ISPRS Journal of Photogrammetry and Remote Sensing 220 (2025) 799–823

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Nothing Stands Still: A spatiotemporal benchmark on 3D point cloud registration under large geometric and temporal change

Tao Sun [b], Yan Hao [a], Shengyu Huang [a], Silvio Savarese [b], Konrad Schindler [a], Marc Pollefeys [a,c], Iro Armeni [b,*]

[a] ETH Zurich, Switzerland
[b] Stanford University, United States of America
[c] Microsoft Mixed Reality & AI Lab, Zurich, Switzerland

ARTICLE INFO

MSC:
00-01
99-00

Keywords:
Point cloud
Spatiotemporal registration
Pairwise registration
Multiway registration
Construction

ABSTRACT

Building 3D geometric maps of man-made spaces is a well-established and active field that is fundamental to numerous computer vision and robotics applications. However, considering the continuously evolving nature of built environments, it is essential to question the capabilities of current mapping efforts in handling temporal changes. In addition to the above, the ability to create spatiotemporal maps holds significant potential for achieving sustainability and circularity goals. Existing mapping approaches focus on small changes, such as object relocation within common living spaces or self-driving car operation in outdoor spaces; all cases where the main structure of the scene remains fixed. Consequently, these approaches fail to address more radical change in the structure of the built environment, such as on the geometry and topology of it. To promote advancements on this front, we introduce the **Nothing Stands Still (NSS)** benchmark, which focuses on the spatiotemporal registration of 3D scenes undergoing large spatial and temporal change, ultimately creating one coherent spatiotemporal map. Specifically, the benchmark involves registering within the same coordinate system two or more partial 3D point clouds (fragments) originating from the same scene but captured from different spatiotemporal views. In addition to the standard task of *pairwise* registration, we assess *multi-way* registration of multiple fragments that belong to the same indoor environment and any temporal stage. As part of **NSS**, we introduce a dataset of 3D point clouds recurrently captured in large-scale building indoor environments that are under construction or renovation. The **NSS** benchmark presents three scenarios of increasing difficulty, with the goal to quantify the generalization ability of point cloud registration methods over space (within one building and across buildings) and time. We conduct extensive evaluations of state-of-the-art methods on **NSS** over all tasks and scenarios. The results demonstrate the necessity for novel methods specifically designed to handle large spatiotemporal changes. The homepage of our benchmark is at http://nothing-stands-still.com.

1. Introduction

"Everything flows, nothing stands still" – as Heraclitus[1] advocated, a critical property of the world around us is that it changes over time. The temporal dimension and its impact on the built environment have not been ignored by the field of computer vision, photogrammetry, and robotics. Its study appears in different tasks, such as those related to video understanding (Simonyan and Zisserman, 2014; Purushwalkam et al., 2020; Haresh et al., 2021; Kwon et al., 2022; Deng et al., 2024), self-driving cars (Qi et al., 2021; Huang et al., 2022b; Zhang et al., 2023), change detection in 2D images acquired at scene-level (Sakurada et al., 2020; Lei et al., 2020; Ru et al., 2020; Prabhakar et al., 2020; Wang et al., 2021b; Li et al., 2022) or in satellite imagery (Bourdis et al., 2011; Peng et al., 2019; Chen et al., 2020; Zheng et al., 2021; Cheng et al., 2024), change detection in 3D scans (Xiao et al., 2015; Qin et al., 2016; Kharroubi et al., 2022; Gehrung et al., 2022; Huang et al., 2022c; de Gélis et al., 2023; Stilla and Xu, 2023; López-Armenta and Nespeca, 2024), object relocalization in recaptured 3D indoor scenes (Wald et al., 2019, 2020; Halber et al., 2019; Zhu et al., 2024), and robot navigation in dynamic environments (Droeschel et al., 2017; Wang et al., 2020a, 2021a). However, the examined change, especially

* Corresponding author.
E-mail address: iarmeni@stanford.edu (I. Armeni).
[1] An ancient Greek philosopher, 501 B.C.

https://doi.org/10.1016/j.isprsjprs.2025.01.010
Received 2 July 2024; Received in revised form 23 November 2024; Accepted 7 January 2025
Available online 31 January 2025

in the 3D indoor domain, often focuses on small spatial (e.g., that of a room Straub et al., 2019; Halber et al., 2019; Wald et al., 2019, 2020) and temporal (e.g., that of a few minutes Droeschel et al., 2017; Prabhakar et al., 2020; Wang et al., 2021b; Huang et al., 2022b; Kwon et al., 2022; Deng et al., 2024) scales, and mainly limited to object relocation (movement of objects) (Straub et al., 2019; Halber et al., 2019; Wald et al., 2019, 2020; Li et al., 2022).

The built environment undergoes various changes throughout its lifecycle, starting from construction, through operation, and finally reaching the end-of-life phase. These changes go beyond simple relocation and involve differences in geometry, appearance, and topology of the building elements. Examples of such changes include the installation of pipes on ceilings, the transformation of floors before and after carpeting, and the gradual development of walls from a group of studs to their final structure visible to users. Among the different lifecycle phases, the most significant differences can be observed during construction and before/after renovation.

Understanding and addressing these dynamic changes opens up new research directions, shifting the predominant static perspective of scene understanding. An instance of these directions is evident in robotics. Robots are frequently required to localize themselves within pre-mapped 3D structures that may have experienced alterations over time. Proficiently identifying and adapting to these changes improves not only the robot's navigation but also its interaction with the environment (Tsamis et al., 2021; Stefanini et al., 2023). Moreover, acquiring a spatiotemporal understanding of how buildings evolve over time is crucial for achieving sustainability and circular economy goals in the built environment (Munaro et al., 2020). For instance, it enables quantitative monitoring and quality control of construction progress, leading to a reduction in out-of-estimate construction costs associated with rework. Currently, progress monitoring is often assessed in a rough manner by project managers, with rework accounting for 52% of the total out-of-estimate costs (Love et al., 2002). Furthermore, a spatiotemporal understanding of building changes can play a significant role in establishing workflows for material reuse. It is estimated that 95% of non-hazardous construction and demolition waste is reusable or recyclable (Ma et al., 2020). However, a large amount of this material ends up in landfills due to a lack of information about materials within buildings. This is due to raw materials getting hidden behind surfaces or paint as construction progresses without proper documentation.

To this end, we propose **Nothing Stands Still (NSS)**, a novel spatiotemporal benchmark utilizing 3D point cloud captures of indoor environments in the aforementioned lifecycle phases. These captures encompass a large spatial and temporal scale and contain changes that extend beyond object relocation. As part of the benchmark, we introduce a spatiotemporal point cloud dataset comprising 6 large-scale building areas (i.e., , distinct buildings or large sections of them referred to as *areas*) in multiple temporal stages (referred to as *stages*) spanning several months (Section 4). We focus on the problem of spatiotemporal point cloud registration and design a series of experiments to demonstrate the inherent challenges of this setup and highlight the limitations of existing methods in addressing them. Notably, spaces under construction are commonly of low-texture and highly repetitive geometry, posing challenges for local feature-based algorithms in computer vision. These algorithms struggle in the spatiotemporal registration task, where similar local features may not correspond to the same location over time.

To evaluate the generalization ability of methods across space and time, we define three scenarios that involve different spatial and temporal data splits (Section 5). Unlike typical spatial registration setups, our training and testing process includes not only point cloud pairs from the same *stage* and *area*, but also pairs from different *stages* and *areas*. When referring to 'pairs from different areas' in a data split, it means point cloud pairs originating from distinct buildings. For instance, one point cloud pair may come from *area* A, while another may come from *area* B. Within a pair, the two point clouds can represent either the same or different *stages* of their respective *area* of origin. Given the large-scale nature of each *area*, the input pairs represent partial observations, namely *fragments*, of the complete area. Consequently, the pairwise registration task is constrained to achieving local alignment between the input pair. To achieve global alignment in the context of entire *areas*, we incorporate the task of multi-way registration that considers all input *fragments* belonging to the same *area* at any *stage*.

We evaluate several state-of-the-art algorithms (Rusu et al., 2009; Bai et al., 2020; Choy et al., 2019b; Huang et al., 2021; Qin et al., 2022) on the **NSS** benchmark. We also evaluate these algorithms on a state-of-the-art spatiotemporal 3D point cloud dataset (Wald et al., 2020), which captures changes in inhabited indoor scenes related to furniture addition, removal, or relocation within rooms. This comparative analysis showcases the need for more challenging setups when addressing the problem of understanding and operating in dynamic environments.

The contributions of this paper can be summarized as:

- We introduce a new spatiotemporal dataset that captures large spatial and temporal changes in the geometry, appearance, and topology of building elements. The dataset comprises 6 indoor areas undergoing construction and renovation with recurrent captures spaced months apart (2–6 per area).
- We propose a novel benchmark, **NSS**, for spatiotemporal 3D point cloud registration, which includes both pairwise and multi-way registration. The evaluation employs diverse data splits, where training and testing pairs originate from across areas *and* stages.
- We provide extensive experimental analysis and insights into the performance of state-of-the-art registration algorithms on the **NSS** benchmark. We also provide evaluation results on a state-of-the-art spatiotemporal 3D point cloud dataset (Wald et al., 2020), following the same evaluation protocol.

We also provide the community with a server for evaluating their algorithms on the test sets, which we keep hidden. A leaderboard showcases the latest results and progress on the benchmark. For more details, please visit nothing-stands-still.com.

2. Related work

We first review works on spatiotemporal reasoning from visual data and the employed datasets before proceeding to pairwise and multi-way registration methods. Finally, we briefly discuss synthetic point cloud generation since our point cloud registration benchmark was created from 3D mesh data of real-world captures.

2.1. Spatiotemporal reasoning

Spatiotemporal reasoning from visual data is a fundamental problem in computer vision, photogrammetry, and robotics, and can be examined at various levels of detail. Change detection methods categorize scenes into stationary and changed regions, through 2D pixel (Hussain et al., 2013), 3D point (Yew and Lee, 2021a; de Gélis et al., 2023), or 3D voxel (Pollard and Mundy, 2007; Xiao et al., 2015) classification tasks. Motion segmentation approaches (Chen et al., 2021; Huang et al., 2022b) segment scenes into static and dynamic parts. Optical flow estimation techniques (Horn and Schunck, 1981; Black and Anandan, 1993; Brox et al., 2009) model fine-grained motion information by associating pixels across frames, typically formulated as optimization tasks (Black and Anandan, 1993). Modern learning-based methods (Ilg et al., 2017; Hui et al., 2018; Fischer et al., 2015; Teed and Deng, 2020) directly learn flow prediction with enhanced accuracy and efficiency from large datasets. Scene flow estimation (Vedula et al., 1999; Sun et al., 2010) additionally provides depth information of objects in a 3D scene. Traditional methods (Vogel et al., 2011, 2013, 2015) leverage motion smoothness priors within optimization frameworks, while learning-based methods (Liu et al., 2019a; Puy et al., 2020) learn directly from large-scale datasets.

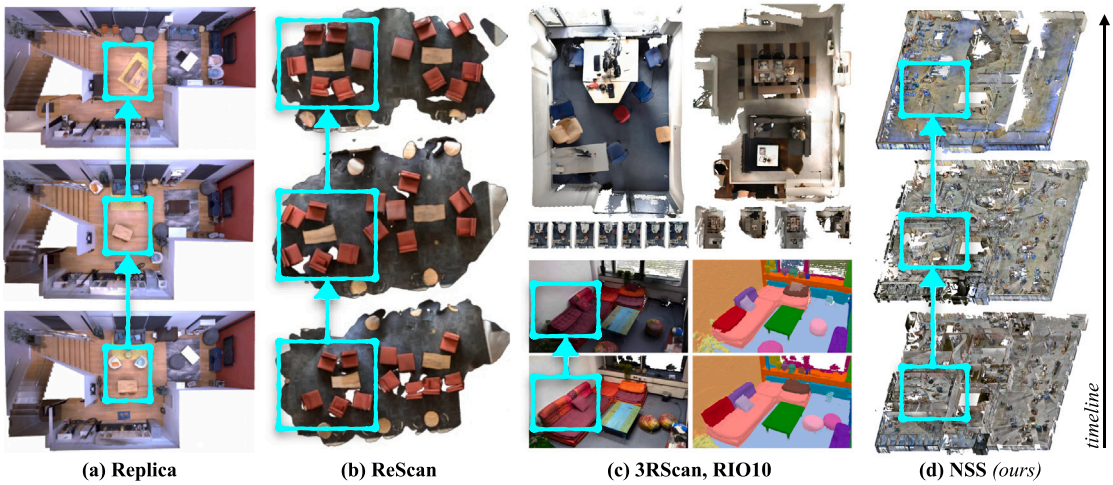


Fig. 1. Qualitative examples of existing indoor spatiotemporal datasets and **Nothing Stands Still** (**NSS**). As shown, existing datasets focus on small and daily changes in living environments, whereas **NSS** exhibits drastic changes over time. Examples of such changes over different *stages* are highlighted with a *cyan box*.

Spatiotemporal reasoning encompasses various downstream tasks, including 3D change detection (Kharroubi et al., 2022; Stilla and Xu, 2023), action recognition from videos (Simonyan and Zisserman, 2014; Wang et al., 2019), multi-object visual tracking (Milan et al., 2016; Xiang et al., 2015), dynamic reconstruction (Rempe et al., 2020; Huang et al., 2022a), and novel view synthesis (Pumarola et al., 2021; Martin-Brualla et al., 2021). Most tasks address spatiotemporal changes at video frame rate. Going beyond the temporal change induced in milliseconds, D^4R (Golparvar-Fard et al., 2009) focuses on aligning images captured in construction sites on a daily basis for the purpose of construction progress monitoring. In a non-built setting, Dong et al. (2017) perform 4D reconstruction of agricultural crops for monitoring growth. Furthermore, works like Griffith et al. (2020) and Schindler and Dellaert (2010), Schindler et al. (2007) aim to sequence online image collections spanning a year or decades based on visibility and temporal occupancy. Scene chronology (Matzen and Snavely, 2014) scales this sequencing operation to millions of photos and reasons about finer appearance changes due to denser sampling. While Matzen and Snavely (2014) is limited to rendering planar regions, it was improved upon by Martin-Brualla et al. (2015), which represents scene geometry using time-varying depth maps, enabling the generation of high-quality time-lapse videos. Recently, Matzen and Snavely (2014) has been extended by employing neural fields (Lin et al., 2023) to achieve photo-realistic renderings with higher fidelity.

Few works specifically handle point cloud sequences as inputs. Spatiotemporal reasoning from point cloud data is predominantly achieved by propagating temporal information via flow estimations (Liu et al., 2019b; Choy et al., 2019a; Fan et al., 2020). "Objects can move" (Adam et al., 2022) has similarities to our benchmark and addresses indoor 3D change detection via geometric transformation consistency. However, it is limited to object relocation and cannot handle changes in the structure of the scene.

2.2. Spatiotemporal 3D datasets

In recent years, several 3D datasets (Armeni et al., 2016, 2017; Chang et al., 2017; Hua et al., 2016; Dai et al., 2017; Straub et al., 2019; Wald et al., 2019; Halber et al., 2019; Wald et al., 2020) have emerged for indoor scene understanding, with some of them also considering the temporal aspect (Straub et al., 2019; Wald et al., 2019; Halber et al., 2019; Wald et al., 2020). However, not all of these datasets contain real-world scenes (Park et al., 2021). In Qiu et al. (2020), the authors generate a change dataset by leveraging an existing real-world static 3D dataset (Chang et al., 2017). In order to generate change, they add synthetic models of small objects in the scenes (e.g., a cup or a car toy). There are three main datasets capturing real-world change in inhabited indoor spaces, namely Replica (Straub et al., 2019), ReScan (Halber et al., 2019), and 3RScan (Wald et al., 2019), that focus on the relocation, addition, or removal of furniture. RIO10 (Wald et al., 2020) is a smaller version of 3RScan. These datasets capture aspects of daily human interaction with the built environment and are limited in spatial scale compared to **NSS** (one room versus one building floor with multiple rooms). For more details see Table 1 and for visual samples Fig. 1.

LAMAR (Sarlin et al., 2022) is similar to our benchmark. It is a large-scale dataset captured in diverse environments over an extended temporal horizon. However, it focuses on the task of visual localization from images and radios, while **NSS** concentrates on registration using point clouds. Additionally, the scenes captured in LAMAR do not exhibit significant changes in the environment's geometry and are more related to relocation scenarios. Other datasets focus on outdoor scenes and capture seasonal changes in real-world data (Wenzel et al., 2020) or are tailored to self-driving cars (Ros et al., 2016; Hernandez-Juarez et al., 2017; Zolfaghari Bengar et al., 2019; Geiger et al., 2013; Cordts et al., 2016). However, their review is outside the scope of this paper.

2.3. 3D point cloud registration

The field of 3D point cloud registration is well-established and active. Here, we discuss both pairwise registration and multi-way registration methods.

Pairwise registration. Approaches here can be mainly grouped as *feature-based* and *end-to-end* registration.

(a) *Feature-based methods* typically involve two steps: local feature extraction and pose estimation. The pose estimation step uses either a robust estimator such as RANSAC (Fischler and Bolles, 1981) or globally optimal estimators (Li and Hartley, 2007; Hartley and Kahl, 2009; Cai et al., 2019). For local feature extraction, traditional methods use hand-crafted features (Johnson and Hebert, 1999; Rusu et al., 2008, 2009; Tombari et al., 2010b,a; Theiler et al., 2014) to capture local geometry and, while having good generalization abilities across scenes, they often lack robustness against occlusions. In contrast, learned local features have taken over in the past few years, and, instead of using heuristics, they rely on deep models and metric learning (Hermans et al., 2017; Sun et al., 2020) to extract dataset-specific discriminative local descriptors. Depending on the input to models, these learned descriptors can be divided into patch-based and fully convolutional methods. Patch-based methods (Gojcic et al., 2019; Ao et al.,

Table 1
Comparison of existing indoor spatiotemporal 3D point cloud datasets. **NSS** focuses on scenes that demonstrate large changes. Since the scale is on the building level, the number of areas (and temporal stages) is less than that of the other datasets but a single area contains numerous scenes on the scale of them.

Dataset	Area		Temporal stage		Change scale
	Num	Scale (*Type*)	Total	Per scene	
Replica (Straub et al., 2019)	1	Room *(typical living)*	6	6	Small
ReScan (Halber et al., 2019)	13	Room *(typical living)*	45	3–5	Small
3RScan (Wald et al., 2019)	478	Room *(typical living)*	1482	2–12	Small
RIO10 (Wald et al., 2020)	10	Room *(typical living)*	74	5–12	Small
NSS *(ours)*	6	Building *(construction)*	27	2–6	Large

2021) treat each point independently, while fully convolutional methods (Choy et al., 2019b; Bai et al., 2020) can extract all local descriptors for the whole scene in a single forward pass. PREDATOR (Huang et al., 2021) is the first work that pays special attention to low-overlap pairs and proposes an overlap-attention module to robustify registration by learning to sample interest points in the overlap region only. Qin et al. (2022) and Yu et al. (2021) improved PREDATOR by operating in a coarse-to-fine manner. DPFM (Attaiki et al., 2021) adopts this idea to non-rigid registration via an overlap attention mechanism in function space.

(b) *End-to-end methods* integrate differentiable pose estimators into the feature extraction pipeline (Wang and Solomon, 2019a,b; Yew and Lee, 2020; Aoki et al., 2019; Wei et al., 2023), providing an alternative to the feature-based methods. With the weighted Kabsch solver (Arun et al., 1987) or the generalized differentiable RANSAC (Wei et al., 2023), training can be directly supervised by ground truth poses. However, they mostly work on synthetic datasets (Wu et al., 2015) due to weak feature extractors.

In our experiments, we evaluate the performance on **NSS** of traditional (Rusu et al., 2008), fully convolutional (Bai et al., 2020; Choy et al., 2019b), attention-based (Huang et al., 2021), and coarse-to-fine (Qin et al., 2022) methods.

Multi-way registration This task aims to resolve the ambiguities in pairwise registration by leveraging multi-view constraints. Traditional methods (Huber and Hebert, 2003; Fantoni et al., 2012) simply refine the initial pose estimations by extending ICP to the multi-way setting. However, this approach becomes computationally intractable due to the quadratically increased number of pairwise registrations as the views increase. Modern methods (Torsello et al., 2011; Choi et al., 2015; Theiler et al., 2015; Bernard et al., 2015; Zhou et al., 2016; Bhattacharya and Govindu, 2019) optimize the initial relative poses by incorporating global cycle consistency. This optimization is typically achieved by applying *synchronization* techniques within the pose graphs using methods such as Iterative Reweighted Least Square (IRLS), Triggs' correction (Triggs et al., 2000), and other lifting-based approaches (Zach and Bourmaud, 2018). Recent studies have introduced learning-based methods to improve synchronization. For instance, Huang et al. (2019) and Gojcic et al. (2020) propose to learn the edge weights for the transformation synchronization, with Gojcic et al. (2020) additionally learning the pairwise registration (Choy et al., 2019b) and outlier rejection (Zhang et al., 2019). Wang et al. (2023) propose constructing a sparse but reliable pose graph by first estimating the pairwise overlap ratios. They further improve the robustness of outlier edges by incorporating a history reweighting function in the IRLS scheme.

In our experiments, we explore a widely-used pose graph synchronization method by Choi et al. (2015), which is known for its efficiency in avoiding local minima by leveraging noisy-free odometry poses for initializing the nodes in the pose graph. Additionally, we investigate a recent synchronization method by Yew and Lee (2021b), PoseGraphNet, that does not assume prior knowledge of pairwise registration. Instead, it utilizes a recurrent Graph Neural Network (GNN) to progressively refine poses, beginning with node initializations as identity matrices. Note that for both methods, the edges in the graph are initialized using the results from pairwise registration, a common ground for constructing the pose graph.

2.4. Synthetic generation of point clouds

Generating synthetic visual data is a largely explored field. Here, we limit the scope to generating 3D point cloud data and depth images from two perspectives: sensor pose definition and sensor noise simulation.

Sensor pose definition. Three main approaches are identified: (i) **Manual definition:** The user manually specifies the location of the sensor either by playing a video game (Richter et al., 2016; Shafaei et al., 2016; Hu et al., 2021) or via a graphical interface (Handa et al., 2015; Noichl et al., 2021); (ii) **Real-world trajectory inserted in simulation:** A trajectory captured in the real world is inserted and transformed to simulate a trajectory in a synthetic scene (Handa et al., 2012, 2014); and (iii) **Random sampling in the synthetic scene:** Here, sensor locations and poses are randomly sampled in the simulation environment to address certain criteria. Methods use physics-based simulation of sensor trajectories (McCormac et al., 2017; Roberto de Souza et al., 2017), react to dynamic movement of other objects in the scene (Ros et al., 2016; Hernandez-Juarez et al., 2017; Zolfaghari Bengar et al., 2019), or densely sample in the free 3D space (Kundu et al., 2020; Qiu and Yuille, 2016). Closer to the latter and to our approach is a group of methods that define sensor positions with the use of a 2D occupancy map of the scene (Song et al., 2017; Zhang et al., 2017; Wang et al., 2020b; Biswasa et al., 2015; Díaz-Vilariño et al., 2018; Frías et al., 2019). These methods commonly employ a set of constraints, criteria, and heuristics to exclude non-informative views from the final selection. Similar to them, we sample locations on a 2D occupancy grid of the scene and constrain the sensor location sampling based on the properties and way of use of a real-world sensor (i.e., height position and distance between locations). However, instead of setting heuristics to exclude non-informative views, we define a probability map that favors more realistic locations (e.g., further away from obstacles). In our final set of generated data, we include all possible scenarios from more to less informative.

Sensor noise simulation. A well-known property of simulation environments is the possible mismatch in the distribution of the noise characteristics in the data. In practice, multiple scans of the same fragment may exhibit different noises. When simulating point clouds from any underlying geometry (e.g., , reconstructed mesh, geometric primitives), to replicate this and prevent models from exploiting the consistent scanning artifacts during registration, works have been creating statistical noise models of sensors to utilize in simulation (Noichl et al., 2021; Handa et al., 2015, 2014; Barron and Malik, 2013). We follow the implementation in Handa et al. (2015) and Gschwandtner et al. (2011) to simulate the sensor noise during our synthetic data generation.

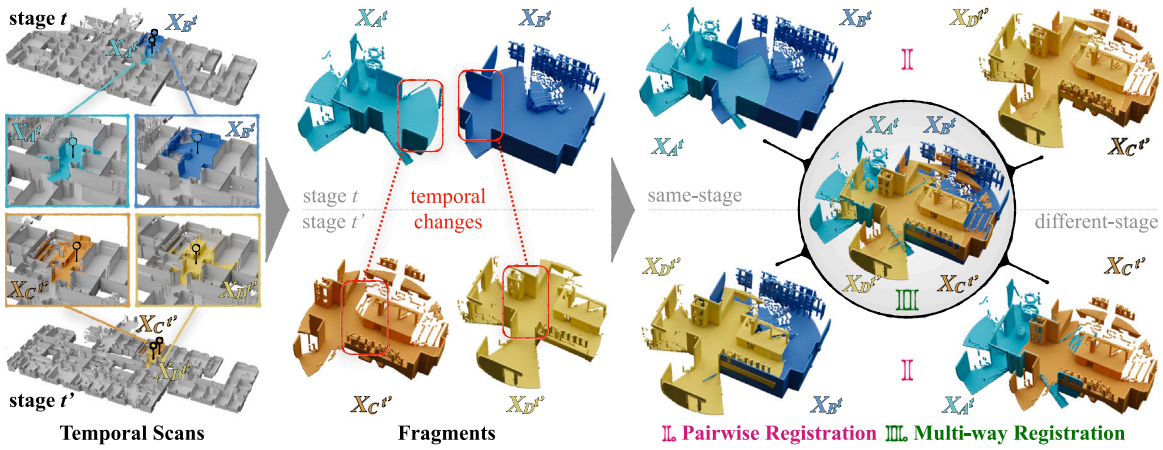


Fig. 2. Overview of the **Nothing Stands Still** (**NSS**) benchmark: fragments ($X_A{}^t$, $X_B{}^t$, $X_C{}^{t'}$, $X_D{}^{t'}$) captured in a construction site are spatiotemporally registered. First, a pairwise registration step registers individually the pairs of fragments belonging to the same (($X_A{}^t$, $X_B{}^t$) and ($X_C{}^{t'}$, $X_D{}^{t'}$)) or different stages (($X_A{}^t$, $X_C{}^{t'}$) and ($X_B{}^t$, $X_D{}^{t'}$)). Then, a multi-way registration step creates a single and coherent spatiotemporal map of all fragments. Given current methods, this step is initialized by the results of the pairwise one. In this example, we assume overlap occurs for pairs ($X_A{}^t$, $X_B{}^t$), ($X_B{}^t$, $X_D{}^{t'}$), ($X_C{}^{t'}$, $X_D{}^{t'}$), and ($X_C{}^{t'}$, $X_A{}^t$). We define the overlapping pairs for entire areas using spatiotemporal graphs, as detailed in Section 6.2.

3. Spatiotemporal point cloud registration

Before introducing the dataset and benchmark, we clarify all used terminology here: *area* refers to a building's (large) indoor space that was recurrently captured over time; *individual 3D point clouds* or else *fragments* refer to partial 3D observations of the area in the form of point clouds; *entire area point clouds* or else *scans* refer to the entire captured area reconstructed in the form of a 3D point cloud at one point in time; and *stages* denote discrete points in time when an area was captured.

Given multiple *fragments* of an *area* that are captured at different *stages* and 3D locations, the goal is to spatiotemporally align them and achieve a 3D *scan* of the *area* over time (i.e., a 4D scan). This includes two tasks[2] (Fig. 2): pairwise registration of *fragments* that can belong to the same or different *stages* (Fig. 2(I)), followed by multi-way registration of all *fragments* to result in the final spatiotemporal alignment of them (Fig. 2(II)).

Pairwise registration. Consider source fragment $\mathbf{X}^{(S,t)} = \{\mathbf{x}_i^{(S,t)} \in \mathbb{R}^3\}_{i=1}^{n}$ and target fragment $\mathbf{X}^{(T,t')} = \{\mathbf{x}_j^{(T,t')} \in \mathbb{R}^3\}_{j=1}^{m}$ captured at t and t', respectively. The spatiotemporal pairwise registration task is to recover a rigid transformation $\mathbf{M}^* = [\mathbf{R}^*, \mathbf{v}^*]$ where rotation matrix $\mathbf{R}^* \in \mathrm{SO}(3)$ and translation vector $\mathbf{v}^* \in \mathbb{R}^3$, such that:

$$\mathbf{M}^* = \arg\min_{\mathbf{M}} \sum_{i=1}^{n} \left\| \mathbf{M}(\mathbf{x}_i^{(S,t)}) - \mathrm{NN}(\mathbf{M}(\mathbf{x}_i^{(S,t)}), \mathbf{X}^{(T,t')}) \right\|_2 \quad (1)$$

where $\mathbf{M}(\mathbf{x}) := \mathbf{R}\mathbf{x} + \mathbf{v}$ is the rigid transformation applied on point $\mathbf{x}$ and $\mathrm{NN}(\mathbf{x}, \mathbf{X})$ represents the nearest neighbor of point $\mathbf{x}$ in point cloud $\mathbf{X}$ in Euclidean space.

Multi-way registration. Consider a set of fragments $\{X_i^t\}$ where each fragment could be captured at any stage t. The spatiotemporal multi-way registration task is to recover a set of rigid transformations $\{\mathbf{M}_i^t\}$ between each fragment in $\{X_i^t\} \backslash \{X_{i=1}^{t=1}\}$ and $X_{i=1}^{t=1}$, such that all fragments achieve a globally optimal alignment in the same reference system. Different from existing settings (Choi et al., 2015; Huang et al., 2019; Gojcic et al., 2020) that consider fragments from the same stage only, the proposed spatiotemporal multi-way registration contains fragments from different stages and is thus a more challenging optimization task. This step results in the spatiotemporal 3D reconstruction of the area.

[2] State-of-the-art multi-way registration algorithms depend on initialization of the alignment between fragments, which can be acquired from the pairwise registration task. In the future, methods can solve the two spatiotemporal tasks independently without jeopardizing the structure of the benchmark.

4. Nothing stands still dataset

The **Nothing Stands Still** (**NSS**) dataset consists of 3D fragments captured over time in 6 large-scale indoor areas, along with their corresponding scans. The dataset focuses on the construction of interior layouts, where the exterior shell of the areas has been erected, and the interior space is empty. The captures chronicle the progression of various construction activities, including the creation of walls, installation of mechanical, electrical, and plumbing elements, movement of materials, temporary structures, machinery, and more. Fig. 3 provides a snapshot of all 6 areas at a single stage. Five of the areas in the dataset depict stages under construction, while one area (Area D) includes only before-and-after renovation stages with no visible construction. Still, the renovation stages involve significant structural changes like wall removal, functional changes like transforming a conference room into an office, and furniture replacement like desks and carpeting.

It is important to note that fragments belonging to different areas but annotated as being in the same stage (e.g., stage t') do not depict identical changes, as construction progresses differently across areas. Similarly, construction progress may vary even across fragments within the same area and stage. Fig. 4 illustrates examples of areas and their stages. For all fragments in an area, the **NSS** dataset provides ground truth pose annotations that describe their spatial and temporal information.

4.1. Dataset acquisition

Each area in the **NSS** dataset covers on average 2500 m^2 and consists of 2–6 stages. The time intervals between stages can range from weeks to months, since the data collection is not based on a fixed schedule but rather follows the completion of significant construction tasks. The timing of data collection is determined in consultation with the project manager to ensure access to the construction site at appropriate and safe times. The objective is to capture the data just before crucial building information, such as pipes and structural elements, becomes inaccessible once covered by surfaces. Table 2 provides details on the floorplan coverage for each area in m^2.

The dataset was collected using the Matterport Camera v1 (Matterport, 2025). The Matterport Camera is a tripod-based reality capture system that acquires 360° fragments from static locations.

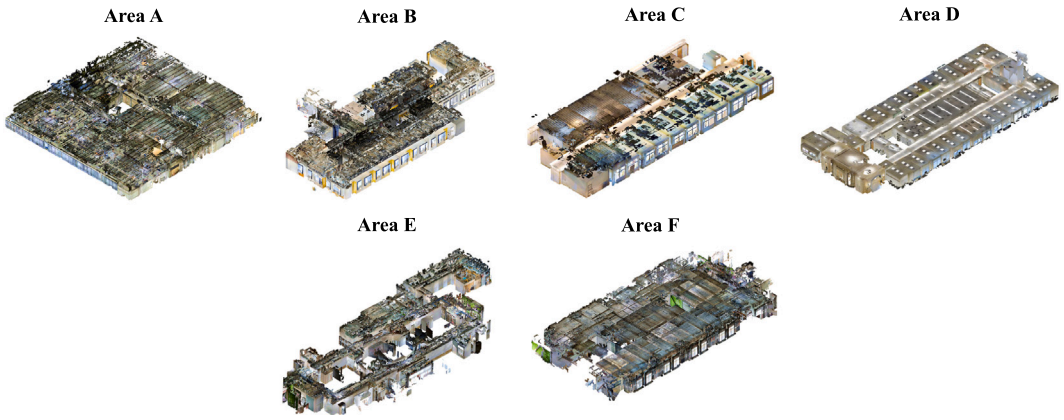


Fig. 3. Areas in the **Nothing Stands Still dataset** at first temporal stage. The building layout and size ranges across areas.

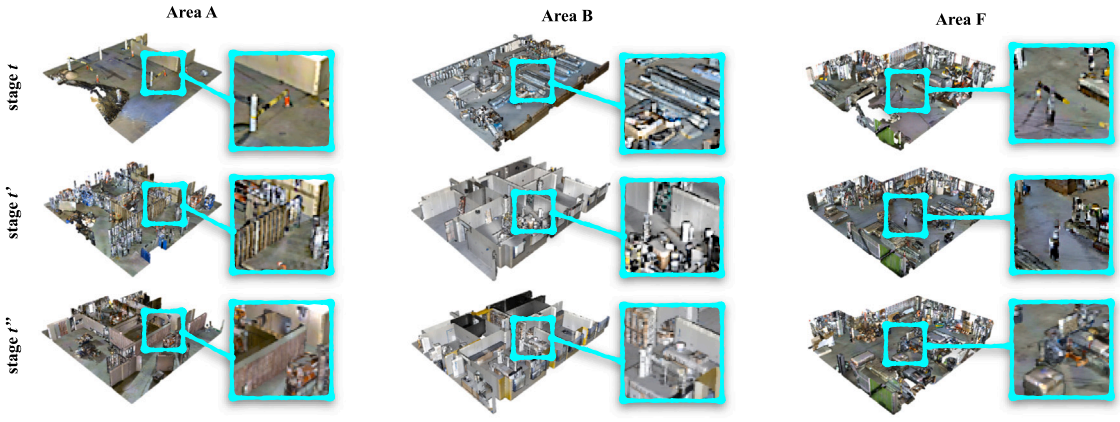


Fig. 4. Sample close-up snapshots of areas in the **Nothing Stands Still dataset.** Significant changes are occurring per area, starting from an empty scene and reaching the construction of rooms.

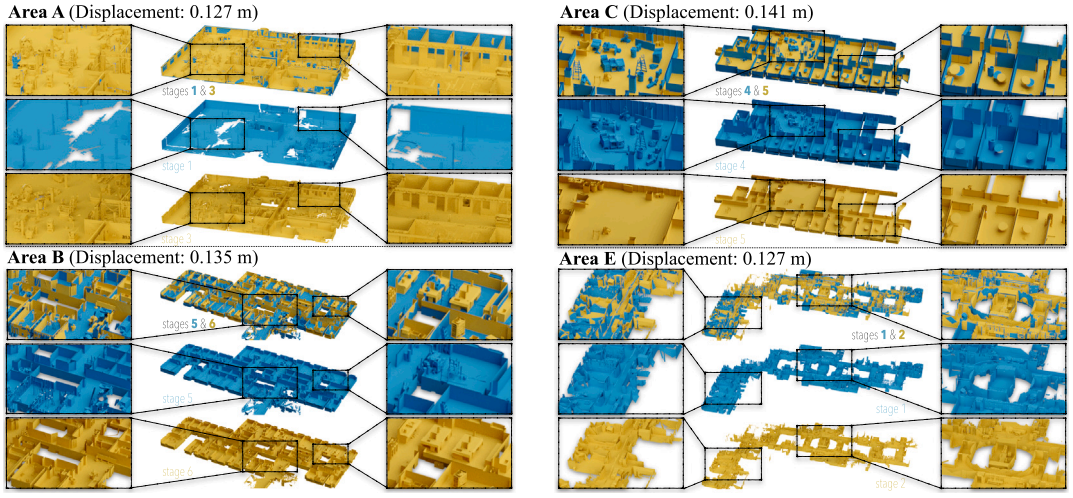
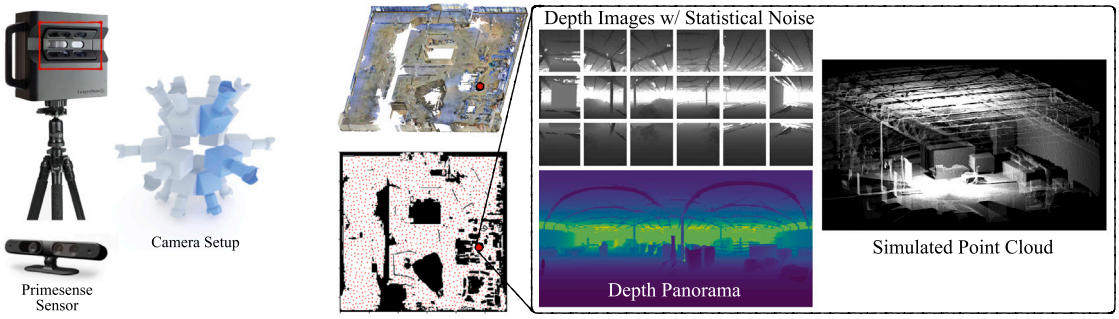


Fig. 5. Global registration ground truth, for example, scans in four areas in the **Nothing Stands Still** dataset. Details about the alignment method for the global ground truth are provided in Section 4.2.1.

These fragments are subsequently registered together to create the final 3D scan of the captured area using proprietary software. The proprietary software is not accessible by or disclosed to users. Matterport3D automatically performs the registration upon uploading the data and provides to the user the final result. According to specifications, Matterport3D has a geometry error of around 1 inch from reality. Hence, while users have access to the 3D scans, they do not have access to the individual fragments. Furthermore, the 3D scans from different stages depicting the same area are not spatiotemporally aligned in the same coordinate system because there is no geolocalization information available. Therefore, two important steps are undertaken in creating the dataset: (a) *alignment of 3D scans*: different-stage scans of the same area are aligned in the same coordinate system to acquire all ground truth poses; and (b) *fragment generation*: as the original fragments are



(a) Matterport Camera Setup (b) An example of generating synthetic fragments.

Fig. 6. Settings of sensor utilized for generating fragments. (a) Given the used sensor settings, (b) we simulate 3 depth sensors with different pitch angles and statistical noise to capture the individual fragments per location.

not accessible, novel ones are generated based on the provided scans.

4.2. Dataset generation

In this section, we describe the process of generating spatiotemporal ground truth pose information for all fragments that will serve the pairwise and multi-way registration tasks. Additionally, we explain how to generate fragments and create pairs, as well as the formulation of the final dataset.

4.2.1. Alignment of 3D scans

To establish a single, global spatiotemporal coordinate system for all scans that correspond to the same area, a manual rough alignment of them is initially performed. This involves using the "Align (point pairs picking)" tool in CloudCompare (Cloud Compare, 2025), a point cloud processing software. By manually selecting 10–15 correspondences between scans, we obtain an initial alignment. In cases where an area consists of more than two stages, an *anchor* stage is selected based on the highest area coverage across all stages. The remaining stages are then aligned with respect to this anchor stage. To refine the above alignment results, we programmatically employ the iterative closest point (ICP) (Besl and McKay, 1992) algorithm. ICP aims to minimize the root mean square error (RMSE) between the input stages, ensuring a more accurate result. Examples of the global registration results are shown in Fig. 5.

Why ICP. The use of ICP for this change-depicting data is not ideal, since one of ICP's main assumptions is that the scene is static. A better circumstance would be to identify changed points and exclude them from the optimization process. However, this is non-trivial because determining the changed parts requires aligning the data first, leading to a circular dependency problem. As a compromise, we choose ICP to refine the rough initial alignment and assume that non-changing points dominate the optimization process. The median cross-stage displacements (i.e., the distance between a point in a non-anchor stage and its corresponding point in the anchor stage) after applying ICP for each area are as follows: 0.127, 0.135, 0.141, 0.130, 0.127, and 0.117 m, respectively. The effect of ICP can also be qualitatively evaluated in Fig. 5.

4.2.2. Fragment generation

We utilize the available 3D scans to generate synthetic fragments that mimic real-world conditions, such as sensor settings and the capturing process.

Sensor settings. The Matterport Camera consists of three RGBD sensors (Primesense, 2025). The configuration of the sensors is optimized to achieve maximum vertical coverage of the scene from a single viewpoint, with a pitch range of $\pm 30°$. As the sensors rotate around the gravity axis, the system captures data at intervals of 60° (Fig. 6(a)). This process generates 18 RGBD images, which are then stitched together to form an equirectangular RGBD image. Further projecting the equirectangular image in the 3D space results in the fragment captured at that location (Fig. 6(b)).

To simulate this sensor setup, we use the Blender (2025) software and model it according to the described configuration. We incorporate statistical noise models (Handa et al., 2015; Gschwandtner et al., 2011) for the Primesense sensor to mimic the real-world characteristics of the captured data. The depth images are sampled based on the reconstructed mesh of the scene, allowing us to closely simulate the raw output of the actual sensor at a specific location. In our simulation setup, we focus on simulating the depth sensor only, as accurately simulating realistic textures from the reconstructed 3D mesh is a challenging task. This does not affect the benchmark, since pairwise and multi-way registration tasks rely on geometric information rather than color.

Finding fragment locations. The next step is to sample possible 3D locations of the sensor in each area and stage, so as to achieve maximum coverage. We compute these locations on a 2D occupancy probabilistic map of each stage (Fig. 7), by taking into account constraints imposed by the sensor system. First, we calculate a 2D occupancy map for each stage by taking into account the obstacle information in the vertical space. We exclude any data outside the height range of $[0.5, 2]$ m to remove occupancy resulting from floor or ceiling points. The maximum sensor height is set at 1.75 m, so any location below 2 m should be unobstructed for it to be considered valid. To create the occupancy map, we densely sample 3D points from the underlying mesh in a uniform manner. The map is defined by a grid with a cell size of 0.10 m $\times$ 0.10 m, and if a point falls within a cell, the cell is marked as occupied.

Next, we enrich the occupancy map with probabilistic information that prioritizes free cells that are further away from occupied ones. This ensures that the sensor locations are preferentially placed further away from obstacles, as would occur in a real-world setting. The probabilistic occupancy map is then used to densely sample sensor locations. The sampling process starts by randomly selecting the first point, and subsequent sensor locations are placed within a 2D Euclidean distance uniformly distributed in the range of $[1, 4]$ m. The sensor height is varied in the range of $[1.5, 1.75]$ m, also uniformly distributed. The objective is to achieve maximum coverage of the 2D map, taking into account that the maximum depth sensing range of the employed sensor is 4.5 m.

4.2.3. Pairwise registration dataset generation

To select fragment pairs for the pairwise registration task, we aim to create a diverse and balanced dataset that represents various scenarios of overlap. We employ three metrics to guide the selection process:

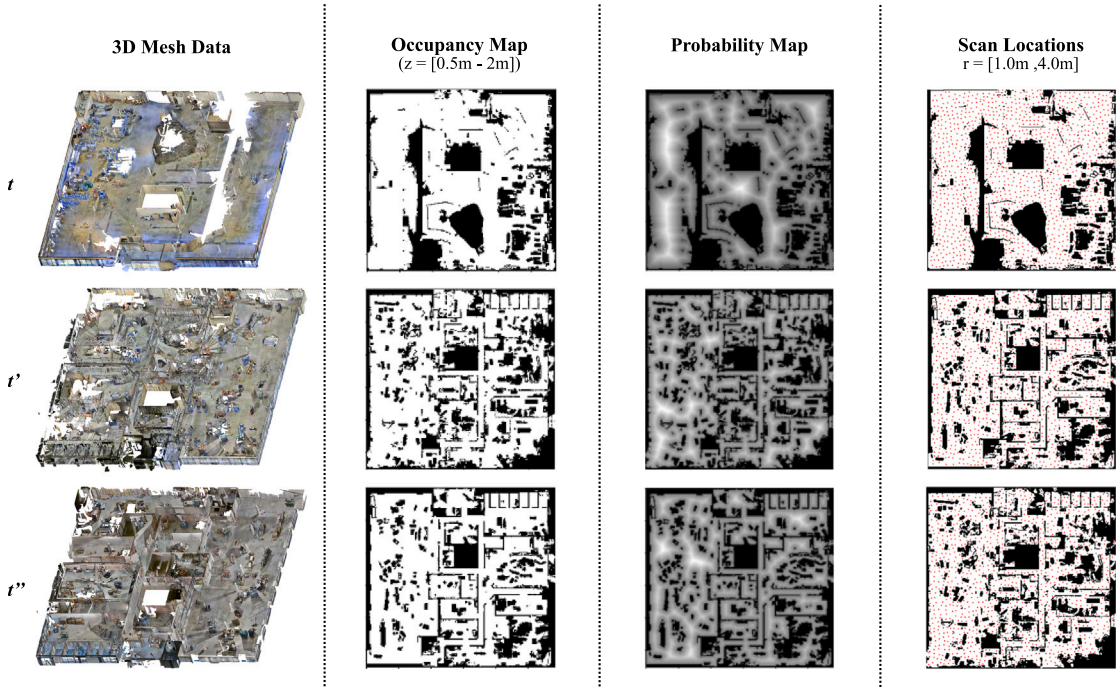


Fig. 7. Sampling fragment locations on scans. Locations are selected on the basis of a probabilistic 2D occupancy map, taking into account the employed sensor characteristics and real-world settings.

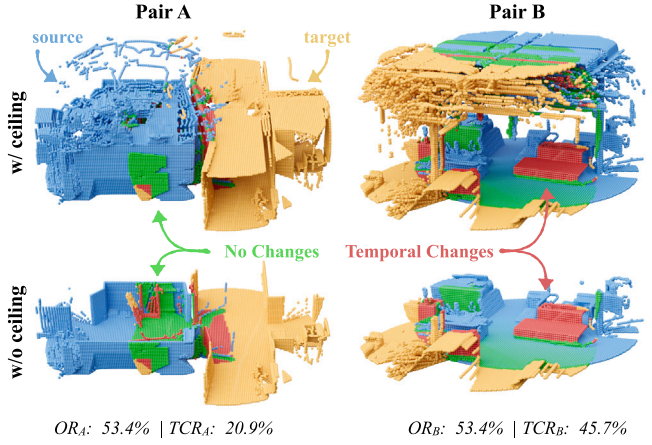


Fig. 8. Significance of overlap (OR) and temporal change (TCR) ratios in **NSS**. Although the OR is the same in both pairs, TCR is substantially higher in pair B with almost half of the points in the overlapping area having changed.

Overlap Ratio (OR). This is an existing metric in the spatial registration domain (Huang et al., 2021) and refers to the ratio of spatially overlapping points between two fragments, regardless of whether they belong to the same or different stage. Given a pair of registered fragments, it measures the ratio of overlapping points over the whole point cloud (Fig. 9(a)). Specifically, given source $\mathbf{X}^{(S)}$ and target $\mathbf{X}^{(T)}$ fragments, the overlapping part between them $\mathbf{O}(\mathbf{X}^{(S)}, \mathbf{X}^{(T)})$ is calculated as:

$$\mathbf{O}(\mathbf{X}^{(S)}, \mathbf{X}^{(T)}; \tau) := \{\mathbf{x} \in \mathbf{X}^{(S)} \mid \mathrm{NN}(\mathbf{x}, \mathbf{X}^{(T)}) \leq \tau\} \tag{2}$$

Then, the overlap ratio is defined as:

$$\text{Overlap Ratio} := \frac{|\mathbf{O}(\mathbf{X}^{(S)}, \mathbf{X}^{(T)}; \tau)|}{|\mathbf{X}^{(S)}|} \tag{3}$$

Note that, in the case of different stage registration, the overlap ratio reflects the ratio of no-change points under the threshold τ. For all evaluations, we set $\tau = 0.2$ m, a threshold commonly used in scanning-based 3D datasets to determine a sufficiently close to the ground-truth transformation (Zeng et al., 2017).

Temporal Change Ratio (TCR). *OR* falls short in providing information about possible temporal changes that might have occurred in the overlapping region between two fragments that originate from different stages. To counter this limitation, we define and introduce the concept of the *temporal change ratio*. This ratio denotes the proportion of points that have undergone changes within the overlap region encapsulated by a 3D convex hull (Fig. 9(b)). Following the same threshold used in OR, we consider a point as changed if it lacks neighbors within the $\tau = 0.2$ m Euclidean range in the other stage.

More specifically, given source fragment $\mathbf{X}^{(S,t)}$ from stage t and target fragment $\mathbf{X}^{(T,t')}$ from stage t', the temporal change ratio is defined as:

$$\mathrm{TCR} := 1 - \frac{|\mathbf{O}(\mathbf{X}^{(S,t)}, \mathbf{X}^{(T,t')}; \tau)|}{|\mathbf{H}(\mathbf{X}^{(S,t)}, \mathbf{X}^{(T,t')})|} \tag{4}$$

Here, the convex envelope $\mathbf{H}$ represents the boundary of the overlap region between the two fragments, and is defined as:

$$\mathbf{H}(\mathbf{X}^{(S,t)}, \mathbf{X}^{(T,t')}) := \{\mathbf{x} \in \mathbf{X}^{(S,t)} \mid \mathrm{hull}(\mathbf{X}^{(T,t')}) = \mathrm{hull}(\mathbf{X}^{(T,t')} \cup \mathbf{x})\} \tag{5}$$

where $\mathrm{hull}(\cdot)$ is the convex hull of a given fragment.

Fig. 8 showcases two examples of fragment pairs, along with their respective overlap and temporal change ratios. While the overlap ratio remains consistent in both cases, a notable difference can be observed in the temporal change ratio. This discrepancy indicates that registering (b) is more difficult than (a) due to the scarcity of static points available for deriving correspondences within the overlapping region. Adding to the challenge is the fact that the majority of static points are associated with flat surfaces, which further restricts finding correspondences.

Geometric complexity. The amount of geometric complexity of points in the overlapping region between two fragments plays an important role in defining easier versus more challenging registration pairs. To assess it, we calculate the surface variation or else curvature (Weinmann et al., 2013), which provides information about the local shape, using

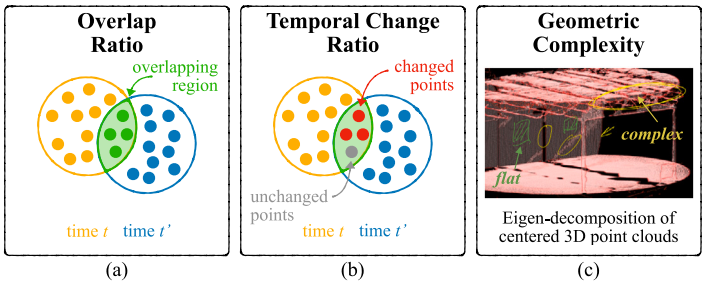


Fig. 9. Metrics for selecting fragment pairs. We employ three metrics that evaluate (a) spatial, (b) temporal, and (c) geometric characteristics of fragment pairs.

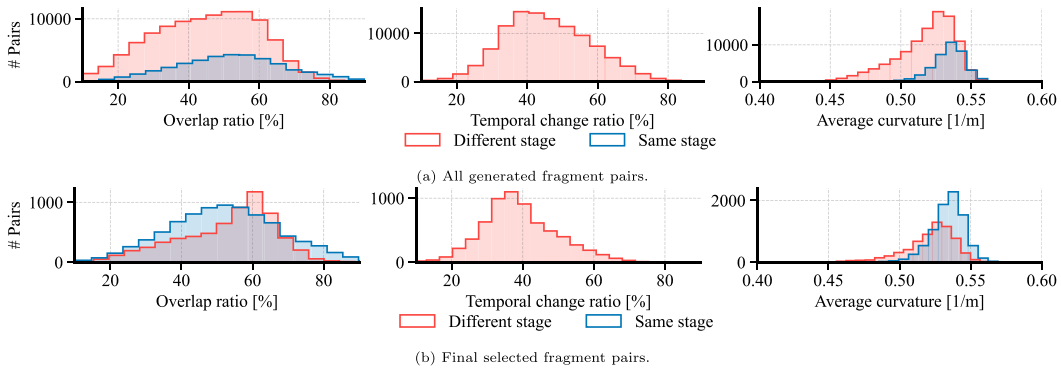


Fig. 10. **NSS** dataset statistics. The histograms showcase the distribution of fragment pairs with respect to spatial and temporal characteristics. These are the overlap ratio, temporal change ratio, and average curvature.

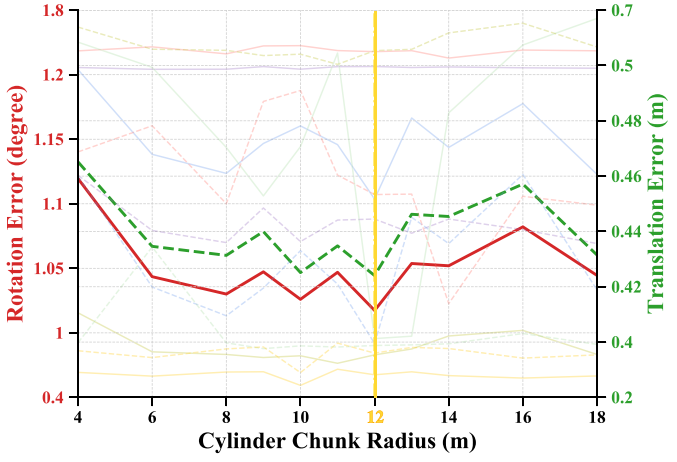


Fig. 11. Refinement of fragment pair ground truth registration using cylinders of different radius. The 12 m radius is the one that provides the smallest error correction with respect to the initial global alignment. We plot the average error curves in **red** and **green**. Error curves per area are colorized as: A: purple | B: blue | C: pink | D : green | E : yellow | F :orange.

eigendecomposition (Fig. 9(c)):

$$C_\lambda(\mathbf{x}) = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \tag{6}$$

where λ_i is the ith eigenvalue for the 3D Structure Tensor (Bigun, 1987) over the points within a sphere of radius $r = 0.5$ m centered at $\mathbf{x}$. This radius is empirically chosen to include enough points to reduce noise while maintaining sensitivity to local features. For every pair of fragments, we provide the averaged curvature value for all points located in their overlapping part, i.e.,

$$C_\lambda(\mathbf{X}^{(S)}, \mathbf{X}^{(T)}) = \frac{1}{|O|} \sum_{\mathbf{x} \in O} C_\lambda(\mathbf{x}), \qquad \text{where } O := \mathbf{O}(\mathbf{X}^{(S)}, \mathbf{X}^{(T)}). \tag{7}$$

Table 2
Details on area coverage and the total number of fragment pairs in the **NSS** dataset. The coverage of each area in the dataset may vary slightly at each stage due to construction activities, since certain parts may be inaccessible or obstructed.

Area	Stages	Area [m²]		Pairs	
		Min	Max	Non-temporal	Temporal
A	3	3159.8	3342.0	9604	3825
B	6	1482.4	2191.9	6801	5876
C	5	652.9	812.9	2185	1682
D	2	1112.1	1129.1	1557	844
E	4	944.3	5019.5	5224	2226
F	5	1322.4	2661.1	12 060	14 359

Regions with higher curvature values indicate more intricate and complex geometry and are generally easier to align. Regions with flatter geometry, characterized by lower curvature values, make the registration task more challenging since they exhibit simpler geometric shapes with less variability.

Final fragment pairs. To determine the final set of fragment pairs for the **Nothing Stands Still** dataset, we compute the metrics described above for all possible pairs within and across stages. After computing them, we create the distribution curves which provide insights into the data characteristics (Fig. 10(a)). To ensure a diverse and balanced dataset, we sample data in a uniform manner from them (Fig. 10(b)), i.e., we select fragment pairs so that they represent a range of overlap ratios, temporal change ratios, and geometric complexities. Details on the final number of fragment pairs for the **Nothing Stands Still** dataset are shown in Table 2.

4.2.4. Fragment alignment

While the global alignment achieved in Section 4.2.1 provides a globally minimum registration error among scans, it does not guarantee a locally optimal registration between fragment pairs. This is illustrated in Fig. 5. Using this initial and imperfect alignment as ground truth to systems will result in a very noisy learning process and ultimately

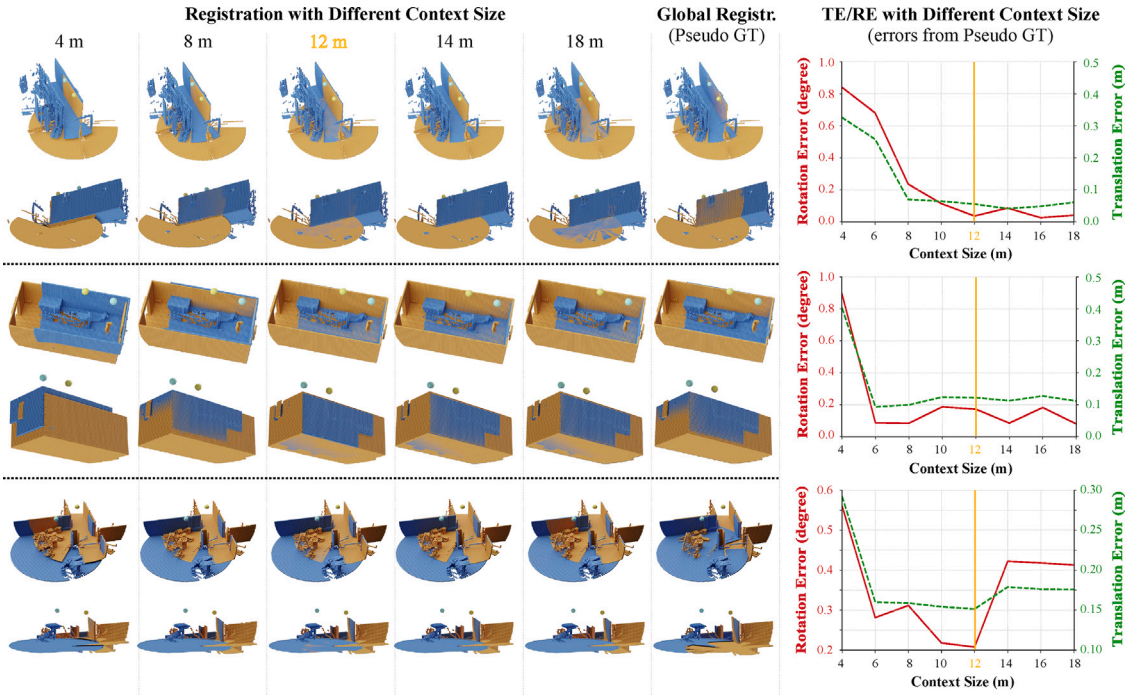


Fig. 12. Qualitative examples of the fine-tuned pairwise alignment using different context sizes. The 12 m radius provides on average the lowest translation (TE) and rotation (RE) errors between the pseudo ground truth and the fine-tuned result. This is particularly noticeable in the alignment of floors and walls. Even in the last row, where the TE is higher, the 12 m radius still achieves a correct alignment. Two viewpoints are shown for a better understanding of the results. The ceiling has been removed for visualization purposes.

to gross errors in registration. To address this, individual refinement of transformations is performed for each pair to achieve a locally minimum solution.

However, directly refining the fragment pairs alone does not always yield the optimal solution for that pair. Context, particularly overlapping regions, plays a crucial role in registration. More context can provide more static anchors, but excessive context can hinder the process if there is significant temporal change. To determine the optimal context size around a fragment location, cylindrical chunks are cropped from the scans centered around the sensor locations of each fragment. Cylinders of different radii in the range of $[2, 18]$ m are used, with a step size of 2 m. The range is selected empirically to allow for sufficient hyper-parameter tuning, with the upper limit encompassing the entire building in most cases. This results in 9 different-sized cylinders per sensor location, with the height of each cylinder being the total height of the scan at that location.

The alignment of the cylinder pairs is refined using the global alignment computed in Section 4.2.1 as the initialization (pseudo ground truth). The relative reconstruction error (translation error (TE) and rotation error (RE)) is computed with respect to the pseudo ground truth. The assumption is that the optimal radius will have a minimal deviation in terms of reconstruction error from it. Based on the results (Fig. 11), it is determined that a radius of 12 m provides a balance of context for refined fragment registration. This choice is confirmed by visualizing various random samples of refined fragment pairs (Fig. 12). Even when there is larger-than-average displacement from the pseudo ground truth, the final registration results are improved. Finally, the pairwise registration ground truth is created using the refined local transformations on the 12 m cylinders for all fragment pairs.

4.2.5. Multi-way registration dataset generation

For the multi-way registration task, the ground truth transformation across fragments is obtained from the global alignment achieved in Section 4.2.1. It is worth noting that the train and test sets for the multi-way registration are a subset of those used in the pairwise registration. This is due to certain fragment pairs in the pairwise registration task not having sufficient overlap with other pairs (i.e., at least 10%). This results in many disjoint components per area instead of a coherent and connected global spatiotemporal map. While the goal is to have high intra-fragment overlap, we keep disjoint components that contain enough fragments (at least 50). As a result, in certain areas, there may be more than one disjoint spatiotemporal map. During the simulation process, it is possible to generate fragments that create a single map per area, however, this scenario is not always realistic. We note that, in the multi-way registration experiments in Section 6.2, we use the subset test data but not the subset training data. State-of-the-art methods for this task aim to globally optimize the pairwise registration results, which does not require new training. The multi-way training annotations are provided as part of the **NSS** dataset for future work that may address this task independently of a prior pairwise registration step.

5. Nothing stands still benchmark

The **Nothing Stands Still (NSS) benchmark** consists of the tasks of pairwise and multi-way registration.

Data splits. To evaluate the generalization ability of the methods, we define three different data splits:

- *Original*: This is the standard data split in the spatial registration domain. The training and testing are performed on fragment pairs from all areas and stages. Although the train–test fragments are not duplicates, they originate from the same area and stage, allowing methods to learn about the composition of an area during training.
- *Cross-area*: In this split, the training is done on fragments from three areas (all stages), and the testing is performed on the remaining areas. This evaluates a method's generalization ability to unseen areas.

Table 3
Generalization type in **NSS** data splits. Each split receives during testing, with respect to training, data from *unseen areas*, and *unseen stages* from the same area.

Data split	Unseen stage	Unseen area
Original	✗	✗
Cross-Area	✗	✓
Cross-Stage	✓	✗

Table 4
Area split per evaluation task on the **NSS** dataset.

	Cross-Area			Cross-Stage					Original
Training									
Area	A	B	F	A	B	C	E	F	All
Stage	All	All	All	1–2	1–3	1–3	1–2	1–3	All
Testing									
Area	C	D	E	A	B	C	E	F	All
Stage	All	All	All	3	4–6	4–5	3–4	4–5	All

- *Cross-stage*: Here, the training is conducted on the first 50% of stages of each area, and the testing is performed on the remaining ones. This split aims to assess the domain gap across stages.

Table 3 provides an overview of the splits, and Table 4 offers more detailed information about them. Both the pairwise registration and multi-way registration tasks are evaluated on all three data splits.

Evaluation metrics. To evaluate both registration tasks, we follow the same evaluation metrics as in the spatial registration domain (Huang et al., 2021) and use: registration recall (Recall), relative translation error (RTE), and relative rotation error (RRE). For the RTE and RRE metrics, their formal definitions are,

$$\text{RRE} = \angle(\mathbf{R}_{GT}^{-1}\hat{\mathbf{R}}), \quad \text{RTE} = \|\mathbf{v}_{GT} - \hat{\mathbf{v}}\|_2, \tag{8}$$

where $\{\mathbf{R}_{GT}, \mathbf{v}_{GT}\}$ and $\{\hat{\mathbf{R}}, \hat{\mathbf{v}}\}$ denote the groundtruth and estimated rigid transformation, respectively. Here,

$$\angle(X) = \arccos\left(\frac{\text{trace}(X) - 1}{2}\right), \tag{9}$$

returns the angle of rotation matrix X in degrees.

The registration recall is defined as the ratio of the number of successfully registered point cloud pairs to the total number of point cloud pairs. In our benchmark, a pair is considered successfully registered if it satisfies two criteria: the relative rotation error (RRE) is less than 10 degrees, and the relative translation error (RTE) is less than 0.2 m. The thresholds are common values for indoor point cloud registration (Zeng et al., 2017). This metric provides a comprehensive measure of the registration algorithm's accuracy with varying degrees of spatial displacement.

To measure the performance of methods in spatiotemporal registration, we employ overlap ratio and temporal change ratio as ablation metrics. Please refer to Section 4.2.3 for their definitions.

6. Experiments

In the pairwise registration task of the Nothing Stands Still benchmark, we evaluate state-of-the-art approaches that include both handcrafted and learned features: FPFH (Rusu et al., 2009), D3Feat (Bai et al., 2020), FCGF (Choy et al., 2019b), Predator (Huang et al., 2021), and GeoTransformer (Qin et al., 2022).[3] In the multi-way registration task, we evaluate the state-of-the-art methods in Choi et al. (2015) and Yew and Lee (2021b), which we initialize with the registration results of the two best performing methods on the pairwise registration task.

6.1. Pairwise spatiotemporal registration

We report the results in Table 5. Overall, Predator performs the best in the majority of the metrics for all data splits and temporal ablations. However, different-stage pairs pose significant challenges for all registration methods. The average performance drops between same-stage and different-stage pairs over all methods by 40.8 p.p., 41.5 p.p., 37.4 p.p. for the cross-area, cross-stage, and original splits respectively. We can also observe that learning methods achieve better performance than the hand-engineered FPFH, especially for temporal registration. FPFH only successfully registers about 1% of the different-stage pairs, significantly lagging behind other learning-based methods. Although FPFH's RE for successfully registered pairs is very low, the low registration success rate suggests that these results are unclear. Indeed, when computing the RE over all pairs in the dataset, we see that the orders are higher in magnitude. Hence, RE calculation on only successfully registered pairs, does not fully showcase the robustness of a method. We also observe that the modeling of interactions between fragments of different stages may play a key role in temporal registration. For example, compared to D3Feat, Predator and GeoTransformer show a large margin of 23.7 and 12.8 p.p. in different-stage pairs in the Original split. We hypothesize that the attention mechanism they utilize between the inputs enables them to capture more temporal-related patterns, while other methods treat the fragment input pairs independently.

When comparing the results of different data splits, we notice that methods perform the best on the cross-stage split and worst on the cross-area split. This behavior is expected for both cases. In the cross-stage split, methods learn the general structure and characteristics of the area during training and are able to make predictions on unseen stages more accurately. This has practical applications in industries such as construction or building management, where a small initial annotation effort can lead to significant future gains. In the cross-area split, methods struggle to generalize to unseen environments, which is a common challenge in various computer vision tasks. As mentioned above, the registration of fragment pairs from different stages poses difficulties, which is further emphasized in the cross-area split.

Fig. 13 provides a histogram of registration recall for all data splits based on the overlap ratio of fragment pairs. The three best-performing methods in the pairwise registration task, namely D3Feat, GeoTransformer, and Predator, are included in the evaluation. The results show a clear trend where higher overlap ratios correspond to higher recall values across all splits. Fig. 14 presents the histogram analysis based on the temporal change ratio. It demonstrates that as the temporal change increases, the registration problem becomes more difficult. Among the three dataset splits, the cross-area split exhibits the least robustness against large temporal changes. It is noteworthy that these methods perform exceptionally well in the prominent spatial registration benchmarks of 3DMatch (Zeng et al., 2017) and 3DLoMatch (Huang et al., 2021), achieving high accuracy rates in the range of 80% to 90% (Table 6). However, their performance drops by around 50% in the **NSS** benchmark. While the difference in performance between 3DLoMatch and **NSS** is less pronounced for pairs with low overlap (10%–30%), for different-stage pairs, regardless of the overlap percentage, it is significantly lower.

Figs. 19 and 20 provide example results of spatiotemporal pairwise registration for D3Feat, GeoTransformer, and Predator. Consistent with the quantitative results, Predator demonstrates more accurate registration compared to the other methods. In cases where the overlap ratio is very high and the temporal changes have minimal impact on the main structure of the scene (row *b*) or do not exist (row *c*), all three methods achieve similarly good results, which is an expected behavior. However, there are scenarios where D3Feat struggles to register pairs correctly,

[3] We follow the original training protocol per method, and integrate all the evaluated methods in our point cloud registration codebase. The codebase is open-sourced together with the benchmark.

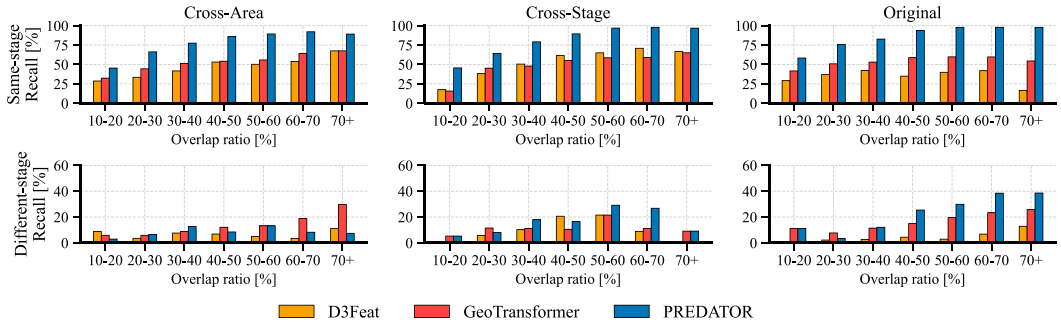


Fig. 13. Registration recall [%] per overlap ratio (OR) bin for existing 3D point cloud registration methods. A clear performance gap is visible between same-stage pairs (top row) and different-stage pairs (bottom row) for these methods.

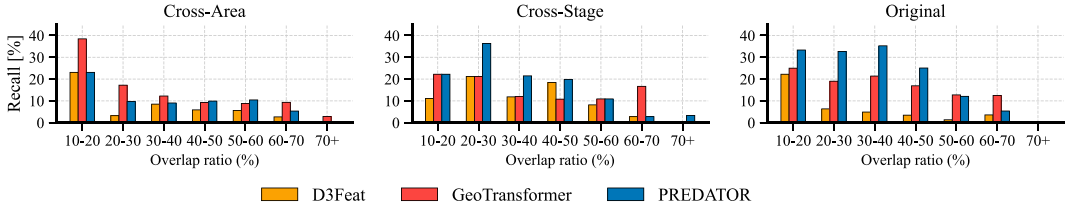


Fig. 14. Registration recall [%] per temporal change ratio (TCR) bin for existing 3D point cloud registration methods on different-stage pairs. It is evident that larger temporal changes pose greater challenges for these methods, and that the cross-area split is a setting with increased challenges.

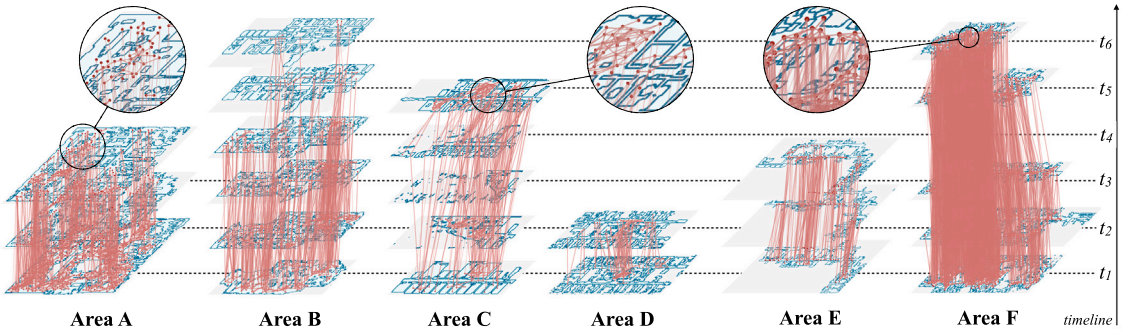


Fig. 15. Spatiotemporal graphs for the multi-way registration task. Nodes represent fragment locations and edges denote that these pairs are overlapping in the 3D spatiotemporal map of each area. The resulting graphs are dense both in nodes and edges.

even with a high overlap ratio. This is particularly evident in cases where there are significant temporal changes and the scene geometry contains repetitive elements, such as studs (rows *a* and *f*). This limitation is attributed to D3Feat's reliance on local geometry constraints and independent fragment processing. In a scenario with low overlap and no temporal change (row *d*), both D3Feat and GeoTransformer fail to find the correct alignment, while PREDATOR performs better. Lastly, in row *e*, all three methods encounter a failure case. Here, not only is the overlap low, but the two rooms are closely similar, making it a challenging scenario to solve. The main differences are the cut-out corners in the blue-colored fragment and the mirrored location of the doors.

6.1.1. *Effect of temporal data*

To further investigate the impact of training on both same-stage and different-stage pairs on the registration recall of D3Feat, GeoTransformer, and PREDATOR, we conducted the following experiment: we only trained on the same-stage pairs available in different data splits. The evaluation was still performed on the entire test set, which includes different-stage pairs. The results in Table 7 indicate that the presence of different-stage pairs hinders the training process for some methods. When trained exclusively with same-stage data, D3Feat demonstrates significantly better performance during testing. We hypothesize that this improvement is due to D3Feat relying on the local geometric assumption that similar local geometric structures are expected to be registered together. As expected, the recall for all methods and splits in same-stage registration is higher when trained solely on same-stage pairs. This suggests that the methods struggle to effectively distinguish between the spatial and temporal characteristics of the data, thereby

Table 5
Pairwise registration results of existing 3D point cloud registration methods on **Nothing Stands Still**. We report registration recall (Recall) and translation (TE) and rotation (RE) errors. For TE and RE, we report the average measurements among: [successfully registered pairs]/[all pairs]. The first value is the standard evaluation setting.

Method	Cross-Area				Cross-Stage				Original			
	Recall [% ↑]	RMSE [m ↓]	TE [m ↓]	RE [° ↓]	Recall [% ↑]	RMSE [m ↓]	TE [m ↓]	RE [° ↓]	Recall [% ↑]	RMSE [m ↓]	TE [m ↓]	RE [° ↓]
All spatiotemporal pairs												
FPFH (Rusu et al., 2009)	22.83	3.30	0.66/3.08	**0.20**/43.21	18.73	2.53	0.80/2.43	**0.16**/45.87	11.70	2.52	0.44/2.43	**0.10**/45.32
FCGF (Choy et al., 2019b)	28.22	2.07	1.83/2.01	0.62/29.25	37.70	1.81	1.29/1.78	0.52/41.04	24.43	2.24	1.09/2.04	0.76/39.89
D3Feat Bai et al. (2020)	31.77	1.98	0.08/1.95	1.44/**24.22**	51.37	1.62	0.07/1.57	1.19/32.09	22.73	2.37	0.09/2.26	1.45/33.09
PREDATOR (Huang et al., 2021)	**55.53**	**1.09**	**0.05/1.08**	0.98/25.05	**76.73**	**0.77**	**0.04/0.68**	0.74/**15.27**	**64.97**	**0.71**	**0.06/0.65**	0.79/**13.52**
GeoTransformer (Qin et al., 2022)	38.13	1.24	0.14/1.28	0.64/27.90	47.78	0.98	0.14/0.98	0.39/22.27	39.07	0.96	0.14/0.99	0.41/22.93
Only same-stage pairs												
FPFH (Rusu et al., 2009)	32.86	2.46	1.57/ 2.34	**0.28**/34.41	46.40	1.94	1.12/1.90	0.38/33.42	30.82	2.58	1.13/2.42	**0.27**/29.35
FCGF (Choy et al., 2019b)	39.32	1.88	1.78/1.84	0.55/28.01	44.65	1.77	0.98/1.76	0.41/30.47	42.86	2.24	0.56/2.23	0.44/32.12
D3Feat Bai et al. (2020)	43.62	1.91	0.08/1.93	1.31/24.05	58.47	1.48	0.07/1.48	1.10/28.39	36.51	2.09	0.08/2.05	1.36/27.22
PREDATOR (Huang et al., 2021)	**76.80**	**0.81**	**0.05/0.83**	0.86/**18.41**	**87.49**	**0.44**	**0.04/0.48**	0.69/**9.89**	**92.99**	**0.27**	**0.04/0.27**	0.67/**4.83**
GeoTransformer (Qin et al., 2022)	50.88	1.07	0.13/1.13	0.54/23.73	54.07	0.79	0.14/0.83	**0.37**/17.26	55.59	0.69	0.14/0.73	0.35/17.02
Only different-stage pairs												
FPFH (Rusu et al., 2009)	1.06	4.88	**0.07**/4.32	**0.03**/65.89	0.82	4.23	**0.09**/4.06	**0.02**/72.43	0.42	4.21	0.03/4.06	**0.00**/78.01
FCGF (Choy et al., 2019b)	5.21	3.22	2.13/3.21	2.17/45.61	14.06	4.15	2.40/4.02	0.93/62.15	10.52	3.28	2.75/3.23	1.74/53.24
D3Feat Bai et al. (2020)	6.12	2.01	0.16/2.01	3.48/**24.57**	12.85	2.40	0.13/2.03	3.56/52.18	4.76	2.75	**0.12**/2.53	2.43/40.76
PREDATOR (Huang et al., 2021)	9.49	1.71	0.16/1.62	3.08/39.42	**18.42**	**2.03**	0.10/**1.77**	2.08/**44.46**	**28.42**	**1.28**	0.13/**1.16**	1.29/**24.85**
GeoTransformer (Qin et al., 2022)	**10.55**	**1.62**	0.15/**1.59**	1.63/36.91	13.39	2.25	0.16/1.81	0.96/49.66	17.51	1.31	0.13/1.34	0.66/30.62

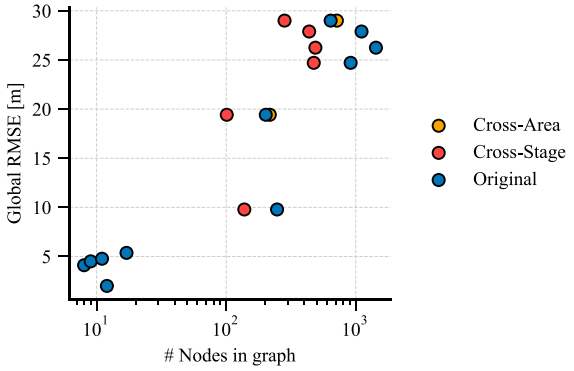


Fig. 16. The measured Global RMSE of the multiway spatiotemporal registration using PoseGraphNet.

Table 6
Comparison of performance on **NSS** (original split) with that on 3DMatch and 3DLoMatch. We compare the registration recall of the three best-performing methods on **NSS** and clearly observe that their results on the standard spatial registration benchmarks are substantially higher on ours.

Benchmark	D3Feat	PREDATOR	GeoTrans.
Standard overlap [30%+]			
3DMatch (Zeng et al., 2017)	**82.2**	**89.0**	**92.0**
NSS *(all)*	34.1	58.8	40.3
NSS *(same-stage only)*	47.9	83.1	54.3
NSS *(different-stage only)*	6.6	10.6	12.5
Low overlap [10–30%]			
3DLoMatch (Huang et al., 2021)	**37.2**	59.8	**75.0**
NSS *(all)*	25.2	47.6	32.6
NSS *(same-stage only)*	32.6	**62.4**	42.2
NSS *(different-stage only)*	4.4	5.9	5.9

affecting same-stage registration due to *temporal noise*. However, in the case of different-stage registration, the methods benefit from the presence of such pairs in training, indicating that some learning is occurring. Fig. 13 illustrates that these benefits primarily stem from the highly overlapping fragment pairs. Despite the advantages of utilizing all spatiotemporal data during training, there is still significant room for the methods to fully exploit the potential of different-stage pairs.

6.1.2. Comparison to RIO10 spatiotemporal dataset

RIO10 (Wald et al., 2020) is a recent indoor point cloud dataset specifically designed for camera re-localization tasks in changing environments. We follow the **NSS** benchmark protocol to evaluate the spatiotemporal registration performance for D3Feat, PREDATOR, and GeoTransformer on this dataset. We consider only the *original* and *cross-stage* splits, since, due to the spatial scale of each area, the cross-area and original splits overlap. The results are presented in Table 8. In comparison to the performance on the **NSS** dataset (refer to Table 5), we observe better registration performance across all metrics on average when evaluating on RIO10. Particularly, the drop in performance for different-stage pair registration is significantly smaller. This suggests that the domain gap between same-stage and different-stage pairs in this dataset is not as drastic, which is expected considering the smaller changes depicted in the scenes.

6.2. Multi-way spatiotemporal registration

Both Choi et al. (2015) and PoseGraphNet (Yew and Lee, 2021b) require the construction of a pose graph, where nodes represent fragments and edges denote the predicted transformation between two connected fragments. We define the pose graph including the fragments from all stages within an area. In both cases, the edges are initialized with the relative transformations from the pairwise registration results for the two best-performing methods, PREDATOR and GeoTransformer. Suppose no edge exists between two fragments in the graph, it implies that they either do not overlap or overlap by less than 10%, which is considered insufficient for meaningful registration for current methods. Fig. 15 displays the dataset pose graphs for all areas in the original split, which comprise numerous nodes and edges, thereby making the optimization process for finding a globally optimal alignment challenging.

In Choi et al. (2015), the authors distinguish between odometry and uncertainty edges. They acquire this distinction directly from the input data, which is an RGBD video sequence. Since this is not applicable to our case, we select odometry edges by constructing a minimum spanning tree from a randomly chosen node. We experimented with various edge selection methods but did not observe significant differences in the results. Next, we initialize the edge weights using the predicted average matchability scores from the two best-performing pairwise registration methods (GeoTransformer and PREDATOR). We also select all non-temporal pairs as constrained pairs, i.e., they will not

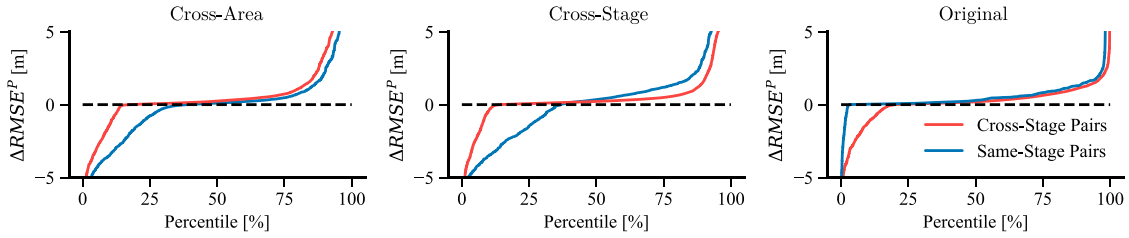


Fig. 17. Percentile distribution of the change in Pairwise RMSE ($\Delta RMSE^P$) across all graph edges using PoseGraphNet for multi-way registration, for all dataset splits. The area under the dashed line demonstrates a decrease in RMSE and the area above demonstrates an increase. A curve's slope points to the rate of change. Areas created between the dashed line and a curve also demonstrate the amount of edges changed — larger area means more edges changed.

Table 7
Effect of training with temporal data on registration recall [%]. Methods are trained using either *all* training data or *same-only* stage pairs. Testing is evaluated on all pairs. Values in red denote a drop in performance, whereas values in green an increase.

Method	Cross-Area			Cross-Stage			Original		
	All	Same-only	Diff	All	Same-only	Diff	All	Same-only	Diff
All testing pairs									
D3Feat (Bai et al., 2020)	31.77	**49.53**	−17.76	51.37	**66.07**	−14.70	22.73	**46.60**	−23.87
Predator (Huang et al., 2021)	55.90	**58.50**	−2.60	76.63	**77.60**	−0.97	**64.67**	62.77	+1.90
GeoTransformer (Qin et al., 2022)	38.30	**38.87**	−0.57	47.24	**47.38**	−0.14	**40.37**	36.08	+4.29
Same-stage testing pairs									
D3Feat (Bai et al., 2020)	43.62	**67.06**	−23.44	58.47	**74.62**	−16.15	36.51	**70.38**	−33.87
Predator (Huang et al., 2021)	76.80	**80.51**	−3.71	87.49	**88.59**	−1.10	92.99	**93.23**	−0.24
GeoTransformer (Qin et al., 2022)	50.88	**51.71**	−0.83	**54.07**	**54.07**	0.00	55.59	**56.24**	−0.65
Different-stage testing pairs									
D3Feat (Bai et al., 2020)	6.12	**11.60**	−5.48	12.85	**19.70**	−6.85	4.76	**15.59**	−10.83
Predator (Huang et al., 2021)	9.49	**10.86**	−1.37	**18.42**	17.99	+0.43	**28.42**	23.04	+5.38
GeoTransformer (Qin et al., 2022)	**10.55**	9.08	+1.47	**13.39**	10.80	+2.59	**17.51**	9.76	+7.75

Table 8
Registration performance on RIO10 dataset (Wald et al., 2020). We follow the same data generation and evaluation protocol as in **NSS** and report registration recall (Recall) and translation (TE) and rotation (RE) errors. For TE and RE, we report the average measurements among: [successfully registered pairs]/[all pairs]. The first value is the standard evaluation setting.

Method	Cross-Stage			Original		
	Recall [% ↑]	TE [m ↓]	RE [° ↓]	Recall [% ↑]	TE [m ↓]	RE [° ↓]
All testing pairs						
D3Feat (Bai et al., 2020)	41.76	**0.04**/0.68	2.46/48.94	45.26	**0.04**/0.63	2.39/44.35
Predator (Huang et al., 2021)	**64.17**	0.05/**0.45**	**2.40**/**27.86**	**73.70**	**0.04**/**0.34**	**2.11**/**19.62**
GeoTransformer (Qin et al., 2022)	46.04	0.09/0.70	4.10/44.83	47.21	0.09/0.61	4.16/38.95
Same-stage testing pairs						
D3Feat (Bai et al., 2020)	66.33	0.03/0.51	1.57/35.21	67.40	**0.02**/0.50	**1.24**/32.56
Predator (Huang et al., 2021)	**75.94**	**0.02**/**0.37**	**1.42**/**21.91**	**81.18**	**0.02**/**0.29**	1.40/**15.93**
GeoTransformer (Qin et al., 2022)	60.97	0.08/0.69	3.90/42.68	61.71	0.08/0.66	4.02/40.69
Different-stage testing pairs						
D3Feat (Bai et al., 2020)	21.68	0.08/0.81	4.70/60.17	27.39	0.09/0.75	4.67/53.87
Predator (Huang et al., 2021)	**54.55**	**0.07**/**0.52**	**3.53**/**32.73**	**67.67**	**0.06**/**0.37**	**2.79**/**22.60**
GeoTransformer (Qin et al., 2022)	33.84	0.10/0.71	4.32/46.58	35.51	0.10/0.58	4.28/37.55

Table 9
Multi-way registration results of existing 3D optimization methods on **Nothing Stands Still**. We report pairwise ($RMSE^P$) and global registration metrics ($RMSE^G$, Recall, TE, RE) on the testing pairs of this task and compare with the pairwise registration results per split since they correspond to the performance *before* the multi-way pose graph optimization. Best values per metric and split are highlighted in **bold**.

Method	Pairwise outputs				Choi (Choi et al., 2015)					PoseGraphNet (Yew and Lee, 2021b)				
	$RMSE^P$ [m ↓]	$Recall^G$ [% ↑]	TE^G [m ↓]	RE^G [° ↓]	$RMSE^P$ [m ↓]	$RMSE^G$ [m ↓]	$Recall^G$ [% ↑]	TE^G [m ↓]	RE^G [° ↓]	$RMSE^P$ [m ↓]	$RMSE^G$ [m ↓]	$Recall^G$ [% ↑]	TE^G [m ↓]	RE^G [° ↓]
Cross-Area														
Predator (Huang et al., 2021)	**1.21**	56.23	**0.05**/**1.04**	0.95/24.89	1.91	33.43	53.61	**0.05**/1.80	0.69/21.67	1.43	28.29	**77.28**	**0.05**/1.11	**0.13**/**0.44**
GeoTransformer (Qin et al., 2022)	1.33	38.52	0.14/1.26	1.18/26.64	2.26	39.90	42.51	0.11/2.13	0.83/24.73	1.64	**27.63**	53.31	0.06/1.28	0.16/0.57
Cross-Stage														
Predator (Huang et al., 2021)	**0.88**	75.85	0.04/**0.71**	0.75/15.90	1.48	19.68	71.07	**0.03**/1.37	0.41/14.86	1.04	**18.45**	**79.94**	0.05/0.79	0.08/0.24
GeoTransformer (Qin et al., 2022)	1.05	46.62	0.14/0.99	0.97/22.55	1.74	28.19	56.90	0.10/1.64	0.63/19.84	1.13	20.93	62.93	0.04/0.88	**0.07**/**0.23**
Original														
Predator (Huang et al., 2021)	0.83	66.65	0.06/0.65	0.81/13.57	1.84	20.72	62.35	0.06/1.73	0.69/18.24	**0.53**	16.14	**82.34**	**0.05**/**0.43**	0.08/**0.37**
GeoTransformer (Qin et al., 2022)	1.23	40.23	0.14/1.00	0.97/24.00	2.83	27.71	40.49	0.12/2.54	1.04/30.58	1.08	**14.55**	65.35	0.09/0.84	**0.07**/0.60

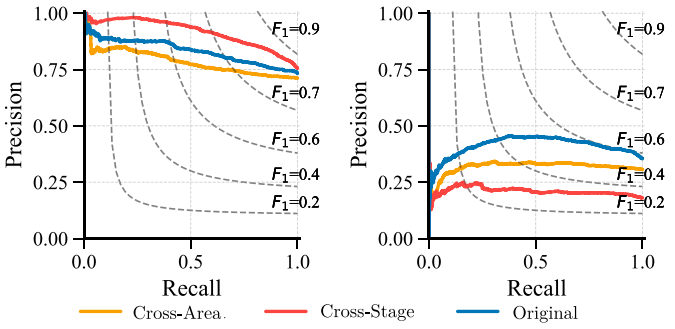


Fig. 18. Precision–recall curve on overlap classification with PREDATOR. Results are shown for same- (left) and different- (right) stage pairs. Temporal changes affect overlap detection accuracy greatly.

be pruned during optimization. In each iteration, edges with weights below a threshold are considered non-valid and are pruned from the graph. For further details on the pose graph optimization, we refer the reader to Choi et al. (2015). The objective here is to achieve a set of consistent registrations that minimize the weighted average root mean square error (RMSE) of all nodes in the pose graph.

PoseGraphNet removes the reliance on odometry data. It employs a GNN to learn to perform transformation synchronization. We train their method on the training set of the multi-way registration task with the objective of minimizing absolute rotation and translation errors. Unlike the pose graph setup for Choi et al. (2015), which initializes nodes with odometry information, PoseGraphNet starts with identity matrices for the nodes, and then refines their poses through incremental updates. All weights in this GNN are learned during training, ensuring robustness against outliers and noisy data. During inference, the trained model applies this learned synchronization recurrently for multi-way registration.

Table 9 presents the results for both methods with respect to global (RMSE^G, Recall^G, TE^G, and RE^G) and pairwise (RMSE^P) registration metrics for all dataset splits. We also compare to the pairwise registration outputs since they correspond to the performance *before* the multi-way pose graph optimization. Before analyzing them, it is important to note the reasons that the *pairwise* outputs showcase differences with respect to Table 5. Firstly, different ground truth poses are used in the two tasks. As mentioned in Section 4, the discrepancy arises from the distinction in global and local alignment for fragments and scans. Secondly, the multi-way registration test set is a subset of the pairwise one, as not all fragment pairs could be connected into a larger pose graph.

For both methods, the errors of global RMSE (RMSE^G) are a magnitude higher than in pairwise RMSE (RMSE^P), showcasing the complexity of the multi-way problem without excluding that a pairwise-to-global approach might not be the most effective one. Consistent to the pairwise evaluation trends: (i) the cross-area split is the hardest here too; (ii) evaluating TE^G and RE^G only on successfully registered pairs is not a sufficient indication of performance; and (iii) overall PREDATOR leads to improved performance than GeoTransformer. In comparison to Choi et al. PoseGraphNet performs the best overall in global metrics. This can be attributed to (i) using learning and (ii) having an objective that is closer to registration recall than weighted RMSE minimization. Particularly in cases where the pairwise registration results are low, such as in the cross-area split or in the case of GeoTransformer, PoseGraphNet brings the most benefits. We note that even though the gap between PREDATOR and GeoTransfomer decreases after multi-way registration, the difference is still substantial between the two methods, especially on registration recall. However, there is a connection between PoseGraphNet performance and the size of the pose graphs. As shown in Fig. 16, its performance is fine for graphs with fewer than 30 nodes but deteriorates with very large graphs. This is aligned with expectations since PoseGraphNet was primarily developed for datasets with smaller graphs (e.g., most scenes in ScanNet Dai et al., 2017 have fewer than 30 nodes).

It is important to consider the challenge of selecting constrained pairs in Choi et al. for our specific setting. The current selection process results in a high number of non-valid edges in our complex spatiotemporal registration scenarios. Specifically, the number of valid edges per split is: *Cross-Area*: 32.03%; *Cross-Stage*: 26.48%; *Original*: 28.30%. The cross-area split is less affected by temporal changes and consequently yields a higher number of valid edges in this constrained optimization task.

Fig. 17 ablates the percentile distribution of change in pairwise RMSE ($\Delta RMSE^P$) across cross- and same-stage graph edges for all dataset splits, when using PoseGraphNet for multi-way registration. While Table 9 shows that RMSE^G is high and RMSE^P only improves in the original split, we can still observe RMSE^P improving over some the graph edges in all splits. Areas below the zero dashed line point to improvements after the multi-way registration.

- **Decrease in RMSE^P**: In the cross-area and cross-stage splits, improvements mainly come from the same-stage edges. Cross-stage ones show a lower rate of improvement, i.e., these cases are more challenging to solve. In the original split, where the recall is already high after pairwise registration, the improvement of both same- and cross-stage pairs is smaller, with the former having minimal impact and the latter contributing the most.
- **No change:** In all splits, most cross-stage edges undergo almost no improvement (the curve is close to parallel to the dashed line), pointing to the particular challenge in temporal registration. Same-stage edges showcase a similar behavior mainly in the original split. In the cross-stage split, same-stage edges barely remain the same; they either improve or deteriorate.
- **Increase in RMSE^P**: Cross-stage and same-stage edges have a similar rate of deterioration per split. In the cross-area split, cross-stage edges show the most deterioration versus the same-stage ones, in contrast to the cross-stage and original splits. However, for the latter, the difference between cross- and same-stage edges in terms of most deterioration is small.

Fig. 21 illustrates the spatiotemporal registration after the pairwise and multi-way tasks for Area F, where different color hues represent different stages. As depicted, multi-way registration achieves superior alignment, particularly evident in the elevator shafts where pairwise registration fails to recover the alignment. However, not all areas exhibit such improvement. Figs. 22–26 present the spatiotemporal registration results for all areas per temporal stage. We observe that when pairwise registration provides a relatively good initialization of the pose graph, the multi-way step can further improve the results. However, when the pairwise step fails, no further alignment improvement can be achieved (e.g., Fig. 23). This limitation persists even when the initialization is relatively good in a few stages. The failed alignment of other stages pulls the fragments away from their initial positions since the optimization goal is to attain a globally plausible solution (Fig. 24(a)).

6.3. Overlap/non-overlap classification

In order to investigate a method's behavior when dealing with non-overlapping fragments or fragments with extremely low overlap ratios (below 10%), we conduct a binary classification task. In this task, a pair of fragments is considered overlapping if the overlap ratio exceeds a certain threshold θ (set to 0.1 in our experiment). We employ PREDATOR, the best-performing model on the **NSS** dataset, for this analysis. Instead of employing the average overlap score to perform the classification, which is available in Huang et al. (2021), we opted to use the average matching probability of all points in a fragment pair as the output probability for classification.

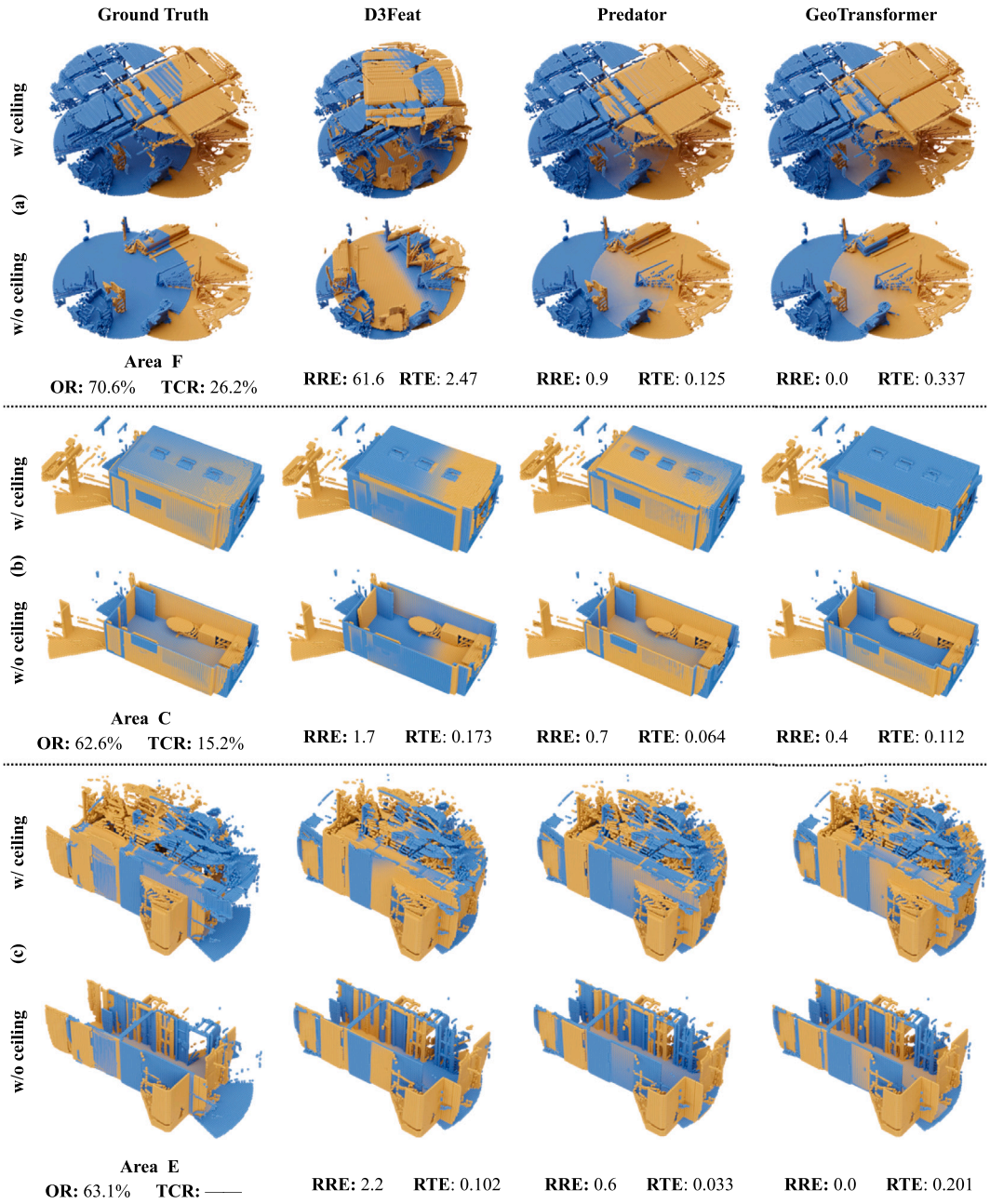


Fig. 19. Qualitative results on pairwise registration. Results on a fragment pair are showcased for D3Feat (Bai et al., 2020), Predator (Huang et al., 2021), and GeoTransformer (Qin et al., 2022). The overlap (OR) and temporal change (TCR) ratios are reported per input pair, as well as the translation (TE) and rotation (RE) errors per method (in meters and degrees respectively).

We evaluate the classification performance using the mean Average Precision (mAP) and the Area under ROC Curve (AUROC) scores. The mAP score offers an averaged measure of precision at different recall levels, which provides a comprehensive assessment of the model's ability to correctly predict overlapping pairs even when these are unevenly distributed in the data. On the other hand, the AUROC provides a measure of the model's discriminative capacity, irrespective of the decision threshold of the matching probability. Table 10 demonstrates the potential of the method to classify overlapping/non-overlapping pairs and highlights areas for improvement, particularly in scenarios with significant temporal changes that occur in real-world *in-the-wild* registration settings (Fig. 18).

We observe that the cross-stage split exhibits the best overall and same-stage scores but encounters challenges with temporal cases. This indicates that temporal changes that were not encountered during training can impact overlap classification. It is important to note that the cross-stage split only includes the first half of temporal stages per area, meaning that the method lacks knowledge of how the construction may progress in the future.[4] In this classification task, pairs from the test set are considered overlapping pairs, while pairs randomly selected from different locations are regarded as non-overlapping pairs. This task can be valuable when considering practical applications of a registration algorithm, where prior knowledge of fragment overlap may not be available from the outset.

[4] Stages across areas, although not identical, exhibit a certain pattern as construction tasks often follow a specific sequence.

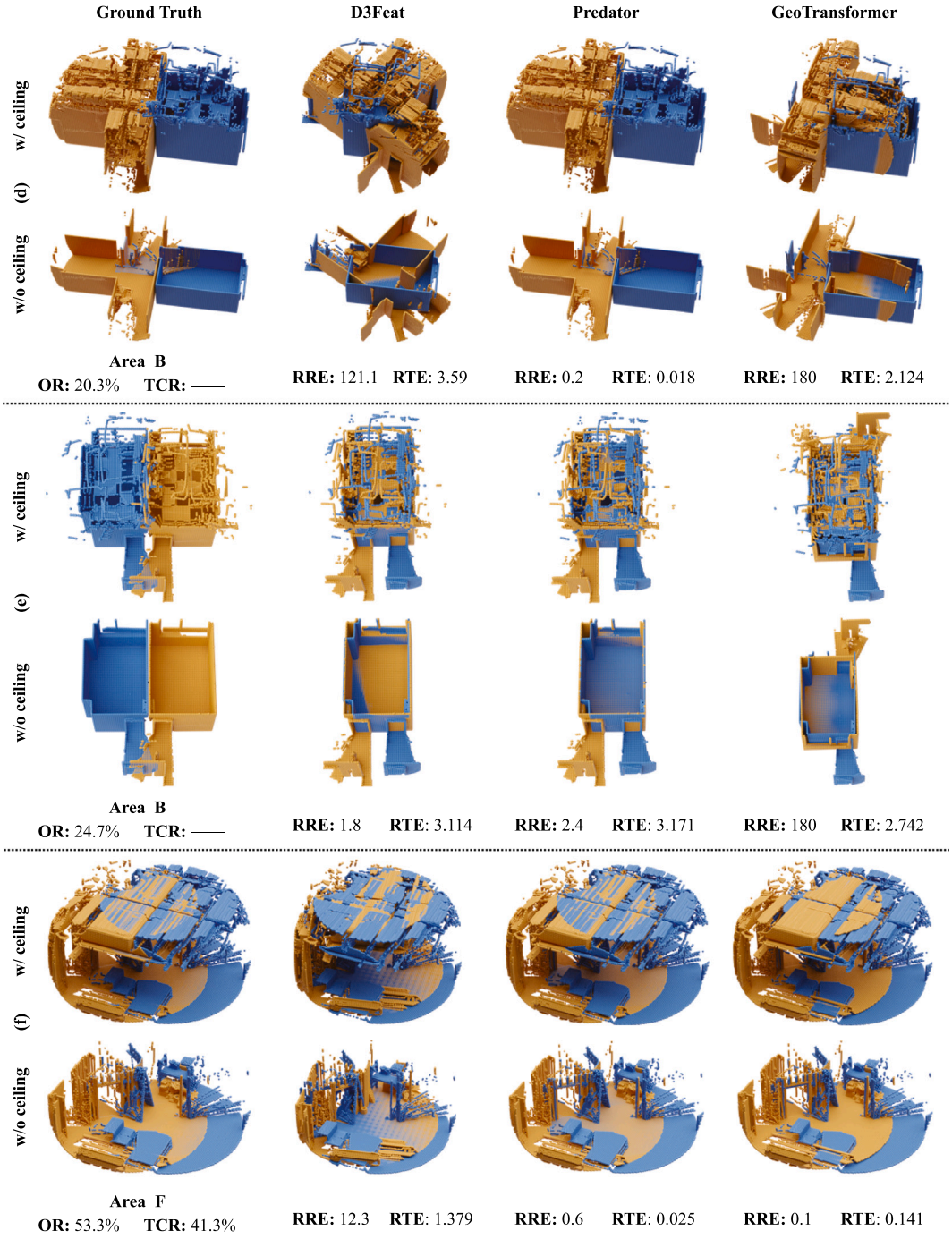


Fig. 20. Additional qualitative results on pairwise registration. Results on a fragment pair are showcased for D3Feat (Bai et al., 2020), Predator (Huang et al., 2021), and GeoTransformer (Qin et al., 2022). The overlap (OR) and temporal change (TCR) ratios are reported per input pair, as well as the translation (TE) and rotation (RE) errors per method (in meters and degrees respectively).

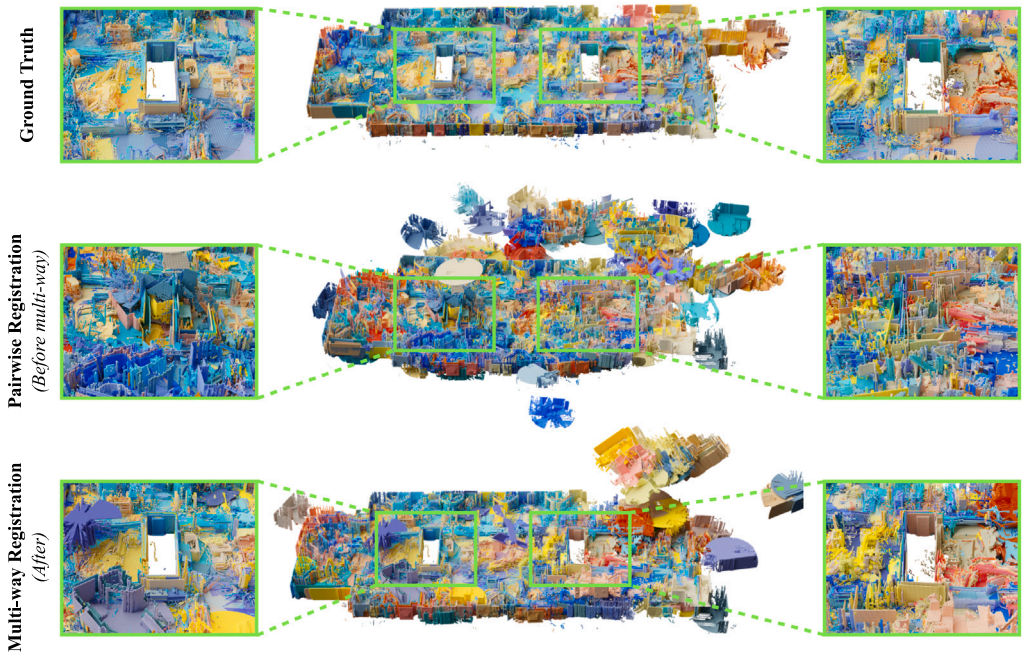


Fig. 21. Spatiotemporal registration after the pairwise (Huang et al., 2021) and multi-way (Choi et al., 2015) tasks for Area F. Different color hues represent different stages. Note the improved alignment in the elevator shafts after multi-way registration.

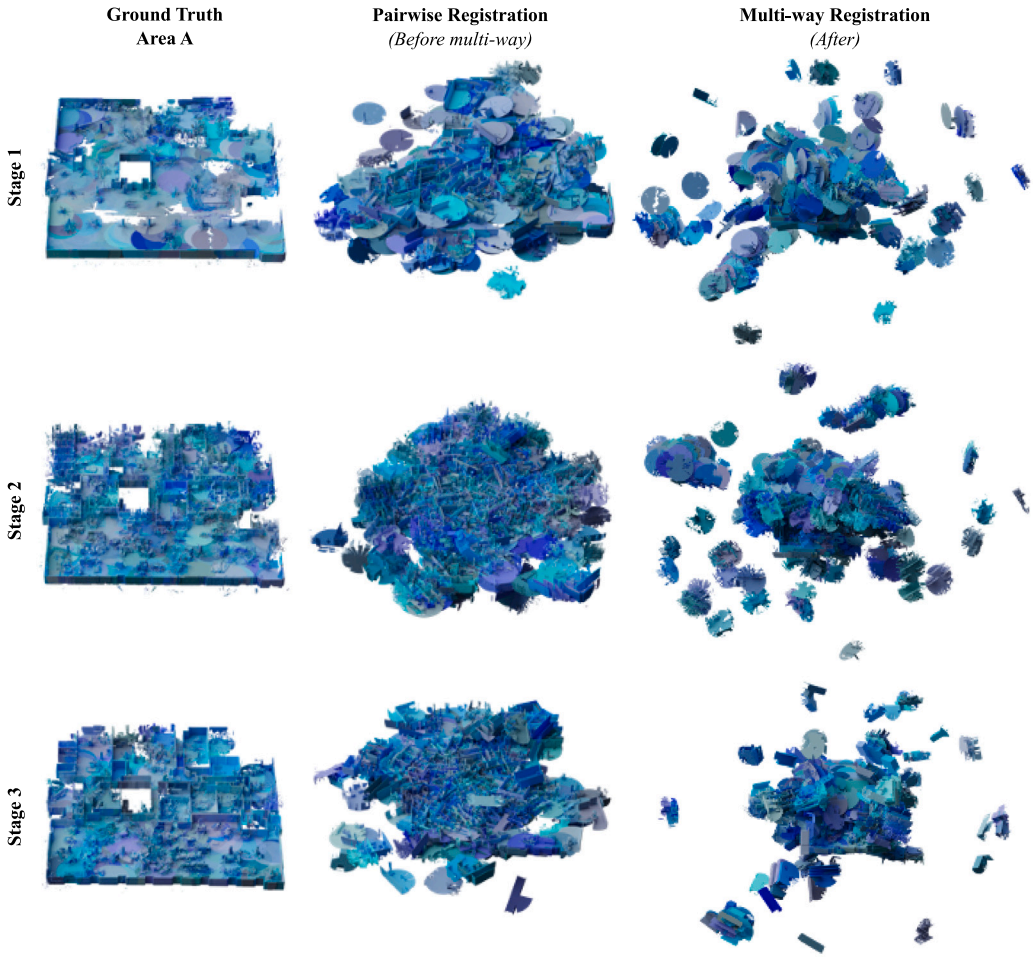


Fig. 22. Spatiotemporal registration results for Area A per temporal stage. Different blue hues denote independent fragment locations. When the pairwise registration from Predator (Huang et al., 2021) fails to recover a rough initial alignment, the multi-way step (Choi et al., 2015) cannot recover it.

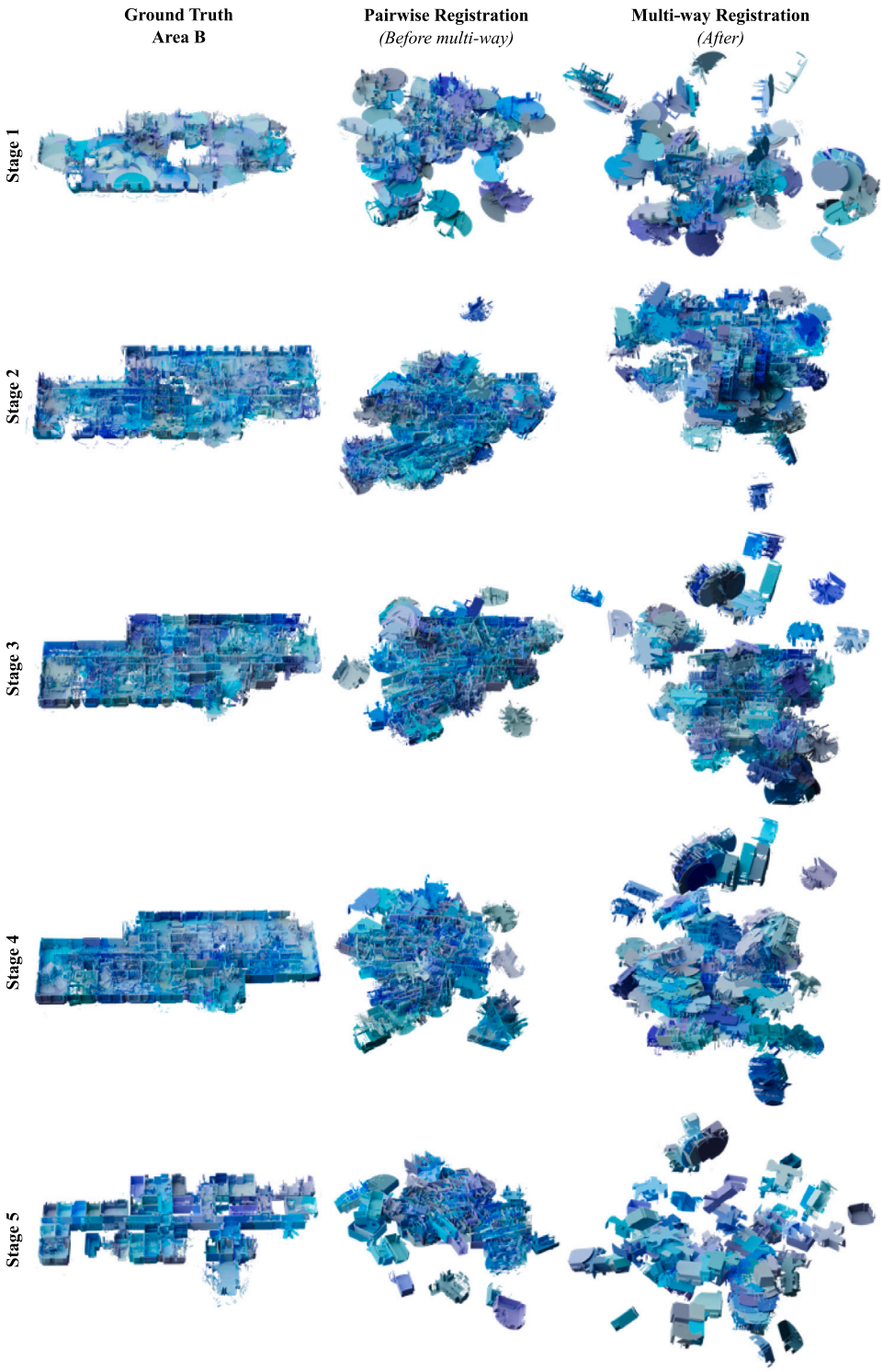
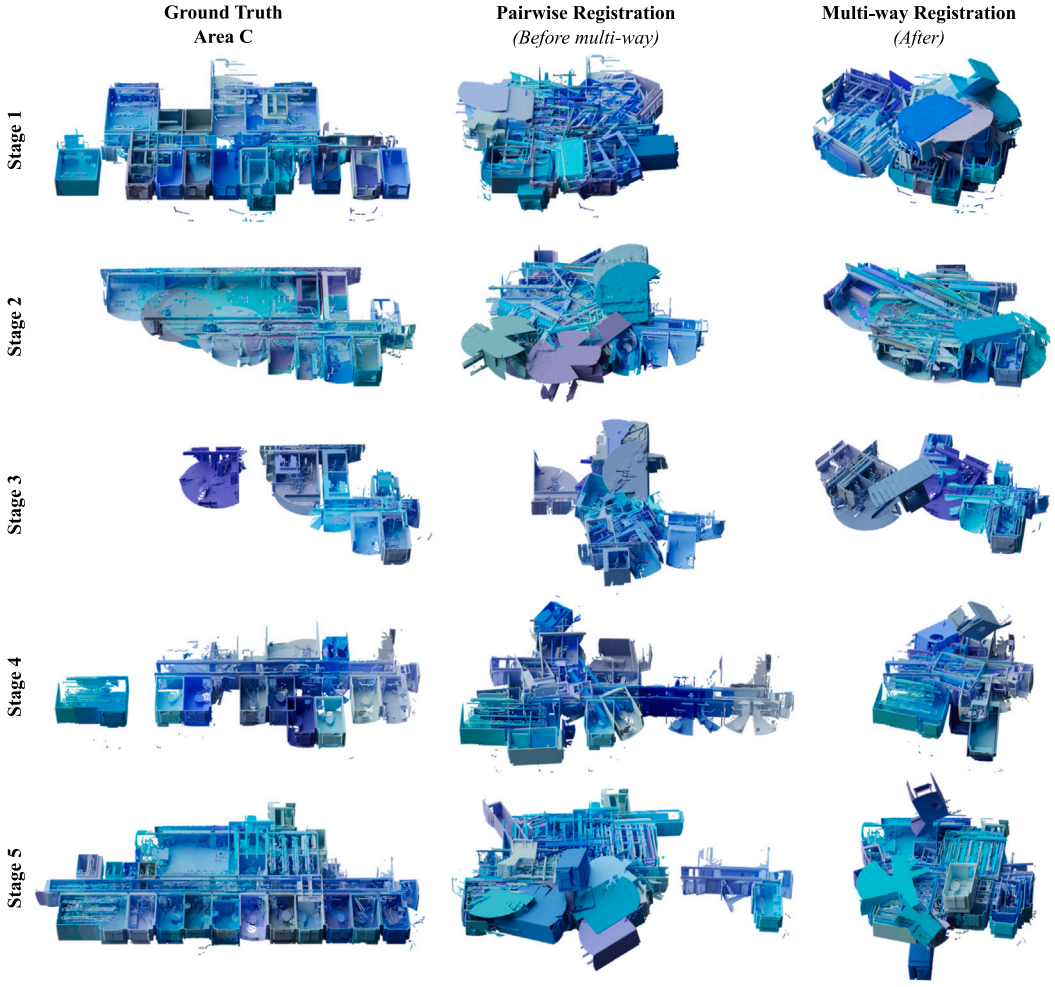
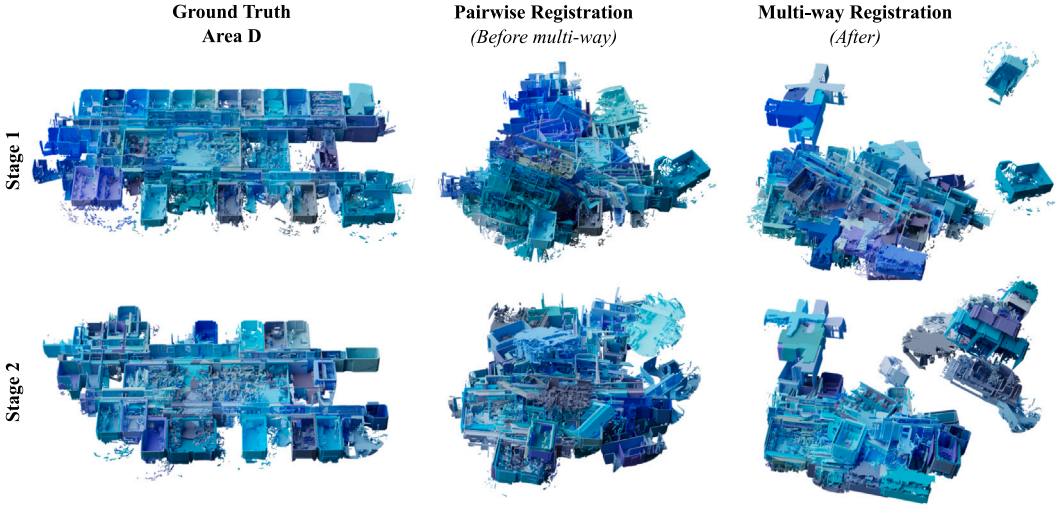


Fig. 23. Spatiotemporal registration results for Area B per temporal stage. Different blue hues denote independent fragment locations. Pairwise registration results from Predator (Huang et al., 2021) and multi-way from Choi et al. (2015).



(a) Spatiotemporal registration results for Area C per temporal stage.



(b) Spatiotemporal registration results for Area D per temporal stage.

Fig. 24. Spatiotemporal registration results for Area C and D per temporal stage. Different blue hues denote independent fragment locations. Pairwise registration results from PREDATOR (Huang et al., 2021) and multi-way from Choi et al. (2015).

Fig. 25. Spatiotemporal registration results for Area E per temporal stage. Different blue hues denote independent fragment locations. Pairwise registration results from PREDATOR (Huang et al., 2021) and multi-way from Choi et al. (2015).

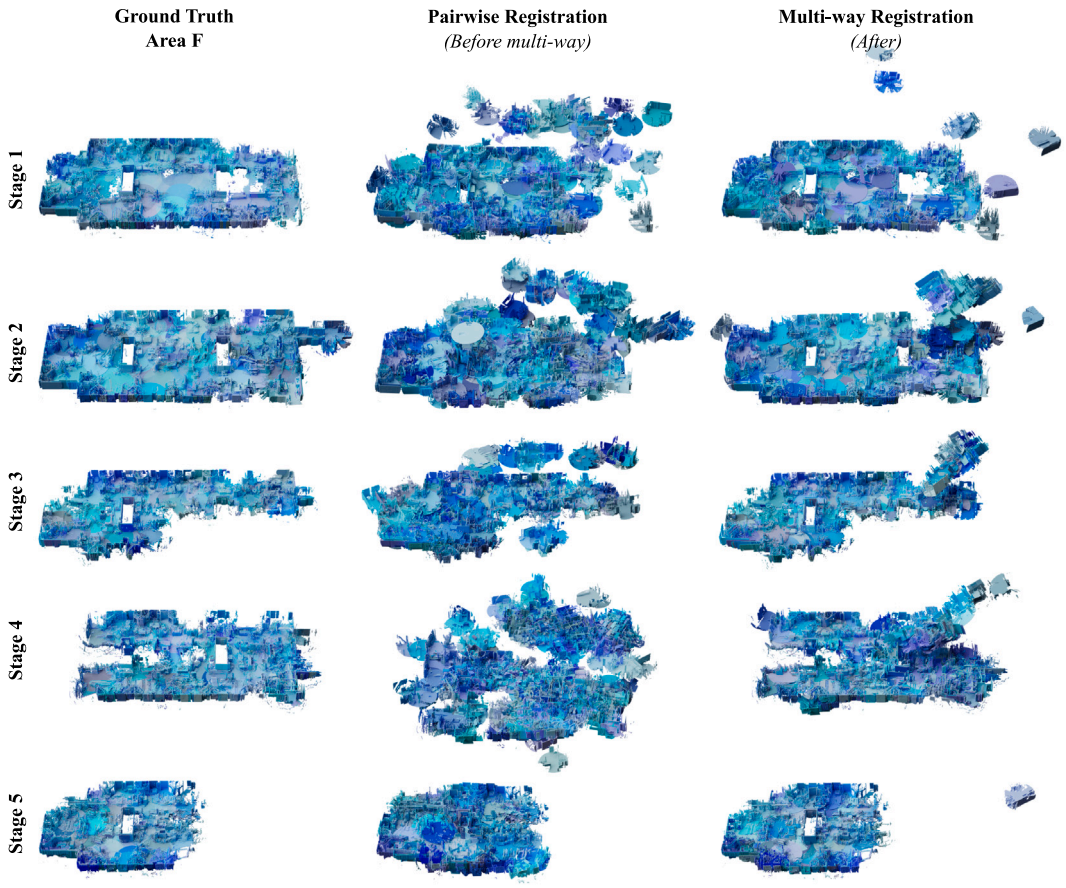


Fig. 26. Spatiotemporal registration results for Area F per temporal stage. Different blue hues denote independent fragment locations. When the pairwise registration results from PREDATOR (Huang et al., 2021) achieve better initialization, the multi-way registration from Choi et al. (2015) can recover an improved alignment.

Table 10
Overlap Classification on **NSS** with PREDATOR. Pairs from the test set are overlapping samples, whereas pairs randomly selected from different locations are non-overlapping ones.

Metrics	Cross-Area	Cross-Stage	Original
All pairs			
mAP [↑]	0.577	**0.774**	0.612
AUROC [↑]	0.582	**0.759**	0.642
Same-stage pairs			
mAP [↑]	0.784	**0.916**	0.833
AUROC [↑]	0.595	**0.786**	0.657
Different-stage pairs			
mAP [↑]	0.318	0.211	**0.410**
AUROC [↑]	0.541	0.575	**0.616**

7. Conclusion

In this study, we introduced a new benchmark called **Nothing Stands Still** for evaluating the performance of 3D point cloud registration in spatiotemporal scenarios. This benchmark assesses methods' capabilities across space, time, and generalization. To support this benchmark, we also presented a novel spatiotemporal dataset containing indoor areas captured over time, exhibiting significant geometric changes. Our findings, as discussed in Section 6, indicate that existing 3D registration methods have limited ability to handle temporal changes effectively. Moreover, the conflicting objectives between pairwise and multi-way registration tasks currently pose challenges in developing end-to-end algorithms. This paper highlights the substantial room for improvement in this field. In addition, both the benchmark and dataset hold great potential for various applications such as in robotic navigation, virtual and augmented reality applications, construction progress monitoring, and learning and detecting change.

CRediT authorship contribution statement

Tao Sun: Writing-original draft, Methodology, Visualization, Validation, Software, Investigation, Formal analysis, Data curation. **Yan Hao:** Visualization, Software, Investigation, Formal analysis, Data curation. **Shengyu Huang:** Writing – review & editing, Writing-original draft, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Silvio Savarese:** Resources, Investigation, Conceptualization. **Konrad Schindler:** Writing – review & editing, Supervision, Resources, Investigation. **Marc Pollefeys:** Writing – review & editing, Supervision, Resources, Investigation. **Iro Armeni:** Writing – review & editing, Writing–original draft, Validation, Supervision, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research was supported by an ETHZurich Postdoctoral Fellowship and the SNSF Adv. Grant 216260: "Beyond Frozen Worlds: Capturing Functional 3D Digital Twins from the Real World."

References

Adam, A., Sattler, T., Karantzalos, K., Pajdla, T., 2022. Objects can move: 3D change detection by geometric transformation consistency. In: ECCV.

Ao, S., Hu, Q., Yang, B., Markham, A., Guo, Y., 2021. SpinNet: Learning a general surface descriptor for 3D point cloud registration. In: CVPR.

Aoki, Y., Goforth, H., Srivatsan, R.A., Lucey, S., 2019. PointnetLK: Robust & efficient point cloud registration using Pointnet. In: CVPR.

Armeni, I., Sax, S., Zamir, A.R., Savarese, S., 2017. Joint 2d-3d-semantic data for indoor scene understanding. arXiv preprint arXiv:1702.01105.

Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S., 2016. 3D semantic parsing of large-scale indoor spaces. In: CVPR.

Arun, K.S., Huang, T.S., Blostein, S.D., 1987. Least-squares fitting of two 3-D point sets. IEEE Trans. Pattern Anal. Mach. Intell..

Attaiki, S., Pai, G., Ovsjanikov, M., 2021. Dpfm: Deep partial functional maps. In: 3DV.

Bai, X., Luo, Z., Zhou, L., Fu, H., Quan, L., Tai, C.-L., 2020. D3Feat: Joint learning of dense detection and description of 3D local features. In: CVPR.

Barron, J.T., Malik, J., 2013. Intrinsic scene properties from a single rgb-d image. In: CVPR. pp. 17–24.

Bernard, F., Thunberg, J., Gemmar, P., Hertel, F., Husch, A., Goncalves, J., 2015. A solution for multi-alignment by transformation synchronisation. In: CVPR.

Besl, P.J., McKay, N.D., 1992. Method for registration of 3-D shapes. In: Sensor Fusion IV: Control Paradigms and Data Structures. Vol. 1611, International Society for Optics and Photonics, pp. 586–606.

Bhattacharya, U., Govindu, V.M., 2019. Efficient and robust registration on the 3d special euclidean group. In: ICCV.

Bigun, J., 1987. Optimal Orientation Detection of Linear Symmetry. Linköping University Electronic Press.

Biswasa, H.K., Boschéa, F., Suna, M., 2015. Planning for scanning using building information models: A novel approach with occlusion handling. In: Symposium on Automation and Robotics in Construction and Mining. ISARC 2015, Vol. 15, p. 18.

Black, M.J., Anandan, P., 1993. A framework for the robust estimation of optical flow. In: ICCV.

2025. Blender.org - Home of the blender project. https://www.blender.org. (Accessed 15 January 2025).

Bourdis, N., Marraud, D., Sahbi, H., 2011. Constrained optical flow for aerial image change detection. In: 2011 IEEE International Geoscience and Remote Sensing Symposium. IEEE, pp. 4176–4179.

Brox, T., Bregler, C., Malik, J., 2009. Large displacement optical flow. In: CVPR.

Cai, Z., Chin, T.-J., Bustos, A.P., Schindler, K., 2019. Practical optimal registration of terrestrial LiDAR scan pairs. ISPRS J. Photogramm. Remote Sens. 147, 118–131.

Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y., 2017. Matterport3d: Learning from rgb-d data in indoor environments. arXiv preprint arXiv:1709.06158.

Chen, X., Li, S., Mersch, B., Wiesmann, L., Gall, J., Behley, J., Stachniss, C., 2021. Moving object segmentation in 3D LiDAR data: A learning-based approach exploiting sequential data. IEEE Robot. Autom. Lett..

Chen, J., Yuan, Z., Peng, J., Chen, L., Huang, H., Zhu, J., Liu, Y., Li, H., 2020. Dasnet: Dual attentive fully convolutional siamese networks for change detection in high-resolution satellite images. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 14, 1194–1206.

Cheng, G., Huang, Y., Li, X., Lyu, S., Xu, Z., Zhao, H., Zhao, Q., Xiang, S., 2024. Change detection methods for remote sensing in the last decade: A comprehensive review. Remote Sens. 16 (13), 2355.

Choi, S., Zhou, Q.-Y., Koltun, V., 2015. Robust reconstruction of indoor scenes. In: CVPR.

Choy, C., Gwak, J., Savarese, S., 2019a. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In: CVPR.

Choy, C., Park, J., Koltun, V., 2019b. Fully convolutional geometric features. In: ICCV.

2025. Cloud compare 3D point cloud and mesh processing software. https://www.danielgm.net/cc/. (Accessed 15 January 2025).

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., 2016. The cityscapes dataset for semantic urban scene understanding. In: CVPR.

Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M., 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: CVPR.

de Gélis, I., Lefèvre, S., Corpetti, T., 2023. Siamese kpconv: 3D multiple change detection from raw point clouds using deep learning. ISPRS J. Photogramm. Remote Sens. 197, 274–291.

Deng, Z., Li, X., Li, X., Tong, Y., Zhao, S., Liu, M., 2024. VG4D: Vision-language model goes 4D video recognition. In: ICRA.

Díaz-Vilariño, L., Frías, E., Balado, J., González-Jorge, H., 2018. Scan planning and route optimization for control of execution of as-designed bim. Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci..

Dong, J., Burnham, J.G., Boots, B., Rains, G., Dellaert, F., 2017. 4D crop monitoring: Spatio-temporal reconstruction for agriculture. In: 2017 IEEE International Conference on Robotics and Automation. ICRA, IEEE, pp. 3878–3885.

Droeschel, D., Schwarz, M., Behnke, S., 2017. Continuous mapping and localization for autonomous navigation in rough terrain using a 3D laser scanner. Robot. Auton. Syst. 88, 104–115.

Fan, H., Yu, X., Ding, Y., Yang, Y., Kankanhalli, M., 2020. PSTNet: Point spatio-temporal convolution on point cloud sequences. In: ICLR.

Fantoni, S., Castellani, U., Fusiello, A., 2012. Accurate and automatic alignment of range surfaces. In: International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission. IEEE.

Fischer, P., Dosovitskiy, A., Ilg, E., Häusser, P., Hazırbas, C., Golkov, V., Van der Smagt, P., Cremers, D., Brox, T., 2015. FlowNet: Learning optical flow with convolutional networks. arXiv preprint arXiv:1504.06852.

Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM.

Frías, E., Díaz-Vilariño, L., Balado, J., Lorenzo, H., 2019. From BIM to scan planning and optimization for construction control. Remote Sens. 11 (17), 1963.

Gehrung, J., Hebel, M., Arens, M., Stilla, U., 2022. Change detection in street environments based on mobile laser scanning: A fuzzy spatial reasoning approach. ISPRS Open J. Photogramm. Remote Sens. 5, 100019.

Geiger, A., Lenz, P., Stiller, C., Urtasun, R., 2013. Vision meets robotics: The kitti dataset. Int. J. Robot. Res..

Gojcic, Z., Zhou, C., Wegner, J.D., Guibas, L.J., Birdal, T., 2020. Learning multiview 3d point cloud registration. In: CVPR.

Gojcic, Z., Zhou, C., Wegner, J.D., Wieser, A., 2019. The perfect match: 3d point cloud matching with smoothed densities. In: CVPR.

Golparvar-Fard, M., Peña-Mora, F., Savarese, S., 2009. D4AR–a 4-dimensional augmented reality model for automating construction progress monitoring data collection, processing and communication. J. Inf. Technol. Constr. 14 (13), 129–153.

Griffith, S., Dellaert, F., Pradalier, C., 2020. Transforming multiple visual surveys of a natural environment into time-lapses. Int. J. Robot. Res. 39 (1), 100–126.

Gschwandtner, M., Kwitt, R., Uhl, A., Pree, W., 2011. BlenSor: Blender sensor simulation toolbox. In: International Symposium on Visual Computing. Springer, pp. 199–208.

Halber, M., Shi, Y., Xu, K., Funkhouser, T., 2019. Rescan: Inductive instance segmentation for indoor RGBD scans. In: ICCV.

Handa, A., Newcombe, R.A., Angeli, A., Davison, A.J., 2012. Real-time camera tracking: When is high frame-rate best? In: ECCV. Springer, pp. 222–235.

Handa, A., Patraucean, V., Badrinarayanan, V., Stent, S., Cipolla, R., 2015. SceneNet: understanding real world indoor scenes with synthetic data. arxiv preprint (2015). arXiv preprint arXiv:1511.07041.

Handa, A., Whelan, T., McDonald, J., Davison, A.J., 2014. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In: ICRA. IEEE, pp. 1524–1531.

Haresh, S., Kumar, S., Coskun, H., Syed, S.N., Konin, A., Zia, Z., Tran, Q.-H., 2021. Learning by aligning videos in time. In: CVPR. pp. 5548–5558.

Hartley, R.I., Kahl, F., 2009. Global optimization through rotation space search. Int. J. Comput. Vis. 82 (1), 64–79.

Hermans, A., Beyer, L., Leibe, B., 2017. In defense of the triplet loss for person re-identification. arXiv preprint arXiv:1703.07737.

Hernandez-Juarez, D., Schneider, L., Espinosa, A., Vazquez, D., Lopez, A.M., Franke, U., Pollefeys, M., Moure, J.C., 2017. Slanted stixels: Representing san francisco's steepest streets. In: BMVC.

Horn, B.K., Schunck, B.G., 1981. Determining optical flow. Artificial Intelligence.

Hu, Y.-T., Wang, J., Yeh, R.A., Schwing, A.G., 2021. Sail-vos 3D: A synthetic dataset and baselines for object detection and 3D mesh reconstruction from video data. In: CVPR. pp. 1418–1428.

Hua, B.-S., Pham, Q.-H., Nguyen, D.T., Tran, M.-K., Yu, L.-F., Yeung, S.-K., 2016. Scenenn: A scene meshes dataset with annotations. In: 3DV.

Huang, J., Birdal, T., Gojcic, Z., Guibas, L.J., Hu, S.-M., 2022a. Multiway non-rigid point cloud registration via learned functional map synchronization. IEEE Trans. Pattern Anal. Mach. Intell..

Huang, S., Gojcic, Z., Huang, J., Wieser, A., Schindler, K., 2022b. Dynamic 3D scene analysis by point cloud accumulation. In: ECCV.

Huang, S., Gojcic, Z., Usvyatsov, M., Wieser, A., Schindler, K., 2021. PREDATOR: Registration of 3D point clouds with low overlap. In: CVPR.

Huang, X., Liang, Z., Zhou, X., Xie, Y., Guibas, L.J., Huang, Q., 2019. Learning transformation synchronization. In: CVPR.

Huang, R., Xu, Y., Hoegner, L., Stilla, U., 2022c. Semantics-aided 3D change detection on construction sites using UAV-based photogrammetric point clouds. Autom. Constr. 134, 104057.

Huber, D.F., Hebert, M., 2003. Fully automatic registration of multiple 3D data sets. Image Vis. Comput. 21 (7), 637–650.

Hui, T.-W., Tang, X., Loy, C.C., 2018. LiteFlowNet: A lightweight convolutional neural network for optical flow estimation. In: CVPR.

Hussain, M., Chen, D., Cheng, A., Wei, H., Stanley, D., 2013. Change detection from remotely sensed images: From pixel-based to object-based approaches. ISPRS J. photogramm. Remote Sens..

Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T., 2017. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In: CVPR.

Johnson, A., Hebert, M., 1999. Using spin images for efficient object recognition in cluttered 3D scenes. IEEE Trans. Pattern Anal. Mach. Intell..

Kharroubi, A., Poux, F., Ballouch, Z., Hajji, R., Billen, R., 2022. Three dimensional change detection using point clouds: A review. Geomatics 2 (4), 457–485.
Kundu, A., Yin, X., Fathi, A., Ross, D., Brewington, B., Funkhouser, T., Pantofaru, C., 2020. Virtual multi-view fusion for 3d semantic segmentation. In: ECCV. Springer, pp. 518–535.
Kwon, T., Tekin, B., Tang, S., Pollefeys, M., 2022. Context-aware sequence alignment using 4D skeletal augmentation. In: CVPR. pp. 8172–8182.
Lei, Y., Peng, D., Zhang, P., Ke, Q., Li, H., 2020. Hierarchical paired channel fusion network for street scene change detection. IEEE Trans. Image Process. 30, 55–67.
Li, H., Hartley, R., 2007. The 3D-3D registration problem revisited. In: 2007 IEEE 11th International Conference on Computer Vision. IEEE, pp. 1–8.
Li, J., Tang, P., Wu, Y., Pan, M., Tang, Z., Hui, G., 2022. Scene change detection: semantic and depth information. Multimedia Tools Appl..
Lin, H., Wang, Q., Cai, R., Peng, S., Averbuch-Elor, H., Zhou, X., Snavely, N., 2023. Neural scene chronology. In: CVPR.
Liu, X., Qi, C.R., Guibas, L.J., 2019a. FlowNet3D: Learning scene flow in 3D point clouds. In: CVPR.
Liu, X., Yan, M., Bohg, J., 2019b. Meteornet: Deep learning on dynamic 3d point cloud sequences. In: ICCV.
López-Armenta, M.F., Nespeca, R., 2024. 3D change detection for cultural heritage monitoring: Two case studies of underground sculptural reliefs. Digit. Appl. Archaeol. Cult. Herit. 33, e00328.
Love, P.E., Holt, G.D., Shen, L.Y., Li, H., Irani, Z., 2002. Using systems dynamics to better understand change and rework in construction project management systems. Int. J. Proj. Manage. 20 (6), 425–436.
Ma, M., Tam, V.W., Le, K.N., Li, W., 2020. Challenges in current construction and demolition waste recycling: A China study. Waste Manage. 118, 610–625.
Martin-Brualla, R., Gallup, D., Seitz, S.M., 2015. 3D time-lapse reconstruction from internet photos. In: ICCV.
Martin-Brualla, R., Radwan, N., Sajjadi, M.S., Barron, J.T., Dosovitskiy, A., Duckworth, D., 2021. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In: CVPR.
2025. Matterport. https://matterport.com/. (Accessed 15 January 2025).
Matzen, K., Snavely, N., 2014. Scene chronology. In: ECCV.
McCormac, J., Handa, A., Leutenegger, S., Davison, A.J., 2017. Scenenet rgb-d: Can 5 m synthetic images beat generic imagenet pre-training on indoor segmentation? In: ICCV. pp. 2678–2687.
Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K., 2016. Mot16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831.
Munaro, M.R., Tavares, S.F., Bragança, L., 2020. Towards circular and more sustainable buildings: A systematic literature review on the circular economy in the built environment. J. Clean. Prod. 260, 121134.
Noichl, F., Braun, A., Borrmann, A., 2021. "BIM-to-scan" for scan-to-BIM: Generating realistic synthetic ground truth point clouds based on industrial 3D models. In: Proceedings of the 2021 European Conference on Computing in Construction.
Park, J.-M., Jang, J.-H., Yoo, S.-M., Lee, S.-K., Kim, U.-H., Kim, J.-H., 2021. ChangeSim: towards end-to-end online scene change detection in industrial indoor environments. In: IROS.
Peng, D., Zhang, Y., Guan, H., 2019. End-to-end change detection for high resolution satellite images using improved unet++. Remote Sens. 11 (11), 1382.
Pollard, T., Mundy, J.L., 2007. Change detection in a 3-d world. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition. Ieee, pp. 1–6.
Prabhakar, K.R., Ramaswamy, A., Bhambri, S., Gubbi, J., Babu, R.V., Purushothaman, B., 2020. Cdnet++: Improved change detection with deep neural network feature correlation. In: 2020 International Joint Conference on Neural Networks. IJCNN.
2025. Primesense sensor. http://xtionprolive.com/primesense-carmine-1.09. (Accessed 15 January 2025).
Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F., 2021. D-nerf: Neural radiance fields for dynamic scenes. In: CVPR.
Purushwalkam, S., Ye, T., Gupta, S., Gupta, A., 2020. Aligning videos in space and time. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16. Springer, pp. 262–278.
Puy, G., Boulch, A., Marlet, R., 2020. Flot: Scene flow on point clouds guided by optimal transport. In: ECCV.
Qi, C.R., Zhou, Y., Najibi, M., Sun, P., Vo, K., Deng, B., Anguelov, D., 2021. Offboard 3D object detection from point cloud sequences. In: CVPR.
Qin, R., Tian, J., Reinartz, P., 2016. 3D change detection–approaches and applications. ISPRS J. Photogramm. Remote Sens. 122, 41–56.
Qin, Z., Yu, H., Wang, C., Guo, Y., Peng, Y., Xu, K., 2022. Geometric transformer for fast and robust point cloud registration. In: CVPR.
Qiu, Y., Satoh, Y., Suzuki, R., Iwata, K., Kataoka, H., 2020. Indoor scene change captioning based on multimodality data. Sensors (Switzerland).
Qiu, W., Yuille, A., 2016. Unrealcv: Connecting computer vision to unreal engine. In: ECCV. Springer, pp. 909–916.
Rempe, D., Birdal, T., Zhao, Y., Gojcic, Z., Sridhar, S., Guibas, L.J., 2020. Caspr: Learning canonical spatiotemporal point cloud representations. In: NeurIPS.
Richter, S.R., Vineet, V., Roth, S., Koltun, V., 2016. Playing for data: Ground truth from computer games. In: ECCV. Springer, pp. 102–118.
Roberto de Souza, C., Gaidon, A., Cabon, Y., Manuel Lopez, A., 2017. Procedural generation of videos to train deep action recognition networks. In: CVPR. pp. 4757–4767.
Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M., 2016. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: CVPR.
Ru, L., Du, B., Wu, C., 2020. Multi-temporal scene classification and scene change detection with correlation based fusion. IEEE Trans. Image Process. 30, 1382–1394.
Rusu, R.B., Blodow, N., Beetz, M., 2009. Fast point feature histograms (FPFH) for 3D registration. In: ICRA.
Rusu, R.B., Blodow, N., Marton, Z.C., Beetz, M., 2008. Aligning point cloud views using persistent feature histograms. In: IROS.
Sakurada, K., Shibuya, M., Wang, W., 2020. Weakly supervised silhouette-based semantic scene change detection. In: ICRA.
Sarlin, P.-E., Dusmanu, M., Schönberger, J.L., Speciale, P., Gruber, L., Larsson, V., Miksik, O., Pollefeys, M., 2022. LaMAR: Benchmarking localization and mapping for augmented reality. In: ECCV.
Schindler, G., Dellaert, F., 2010. Probabilistic temporal inference on reconstructed 3d scenes. In: CVPR.
Schindler, G., Dellaert, F., Kang, S.B., 2007. Inferring temporal order of images from 3D structure. In: CVPR.
Shafaei, A., Little, J.J., Schmidt, M., 2016. Play and learn: Using video games to train computer vision models. arXiv preprint arXiv:1608.01745.
Simonyan, K., Zisserman, A., 2014. Two-stream convolutional networks for action recognition in videos. In: NeurIPS.
Song, S., Yu, F., Zeng, A., Chang, A.X., Savva, M., Funkhouser, T., 2017. Semantic scene completion from a single depth image. In: CVPR. pp. 1746–1754.
Stefanini, E., Ciancolini, E., Settimi, A., Pallottino, L., 2023. Safe and robust map updating for long-term operations in dynamic environments. Sensors 23 (13), http://dx.doi.org/10.3390/s23136066, URL https://www.mdpi.com/1424-8220/23/13/6066.
Stilla, U., Xu, Y., 2023. Change detection of urban objects using 3D point clouds: A review. ISPRS J. Photogramm. Remote Sens. 197, 228–255.
Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J.J., Mur-Artal, R., Ren, C., Verma, S., et al., 2019. The Replica dataset: A digital replica of indoor spaces. arXiv preprint arXiv:1906.05797.
Sun, Y., Cheng, C., Zhang, Y., Zhang, C., Zheng, L., Wang, Z., Wei, Y., 2020. Circle loss: A unified perspective of pair similarity optimization. In: CVPR.
Sun, D., Roth, S., Black, M.J., 2010. Secrets of optical flow estimation and their principles. In: CVPR.
Teed, Z., Deng, J., 2020. Raft: Recurrent all-pairs field transforms for optical flow. In: ECCV.
Theiler, P.W., Wegner, J.D., Schindler, K., 2014. Keypoint-based 4-points congruent sets – automated marker-less registration of laser scans. ISPRS J. Photogramm. Remote Sens. 96, 149–163. http://dx.doi.org/10.1016/j.isprsjprs.2014.06.015, URL https://www.sciencedirect.com/science/article/pii/S0924271614001701.
Theiler, P.W., Wegner, J.D., Schindler, K., 2015. Globally consistent registration of terrestrial laser scans via graph optimization. ISPRS J. Photogramm. Remote Sens. 109, 126–138.
Tombari, F., Salti, S., Di Stefano, L., 2010a. Unique shape context for 3D data description. In: ACM Workshop on 3D Object Retrieval.
Tombari, F., Salti, S., Di Stefano, L., 2010b. Unique signatures of histograms for local surface description. In: ECCV.
Torsello, A., Rodola, E., Albarelli, A., 2011. Multiview registration via graph diffusion of dual quaternions. In: CVPR.
Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W., 2000. Bundle adjustment—a modern synthesis. In: Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings. Springer, pp. 298–372.
Tsamis, G., Kostavelis, I., Giakoumis, D., Tzovaras, D., 2021. Towards life-long mapping of dynamic environments using temporal persistence modeling. In: 2020 25th International Conference on Pattern Recognition. ICPR, pp. 10480–10485. http://dx.doi.org/10.1109/ICPR48806.2021.9413161.
Vedula, S., Baker, S., Rander, P., Collins, R., Kanade, T., 1999. Three-dimensional scene flow. In: ICCV.
Vogel, C., Schindler, K., Roth, S., 2011. 3D scene flow estimation with a rigid motion prior. In: ICCV.
Vogel, C., Schindler, K., Roth, S., 2013. Piecewise rigid scene flow. In: ICCV.
Vogel, C., Schindler, K., Roth, S., 2015. 3D scene flow estimation with a piecewise rigid scene model. Int. J. Comput. Vis..
Wald, J., Avetisyan, A., Navab, N., Tombari, F., Niessner, M., 2019. RIO: 3D object instance re-localization in changing indoor environments. In: ICCV.
Wald, J., Sattler, T., Golodetz, S., Cavallari, T., Tombari, F., 2020. Beyond controlled environments: 3d camera re-localization in changing indoor scenes. In: ECCV.
Wang, H., Liu, Y., Dong, Z., Guo, Y., Liu, Y.-S., Wang, W., Yang, B., 2023. Robust multiview point cloud registration with reliable pose graph initialization and history reweighting. In: CVPR.
Wang, B., Liu, Z., Li, Q., Prorok, A., 2020a. Mobile robot path planning in dynamic environments through globally guided reinforcement learning. IEEE Robot. Autom. Lett. 5 (4), 6932–6939.

Wang, X., Mizukami, Y., Tada, M., Matsuno, F., 2021a. Navigation of a mobile robot in a dynamic environment using a point cloud map. Artif. Life Robot. 26, 10–20.
Wang, Y., Solomon, J.M., 2019a. Deep closest point: Learning representations for point cloud registration. In: ICCV.
Wang, Y., Solomon, J.M., 2019b. Prnet: Self-supervised learning for partial-to-partial registration. Adv. Neural Inf. Process. Syst. 32.
Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L., 2019. Temporal segment networks for action recognition in videos. IEEE Trans. Pattern Anal. Mach. Intell..
Wang, Z., Zhang, Y., Luo, L., Wang, N., 2021b. Transcd: scene change detection via transformer-based architecture. Opt. Express.
Wang, W., Zhu, D., Wang, X., Hu, Y., Qiu, Y., Wang, C., Hu, Y., Kapoor, A., Scherer, S., 2020b. Tartanair: A dataset to push the limits of visual slam. In: IROS. IEEE, pp. 4909–4916.
Wei, T., Patel, Y., Shekhovtsov, A., Matas, J., Barath, D., 2023. Generalized differentiable RANSAC. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. ICCV, pp. 17649–17660.
Weinmann, M., Jutzi, B., Mallet, C., 2013. Feature relevance assessment for the semantic interpretation of 3D point cloud data. ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci. 2, 313–318.
Wenzel, P., Wang, R., Yang, N., Cheng, Q., Khan, Q., von Stumberg, L., Zeller, N., Cremers, D., 2020. 4Seasons: A cross-season dataset for multi-weather SLAM in autonomous driving. In: GCPR.
Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J., 2015. 3D shapenets: A deep representation for volumetric shapes. In: CVPR.
Xiang, Y., Alahi, A., Savarese, S., 2015. Learning to track: Online multi-object tracking by decision making. In: ICCV.
Xiao, W., Vallet, B., Brédif, M., Paparoditis, N., 2015. Street environment change detection from mobile laser scanning point clouds. ISPRS J. Photogramm. Remote Sens. 107, 38–49.
Yew, Z.J., Lee, G.H., 2020. RPM-Net: Robust point matching using learned features. In: CVPR.
Yew, Z.J., Lee, G.H., 2021a. City-scale scene change detection using point clouds. In: ICRA.
Yew, Z.J., Lee, G.H., 2021b. Learning iterative robust transformation synchronization. In: 3DV.
Yu, H., Li, F., Saleh, M., Busam, B., Ilic, S., 2021. CoFiNet: Reliable coarse-to-fine correspondences for robust PointCloud registration. In: NeurIPS. Vol. 34.
Zach, C., Bourmaud, G., 2018. Descending, lifting or smoothing: Secrets of robust cost optimization. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 547–562.
Zeng, A., Song, S., Nießner, M., Fisher, M., Xiao, J., Funkhouser, T., 2017. 3DMatch: learning local geometric descriptors from RGB-D reconstructions. In: CVPR.
Zhang, Y., Song, S., Yumer, E., Savva, M., Lee, J.-Y., Jin, H., Funkhouser, T., 2017. Physically-based rendering for indoor scene understanding using convolutional neural networks. arXiv preprint arXiv:1612.07429.
Zhang, J., Sun, D., Luo, Z., Yao, A., Zhou, L., Shen, T., Chen, Y., Quan, L., Liao, H., 2019. Learning two-view correspondences and geometry using order-aware network. In: CVPR.
Zhang, L., Yang, A.J., Xiong, Y., Casas, S., Yang, B., Ren, M., Urtasun, R., 2023. Towards unsupervised object detection from LiDAR point clouds. In: CVPR.
Zheng, Z., Ma, A., Zhang, L., Zhong, Y., 2021. Change is everywhere: Single-temporal supervised object change detection in remote sensing imagery. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15193–15202.
Zhou, Q.-Y., Park, J., Koltun, V., 2016. Fast global registration. In: ECCV. Springer.
Zhu, L., Huang, S., Schindler, K., Armeni, I., 2024. Living scenes: Multi-object relocalization and reconstruction in changing 3D environments. In: The IEEE Conference on Computer Vision and Pattern Recognition. CVPR.
Zolfaghari Bengar, J., Gonzalez-Garcia, A., Villalonga, G., Raducanu, B., Aghdam, H.H., Mozerov, M., Lopez, A.M., van de Weijer, J., 2019. Temporal coherence for active learning in videos. In: ICCV Workshops.