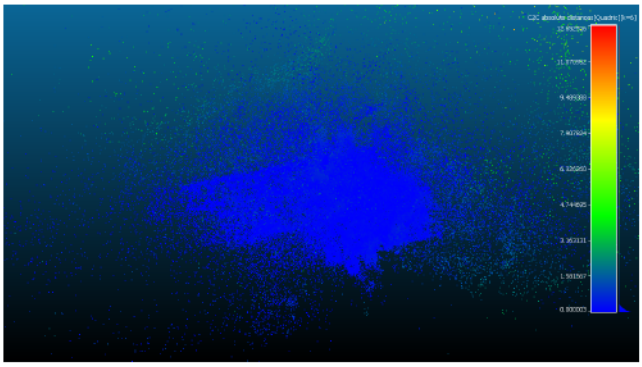
Figure 4.13: Hilti — Construction Upper Level 1: Traditional Reconstructions



(a) D2-Net+NN-Distance

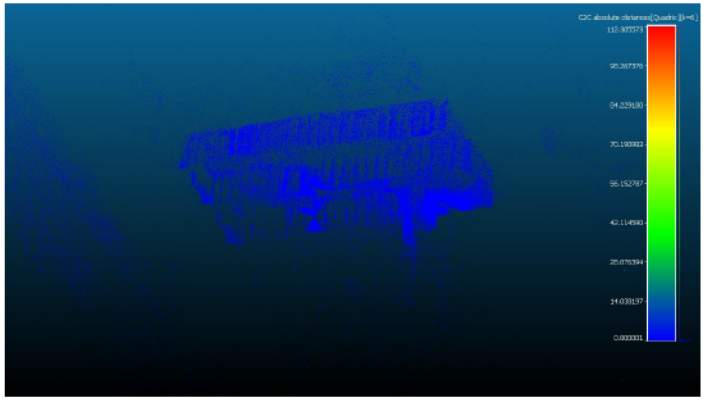


(b) D2-Net+NN-Ratio

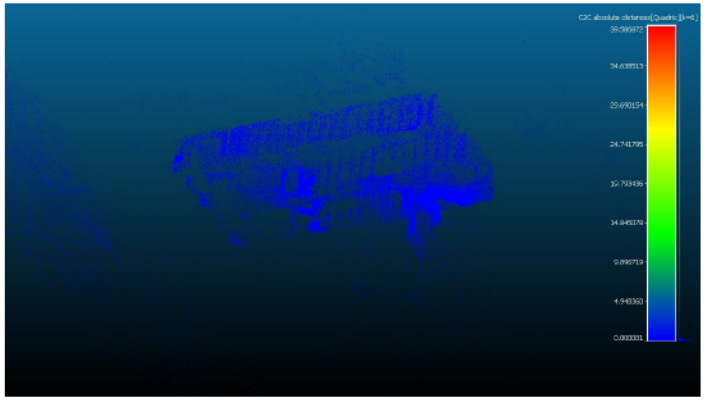


(c) D2-Net+NN-Mutual

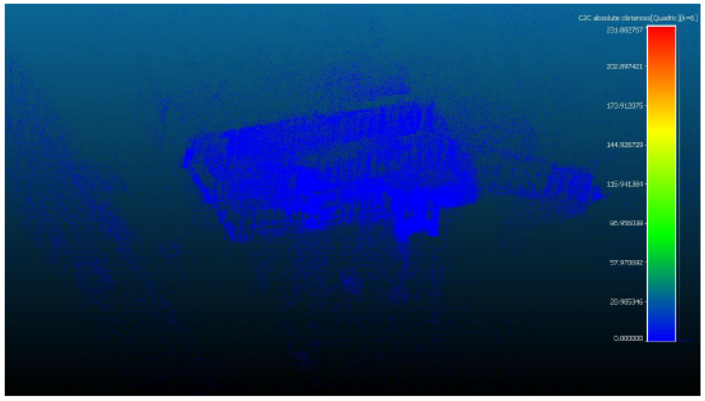
Figure 4.14: Hilti — Construction Upper Level 1: D2-Net Reconstructions



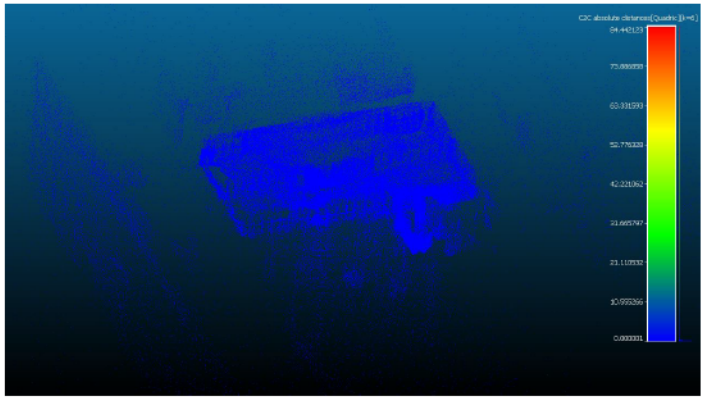
(a) DISK+NN-Distance



(b) DISK+NN-Ratio

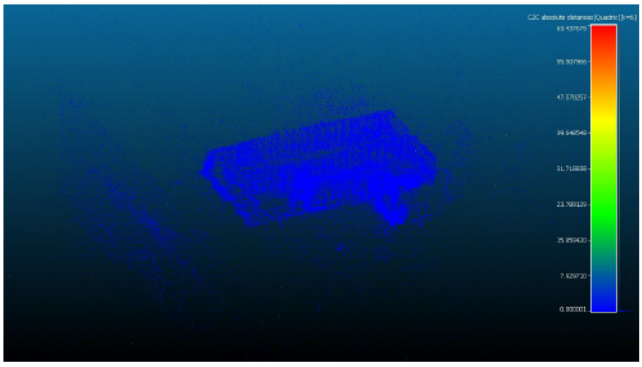


(c) DISK+NN-Mutual

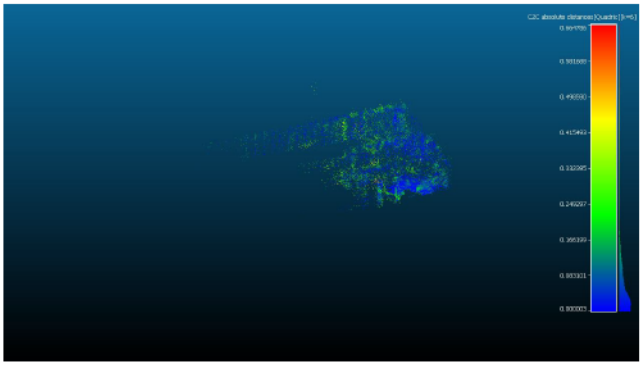


(d) DISK+LightGlue

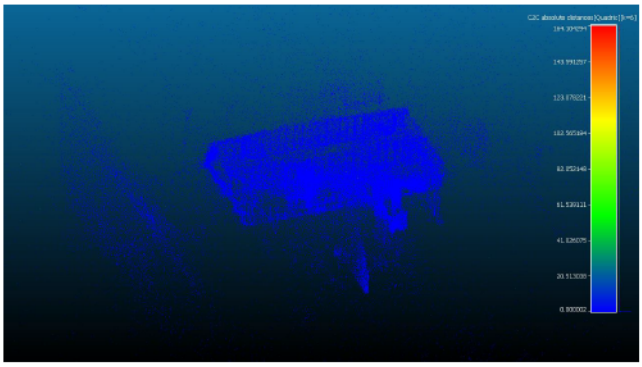
Figure 4.15: Hilti - Construction Upper Level 1: DISK Reconstructions



(a) R2D2+NN-Distance

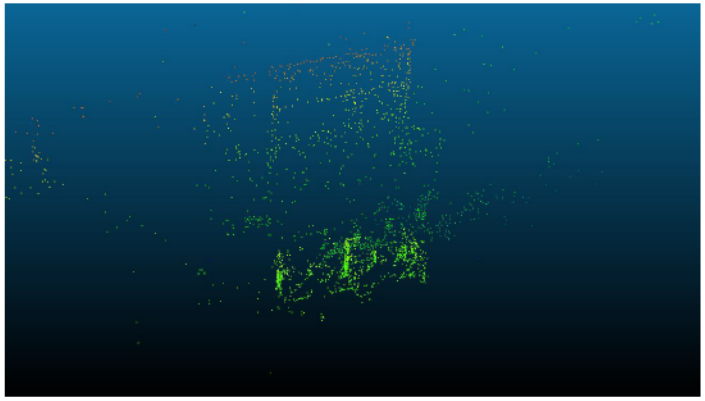


(b) R2D2+NN-Ratio



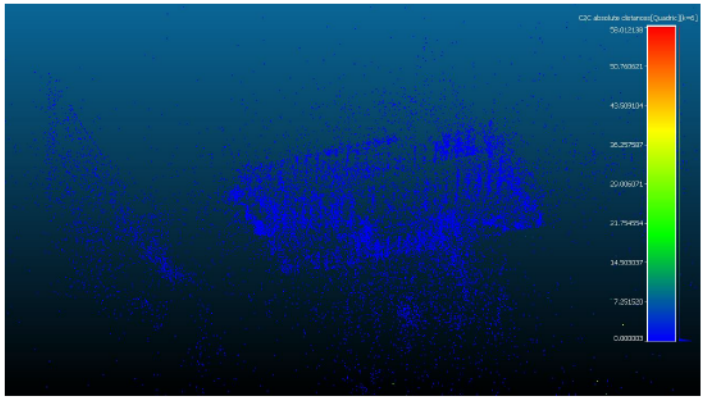
(c) R2D2+NN-Mutual

Figure 4.16: Hilti — Construction Upper Level 1: R2D2 Reconstructions



(a) SOSNet+NN-Distance

(b) SOSNet+NN-Ratio

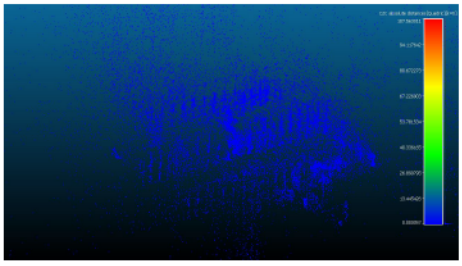


(c) SOSNet+NN-Mutual

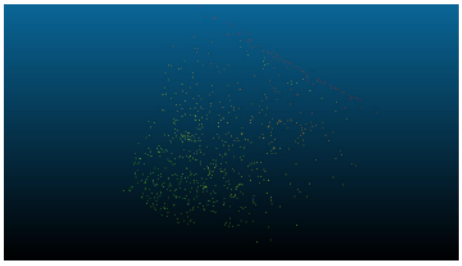
(d) SOSNet+AdaLAM

Figure 4.17: Hilti — Construction Upper Level 1: SOSNet Reconstructions

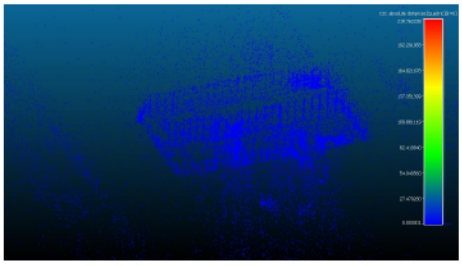
(a) SuperPoint+NN-Distance



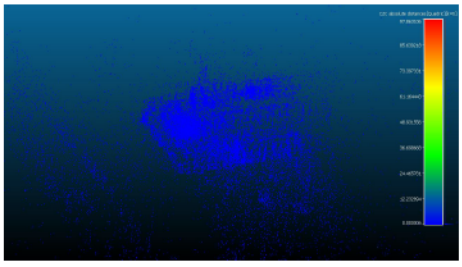
(b) SuperPoint+LightGlue



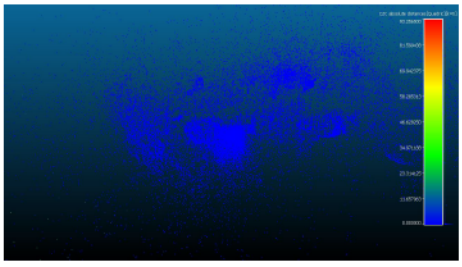
(c) SuperPoint+NN-Ratio



(d) SuperPoint+SuperGlue



(e) SuperPoint+NN-Mutual



(f) SuperPoint+SuperGlue-Fast

Figure 4.18: Hilti — Construction Upper Level 1: SuperPoint Reconstructions

4.1.3 Cloud-to-Cloud Distances

Table 4.3 shows the cloud-to-cloud distances for the indoor scenes, providing insights into the accuracy of the 3D reconstructions. Assessing the cloud-to-cloud distances was a difficult task in ConSLAM's sequence due to the mentioned misaligned and missed areas in the point clouds, as this error was not reflected in the cloud-to-cloud distances as higher errors, the cloud-to-cloud distances were computed using the available points.

Table 4.3: Cloud-to-Cloud Distances for Indoor Scenes

			ConSLAM - Sequence 2			Hilti - Construction Upper Level 1		
Baseline	**Extractor**	**Matcher**	**ICP Scale**	**Mean**	**STD**	**ICP Scale**	**Mean**	**STD**
SIFT	AKAZE	NN-BruteForce	0.70	**0.07**	0.19	1.00	0.33	0.47
SIFT	ORB	NN-BruteForce	-	-	-	-	-	-
SIFT	D2-Net	NN-Mutual	1.00	0.10	0.32	0.67	0.23	0.49
SIFT	D2-Net	NN-Ratio	1.00	0.23	0.38	-	-	-
SIFT	D2-Net	NN-Distance	1.00	0.26	0.73	1.00	0.21	0.34
SIFT	DISK	LightGlue	1.00	0.12	0.21	1.00	0.26	1.40
SIFT	DISK	NN-Mutual	1.00	0.08	0.17	1.00	0.30	2.22
SIFT	DISK	NN-Ratio	1.00	0.11	0.28	1.00	0.18	0.79
SIFT	DISK	NN-Distance	1.00	0.18	0.53	1.00	0.20	1.22
SIFT	R2D2	NN-Mutual	1.00	0.10	0.34	1.00	0.55	3.38
SIFT	R2D2	NN-Ratio	1.00	0.11	**0.12**	0.91	**0.07**	**0.07**
SIFT	R2D2	NN-Distance	1.00	0.09	0.35	1.00	0.28	1.47
SIFT	SOSNet	Adalam	1.00	0.43	0.83	1.00	0.34	0.63
SIFT	SOSNet	NN-Mutual	1.00	0.10	0.16	1.00	0.45	2.51
SIFT	SOSNet	NN-Ratio	1.00	0.36	0.90	1.00	0.45	0.36
SIFT	SOSNet	NN-Distance	1.00	0.30	0.75	-	-	-
SIFT	SuperPoint	NN-Mutual	1.00	0.17	1.45	0.68	0.30	1.90
SIFT	SuperPoint	NN-Ratio	1.00	0.07	0.20	-	-	-
SIFT	SuperPoint	NN-Distance	1.00	0.10	0.28	0.55	0.44	2.84
SIFT	SuperPoint	SuperGlue	1.00	0.14	0.42	1.00	0.67	5.17
SIFT	SuperPoint	SuperGlue-Fast	1.00	0.26	0.48	1.00	0.62	2.61
SIFT	SuperPoint	LightGlue	1.00	0.15	0.33	1.00	1.15	4.54

ConSLAM

For the ConSLAM dataset, the combination of AKAZE and R2D2 (NN-Mutual) produced the most accurate results. AKAZE achieved the lowest mean distance of 0.07 units, while

R2D2 exhibited the lowest standard deviation of 0.12 units. The ORB map was unable to align with SIFT, as most parts were not generated correctly (see sub-figure 4.6b). The Cloud-to-cloud error distances do not accurately represent the map's quality. For instance, DISK with NN-Mutual achieves a mean distance of 0.08 and a standard deviation of 0.17, closely aligning with SIFT's accuracy. However, considering the previous reconstruction error, it can be inferred that the map is not as precise as the cloud-to-cloud error suggests. On the contrary, SuperPoint (with SuperGlue-Fast) shows a higher mean distance of 0.26 and a standard deviation of 0.48, indicating a map that, while less accurate in quantitative terms, is qualitatively better than those produced by the other methods.

Hilti

In the Hilti dataset, the R2D2 (NN-Distance) method demonstrated the best performance, achieving a mean distance and a standard deviation of 0.07 units. In this dataset, ORB and D2-Net (NN-Ratio) generated incorrect results and could not be evaluated using this metric (see sub-figures 4.13b and 4.14b).

4.1.4 Performance Evaluation

Table 4.4 presents the performance metrics for the indoor scenes, including Elapsed Time, Mean Runtime for Feature Extraction, Feature Matching, and Global Search, as well as CPU, RAM, GPU, and Disk usage.

ConSLAM

Traditional Methods. SIFT, as the baseline method, displayed a relatively balanced performance across the evaluated metrics, recording an elapsed time of 1.20 hours, except for a significant CPU usage of 70.77% and RAM usage of 25.50 GB. AKAZE, despite its lower feature extraction and matching times, still exhibited considerable CPU and RAM usage, making it a faster choice than SIFT at the cost of more resource usage. ORB, despite having the longest elapsed time of 3.39 hours, demonstrated the quickest feature extraction time, though its matching time was the highest, indicating potential inefficiencies in certain stages of the process. Traditional methods showed lower GPU usage, which was expected since the only GPU-intensive step in the pipeline was the use of NETVLAD for pair matching. Additionally, the methods indicated higher CPU and memory consumption compared to deep learning methods, underscoring their reliance on CPU resources for processing.

Deep Learning-Based Methods. Overall, these methods displayed a significant improvement in processing times and resource efficiency compared to traditional methods. SuperPoint, in particular, exhibited exceptionally fast feature extraction runtimes (around 11 ms) and matching runtimes (approximately 1.55–1.70 ms), interestingly when paired with nearest neighbors techniques, not with tailored ones. The method's efficient use of system resources was further evidenced by its lower CPU and RAM usage, though GPU usage and memory were relatively higher, especially when paired with SuperGlue and SuperGlue-Fast. The shift towards higher GPU utilization in deep learning meth-

ods highlights the advancements in leveraging modern hardware capabilities to enhance performance.

Figure 4.19 presents a radar plot between the baseline and SuperPoint with SuperGlue-Fast. The values were scaled and translated into the range [0, 1] for improved visualization using min-max scaling according to the following formula:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{4.3}$$

where x is the original value and x' is the scaled value. The minimum and maximum values were calculated separately for each column. The original values are provided in Table 4.4.

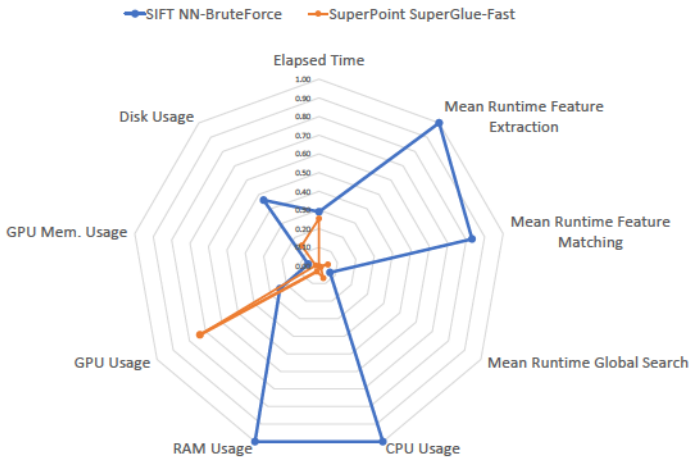


Figure 4.19: ConSLAM — Sequence 2: Performance Comparison

Hilti

Traditional Methods. The elapsed time for these methods varied, with SIFT exhibiting the shortest processing time. AKAZE demonstrated a notably low feature extraction runtime, significantly outperforming SIFT in this regard, making it a more efficient choice in scenarios with limited computational resources. CPU usage across these methods was relatively high, particularly for SIFT, which also recorded the highest RAM usage among the traditional methods. Despite these variations, disk usage remained consistently low across all traditional methods, with ORB occupying slightly more space. ORB, while fast in feature extraction, had the slowest feature matching runtime, indicating potential inefficiencies in certain stages of the process. SIFT showed a balanced yet higher resource consumption profile, suggesting its robustness but also highlighting the trade-offs in terms of processing speed and memory usage.

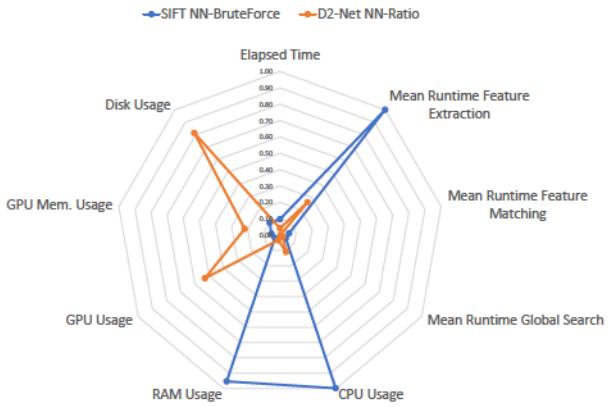
Deep Learning-Based Methods. Resource utilization for deep learning methods showed a higher GPU usage, which correlated with reduced CPU load. This trend was evident in methods like DISK and SuperPoint with LightGlue, which leveraged GPU resources, leading to improved processing speeds. RAM usage, while mostly higher in deep learning methods, was within acceptable limits, and disk usage remained slightly elevated due to the increased number of keypoints per images. SuperPoint, also demonstrated exceptionally fast feature extraction and matching runtimes when paired with nearest neighbors matching techniques

Figure 4.20 presents radar plots between the baseline and the top three deep learning methods (D2-Net, DISK, and R2D2). The values were scaled and translated into the range [0, 1] for improved visualization using min-max scaling according to the following formula:

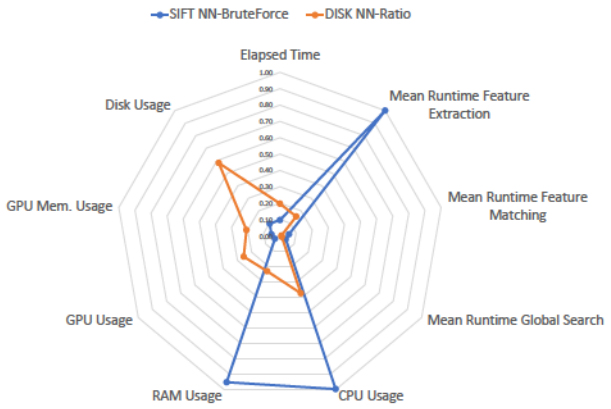
$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{4.4}$$

where x is the original value and x' is the scaled value. The minimum and maximum

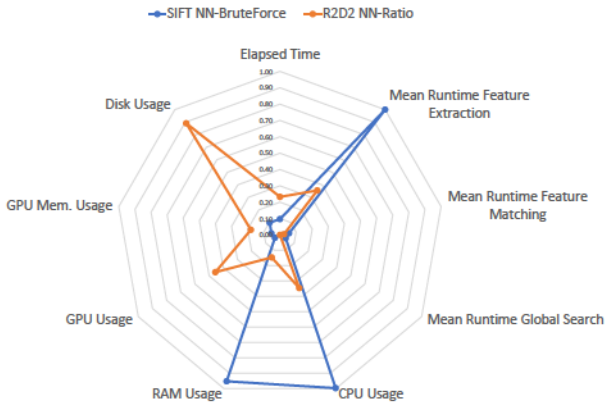
values were calculated separately for each column. The original values are provided in Table 4.4.



(a) D2-Net+NN-Ratio vs. Baseline



(b) DISK+Ratio vs. Baseline



(c) R2D2+NN-Ratio vs. Baseline

Figure 4.20: Hilti — Construction Upper Level 1: Performance Comparison

Table 4.4: Performance Metrics for Indoor Scenes

		ConSLAM - Sequence 2									Hilti - Construction Upper Level 1								
Extractor	**Matcher**	**Elapsed Time (hr)**	**Mean Runtime Feature Extraction (ms)**	**Mean Runtime Feature Matching (ms)**	**Mean Runtime Global Search (ms)**	**CPU Usage (%)**	**RAM Usage (GB)**	**GPU Usage (%)**	**GPU Mem. Usage (GB)**	**Disk Usage (GB)**	**Elapsed Time (hr)**	**Mean Runtime Feature Extraction (ms)**	**Mean Runtime Feature Matching (ms)**	**Mean Runtime Global Search (ms)**	**CPU Usage (%)**	**RAM Usage (GB)**	**GPU Usage (%)**	**GPU Mem. Usage (GB)**	**Disk Usage (GB)**
AKAZE	NN-BruteForce	1.71	2,156.18	799.00	3.08	64.33	14.30	0.19	2.58	1.08	0.32	149.30	3.02	1.89	70.50	2.91	0.61	2.03	0.23
SIFT	NN-BruteForce	1.20	2,229.80	1,236.31	3.32	70.77	25.50	3.49	2.73	2.20	0.27	200.21	9.89	1.85	70.31	5.84	0.71	2.03	0.49
ORB	NN-BruteForce	3.39	106.05	1,485.64	3.11	66.67	5.15	**0.12**	2.89	1.33	1.07	28.63	151.73	1.85	64.75	3.16	**0.23**	2.26	0.61
D2-Net	NN-Mutual	1.80	119.20	3.88	2.97	43.38	4.68	2.94	2.80	4.29	1.19	59.30	1.95	1.76	36.67	3.77	1.04	2.52	2.90
D2-Net	NN-Ratio	**0.31**	120.85	3.93	3.02	21.02	2.90	8.30	2.80	3.93	0.16	60.09	2.03	1.78	12.19	2.88	7.46	2.47	2.62
D2-Net	NN-Distance	1.22	120.85	3.89	2.97	39.71	4.07	3.41	2.80	4.09	0.66	60.27	1.97	1.79	27.54	2.86	1.88	2.45	2.73
DISK	LightGlue	2.58	79.04	41.92	3.02	38.19	7.03	7.55	5.52	2.64	1.35	39.83	26.77	1.77	26.67	5.62	8.38	3.42	2.16
DISK	NN-Mutual	2.15	79.16	4.96	**2.88**	45.31	6.30	3.02	3.33	2.56	1.69	39.76	2.78	1.79	37.75	5.38	1.03	2.42	2.18
DISK	NN-Ratio	0.81	79.16	5.01	2.88	34.27	4.76	5.13	3.32	2.32	0.47	39.76	2.82	1.75	29.46	3.51	3.71	2.44	1.93
DISK	NN-Distance	1.20	79.16	4.96	2.91	39.44	5.38	4.03	3.32	2.39	0.69	39.76	2.80	1.75	32.38	3.67	2.53	2.68	1.96
R2D2	NN-Mutual	1.85	173.22	6.93	2.95	44.65	6.18	4.16	3.48	2.92	2.06	78.00	5.23	1.71	36.75	6.00	1.71	2.67	3.16
R2D2	NN-Ratio	0.57	173.22	7.00	2.94	24.27	3.80	9.46	3.23	2.65	0.54	77.82	5.28	**1.71**	27.63	3.25	6.46	2.37	2.84
R2D2	NN-Distance	1.92	172.85	6.95	2.97	45.21	6.13	4.07	3.23	2.92	1.65	77.94	5.22	1.71	33.90	5.80	2.12	4.51	3.11
SOSNet	Adalam	0.34	272.46	8.77	9.33	25.98	3.74	4.88	3.05	0.48	0.17	117.13	7.40	5.01	33.71	3.71	4.96	3.68	0.26
SOSNet	NN-Mutual	0.51	278.08	**1.50**	9.22	39.67	3.74	2.64	3.04	0.51	0.29	117.98	1.43	5.01	40.00	3.70	1.08	3.69	0.25
SOSNet	NN-Ratio	0.35	273.19	1.68	9.11	24.88	3.70	3.08	3.04	**0.42**	0.12	120.36	1.64	5.04	17.48	3.71	2.80	3.67	**0.21**
SOSNet	NN-Distance	0.38	269.04	1.60	9.41	27.00	3.76	3.04	3.04	0.43	0.13	118.23	1.55	5.03	15.38	3.67	2.32	3.69	0.21
SuperPoint	NN-Mutual	0.74	**11.09**	1.55	2.98	39.08	2.82	2.50	**2.55**	1.01	0.47	**10.44**	**1.41**	1.75	34.85	2.78	0.91	**1.90**	0.49
SuperPoint	NN-Ratio	0.37	11.12	1.70	2.98	23.60	2.78	3.20	**2.55**	0.88	**0.08**	10.48	1.64	1.74	**4.94**	**2.77**	5.19	1.93	0.42
SuperPoint	NN-Distance	0.60	11.15	1.65	3.01	33.42	**2.77**	2.64	**2.55**	0.93	0.19	10.51	1.55	1.75	16.50	2.79	2.34	1.96	0.45
SuperPoint	SuperGlue	1.08	11.11	84.37	2.93	**15.56**	3.38	14.08	2.61	0.97	1.05	10.48	69.04	1.75	11.17	2.78	6.27	2.73	0.47
SuperPoint	SuperGlue-Fast	1.09	11.11	73.92	2.95	19.35	3.48	10.38	2.61	0.98	0.93	10.51	62.23	1.73	11.44	2.78	5.23	2.40	0.47
SuperPoint	LightGlue	0.59	11.28	15.27	3.05	26.65	3.24	7.63	2.67	0.97	0.37	10.89	14.47	1.77	15.71	2.79	13.89	2.24	0.47

4.2 Outdoor Scenes

4.2.1 Dataset Evaluation

Figures and 4.21 illustrate the box plots of co-visibility ratios, and Table 4.5 provides descriptive statistics, including count, mean, standard deviation, minimum, 25th percentile, median, 75th percentile, and maximum.

Private

High Co-Visibility Performance (0.8–1.0). The three traditional methods fall within this category, with SIFT identified as the most effective, exhibiting a mean co-visibility ratio of 0.947 and a low standard deviation of 0.028, indicating dependable and consistent performance. In contrast, AKAZE and ORB, although within the same range, present lower mean values and slightly higher standard deviations, reflecting reduced consistency. Deep learning-based approaches, particularly D2-Net with NN-Mutual matching, demonstrate the highest performance among all methods, with a mean co-visibility ratio of 0.976 and an exceptionally low standard deviation of 0.004. This suggests that D2-Net, when integrated with the appropriate matching strategy, significantly improves scene visibility overlap. SOSNet, R2D2, and SuperPoint, utilizing either NN-Mutual or NN-Distance, also show high co-visibility ratios with minimal variability.

Moderate Co-visibility Performance (0.4-0.8). Methods such as DISK, D2-Net with NN-Distance, and SuperPoint with its specialized techniques (LightGlue and SuperGlue) exhibit moderate co-visibility performance. All show a very similar mean co-visibility ratio of between 0.4 and 0.5, suggesting a reasonable overlap in scene visibility, which could be adequate for certain 3D reconstruction tasks. However, these methods may lack the robustness and consistency observed in higher-performing techniques, further emphasizing their reliance on the chosen matching strategy.

Low Co-visibility Performance (0-0.4). All NN-Ratio combinations, along with SOS-Net (NN-Distance) and SuperPoint (LightGlue), are categorized under low co-visibility performance. These methods display the lowest mean co-visibility ratios, signifying a substantial deficiency in scene visibility overlap. Consequently, these approaches may be less effective in identifying and matching image pairs with common visual elements, resulting in diminished accuracy and consistency in 3D reconstruction applications.

Outliers and Extremes. The analysis of the box plots in Figure 4.21 reveals extended whiskers and a significant number of outliers, indicating inconsistent performance for methods such as D2-Net, DISK, R2D2, SOSNet, and SuperPoint with NN-Ratio. This variability highlights the critical importance of selecting appropriate matching strategies and potentially necessitates method-specific tuning to attain optimal performance.

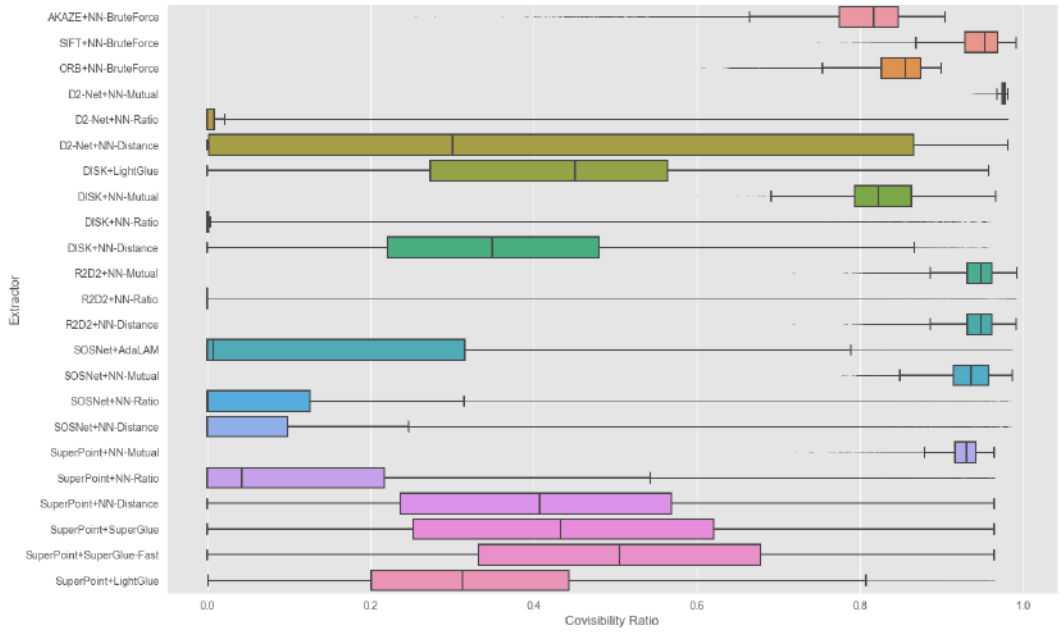


Figure 4.21: Private: Construction Site Outdoor Co-visibility Ratios

Hilti

Due to time constraints, the evaluation could not be conducted. Each map would have required approximately 60 hours to execute, and with a total of 23 combinations, the

cumulative time needed would have been 1380 hours.

Table 4.5: Descriptive Statistics of Co-visibility Ratios for Outdoor Scenes

		Private - Construction Site Outdoor							
Extractor	**Matcher**	**count**	**mean**	**std**	**min**	**25%**	**50%**	**75%**	**max**
AKAZE	NN-BruteForce	113,526	0.800	0.067	0.256	0.775	0.817	0.847	0.905
SIFT	NN-BruteForce	113,526	0.947	0.028	0.748	0.929	0.953	0.969	0.992
ORB	NN-BruteForce	113,526	0.844	0.043	0.606	0.826	0.856	0.874	0.900
D2-Net	NN-Mutual	113,526	**0.976**	**0.004**	0.938	0.974	0.977	0.979	0.982
D2-Net	NN-Ratio	113,526	0.081	0.209	-	-	0.000	0.009	0.981
D2-Net	NN-Distance	113,526	0.411	0.398	-	0.002	0.300	0.866	0.981
DISK	LightGlue	113,526	0.420	0.185	-	0.273	0.451	0.564	0.958
DISK	NN-Mutual	113,526	0.829	0.050	0.601	0.794	0.823	0.864	0.969
DISK	NN-Ratio	113,526	0.041	0.124	-	-	-	0.002	0.958
DISK	NN-Distance	113,526	0.352	0.177	0.000	0.221	0.349	0.480	0.958
R2D2	NN-Mutual	113,526	0.943	0.025	0.719	0.931	0.948	0.961	0.992
R2D2	NN-Ratio	113,526	0.041	0.128	-	-	-	0.000	0.992
R2D2	NN-Distance	113,526	0.943	0.025	0.719	0.931	0.948	0.961	0.992
SOSNet	AdaLAM	113,526	0.231	0.358	-	-	0.007	0.316	0.986
SOSNet	NN-Mutual	113,526	0.933	0.030	0.778	0.914	0.936	0.958	0.987
SOSNet	NN-Ratio	113,526	0.097	0.174	-	-	0.001	0.126	0.984
SOSNet	NN-Distance	113,526	0.086	0.165	-	-	0.000	0.099	0.984
SuperPoint	NN-Mutual	113,526	0.927	0.020	0.720	0.916	0.931	0.942	0.965
SuperPoint	NN-Ratio	113,526	0.137	0.191	-	0.000	0.042	0.217	0.965
SuperPoint	NN-Distance	113,526	0.404	0.218	-	0.237	0.408	0.569	0.965
SuperPoint	SuperGlue	113,526	0.439	0.228	-	0.253	0.433	0.621	0.965
SuperPoint	SuperGlue-Fast	113,526	0.503	0.216	-	0.333	0.506	0.678	0.965
SuperPoint	LightGlue	113,526	0.333	0.175	0.001	0.201	0.313	0.444	0.965

4.2.2 Reconstruction Evaluation

Table 4.6 presents the reconstruction results, detailing the number of registered images, points, observations, mean track length, mean observations per image, mean reprojection error, mean number of keypoints, and mean number of matches. A visual examination of the reconstructed point clouds offers insights into the quality and completeness of the reconstructions can be found in Figures 4.23 to 4.28 for Hilti and from 4.30 to 4.35 for the Private dataset, which display the generated point clouds from a front isometric view.

Table 4.6: Reconstruction Results for Outdoor Scenes

		Private - Construction Site Outdoor								Hilti - Construction Site Outdoor 1							
Extractor	Matcher	# Reg. Img.	# Points	# Obs.	Mean Track Length	Mean Obs. Img	Mean Reproj. Error	Mean # Keypoints	Mean # Matches	# Reg. Img.	# Points	# Obs.	Mean Track Length	Mean Obs. Img	Mean Reproj. Error	Mean # Keypoints	Mean # Matches
AKAZE	NN-BruteForce	477	49,652	597,559	12.03	1,252.74	0.85	1,569.00	557.31	321	13,797	90,506	6.56	281.95	1.16	458.68	144.55
SIFT	NN-BruteForce	477	158,408	1,121,996	7.08	2,352.19	**0.62**	3,515.05	1,142.23	452	24,355	130,712	5.37	289.19	0.80	580.85	162.72
ORB	NN-BruteForce	477	174,582	2,677,935	**15.34**	5,614.12	1.00	7,614.78	**2,204.80**	953	87,726	749,104	8.54	786.05	1.10	2,140.84	551.43
D2-Net	NN-Mutual	477	380,502	2,129,293	5.60	4,463.93	1.55	5,984.59	1,623.07	651	79,271	552,331	6.97	848.43	1.41	2,312.98	218.24
D2-Net	NN-Ratio	477	108,983	654,832	6.01	1,372.81	1.32	5,984.59	170.39	43	3,871	48,141	12.44	1,119.56	1.13	2,312.98	91.12
D2-Net	NN-Distance	477	288,048	1,604,316	5.57	3,363.35	1.51	5,984.59	546.26	957	216,417	1,308,981	6.05	1,367.80	1.38	2,312.98	579.80
DISK	LightGlue	477	331,537	**3,582,694**	10.81	**7,510.89**	1.04	7,967.76	1,880.21	673	115,683	1,199,493	10.37	1,782.31	0.64	4,128.47	404.79
DISK	NN-Mutual	477	**399,953**	3,450,039	8.63	7,232.79	0.90	7,967.76	1,757.26	704	136,753	1,421,581	10.40	2,019.29	0.76	4,128.47	454.50
DISK	NN-Ratio	477	376,608	2,973,756	7.90	6,234.29	0.75	7,967.76	825.97	967	**309,370**	2,799,201	9.05	2,894.73	0.88	4,128.47	866.34
DISK	NN-Distance	477	386,539	3,015,828	7.80	6,322.49	0.77	7,967.76	867.81	920	288,775	3,005,325	10.41	3,266.66	0.99	4,128.47	775.60
R2D2	NN-Mutual	477	237,509	2,974,663	12.52	6,236.19	1.14	**8,000**	1,015.63	233	34,962	510,494	14.60	2,190.96	0.67	**6,618.23**	349.81
R2D2	NN-Ratio	477	212,481	1,590,765	7.49	3,334.94	0.87	**8,000**	325.12	956	273,521	**3,364,291**	12.30	**3,519.13**	0.91	**6,618.23**	**878.93**
R2D2	NN-Distance	477	237,136	2,974,582	12.54	6,236.02	1.14	**8,000**	1,015.04	**974**	243,966	3,217,281	13.19	3,303.16	0.90	**6,618.23**	767.43
SOSNet	Adalam	477	55,904	481,336	8.61	1,009.09	0.86	1,325.63	213.92	121	2,652	18,944	7.14	156.56	0.76	290.41	25.98
SOSNet	NN-Mutual	477	65,807	507,848	7.72	1,064.67	0.86	1,325.63	506.20	576	21,161	118,795	5.61	206.24	0.92	290.41	97.47
SOSNet	NN-Ratio	477	50,713	417,985	8.24	876.28	0.74	1,325.63	158.90	55	1,251	6,153	4.92	111.87	0.62	290.41	25.47
SOSNet	NN-Distance	357	38,796	310,958	8.02	871.03	0.68	1,325.63	147.91	66	2,555	10,310	4.04	156.21	0.66	290.41	65.81
SuperPoint	NN-Mutual	477	41,489	433,872	10.46	909.58	1.18	1,048.51	447.31	43	698	12,696	**18.19**	295.26	**0.57**	509.43	46.49
SuperPoint	NN-Ratio	477	33,026	356,243	10.79	746.84	1.07	1,048.51	160.99	763	38,158	310,980	8.15	407.58	1.05	509.43	95.37
SuperPoint	NN-Distance	477	36,591	400,741	10.95	840.13	1.14	1,048.51	229.74	260	10,022	90,903	9.07	349.63	0.97	509.43	81.25
SuperPoint	SuperGlue	477	38,658	448,101	11.59	939.42	1.26	1,048.51	343.53	689	37,469	260,311	6.95	377.81	0.99	509.43	162.16
SuperPoint	SuperGlue-Fast	477	39,565	450,761	11.39	944.99	1.26	1,048.51	355.03	705	46,798	261,746	5.59	371.27	1.17	509.43	100.28
SuperPoint	LightGlue	477	38,147	447,215	11.72	937.56	1.26	1,048.51	338.02	757	37,866	319,856	8.45	422.53	1.06	509.43	98.37

Hilti

Traditional Methods. The traditional methods demonstrated distinct trends across various reconstruction quality metrics. AKAZE's performance was similar to SIFT in mean track length, mean observations per image, mean number of keypoints, and matches. However, AKAZE exhibited a higher mean reprojection error (1.16) compared to SIFT (0.80). While SIFT had better performance in terms of mean reprojection error (0.8), it had lower scores in all other metrics compared to ORB. ORB outperformed both AKAZE and SIFT in all metrics except for mean reprojection error.

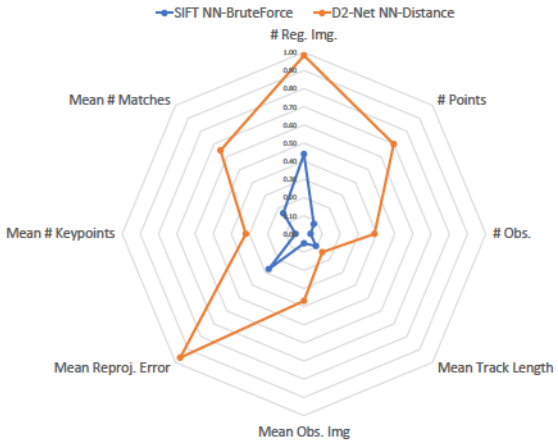
Deep Learning-Based Methods. Deep learning-based methods demonstrate superior performance across all metrics compared to traditional approaches, highlighting their effectiveness in complex 3D reconstruction tasks. Nevertheless, notable variability in performance is observed within these methods based on their specific configurations. Consistent with previous datasets, DISK, D2-Net, and R2D2 exhibit strong performance. DISK and R2D2, in particular, excel in the number of images and points generated, as well as in mean keypoints per image. Nearest Neighbor techniques are the most effective matchers for all three methods, with NN-Ratio being the most effective for DISK and R2D2, and NN-Distance proving most effective for D2-Net. For example, R2D2 combined with the NN-Ratio matcher registers 956 images, 273,521 points, and 3,364,291 observations, achieving a mean track length of 12.30 and mean observations per image of 3,519.13, the highest among all methods. DISK, when paired with the NN-Ratio matcher, registers a similar number of images (967) and generates 309,370 points and 2,799,201 observations, with a mean track length of 9.05 and mean observations per image of 2,894.73. While SuperPoint shows strong performance with its tailored matching techniques (SuperGlue and LightGlue), its performance does not match that of DISK, D2-Net, and R2D2.

Figure 4.22 presents radar plots of the reconstruction metrics between the baseline and the top three deep learning methods (D2-Net, DISK, and R2D2). The values were scaled

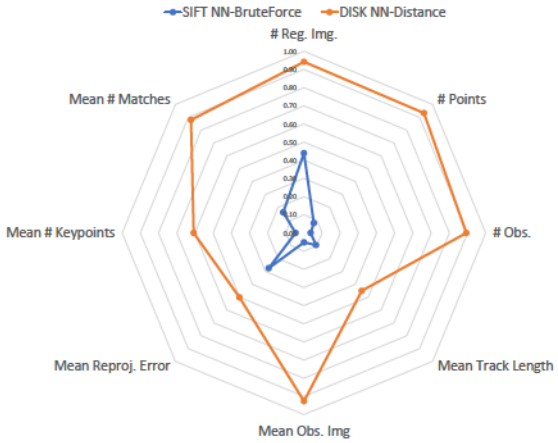
and translated into the range [0, 1] for improved visualization using min-max scaling according to the following formula:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{4.5}$$

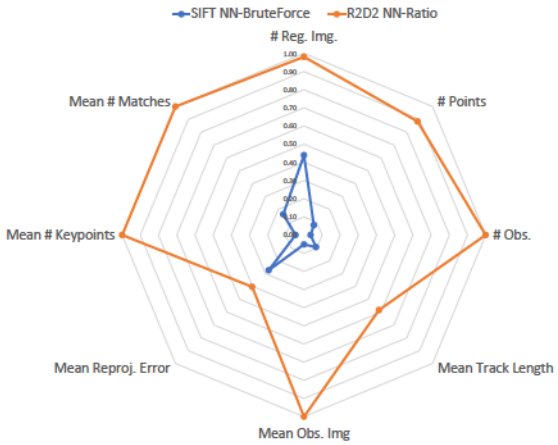
where x is the original value and x' is the scaled value. The minimum and maximum values were calculated separately for each column. The original values are provided in Table 4.6.



(a) D2-Net+NN-Distance vs. Baseline

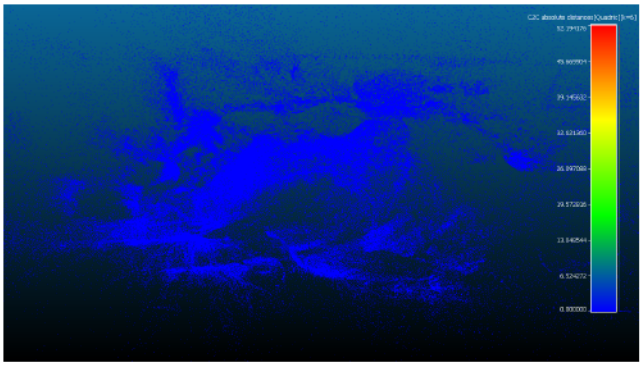


(b) DISK+NN-Distance vs. Baseline

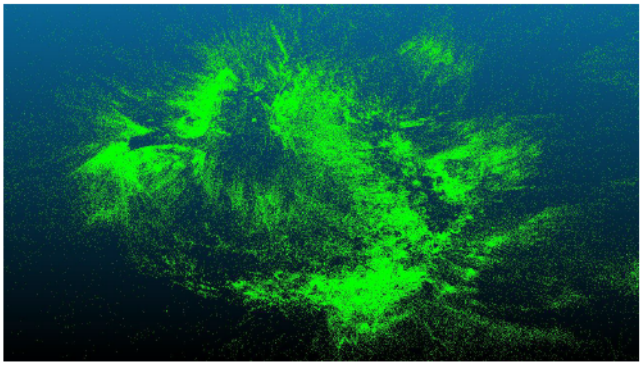


(c) R2D2+NN-Ratio vs. Baseline

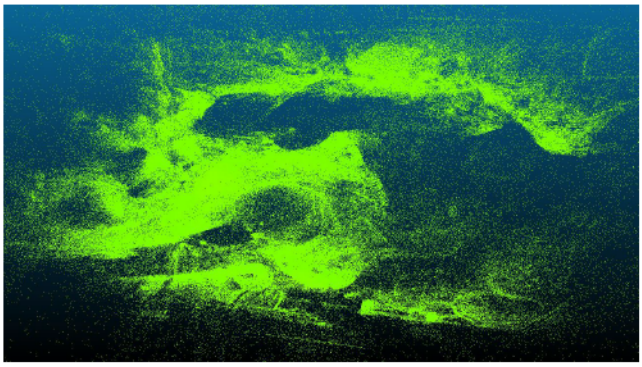
Figure 4.22: Hilti — Construction Site Outdoor 1: Reconstruction Comparison



(a) AKAZE+NN-BruteForce

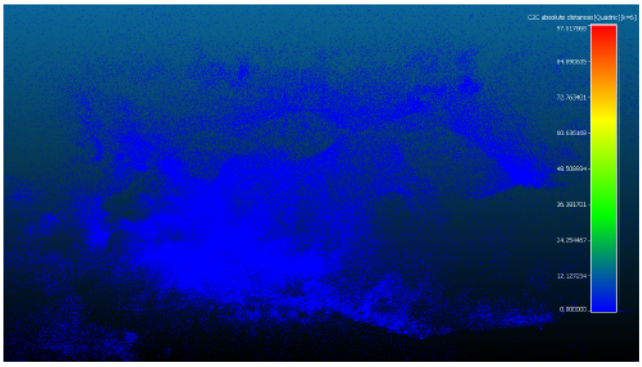


(b) ORB+NN-BruteForce

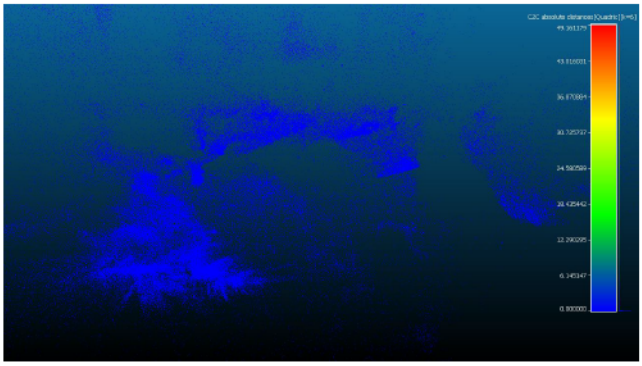


(c) SIFT+NN-BruteForce (Baseline)

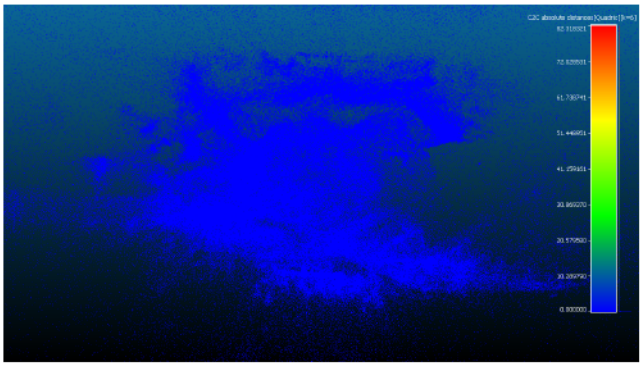
Figure 4.23: Hilti — Construction Site Outdoor 1: Traditional Reconstructions



(a) D2-Net+NN-Distance

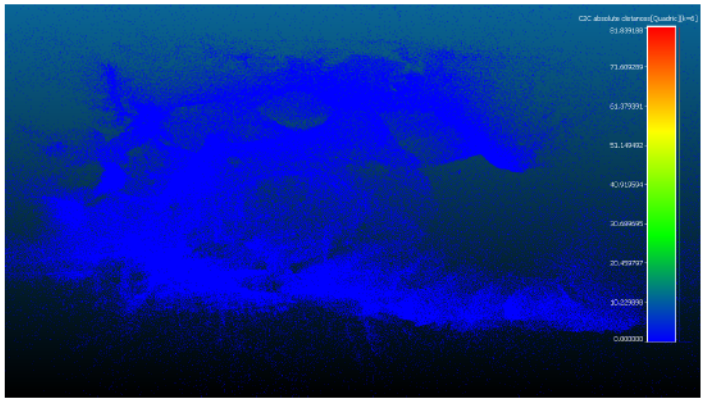


(b) D2-Net+NN-Ratio

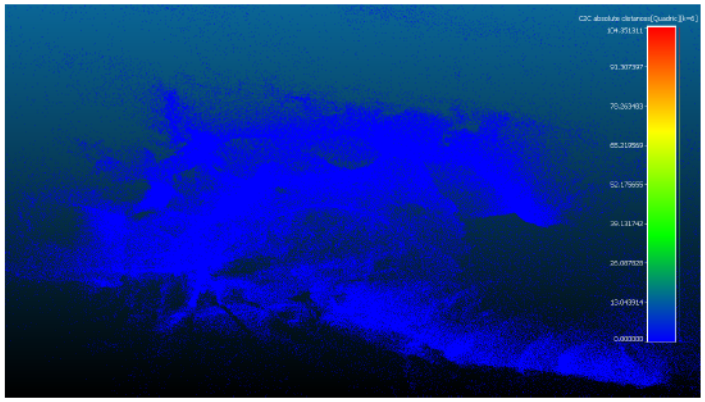


(c) D2-Net+NN-Mutual

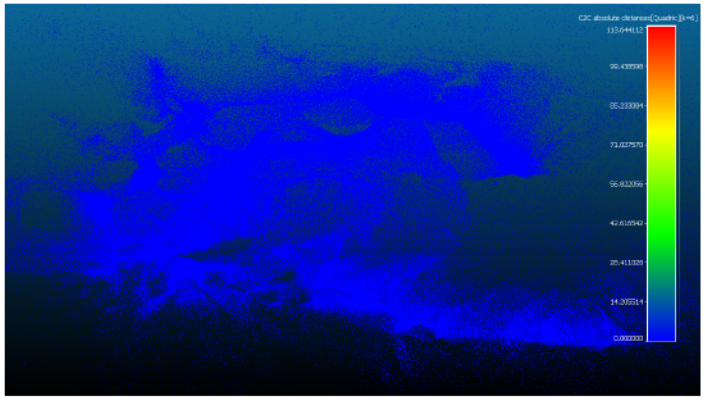
Figure 4.24: Hilti — Construction Site Outdoor 1: D2-Net Reconstructions



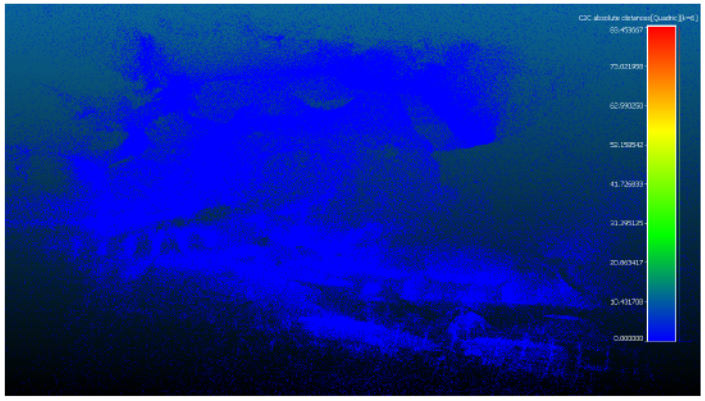
(a) DISK+NN-Distance



(b) DISK+NN-Ratio

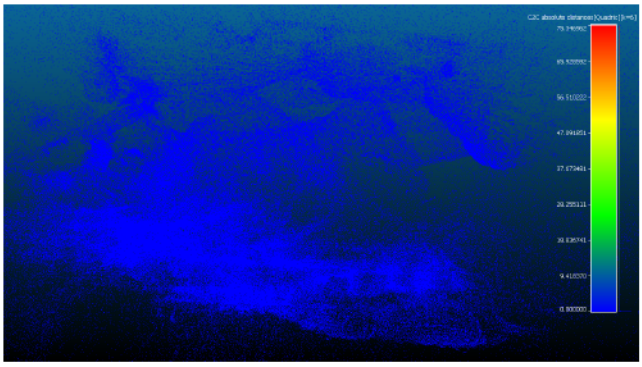


(c) DISK+NN-Mutual

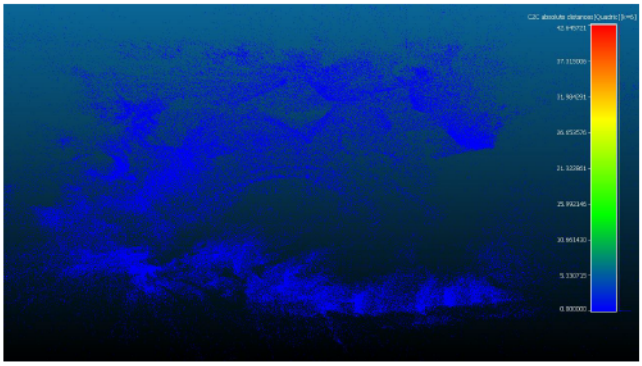


(d) DISK+LightGlue

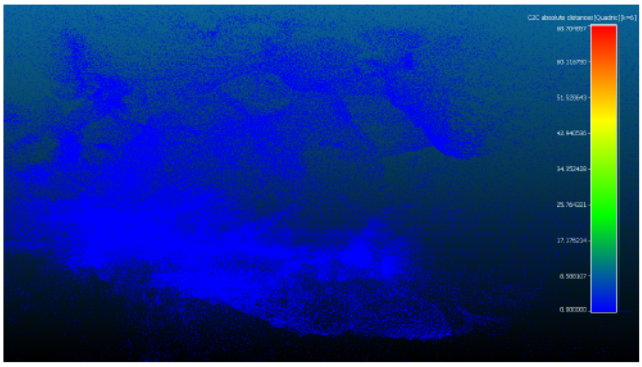
Figure 4.25: Hilti — Construction Site Outdoor: DISK Reconstructions



(a) R2D2+NN-Distance

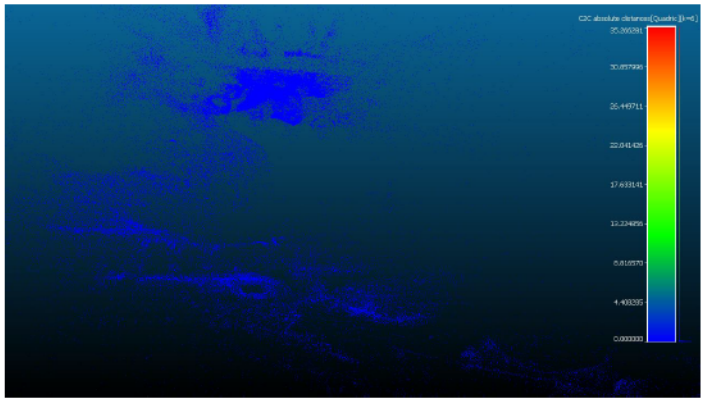


(b) R2D2+NN-Ratio

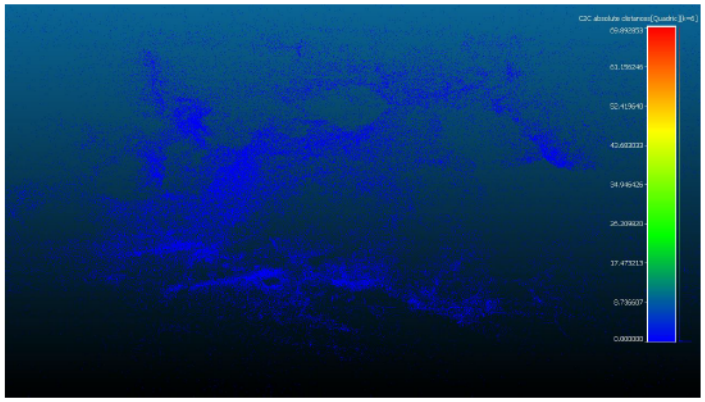


(c) R2D2+NN-Mutual

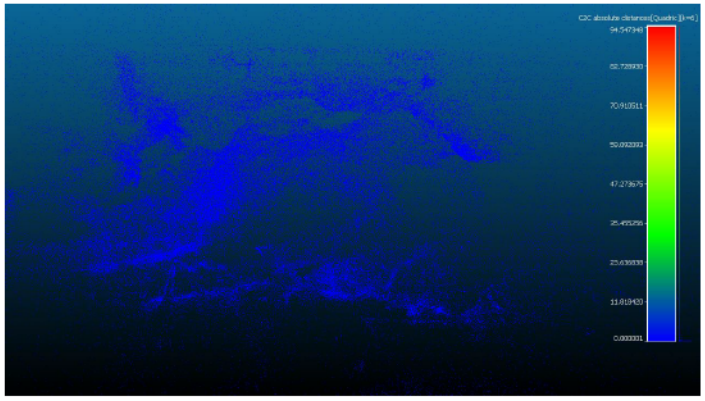
Figure 4.26: Hilti — Construction Site Outdoor: R2D2 Reconstructions



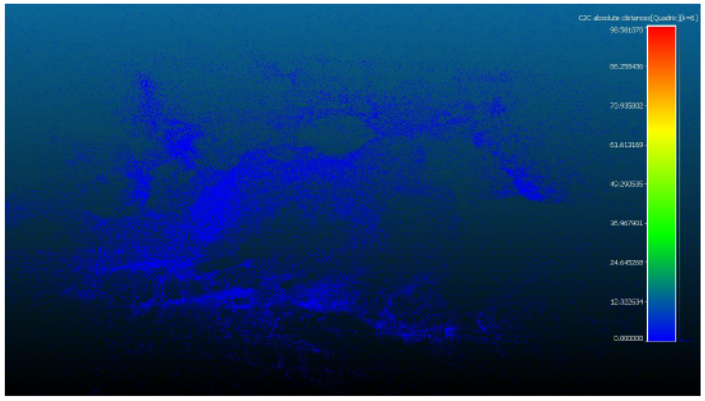
(a) SOSNet+NN-Distance



(b) SOSNet+NN-Ratio

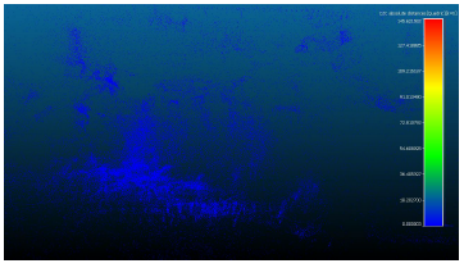


(c) SOSNet+NN-Mutual

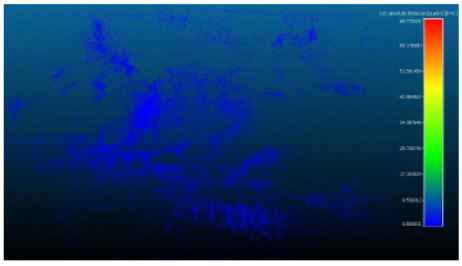


(d) SOSNet+AdaLAM

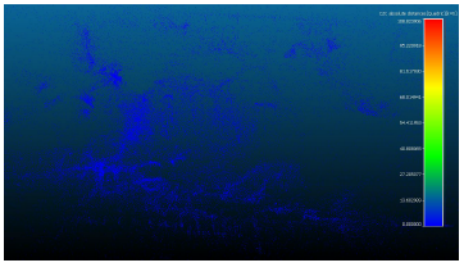
Figure 4.27: Hilti — Construction Site Outdoor: SOSNet Reconstructions



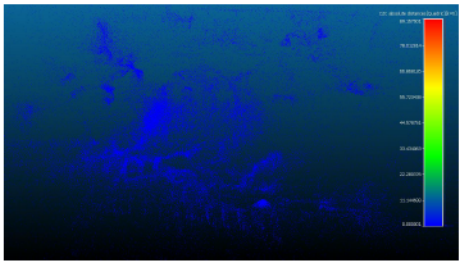
(a) SuperPoint+NN-Distance



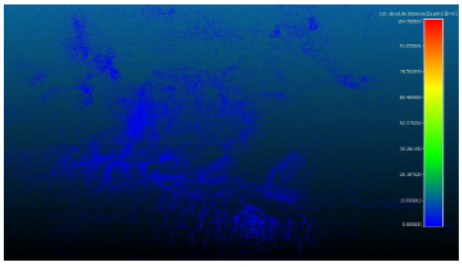
(b) SuperPoint+LightGlue



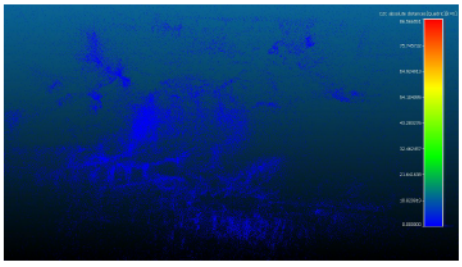
(c) SuperPoint+NN-Ratio



(d) SuperPoint+SuperGlue



(e) SuperPoint+NN-Mutual



(f) SuperPoint+SuperGlue-Fast

Figure 4.28: Hilti — Construction Site Outdoor: SuperPoint Reconstructions

Private

Traditional Methods. In the evaluation of traditional methods, ORB produced the highest number of 3D points and observations, suggesting a more robust feature detection and matching capabilities. In contrast, SIFT, as the baseline method, while generating fewer points, it achieved a lower mean reprojection error, indicative of higher accuracy in 3D point projection back onto 2D images. AKAZE, with the fewest points and observations, exhibited a mean track length of 12.03, indicating reasonable feature tracking across images. However, its mean reprojection error was higher than that of SIFT, implying less precise 3D point projections. In terms of cloud-to-cloud error distances, ORB and AKAZE demonstrated moderate performance with standard deviations of 0.80 and 0.95 respectively, whereas SIFT exhibited better consistency and lower mean error values.

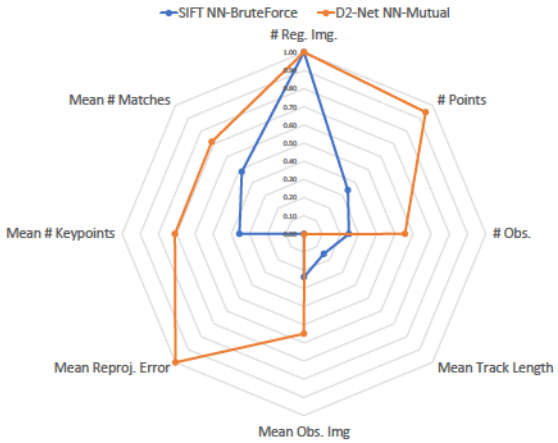
Deep Learning-Based Methods. The deep learning-based methods, consistently registered 477 images (except SOSNet with NN-Distance). Among these, DISK with NN-Mutual and LightGlue stood out, producing the highest number of 3D points and observations, highlighting their effectiveness in feature extraction and matching. D2-Net, when paired with NN-Mutual, also demonstrated a significant number of points and observations (380,502 and 2,129,293), albeit with a higher mean reprojection error compared to DISK methods. The mean track length for deep learning methods generally varied, with DISK, D2-net and R2D2 achieving similar values, ranging between 0.7-1.2), indicating strong feature tracking abilities. Mean observations per image were high for DISK with LightGlue, showing its superior feature match density. Compared to SIFT point cloud, the deep learning methods (R2D2 and DISK) generated more complete point clouds where even the gravel and grass deformations were captured as the excavator moved, which was not the case for SIFT (see figures 4.32 and 4.33).

Figure 4.29 presents radar plots between the baseline method and the top three deep learning methods (D2-Net, DISK, and R2D2). The values were scaled and translated

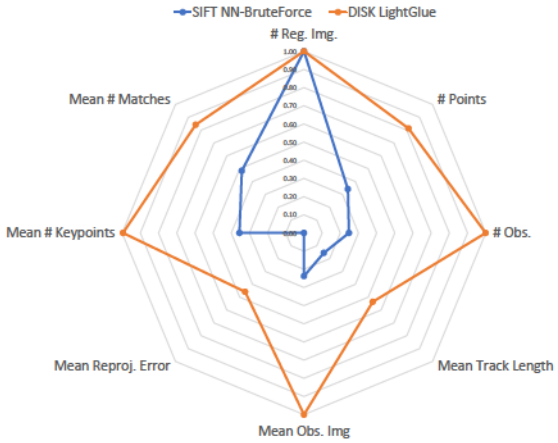
into the range [0, 1] for improved visualization using min-max scaling according to the following formula:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{4.6}$$

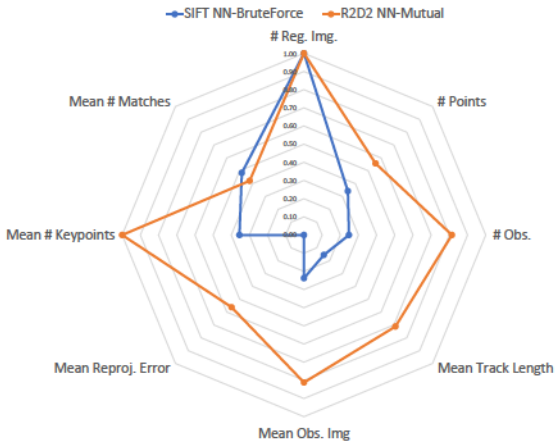
where x is the original value and x' is the scaled value. The minimum and maximum values were calculated separately for each column. The original values are provided in Table 4.6.



(a) D2-Net+NN-Mutual vs. Baseline

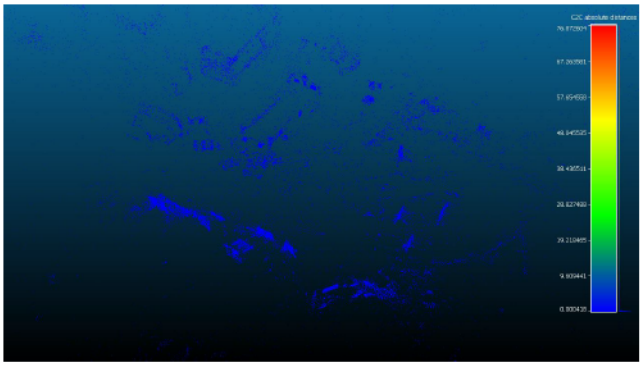


(b) DISK+LightGlue vs. Baseline

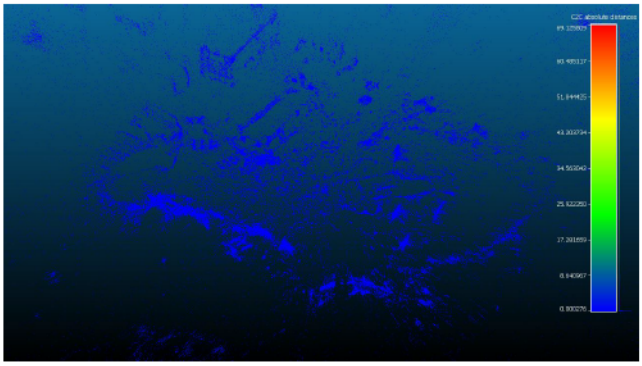


(c) R2D2+NN-Mutual vs. Baseline

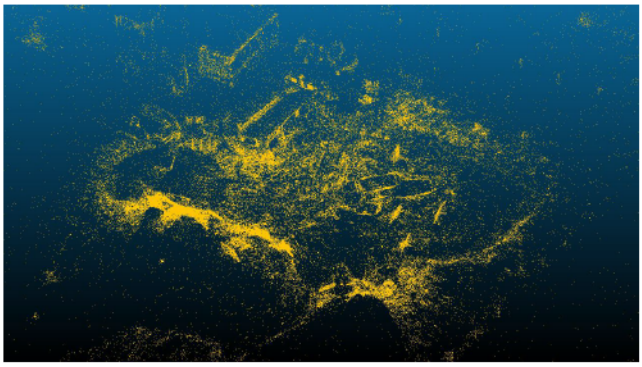
Figure 4.29: Private — Construction Site Outdoor: Reconstruction Comparison



(a) AKAZE

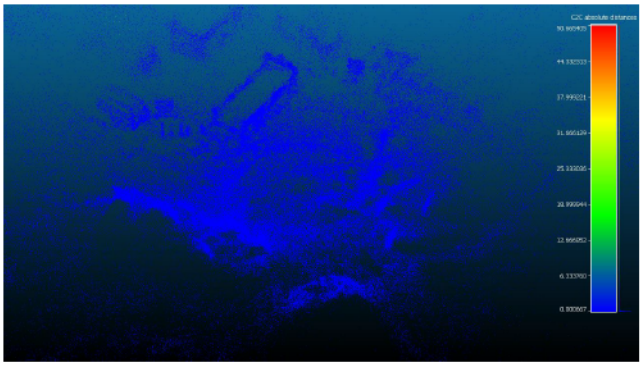


(b) ORB

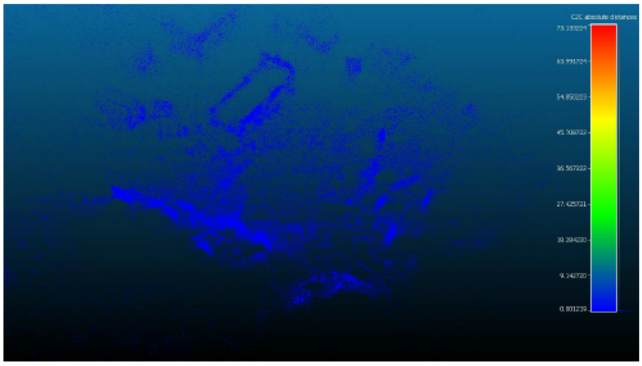


(c) SIFT (Baseline)

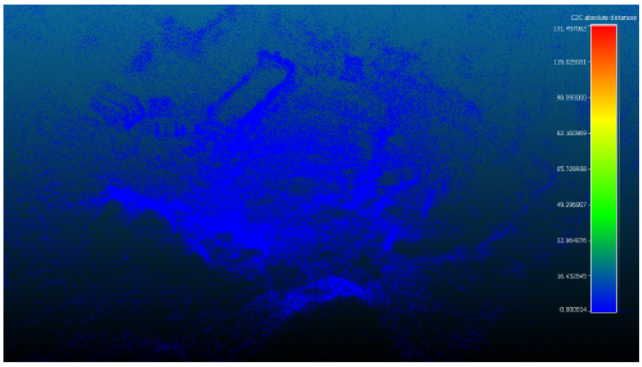
Figure 4.30: Private — Construction Site Outdoor: Traditional Reconstructions



(a) D2-Net+NN-Distance

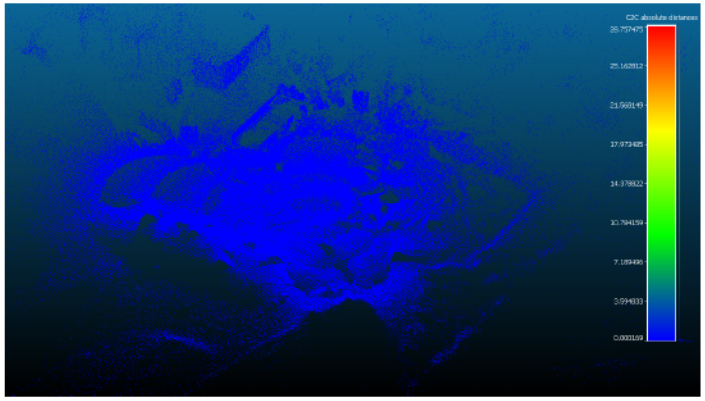


(b) D2-Net+NN-Ratio

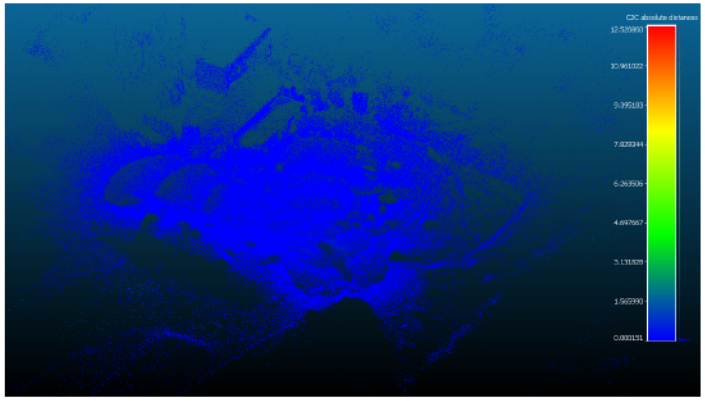


(c) D2-Net+NN-Mutual

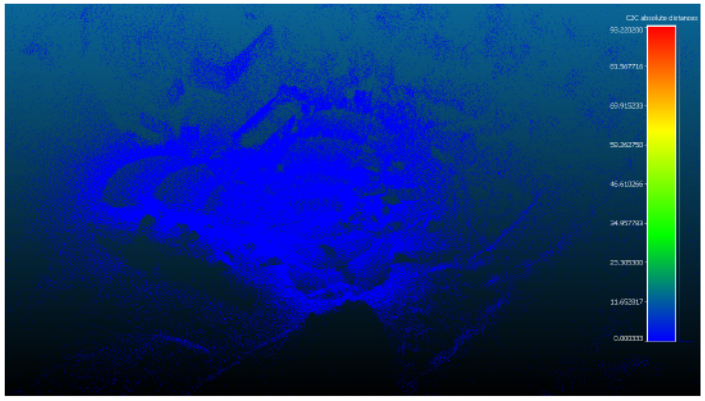
Figure 4.31: Private — Construction Site Outdoor: D2-Net Reconstructions



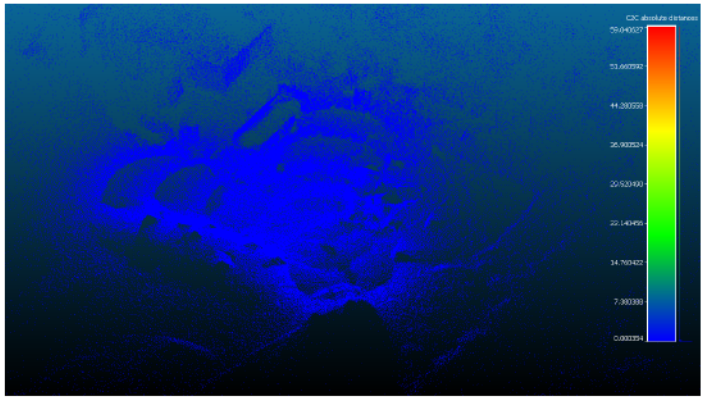
(a) DISK+NN-Distance



(b) DISK+NN-Ratio

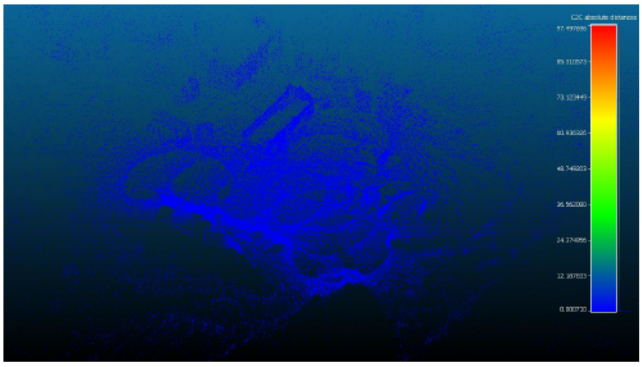


(c) DISK+NN-Mutual

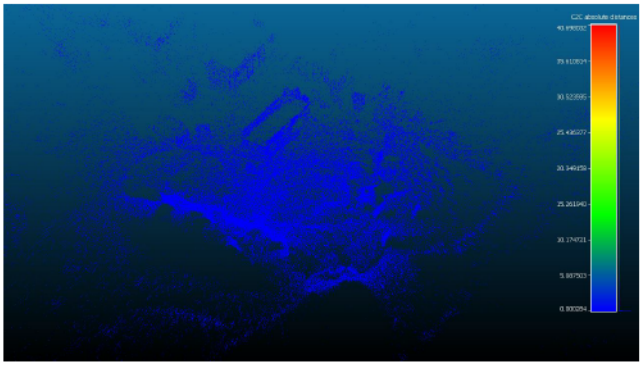


(d) DISK+LightGlue

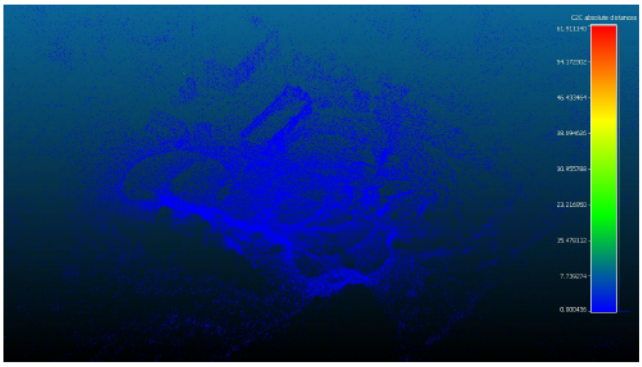
Figure 4.32: Private — Construction Site Outdoor: DISK Reconstructions



(a) R2D2+NN-Distance

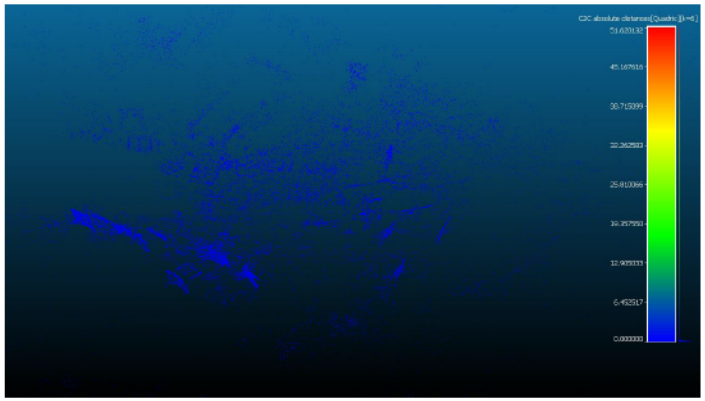


(b) R2D2+NN-Ratio

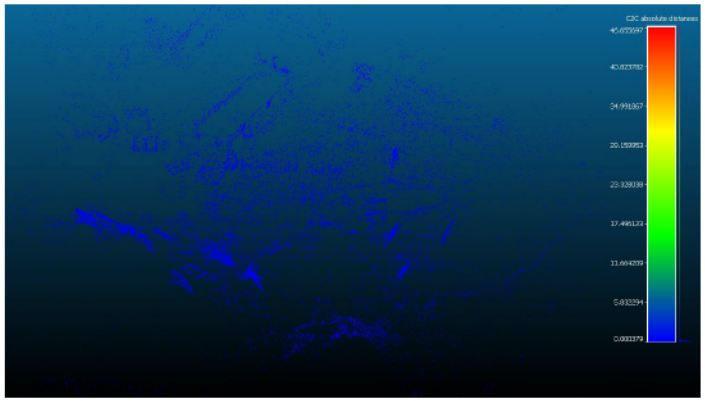


(c) R2D2+NN-Mutual

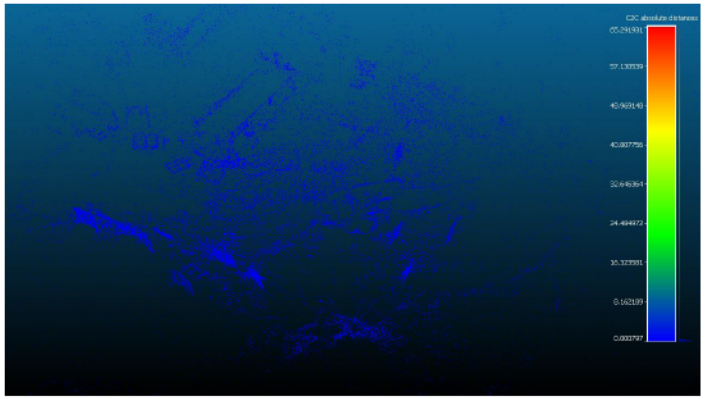
Figure 4.33: Private — Construction Site Outdoor: R2D2 Reconstructions



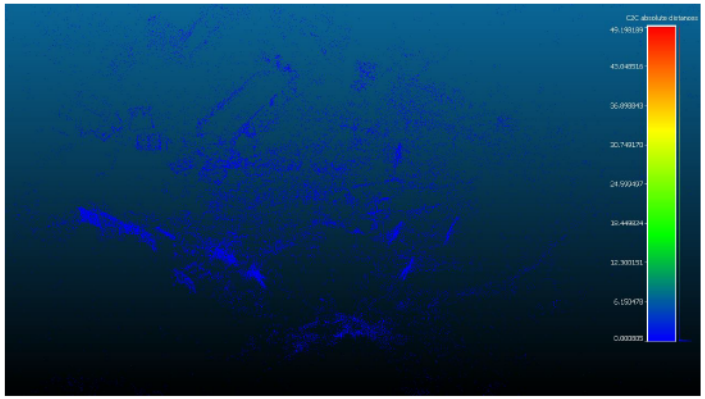
(a) SOSNet+NN-Distance



(b) SOSNet+NN-Ratio

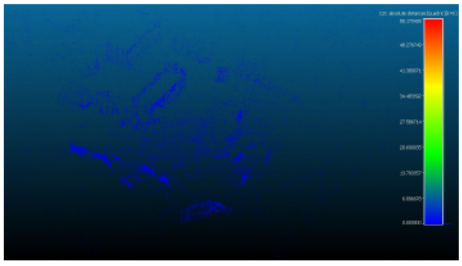


(c) SOSNet+NN-Mutual

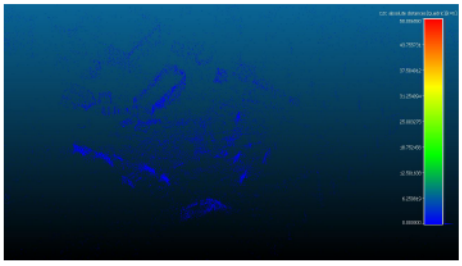


(d) SOSNet+AdaLAM

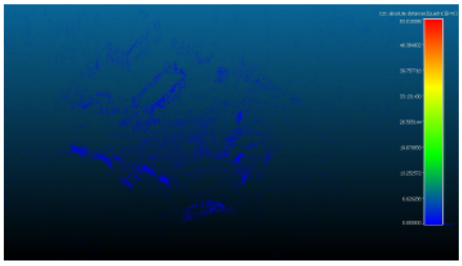
Figure 4.34: Private — Construction Site Outdoor: SOSNet Reconstructions



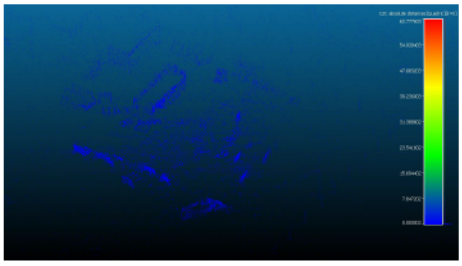
(a) SuperPoint+NN-Distance



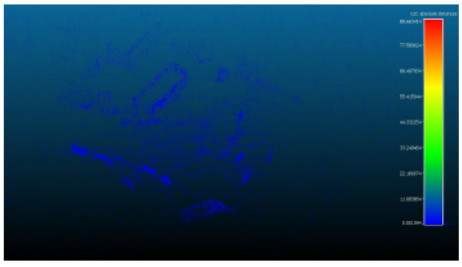
(b) SuperPoint+LightGlue



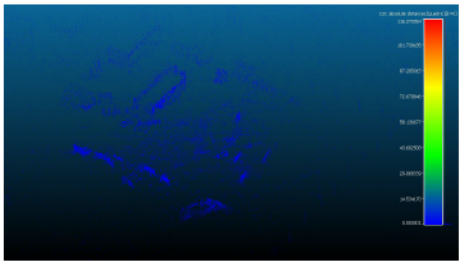
(c) SuperPoint+NN-Ratio



(d) SuperPoint+SuperGlue



(e) SuperPoint+NN-Mutual



(f) SuperPoint+SuperGlue-Fast

Figure 4.35: Private — Construction Site Outdoor: SuperPoint Reconstructions

4.2.3 Cloud-to-Cloud Distances

Table 4.7 shows the cloud-to-cloud distances for the indoor scenes, providing insights into the accuracy of the 3D reconstructions.

Table 4.7: Cloud-to-Cloud Distances for Outdoor Scenes

			Private — Construction Site Outdoor			Hilti — Construction Site Outdoor 1		
Baseline	**Extractor**	**Matcher**	**ICP Scale**	**Mean**	**STD**	**ICP Scale**	**Mean**	**STD**
SIFT	AKAZE	NN-BruteForce	1.00	0.19	0.95	1.00	0.09	0.32
SIFT	ORB	NN-BruteForce	1.00	0.16	0.80	-	-	-
SIFT	D2-Net	NN-Mutual	1.00	0.25	1.02	1.00	0.10	0.58
SIFT	D2-Net	NN-Ratio	1.00	0.14	0.47	1.00	0.11	0.39
SIFT	D2-Net	NN-Distance	1.00	0.18	0.64	1.00	0.16	0.72
SIFT	DISK	LightGlue	1.00	0.12	0.43	1.00	0.09	0.72
SIFT	DISK	NN-Mutual	1.00	0.10	0.45	1.00	0.08	1.13
SIFT	DISK	NN-Ratio	1.00	**0.05**	**0.17**	1.00	0.09	1.06
SIFT	DISK	NN-Distance	1.00	0.06	0.27	1.00	**0.07**	0.52
SIFT	R2D2	NN-Mutual	1.00	0.14	0.62	1.00	0.10	0.58
SIFT	R2D2	NN-Ratio	1.00	0.06	0.22	1.00	0.08	0.30
SIFT	R2D2	NN-Distance	1.00	0.15	0.63	1.00	0.10	0.65
SIFT	SOSNet	Adalam	1.00	0.15	0.74	1.00	0.07	0.60
SIFT	SOSNet	NN-Mutual	1.00	0.18	0.96	1.00	0.18	1.35
SIFT	SOSNet	NN-Ratio	1.00	0.10	0.59	1.00	0.07	0.43
SIFT	SOSNet	NN-Distance	0.82	0.06	0.41	1.00	0.07	**0.25**
SIFT	SuperPoint	NN-Mutual	1.00	0.24	1.00	1.00	0.13	0.96
SIFT	SuperPoint	NN-Ratio	1.00	0.16	0.87	1.00	0.13	1.08
SIFT	SuperPoint	NN-Distance	1.00	0.21	1.07	1.00	0.16	1.25
SIFT	SuperPoint	SuperGlue	1.00	0.31	1.24	1.00	0.12	0.76
SIFT	SuperPoint	SuperGlue-Fast	1.00	0.31	1.30	1.00	0.12	0.72
SIFT	SuperPoint	LightGlue	1.00	0.29	1.10	1.00	0.12	0.75

Hilti

In Hilti's dataset, the DISK (NN-Ratio) method yielded the most accurate results, with a mean cloud-to-cloud distance of 0.07 units. However, the SOSNet with NN-Distance method exhibited the lowest variability.

Private

For the Private dataset, the method DISK (NN-Ratio) produced the most accurate results, with a mean cloud-to-cloud distance of 0.05 units and a standard deviation of 0.32 17.

4.2.4 Performance Evaluation

Table 4.8 presents the performance metrics for the indoor scenes, including Elapsed Time, Mean Runtime for Feature Extraction, Feature Matching, and Global Search, as well as CPU, RAM, GPU, and Disk usage.

Hilti

Traditional Methods. Traditional methods achieved the highest CPU usage, with a maximum of 70% and a minimum of 64%. All show a Ram usage in line with deep learning methods. SIFT, as the baseline method, exhibits a lower elapsed time of 15.94 hours but higher RAM usage at 5.84 GB and a significant feature extraction runtime of 200.21 ms while having a feature matching runtime for SIFT is 9.89 ms. ORB stands out with the shortest feature extraction runtime of 28.63 ms; however, it has the highest feature matching time at 151.73 ms. The overall elapsed time for ORB is the longest.

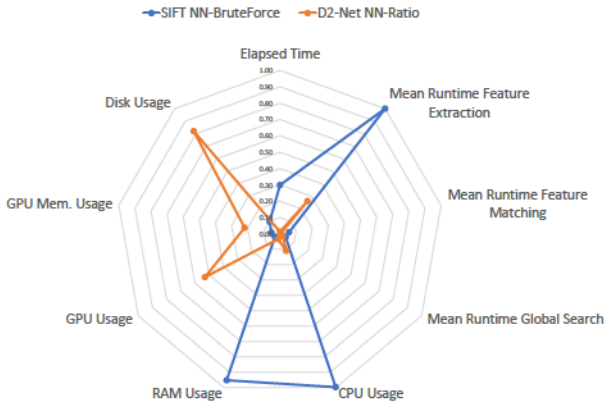
Deep Learning-Based Methods. Deep learning-based methods exhibit a range of performance metrics, with elapsed times varying from as long as 41 hours to as short as 5 hours. D2-Net with NN-Ratio showed the shortest elapsed time among deep learning methods at 5.59 hours, accompanied by a CPU and RAM usage of 12.19% and 2.88 GB, respectively. Its feature extraction and matching runtimes are efficient, at 60.09 ms and 2.03 ms. In contrast, the same extractor with NN-Distance shows slightly higher GPU usage of 1.04% and GPU memory usage of 2.52 GB. DISK, when combined with LightGlue, exhibits the longest elapsed time of 41.34 hours, with a feature extraction time of 39.76 ms and a slightly higher feature matching time of 2.82 ms. This configuration also

demonstrates notable GPU usage at 3.71% and GPU memory usage at 2.44 GB. In contrast, SuperPoint, in all its combinations, shows faster elapsed times ranging between 7 and 9 hours. And, it also has the lowest CPU and RAM usage of all methods, at the cost of significant GPU use at 13.89%.

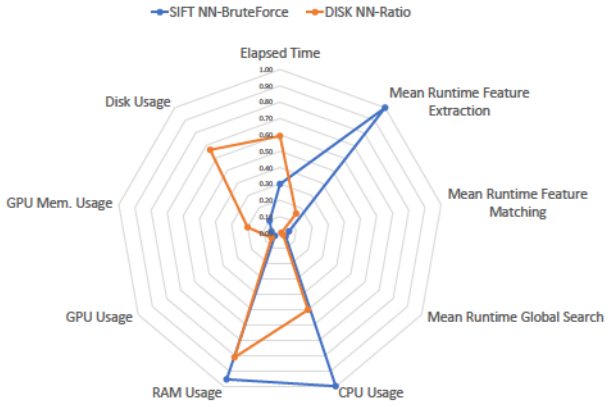
Figure 4.36, presents radar plots between the baseline and the top three deep learning methods (D2-Net, DISK, and R2D2). The values were scaled and translated into the range [0, 1] for improved visualization using min-max scaling according to the following formula:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{4.7}$$

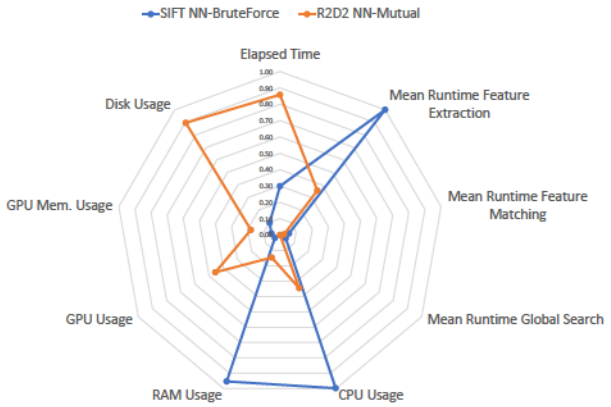
where x is the original value and x' is the scaled value. The minimum and maximum values were calculated separately for each column. The original values are provided in Table 4.8.



(a) D2-Net+NN-Ratio vs. Baseline



(b) DISK+NN-Ratio vs. Baseline



(c) R2D2+NN-Mutual vs. Baseline

Figure 4.36: Hilti — Construction Site Outdoor 1: Performance Comparison

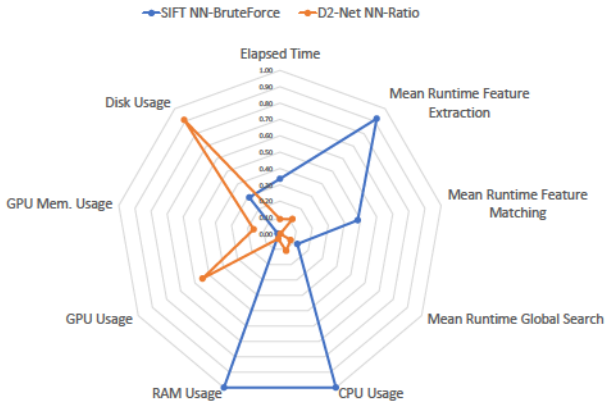
Private

Traditional Methods. SIFT, as the baseline method, exhibited relatively high resource consumption across the evaluated metrics, with an elapsed time of 0.68 hours, CPU usage of 58.41%, RAM usage of 16.26 GB, and disk usage of 1.24 GB. Although AKAZE demonstrated faster matching and elapsed times, it still recorded the highest extraction time. On the contrary, ORB showed the lowest resource usage in terms of CPU and RAM. GPU utilization remained minimal, consistent with the limited dependence on GPU resources typical of traditional methods.

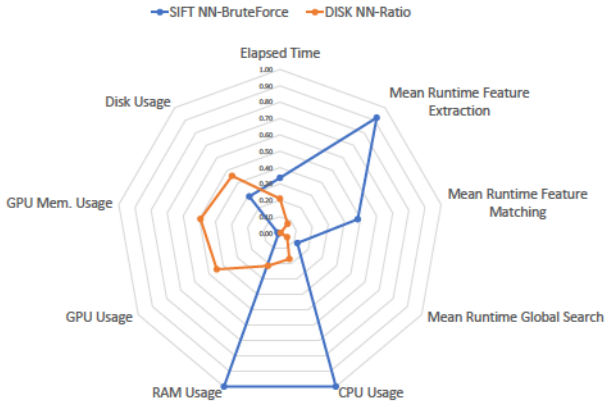
Deep Learning-Based Methods. For deep learning approaches, the elapsed times were significantly shorter than those of traditional methods, underscoring the efficiency of deep learning-based feature extraction. CPU usage was generally lower, with SOSNet (NN-Mutual) having the highest usage at 35.73%. RAM usage was relatively low, with DISK (LightGlue) reaching the maximum of 6.56 GB; despite this low RAM usage, this method also exhibited the highest GPU memory usage at 4.10 GB. In contrast, SuperPoint demonstrated exceptional efficiency in feature extraction and matching across various techniques, resulting in the shortest processing times and the lowest resource consumption among all methods. However, GPU usage was still high, due to the specialized matching techniques employed. Compared to traditional methods, deep learning approaches are more efficient in terms of processing time and resource utilization. Figure 4.37, presents radar plots between the baseline and the top three deep learning methods (D2-Net, DISK, and R2D2). The values were scaled and translated into the range [0, 1] for improved visualization using min-max scaling according to the following formula:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{4.8}$$

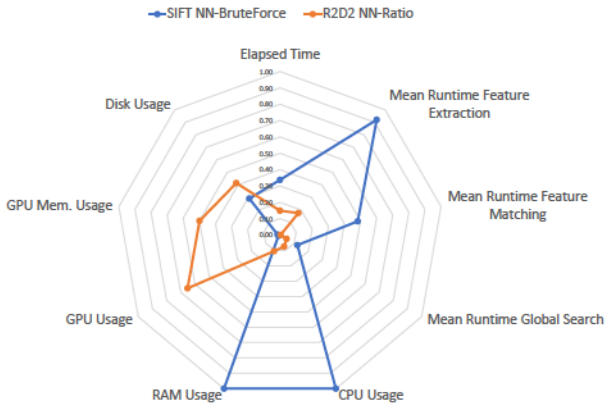
where x is the original value and x' is the scaled value. The minimum and maximum values were calculated separately for each column. The original values are provided in Table 4.8.



(a) D2-Net+NN-Ratio vs. Baseline



(b) DISK+Ratio vs. Baseline



(c) R2D2+NN-Ratio vs. Baseline

Figure 4.37: Private — Construction Site Outdoor: Performance Comparison

Table 4.8: Performance Metrics for Outdoor Scenes

		Private - Construction Site Outdoor									Hilti - Construction Site Outdoor 1								
Extractor	**Matcher**	**Elapsed Time (hr)**	**Mean Runtime Feature Extraction (ms)**	**Mean Runtime Feature Matching (ms)**	**Mean Runtime Global Search (ms)**	**CPU Usage (%)**	**RAM Usage (GB)**	**GPU Usage (%)**	**GPU Mem. Usage (GB)**	**Disk Usage (GB)**	**Elapsed Time (hr)**	**Mean Runtime Feature Extraction (ms)**	**Mean Runtime Feature Matching (ms)**	**Mean Runtime Global Search (ms)**	**CPU Usage (%)**	**RAM Usage (GB)**	**GPU Usage (%)**	**GPU Mem. Usage (GB)**	**Disk Usage (GB)**
AKAZE	NN-BruteForce	0.47	930.34	64.26	3.33	56.23	6.55	0.35	2.59	**0.30**	19.07	149.30	3.02	1.89	70.50	2.91	0.61	2.03	0.23
SIFT	NN-BruteForce	0.68	857.15	720.35	3.26	58.46	16.27	0.24	2.59	1.24	15.94	200.21	9.89	1.85	70.31	5.84	0.71	2.03	0.49
ORB	NN-BruteForce	1.73	94.00	1491.12	3.20	55.71	5.02	**0.11**	2.84	0.94	25.18	28.63	151.73	1.85	64.75	3.16	**0.23**	2.26	0.60
D2-Net	NN-Mutual	1.20	122.01	5.91	2.98	32.33	5.30	3.11	2.82	3.55	20.63	60.27	1.97	1.79	27.54	2.86	1.88	2.45	2.73
D2-Net	NN-Ratio	0.29	120.79	5.93	3.07	16.65	3.20	7.26	2.82	3.25	5.59	60.09	2.03	1.78	12.19	2.88	7.46	2.47	2.62
D2-Net	NN-Distance	0.53	122.38	5.94	3.02	17.98	4.48	4.75	2.82	3.37	20.29	59.30	1.95	1.76	36.67	3.77	1.04	2.52	2.89
DISK	LightGlue	0.98	78.25	54.35	3.10	17.04	6.56	13.20	4.10	2.02	41.34	39.76	2.82	1.75	29.46	3.51	3.71	2.44	1.92
DISK	NN-Mutual	0.81	78.61	6.94	3.07	29.71	6.20	4.23	3.33	1.95	40.28	39.76	2.80	1.75	32.38	3.67	2.53	2.68	1.95
DISK	NN-Ratio	0.48	78.31	7.04	2.97	19.40	5.67	5.94	3.32	1.78	26.61	39.76	2.78	1.79	37.75	5.38	1.03	2.42	2.15
DISK	NN-Distance	0.69	78.31	7.00	2.85	29.15	5.77	4.74	3.32	1.79	32.09	39.83	26.77	1.77	26.67	5.62	8.38	3.42	2.15
R2D2	NN-Mutual	0.99	172.24	7.07	2.83	26.35	5.41	4.49	3.58	1.81	36.19	77.82	5.28	**1.71**	27.63	3.25	6.46	2.37	2.84
R2D2	NN-Ratio	0.38	172.00	7.17	2.95	14.98	4.19	8.65	3.33	1.64	19.40	78.00	5.23	1.71	36.75	6.00	1.71	2.67	3.14
R2D2	NN-Distance	0.80	172.12	7.09	**2.76**	21.50	5.41	5.08	3.33	1.81	31.88	77.94	5.22	1.71	33.90	5.80	2.12	4.51	3.09
SOSNet	Adalam	0.21	247.19	9.63	6.48	18.69	3.51	5.66	3.04	0.38	7.44	118.23	1.55	5.03	15.38	3.67	2.32	3.69	0.21
SOSNet	NN-Mutual	0.32	253.17	1.49	6.87	35.73	3.49	2.88	3.03	0.40	8.16	117.98	1.43	5.01	40.00	3.70	1.08	3.69	0.25
SOSNet	NN-Ratio	0.18	250.00	1.65	6.20	18.04	3.53	3.61	3.03	0.34	6.46	120.36	1.64	5.04	17.48	3.71	2.80	3.67	**0.21**
SOSNet	NN-Distance	0.23	247.56	1.59	6.53	19.00	3.53	3.24	3.03	0.34	5.97	117.13	7.40	5.01	33.71	3.71	4.96	3.68	0.26
SuperPoint	NN-Mutual	0.33	11.41	**1.48**	3.04	32.21	2.80	2.56	**2.57**	0.46	7.78	10.48	1.64	1.74	**4.94**	**2.77**	5.19	1.93	0.42
SuperPoint	NN-Ratio	**0.15**	11.36	1.67	2.90	11.98	**2.77**	3.57	**2.57**	0.40	**5.13**	10.89	14.47	1.77	15.71	2.79	13.89	2.24	0.47
SuperPoint	NN-Distance	0.18	11.42	1.60	3.09	14.35	2.78	3.32	**2.57**	0.42	7.61	10.51	1.55	1.75	16.50	2.79	2.34	1.96	0.45
SuperPoint	SuperGlue	0.48	11.36	44.90	2.92	19.08	2.78	10.90	2.63	0.46	8.91	**10.44**	**1.41**	1.75	34.85	2.78	0.91	**1.90**	0.49
SuperPoint	SuperGlue-Fast	0.32	11.37	37.84	2.97	**11.46**	2.80	12.00	2.63	0.46	9.57	10.51	62.23	1.73	11.44	2.78	5.23	2.40	0.47
SuperPoint	LightGlue	0.23	**11.30**	15.33	2.89	15.50	2.80	9.07	2.63	0.46	7.59	10.48	69.04	1.75	11.17	2.78	6.27	2.73	0.47

4.3 Summary

This chapter presented the evaluation results of the 3D reconstruction methodologies across indoor and outdoor scenes, providing insights into their performance, robustness, and limitations. The analysis encompassed both indoor and outdoor scenes, offering a comprehensive understanding of how different feature extractors and matching algorithms operate under diverse conditions.

The performance of the employed 3D reconstruction methods exhibited considerable variability across different datasets. Deep learning-based techniques, such as R2D2, DISK, and D2-Net, generally outperformed traditional methods like SIFT and ORB. Specifically, these advanced techniques consistently achieved higher co-visibility ratios and produced high-quality reconstructions with more detailed textures and accurate geometries, demonstrating their efficacy in identifying and matching features across image pairs. This was particularly evident in the Private dataset, where fewer missing areas were visually identifiable. On the other hand, traditional methods such as SIFT, while robust, displayed greater variability and lower overall performance with more missing areas. This variability underscores the challenge of maintaining consistent feature matching in complex scenes, where visual overlap may be limited or unevenly distributed. This was confirmed when analyzing in the box plots for the indoor scenes, where the median co-visibility ratio was lower than in the outdoor scenes.

Overall, the evaluation of 3D reconstruction methods across diverse datasets provided valuable insights into the strengths and weaknesses of various feature extractors and matching algorithms, emphasizing the importance of selecting appropriate techniques based on the specific characteristics of the scene and the desired outcome. The next chapter presents conclusions, discussion, and future work based on the results obtained in this chapter.

5 Conclusion

This chapter provides conclusions based on the evaluations results presented in Chapter 4, as well as discussions on the findings and potential future research directions. The chapter is structured into two sections as follows, Summary and Future Works. The findings are discussed in the context of the research objectives, highlighting the strengths and limitations of the 3D reconstruction methods across indoor and outdoor scenes.

5.1 Summary

The purpose of this thesis was to evaluate the performance of both traditional and deep learning-based methods for 3D reconstruction in indoor and outdoor scenes, focusing on construction sites. The evaluation was conducted using three datasets: ConSLAM, Hilti, and a Private Construction Site dataset. These datasets were selected to represent a wide range of challenging scenarios, including varying lighting conditions, feature scarcity, and occlusions. The methods evaluated include traditional feature extractors like SIFT, AKAZE, and ORB, along deep learning-based methods such as R2D2, D2-Net, SuperPoint and DISK, paired with various matching algorithms ranging from different nearest neighbors to specialized approaches like LightGlue and SuperGlue. This evaluation was motivated by the increasing demand for accurate and efficient 3D reconstruction methods in construction applications, where traditional methods may struggle to cope with the complexity and variability of real-world environments and the lack of dedicated evaluation benchmarks on construction sites. The evaluation criteria included dataset evaluation

metrics, reconstruction quality metrics, and performance evaluation metrics.

The initial phase of the evaluation, presented as the dataset evaluation, highlights the variability and performance of different methods in terms of co-visibility ratios, which provide insights into the reliability and consistency of scene visibility overlap across image pairs. The second phase, focused on the reconstruction evaluation, providing insights into the quality and completeness of the reconstructions by using cloud-to-cloud distances with respect a baseline and visual inspection of the point clouds, as well as, the statistics for each method in terms of registered images, points, observations, mean track length, mean observations per image, mean reprojection error, mean number of keypoints, and mean number of matches. Lastly, the performance evaluation, consisted of analyzing the resource usage and efficiency of the methods in terms of elapsed time, mean runtime for feature extraction and feature matching, as well as CPU, RAM, GPU, and Disk usage. Key findings from the evaluations include:

- Deep learning-based methods consistently outperformed traditional methods in terms of co-visibility ratios and reconstruction quality, particularly in challenging scenarios with limited visual overlap and complex lighting conditions. This was more evident in the Hilti dataset, where over and under exposure were common and the particularly in sections where featureless areas like walls and floors were present.

- One thing to note about the co-visibility ratio is that it is not a direct measure of the quality of the reconstruction, as multiple methods that fall on the higher end of the spectrum in terms of co-visibility ratio, like AKAZE, ORB, and SIFT on the Hilti dataset for the indoor scenes, did not produce the best reconstructions. An alternative metric could be the cloud-to-cloud distances, which provide a more direct measure of the accuracy of the 3D reconstructions, as long as, the reconstructions were successfully generated and are not as sparse.

- Traditional methods like SIFT and ORB demonstrated robustness and consistency

in feature matching, albeit with low overall performance. These methods were more effective in maintaining reliable scene overlap and producing accurate reconstructions in scenarios with sufficient visual features and lighting conditions not as complex, as seen in the outdoor scenes from the Private dataset. However, compared to point clouds generated by deep learning methods, they were less detailed and with a larger number of missing areas.

- Deep learning methods were more efficient in terms of processing time and resource consumption, leveraging modern hardware capabilities to enhance performance like SuperPoint, which demonstrated exceptional efficiency in feature extraction and matching thanks to the GPU acceleration.

- Deep Learning-based reconstructions were more detailed and accurate with fewer missing areas, especially within the Private dataset (outdoor scenes). This outcome can be attributed to the sophisticated feature extraction inherent in these methods, which facilitated more effective identification and matching of features across image pairs and to the datasets they were trained on. Specifically, R2D2 and DISK, trained on datasets such as MegaDepth [49] and Aachen [11], [54], were identified as the most effective methods in terms of reconstruction quality and performance within the Private dataset, indicating that the training data plays a significant role in the performance of the methods.

- The map error observed in the ConSLAM dataset highlighted the challenges of maintaining consistent feature matching in complex scenes, emphasizing the importance of selecting appropriate techniques based on the specific characteristics of the scene. This error was attributed to the similarity of the images taken in specific regions of the scene, leading to keypoints being mistakenly identified as valid matches between image pairs. Notably, a tailored extraction and matching approach like SuperPoint with SuperGlue-Fast was able to reconstruct the map successfully,

indicating that specific configurations and tuning may be necessary to achieve optimal performance in challenging scenarios. Metrics from this dataset, should be considered with caution, as the only method that was able to reconstruct the map was SuperPoint with SuperGlue-Fast, and the rest of the methods failed to reconstruct the map.

- The evaluation of the cloud-to-cloud distances provided insights into the accuracy of the 3D reconstructions, with deep learning-based methods like DISK (NN-Ratio) and R2D2 (NN-Mutual) demonstrating the best performance in terms of mean distance and standard deviation. These methods exhibited high precision and consistency. Compared to the baseline (SIFT), DISK, R2D2 and in most cases D2-Net, tend to generate point clouds with superior appearance and fewer missing areas.

- Matching techniques like Nearest Neighbor (NN) when used with deep learning showed comparable performance to tailored matching techniques like LightGlue and SuperGlue, indicating that traditional matching techniques can still be effective in certain scenarios. However, the tailored matching techniques were more consistent and robust across different datasets, suggesting that they may be more suitable for challenging scenarios with limited visual overlap and complex lighting conditions.

To conclude, this evaluation of 3D reconstruction methods across diverse datasets provided valuable insights into the strengths and limitations of various feature extractors and matching algorithms, emphasizing the importance of selecting appropriate techniques based on the specific characteristics of the scene and the desired outcome. From previous works, it is known that traditional hand-crafted methods might still perform similarly to deep learning methods in some scenarios, as shown in [28] or [27]. However, in this work, more recent deep learning methods than those evaluated in previous works were evaluated, and the results indicated that deep learning methods consistently outperformed

the baseline (SIFT) and traditional methods across all the metrics evaluated. Furthermore, deep learning methods were more efficient in terms of processing time and resource consumption, leveraging modern hardware capabilities to enhance performance.

5.2 Future works

Currently, this evaluation does not account for different configurations of the methods. Future research could include an ablation study to examine how various parameters of the extractors, matchers, and different configuration within the Incremental SfM influence the reconstruction process and performance. Additionally, as new methods based on Transformers are developed for feature extraction [68], [69] or matching [70], [71], utilizing attention maps to enhance performance, these could be evaluated similarly to the methods assessed in this study.

Furthermore, considering the complexity of the scene, ConSLAM also provided NIR Images, suggesting that a fusion technique could be employed to enhance results [72], [73]. NIR images could be integrated with RGB images to reduce the number of overexposed and underexposed areas. Moreover, evaluating these methods on datasets with varying image overlap according to the scene complexity would provide more insights into their performance in such scenarios.

Finally, a hybrid approach could be explored, where techniques are applied on a case-by-case basis or area-by-area basis to leverage their strengths and improve overall results.

References

[1] G. Marullo, L. Tanzi, P. Piazzolla, and E. Vezzetti, "6d object position estimation from 2d images: A literature review", *Multimedia Tools and Applications*, vol. 82, no. 16, pp. 24 605–24 643, 2023.

[2] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry", in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, Ieee, vol. 1, 2004, pp. I–I.

[3] O. Álvarez-Tuñón, Y. Brodskiy, and E. Kayacan, "Monocular visual simultaneous localization and mapping:(r) evolution from geometry to deep learning-based pipelines", *IEEE Transactions on Artificial Intelligence*, 2023.

[4] C. J. Iheaturu, E. G. Ayodele, and C. J. Okolie, "An assessment of the accuracy of structure-from-motion (sfm) photogrammetry for 3d terrain mapping", *Geomatics, landmanagement and landscape*, vol. 2, pp. 65–82, 2020.

[5] P. Musialski, P. Wonka, D. G. Aliaga, M. Wimmer, L. Van Gool, and W. Purgathofer, "A survey of urban reconstruction", in *Computer graphics forum*, Wiley Online Library, vol. 32, 2013, pp. 146–177.

[6] A. Corradetti, T. Seers, M. Mercuri, C. Calligaris, A. Busetti, and L. Zini, "Benchmarking different sfm-mvs photogrammetric and ios lidar acquisition methods for the digital preservation of a short-lived excavation: A case study from an area of sinkhole related subsidence", *Remote Sensing*, vol. 14, no. 20, p. 5187, 2022.

[7] J. Xue, X. Hou, and Y. Zeng, "Review of image-based 3d reconstruction of building for automated construction progress monitoring", *Applied Sciences*, vol. 11, no. 17, p. 7840, 2021.

[8] A. Khaloo, D. Lattanzi, K. Cunningham, R. Dell'Andrea, and M. Riley, "Unmanned aerial vehicle inspection of the placer river trail bridge through image-based 3d modelling", *Structure and Infrastructure Engineering*, vol. 14, no. 1, pp. 124–136, 2018.

[9] W. Jiang, Y. Zhou, L. Ding, C. Zhou, and X. Ning, "Uav-based 3d reconstruction for hoist site mapping and layout planning in petrochemical construction", *Automation in Construction*, vol. 113, p. 103 137, 2020.

[10] Z. Liu, D. Kim, S. Lee, L. Zhou, X. An, and M. Liu, "Near real-time 3d reconstruction and quality 3d point cloud for time-critical construction monitoring", *Buildings*, vol. 13, no. 2, p. 464, 2023.

[11] T. Sattler, W. Maddern, C. Toft, *et al.*, "Benchmarking 6dof outdoor visual localization in changing conditions", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8601–8610.

[12] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors", *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[13] S. A. K. Tareen and Z. Saleem, "A comparative analysis of sift, surf, kaze, akaze, orb, and brisk", in *2018 International conference on computing, mathematics and engineering technologies (iCoMET)*, IEEE, 2018, pp. 1–10.

[14] M. Dusmanu, I. Rocco, T. Pajdla, *et al.*, "D2-net: A trainable cnn for joint description and detection of local features", in *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 2019, pp. 8092–8101.

[15] J. Revaud, C. De Souza, M. Humenberger, and P. Weinzaepfel, "R2d2: Reliable and repeatable detector and descriptor", *Advances in neural information processing systems*, vol. 32, 2019.

[16] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4938–4947.

[17] M. Tyszkiewicz, P. Fua, and E. Trulls, "Disk: Learning local features with policy gradient", *Advances in Neural Information Processing Systems*, vol. 33, pp. 14 254–14 265, 2020.

[18] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description", in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.

[19] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction", *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–13, 2017.

[20] A. Martell, H. A. Lauterbach, A. Nuchtcer, *et al.*, "Benchmarking structure from motion algorithms of urban environments with applications to reconnaissance in search and rescue scenarios", in *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, IEEE, 2018, pp. 1–7.

[21] S. Ruano and A. Smolic, "A benchmark for 3d reconstruction from aerial imagery in an urban environment.", in *VISIGRAPP (5: VISAPP)*, 2021, pp. 732–741.

[22] E. K. Stathopoulou, M. Welponer, and F. Remondino, "Open-source image-based 3d reconstruction pipelines: Review, comparison and evaluation", *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLII-2/W17*, pp. 331–338, 2019.

[23] A. Keyvanfar, A. Shafaghat, and M. S. Rosley, "Performance comparison analysis of 3d reconstruction modeling software in construction site visualization and mapping", *International Journal of Architectural Computing*, vol. 20, no. 2, pp. 453–475, 2022.

[24] B. Fan, Q. Kong, X. Wang, *et al.*, "A performance evaluation of local features for image-based 3d reconstruction", *IEEE Transactions on Image Processing*, vol. 28, no. 10, pp. 4774–4789, 2019.

[25] Y. Tian, B. Fan, and F. Wu, "L2-net: Deep learning of discriminative patch descriptor in euclidean space", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 661–669.

[26] K. Bartol, D. Bojanić, T. Pribanić, T. Petković, Y. Donoso, and J. Mas, *On the comparison of classic and deep keypoint detector and descriptor methods. arxiv*, 2020.

[27] F. Remondino, F. Menna, and L. Morelli, "Evaluating hand-crafted and learning-based features for photogrammetric applications", *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 43, pp. 549–556, 2021.

[28] J. L. Schonberger, H. Hardmeier, T. Sattler, and M. Pollefeys, "Comparative evaluation of hand-crafted and learned local features", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1482–1491.

[29] K. Karsch, M. Golparvar-Fard, and D. Forsyth, "Constructaide: Analyzing and visualizing construction sites through photographs and building models", *ACM Transactions on Graphics (TOG)*, vol. 33, no. 6, pp. 1–11, 2014.

[30] Y.-m. Wei, L. Kang, B. Yang, and L.-d. Wu, "Applications of structure from motion: A survey", *Journal of Zhejiang University SCIENCE C*, vol. 14, no. 7, pp. 486–494, 2013.

[31] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.

[32] D. Barath, D. Mishkin, I. Eichhardt, I. Shipachev, and J. Matas, "Efficient initial pose-graph generation for global sfm", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 546–14 555.

[33] Y. Chen, S. Shen, Y. Chen, and G. Wang, "Graph-based parallel large scale structure from motion", *Pattern Recognition*, vol. 107, p. 107 537, 2020.

[34] B. Bhowmick, S. Patra, A. Chatterjee, V. M. Govindu, and S. Banerjee, "Divide and conquer: A hierarchical approach to large-scale structure-from-motion", *Computer Vision and Image Understanding*, vol. 157, pp. 190–205, 2017.

[35] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo", in *European Conference on Computer Vision (ECCV)*, 2016.

[36] J. M. Martínez-Otzeta, I. Rodríguez-Moreno, I. Mendialdua, and B. Sierra, "Ransac for robotic applications: A survey", *Sensors*, vol. 23, no. 1, p. 327, 2022.

[37] X. X. Lu, "A review of solutions for perspective-n-point problem in camera pose estimation", in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1087, 2018, p. 052 009.

[38] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis", in *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings*, Springer, 2000, pp. 298–372.

[39] J. Wu, Z. Cui, V. S. Sheng, P. Zhao, D. Su, and S. Gong, "A comparative study of sift and its variants", *Measurement science review*, vol. 13, no. 3, pp. 122–131, 2013.

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

[41] D. G. Lowe, *Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image*, US Patent 6,711,293, Mar. 2004.

[42] M. Trzeciak, K. Pluta, Y. Fathy, *et al.*, "Conslam: Periodically collected real-world construction dataset for slam and progress monitoring", in *European Conference on Computer Vision*, Springer, 2022, pp. 317–331.

[43] 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.

[44] 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.

[45] 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.

[46] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points", in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, IEEE, vol. 1, 2001, pp. 525–531.

[47] P. F. Alcantarilla and T. Solutions, "Fast explicit diffusion for accelerated features in nonlinear scale spaces", *IEEE Trans. Patt. Anal. Mach. Intell*, vol. 34, no. 7, pp. 1281–1298, 2011.

[48] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "Kaze features", in *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12*, Springer, 2012, pp. 214–227.

[49] Z. Li and N. Snavely, "Megadepth: Learning single-view depth prediction from internet photos", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2041–2050.

[50] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation", in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, Springer, 2015, pp. 234–241.

[51] K. He, Y. Lu, and S. Sclaroff, "Local descriptors optimized for average precision", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 596–605.

[52] D. Novotny, S. Albanie, D. Larlus, and A. Vedaldi, "Self-supervised learning of geometrically stable features through probabilistic introspection", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3637–3645.

[53] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5173–5182.

[54] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt, "Image retrieval for image-based localization revisited.", in *BMVC*, vol. 1, 2012, p. 4.

[55] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", *arXiv preprint arXiv:1409.1556*, 2014.

[56] Y. Tian, X. Yu, B. Fan, F. Wu, H. Heijnen, and V. Balntas, "Sosnet: Second order similarity regularization for local descriptor learning", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 11 016–11 025.

[57] Y. Jin, D. Mishkin, A. Mishchuk, *et al.*, "Image matching across wide baselines: From paper to practice", *International Journal of Computer Vision*, vol. 129, no. 2, pp. 517–547, 2021.

[58] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 716–12 725.

[59] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5297–5307.

[60] M. R. Abbasifard, B. Ghahremani, and H. Naderi, "A survey on nearest neighbor search methods", *International Journal of Computer Applications*, vol. 95, no. 25, 2014.

[61] L. Cavalli, V. Larsson, M. Oswald, T. Sattler, and M. Pollefeys, "Adalam: Revisiting handcrafted outlier detection. arxiv 2020", *arXiv preprint arXiv:2006.04250*,

[62] R. Sinkhorn and P. Knopp, "Concerning nonnegative matrices and doubly stochastic matrices", *Pacific Journal of Mathematics*, vol. 21, no. 2, pp. 343–348, 1967.

[63] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport", *Advances in neural information processing systems*, vol. 26, 2013.

[64] G. Peyré, M. Cuturi, *et al.*, "Computational optimal transport: With applications to data science", *Foundations and Trends® in Machine Learning*, vol. 11, no. 5-6, pp. 355–607, 2019.

[65] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys, "Lightglue: Local feature matching at light speed", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 17 627–17 638.

[66] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey", *ACM computing surveys (CSUR)*, vol. 54, no. 10s, pp. 1–41, 2022.

[67] M. Helmberger, K. Morin, B. Berner, N. Kumar, G. Cioffi, and D. Scaramuzza, "The hilti slam challenge dataset", *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7518–7525, 2022.

[68] M. Oquab, T. Darcet, T. Moutakanni, *et al.*, "Dinov2: Learning robust visual features without supervision", *arXiv preprint arXiv:2304.07193*, 2023.

[69] H. Chen, Z. Luo, L. Zhou, *et al.*, "Aspanformer: Detector-free image matching with adaptive span transformer", in *European Conference on Computer Vision*, Springer, 2022, pp. 20–36.

[70] H. Jiang, A. Karpur, B. Cao, Q. Huang, and A. Araujo, "Omniglue: Generalizable feature matching with foundation model guidance", *arXiv preprint arXiv:2405.12979*, 2024.

[71] Q. Wang, J. Zhang, K. Yang, K. Peng, and R. Stiefelhagen, "Matchformer: Interleaving attention in transformers for feature matching", in *Proceedings of the Asian Conference on Computer Vision*, 2022, pp. 2746–2762.

[72] J. Ying, C. Tong, Z. Sheng, *et al.*, "Region-aware rgb and near-infrared image fusion", *Pattern Recognition*, vol. 142, p. 109 717, 2023.

[73] D. Zou, B. Yang, Y. Li, X. Zhang, and L. Pang, "Visible and nir image fusion based on multiscale gradient guided edge-smoothing model and local gradient weight", *IEEE Sensors Journal*, vol. 23, no. 3, pp. 2783–2793, 2023.