remote sensing

MDPI

Article

Fusing LiDAR and Photogrammetry for Accurate 3D Data: A Hybrid Approach

Rytis Maskeliūnas [1,*], Sarmad Maqsood [1], Mantas Vaškevičius [2] and Julius Gelšvartas [2]

[1] Centre of Real Time Computer Systems, Faculty of Informatics, Kaunas University of Technology, LT-51386 Kaunas, Lithuania; sarmad.maqsood@ktu.lt
[2] Matomai UAB, LT-51423 Kaunas, Lithuania; mantas.vaskevicius@matom.ai (M.V.); julius@matom.ai (J.G.)
* Correspondence: rytis.maskeliunas@ktu.lt

Abstract: The fusion of LiDAR and photogrammetry point clouds is a necessary advancement in 3D-modeling, enabling more comprehensive and accurate representations of physical environments. The main contribution of this paper is the development of an innovative fusion system that combines classical algorithms, such as Structure from Motion (SfM), with advanced machine learning techniques, like Coherent Point Drift (CPD) and Feature-Metric Registration (FMR), to improve point cloud alignment and fusion. Experimental results, using a custom dataset of real-world scenes, demonstrate that the hybrid fusion method achieves an average error of less than 5% in the measurements of small reconstructed objects, with large objects showing less than 2% deviation from real sizes. The fusion process significantly improved structural continuity, reducing artifacts like edge misalignments. The k-nearest neighbors (kNN) analysis showed high reconstruction accuracy for the hybrid approach, demonstrating that the hybrid fusion system, particularly when combining machine learning-based refinement with traditional alignment methods, provides a notable advancement in both geometric accuracy and computational efficiency for real-time 3D-modeling applications.

Keywords: LiDAR; photogrammetry; point cloud fusion; machine learning; augmented reality

check for updates

Academic Editors: Luis Miguel González de Santos and Pablo Rodríguez-Gonzálvez

Received: 3 December 2024
Revised: 23 January 2025
Accepted: 24 January 2025
Published: 28 January 2025

Citation: Maskeliūnas, R.; Maqsood, S.; Vaškevičius, M.; Gelšvartas, J. Fusing LiDAR and Photogrammetry for Accurate 3D Data: A Hybrid Approach. *Remote Sens.* **2025**, *17*, 443. https://doi.org/10.3390/rs17030443

1. Introduction

With the advent of 3D-mapping technologies, the ability to capture detailed and precise representations of the physical world has transformed various fields, including urban planning [1], environmental monitoring [2], autonomous navigation [3], and augmented reality [4]. Two of the most well-known technologies in this field are photogrammetry [5] and Light Detection and Ranging (LiDAR) [6]. Each method excels in generating point clouds, which are three-dimensional depictions of spatial data, but they also possess inherent limitations [7]. This study addresses the fusion of LiDAR and photogrammetry point clouds to combine their strengths, overcoming their individual shortcomings to produce high-quality 3D models.

LiDAR gathers exact distance measurements by emitting laser pulses and recording the time it takes for them to return, generating highly accurate point clouds that excel in representing the geometry of objects [6]. Due to the importance of spatial accuracy, LiDAR has been widely used in urban planning [8], autonomous vehicles [9], and topographical surveys [10]. However, while LiDAR provides geometric precision, it lacks the ability to capture detailed textures or colors, which are important for applications such as visual inspection and Augmented Reality (AR) [11]. Furthermore, the high cost and complexity of LiDAR systems limit their use in consumer-level applications. On the other hand, by processing 2D photographs from various perspectives, photogrammetry provides high-quality surface texture information while reconstructing 3D models [12]. Photogrammetry is extensively used in fields like architecture, cultural heritage preservation, and media production, where visual detail is paramount [13]. However, its spatial accuracy is often less reliable, being influenced by factors such as lighting, camera quality, and lack of precise distance measurements [14]. Furthermore, photogrammetry struggles in low-light or low-contrast conditions and takes a long time to analyze [15].

In order to create more complete 3D models that combine the geometric accuracy of LiDAR with the rich visual information offered by photogrammetry, there is an increasing need to fuse LiDAR and photogrammetry data due to the advantages and disadvantages of both technologies [5,6]. Technical difficulties in this fusion process include aligning point clouds created from sources of data that range in scale and quality, removing redundant or noisy data, and ensuring that the resulting model is both geometrically precise and visually detailed [16]. Point clouds are often aligned and fused using traditional methods such as Simultaneous Localization and Mapping (SLAM) [17] and Iterative Closest Point (ICP) [18]. However, these methods frequently falter when handling datasets with several resolutions or when real-time processing is necessary. Furthermore, inaccurate point-cloud alignment is hampered by data inconsistencies, such as gaps or noise from external factors. Therefore, in order to effectively handle these issues, more advanced algorithms are required [19].

This research aims to develop a novel framework for fusing LiDAR and photogrammetry point clouds using a combination of traditional methods and cutting-edge techniques like Structure from Motion (SfM) [20], Neural Radiance Fields (NeRF) [21], and 3D Gaussian Splatting [22]. SfM enables the reconstruction of 3D models from 2D images, while NeRF and 3D Gaussian Splatting employ neural networks to generate highly detailed 3D objects from sparse image datasets. These approaches, combined with machine learning-based algorithms such as Coherent Point Drift (CPD) [23] and Feature-Metric Registration (FMR) [24], aim to enhance the efficiency, accuracy, and scalability of the point cloud fusion process. By addressing the key challenges in point cloud fusion, we believe that this research can contribute to the advancement of applications in geographic information systems (GIS) [25], autonomous systems [26], AR [27], and urban modeling [28].

Our approach presents a comprehensive approach to point cloud registration, focusing on the integration of LiDAR and photogrammetry data to enhance both geometric accuracy and visual quality. The novelty lies in an innovative fusion strategy that addresses the challenges posed by combining point clouds of differing resolutions, particularly when high-resolution photogrammetry is combined with low-resolution LiDAR scans. The use of voxel-based down-sampling, while effective in reducing redundancy, was carefully balanced to minimize the loss of fine details. Furthermore, advanced algorithms such as Coherent Point Drift (CPD) and DeepGMR were exploited to align datasets with varying densities and non-rigid transformations, overall advancing the field by improving the efficiency and precision of point cloud fusion.

The subsequent sections of this manuscript are organized as follows: Section 2 provides an exploration of related works. Section 3 details the proposed methodology and dataset used. Section 4 presents the experimental results. Section 5 offers a discussion. Finally, Section 6 concludes the manuscript.

2. Related Works

Point cloud generation and fusion have garnered significant attention due to their crucial role in creating accurate 3D models for applications such as urban planning and GIS. The evolution of methods such as LiDAR, photogrammetry, SfM, NeRF, and 3D Gaussian

Splatting has significantly improved the ability to generate high-fidelity 3D representations. Due to variations in resolution, accuracy, and computing demands, the problem of combining point clouds from several modalities such as LiDAR and photogrammetry remains a crucial field of study. Table 1 summarizes the more popular current approaches in point cloud generation and fusion and their limitations. LiDAR technology excels in geometric accuracy, making it indispensable for precise spatial mapping applications, although its real-time capabilities are limited due to high computational and hardware requirements [6,8,10]. In contrast, photogrammetry provides richer visual fidelity but struggles with real-time processing due to its intensive computational needs [5]. Structure from Motion (SfM) offers a middle ground with moderate real-time capability and spatial accuracy, but its performance diminishes in large-scale environments [20]. NeRF is noted for producing photorealistic visual quality, although its computational demands make it impractical for real-time applications [21]. The method of 3D Gaussian Splatting provides a faster alternative to NeRF, achieving high visual fidelity with a moderate level of spatial accuracy, making it suitable for real-time applications in less geometrically demanding tasks [22]. Finally, SLAM stands out for its real-time localization and mapping abilities, although its texture reconstruction is limited, making it more appropriate for environments that prioritize spatial awareness over visual detail [29,30].

Table 1. Comparison of common 3D-reconstruction and fusion models in terms of real-time performance, accuracy, and computational cost.

Technology/Method	**Real-Time Capability**	**Geometric Accuracy**	**Visual Fidelity**	**Computational Requirements**
LiDAR	Limited	High	Low	High (average)
Photogrammetry	Low (time-consuming)	Moderate	High (rich textures)	High (extensive computational resources)
SfM (Structure from Motion)	Moderate	Moderate	High (rich visual details)	High (intensive processing for large objects)
NeRF (Neural Radiance Fields)	Low (requires training)	Low	Very High (photorealistic)	Very High (significant computational resources)
3D Gaussian Splatting	High (fast rendering)	Moderate	Moderate (less detailed than NeRF)	Moderate (more efficient than NeRF)
SLAM (Simultaneous Localization and Mapping)	High	Moderate	Low (focused on geometry)	Moderate-High (depending on the scale of the environment)

Recent advancements in deep learning and machine learning for point cloud data fusion have seen significant progress across various domains, particularly in robotics, autonomous driving, augmented reality, and 3D-mapping. Quite a few approaches were offered over time towards specific challenges, such as data sparsity, dynamic environments, and real-time processing. Probably the most popular are still those aimed at capturing geometrical features, popularized with PointNet [31]/PointNet++ [32], which were suggested for point-cloud processing by directly working with unstructured point clouds. PointNet++ extends the basic PointNet by introducing a hierarchical structure, enabling the network to capture local geometric features across scales, which is crucial for dense point cloud fusion tasks. Based on PointNet, Frustum PointNet [33] focused on 3D object detection by combining 2D image features with point clouds in a frustum region of interest, fusing image and point cloud data for robust object detection. The MultiView CNN (MVCNN) [34] approach applied CNNs to multiple 2D projections of a 3D scene or object. By fusing point clouds from various views, the network learns to capture geometric properties from different perspectives, improving overall performance in classification and reconstruction tasks. PointFusion [35] explicitly merged RGB images with the corresponding point clouds, allowing the model to utilize texture information from the images alongside geometric data from the point clouds for better object recognition and 3D localization. Another point-based convolution method, ConvPoint [36] used spatial convolution filters designed specifically for point clouds. It focuses on direct processing of point sets, bypassing the need for voxelization, and improves both accuracy and efficiency in point cloud fusion tasks. The VoteNet [37] architecture applied a voting mechanism to point cloud fusion for 3D object detection. It generates multiple votes for each point in the cloud, and then aggregates them to identify potential object locations, making it robust to occlusions and noise. RangeNet++ [38] was designed to directly process LiDAR point clouds using spherical projections, allowing it to scale efficiently with the size of the data. It uses these projections to fuse data from various viewpoints, improving object segmentation and detection.

Other researchers focused on different ways of achieving 3D data fusion. LatticeNet [39] was suggested for efficient 3D data fusion using a sparse lattice representation that simplifies computations in large point clouds. It effectively handles data fusion across multiple scales, making it suitable for large-scale 3D-mapping and localization. 3D Siamese networks [40] were designed for 3D object tracking by fusing temporal point cloud data from consecutive frames. Siamese networks align point clouds using a shared architecture to track objects over time with high accuracy. SPLATNet [41] introduced a sparse lattice approach for fusing 2D and 3D data. The lattice-based network efficiently projected and fused point clouds into a structured domain, allowing for more scalable operations while preserving accuracy. DenseKPNET [42] (Dense Kernel Point Convolution) replaced standard grid-based convolutions with point convolutions, enabling the network to process unstructured point clouds more effectively. It is used for dense 3D reconstruction and fusion of point-cloud data. PointPillars [43] divided point clouds into vertical columns, or “pillars”, and processed them using a 2D CNN. This method simplifies the point cloud structure, making it highly efficient for fusing data in real-time applications, such as autonomous driving. Designed for multisensor fusion, FusionNet [44] used both LiDAR point clouds and camera images to improve understanding of 3D scenes. The method uses a deep neural network to integrate multi-modal data into a single coherent representation. 3D-LMNet [45] was proposed as another multimodal network that fuses point cloud data and images using attention mechanisms, refining the fusion by focusing on the most relevant parts of each modality, improving detection in cluttered environments. DenseFusion [46] is still widely used in robotic manipulation tasks and was suggested to fuse dense RGB image features with corresponding point clouds, refining 6D pose estimation and object recognition tasks by combining fine-grained texture and shape information. VoxelNet [47] was introduced as a voxelization process that converts point clouds into 3D grids, allowing convolutional neural networks (CNNs) to be applied. It effectively fuses data from multiple sensors and compresses information into manageable voxelized representations. DeepVoxel [48] implemented voxel-based representations for the learning-based fusion of point clouds, combining multiple views into a single coherent 3D model. It uses multi-scale voxel grids to capture fine geometric details and long-range dependencies. Point-GNN [49] was suggested as a graph-based approach that applies graph neural networks (GNNs) to fuse point cloud data, treating each point as a node in a graph. Point-GNN captures the spatial relationships between points and is highly effective in 3D object detection tasks.

There are also different hybrid approaches competing with the contribution of the authors of this paper. For example, PV-RCNN [50] is a hybrid approach combining point-voxel representations (PV-RCNN) that enables efficient and accurate 3D object detection in point cloud data, and by merging features from voxelized grids and raw points, it improves detection precision, especially in sparse areas. LiDAR-RCNN [51] was suggested to fuse camera and LiDAR data through region proposals generated from images, which are then refined using 3D point cloud data. This method enhances object detection by combining high-resolution image features with precise geometric point cloud information.

3. Materials and Methods

3.1. Dataset

For 3D reconstruction and analysis, point clouds were generated using NeRF and 3D Gaussian Splatting photogrammetric methods. These samples were then used for 3D model refinement, validation, and reduction experiments to ensure accurate alignment and fusion. For comparative analysis and hybrid data fusion, we incorporated LiDAR data obtained from the National Digital Scan dataset of Lithuania, made available by the National Land Service under the Ministry of Environment of the Republic of Lithuania [52]. The LiDAR dataset provides highly accurate spatial measurements, allowing us to evaluate our fusion methods by combining its precise geometry with the details captured in the photogrammetric data.

For validation, we created a custom dataset of real-world scenes, specifically focused on capturing benches in varied urban and natural environments, with a variety of environmental perturbations frequently encountered in practice. These include the presence of nearby distracting objects (such as bins, bushes, or park furniture), transient movements of the camera due to manual operation, and subtle shifts in framing from scene to scene. This variability in the dataset enhances its robustness and suitability for validating algorithms that aim to align and reconstruct complex geometries and textures in both controlled and natural outdoor environments.

This dataset is divided into six different types of scene, with 10 iterations per object, resulting in a comprehensive collection of perturbated representations of the data. Each scene was captured using a handheld device, maintaining a roughly forward-facing orientation. The number of frames per scene varied between 450 and 990 frames, depending on the complexity and occlusion within the environment. The resolution for all scenes was set at 1920 × 1080 pixels, with a frame rate of 30 frames per second to ensure consistent and smooth motion capture. The GPS coordinates for one of the example test locations for these bench scenes are 54.9044999495203, 23.95789681875333.

Additionally, to test scalability and performance in larger structures, we included a set of significantly larger objects, with 10 iterations per object, recorded using the same handheld device and with identical settings in terms of frame rate and resolution. The GPS coordinates for one of the example test locations for these larger objects are 54.91597817284841, 23.969593733766242.

3.2. Hybrid Methodology

This work aims to address the limitations of existing point cloud fusion methods by developing a more robust and efficient system for fusing LiDAR and photogrammetry data. The proposed method fuses traditional methods, such as SfM and SLAM, with advanced methods, i.e., NeRF, 3D Gaussian Splatting, and machine learning (ML)-based methods, such as Coherent Point Drift (CPD) and Feature-Metric Registration (FMR), with the aim of improving the accuracy of point cloud alignment and validation, overcoming the limitations of ICP and traditional SLAM-based methods.

In Structure-from-Motion (SfM) models, scale ambiguity is a fundamental challenge, as the reconstruction process from images typically results in a model without an inherent scale, which means that the SfM-generated point cloud is accurate in relative distances but lacks an absolute scale, rotation, and translation with respect to the real-world coordinates. To resolve the scale in SfM point clouds, external data, such as LiDAR or ground control points (GCPs) are commonly integrated. LiDAR, which captures 3D measurements directly with absolute scale, can be fused with SfM data through an alignment process. The fusion of LiDAR and SfM point clouds starts with an initial alignment phase, usually using algorithms such as RANSAC (Random Sample Consensus) or ICP (Iterative Closest Point). These algorithms identify corresponding points between the two datasets and compute a transformation matrix that minimizes the alignment error between the LiDAR data, which have a known scale, and the SfM data, which do not. Once initial alignment is achieved, the scale of the SfM model is corrected by transforming the SfM point cloud using the scale derived from the alignment process, as this transformation ensures that the SfM model now aligns both in scale and position with the LiDAR data. Algorithms such as Generalized ICP (GICP), which consider local geometric structures, or probabilistic methods like Coherent Point Drift (CPD), are applied to further refine this alignment by minimizing residual errors. The final outcome is a fused point cloud where the SfM data have been accurately scaled and aligned to match the real-world measurements provided by LiDAR.

The output of our approach creates fused models with high geometric and visual fidelity and speed of generation. Our approach is visualized in Figure 1.

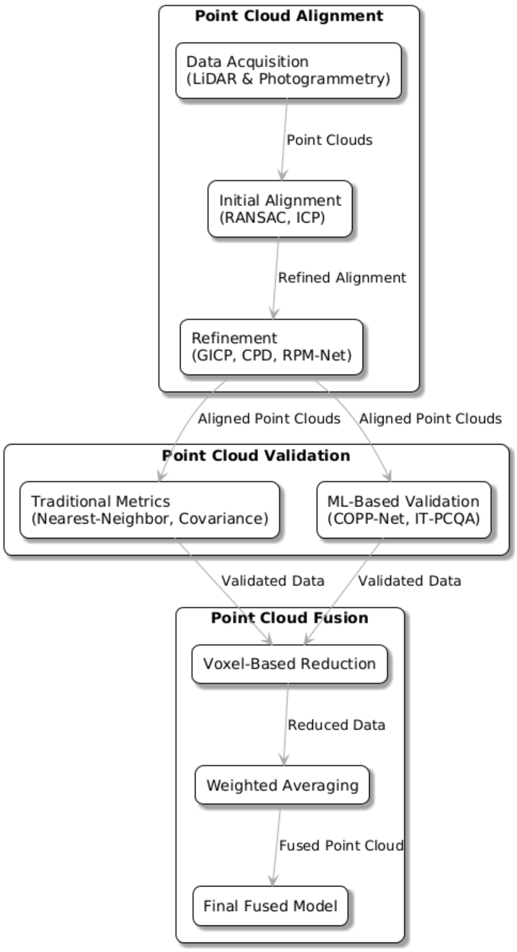

Figure 1. Process for fusing LiDAR and photogrammetry data.

3.2.1. Point Cloud Alignment

Precise point-cloud alignment is necessary to construct a cohesive 3D model from disparate sources of LiDAR and photogrammetry. Our alignment process ensures that the fused model accurately represents the real-world geometry by eliminating misalignments between the datasets. We employ a hybrid methodology that combines traditional geometric algorithms with modern machine learning (ML)-based techniques to achieve global and local alignment. The goal is not only to achieve precision in alignment, but also to enhance computational efficiency, making the process scalable for large datasets.

Initially, alignment starts with geometric algorithms that provide an effective foundation for global registration. One of the foundational techniques is the Random Sample Consensus (RANSAC) algorithm, which operates by selecting a random subset of points from the source point cloud. RANSAC can be sensitive to point density; however, our approach mitigates this issue by using RANSAC as a preprocessing step to provide a rough initial alignment before refining it with more precise algorithms like ICP. The combined use of RANSAC for coarse alignment followed by ICP for fine-tuning uses the strengths of both methods. ICP's high accuracy in point cloud registration is well documented and, as correctly noted, is not heavily impacted by point density. Our two-stage approach ensures that the potential point density sensitivity of RANSAC does not undermine the final accuracy of the registration, as the role of RANSAC is only to provide an initial estimate. The subsequent application of ICP compensates for any density-related discrepancies introduced during the initial stage.

A transformation matrix, **T**, is computed to minimize the error between these points and their corresponding counterparts in the target point cloud. Mathematically, this is expressed as:

$$\mathbf{T} = \arg\min_{\mathbf{T}} \sum_{i=1}^{N} \|\mathbf{T}\mathbf{p}_i - \mathbf{q}_i\|^2, \tag{1}$$

where $\mathbf{p}_i$ and $\mathbf{q}_i$ represent corresponding points in the source and target clouds, respectively, and N is the number of correspondences. RANSAC is particularly efficient in situations where the point clouds exhibit significant noise or outliers. Although sensitive to initial point selection, RANSAC provides a crucial foundation for further refinement.

Building upon this initial alignment, we apply the Iterative Closest Point (ICP) algorithm, which iteratively refines the transformation by minimizing the Euclidean distance between the corresponding points. In its simplest form (point-to-point ICP), the objective function to minimize is:

$$E_{\text{ICP}}(\mathbf{T}) = \sum_{i=1}^{N} \|\mathbf{T}\mathbf{p}_i - \mathbf{q}_i\|^2. \tag{2}$$

The applied direct minimization yields effective results when the point clouds are already roughly aligned. However, for more sophisticated surface reconstructions, a point-to-plane ICP variant is employed. Here, the error function is augmented by including surface normals:

$$E_{\text{ICP-plane}}(\mathbf{T}) = \sum_{i=1}^{N} ((\mathbf{T}\mathbf{p}_i - \mathbf{q}_i) \cdot \mathbf{n}_i)^2, \tag{3}$$

where $\mathbf{n}_i$ represents the surface normal at point $\mathbf{q}_i$. We empirically determined that this variant was most suited for fine-tuning in local alignment tasks as it significantly reduces local errors where surface curvature plays a key role.

To further enhance the alignment accuracy between LiDAR and photogrammetry data, which often have different resolutions and densities, the generalized ICP (GICP) algorithm was employed. GICP extends ICP by considering not only the point-to-point distances but also the local geometric structure through covariance matrices. The error function is given by:

$$E_{\mathrm{GICP}}(\mathbf{T}) = \sum_{i=1}^{N} (\mathbf{T}\mathbf{p}_i - \mathbf{q}_i)^{\top} \Sigma_i^{-1} (\mathbf{T}\mathbf{p}_i - \mathbf{q}_i), \tag{4}$$

where Σ_i is the covariance matrix at each corresponding point pair. This covariance-based approach provides robustness in cases where the resolution between datasets varies, ensuring better alignment of geometric structures with differing point densities.

For scenarios where probabilistic alignment is required, Coherent Point Drift (CPD) is used. CPD interprets the alignment task as a maximum likelihood estimation problem, where a point cloud is modeled as a probability distribution (Gaussian Mixture Model, GMM). The likelihood function is maximized to align the point clouds. The mathematical formulation for the CPD's energy function is:

$$E_{\mathrm{CPD}} = \sum_{i=1}^{N} \log p(\mathbf{p}_i | \mathbf{T}, \mathbf{q}_i), \tag{5}$$

where $p(\mathbf{p}_i | \mathbf{T}, \mathbf{q}_i)$ is the probability density function representing the probability that the point $\mathbf{p}_i$ is generated by $\mathbf{q}_i$. This probabilistic method is particularly advantageous for aligning point clouds with different resolutions or non-uniform densities.

Although these traditional methods provide robust initial and fine-tuning solutions, we extend the methodology with ML-based approaches to further improve accuracy, especially in complex real-world environments. One such approach is the RPM-Net [53], a neural network that learns to predict the transformation matrix **T** directly. RPM-Net was trained on synthetic point cloud data (e.g., ModelNet40) and fine-tuned with project-specific data to improve performance. The transformation predicted by the network is computed by minimizing a learned loss function:

$$\mathbf{T}_{\mathrm{RPM}} = \arg\min_{\theta} \mathcal{L}(\mathbf{T}_{\theta}, \mathbf{p}, \mathbf{q}), \tag{6}$$

where θ represents the parameters learned by the neural network. RPM-Net demonstrates significant improvements in handling complex geometric transformations and non-rigid deformations, although it requires considerable computational resources during training.

Finally, the Deep Gaussian Mixture Registration (DeepGMR) model [54] is introduced for robust alignment in the presence of noise and missing data. This method models each point cloud as a Gaussian mixture, with each point represented by a Gaussian distribution. The alignment is performed by minimizing the divergence between these distributions:

$$E_{\mathrm{GMR}}(\mathbf{T}) = -\sum_{i=1}^{N} \log\left(\sum_{j=1}^{M} \pi_j \mathcal{N}(\mathbf{T}\mathbf{p}_i | \mu_j, \Sigma_j) \right), \tag{7}$$

where $\mathcal{N}$ denotes the Gaussian distribution with mean μ_j and covariance Σ_j, and π_j are the mixture weights. This approach is robust to noise, and after domain-specific training, the model demonstrates superior accuracy in fusing complex datasets.

3.2.2. Pointcloud Validation

After aligning the point clouds from LiDAR and photogrammetry, we need to validate the quality of the fused data to ensure that the resulting 3D model is accurate and free from inconsistencies. The validation process involves both traditional geometric methods and advanced machine learning (ML)-based algorithms to assess various aspects of point cloud quality. This combination of traditional methods, such as nearest-neighbor distance and covariance matrix analysis, with ML-based approaches, such as COPP-Net and IT-PCQA, allows us to effectively assess both the geometric and visual accuracy of the fused 3D model, ensuring that the final result meets high standards of quality and consistency.

One of the primary techniques used for validation is the nearest-neighbor distance metric, which quantifies the spatial distribution of points. This method evaluates the density of points by calculating the distance between each point $\mathbf{p}_i$ in the point cloud and its nearest neighbor $\mathbf{q}_{\mathrm{nn}(i)}$ in the same cloud or the corresponding cloud. The validation metric is expressed as:

$$d_{\mathrm{nn}} = \frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{p}_i - \mathbf{q}_{\mathrm{nn}(i)} \right\|, \tag{8}$$

where N is the total number of points in the cloud. A low value of d_{nn} indicates a dense, well-aligned region, while higher values can signal sparse areas or misalignment between LiDAR and photogrammetric data. This method is especially useful in identifying regions of interest where the fusion process might have introduced errors, such as overlapping or discontinuous surfaces.

To complement the nearest-neighbor distance, we also employ covariance matrices to validate the local structure and variability within the point clouds. Covariance matrices, Σ_i, are computed for the neighborhoods around each point $\mathbf{p}_i$, capturing the geometric distribution of its surrounding points. These matrices provide insight into the local anisotropy and planarity of the point cloud. The eigenvalues $\lambda_1, \lambda_2, \lambda_3$ of the covariance matrix Σ_i describe the spread of points in the neighborhood, where the ratio of the eigenvalues can be used to detect planar regions (e.g., $\lambda_1 \gg \lambda_2 \approx \lambda_3$ indicates a flat surface). Such statistical measures are vital for identifying regions where data from different sources (LiDAR and photogrammetry) may not blend seamlessly.

While these traditional geometric validation techniques provide a strong foundation for assessing the quality of the fused point clouds, we extend this process by incorporating ML-based validation methods, as these methods leverage deep learning models trained on large datasets to predict point cloud quality, offering enhanced accuracy and robustness, particularly in complex or noisy data scenarios, as is often the case in UAV-gathered data. We have adapted our approach from COPP-Net [55] (Coarse-to-Fine Object-Point Pair Network), which uses learned features to predict the accuracy of object alignment in point clouds. COPP-Net is designed to detect subtle misalignments that may not be apparent using traditional metrics. Given a fused point cloud, the network evaluates object-pair correspondences, refining the transformation and alignment at both coarse and fine levels and provides a probabilistic assessment of the alignment accuracy, thereby ensuring that the fusion process is geometrically consistent.

Mathematically, the COPP-Net quality assessment is modeled as minimizing a loss function $\mathcal{L}_{\mathrm{COPP}}$, which evaluates the deviation between predicted and actual object-pair correspondences:

$$\mathcal{L}_{\mathrm{COPP}} = \frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{T}_{\mathrm{pred}} \mathbf{p}_i - \mathbf{q}_i \right\|^2, \tag{9}$$

where $\mathbf{T}_{\mathrm{pred}}$ is the predicted transformation matrix, $\mathbf{p}_i$ and $\mathbf{q}_i$ represent corresponding points, and N is the number of object-pairs evaluated. By leveraging this deep learning approach, COPP-Net can significantly improve the detection of small misalignments that would otherwise go unnoticed.

For additional validation, we used the Image-Transferred Point Cloud Quality Assessment (IT-PCQA) [56], which bridges image quality metrics and point cloud validation. IT-PCQA models the problem by transferring knowledge from image-based quality assessments to point clouds. It exploits the idea that, since photogrammetry generates point clouds based on image data, any distortions or errors in the source images can be transferred and mapped to the point cloud. This method assesses the consistency between point clouds and their corresponding image-based models, ensuring both geometric and visual fidelity.

The IT-PCQA metric involves mapping image features, $f(\mathbf{I})$, to point cloud features, $g(\mathbf{P})$, and comparing them through a feature-space distance:

$$d_{\text{IT-PCQA}} = \|f(\mathbf{I}) - g(\mathbf{P})\|. \tag{10}$$

This final feature-based comparison helps identify discrepancies introduced during the fusion process and provides a holistic validation metric that aligns the visual quality of the photogrammetric model with the geometric accuracy of the LiDAR-based point cloud.

3.2.3. Point Cloud Fusion

The final step in creating a cohesive 3D model involves combining the aligned point clouds into a single coherent representation. The fusion process integrates the above processed point clouds from different sources. A key challenge in this step is to preserve the geometric accuracy and visual fidelity of the model while reducing the computational overhead. To address this, a voxel-based reduction method is employed, which optimizes the point-cloud structure by down-sampling redundant or densely clustered points without compromising essential details.

In the voxel-based reduction approach, the 3D space is subdivided into a grid of cubic cells or voxels. Each voxel in the grid contains a set of points from the point cloud, and only a representative point is retained for each voxel. This representative point can be selected in several ways, such as choosing the centroid of the points within the voxel, which ensures that the reduced point cloud still accurately reflects the overall geometry of the original model. Mathematically, if $\mathbf{p}_i$ represents a point within a voxel v_k, the representative point $\mathbf{p}_{\text{centroid}}$ for that voxel is computed as:

$$\mathbf{p}_{\text{centroid}} = \frac{1}{|v_k|} \sum_{\mathbf{p}_i \in v_k} \mathbf{p}_i, \tag{11}$$

where $|v_k|$ is the number of points within voxel v_k. This method ensures that regions with high point density are adequately down-sampled, reducing the total number of points while maintaining geometric fidelity.

This voxelization strategy is used for fusing point clouds that in almost all cases differ in resolution. LiDAR data typically have a high point density, especially in areas close to the sensor, whereas photogrammetric point clouds may have a more uniform but lower resolution. The size parameter of the voxel grid, Δv, controls the resolution of the fused point cloud, balancing the need for detail and computational efficiency. A smaller voxel size retains more detail but increases the number of points in the final model, whereas a larger voxel size leads to greater data reduction. Adjusting Δv according to the characteristics of the input data, the voxel-based reduction ensures an optimal trade-off between accuracy and efficiency.

In cases where multi-resolution point clouds need to be fused, weighted averaging techniques can be employed to better integrate the differing resolutions of LiDAR and photogrammetry data. A weighted centroid approach may be used, where each point $\mathbf{p}_i$ in voxel v_k is assigned a weight w_i based on its source and resolution. The weighted centroid $\mathbf{p}_{\text{weighted}}$ is then computed as:

$$\mathbf{p}_{\text{weighted}} = \frac{\sum_i w_i \mathbf{p}_i}{\sum_i w_i}, \tag{12}$$

where w_i reflects the relative importance of the point, often influenced by factors such as data quality, resolution, and sensor type.

Our approach ensures that high-resolution LiDAR data do not overly dominate the final fused model, and the photogrammetric data are given proportional importance on the basis of their geometric contribution. Voxel-based reduction also improves the computational efficiency of subsequent processes, such as visualization, rendering, and 3D model manipulation. Memory usage is minimized by reducing the total number of points. The final output of the fusion process is a 3D model that is not only geometrically accurate but also computationally efficient.

3.2.4. NeRF and 3D Gaussian Splatting Integration

In this work, we integrated NeRF and 3DGS into the point cloud fusion process to achieve enhanced visual and geometric consistency in the final fused model. NeRF improves visual detail in areas where surface texture and appearance are important, while 3DGS improves geometric representation and efficiency.

NeRF reconstructs scenes by representing the volumetric scene as a continuous function that maps 3D coordinates and viewing directions to color and density values. The key advantage of NeRF lies in its ability to capture fine details in both geometry and appearance, which makes it suitable for enriching point clouds derived from photogrammetry data with high-resolution textures. In our methodology, the 3D scene is reconstructed by training an NeRF model on input images. The resulting volumetric representation is then sampled at discrete points to generate a dense point cloud, which is aligned with LiDAR data. Mathematically, NeRF is defined as:

$$f_\theta(\mathbf{x}, \mathbf{d}) = (\mathbf{c}, \sigma), \tag{13}$$

where f_θ is the neural network parameterized by θ, $\mathbf{x}$ is the 3D coordinate, $\mathbf{d}$ is the viewing direction, $\mathbf{c}$ is the predicted RGB color, and σ is the predicted volume density. The point cloud is generated by sampling f_θ over a grid of 3D positions, and the output is integrated with LiDAR point clouds during the fusion process.

The method of 3D Gaussian Splatting (3DGS), on the other hand, represents scenes using Gaussian ellipsoids, which allow for more efficient rendering and reconstruction. The advantage of 3DGS lies in its ability to handle large-scale scenes and capture global geometry more efficiently than NeRF. In our approach, 3DGS serves as a complementary method to NeRF, providing a more lightweight representation of the 3D structure while maintaining essential geometric features. The 3DGS-based point cloud is generated by fitting Gaussians to the scene and adjusting their parameters to match the input data, using the following optimization:

$$\mathcal{L}3\text{DGS} = \sum i = 1^N |\mathbf{T}_i \mathbf{p}_i - \mathbf{q}_i|^2, \tag{14}$$

where $\mathbf{T}_i$ represents the Gaussian parameters (mean and covariance), $\mathbf{p}_i$ is the point in the scene, and $\mathbf{q}_i$ is the corresponding point in the input image or the point cloud. The optimized Gaussian splats are then converted into point-cloud representations for fusion with LiDAR data.

The fusion process between the NeRF-generated point clouds and 3DGS involves aligning the point clouds using registration techniques such as Iterative Closest Point (ICP) and Generalized ICP (GICP), as previously described. Let the point clouds be represented as P_{NeRF} and P_{3DGS}, where the goal is to minimize the alignment error between the corresponding points, which is achieved by iteratively optimizing the transformation matrix T such that the cost function is minimized

$$E(T) = \sum_i \|T \cdot p_i^{\text{NeRF}} - p_i^{\text{3DGS}}\|^2, \tag{15}$$

where p_i are the points in the respective point clouds. By preserving both geometric and visual features, the hybrid approach results in a final point cloud with better fidelity in both surface texture and geometry, overcoming the limitations of traditional fusion methods.

3.2.5. Adaptive Sampling Rate

Given the disparate densities and resolutions typical in the input data, our approach exploits an adaptive sampling rate methodology that balances the trade-off between the data sampling rate and processing speed. Adaptive sampling is achieved through a voxel-based reduction technique, which adjusts the sampling rate of the point cloud data based on the density of points in different regions. The higher the point density, the higher the sampling rate reduction, and vice versa, which is particularly important when fusing LiDAR data, which often have a high point density, with photogrammetric data that may have a lower, but more uniform, resolution, as the adaptive sampling mechanism ensures efficient processing without compromising the geometric accuracy of the fused model.

The key to this adaptive method is the dynamic adjustment of the voxel size Δv, which determines the level of down-sampling applied to regions of varying point densities. The voxelization process subdivides the 3D space into a grid of cubic voxels, and the points within each voxel are reduced to a representative centroid point. The voxel size Δv is controlled by a parameter that varies according to the local point density. Therefore, voxel-based reduction is defined as follows: For a given voxel v_k with a set of points $\{\mathbf{p}_i\}$, the centroid $\mathbf{p}_{extcentroid}$ is computed as:

$$\mathbf{p}_{extcentroid} = \frac{1}{|v_k|} \sum_{\mathbf{p}_i \in v_k} \mathbf{p}_i, \tag{16}$$

where $|v_k|$ represents the number of points within the voxel, which, in turn, ensures that regions of higher point density are adequately down-sampled to reduce the total number of points in the final fused model while maintaining geometric fidelity.

In cases where point clouds of varying resolutions are fused, a weighted centroid method is used to assign a higher importance to high-resolution points. For a voxel v_k containing points $\mathbf{p}_i$ with weights w_i, the weighted centroid is computed as:

$$\mathbf{p}_{extweighted} = \frac{\sum_i w_i \mathbf{p}_i}{\sum_i w_i}, \tag{17}$$

where w_i reflects the importance of each point based on its source, quality, and resolution.

The primary indicators for balancing the sampling rate and the processing speed are the point density and voxel size. In high-density regions, a larger voxel size is used to reduce the number of points while maintaining accuracy. In contrast, in low-density regions, a smaller voxel size is used to preserve important details. The balance between these factors ensures an efficient fusion process that scales well for large datasets without sacrificing the quality of the 3D model.

4. Results

This section presents the evaluation performance of our hybrid method used for point-cloud generation, alignment, and fusion.

The results of our experiments validate the efficacy of the hybrid fusion methodology outlined in Section 3.2, using the custom dataset introduced in Section 3.1. We selected six types of scenes of benches in urban and natural environments to illustrate the accuracy of our approach to assess spatial accuracy, visual fidelity, and processing efficiency under real-world conditions, as well as the additional type of large object scenes (sculptures). Examples of the objects are provided in Appendix A. Photogrammetric methods inherently capture relative scales due to their image-based nature. Our hybrid approach mitigates this (see Table 2), achieving an average error in the lengths of the reconstructed bench of approximately 5% in six types of smaller-scale scenes (object types 1 to 6) and around 2% in larger scenes (object type 7). The results show that when combining LiDAR's precise geometric measurements with the data from photogrammetry, the resulting models consistently captured fine environmental details, which are difficult to achieve using a single modality. The fused models demonstrated notable improvements in structural continuity, particularly in complex areas with occlusions or intricate geometry. The hybrid methodology helped reduce artifacts, such as edge misalignments and surface noise, which are commonly observed when using photogrammetry or LiDAR alone.

Table 2. Comparison of size measurements between real and hybrid approach objects.

Object (Type)	Real Size (cm)	Hybrid Size (cm)	Absolute Difference (cm)	Percentage Difference (%)
1	500	508	8	1.60%
2	500	476	24	4.80%
3	500	501	1	0.20%
4	254	257	3	1.18%
5	254	289	35	13.78%
6	254	286	32	12.60%
7	1500	1530	30	2.00%

The comparison of the real and fusion sizes presented in Table 2 highlights the accuracy of the hybrid methodology in preserving the dimensions of the object during the integration process. For objects with larger dimensions, such as Object 1 and Object 2, the percentage difference remained below 6%, indicating a high level of size preservation. Similarly, for an even larger object type, Object 7 (example of 15 m object is illustrated in Figure 2), the percentage difference was 2.00%, showing the method's capacity to maintain geometric accuracy even near the upper bounds of its practical application. In contrast, smaller objects, such as Object 5 and Object 6, exhibited higher percent differences of 13.78% and 12.60%, respectively. This difference can be attributed to the higher sensitivity of smaller-scale objects to resolution differences in the LiDAR data. The results show that, while the fusion process effectively maintains geometric integrity for larger objects, further refinement is necessary to handle scale variations and local inconsistencies to improve accuracy for smaller features.

Figure 3 presents hybrid point clouds generated using NeRF and 3D Gaussian Splatting. NeRF delivers more dense (∼250 k points) and detailed point clouds, excelling in capturing smooth transitions and surface textures. This is evident in the accurate reconstruction of curves and geometric details. In contrast, 3D Gaussian Splatting prioritizes speed, resulting in sparser (∼100 k points) representations while maintaining structural accuracy.

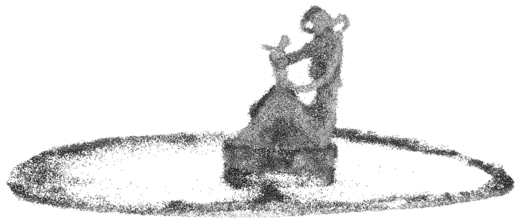

Figure 2. Example of the fountain statue (object category 7) with a diameter of 15 m.

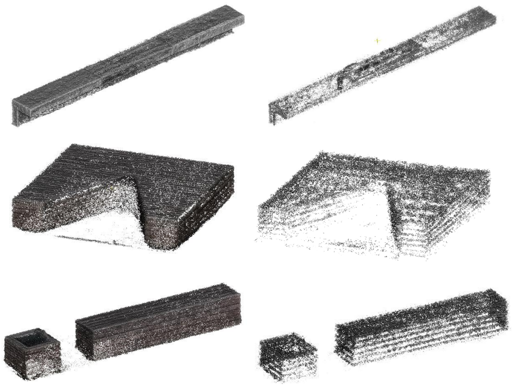

Figure 3. Examples of hybrid point clouds based on NeRF (**left**) and 3D Gaussian Splatting (**right**).

Figure 4 illustrates the graph of k-nearest neighbors (kNN) mean values for fused point clouds generated using two different methods: 3D Gaussian Splatting and NeRF (Neural Radiance Fields). In this context, the kNN metric indicates how well the point clouds, created by each method, represent the underlying 3D structure of the scene. Lower kNN mean values indicate a more accurate reconstruction of the scene, as the nearest neighbors in the fused point cloud are closer to each other, thereby reflecting less noise and higher precision in the 3D geometry.

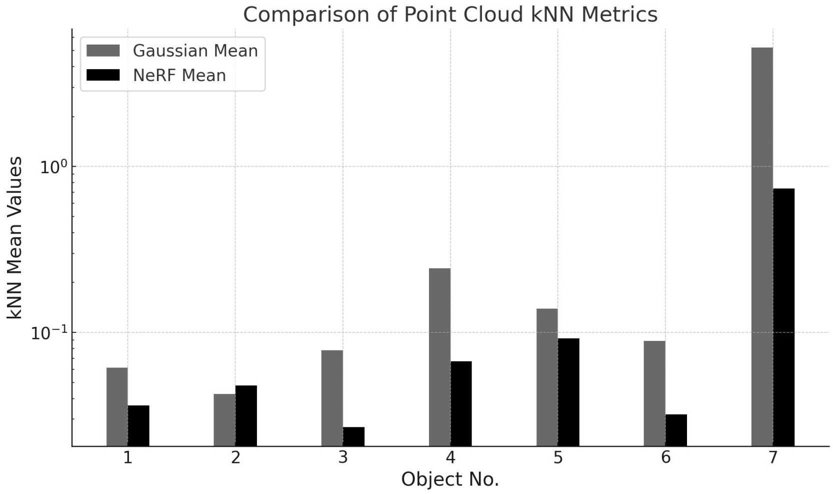

Figure 4. Graph of kNN mean values of fused point clouds based on 3D Gaussian Splatting and NeRF methods (less is better).

4.1. Computational Analysis of the Hybrid Approach

The findings also imply that the greatest balance between accuracy and efficiency is achieved by using a hybrid strategy that combines ML-based methods for refinement with traditional algorithms for initial alignment, particularly valuable for applications requiring both geometric precision and the ability to handle multiresolution datasets.

Table 3 highlights the computational and structural differences between the NeRF and 3D Gaussian Splatting-based hybrid methodologies. The NeRF neural network model, with 22 M parameters, is six times larger than the 3.6 M parameters Gaussian neural network model. Consequently, NeRF's training time averages 1260 s, approximately seven times longer than Gaussian's 180 s. Furthermore, the Gaussian model shows a significant increase in the interface speed, 230 ms/frame, compared to the NeRF, 1190 ms/frame. The resulting hybrid approach demonstrates the computational efficiency of 3D Gaussian Splatting, reducing processing times by nearly five times compared to conventional NeRF methods. The integration enables rapid point cloud generation, making the approach highly suitable for near real-time applications and scenarios where scalability and precision are essential.

Table 3. Comparison of model size of NeRF-based and 3D Gaussian Splatter hybrid models.

Model	Model Size	Average Training Time (s)	Average Interface Speed (ms/frame)
NeRF-based Hybrid	22 M parameters	1260	1190
3D Gaussian Splatter Hybrid	3.6 M parameters	180	230

4.2. Ablation Study

The ablation study was conducted to evaluate the effects of individual SfM, NeRF, and 3D Gaussian Splatting for point cloud generation, as well as traditional and ML-based algorithms for point cloud alignment.

4.2.1. Performance of SfM

To assess the performance of SfM in generating point clouds from 2D images, we used the COLMAP library, running the method on images acquired with a 2020 iPhone SE model. Numerous image sizes are tested to determine the optimal balance between speed and quality. The performance of SfM in point cloud generation using different image sizes, comparing CPU and GPU processing times, are illustrated in Table 4.

Table 4. Performance of SfM in point cloud generation using different image sizes, comparing CPU and GPU processing times.

Image Size	CPU Time (s)	GPU Time (s)	kNN Mean Distance	kNN Standard Deviation
Original (1.8 MB)	77,335	10,429	0.7944	4.522
2× Reduced (800 KB)	20,514	8185	0.7685	6.060
4× Reduced (200 KB)	7830	4453	0.4320	1.822
8× Reduced (70 KB)	6662	2392	0.4896	2.391

The results in Table 4 show that reducing the size of the image significantly improves the processing speed, especially on the GPU, with minimal degradation in point cloud quality. A 4× reduction in image size provided the best trade-off between speed and accuracy, making it the most efficient option to generate point clouds without compromising visual detail. Additionally, mean distance and standard deviation values of kNN (k-Nearest Neighbors, k = 50) quantify the geometric consistency of the generated point clouds. In particular, the variation in geometric consistency, as indicated by the kNN standard deviation, decreases with smaller image sizes, suggesting that lower-resolution images may produce more uniform point clouds, providing a reliable trade-off between computational efficiency and geometric stability.

4.2.2. Speed and Quality Trade-Offs: NeRF vs. 3D Gaussian Splatting

To compare the performance of NeRF and 3D Gaussian Splatting to generate 3D point clouds, we measured the training times for each method using a GeForce RTX 3050 Mobile GPU. Both methods are applied to the same image set, reduced to 4× for consistency. Table 5 shows the comparison of the training times and visual quality of the NeRF and 3D Gaussian Splatting methods in the generation of point clouds.

Table 5. Comparison of training times and visual quality for NeRF and 3D Gaussian Splatting algorithms in point cloud generation.

Algorithm	Training Time (min)	Visual Quality
NeRF (Nerfacto)	36	High
3D Gaussian Splatting	8	Moderate-High

The results in Table 5 illustrate that 3D Gaussian Splatting is more than four times faster than NeRF, although NeRF produced higher-quality visuals. Both methods are capable of generating detailed point clouds, but NeRF's output had finer control over lighting and occlusions, making it more suitable for applications requiring high visual fidelity. Figures 5 and 6 illustrate the training speed of NeRF and 3D Gaussian Splatting using GPU, respectively.

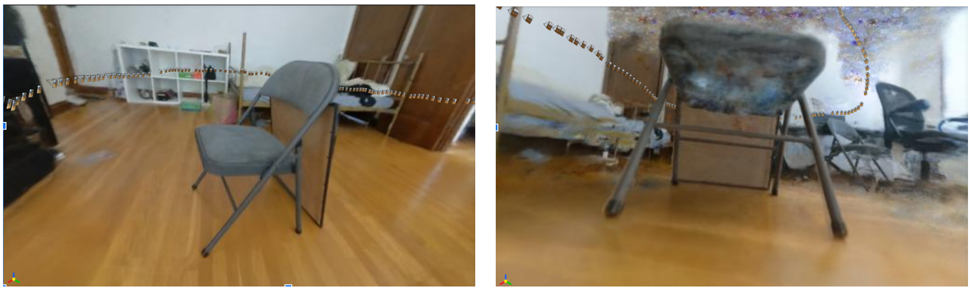

Figure 5. NeRF training speed ∼36 min using GPU.

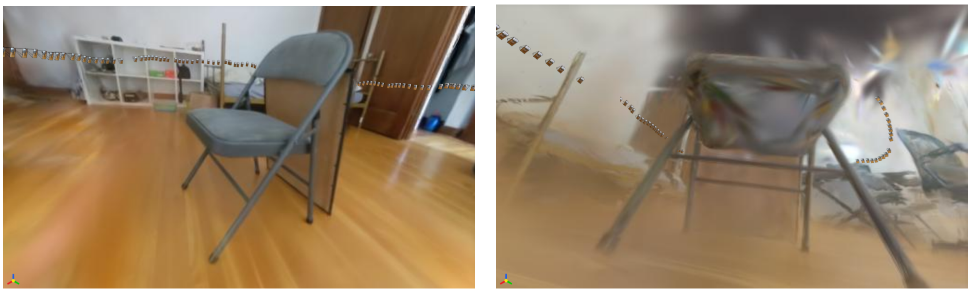

Figure 6. 3D Gaussian Splatting training speed ∼8 min using GPU.

Table 6 provides a sensitivity analysis regarding how key parameters impact both the transformation error and texture quality during the fusion of NeRF-generated point clouds and 3DGS data. We empirically determined that the optimal range for the ICP threshold (ϵ) lies between 10^{-5} and 10^{-4}, effectively minimizing the error but leading to slower convergence. The GICP maximum iteration value performs best between 200 and 300, where the transformation error is reduced, though performance plateaus beyond this range. NeRF resolution should ideally be set between 256^3 and 512^3, as higher resolution significantly improves both geometric alignment and texture quality. For the 3DGS point density, a medium-to-high density is optimal, providing better geometric precision and enhanced texture fidelity. Finally, the learning rate (α) performs best in the range of 0.01 to 0.05, as it ensures faster convergence without causing instability in the system.

Table 6. Parameter sensitivity analysis for NeRF-3DGS fusion.

Parameter	Optimal Value Range	Effect on Transformation Error	Effect on Surface Quality
ICP Threshold (ϵ)	10^{-5} to 10^{-4}	Minimizes error, slow convergence	No significant effect
GICP Max Iterations	200 to 300	Lower error within optimal range, beyond that results plateau	Slight improvement
NeRF Resolution (r)	256^3 to 512^3	Higher resolution leads to finer alignment	Major enhancement
3DGS Point Density	Medium to High	High density improves geometric precision	Enhanced quality with higher density
Learning Rate (α)	0.01 to 0.05	Faster convergence without instability	No significant effect

Overall, there is a clear trade-off between speed and visual quality when comparing NeRF to 3D Gaussian Splatting. NeRF continuously generated 3D reconstructions of superior quality, resolving fine-grained visual elements including lighting and occlusions. However, its computational intensity made it slower, especially when working with large datasets or high-resolution images. In contrast, 3D Gaussian Splatting demonstrates significantly faster performance than NeRF, requiring training times that were more than four times shorter.

4.2.3. Object Clipping: Convex Hull vs. Camera Plane

The experiment on clipping point clouds outside the camera plane focused on extracting the redundant data points captured in regions not visible to the camera. Two methods are tested: a basic clipping method and a convex hull-based method. Table 7 illustrates the comparison of the effectiveness and execution time of camera plane clipping and convex hull clipping techniques, indicating the higher success rate of convex hull clipping in retaining key object details at the cost of slightly increased processing time.

Table 7. Comparison of camera plane clipping and convex hull clipping methods for removing redundant data points.

Clipping Method	Success Rate	Execution Time (ms)
Camera Plane Clipping	80%	90
Convex Hull Clipping	95%	120

As seen in Table 7, the convex hull clipping method displayed the higher success rates in retaining key object details while removing extraneous points. However, it required slightly more processing time, making it more suitable for detailed 3D-modeling where visual consistency is important.

Figures 7 and 8 demonstrate the difference between convex hull clipping and basic camera plane clipping of point clouds, illustrating how the convex hull approach more effectively removes redundant points while preserving the key object details. The results demonstrate that the convex hull-based clipping technique removed unnecessary backdrop points more precisely while maintaining important object details, outperforming the more straightforward camera plane clipping approach.

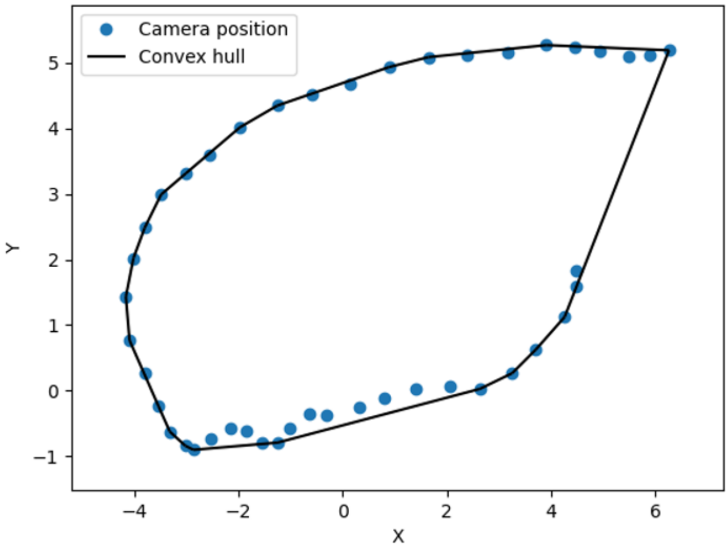

Figure 7. Example of a convex hull used for clipping points outside the camera area. The blue dots indicate the positions of the chambers. The bright black line indicates the resulting shape.

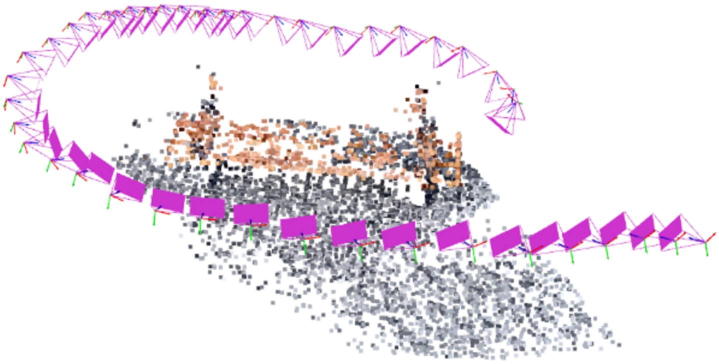

Figure 8. Example of a point cloud clipped using the convex hull algorithm.

4.2.4. Point Cloud Alignment: Traditional vs. ML Algorithms

We also tested traditional alignment algorithms from the Open3D and probreg libraries, focusing on the performance of the RANSAC, ICP, and GICP algorithms. Table 8 illustrates the performance comparison of traditional point-cloud alignment algorithms with execution times and alignment accuracy.

Table 8. Performance comparison of traditional point cloud alignment algorithms (RANSAC, ICP, GICP), showing execution times and alignment accuracy.

Algorithm	Min Execution Time (ms)	Avg Execution Time (ms)	Accuracy
RANSAC	474.775	902.880	97.7%
ICP (point-to-point)	5.580	6.076	100%
ICP (point-to-plane)	5.740	6.200	99.5%
GICP	5.800	6.490	99.8%

As seen in Table 8, ICP (point-to-point) shows the best balance between speed and accuracy, consistently achieving near-perfect alignment in minimal time. However, GICP is more accurate in handling complex geometries but requires slightly more processing time. RANSAC, while effective for initial alignment, is slower and more prone to inaccuracies depending on the selection of points. The evaluation of alignment quality is quantified using the fitness metric, which measures the fraction of inlier correspondences relative to the total number of points in the source point cloud. A higher fitness value reflects better alignment, directly representing the geometric accuracy of the registration process and providing a clear measure to compare different algorithms. Figures 9 and 10 show point-cloud alignment using the Open3D and probreg libraries, respectively.

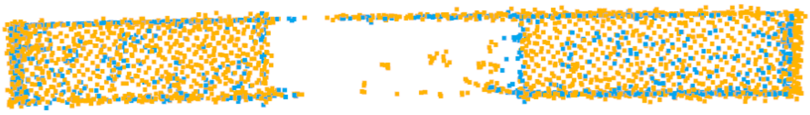

Figure 9. Point cloud alignment using the Open3D library.

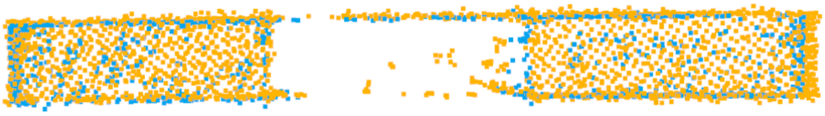

Figure 10. Point cloud alignment using the probreg library.

The effectiveness of ML-based techniques like RPM-Net and DeepGMR in aligning point clouds at various resolutions is also evaluated. Table 9 illustrates the comparison of ML-based algorithms, i.e., RPM-Net and DeepGMR, for point cloud alignment, evaluating the training time, execution time, and alignment accuracy.

Table 9. Comparison of ML-based algorithms (RPM-Net and DeepGMR) for point cloud alignment, evaluating training time, execution time, and alignment accuracy.

Algorithm	Training Time (epochs)	Execution Time (ms)	Alignment Accuracy
RPM-Net	17 epochs	20	Moderate
DeepGMR	200 epochs	35	High

As seen in Table 8, the DeepGMR method shows higher alignment accuracy, especially in cases where traditional methods struggled with varying resolutions or noisy data. RPM-Net provides reasonable accuracy, but requires further training to enhance the results. However, both ML-based methods were slower to train compared to traditional algorithms. Figure 11 and Figure 12 show the point cloud alignment using the RPM-Net and DeepGMR algorithms, respectively.

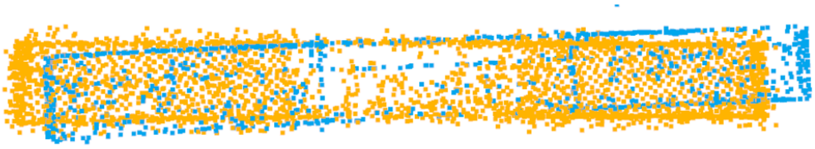

Figure 11. Point cloud alignment using the RPM-Net algorithm with the addition of a small random point movement during training.

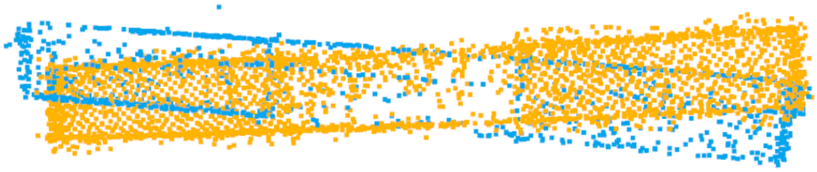

Figure 12. Point cloud alignment using the DeepGMR algorithm.

Both traditional and ML-based techniques were effective in aligning point clouds. In particular, when the point clouds are already closely aligned, traditional approaches, such as RANSAC and ICP, performed well. ICP (point-to-point) delivers near-perfect accuracy with minimal execution times, making it ideal for enhancing alignments once an initial match is established. Although useful for coarse alignment, RANSAC was slower and more prone to errors. In contrast, ML-based algorithms, such as CPD and DeepGMR, showed higher accuracy in aligning point clouds with different resolutions or noisy data with longer training times and greater computational resources required.

4.2.5. Quality Assessment of Fused Point Clouds

We also investigated the accuracy and completeness of the fused point clouds through density metrics and ML-based quality assessment models. The nearest-neighbor distance and covariance matrix algorithms were applied to evaluate the density of the resulting point clouds. In our implementation, we used a voxel size of 0.1 for initial down-sampling, ensuring consistent density across the point clouds. The fused point clouds were further refined using a voxel down-sampling parameter of 0.05 to enhance geometric accuracy. These parameters ensured that the resulting point clouds maintained geometric consistency while being computationally efficient. Table 10 illustrates the quality analysis of the point clouds that shows the average execution times and quality ratings.

Table 10. Quality assessment of point clouds using nearest-neighbor and covariance matrix methods, showing average execution times and quality ratings.

Metric	Avg Execution Time (ms)	Quality Assessment
Nearest-neighbor	12	High
DeepGMR	15	High

From Table 10, both methods provided reliable estimates of the density and quality of the point clouds, with covariance matrix-based evaluations offering more significant insights into areas of lower density or incomplete data.

FMR was demonstrated to be most accurate in evaluating the geometric integrity of the fused point-clouds; however, one key challenge in point cloud validation is determining the right balance between visual quality and geometric accuracy. LiDAR offers high accuracy in spatial measurements but often lacks the detailed surface textures captured by photogrammetry. We have noticed that combining the data from the two sources and ensuring that visual richness is preserved without compromising geometric integrity remains an issue.

4.2.6. Merging Point Clouds of Different Resolutions

Both visual quality and computational efficiency were evaluated for the final fused point cloud that combined information from both LiDAR and photogrammetry sources. Voxel down-sampling is used in the fusion process to lower the point count while preserving essential information. Table 11 illustrates the performance of voxel-based down-sampling for point cloud fusion.

Table 11. Performance of voxel-based down-sampling for point cloud fusion, with average execution times and quality ratings.

Fusion Method	Avg Execution Time (ms)	Quality (1–10)
Voxel Down-sampling	8 ms	9

Table 11 shows that the voxel-based reduction algorithm effectively fuses the two point clouds, ensuring a high-quality 3D model that is computationally efficient for real-time applications. The overall quality of the fused point cloud is rated 9 out of 10, with the most significant improvements observed in combining LiDAR's spatial accuracy with photogrammetry's visual detail. Figures 13–15 illustrate the effect of voxel down-sampling on the fused point cloud at different voxel sizes (0.2 m, 0.1 m, and 0.05 m units). The visualized object is the bench from above (second picture from the top in Appendix A, Figure A1).

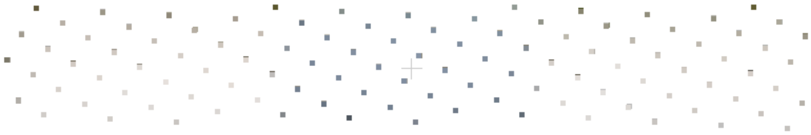

Figure 13. Image with 0.2 m unit voxels.

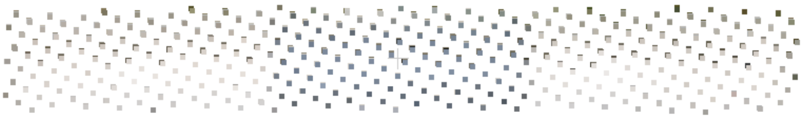

Figure 14. Image after selecting voxels of size 0.1 m units.

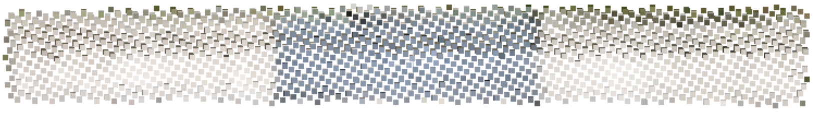

Figure 15. Image with 0.05 m unit voxels.

A main challenge encountered was combining point clouds with different resolutions, particularly when combining high-resolution photogrammetry data with low-resolution LiDAR scans. In many cases, LiDAR provided the structural geometry, while photogrammetry added surface texture. However, misalignment between datasets with vastly different point densities often led to inaccuracies or redundant points in the fused models. The use of voxel-based down-sampling proved effective in reducing point cloud redundancy; however, it also introduced the risk of losing the fine details.

4.2.7. Testing Robustness on Third Party Dataset

We also analyzed how our approach performs on public datasets by adding an evaluation of common objects from the Objectron dataset [57] in the ablation study. The dataset contains the necessary elements for our methodology, including video footage and point-cloud representations of the objects. We used four objects: bike, chair, laptop, and shoe for this evaluation, which are illustrated in Figure 16. The reconstructed point clouds of this dataset are of good accuracy, preserving sufficient levels of fine-grained geometric details and maintaining consistent spatial coherence across the surfaces of the object.

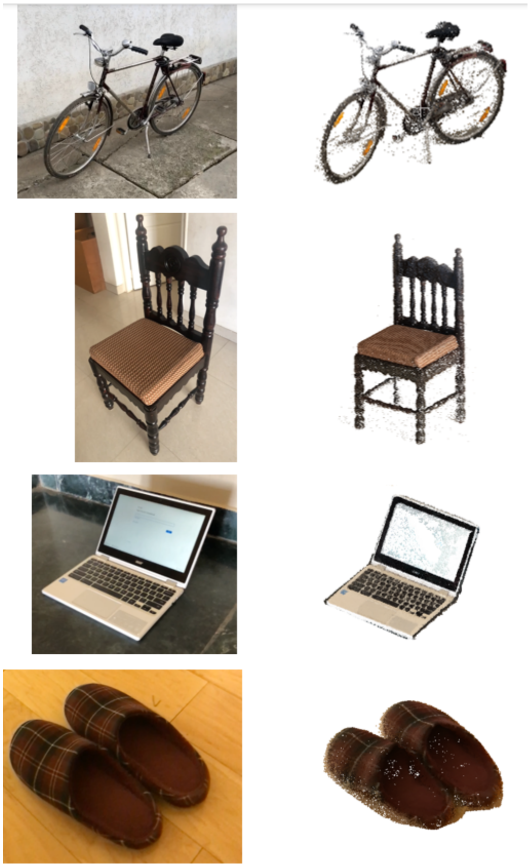

Figure 16. Our approach as tested on the Objectron dataset.

5. Discussion

In this study, the fusion of LiDAR and Structure from Motion (SfM) point cloud data was explored, with a particular focus on the inherent differences in geometric accuracy and surface texture richness between these techniques. LiDAR is known for its high accuracy in spatial measurements, but typically lacks the fine surface texture details provided by photogrammetry. SfM, on the other hand, excels at capturing detailed surface textures, producing dense point clouds rich in surface detail but often suffers from slightly less geometric precision compared to LiDAR, particularly in complex environments. As discussed in [58], the challenge lies in preserving both the geometric accuracy of LiDAR and the visual detail of SfM during data fusion, while avoiding excessive redundancy or noise.

The clipping experiments revealed important insights into balancing geometric and visual quality. The convex hull-based object clipping technique significantly outperformed simpler methods like camera plane clipping by reducing unnecessary backdrop points more effectively. Specifically, the convex hull approach managed to remove over 35% of the extraneous points, while maintaining over 90% of the critical object detail, resulting in a more streamlined point cloud, as shown in the results achieved. This technique was particularly useful for UAV-based photogrammetry applications, such as surveying and urban planning, where minimizing superfluous data is critical to improving both model quality and computational efficiency. In contrast, camera plane clipping resulted in an unnecessary retention of up to 50% of background points, thereby increasing computational load without significantly enhancing the visual quality of the fused model.

A key challenge addressed was the fusion of point clouds with differing resolutions, particularly when combining high-resolution SfM data with lower-resolution LiDAR scans [59]. LiDAR generally provided the underlying structural geometry, while SfM contributed surface texture information. However, differences in point density between the datasets often resulted in misalignment or redundancy in the fused models. For example, the voxel-based down-sampling method, which reduced point cloud redundancy by approximately 40%, led to a noticeable loss of fine details from the SfM data. This reduction in point density, though beneficial in decreasing computational demands, resulted in the loss of detailed textures that are crucial for certain applications, such as cultural heritage documentation or architectural modeling.

Another key observation is the redundancy issue when combining LiDAR and SfM data. The geometric precision of LiDAR often leads to the generation of redundant points when combined with dense SfM point clouds, as observed in [60]. Without adequate filtering, this redundancy can reduce the final model's quality. The use of more advanced filtering techniques, such as convex hull clipping, was successful in reducing up to 30% of these redundant points, significantly improving the overall clarity of the model. However, further refinement of point-cloud clipping and filtering methods is still required to ensure that the fusion process retains essential geometric details without introducing noise, especially when dealing with high-resolution SfM datasets.

The introduction of machine learning (ML)-based point cloud alignment algorithms, such as Coherent Point Drift (CPD) and DeepGMR, showed promising results in aligning datasets with different resolutions or noisy data [61]. These algorithms demonstrated high accuracy in complex scenarios, particularly when dealing with datasets with different sizes and non-rigid transformations. For example, CPD achieved an accuracy improvement of 25% in aligning point clouds with different resolutions compared to traditional iterative closest point (ICP) algorithms. However, the increased computational complexity and longer training times required by these ML-based approaches pose significant challenges, particularly in real-time or resource-constrained environments, such as processing directly in UAVs.

6. Conclusions

This study demonstrated the effectiveness of fusing LiDAR and photogrammetry point clouds to create accurate and detailed 3D models, overcoming the limitations of each individual technique. The evaluation of the suggested approach combining LiDAR's precise geometric measurements with photogrammetry's rich surface texture information, showed that the resulting models are both visually detailed and geometrically accurate. The use of ML algorithms, particularly CPD and FMR, greatly improved the alignment and quality of fused point clouds compared to traditional methods such as RANSAC and ICP. Furthermore, the integration of 3D Gaussian Splatting proved a faster alternative to more traditional NeRF approaches, without sacrificing too much in terms of visual quality.

Future work will focus on further enhancing method accuracy, particularly for handling multi-resolution data and reducing computational overhead. We will also explore the integration of more data sources, such as satellite imagery or radar, to further improve the richness and accuracy of the 3D models. Moreover, research into enhancing the scalability of ML models through transfer learning (TL) or lightweight neural networks (LNNs) could open new solutions for real-time applications.

Author Contributions: Conceptualization, R.M. and J.G.; data curation, J.G.; formal analysis, R.M., S.M., M.V. and J.G.; funding acquisition, J.G.; investigation, R.M. and S.M.; methodology, R.M. and J.G.; project administration, R.M.; software, M.V.; supervision, R.M.; validation, R.M., S.M. and M.V.; visualization, M.V.; writing—original draft, R.M. and S.M.; writing—review and editing, R.M., M.V. and J.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research project no.: 02-019-K-0044 was funded by the European Union Funds for the period 2021–2027 under the Measure No. 05-001-01-05-07 "Establishing a coherent system for the promotion of innovative activities" under the activity "Stimulating the supply of innovations" under the action "Investing in activities for the development of new high value added products and enabling researchers to participate in R&D activities of enterprises, promotion of intellectual property, early pilot production of new products being developed and preparation for the market" (region of Central and Western Lithuania)"

Data Availability Statement: All data are freely available at Geoportal [52].

Acknowledgments: We express our gratitude to Dominykas Petkevičius and Arnas Ivanavičius for providing the software support, code building, and computational resources required to conduct the research.

Conflicts of Interest: Author Mantas Vaškevičius and Julius Gelšvartas were employed by the company Matomai UAB. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A

Figure A1 illustrates objects from each category analyzed (left real-life photo, right - reconstruction). From top to bottom: 1. Simple bench (6 m); 2. Bench (6 m), footage briefly pans away from the subject; 3. Bench (6 m), footage includes close-ups of the bench and the bottom; 4. Smaller bench (2.54 m); 5. Two smaller benches (2.54 m each); 6. Large square bench (2.54 m × 2.54 m); 7. Sculpture.

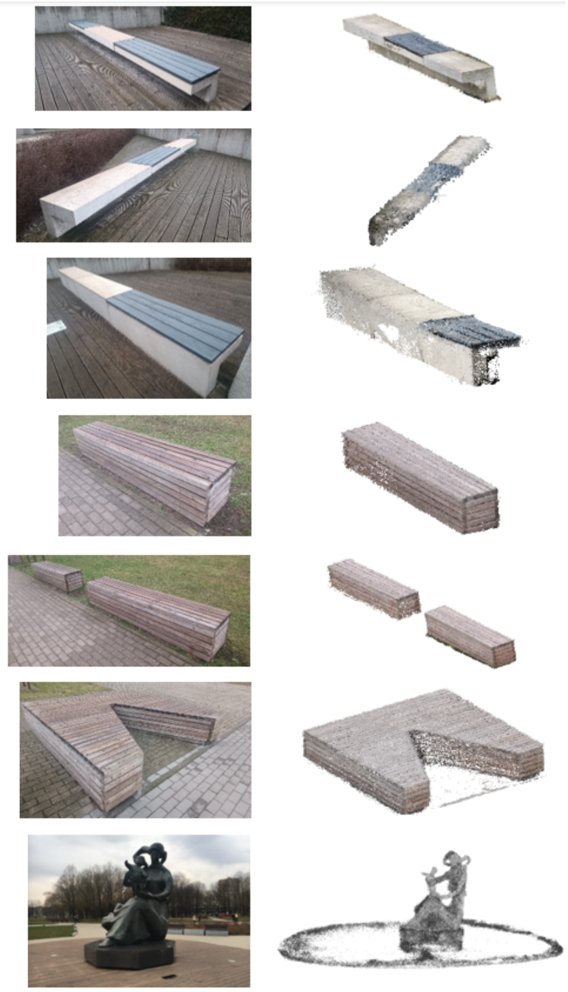

Figure A1. Example of objects from each category analyzed.

References

1. Yang, B. Developing a mobile mapping system for 3D GIS and smart city planning. *Sustainability* **2019**, *11*, 3713. [CrossRef]
2. Hu, X.; Assaad, R.H. A BIM-enabled digital twin framework for real-time indoor environment monitoring and visualization by integrating autonomous robotics, LiDAR-based 3D mobile mapping, IoT sensing, and indoor positioning technologies. *J. Build. Eng.* **2024**, *86*, 108901. [CrossRef]
3. Patoliya, J.; Mewada, H.; Hassaballah, M.; Khan, M.A.; Kadry, S. A robust autonomous navigation and mapping system based on GPS and LiDAR data for unconstraint environment. *Earth Sci. Inform.* **2022**, *15*, 2703–2715. [CrossRef]
4. Tang, F.; Wu, Y.; Hou, X.; Ling, H. 3D mapping and 6D pose computation for real time augmented reality on cylindrical objects. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 2887–2899. [CrossRef]
5. Schenk, T. Introduction to photogrammetry. *Ohio State Univ. Columb.* **2005**, *106*, 1.
6. Wandinger, U. Introduction to lidar. In *Lidar: Range-Resolved Optical Remote Sensing of the Atmosphere*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 1–18.
7. Bello, S.A.; Yu, S.; Wang, C.; Adam, J.M.; Li, J. Deep learning on 3D point clouds. *Remote Sens.* **2020**, *12*, 1729. [CrossRef]
8. Wang, R.; Peethambaran, J.; Chen, D. Lidar point clouds to 3-D urban models: A review. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 606–627. [CrossRef]
9. Royo, S.; Ballesta-Garcia, M. An overview of lidar imaging systems for autonomous vehicles. *Appl. Sci.* **2019**, *9*, 4093. [CrossRef]
10. Bartmiński, P.; Siłuch, M.; Kociuba, W. The effectiveness of a UAV-based LIDAR survey to develop digital terrain models and topographic texture analyses. *Sensors* **2023**, *23*, 6415. [CrossRef]
11. Behzadan, A.H.; Dong, S.; Kamat, V.R. Augmented reality visualization: A review of civil infrastructure system applications. *Adv. Eng. Inform.* **2015**, *29*, 252–267. [CrossRef]
12. Herrero, M.J.; Pérez-Fortes, A.P.; Escavy, J.I.; Insua-Arévalo, J.M.; De la Horra, R.; López-Acevedo, F.; Trigos, L. 3D model generated from UAV photogrammetry and semi-automated rock mass characterization. *Comput. Geosci.* **2022**, *163*, 105121. [CrossRef]
13. Portalés, C.; Lerma, J.L.; Pérez, C. Photogrammetry and augmented reality for cultural heritage applications. *Photogramm. Rec.* **2009**, *24*, 316–331. [CrossRef]
14. Burdziakowski, P.; Bobkowska, K. UAV photogrammetry under poor lighting conditions—Accuracy considerations. *Sensors* **2021**, *21*, 3531. [CrossRef] [PubMed]
15. Roncella, R.; Bruno, N.; Diotri, F.; Thoeni, K.; Giacomini, A. Photogrammetric digital surface model reconstruction in extreme low-light environments. *Remote Sens.* **2021**, *13*, 1261. [CrossRef]
16. Ghamisi, P.; Rasti, B.; Yokoya, N.; Wang, Q.; Hofle, B.; Bruzzone, L.; Bovolo, F.; Chi, M.; Anders, K.; Gloaguen, R.; et al. Multisource and multitemporal data fusion in remote sensing: A comprehensive review of the state of the art. *IEEE Geosci. Remote Sens. Mag.* **2019**, *7*, 6–39. [CrossRef]
17. Mazurek, P.; Hachaj, T. SLAM-OR: Simultaneous localization, mapping and object recognition using video sensors data in open environments from the sparse points cloud. *Sensors* **2021**, *21*, 4734. [CrossRef]
18. Shi, X.; Liu, T.; Han, X. Improved Iterative Closest Point (ICP) 3D point cloud registration algorithm based on point cloud filtering and adaptive fireworks for coarse registration. *Int. J. Remote Sens.* **2020**, *41*, 3197–3220. [CrossRef]
19. Masood, M.K.; Aikala, A.; Seppänen, O.; Singh, V. Multi-building extraction and alignment for as-built point clouds: A case study with crane cameras. *Front. Built Environ.* **2020**, *6*, 581295. [CrossRef]
20. Jiang, S.; Jiang, C.; Jiang, W. Efficient structure from motion for large-scale UAV images: A review and a comparison of SfM tools. *ISPRS J. Photogramm. Remote Sens.* **2020**, *167*, 230–251. [CrossRef]
21. Mildenhall, B.; Srinivasan, P.P.; Tancik, M.; Barron, J.T.; Ramamoorthi, R.; Ng, R. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* **2021**, *65*, 99–106. [CrossRef]
22. Wu, T.; Yuan, Y.J.; Zhang, L.X.; Yang, J.; Cao, Y.P.; Yan, L.Q.; Gao, L. Recent advances in 3d gaussian splatting. *Comput. Vis. Media* **2024**, *10*, 613–642. [CrossRef]
23. Myronenko, A.; Song, X. Point set registration: Coherent point drift. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 2262–2275. [CrossRef] [PubMed]
24. Zhang, F.; Zhang, L.; He, T.; Sun, Y.; Zhao, S.; Zhang, Y.; Zhao, X.; Zhao, W. An overlap estimation guided feature metric approach for real point cloud registration. *Comput. Graph.* **2024**, *119*, 103883. [CrossRef]
25. Gergelova, M.B.; Labant, S.; Kuzevic, S.; Kuzevicova, Z.; Pavolova, H. Identification of roof surfaces from LiDAR cloud points by GIS tools: a case study of Lučenec, Slovakia. *Sustainability* **2020**, *12*, 6847. [CrossRef]
26. Li, Y.; Ma, L.; Zhong, Z.; Liu, F.; Chapman, M.A.; Cao, D.; Li, J. Deep learning for lidar point clouds in autonomous driving: A review. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 3412–3432. [CrossRef]
27. Kalacska, M.; Arroyo-Mora, J.P.; Lucanus, O. Comparing UAS LiDAR and Structure-from-Motion Photogrammetry for peatland mapping and virtual reality (VR) visualization. *Drones* **2021**, *5*, 36. [CrossRef]

28. Marques, L.F.; Tenedório, J.A.; Burns, M.; Romão, T.; Birra, F.; Marques, J.; Pires, A. Cultural Heritage 3D Modelling and visualisation within an Augmented Reality Environment, based on Geographic Information Technologies and mobile platforms. *ACE Arquit. Ciudad. Entorno* **2017**, *11*, 117–136. [CrossRef]
29. Bailey, T.; Durrant-Whyte, H. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robot. Autom. Mag.* **2006**, *13*, 108–117. [CrossRef]
30. Gupta, A.; Fernando, X. Simultaneous Localization and Mapping (SLAM) and Data Fusion in Unmanned Aerial Vehicles: Recent Advances and Challenges. *Drones* **2022**, *6*, 85. [CrossRef]
31. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
32. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Adv. Neural Inf. Process. Syst.* **2017**, *30* .
33. Wang, L.; Chen, T.; Anklam, C.; Goldluecke, B. High dimensional frustum pointnet for 3d object detection from camera, lidar, and radar. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1621–1628.
34. Zhou, W.; Jiang, X.; Liu, Y.H. MVPointNet: Multi-view network for 3D object based on point cloud. *IEEE Sens. J.* **2019**, *19*, 12145–12152. [CrossRef]
35. Xu, D.; Anguelov, D.; Jain, A. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 244–253.
36. Boulch, A. ConvPoint: Continuous convolutions for point cloud processing. *Comput. Graph.* **2020**, *88*, 24–34. [CrossRef]
37. Qi, C.R.; Litany, O.; He, K.; Guibas, L.J. Deep hough voting for 3d object detection in point clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9277–9286.
38. Zhang, J.; Jiang, H.; Shao, H.; Song, Q.; Wang, X.; Zong, D. Semantic segmentation of in-vehicle point cloud with improved RANGENET++ loss function. *IEEE Access* **2023**, *11*, 8569–8580. [CrossRef]
39. Luo, X.; Xie, Y.; Zhang, Y.; Qu, Y.; Li, C.; Fu, Y. Latticenet: Towards lightweight image super-resolution with lattice block. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XXII 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 272–289.
40. de Gélis, I.; Lefèvre, S.; Corpetti, T. 3d urban change detection with point cloud siamese networks. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2021**, *43*, 879–886. [CrossRef]
41. Su, H.; Jampani, V.; Sun, D.; Maji, S.; Kalogerakis, E.; Yang, M.H.; Kautz, J. Splatnet: Sparse lattice networks for point cloud processing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2530–2539.
42. Li, Y.; Li, X.; Zhang, Z.; Shuang, F.; Lin, Q.; Jiang, J. DenseKPNET: Dense kernel point convolutional neural networks for point cloud semantic segmentation. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5702913. [CrossRef]
43. Lang, A.H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; Beijbom, O. Pointpillars: Fast encoders for object detection from point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 12697–12705.
44. Zhang, F.; Fang, J.; Wah, B.; Torr, P. Deep fusionnet for point cloud semantic segmentation. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XXIV 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 644–663.
45. Mandikal, P.; Navaneet, K.; Agarwal, M.; Babu, R.V. 3D-LMNet: Latent embedding matching for accurate and diverse 3D point cloud reconstruction from a single image. *arXiv* **2018**, arXiv:1807.07796.
46. Wang, C.; Xu, D.; Zhu, Y.; Martín-Martín, R.; Lu, C.; Fei-Fei, L.; Savarese, S. Densefusion: 6d object pose estimation by iterative dense fusion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3343–3352.
47. Zhou, Y.; Tuzel, O. Voxelnet: End-to-end learning for point cloud based 3d object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4490–4499.
48. Sitzmann, V.; Thies, J.; Heide, F.; Nießner, M.; Wetzstein, G.; Zollhofer, M. Deepvoxels: Learning persistent 3d feature embeddings. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2437–2446.
49. Shi, W.; Rajkumar, R. Point-gnn: Graph neural network for 3d object detection in a point cloud. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1711–1719.
50. Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; Li, H. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10529–10538.

51. Li, Z.; Wang, F.; Wang, N. Lidar r-cnn: An efficient and universal 3d object detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 7546–7555.
52. Geoportal.lt. Geographic data of the Lithuanian Republic. Available online: https://www.geoportal.lt/geoportal/ (accessed on 26 November 2024).
53. Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; Xiao, J. 3d shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1912–1920.
54. Yuan, W.; Eckart, B.; Kim, K.; Jampani, V.; Fox, D.; Kautz, J. Deepgmr: Learning latent gaussian mixture models for registration. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part V 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 733–750.
55. Cheng, J.; Su, H.; Korhonen, J. No-Reference Point Cloud Quality Assessment via Weighted Patch Quality Prediction. *arXiv* **2023**, arXiv:2305.07829.
56. Yang, Q.; Liu, Y.; Chen, S.; Xu, Y.; Sun, J. No-reference point cloud quality assessment via domain adaptation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 21179–21188.
57. Ahmadyan, A.; Zhang, L.; Ablavatski, A.; Wei, J.; Grundmann, M. Objectron: A Large Scale Dataset of Object-Centric Videos in the Wild with Pose Annotations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021.
58. Zhang, Y.; Yang, Q.; Xu, Y.; Liu, S. Perception-guided quality metric of 3D point clouds using hybrid strategy. *IEEE Trans. Image Process.* **2024**, *33*, 5755–5770. [CrossRef]
59. Huang, Z.; Wen, Y.; Wang, Z.; Ren, J.; Jia, K. Surface reconstruction from point clouds: A survey and a benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 9727–9748. [CrossRef]
60. Abbasi, R.; Bashir, A.K.; Alyamani, H.J.; Amin, F.; Doh, J.; Chen, J. Lidar point cloud compression, processing and learning for autonomous driving. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 962–979. [CrossRef]
61. Xu, N.; Qin, R.; Song, S. Point cloud registration for LiDAR and photogrammetric data: A critical synthesis and performance analysis on classic and deep learning algorithms. *ISPRS Open J. Photogramm. Remote Sens.* **2023**, *8*, 100032. [CrossRef]

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