remote sensing

MDPI

Article

Building Better Models: Benchmarking Feature Extraction and Matching for Structure from Motion at Construction Sites

Carlos Roberto Cueto Zumaya [1], Iacopo Catalano [1] and Jorge Peña Queralta [1,2,*]

[1] Department of Computing, Faculty of Technology, University of Turku, 20014 Turku, Finland; carlos.r.cuetozumaya@utu.fi (C.R.C.Z.); iacopo.catalano@utu.fi (I.C.)
[2] Institute of Robotics and Intelligent Systems, ETH Zurich, 8092 Zurich, Switzerland
* Correspondence: jorge.penaqueralta@hest.ethz.ch

Abstract: The popularity of Structure from Motion (SfM) techniques has significantly advanced 3D reconstruction in various domains, including construction site mapping. Central to SfM, is the feature extraction and matching process, which identifies and correlates keypoints across images. Previous benchmarks have assessed traditional and learning-based methods for these tasks but have not specifically focused on construction sites, often evaluating isolated components of the SfM pipeline. This study provides a comprehensive evaluation of traditional methods (e.g., SIFT, AKAZE, ORB) and learning-based methods (e.g., D2-Net, DISK, R2D2, SuperPoint, SOSNet) within the SfM pipeline for construction site mapping. It also compares matching techniques, including SuperGlue and LightGlue, against traditional approaches such as nearest neighbor. Our findings demonstrate that deep learning-based methods such as DISK with LightGlue and SuperPoint with various matchers consistently outperform traditional methods like SIFT in both reconstruction quality and computational efficiency. Overall, the deep learning methods exhibited better adaptability to complex construction environments, leveraging modern hardware effectively, highlighting their potential for large-scale and real-time applications in construction site mapping. This benchmark aims to assist researchers in selecting the optimal combination of feature extraction and matching methods for SfM applications at construction sites.

Keywords: structure from motion; benchmark; feature extraction; feature matching; 3D reconstruction; construction sites

Citation: Cueto Zumaya, C.R.; Catalano, I.; Queralta, J.P. Building Better Models: Benchmarking Feature Extraction and Matching for Structure from Motion at Construction Sites. *Remote Sens.* **2024**, *16*, 2974. https://doi.org/10.3390/rs16162974

Academic Editors: Ayman F. Habib, Fangning He, Hongzhou Yang, Shengjun Tang and Ding Ma

Received: 10 June 2024
Revised: 4 August 2024
Accepted: 10 August 2024
Published: 14 August 2024

1. Introduction

Benchmarks serve as essential tools for objectively evaluating and comparing algorithms, fostering advancements in various fields of computer vision. By providing standardized datasets and evaluation metrics, benchmarks have proven instrumental in pushing the boundaries of the state of the art. This is particularly evident in 3D map reconstruction, where the increasing availability and affordability of camera sensors have prompted the adoption of Structure from Motion (SfM) techniques.

SfM, a photogrammetry technique capable of generating impressive 3D scene models from image sets [1–4], finds application across diverse domains such as robotics and augmented reality for tasks such as robot navigation or pose estimation [5]. Its adaptability to varying image sets and environments underscores its versatility. SfM facilitates the creation of 3D models, enhancing spatial information planning, monitoring, quality control, and asset visualization [6,7]. This innovation presents a transformative alternative to costly methods like Terrestrial Laser Scanning (TLS), making spatial reconstruction more accessible [8–10].

However, SfM's efficacy hinges on factors like feature quality and matching accuracy. Traditionally, descriptors such as SIFT [11], SURF [12], AKAZE [13], and ORB [14], alongside matching techniques like brute-force or nearest neighbor approaches, have dominated SfM

feature extraction and matching [15–18]. Yet, challenges arise in complex scenarios, such as low-light conditions or dynamic environments, when more complex transformations are present, revealing limitations in these methods.

This prompted an exploration into learned features and matching techniques, exhibiting promise in various applications [19–22]. Learned methods leverage neural networks (NNs) to automatically learn optimal feature representations directly from the data, improving the ability to handle even more complex variations in image conditions, such as occlusions and viewpoint changes. By combining learned features like SuperPoint [23] with tailored techniques like SuperGlue [21], state-of-the-art performance can be achieved. However, the superiority of learned methods over traditional ones remains a subject of inquiry, particularly in challenging environments like construction sites.

Current SfM benchmarks have limitations regarding feature extraction and matching evaluation [17,18,24,25]:

(i) **Metric limitations:** Existing metrics like accuracy and completeness might not fully capture the impact of feature extractors and matchers on reconstruction quality. More nuanced metrics are needed to assess factors like efficiency, robustness to challenging conditions (e.g., low-quality images), and the influence on downstream tasks within the SfM pipeline.

(ii) **Dataset limitations:** Existing benchmarks often focus on specific datasets (e.g., HPatches [26]), which are often limited in scope, featuring mostly planar scenes with consistent illumination and texture. While these datasets are useful for evaluating specific aspects of feature extractors and matchers, they might not generalize well to real-world scenarios. More diverse datasets are needed to evaluate the performance of feature extractors and matchers across different environments and conditions.

(iii) **Limited reporting:** Benchmarks often lack focus on how previous work specifies and chooses feature extractors and matchers. Ideally, benchmarks should encourage researchers to report their choices and configurations for better interpretability and comparison of results. Addressing these limitations can lead to more informative benchmarks and advance the field of SfM.

(iv) **Computational constraints:** Benchmarks often do not consider the computational efficiency of feature extractors and matchers, which is crucial for real-time or large-scale applications. Methods that perform well in terms of accuracy may be impractically slow for certain applications.

To address these limitations, this paper makes the following core contributions:

(i) **Comprehensive evaluation:** We provide a thorough evaluation of traditional methods and learning-based methods within the SfM pipeline for construction site mapping. This evaluation covers both feature extraction and matching stages, including a comparison of traditional and learned-based methods.

(ii) **Diverse benchmarking:** Our study specifically focuses on the performance of these methods in challenging environments typical of construction sites, characterized by complex lighting conditions, occlusions, dynamic objects, and varying textures. These datasets better represent real-world scenarios, extending the limitations of existing benchmarks that often use overly controlled or simplistic datasets.

(iii) **Evaluation metrics:** We utilize a comprehensive set of metrics to evaluate the impact of feature extractors and matchers on reconstruction quality, efficiency, and robustness. These metrics provide a more nuanced understanding of the performance of different combinations of methods.

The objectives of this paper are as follows:

(i) Assess how learned-based methods enhance 3D reconstruction quality compared to traditional methods.

(ii) Determine the optimal scenarios for learned-based methods in 3D reconstruction.

(iii) Evaluate the out-of-the-box performance of various feature extraction and matching methods to provide practical insights for end-users and real-world applications without extensive parameter tuning.
(iv) Evaluate the applicability of learned-based methods in challenging environments like construction sites.

The remainder of this document is organized as follows: Section 2 reviews existing benchmarks and applications, followed by their limitations. Section 3 describes the methods to be evaluated. In Section 4, we introduce the methodology and experiments, with results presented in Section 5. Lastly, Section 6 concludes the work and outlines future research directions.

2. Related Works

2.1. Existing Benchmarks on Feature Extractors and Matchers

Evaluations of feature matching methods for image matching tasks may not be sufficiently generalized for image-based 3D reconstruction tasks. In line with this study, Schonberger et al. [18] explored the performance of various local features within image-based 3D reconstruction systems. The authors provided an experimental evaluation of learned and advanced handcrafted feature descriptors, confirming that while learned descriptors often outperformed traditional ones (e.g., SIFT) on image-based 3D reconstruction, advanced versions of handcrafted descriptors performed on par or better than learned ones, especially in more complex SfM scenarios [18]. In contrast, Fan et al. [27] proposed a comprehensive performance comparison of different combinations of keypoints and descriptors in image-based 3D reconstruction; the study encompassed recent advancements in both handcrafted and learning-based features. The findings indicated that binary features were capable of reconstructing scenes from controlled image sequences in a fraction of the time required by float-type features (e.g., SIFT); however, float-type features demonstrated a distinct advantage over binary ones in large-scale image sets with numerous distracting images [27].

It is important to note that none of these studies used construction sites or similar environments for their evaluations, they were composed of a large collection of unordered images from the internet, which is common for general 3D reconstruction scenarios but may not accurately represent the specific challenges posed by construction sites. For a more extensive review of prior work on the evaluation of feature extractors and matchers in other domains, please refer to [28–30].

2.2. Existing Benchmarks on SfM

This section covers a review of the literature both in the wider area of benchmarking image-based feature extraction and matching, as well as in benchmarking SfM-based reconstruction in particular. We also cover a smaller set of related literature that specifically targets construction sites.

Several benchmark studies have systematically compared 3D reconstruction methods, focusing on different aspects such as accuracy, completeness, and robustness. For instance, Knapitsch et al. [25] evaluated the performance of various SfM and Multi-View Stereo (MVS) reconstruction pipelines over a collection of statues, real-scale objects, and large-scale buildings. The accuracy and completeness of the reconstruction were evaluated by measuring how closely reconstructed points lay to the ground truth and to what extent all ground-truth points were covered, respectively. Similarly, Ruano and Smolic [31] conducted an evaluation of final 3D reconstructions in urban environments using aerial images across multiple pipelines. Their study employed precision (P), recall (R), and F-score (F) as key metrics for assessing the performance of these reconstruction pipelines.

In contrast, Martell et al. [24] examined the accuracy and runtime of specific SfM pipelines in urban environments; the evaluation utilized metrics such as total runtime and cloud-to-cloud error distance, with comparisons made against laser scanner models. Lastly, Jin et al. [17] conducted a thorough investigation into image matching, introducing a

benchmark designed to evaluate local features and robust estimation methods based on the precision of reconstructed camera poses. The benchmark aimed to standardize a pipeline for the direct comparison of different methods and configurations, revealing that properly tuned classical solutions can outperform contemporary learning-based approaches.

However, these studies often overlook the specific role of feature extractors and matchers in the reconstruction process.

2.3. Applications of SfM in Construction

In the construction industry, SfM techniques have been utilized for tasks such as site analysis, monitoring, and visualization. For instance, Karsch et al. [6] introduced ConstructAide, a tool that enhances construction site analysis using images alongside 3D building models, allowing users to explore the construction site over time to monitor the progress of construction, assess errors, and create photorealistic architectural visualizations; however, the authors acknowledged limitations in handling large datasets and complex scenes. Similarly, a benchmark study by Corradetti et al. [32] compared various SfM and MVS techniques using consumer-grade mobile devices. The study highlighted the trade-offs between accessibility and accuracy in the context of digital preservation of short-lived excavations. Lastly, Khaloo et al. [7] conducted a comprehensive study on the use of UAVs for bridge inspections using a Dense Structure-from-Motion (DSfM), demonstrating the efficacy of UAV inspections in providing detailed and accurate 3D representations. Despite challenges such as image noise and environmental factors, the study found that the UAV-based method with DSfM outperformed traditional laser scanning (LIDAR) in terms of model completeness and detail resolution.

These studies underscore the pivotal role of emerging technologies in bridging the gap between theoretical research and practical application. However, limitations like computational cost and sensitivity to image quality remain challenges in the field.

3. Evaluated Methods

This section covers the background introduction of the different methods that we benchmarked in this study, with a selection of relevant feature extraction and feature matching methods.

3.1. Feature Extraction Techniques

We start by describing both traditional and learning-based feature extraction methods from a variety of computer vision applications.

SIFT [11] is one of the most renowned methods for extracting distinctive invariant features that are robust to changes in scale, rotation, and illumination and can withstand significant affine distortion and noise. It identifies and describes keypoints using the Difference of Gaussian (DoG) technique in a multi-stage process. First, potential keypoints are detected by identifying local extrema in the DoG scale-space, defined as:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \quad (1)$$

where G is the Gaussian blur function, I is the image, and σ is the scale parameter. These potential keypoints are then refined by filtering out those with low contrast or poorly defined edges. In the next stage, each keypoint is assigned a dominant orientation based on local gradient directions, ensuring rotation invariance.

ORB [14], introduced as a fast alternative to SIFT and SURF while maintaining good accuracy, is based on the FAST detector [33] and the BRIEF descriptor [34]. First, FAST detects corners in the image by using a pixel intensity comparison, then a Harris corner detection [35], which is a score based on the local gradient changes around the pixel, is applied to find the top K corners.

AKAZE [13] leverages non-linear scale spaces for feature detection and description, using a faster approach than KAZE [36]. It employs Fast Explicit Diffusion (FED) schemes to efficiently construct the non-linear scale space through a sequence of n explicit diffusion

steps with varying time steps. 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:

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

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. Similar to SIFT, AKAZE finds the dominant orientation in a circular area of radius r to ensure rotation invariance.

D2-Net [19] is a novel approach to local feature detection and description using a single Convolutional Neural Network (CNN) for both tasks. Unlike traditional methods that detect features first and then describe them, D2-Net postpones keypoint detection until after creating a dense image representation, leading to more stable keypoints based on higher-level information. By combining detection and description, D2-Net typically performs better under difficult conditions, such as significant changes in illumination and weakly textured scenes. The model is trained using pixel correspondences from large-scale SfM reconstructions.

DISK (DIScrete Keypoints) [22] is a novel approach for learning local features using policy gradients. The network leverages reinforcement learning to optimize for a high number of correct feature matches, utilizing a probabilistic model that closely aligns training and inference regimes. DISK extracts dense and discriminative keypoints and descriptors, then builds a distribution over feature matches across images. The learning process is guided by assigning positive and negative rewards based on the geometric ground truth.

R2D2 [20] is a method for local feature detection and description designed to enhance the repeatability and reliability of keypoints. Unlike other methods that handle detection and description separately, R2D2 combines these processes using a CNN that predicts keypoint locations, descriptors, and a reliability map simultaneously. This approach ensures that the selected keypoints are both repeatable and reliable for matching. The authors introduce an unsupervised loss based on Average Precision (AP) [37,38], which promotes repeatability and sparsity while optimizing descriptor reliability. R2D2 outperforms state-of-the-art techniques on benchmarks like HPatches [26] and the Aachen Day-Night localization dataset [39,40].

SuperPoint [23] uses a Fully Convolutional Network (FCN) and self-supervised training to detect interest points and compute descriptors in a single pass. It introduces Homographic Adaptation, a method that enhances the repeatability and adaptability of interest point detection through multiscale and multi-homography transformations. Trained on the MS-COCO [41] dataset, SuperPoint outperforms traditional methods like SIFT and ORB in homography estimation tasks [23]. The network benefits from a self-supervised approach that pre-trains on synthetic data to create a robust detector called MagicPoint, which is further refined with real-world images.

SOSNet [42] enhances the robustness of local descriptors for image matching by using Second-Order Similarity Regularization (SOSR). Unlike other methods focused on First-Order Similarity (FOS), SOSR ensures that positive pairs have similar distances to other points in the embedding space. The authors introduce a triplet loss function combining FOS and SOSR, which significantly improves the learning of matching and non-matching descriptors. They also propose an evaluation method using the von Mises–Fisher distribution to assess descriptor space utilization, showing that SOSR enhances both intra-class compactness and inter-class dispersion. This results in better matching accuracy without adding computational overhead during matching.

3.2. Feature Matching Techniques

We now also review both traditional and learning-based methods for feature matching.

Nearest neighbor (NN) is one of the simplest yet most effective classification techniques, characterized by its long-standing history and robust performance across various applications. NN operates on the principle that similar points are typically close to each

other in the feature space. It identifies the closest data points to a query point using common distance metrics like Euclidean (for matching SIFT features), Manhattan, and Hamming (for matching AKAZE and ORB features), each suitable for different types of data characteristics and dimensionalities [28]. The effectiveness of NN relies heavily on these metrics, which define how "closeness" is measured.

To improve NN's performance and accuracy, two main threshold approaches are used: ratio-based and distance-based [28]. A ratio-based threshold evaluates match validity by comparing the distance ratio between the closest and second-closest matches, as seen in Lowe's ratio test in the SIFT algorithm [11]. This method is robust, especially in spaces with varying scales, as it relies on relative differences rather than absolute distances. A distance-based threshold uses a fixed distance to determine match validity, assuming consistent and meaningful distance measures across the dataset. However, this can be challenging if the metric distorts distances in different parts of the dataset. The choice between these approaches depends on the dataset's characteristics and the desired balance between accuracy and computational efficiency.

AdaLAM [43] revisits traditional outlier filtering methods and proposes a hierarchical pipeline that integrates local affine motion verification with sample-adaptive thresholds. This approach is optimized for modern GPUs, enabling processing times of tens of milliseconds. AdaLAM is based on three core principles: local planarity, locality, and adaptivity. Local planarity assumes that keypoints lie on approximately planar surfaces, simplifying geometric modeling to affine transformations. The locality principle ensures consistent affine transformations within small neighborhoods, enhancing verification robustness. Lastly, adaptivity allows the method to adjust thresholds based on the local match consistency and density, maintaining robustness across different geometric and scene conditions.

SuperGlue [21] utilizes graph neural networks (GNNs) and attention mechanisms to improve the matching process between two sets of local features. Unlike traditional heuristic methods, SuperGlue addresses a differentiable optimal transport problem, allowing it to consider both 3D scene understanding and feature assignments by leveraging spatial relationships and visual appearance. In SuperGlue's graph structure, keypoints are represented as nodes, with two types of edges: self-edges, connecting keypoints within the same image to capture local context, and cross-edges, connecting keypoints between images to identify potential matches. The network was trained end-to-end on image pairs from the Oxford and Paris dataset [44], learning priors over geometric transformations and 3D world regularities. SuperGlue achieves state-of-the-art results in pose estimation tasks across challenging indoor and outdoor environments, significantly outperforming existing methods in terms of accuracy and robustness while operating in real time on modern GPUs [21].

LightGlue [45] builds upon his predecessor (SuperGlue) by optimizing the model architecture in terms of speed, accuracy, and training simplicity. It uses a Transformer-based architecture [46], incorporating layers with self- and cross-attention mechanisms. LightGlue uses self-attention to gather information from local features within the same image, and cross-attention to match corresponding points between images. The model was trained on synthetic homographies and fine-tuned with the MegaDepth [47] dataset, demonstrating superior performance in homography estimation, relative pose estimation, and visual localization tasks compared to existing methods like SuperGlue.

4. Evaluation Procedure

4.1. Platform

Our experiments were conducted on a high-performance computing platform consisting of a server with the components listed in Table 1. It included an AMD Ryzen Threadripper CPU, NVME SSD storage, NVIDIA GPUs, and Ubuntu as the operating system.

Table 1. Components of the reconstruction platform.

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

4.2. Implementation

COLMAP [4] and Hierarchical Localization Toolbox (HLOC) [48] are the basis of this work. COLMAP provides a modular general-purpose SfM-MVS pipeline that incorporates Incremental SfM. The library also provides a set of evaluation metrics that can be used to assess the quality of the reconstruction. The MVS pipeline was not used in this work. On the other hand, HLOC is 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 the correspondences between images. The interaction between these two modules is described in Figure 1.

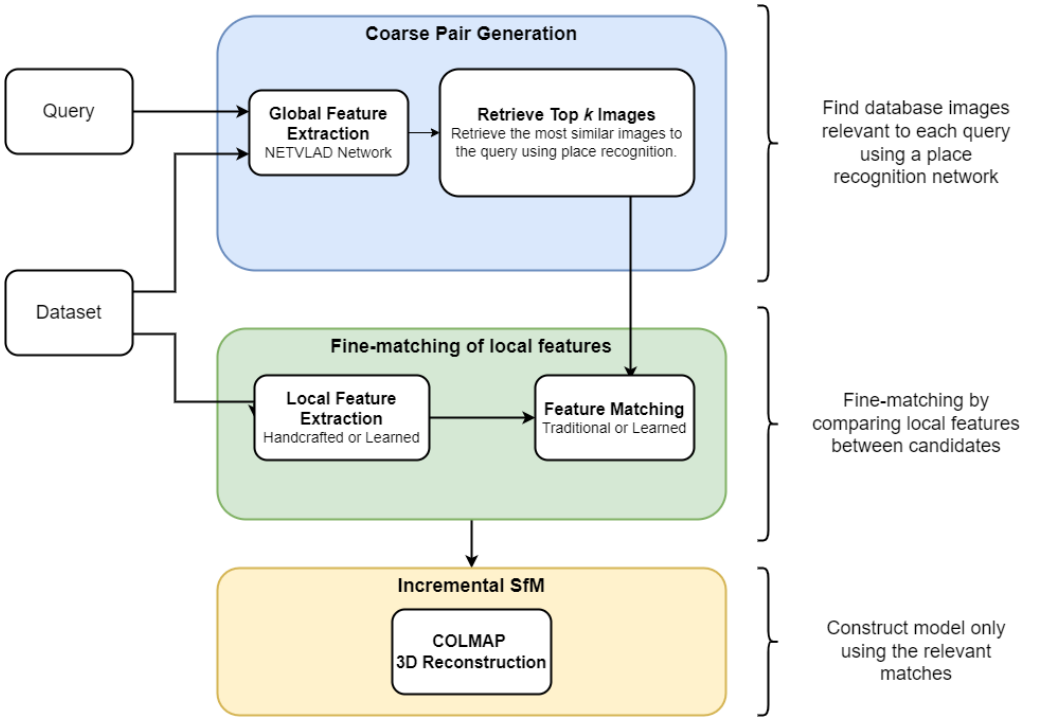


Figure 1. Reconstruction pipeline using HLOC and COLMAP.

Algorithm 1 describes the feature extraction and matching process, while Algorithm 2 outlines the 3D reconstruction process. To visually inspect and further analyze the 3D reconstructions of the samples, Cloud Compare was used.

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 50 candidate images using NETVLAD [48]
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: Three-dimensional reconstruction process

Input: Image pairs and features extracted and matched
Output: Three-dimensional 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

The combinations to be evaluated, as detailed in Table 2, were selected due to their availability in HLOC. The combinations were determined by manually testing the compatibility between feature extractors and matching algorithms. Each extractor produced a specific type of feature descriptor (either floating-point or binary) that required specific matching approaches. The methods were used in their default configurations without parameter tuning. Further implementation details can be found in the HLOC repository (https://github.com/cvg/Hierarchical-Localization, accessed on 2 June 2024) for the deep learning-based methods and in the OpenCV Library (https://docs.opencv.org/4.9.0/db/d27/tutorial_py_table_of_contents_feature2d.html, accessed on 2 June 2024) for the traditional methods.

Guided by insights from [17,48], the following decisions were made for processing and evaluating the 3D reconstructions produced by the tested methods:

(i) Feature extraction and matching were conducted using combinations of methods as detailed in Table 2.
(ii) The retrieval of candidate images through a global search was capped at 50.
(iii) A maximum of 8000 features were extracted per image. It is noted that some algorithms, such as SuperPoint, have a feature extraction limit (e.g., 4096 features).
(iv) Image resizing for deep learning methods was based on a maximum dimension of 1024 pixels to preserve the original aspect ratio.
(v) The model that registered the highest number of images was selected for detailed reconstruction quality evaluation.
(vi) For deep learning feature extractors paired with nearest neighbor matching algorithms, Mutual Check was enabled in all combinations. Similarity was computed using the dot product between descriptors, and thresholds were set at 0.8 for NN-Ratio (ratio-based filtering) and 0.7 for NN-Distance (distance-based filtering); no threshold was used for NN-Mutual.

Table 2. Feature extraction and matching combinations to be evaluated. The symbol "x" indicates that the combination will be evaluated, while "-" denotes that the combination is incompatible or will not be evaluated.

Extractors/Matchers	NN-BruteForce	NN-Ratio	NN-Distance	NN-Mutual	SuperGlue	SuperGlue-Fast	LightGlue	AdaLAM
AKAZE	x	-	-	-	-	-	-	-
ORB	x	-	-	-	-	-	-	-
SIFT	x	-	-	-	-	-	-	-
D2-Net	-	x	x	x	-	-	-	-
DISK	-	x	x	x	-	-	x	-
R2D2	-	x	x	x	-	-		-
SuperPoint	-	x	x	x	x	x	x	
SOSNet	-	x	x	x	-	-	-	x

4.3. Metrics

This section describes the specific metrics used to evaluate both environment reconstruction and algorithmic performance. The choice of metrics for evaluating 3D reconstruction at construction sites is driven by the need to balance accuracy, completeness, and performance. To assess the quality of the 3D reconstructions, a comparative analysis was conducted using SIFT as the baseline. The point clouds from both the baseline and the reconstructed model with the highest number of registered images were aligned and registered using the ICP algorithm, as implemented in Cloud Compare (https://www.cloudcompare.org/doc/wiki/index.php/ICP, accessed on 2 June 2024).

In alignment with the methodologies in the existing literature [18,24], reconstruction metrics such as accuracy and fidelity (e.g., cloud-to-cloud error distance, mean reprojection error) were employed to ensure that the 3D reconstructions closely matched the actual baseline. Metrics for completeness and detail (e.g., the number of points, number of observations, mean track length) were included to assess the density and quality of the reconstructed models, which are important for capturing intricate site details. Coverage and consistency metrics (e.g., the number of registered images, mean observations per registered image) were used to evaluate the extent of the site that was captured and the consistency of the reconstructions across different viewpoints. For performance evaluation, metrics for efficiency and scalability (e.g., elapsed time, average runtime for feature extraction and matching) were considered to evaluate the methods' suitability for real-time or large-scale scenarios typical of construction sites. Additionally, resource utilization metrics (e.g., CPU, RAM, GPU, disk usage) were selected to ensure the methods were practical for deployment on available hardware. This comprehensive set of metrics provided a robust framework for evaluating and optimizing 3D reconstruction techniques for construction site applications.

4.3.1. Reconstruction Metrics

The metrics used to evaluate the quality of the 3D reconstructions are:

Cloud-to-cloud error distance: Measures the geometric accuracy by quantifying the distance between the reconstructed points and the actual points in the target cloud, indicating the geometric accuracy of the reconstruction. In Cloud Compare (https://www.cloudcompare.org/doc/wiki/index.php/Cloud-to-Cloud_Distance, accessed on 2 June 2024), a surface in the target cloud is first approximated using a quadric model in the local region. The distance is then calculated by finding the nearest point on the model surface to each source point. The quadric model in Cloud Compare is defined with a default setting of $k = 6$ neighbors as follows:

$$Z = aX^2 + bY^2 + cXY + dX + eY + f \tag{3}$$

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 model coefficients. After calculating all

distances, the standard deviation and the mean value are calculated and used to report this metric. For construction sites, where precision is critical for the reconstruction analysis, a low cloud-to-cloud error distance indicates that the 3D model closely matches the actual site conditions.

Mean reprojection error: The average reprojection error across all 3D points that have a recorded error in pixels, calculated as

$$MeanReprojectionError = \frac{\sum_{i=1}^{m} Error(i)}{m} \tag{4}$$

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. It indicates the precision of the reconstructed points by comparing their projected positions in the images.

Number of points: The total number of 3D points in the reconstruction that are part of a 3D point track. it reflects the density of the reconstructed model, indicating how detailed and comprehensive the model is.

Number of observations: The total number of observations of 3D points across all registered images, calculated as:

$$N_{obs} = \sum_{i=1}^{n} NumPoints3D(I_i) \tag{5}$$

where $NumPoints3D(I_i)$ is the number of 3D points observed in the image I_i. It represents the total of observed 3D points across multiple images, enhancing the reliability and robustness of the reconstruction.

Mean track length: The average number of images that observe each 3D point, calculated as

$$MeanTrackLength = \frac{N_{obs}}{m} \tag{6}$$

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. It shows the stability of the 3D points by measuring how many images observe each point. Longer track lengths contribute to more stable and accurate models.

Avg. number of keypoints: The average number of keypoints extracted from the images. More keypoints generally lead to better feature matching and more detailed reconstructions.

Avg. number of matches: The average number of valid matches found between pairs of images. More matches indicate better consistency and reliability in feature detection, contributing to more accurate and detailed 3D reconstruction.

Number of registered images: The total number of images that were successfully registered in the reconstruction. It ensures that a sufficient number of images contribute to the reconstruction, providing comprehensive coverage of the site.

Mean observations per registered image: The average number of 3D point observations per registered image, calculated as

$$MeanObsPerImage = \frac{N_{obs}}{n} \tag{7}$$

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

4.3.2. Performance Metrics

The metrics used to evaluate the performance of the 3D reconstruction methods are:

Elapsed time: the total time taken to process the images and generate the 3D reconstruction, critical for practical use in time-sensitive construction projects, calculated using the Linux command `time`.

Avg. runtime feature extraction: the average time taken to extract features per image, which is the time spent on a key step, highlighting areas for optimization to improve overall efficiency.

Avg. runtime feature matching: the average time taken to match features per image pair, which is the time spent on a key step, highlighting areas for optimization to improve overall efficiency.

Avg. runtime global search: the average time taken to retrieve candidate images using NETVLAD [48], which is crucial for handling large datasets typical in construction site surveys.

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,

$$\frac{\text{Total CPU-seconds in user mode} + \text{Total CPU-seconds in kernel mode}}{\text{Elapsed real time}} \quad (8)$$

For instance, if the CPU is fully utilized, the usage on our 48-thread system would be $48 * 100 = 4800\%$. To correct this, the CPU usage was divided by the total number of threads (48).

RAM usage: The amount of RAM used during the process, calculated using the Linux command `time`. It measures the maximum resident set size, which is the peak 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`.

4.4. Datasets

Indoor and outdoor scenes were chosen to evaluate the reconstruction performance of the feature extractors and matchers, presented in Table 2. Indoor scenes are characterized by the presence of textureless regions, repetitive patterns, and occlusions, while outdoor scenes present large displacements, illumination changes, and dynamic objects. To reduce the size of the datasets, the rosbags were downsampled to a rate of 2 Hz using the *topic_tools* package and the *throttle* node (https://wiki.ros.org/topic_tools/throttle, accessed on 2 June 2024) with ROS Noetic (https://wiki.ros.org/noetic/Installation, accessed on 2 June 2024); only RGB and grayscale images were utilized.

4.4.1. Indoor Scenes

Two sequences were chosen: the "Construction Site Upper Level 1" sequence from the Hilti SLAM Challenge dataset [49] and "Sequence 2" from the ConSLAM dataset [50]. Samples of the sequences are shown in Figure 2. For the ConSLAM dataset, no near-infrared images (NIR) were utilized.

(**a**) Image samples from ConSLAM's Sequence 2.

(**b**) Image samples from Hilti's Construction Upper Level 1.

Figure 2. Examples of images from indoor scenes from the ConSLAM and Hilti Datasets.

4.4.2. Outdoor Scenes

Similarly, two sequences were chosen for outdoor scenarios: the "Construction Site Outdoor 1" sequence from the Hilti SLAM Challenge dataset and a private dataset (not publicly available) recorded at an active construction site. Both datasets were characterized by the presence of large displacements and illumination changes. Samples of the sequences are shown in Figure 3.

(**a**) Image samples from the private dataset.

(**b**) Image samples from Hilti's Construction Site Outdoor 1.

Figure 3. Examples of images from outdoor scenes from the Hilti and private datasets.

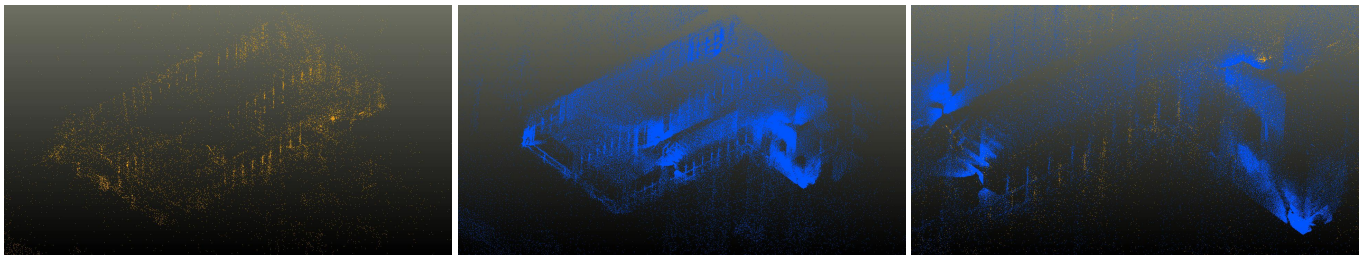
5. Results

The term "ConSLAM" is used to refer to the *ConSLAM—Sequence 2* dataset, while "Hilti" denotes the sequence *Hilti—Construction Upper Level 1* for indoor scenes and *Hilti—Construction Site Outdoor* for outdoor scenes. "Private" denotes *Private—Construction Site Outdoor*.

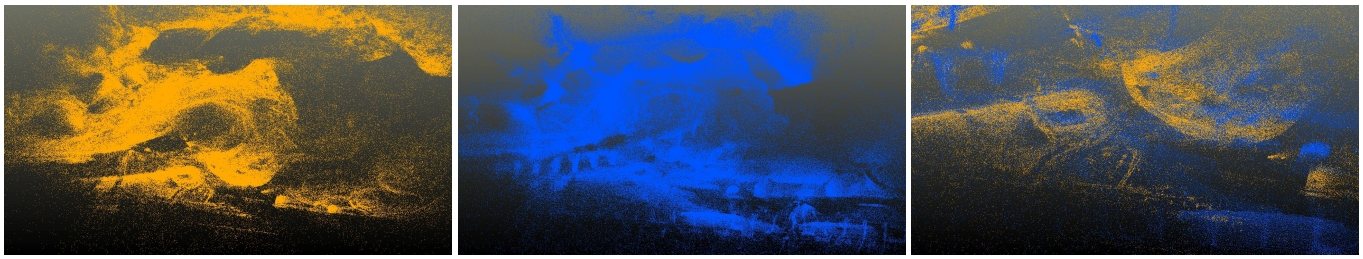
5.1. Reconstruction Evaluation

5.1.1. Visual Qualitative Comparison

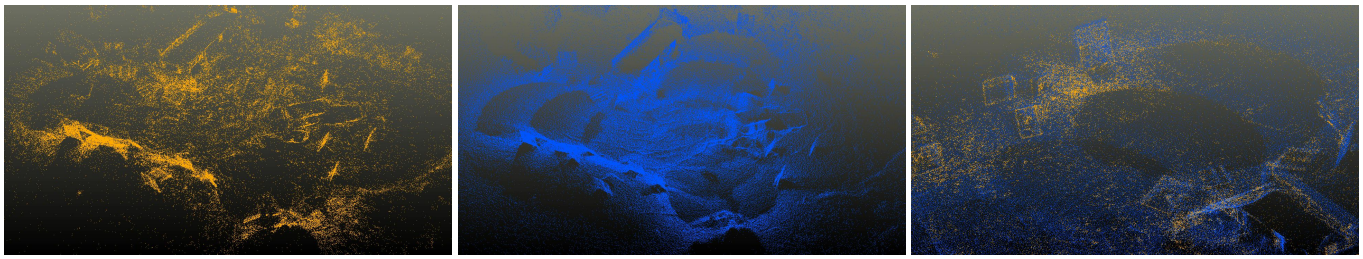
Before conducting a detailed analysis of the datasets, a preliminary visual inspection of the reconstructions generated was performed. Figure 4 illustrates a visual comparison of the 3D reconstruction quality achieved by deep learning methods (DISK + LightGlue) versus the SIFT baseline. Generally, the deep learning methods exhibited superior reconstruction quality, capturing higher detail and accuracy in complex scenes. This figure serves as a demonstration. Detailed values for the reconstruction evaluation, including metrics for indoor and outdoor scenes, can be found in Tables 3 and 5, respectively.



(**a**) Hilti's indoor dataset



(**b**) Hilti's outdoor dataset

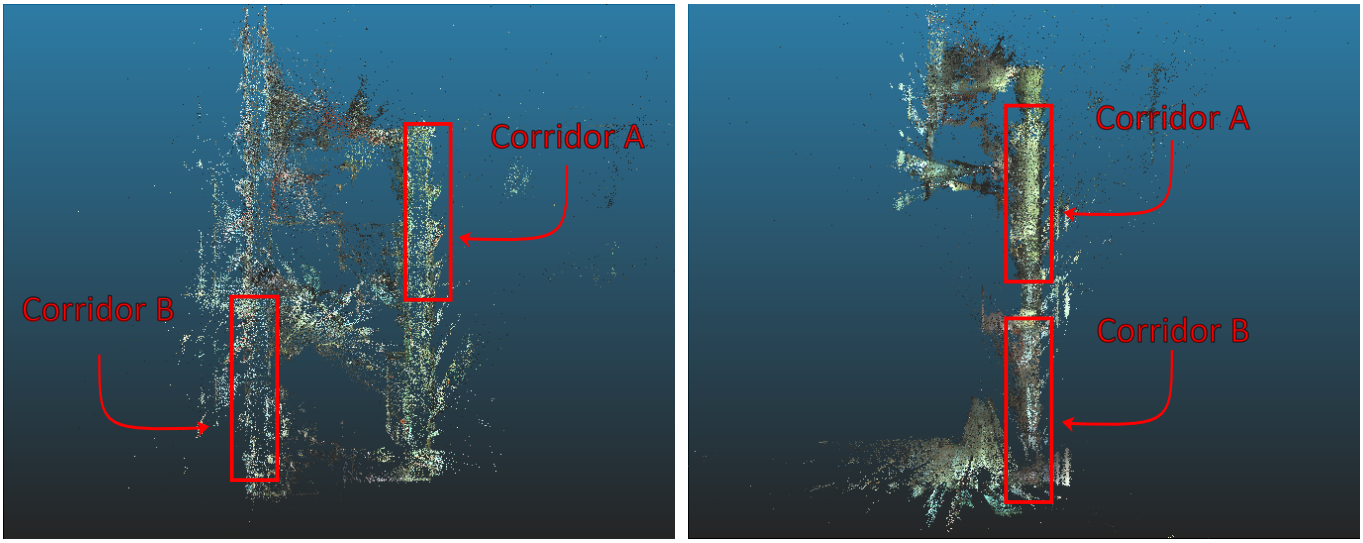


(**c**) Private dataset

Figure 4. Comparison of 3D reconstructions: The left column, highlighted in orange, shows results using the SIFT baseline. The middle column, in blue, presents reconstructions generated with the DISK+LightGlue combination. The right column features a zoomed-in section of the map, highlighting the overlap between both reconstructions.

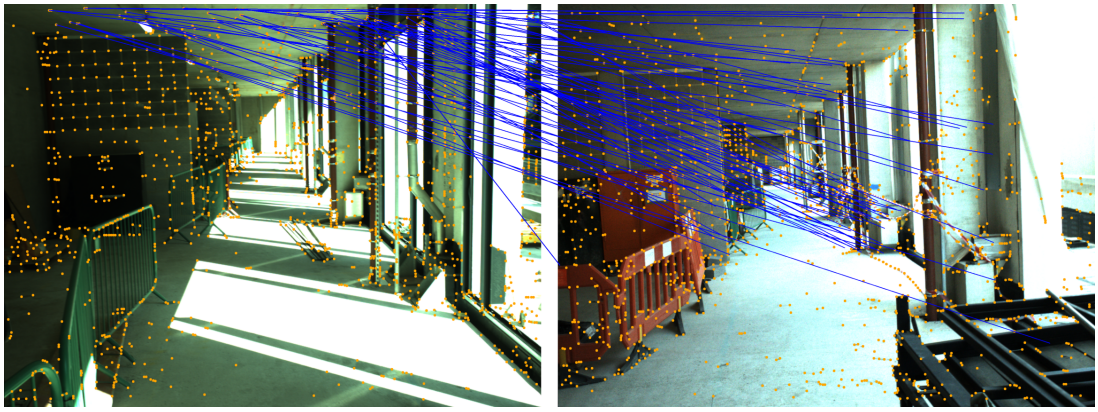
5.1.2. Corridor Misalignment

We observed a reconstruction error in the ConSLAM dataset during the evaluation of the point clouds. The map has two distinct opposite corridors, Corridor A and Corridor B, which should not be interconnected, as depicted in Figure 5a. However, both corridors were joined as a single one when reconstructing the map, as displayed in Figure 5b. This error was observed consistently across all methods except for the combination of SuperPoint with SuperGlue-Fast, suggesting that the problem was likely related to incorrect feature matching and registration, resulting in an inaccurate map reconstruction. The error was 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, as displayed in Figure 6.

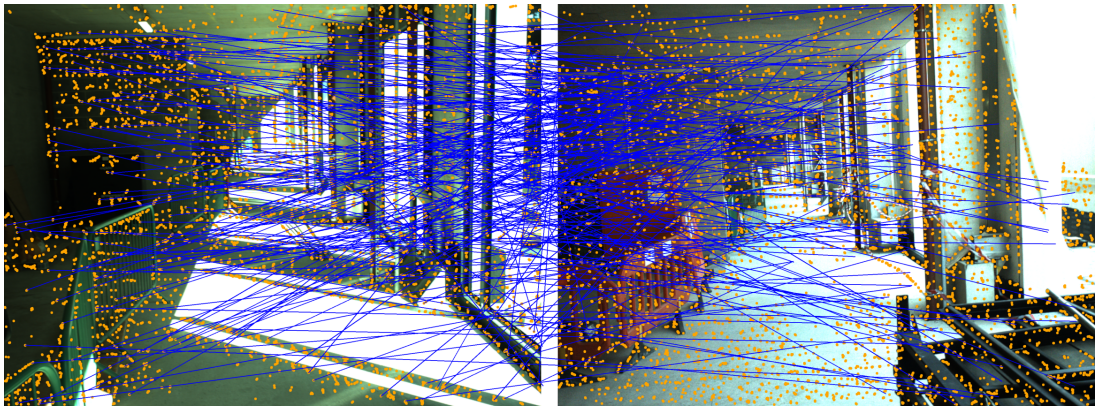


(**a**) Correct reconstruction using SuperPoint+SuperGlue-Fast. (**b**) Incorrect reconstruction using R2D2+NN-Distance.

Figure 5. Top-down point cloud view of ConSLAM with highlighted corridor misalignment.



(**a**) SuperPoint+SuperGlue-Fast—matching.



(**b**) R2D2+NN-Distance—matching.

Figure 6. Matching differences between Corridor A (Left) and Corridor B (right) in the ConSLAM dataset.

5.1.3. Indoor Scenes

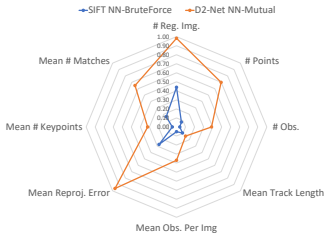
Reconstruction results for both datasets can be found in Table 3. In terms of image registration, most deep learning-based methods outperformed SIFT. For instance, DISK, R2D2, and SuperPoint consistently registered more images across datasets, demonstrating robustness and versatility in handling diverse and complex scenes. This led to improved image alignment and reconstruction coverage. Regarding points and observation counts, deep learning methods like DISK and R2D2 yielded more reconstructed points and observations, reflecting their superior ability to detect and match feature points across images. This resulted in denser, more detailed 3D reconstructions, as evidenced by increased mean track lengths and the average number of observations per image.

In contrast, SIFT lagged in producing detailed correspondences, suggesting a limitation in capturing intricate scene details. However, it generally exhibited a lower reprojection error, particularly on the ConSLAM dataset (with the lowest error), indicating that its fewer points were more accurately reconstructed. Interestingly, some deep learning methods like DISK and R2D2 also showed comparable reprojection errors, particularly on the Hilti dataset. This suggested an effective balance between point quantity and quality. Overall, learning-based methods offered clear advantages in terms of keypoints and match quantity, leading to more robust and reliable image correspondences.

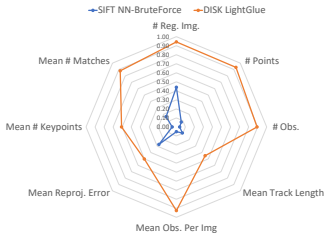
Figure 7 illustrates radar plots comparing SIFT with three of the most representative methods on the Hilti dataset. The values were scaled and translated into the range [0, 1] for improved visualization using min-max scaling according to the following formula:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{9}$$

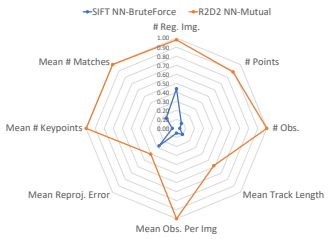
where x is the original value, and x' is the scaled value. The minimum and maximum values were calculated separately for each column. The original values are provided in Table 3.



(**a**) Reconstruction comparison of D2-Net+NN-Mutual and SIFT



(**b**) Reconstruction comparison of DISK+LightGlue and SIFT



(**c**) Reconstruction comparison of R2D2+NN-Mutual and SIFT

Figure 7. Hilti—Construction Upper Level 1: reconstruction comparison between the top three deep learning methods and SIFT.

Table 3. Reconstruction results for the indoor scenes from ConSLAM (a) and Hilti (b) datasets. The bold values highlight the configurations with the most significant performance in specific metrics, such as the highest number of registered images, points, observations, mean track length, mean observations per image, keypoints, or matches and the lowest mean reprojection error.

Extractor	Matcher	No. Reg. Img.	No. Points	No. Obs.	Mean Track Length	Mean Obs. Per Img	Mean Reproj. Error	Mean No. Key-points	Mean No. Matches
(**a**) Reconstruction results on the ConSLAM dataset.									
AKAZE	NN-BruteForce	598	162,116	1,278,721	7.89	2138.33	1.07	4,694.114	1,379.82
SIFT	NN-BruteForce	605	157,949	907,951	5.75	1500.75	**0.71**	4107.1714	1023.93
ORB	NN-BruteForce	768	217,321	2,289,941	**10.54**	2981.69	1.12	7485.8937	**1833.36**
D2-Net	NN-Mutual	780	364,527	1,557,653	4.27	1996.99	1.50	4442.129	1054.82
	NN-Ratio	336	46,033	226,822	4.93	675.07	1.46		90.70
	NN-Distance	780	261,884	1,201,174	4.59	1539.97	1.53		314.36
DISK	LightGlue	**783**	**536,211**	**3,780,568**	7.05	**4828.31**	1.35	6568.7982	1122.41
	NN-Mutual	**783**	469,707	3,378,905	7.19	4315.33	1.29		1029.01
	NN-Ratio	772	263,240	2,155,316	8.19	2791.86	1.16		373.46
	NN-Distance	772	344,770	2,661,758	7.72	3447.87	1.24		547.49
R2D2	NN-Mutual	**783**	349,930	3,553,683	10.16	4538.55	1.44	**7995.1648**	810.12
	NN-Ratio	485	122,945	1,104,995	8.99	2278.34	1.26		202.57
	NN-Distance	**783**	347,882	3,539,670	10.17	4520.65	1.44		804.78
SOSNet	Adalam	481	38,770	248,663	6.41	516.97	1.09	1028.1328	148.30
	NN-Mutual	477	43,176	254,733	5.90	534.03	1.07		341.13
	NN-Ratio	474	29,302	201,975	6.89	426.11	1.02		78.60
	NN-Distance	647	41,542	280,855	6.76	434.09	1.04		79.60
SuperPoint	NN-Mutual	713	102,468	665,417	6.49	933.26	1.34	1532.0166	530.73
	NN-Ratio	648	56,062	454,708	8.11	701.71	1.30		115.75
	NN-Distance	780	93,849	679,938	7.25	871.72	1.36		203.65
	SuperGlue	782	124,437	830,524	6.67	1062.05	1.39		293.23
	SuperGlue-Fast	781	123,910	823,766	6.65	1054.76	1.39		305.16
	LightGlue	781	122,568	825,248	6.73	1056.66	1.39		286.78
(**b**) Reconstruction results on the Hilti dataset.									
AKAZE	NN-BruteForce	321	13,797	90,506	6.56	281.95	1.16	458.6778	144.55
SIFT	NN-BruteForce	452	24,355	130,712	5.37	289.19	0.80	580.8544	162.72
ORB	NN-BruteForce	953	87,726	749,104	8.54	786.05	1.10	2140.8402	551.43
D2-Net	NN-Mutual	957	216,417	1,308,981	6.05	1367.80	1.38	2312.9816	579.80
	NN-Ratio	43	3871	48,141	12.44	1119.56	1.13		91.12
	NN-Distance	651	79,271	552,331	6.97	848.43	1.41		218.24
DISK	LightGlue	920	288,775	3,005,325	10.41	3266.66	0.99	4128.4682	775.60
	NN-Mutual	967	**309,370**	2,799,201	9.05	2894.73	0.88		866.34
	NN-Ratio	704	136,753	1,421,581	10.40	2019.29	0.76		454.50
	NN-Distance	673	115,683	1,199,493	10.37	1782.31	0.64		404.79
R2D2	NN-Mutual	956	273,521	**3,364,291**	12.30	**3519.13**	0.91	**6618.2336**	**878.93**
	NN-Ratio	**974**	243,966	3,217,281	13.19	3303.16	0.90		767.43
	NN-Distance	233	34,962	510,494	14.60	2190.96	0.67		349.81
SOSNet	Adalam	576	21,161	118,795	5.61	206.24	0.92	290.4109	97.47
	NN-Mutual	121	2652	18,944	7.14	156.56	0.76		25.98
	NN-Ratio	66	2555	10,310	4.04	156.21	0.66		65.81
	NN-Distance	55	1251	6153	4.92	111.87	0.62		25.47
SuperPoint	NN-Mutual	757	37,866	319,856	8.45	422.53	1.06	509.4252	98.37
	NN-Ratio	763	38,158	310,980	8.15	407.58	1.05		95.37
	NN-Distance	689	37,469	260,311	6.95	377.81	0.99		162.16
	SuperGlue	705	46,798	261,746	5.59	371.27	1.17		100.28
	SuperGlue-Fast	260	10,022	90,903	9.07	349.63	0.97		81.25
	LightGlue	43	698	12,696	**18.19**	295.26	**0.57**		46.49

Table 4 shows the cloud-to-cloud distances between the reconstructions and SIFT. The absence of values in the table is due to the failure of the method to produce a reconstruction for the corresponding combination. For ConSLAM's dataset, methods such as D2-Net with NN-Mutual matcher and DISK with NN-Mutual matcher demonstrated relatively low mean errors and standard deviations, indicating more accurate and consistent reconstructions. R2D2 with the NN-Ratio matcher also showed promising results with notably low mean and standard deviation values, highlighting its effectiveness in this particular scenario. However, these metrics did not necessarily reflect the quality of the reconstructions, as the reconstruction error found previously was not reflected in the cloud-to-cloud distances.

In contrast, for Hilti's dataset, the performance varied more significantly. D2-Net and DISK, both with NN-Distance or NN-Mutual matchers, exhibited lower mean errors compared to other methods, suggesting better adaptability to the complexity of the construction environment. However, some methods, like R2D2 with NN-Mutual matcher and SuperPoint with SuperGlue matcher, displayed higher standard deviations, indicating more variability in their reconstruction accuracy.

Table 4. Cloud-to-cloud distances between the reconstructions and SIFT for indoor scenes. The bold values highlight the configurations with the most accurate performance, characterized by the lowest mean cloud-to-cloud distance and standard deviation (STD). Missing values in the table are due to the failure of the method to produce a reconstruction for the corresponding sequence.

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

5.1.4. Outdoor Scenes

Reconstruction results for both datasets can be found in Table 5. Similar to previous observations, deep learning-based methods continued to outperform traditional techniques in several aspects. On the private dataset, models like DISK, R2D2, and D2-Net surpassed SIFT in terms of the number of points and observations. Interestingly, on this dataset, all methods registered an equal number of images, except SOSNet with NN-Distance. DISK, R2D2, and D2-Net also achieved greater mean track lengths and more observations

per image, demonstrating their ability to maintain detailed feature tracks across multiple images. On Hilti's dataset, SIFT was found to have comparable mean reprojection errors with fewer registered images than deep learning-based methods like DISK or R2D2, which consistently ranked highest in the number of registered images, points, and observations, indicating an effective balance between point quantity and reconstruction accuracy. Other traditional techniques like AKAZE and ORB generally fell behind in most metrics.

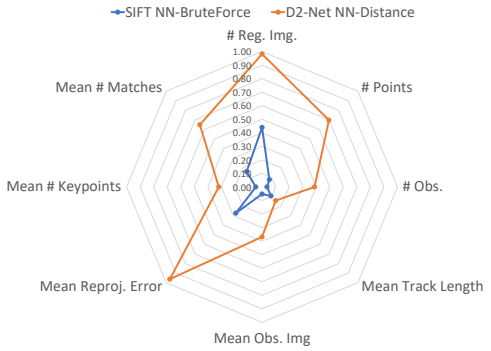
Table 5. Reconstruction results for the outdoor scenes from Hilti (a) and Private (b) datasets. The bold values highlight the configurations with the most significant performance in specific metrics, such as the highest number of registered images, points, observations, mean track length, mean observations per image, keypoints, or matches and the lowest mean reprojection error.

Extractor	Matcher	No. Reg. Img.	No. Points	No. Obs.	Mean Track Length	Mean Obs. Img	Mean Reproj. Error	Mean No. Key-points	Mean No. Matches
(**a**) Reconstruction results of Hilti's dataset.									
AKAZE	NN-BruteForce	321	13,797	90,506	6.56	281.95	1.16	458.68	144.55
SIFT	NN-BruteForce	452	24,355	130,712	5.37	289.19	0.80	580.85	162.72
ORB	NN-BruteForce	953	87,726	749,104	8.54	786.05	1.10	2140.84	551.43
D2-Net	NN-Mutual	651	79,271	552,331	6.97	848.43	1.41	2312.98	218.24
	NN-Ratio	43	3871	48,141	12.44	1119.56	1.13		91.12
	NN-Distance	957	216,417	1,308,981	6.05	1367.80	1.38		579.80
DISK	LightGlue	673	115,683	1,199,493	10.37	1782.31	0.64	4128.47	404.79
	NN-Mutual	704	136,753	1,421,581	10.40	2019.29	0.76		454.50
	NN-Ratio	967	**309,370**	2,799,201	9.05	2894.73	0.88		866.34
	NN-Distance	920	288,775	3,005,325	10.41	3266.66	0.99		775.60
R2D2	NN-Mutual	233	34,962	510,494	14.60	2190.96	0.67	**6618.23**	349.81
	NN-Ratio	956	273,521	**3,364,291**	12.30	**3519.13**	0.91		**878.93**
	NN-Distance	**974**	243,966	3,217,281	13.19	3303.16	0.90		767.43
SOSNet	Adalam	121	2652	18,944	7.14	156.56	0.76	290.41	25.98
	NN-Mutual	576	21,161	118,795	5.61	206.24	0.92		97.47
	NN-Ratio	55	1251	6153	4.92	111.87	0.62		25.47
	NN-Distance	66	2555	10,310	4.04	156.21	0.66		65.81
SuperPoint	NN-Mutual	43	698	12,696	**18.19**	295.26	**0.57**	509.43	46.49
	NN-Ratio	763	38,158	310,980	8.15	407.58	1.05		95.37
	NN-Distance	260	10,022	90,903	9.07	349.63	0.97		81.25
	SuperGlue	689	37,469	260,311	6.95	377.81	0.99		162.16
	SuperGlue-Fast	705	46,798	261,746	5.59	371.27	1.17		100.28
	LightGlue	757	37,866	319,856	8.45	422.53	1.06		98.37
(**b**) Reconstruction results on the private dataset.									
AKAZE	NN-BruteForce	477	49,652	597,559	12.03	1252.74	0.85	1569.00	557.31
SIFT	NN-BruteForce	477	158,408	1,121,996	7.08	2352.19	**0.62**	3515.05	1142.23
ORB	NN-BruteForce	477	174,582	2,677,935	**15.34**	5614.12	1.00	7614.78	**2204.80**
D2-Net	NN-Mutual	477	380,502	2,129,293	5.60	4463.93	1.55	5984.59	1623.07
	NN-Ratio	477	108,983	654,832	6.01	1372.81	1.32		170.39
	NN-Distance	477	288,048	1,604,316	5.57	3363.35	1.51		546.26
DISK	LightGlue	477	331,537	**3,582,694**	10.81	**7510.89**	1.04	7967.76	1880.21
	NN-Mutual	477	**399,953**	3,450,039	8.63	7232.79	0.90		1757.26
	NN-Ratio	477	376,608	2,973,756	7.90	6234.29	0.75		825.97
	NN-Distance	477	386,539	3,015,828	7.80	6322.49	0.77		867.81
R2D2	NN-Mutual	477	237,509	2,974,663	12.52	6236.19	1.14	**8000**	1015.63
	NN-Ratio	477	212,481	1,590,765	7.49	3334.94	0.87		325.12
	NN-Distance	477	237,136	2,974,582	12.54	6236.02	1.14		1015.04
SOSNet	Adalam	477	55,904	481,336	8.61	1009.09	0.86	1325.63	213.92
	NN-Mutual	477	65,807	507,848	7.72	1064.67	0.86		506.20
	NN-Ratio	477	50,713	417,985	8.24	876.28	0.74		158.90
	NN-Distance	357	38,796	310,958	8.02	871.03	0.68		147.91
SuperPoint	NN-Mutual	477	41,489	433,872	10.46	909.58	1.18	1048.51	447.31
	NN-Ratio	477	33,026	356,243	10.79	746.84	1.07		160.99
	NN-Distance	477	36,591	400,741	10.95	840.13	1.14		229.74
	SuperGlue	477	38,658	448,101	11.59	939.42	1.26		343.53
	SuperGlue-Fast	477	39,565	450,761	11.39	944.99	1.26		355.03
	LightGlue	477	38,147	447,215	11.72	937.56	1.26		338.02

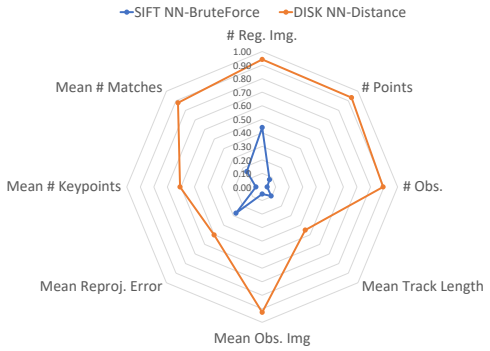
Figures 8 and 9 display radar plots comparing SIFT with three of the most representative methods in the private and Hilti datasets, respectively. The values were scaled and translated into the range [0, 1] for improved visualization using min-max scaling according to the following formula:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{10}$$

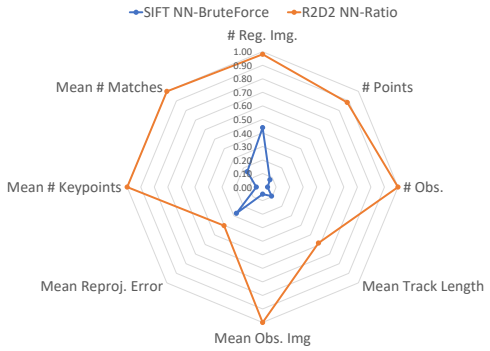
where x is the original value, and x' is the scaled value. The minimum and maximum values were calculated separately for each column. The original values are provided in Table 5.



(**a**) Reconstruction comparison of D2-Net+NN-Distance and SIFT



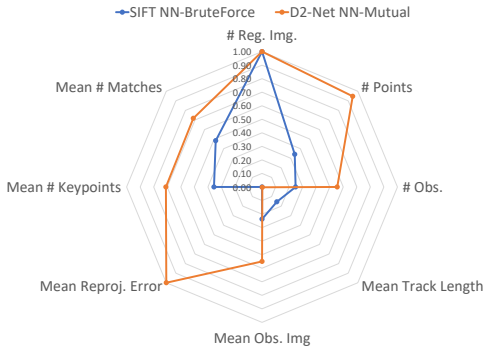
(**b**) Reconstruction comparison of DISK+NN-Distance and SIFT



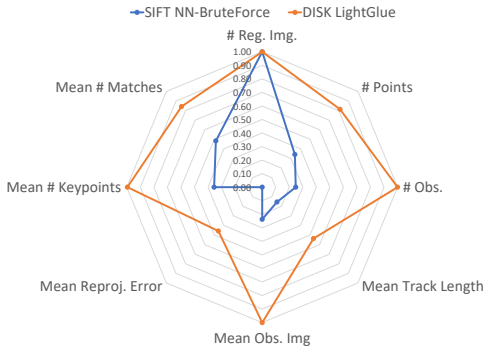
(**c**) Reconstruction comparison of R2D2+NN-Ratio and SIFT

Figure 8. Hilti—Construction Site Outdoor 1: reconstruction comparison between the top three deep learning methods and SIFT.

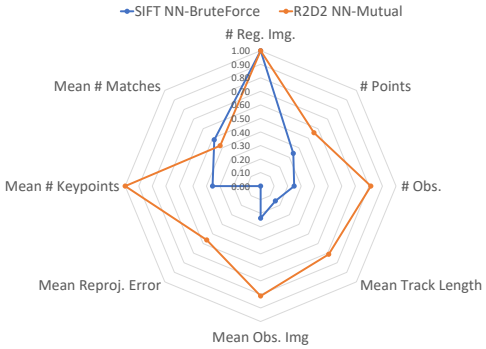
Table 6 shows the cloud-to-cloud distances between the reconstructions and SIFT. The absence of values in the table is due to the failure of the method to produce a reconstruction for the corresponding combination. In both datasets, the deep learning methodologies surpassed traditional approaches regarding the mean and standard deviation of error distances. On the private dataset, methods such as DISK with NN-Ratio and NN-Distance and R2D2 with NN-Ratio exhibited notably low mean error values, indicating high accuracy. Specifically, DISK with NN-Ratio achieved the lowest mean error, which was significantly lower than that of traditional methods. Likewise, on the Hilti dataset, DISK with NN-Mutual and NN-Ratio again demonstrated superior performance, with the mean distances falling within a low range and a relatively low standard deviation, suggesting both high accuracy and consistency.



(**a**) Reconstruction comparison of D2-Net+NN-Mutual and SIFT



(**b**) Reconstruction comparison of DISK+LightGlue and SIFT



(**c**) Reconstruction comparison of R2D2+NN-Mutual and SIFT

Figure 9. Private—Construction Site Outdoor: reconstruction comparison between the top three deep learning methods and SIFT.

Table 6. Cloud-to-cloud distances between the reconstructions and SIFT for outdoor scenes. The bold values highlight the configurations with the most accurate performance, characterized by the lowest mean cloud-to-cloud distance and standard deviation (STD). Missing values in the table are due to the failure of the method to produce a reconstruction for the corresponding sequence.

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

5.2. Performance Evaluation

5.2.1. Indoor Scenes

Performance results for both datasets can be found in Table 7. On the ConSLAM dataset, deep learning-based methods like SuperPoint and DISK outperformed traditional techniques like SIFT in feature extraction times. This speed advantage aligned with their lower CPU and RAM usage, a benefit of GPU acceleration. In contrast, all traditional methods exhibited higher CPU and RAM usage, reflecting slower processing speeds. Interestingly, the feature matching phase did not show as much improvement from learning-based techniques, with nearest neighbor methods exhibiting lower mean runtimes than those paired with SuperGlue or LightGlue.

A similar pattern emerged in Hilti's dataset. Deep learning methods, particularly DISK and SuperPoint, both with LightGlue, continued to demonstrate lower mean runtimes and resource usage for feature extraction and matching. They reduced the overall elapsed time and exhibited lower GPU memory and disk usage, which is beneficial for large-scale reconstructions. However, performance within the deep-learning methods varied, influenced by the specific combinations of feature extractors and matchers.

Table 7. Performance results for the indoor scenes from ConSLAM (a) and Hilti (b) datasets. The bold values highlight the most significant performance in specific metrics, such as the shortest elapsed time, fastest mean runtimes for feature extraction, matching, and global search, and the most efficient CPU, RAM, GPU, and disk usage.

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)
(**a**) Performance results on the ConSLAM dataset.										
AKAZE	NN-BruteForce	1.71	2156.184	799.003	3.0804	64.33	14.30	0.19	2.58	1.08
SIFT	NN-BruteForce	1.20	2229.797	1236.313	3.3245	70.77	25.50	3.49	2.73	2.20
ORB	NN-BruteForce	3.39	106.054	1485.637	3.1052	66.67	5.15	**0.12**	2.89	1.33
D2-Net	NN-Mutual	1.80	119.202	3.884	2.9736	43.38	4.68	2.94	2.80	4.29
	NN-Ratio	**0.31**	120.850	3.926	3.0212	21.02	2.90	8.30	2.80	3.93
	NN-Distance	1.22	120.850	3.893	2.9736	39.71	4.07	3.41	2.80	4.09
DISK	LightGlue	2.58	79.041	41.921	3.0231	38.19	7.03	7.55	5.52	2.64
	NN-Mutual	2.15	79.163	4.965	**2.8763**	45.31	6.30	3.02	3.33	2.56
	NN-Ratio	0.81	79.163	5.006	2.8839	34.27	4.76	5.13	3.32	2.32
	NN-Distance	1.20	79.163	4.964	2.9106	39.44	5.38	4.03	3.32	2.39
R2D2	NN-Mutual	1.85	173.218	6.927	2.9507	44.65	6.18	4.16	3.48	2.92
	NN-Ratio	0.57	173.218	6.996	2.943	24.27	3.80	9.46	3.23	2.65
	NN-Distance	1.92	172.852	6.953	2.9678	45.21	6.13	4.07	3.23	2.92
SOSNet	Adalam	0.34	272.461	8.774	9.3307	25.98	3.74	4.88	3.05	0.48
	NN-Mutual	0.51	278.076	**1.500**	9.2239	39.67	3.74	2.64	3.04	0.51
	NN-Ratio	0.35	273.193	1.679	9.1095	24.88	3.70	3.08	3.04	**0.42**
	NN-Distance	0.38	269.043	1.597	9.407	27.00	3.76	3.04	3.04	0.43
SuperPoint	NN-Mutual	0.74	**11.093**	1.548	2.9755	39.08	2.82	2.50	**2.55**	1.01
	NN-Ratio	0.37	11.124	1.700	2.9755	23.60	2.78	3.20	**2.55**	0.88
	NN-Distance	0.60	11.147	1.653	3.006	33.42	**2.77**	2.64	**2.55**	0.93
	SuperGlue	1.08	11.108	84.371	2.9316	**15.56**	3.38	14.08	2.61	0.97
	SuperGlue-Fast	1.09	11.108	73.917	2.9469	19.35	3.48	10.38	2.61	0.98
	LightGlue	0.59	11.284	15.270	3.046	26.65	3.24	7.63	2.67	0.97
(**b**) Performance results on the Hilti dataset.										
AKAZE	NN-BruteForce	0.32	149.298	3.019	1.886	70.50	2.91	0.61	2.03	0.23
SIFT	NN-BruteForce	0.27	200.208	9.892	1.846	70.31	5.84	0.71	2.03	0.49
ORB	NN-BruteForce	1.07	28.630	151.729	1.849	64.75	3.16	**0.23**	2.26	0.61
D2-Net	NN-Mutual	1.19	59.296	1.948	1.762	36.67	3.77	1.04	2.52	2.90
	NN-Ratio	0.16	60.089	2.025	1.777	12.19	2.88	7.46	2.47	2.62
	NN-Distance	0.66	60.272	1.968	1.791	27.54	2.86	1.88	2.45	2.73
DISK	LightGlue	1.35	39.825	26.771	1.775	26.67	5.62	8.38	3.42	2.16
	NN-Mutual	1.69	39.764	2.782	1.786	37.75	5.38	1.03	2.42	2.18
	NN-Ratio	0.47	39.764	2.819	1.752	29.46	3.51	3.71	2.44	1.93
	NN-Distance	0.69	39.764	2.796	1.754	32.38	3.67	2.53	2.68	1.96
R2D2	NN-Mutual	2.06	78.003	5.232	1.715	36.75	6.00	1.71	2.67	3.16
	NN-Ratio	0.54	77.820	5.276	**1.710**	27.63	3.25	6.46	2.37	2.84
	NN-Distance	1.65	77.942	5.218	1.711	33.90	5.80	2.12	4.51	3.11
SOSNet	Adalam	0.17	117.127	7.396	5.013	33.71	3.71	4.96	3.68	0.26
	NN-Mutual	0.29	117.981	1.431	5.009	40.00	3.70	1.08	3.69	0.25
	NN-Ratio	0.12	120.361	1.640	5.039	17.48	3.71	2.80	3.67	**0.21**
	NN-Distance	0.13	118.225	1.552	5.028	15.38	3.67	2.32	3.69	0.21
SuperPoint	NN-Mutual	0.47	**10.445**	**1.408**	1.745	34.85	2.78	0.91	**1.90**	0.49
	NN-Ratio	**0.08**	10.475	1.636	1.744	**4.94**	**2.77**	5.19	1.93	0.42
	NN-Distance	0.19	10.513	1.550	1.755	16.50	2.79	2.34	1.96	0.45
	SuperGlue	1.05	10.483	69.039	1.747	11.17	2.78	6.27	2.73	0.47
	SuperGlue-Fast	0.93	10.506	62.231	1.733	11.44	2.78	5.23	2.40	0.47
	LightGlue	0.37	10.895	14.467	1.770	15.71	2.79	13.89	2.24	0.47

5.2.2. Outdoor Scenes

Performance results for both datasets can be found in Table 8. On the private dataset, traditional methods like AKAZE outperformed SIFT with lower elapsed time and reduced CPU and RAM usage, indicating their computational efficiency. In contrast, ORB's rapid feature extraction was hindered by prolonged feature matching times, resulting in overall slower performance. This highlights the variability within traditional methods and potential bottlenecks in feature extraction or matching.

Deep learning methods, on the other hand, consistently outperformed SIFT. They achieved faster extraction and matching times, resulting in reduced overall elapsed times, such as in the case of DISK with LightGlue, D2-Net with NN-Ratio, and SuperPoint with various matchers. These findings underscore the efficacy of learning-based models in handling complex tasks and their potential to enhance the speed and efficiency of SfM processes. Hilti's dataset corroborated these trends; for instance, SuperPoint paired with matchers like SuperGlue and LightGlue exhibited exceptional performance. They leveraged GPU resources effectively, resulting in some cases in the fastest feature extraction and matching times, as well as overall elapsed times (compared to SIFT). Additionally, their efficient GPU utilization highlights their potential for large-scale and real-time applications.

Table 8. Performance results for the outdoor scenes from Private (a) and Hilti (b) datasets. The bold values highlight the most significant performance in specific metrics, such as the shortest elapsed time, fastest mean runtimes for feature extraction, matching, and global search, and the most efficient CPU, RAM, GPU, and disk usage.

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)
(**a**) Performance results on the private dataset.										
AKAZE	NN-BruteForce	0.47	930.335	64.256	3.326	56.23	6.55	0.35	2.59	**0.30**
SIFT	NN-BruteForce	0.68	857.155	720.354	3.264	58.46	16.27	0.24	2.59	1.24
ORB	NN-BruteForce	1.73	93.997	1491.121	3.204	55.71	5.02	**0.11**	2.84	0.94
D2-Net	NN-Mutual	1.20	122.009	5.906	2.979	32.33	5.30	3.11	2.82	3.55
	NN-Ratio	0.29	120.789	5.933	3.065	16.65	3.20	7.26	2.82	3.25
	NN-Distance	0.53	122.376	5.936	3.021	17.98	4.48	4.75	2.82	3.37
DISK	LightGlue	0.98	78.247	54.349	3.099	17.04	6.56	13.20	4.10	2.02
	NN-Mutual	0.81	78.613	6.937	3.065	29.71	6.20	4.23	3.33	1.95
	NN-Ratio	0.48	78.308	7.045	2.966	19.40	5.67	5.94	3.32	1.78
	NN-Distance	0.69	78.308	7.002	2.853	29.15	5.77	4.74	3.32	1.79
R2D2	NN-Mutual	0.99	172.241	7.071	2.831	26.35	5.41	4.49	3.58	1.81
	NN-Ratio	0.38	171.997	7.167	2.951	14.98	4.19	8.65	3.33	1.64
	NN-Distance	0.80	172.119	7.086	**2.756**	21.50	5.41	5.08	3.33	1.81
SOSNet	Adalam	0.21	247.192	9.633	6.481	18.69	3.51	5.66	3.04	0.38
	NN-Mutual	0.32	253.174	1.488	6.870	35.73	3.49	2.88	3.03	0.40
	NN-Ratio	0.18	250.000	1.652	6.203	18.04	3.53	3.61	3.03	0.34
	NN-Distance	0.23	247.559	1.593	6.535	19.00	3.53	3.24	3.03	0.34
SuperPoint	NN-Mutual	0.33	11.414	**1.480**	3.037	32.21	2.80	2.56	**2.57**	0.46
	NN-Ratio	**0.15**	11.360	1.673	2.901	11.98	**2.77**	3.57	**2.57**	0.40
	NN-Distance	0.18	11.421	1.597	3.086	14.35	2.78	3.32	**2.57**	0.42
	SuperGlue	0.48	11.360	44.896	2.922	19.08	2.78	10.90	2.63	0.46
	SuperGlue-Fast	0.32	11.368	37.836	2.968	**11.46**	2.80	12.00	2.63	0.46
	LightGlue	0.23	**11.299**	15.328	2.890	15.50	2.80	9.07	2.63	0.46
(**b**) Performance results on the Hilti dataset.										
AKAZE	NN-BruteForce	19.07	149.298	3.019	1.886	70.50	2.91	0.61	2.03	0.23
SIFT	NN-BruteForce	15.94	200.208	9.892	1.846	70.31	5.84	0.71	2.03	0.49
ORB	NN-BruteForce	25.18	28.630	151.729	1.849	64.75	3.16	**0.23**	2.26	0.60
D2-Net	NN-Mutual	20.63	60.272	1.968	1.791	27.54	2.86	1.88	2.45	2.73
	NN-Ratio	5.59	60.089	2.025	1.777	12.19	2.88	7.46	2.47	2.62
	NN-Distance	20.29	59.296	1.948	1.762	36.67	3.77	1.04	2.52	2.89
DISK	LightGlue	41.34	39.764	2.819	1.752	29.46	3.51	3.71	2.44	1.92
	NN-Mutual	40.28	39.764	2.796	1.754	32.38	3.67	2.53	2.68	1.95
	NN-Ratio	26.61	39.764	2.782	1.786	37.75	5.38	1.03	2.42	2.15
	NN-Distance	32.09	39.825	26.771	1.775	26.67	5.62	8.38	3.42	2.15
R2D2	NN-Mutual	36.19	77.820	5.276	**1.710**	27.63	3.25	6.46	2.37	2.84
	NN-Ratio	19.40	78.003	5.232	1.715	36.75	6.00	1.71	2.67	3.14
	NN-Distance	31.88	77.942	5.218	1.711	33.90	5.80	2.12	4.51	3.09
SOSNet	Adalam	7.44	118.225	1.552	5.028	15.38	3.67	2.32	3.69	0.21
	NN-Mutual	8.16	117.981	1.431	5.009	40.00	3.70	1.08	3.69	0.25
	NN-Ratio	6.46	120.361	1.640	5.039	17.48	3.71	2.80	3.67	**0.21**
	NN-Distance	5.97	117.127	7.396	5.013	33.71	3.71	4.96	3.68	0.26
SuperPoint	NN-Mutual	7.78	10.475	1.636	1.744	**4.94**	**2.77**	5.19	1.93	0.42
	NN-Ratio	**5.13**	10.895	14.467	1.770	15.71	2.79	13.89	2.24	0.47
	NN-Distance	7.61	10.513	1.550	1.755	16.50	2.79	2.34	1.96	0.45
	SuperGlue	8.91	**10.445**	**1.408**	1.745	34.85	2.78	0.91	**1.90**	0.49
	SuperGlue-Fast	9.57	10.506	62.231	1.733	11.44	2.78	5.23	2.40	0.47
	LightGlue	7.59	10.483	69.039	1.747	11.17	2.78	6.27	2.73	0.47

6. Conclusions

The purpose of this work 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 were selected to represent a wide range of challenging scenarios, including varying lighting conditions, feature scarcity, and occlusions. Using cloud-to-cloud distances, and reconstruction metrics regarding a baseline (SIFT), as well as a visual inspection of the point clouds, the reconstruction evaluation revealed insights into the quality, accuracy, and completeness of the reconstructions. On the other hand, the performance evaluation analyzed 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 the following:

- Traditional methods 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 good lighting conditions, as seen on the private dataset. However, compared to point clouds generated by deep learning methods, they were less detailed and had a larger number of missing areas.
- Deep learning-based methods consistently outperformed traditional methods in terms of reconstruction quality, particularly in challenging scenarios with complex lighting conditions, where over- and under-exposure were common and featureless areas like walls and floors were present. This outcome can be attributed to the sophisticated feature extraction inherent in these methods, which facilitated a more effective identification and matching of features across image pairs and the datasets they were trained on. For instance, R2D2 and DISK, trained on outdoor datasets such as MegaDepth [47] and Aachen [39,40], were identified as the most effective methods in terms of reconstruction quality and performance within the outdoor scenes, indicating that the training data played a significant role in the performance of the methods.
- Deep learning methods were more efficient in terms of processing time and resource consumption, leveraging modern hardware capabilities to enhance performance, which demonstrated exceptional efficiency in feature extraction and matching thanks to the GPU acceleration.
- 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, 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 only one map was capable of reconstructing the scene.
- The evaluation of cloud-to-cloud distances provided valuable insights into the accuracy of the 3D reconstructions. Most deep learning-based methods demonstrated superior performance in terms of mean distance and standard deviation, indicating high precision and consistency. Compared to SIFT, these methods tended to produce point clouds with a better appearance and fewer missing areas. Cloud-to-cloud distances served as a more direct measure of reconstruction accuracy, provided the reconstructions were successfully generated and sufficiently dense.
- 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, like indoor scenes. However, the tailored matching techniques were

more consistent and robust across both datasets, suggesting that they may be more suitable for an overall reconstruction process without the need for manual tuning or adjustments.

Future Work

While this study provides a comprehensive evaluation of out-of-the-box performance for various feature extraction and matching methods, we recognize the importance of conducting ablation studies to examine the impact of different parameters and configurations. Future work should focus on a detailed parameter analysis and optimization to enhance the performance of specific methods in tailored applications. Additionally, as new Transformer-based methods for feature extraction [51,52] and matching [53,54] are developed, future research could also include new methods and evaluate their effectiveness in 3D reconstruction by leveraging attention maps to enhance performance.

Furthermore, given the complexity of the scenes, the ConSLAM dataset also provides NIR images (not used in this work). These images can be used to mitigate overexposed or underexposed areas. Further research is needed to evaluate the impact and suitability of fusing NIR images with RGB images on reconstruction quality; techniques such as those explored in [55,56] could be utilized for this fusion.

Finally, future research could explore a hybrid approach, applying techniques on a case-by-case or area-by-area basis to leverage the strengths of different methods. For instance, traditional methods could be utilized in areas with sufficient visual features and favorable lighting conditions, while deep learning-based methods could be employed in challenging scenarios with complex lighting. This approach could help researchers and practitioners optimize the reconstruction process based on the specific characteristics of each scene, ensuring high-quality and accurate reconstructions across diverse scenarios.

Author Contributions: Conceptualization, C.R.C.Z., I.C., and J.P.Q.; methodology, C.R.C.Z. and I.C.; software, C.R.C.Z.; validation, C.R.C.Z., I.C., and J.P.Q.; formal analysis, C.R.C.Z.; investigation, C.R.C.Z. and I.C.; writing—original draft preparation, C.R.C.Z., I.C., and J.P.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The code used to conduct the analysis will be made available upon request.

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

References

1. Agarwal, S.; Furukawa, Y.; Snavely, N.; Simon, I.; Curless, B.; Seitz, S.M.; Szeliski, R. Building rome in a day. *Commun. ACM* **2011**, *54*, 105–112. [CrossRef]
2. Frahm, J.M.; Fite-Georgel, P.; Gallup, D.; Johnson, T.; Raguram, R.; Wu, C.; Jen, Y.H.; Dunn, E.; Clipp, B.; Lazebnik, S.; et al. Building rome on a cloudless day. In Proceedings of the Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September 2010; Proceedings, Part IV 11; Springer: Berlin/Heidelberg, Germany, 2010; pp. 368–381.
3. Heinly, J.; Schonberger, J.L.; Dunn, E.; Frahm, J.M. Reconstructing the world* in six days*(as captured by the yahoo 100 million image dataset). In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3287–3295.
4. Schonberger, J.L.; Frahm, J.M. Structure-from-motion revisited. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4104–4113.
5. Wei, Y.m.; Kang, L.; Yang, B.; Wu, L.d. Applications of structure from motion: A survey. *J. Zhejiang Univ. Sci. C* **2013**, *14*, 486–494. [CrossRef]
6. Karsch, K.; Golparvar-Fard, M.; Forsyth, D. ConstructAide: Analyzing and visualizing construction sites through photographs and building models. *ACM Trans. Graph. (TOG)* **2014**, *33*, 1–11. [CrossRef]
7. Khaloo, A.; Lattanzi, D.; Cunningham, K.; Dell'Andrea, R.; Riley, M. Unmanned aerial vehicle inspection of the Placer River Trail Bridge through image-based 3D modelling. *Struct. Infrastruct. Eng.* **2018**, *14*, 124–136. [CrossRef]
8. Xiong, X.; Adan, A.; Akinci, B.; Huber, D. Automatic creation of semantically rich 3D building models from laser scanner data. *Autom. Constr.* **2013**, *31*, 325–337. [CrossRef]

9. Olsen, M.J.; Kuester, F.; Chang, B.J.; Hutchinson, T.C. Terrestrial laser scanning-based structural damage assessment. *J. Comput. Civ. Eng.* **2010**, *24*, 264–272. [CrossRef]
10. Tang, P.; Huber, D.; Akinci, B.; Lipman, R.; Lytle, A. Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. *Autom. Constr.* **2010**, *19*, 829–843. [CrossRef]
11. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
12. Bay, H.; Tuytelaars, T.; Van Gool, L. Surf: Speeded up robust features. In Proceedings of the Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; Proceedings, Part I 9; Springer: Berlin/Heidelberg, Germany, 2006; pp. 404–417.
13. Alcantarilla, P.F.; Solutions, T. Fast explicit diffusion for accelerated features in nonlinear scale spaces. *IEEE Trans. Patt. Anal. Mach. Intell.* **2011**, *34*, 1281–1298.
14. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2564–2571.
15. Pusztai, Z.; Hajder, L. Quantitative Comparison of Feature Matchers Implemented in OpenCV3. In Proceedings of the 21st Computer Vision Winter Workshop, Rimske Toplice, Slovenia, 3–5 February 2016; Slovenian Pattern Recognition Society: Ljubljana, Slovenia, 2016; pp. 1–9.
16. Özyeşil, O.; Voroninski, V.; Basri, R.; Singer, A. A survey of structure from motion*. *Acta Numer.* **2017**, *26*, 305–364. [CrossRef]
17. Jin, Y.; Mishkin, D.; Mishchuk, A.; Matas, J.; Fua, P.; Yi, K.M.; Trulls, E. Image matching across wide baselines: From paper to practice. *Int. J. Comput. Vis.* **2021**, *129*, 517–547. [CrossRef]
18. Schonberger, J.L.; Hardmeier, H.; Sattler, T.; Pollefeys, M. Comparative evaluation of hand-crafted and learned local features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1482–1491.
19. Dusmanu, M.; Rocco, I.; Pajdla, T.; Pollefeys, M.; Sivic, J.; Torii, A.; Sattler, T. 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, Long Beach, CA, USA, 15–20 June 2019; pp. 8084–8093.
20. Revaud, J.; De Souza, C.; Humenberger, M.; Weinzaepfel, P. R2d2: Reliable and repeatable detector and descriptor. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 12414–12424.
21. Sarlin, P.E.; DeTone, D.; Malisiewicz, T.; Rabinovich, A. Superglue: Learning feature matching with graph neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 4938–4947.
22. Tyszkiewicz, M.; Fua, P.; Trulls, E. DISK: Learning local features with policy gradient. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 14254–14265.
23. DeTone, D.; Malisiewicz, T.; Rabinovich, A. Superpoint: Self-supervised interest point detection and description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 224–236.
24. Martell, A.; Lauterbach, H.A.; Nuchtcer, A. Benchmarking structure from motion algorithms of urban environments with applications to reconnaissance in search and rescue scenarios. In Proceedings of the 2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Philadelphia, PA, USA, 6–8 August 2018; pp. 1–7.
25. Knapitsch, A.; Park, J.; Zhou, Q.Y.; Koltun, V. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Trans. Graph. (ToG)* **2017**, *36*, 1–13. [CrossRef]
26. Balntas, V.; Lenc, K.; Vedaldi, A.; Mikolajczyk, K. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5173–5182.
27. Fan, B.; Kong, Q.; Wang, X.; Wang, Z.; Xiang, S.; Pan, C.; Fua, P. A performance evaluation of local features for image-based 3D reconstruction. *IEEE Trans. Image Process.* **2019**, *28*, 4774–4789. [CrossRef] [PubMed]
28. Tareen, S.A.K.; Saleem, Z. A comparative analysis of sift, surf, kaze, akaze, orb, and brisk. In Proceedings of the 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 3–4 March 2018; pp. 1–10.
29. Bartol, K.; Bojanić, D.; Pribanić, T.; Petković, T.; Donoso, Y.; Mas, J. On the comparison of classic and deep keypoint detector and descriptor methods. In Proceedings of the 2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA), Dubrovnik, Croatia, 23–25 September 2019; pp. 64–69.
30. Remondino, F.; Menna, F.; Morelli, L. Evaluating hand-crafted and learning-based features for photogrammetric applications. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2021**, *43*, 549–556. [CrossRef]
31. Ruano, S.; Smolic, A. A Benchmark for 3D Reconstruction from Aerial Imagery in an Urban Environment. In *Proceedings of the VISIGRAPP (5: VISAPP)*; SciTePress: Setúbal, Portugal, 2021; pp. 732–741.
32. Corradetti, A.; Seers, T.; Mercuri, M.; Calligaris, C.; Busetti, A.; Zini, L. 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 Sens.* **2022**, *14*, 5187. [CrossRef]
33. Rosten, E.; Drummond, T. Machine learning for high-speed corner detection. In Proceedings of the Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; Proceedings, Part I 9; Springer: Berlin/Heidelberg, Germany, 2006; pp. 430–443.

34. Calonder, M.; Lepetit, V.; Strecha, C.; Fua, P. Brief: Binary robust independent elementary features. In Proceedings of the Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September 2010; Proceedings, Part IV 11; Springer: Berlin/Heidelberg, Germany, 2010; pp. 778–792.
35. Mikolajczyk, K.; Schmid, C. Indexing based on scale invariant interest points. In Proceedings of the 8th IEEE International Conference on Computer Vision. ICCV 2001, Vancouver, BC, Canada, 7–14 July 2001; Volume 1, pp. 525–531.
36. Alcantarilla, P.F.; Bartoli, A.; Davison, A.J. KAZE features. In Proceedings of the Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; Proceedings, Part VI 12; Springer: Berlin/Heidelberg, Germany, 2012; pp. 214–227.
37. He, K.; Lu, Y.; Sclaroff, S. Local descriptors optimized for average precision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 596–605.
38. Novotny, D.; Albanie, S.; Larlus, D.; Vedaldi, A. Self-supervised learning of geometrically stable features through probabilistic introspection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3637–3645.
39. Sattler, T.; Maddern, W.; Toft, C.; Torii, A.; Hammarstrand, L.; Stenborg, E.; Safari, D.; Okutomi, M.; Pollefeys, M.; Sivic, J.; et al. Benchmarking 6dof outdoor visual localization in changing conditions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8601–8610.
40. Sattler, T.; Weyand, T.; Leibe, B.; Kobbelt, L. Image Retrieval for Image-Based Localization Revisited. In Proceedings of the BMVC, Surrey, UK, 3–7 September 2012; Volume 1, p. 4.
41. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part V 13; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
42. Tian, Y.; Yu, X.; Fan, B.; Wu, F.; Heijnen, H.; Balntas, V. Sosnet: Second order similarity regularization for local descriptor learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 11016–11025.
43. Cavalli, L.; Larsson, V.; Oswald, M.; Sattler, T.; Pollefeys, M. Adalam: Revisiting handcrafted outlier detection. *arXiv* **2020**, arXiv:2006.04250.
44. Radenović, F.; Iscen, A.; Tolias, G.; Avrithis, Y.; Chum, O. Revisiting Oxford and Paris: Large-Scale Image Retrieval Benchmarking. In Proceedings of the CVPR, Salt Lake City, UT, USA, 18–23 June 2018.
45. Lindenberger, P.; Sarlin, P.E.; Pollefeys, M. Lightglue: Local feature matching at light speed. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 1–6 October 2023; pp. 17627–17638.
46. Khan, S.; Naseer, M.; Hayat, M.; Zamir, S.W.; Khan, F.S.; Shah, M. Transformers in vision: A survey. *ACM Comput. Surv. (CSUR)* **2022**, *54*, 1–41. [CrossRef]
47. Li, Z.; Snavely, N. Megadepth: Learning single-view depth prediction from internet photos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2041–2050.
48. Sarlin, P.E.; Cadena, C.; Siegwart, R.; Dymczyk, M. From coarse to fine: Robust hierarchical localization at large scale. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 12716–12725.
49. Helmberger, M.; Morin, K.; Berner, B.; Kumar, N.; Cioffi, G.; Scaramuzza, D. The hilti slam challenge dataset. *IEEE Robot. Autom. Lett.* **2022**, *7*, 7518–7525. [CrossRef]
50. Trzeciak, M.; Pluta, K.; Fathy, Y.; Alcalde, L.; Chee, S.; Bromley, A.; Brilakis, I.; Alliez, P. Conslam: Periodically collected real-world construction dataset for SLAM and progress monitoring. In *Proceedings of the European Conference on Computer Vision*; Springer: Cham, Switzerland, 2022; pp. 317–331.
51. Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; et al. Dinov2: Learning robust visual features without supervision. *arXiv* **2023**, arXiv:2304.07193.
52. Chen, H.; Luo, Z.; Zhou, L.; Tian, Y.; Zhen, M.; Fang, T.; Mckinnon, D.; Tsin, Y.; Quan, L. Aspanformer: Detector-free image matching with adaptive span transformer. In *Proceedings of the European Conference on Computer Vision*; Springer: Cham, Switzerland, 2022; pp. 20–36.
53. Jiang, H.; Karpur, A.; Cao, B.; Huang, Q.; Araujo, A. OmniGlue: Generalizable Feature Matching with Foundation Model Guidance. *arXiv* **2024**, arXiv:2405.12979.
54. Wang, Q.; Zhang, J.; Yang, K.; Peng, K.; Stiefelhagen, R. Matchformer: Interleaving attention in transformers for feature matching. In Proceedings of the Asian Conference on Computer Vision, Macao, China, 4–8 December 2022; pp. 2746–2762.
55. Ying, J.; Tong, C.; Sheng, Z.; Yao, B.; Cao, S.Y.; Yu, H.; Shen, H.L. Region-aware RGB and near-infrared image fusion. *Pattern Recognit.* **2023**, *142*, 109717. [CrossRef]
56. Zou, D.; Yang, B.; Li, Y.; Zhang, X.; Pang, L. Visible and NIR image fusion based on multiscale gradient guided edge-smoothing model and local gradient weight. *IEEE Sens. J.* **2023**, *23*, 2783–2793. [CrossRef]