Living Scenes: Multi-object Relocalization and Reconstruction in Changing 3D Environments

Liyuan Zhu[1] Shengyu Huang[2] Konrad Schindler[2] Iro Armeni[1]
[1]Stanford University [2]ETH Zurich

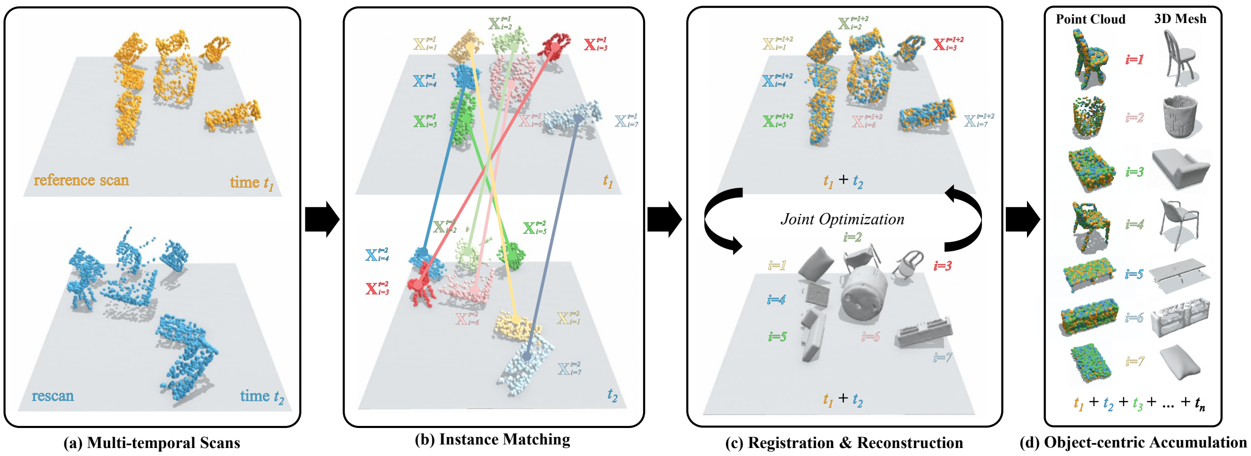

Figure 1. **Living Scenes.** A *living scene* is a 3D environment with multiple moving objects that evolves over time. *(a)* Two temporal observations (scans) represent the scene at times (t_1, t_2) and capture the objects having moved around. To understand the change in the scene, given instance segmentation, we *(b)* match object point clouds from t_1 and t_2 that belong to the same instance; *(c)* register and reconstruct the matches through our joint optimization, *(d)* accumulate all point clouds per instance from the multiple temporal scans, improving the registration and reconstruction quality over time. We illustrate on two scans for simplicity.

Abstract

Research into dynamic 3D scene understanding has primarily focused on short-term change tracking from dense observations, while little attention has been paid to long-term changes with sparse observations. We address this gap with **MoRE**^{2}*, a novel approach for multi-object relocalization and reconstruction in evolving environments. We view these environments as "living scenes" and consider the problem of transforming scans taken at different points in time into a 3D reconstruction of the object instances, whose accuracy and completeness increase over time. At the core of our method lies an SE(3)-equivariant representation in a single encoder-decoder network, trained on synthetic data. This representation enables us to seamlessly tackle instance matching, registration, and reconstruction. We also introduce a joint optimization algorithm that facilitates the accumulation of point clouds originating from the same instance across multiple scans taken at different points in time. We validate our method on synthetic and real-world data and demonstrate state-of-the-art performance in both end-to-end performance and individual subtasks. [project]*

1. Introduction

3D scene reconstruction serves as the foundation for numerous applications of computer vision and robotics, such as mixed reality, navigation, and embodied perception. Many of these applications involve the repeated execution of similar tasks, such as cleaning or searching objects, within a given environment. Consequently, they would benefit from an integrated understanding of the environment, accumulated over multiple 3D scans acquired at different points in time. Such a cumulative scene understanding can enhance interaction with scene objects by progressively improving their geometric completeness and accuracy over time – especially when previously unseen parts are unveiled – and could help to develop a foundational understanding of objects' relocation within the environment. Progressively acquiring a cumulative scene understanding, characterized by increasing geometric completeness and accuracy of constituent objects, can be framed as multi-object relocalization and reconstruction. In this context, object relocalization refers to estimating the 6DoF motion that an object has undergone between two scans, relative to the scene background. This resonates with efforts in dynamic scene understanding. The bulk of previous research focused on real-time observation of dynamic scenes [22, 31, 65], where object relocalization boils down to tracking [4, 14, 31, 37]. Fewer works address the long-term dynamics of 3D scenes [1, 56, 62], where sensor data cannot be captured constantly but rather at irregular points in time. Due to the long gaps between captures, modeling the objects' intermediate motions is infeasible. Alternative methods [62] solve relocalization via point or object matching, followed by object-wise registration [21, 50]. Jointly tackling relocalization and reconstruction [13, 36, 41] over time has been largely overlooked. As we will show in Sec. 4, using separate methods for each task tends to yield increased errors.

To bridge this gap, we introduce MoRE^2, a method for **m**ulti-**o**bject **re**localization and **re**construction of evolving environments over long time spans and from sparse observations, aiming to create a *living scene*. This draws inspiration from the concept of living building information models[1] [29], which treats buildings as living organisms and aspires to maintain their digital twins throughout their lifespan. As shown in Fig. 1(a), our approach takes as input multiple 3D point clouds, acquired at different times and segmented into instances. It addresses the creation of a *living scene* by solving three connected tasks, namely: matching (Fig. 1(b)), registering (Fig. 1(c) top), and reconstructing (Fig. 1(c) bottom) all instances. With a **single encoder-decoder network** trained on synthetic data, our method is able to solve these tasks for real-world scans. At its heart lies an SE(3)-equivariant representation. Additionally, we introduce an optimization scheme that facilitates the accumulation of point clouds originating from the same instance but different scans (Fig. 1(c)). We evaluate MoRE^2 on two long-term living scene datasets, a synthetic one that we generate using 3D object models from ShapeNet [7] and the real-world 3RScan [62], and achieve state-of-the-art performance for both the end-to-end system and each of its subtasks. Our contributions are:

1. A new object-centric formulation of parsing an evolving 3D indoor environment as a *living scene*. It involves instance matching, relocalization, and reconstruction.
2. A novel compact object representation that simultaneously tackles all three tasks. It can generalize to real scenes while being trained on synthetic data only.
3. A joint optimization algorithm that progressively improves the performance of the point cloud registration and reconstruction as more data are accumulated.

2. Related Work

Dynamic point cloud understanding. Dynamic scenes mainly consist of multiple moving instances and a static background. Modeling such complex environments usually starts with estimating the per-point motion – *i.e.*, scene flows [32, 33, 61, 68] – in the scene. Nevertheless, conventional scene flow estimation methods are instance-agnostic, lacking high-level scene representations for downstream tasks related to moving agents. Other methods [3, 12, 20, 22, 54] further disentangle the segmentation and motion estimation for multi-body scenes and object articulation. By combining detection with motion models, 3D multi-object tracking methods [40, 66, 67, 76] directly obtain object instance motions. This line of work relies on regular and constant observations at a high frequency in a short time horizon, such as those in self-driving car datasets [6, 16, 55]. To study long-term changing indoor environments, researchers captured 3D datasets (3RScan [62], ReScan [19], and NSS [56]), where changes between observations are more drastic, making tracking-based methods [31, 53, 76] no longer applicable. To understand long-term changes, Adam *et al*. [1] develop a 3D change detection method using geometric transformation consistency, however they do not have a notion of instances and/or semantics. Halber *et al*. [19] build a spatiotemporal model for temporal point clouds to improve instance segmentation, however they do not tackle the task of relocalization or reconstruction. Wald *et al*. [62] focus on the task of object instance relocalization in two temporal point clouds. They introduce a triplet network for local feature matching, aiming to identify 3D keypoint correspondences between instance patches from two observations. In contrast, our approach goes beyond local matching and infers 6DoF transformations at the entire instance level. It additionally performs reconstruction, which no existing method designed for indoor long-term changing environments addresses.

$\mathrm{SO}(3)$-equivariant networks. Randomly oriented point clouds make coordinate-based networks suffer from inconsistent predictions and poor performance. Such negative effects can be partially alleviated by heavy data augmentation during training [45]. Thus, preserving rotations, *i.e.*, $\mathrm{SO}(3)$-equivariance, in point cloud processing networks becomes a desired property. Thomas *et al*. [58] apply spherical harmonics to constrain the network to achieve $\mathrm{SE}(3)$ equivariance. $\mathrm{SO}(3)$-transformers [15] introduce equivariance to the self-attention mechanism [60] and significantly improve the efficiency of [58]. Deng *et al*. develop Vector Neurons (VNN) [11], a general framework to make MLP-based networks $\mathrm{SO}(3)$-equivariant by vectorizing scalar neurons. GraphOnet [8] extends vector neurons to $\mathrm{SE}(3)$ equivariance. Assaad *et al*. [2] apply rotation equivariant attention for vector neurons. EFEM [28] uses VNN to store shape priors and performs unsupervised object segmentation through expectation maximization. Deng *et al*. [12] develop a Banach-fixed point network with inter-part equivariance for object articulation and multi-object segmentation. Yang *et al*. [70] extend VNN to policy learning and non-

[1] A building information model is a digital representation of a building in terms of semantically meaningful building parts, including their geometry, attributes, and relations.

rigid object manipulation in robotics. We utilize VNNs to comprehend the dynamics of multiple objects.

Point cloud registration. Aligning posed point clouds is essential for 3D perception and mapping. Several hand-crafted 3D feature descriptors have been developed for local feature matching, such as FPFH [49] and SHOT [59]. Following advancements in deep learning, 3DMatch [75], PerfectMatch [17], and RPMNet [71] focus on learning-based descriptors. Predator [21] introduces attention mechanisms [60] into finding 3D correspondences, particularly in low-overlapping regions. In [47, 72], the use of transformer architectures is refined for superpoint matching. [25, 50, 74] leverage additional prior information like surface curvature, 2D image overlap, and scene structure. Apart from the correspondence matching methods above, another line of work is to learn equivariant representations to solve the relative pose. Wang *et al.* [63] develop rotation equivariant descriptors using group equivariant learning [10]. Yu *et al.* [73] develop a rotation invariant transformer method to cope with pose variations in point cloud matching. Zhu *et al.* [77] directly solve the pairwise rotation using rotation equivariant [11] embeddings. Our method also uses equivariant representation but additionally leverages neural implicit surfaces to align two point clouds using test-time optimization.

Multi-object reconstruction. The recent emergence of neural implicit reconstruction [9, 36, 41] has boosted the performance and flexibility of object reconstruction methods. This is attributed to the learned shape prior and differentiability for test-time optimization [13]. FroDo [48] reconstructs object shapes using detection and multi-view optimization. ELLIPSDF [53] and ODAM [30] further introduce geometric representations, specifically superquadrics, to represent shape primitives and constrain multi-view optimization. Irshad *et al.* [23, 24, 35] do not rely on existing detectors but instead develop a single-shot pipeline to regress object pose, shape, and appearance. BundleSDF [65] focuses on single dynamic objects and generalizes to unknown objects using graph optimization. Our reconstruction is built on top of [41]. We further introduce joint optimization on shape and pose to aggregate multiple observations for more accurate and complete reconstruction.

3. Living Scenes

We define a **living scene** as a built environment with dynamic and static objects. Its reconstruction occurs cumulatively over time from temporal scans and showcases how it has been *lived*. Our method, MoRE^2, creates living scenes and is designed to understand the rigid motion and the geometry of objects. It reconstructs each individual 3D object separately, with increased accuracy and completeness as more temporal scans become available (*c.f.* Fig. 1(d)). These reconstructions can be seamlessly positioned within the scans acquired over time and used for other tasks, *e.g.*, learning from historical data or creating 3D assets.

Problem Setting. Consider a collection of scans $\{\mathbf{S}^t\}_{t=1}^T$ of a 3D environment captured at irregular intervals. Scan $\mathbf{S}^t$ represents the environment observed at time t and contains a list of point clouds $\{\mathbf{X}_i^t\}_{i=1}^N$. Hereafter, we term object-level point clouds as *point cloud* and scene-level point clouds as *scan*. We denote the first scan $\mathbf{S}^1$ as the reference scan and the following ones $\{\mathbf{S}^t | t > 1\}$ as temporal rescans. Our goal is formulated as:

1. **Multi-object relocalization**: We aim to compute the 6DoF rigid transformation $\{\mathbf{T}_i^t \in \mathrm{SE}(3) |\ t > 1\}$ between the point clouds belonging to the same instance in the reference scan and rescan respectively. Specifically, we formulate relocalization in two steps: matching of point clouds followed by their registration.
2. **Object reconstruction**: The goal is to reconstruct each instance from the accumulated point clouds $\{\mathbf{X}^1 \circ \mathbf{T}_i^2\mathbf{X}^2 \circ ... \circ \mathbf{T}_i^t\mathbf{X}^t \mid t > 1\}$, where $\circ$ denotes concatenation operation.

Since our method reasons at the instance level, we assume the availability of instance segmentation masks.[2].

Method Overview. MoRE^2 sequentially addresses instance matching (Sec. 3.2), registration (Sec. 3.3), and reconstruction (Sec. 3.4) with a single compact representation obtained from our encoder-decoder network (Sec. 3.1), which is trained solely on the object reconstruction task. We obtain the final accumulated point clouds per instance through joint shape-pose optimization (Sec. 3.5). An overview is provided in Fig. 2.

3.1. Encoder-decoder Network

Vector Neuron Encoder (VN). To obtain rotation equivariant features from point clouds, we follow Deng *et al.* [11] and "lift" the neuron representation from a scalar to a vector and preserve the rotation during forward propagation. $\mathrm{SO}(3)$ equivariance and invariance of function f are expressed as:

$$f(\mathbf{RX}) = \mathbf{R}f(\mathbf{X}), f(\mathbf{RX}) = f(\mathbf{X}), \tag{1}$$

where $\mathbf{R}$ denotes the rotation applied to input point cloud $\mathbf{X}$. [28] extends VNNs from $\mathrm{SO}(3)$- to $\mathrm{SIM}(3)$-equivariance by additionally estimating a scale factor and the centroid of the point cloud. The encoder Φ takes a point cloud $\mathbf{X}$ as input and outputs $\mathbf{F} = (\mathbf{F_{inv}} \in \mathbb{R}^{256}, \mathbf{F_{eqv}} \in \mathbb{R}^{3\times 256}, \mathbf{F_s} \in \mathbb{R}_+, \mathbf{F_c} \in \mathbb{R}^3)$, with the four components representing the invariant embedding, equivariant embed-

[2]In Sec. 4, we offer an experiment using predicted instance masks as input to MoRE^2. Note that MoRE^2 is agnostic to the semantic labels.

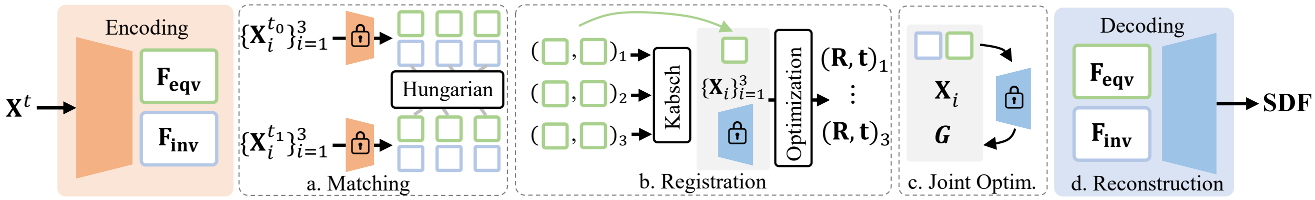

Figure 2. **Overview of the MORE^2 pipeline.** Given two temporal point clouds with instance masks $\{\mathbf{X}_i^{t_0}\}_{i=1}^3$ and $\{\mathbf{X}_i^{t_1}\}_{i=1}^3$, we first use the VN encoder to compute the embeddings for each instance. *a)* Matching solves the pairwise correspondences of the same instances using Hungarian matching [38] on the embeddings. *b)* Registration estimates 6DoF transformations within matched pairs: Kabsch algorithm [26] is employed to compute the initial transform, followed by optimization to further refine the registration. *c)* Joint optimization simultaneously refines the registration and *d)* reconstruction. The output is the signed distance values (SDF) of query coordinates.

ding, scale factor, and centroid, respectively. The $\mathrm{SIM}(3)$-equivariance is then achieved by:

$$\Phi(s\mathbf{R}\mathbf{X}+\mathbf{t}) = (\mathbf{F_{inv}}, \mathbf{R}\mathbf{F_{eqv}}, s\mathbf{F_s}, \mathbf{F_c}+\mathbf{t}). \tag{2}$$

Here Φ is the VN-encoder and $(s, \mathbf{R}, \mathbf{t}) \in \mathrm{SIM}(3)$ denote scale, rotation, and translation respectively. The embedding $\mathbf{F}$ is used to canonicalize the query point $\mathbf{p} \in \mathbb{R}^3$

$$\mathbf{F_q} = \langle \mathbf{F_{eqv}}, (\mathbf{p}-\mathbf{F_c})/\mathbf{F_s} \rangle \tag{3}$$

where $\langle \cdot, \cdot \rangle$ denotes channel-wise inner product. The canonicalized feature $\mathbf{F_q}$ is then fed to the decoder.

Neural Implicit Decoder. Here we use DeepSDF [41] as our neural implicit decoder. It is an auto-decoder network that takes latent code and query coordinates as input and outputs the SDF value at query location.

$$\mathbf{SDF}(\mathbf{p}) = \Psi(\mathbf{F_{inv}}, \mathbf{F_q}), \tag{4}$$

where Ψ represents the DeepSDF decoder. The latent space trained using DeepSDF decoder lays a solid ground for shape interpolation and test-time optimization [13].

Training. We train MORE^2 using the L1 reconstruction loss [28, 41] on individual shapes:

$$\mathcal{L}_{\text{recon}} = \frac{1}{K}\sum_{i=1}^{K} |\overline{\mathbf{SDF}}(\mathbf{p}_i) - \mathbf{SDF}(\mathbf{p}_i)|, \tag{5}$$

where $\overline{\mathbf{SDF}}(\mathbf{p})$ denotes the ground truth SDF value of $\mathbf{p}$ and $\mathbf{SDF}(\mathbf{p})$ the predicted SDF value. $\mathbf{p}_i$ is the sampled point and K denotes the number of SDF samples. Additional details are provided in Supp.

Unlike [13, 41], we make MORE^2 category-agnostic by directly training it across multiple classes. Once the training is complete, we freeze the network's weights. Next, we elaborate on how we flexibly adapt the network and embeddings to address the relocalization and reconstruction tasks.

3.2. Instance Matching

Given two sets of randomly oriented point clouds $\{(\mathbf{X}_i^{t_1})\}_{i=1}^N$, $\{(\mathbf{X}_j^{t_2})\}_{j=1}^M$ of size N and M, the task is to associate them across time (*c.f.* Fig. 1(b) and Fig. 2(a)).

We first compute the cosine similarity matrix $\mathbf{\Lambda} \in \mathbb{R}_+^{N\times M}$ using all the invariant embedding pairs $\{\langle \mathbf{F_{inv}}_i^{t_1}, \mathbf{F_{inv}}_j^{t_2} \rangle\}_{i,j=1}^{N,M}$ as our initial score matrix. Next, since the equivariant embeddings $\mathbf{F_{eqv}}$ can be treated as 3D coordinates in the latent space given the one-to-one correspondences along the feature dimension, we extract the rotations $\mathbf{R}_{i,j}$ between all equivariant embedding pairs via the Kabsch algorithm [26] and consequently factor them out for each pair. Following the factorization, we can compute the alignment residual matrix $\mathbf{E} \in \mathbb{R}_+^{N\times M}$, where

$$\mathbf{E}(i,j) = ||\mathbf{R}_{i,j}\mathbf{F_{eqv}}_{,i}^{t_1} - \mathbf{F_{eqv}}_{,j}^{t_2}||_2 \tag{6}$$

indicates the inverse fitness of equivariant pairs after coarse alignment in the $\mathrm{SO}(3)$ feature space using $\mathbf{R}_{i,j}$. Finally, we compute the aggregated matching score matrix $\mathbf{H} = \mathbf{\Lambda} \oslash \mathbf{E}$, where $\oslash$ denotes element-wise division of matrices. Now the problem is to find the assignment that maximizes the total matching score $\sum_{i,j} \mathbf{H}_{i,j}\mathbf{P}_{i,j}$. In light of existing 2D/3D feature matching paradigms [18, 51], we use the Hungarian Matching [38] to solve this linear assignment. Considering that the numbers of object instances in two sets can differ and some might remain unmatched after Hungarian Matching, we treat unmatched instances as *removed* or *added* based on their appearance order in time.

3.3. Instance Registration

Consider a matched pair ($\mathbf{X}^{t_1} \in \mathbb{R}^{3\times N_1}$, $\mathbf{X}^{t_2} \in \mathbb{R}^{3\times N_2}$) (Fig. 1(c) top). The task is to estimate the relative transformation $\mathbf{T} = (\mathbf{R}, \mathbf{t})$ that aligns the source $\mathbf{X}^{t_2}$ to the target $\mathbf{X}^{t_1}$. To address this, we propose the following optimization-based registration (Fig. 2(b)). We first compute the $\mathrm{SE}(3)$-equivariant embeddings $\mathbf{F}_{se3} = \mathbf{F_{eqv}} + \mathbf{F_c}$ for each point cloud and solve $(\mathbf{R}, \mathbf{t})$ using Kabsch algorithm [26]. This serves as the initialization for our registration. Next, the optimal transformation $(\mathbf{R}^*, \mathbf{t}^*)$ is obtained by minimizing $\mathcal{L}_{\text{reg}}$:

$$(\mathbf{R}^*, \mathbf{t}^*) = \underset{(\mathbf{R},\mathbf{t})}{\arg\min}\, \mathcal{L}_{\text{reg}}(\mathbf{X}^{t_1}, \mathbf{X}^{t_2}), \tag{7}$$

where $\mathcal{L}_{\text{reg}}$ is defined as:

$$\mathcal{L}_{\text{reg}}(\mathbf{X}^{t_1}, \mathbf{X}^{t_2}) = \underbrace{||\Psi(\mathbf{F}^{t_1}_{\mathbf{inv}}, \langle \mathbf{F}^{t_1}_{\mathbf{eqv}}, (\mathbf{R}_{\text{i}}\mathbf{X}^{t_2} + \mathbf{t}_{\text{i}} - \mathbf{F}^{t_1}_{\mathbf{c}})/\mathbf{F}^{t_1}_{\mathbf{s}}\rangle)||_1}_{\mathcal{L}_{\text{sdf}}} + \underbrace{\tilde{CD}(\mathbf{R}_{\text{i}}\mathbf{X}^{t_2} + \mathbf{t}_{\text{i}}, \mathbf{X}^{t_1})}_{\mathcal{L}_{\text{cd}}}. \tag{8}$$

Here $\mathcal{L}_{\text{sdf}}$ denotes the misalignment between the point cloud and zero level-set, and $\mathcal{L}_{\text{cd}}$ the chamfer loss between the current estimate and target point cloud.

In the ith iteration, we use the target embedding $\mathbf{F}^{t_1}_{se3}$ to canonicalize the source point cloud $\mathbf{X}^{t_2}$ transformed by current $(\mathbf{R}_{\text{i}}, \mathbf{t}_{\text{i}})$ and compute $\mathcal{L}_{\text{reg}}$. We directly optimize $(\mathbf{R}, \mathbf{t})$ through back-propagation on $\text{SE}(3)$ manifold using *TorchLie* [44] for faster and more stable convergence [39, 57]. Our optimization iteratively aligns the source point cloud to the *zero level-set* of the target point cloud, together with minimization of point-wise misalignment. After optimization, we refine the point cloud alignment using iterative closest point [5] to obtain the final output. Furthermore, we can **classify static /dynamic** objects in the scene by thresholding their transformation distances.

3.4. Instance Reconstruction

After obtaining the **matched** and **aligned** point cloud pairs $\{(\mathbf{X}^{t_1}, \mathbf{X}^{t_2}, \mathbf{R}, \mathbf{t})_i\}_{i=0}^{M}$, we proceed to reconstructing them (Fig. 1(c) bottom). We first down-sample the accumulated point clouds using farthest point sampling (FPS) [46]. Next, we compute the new embedding $\mathbf{F}_*$ from the down-sampled point cloud. Finally, we query the SDF values of a voxel grid with 64^3 resolution using $\mathbf{F}_*$ and DeepSDF decoder, as is shown in Eq. (4). Following the previous literature [28, 41, 48], we use Multi-resolution IsoSurface Extraction (MISE) [36] to extract the zero level-set as object reconstruction.

3.5. Joint Optimization for Accumulation

So far, we have discussed relocalization and reconstruction between two temporal scans. To leverage observations from multiple scans, we propose a joint optimization algorithm to refine the registration and reconstruction (Fig. 2(c),(d)) and accumulate point clouds with increasing geometric accuracy and completeness over time(Fig. 1(d)).

Initialization. Consider the matched and registered point clouds $\{\mathbf{X}^t\}_{t=t_1}^{t_K}$ and their associated equivariant and invariant embeddings from VN-encoder. For each point cloud $\mathbf{X}^t$, we compute its $\mathcal{L}_{\text{sdf}}$ value and choose the one $\mathbf{X}^*$ with the best agreement between the point cloud and the zero level set defined by its embeddings, as our initialization. Specifically, we initialize $\mathbf{F}$ with the equivariant embedding $\mathbf{F}^*_{\mathbf{eqv}}$ and construct the pose graph $\mathbf{G} = \{\mathbf{T}^t\}_{t=t_1}^{t_K}$. Here $\mathbf{T}^t$ aligns $\mathbf{X}^t$ to $\mathbf{X}^*$ and is computed by the previously introduced registration method.

Optimization Objectives. We jointly optimize the shared equivariant embedding $\mathbf{F}$ and pose graph $\mathbf{G}$ by minimizing $\mathcal{L}_{joint} = \mathcal{L}_{\text{sdf}} + \mathcal{L}_z$. Here $\mathcal{L}_{\text{sdf}}$ denotes the SDF error of accumulated point clouds and $\mathcal{L}_z = ||\mathbf{F}' - \mathbf{F}||_2$ is the regularization term to constrain variations w.r.t. the initial $\mathbf{F}$. Similar to registration, the pose graph is optimized on $\text{SE}(3)$ manifold [44, 57] and the embedding is optimized using Adam optimizer [27] for 200 iterations.

3.6. Implementation Details

We use the VN Transformer [2] and DeepSDF [41] as our encoder and decoder, respectively. We implement MoRE^2 using PyTorch [42] and train it on a single NVIDIA A100 (80GB) GPU for 2×10^5 iterations with batch size = 64. We decay the learning rate (0.0001) by 0.3 at 1.2×10^5, 1.5×10^5, and 1.8×10^5 iterations. For more details, we refer the reader to the Supp.

4. Experiments

We evaluate MoRE^2 on its end-to-end performance on the tasks of multi-object relocalization and reconstruction (Sec. 4.3), as well as on each of the three subtasks individually (Secs. 4.4 to 4.6). When evaluating end-to-end performance, we input to each subsequent task the output of the preceding one, thereby inheriting any accompanying noise and errors. When evaluating on each task independently, we provide as input the ground truth information, i.e., we provide correct instance matches to the registration task and correct registration pairs to the reconstruction task. We identify baseline methods per task and combine the best-performing ones as the end-to-end baseline. We investigate the impact of predicted instance segmentation masks as input to MoRE^2 and the benefit of accumulation in Sec. 5. For more analysis on design choices, see Supp.

4.1. Datasets

In our experiments, for both MoRE^2 and baselines, we train on a synthetic dataset and test on both this and a real-world dataset, evaluating the generalization ability of the methods. We synthesize our own living scenes as there is no available synthetic 3D dataset of indoor scenes that exhibits long-term changes. All data will be made available.

***FlyingShape*.** We synthesize the *FlyingShape* dataset using a ShapeNet [7] *subset*, containing 7 categories: chair, table, sofa, pillow, bench, couch, and trash can. The training-validation split follows ShapeNet [7]. We randomly sample objects from the *subset*'s test set and compose 100 unique 3D scenes from them as our **test set**. We assign random poses to objects while ensuring they touch the scene's ground. To emulate long-term dynamics, we introduce random changes to all object poses and sequentially generate five temporal scans per scene. See Supp. for more details.

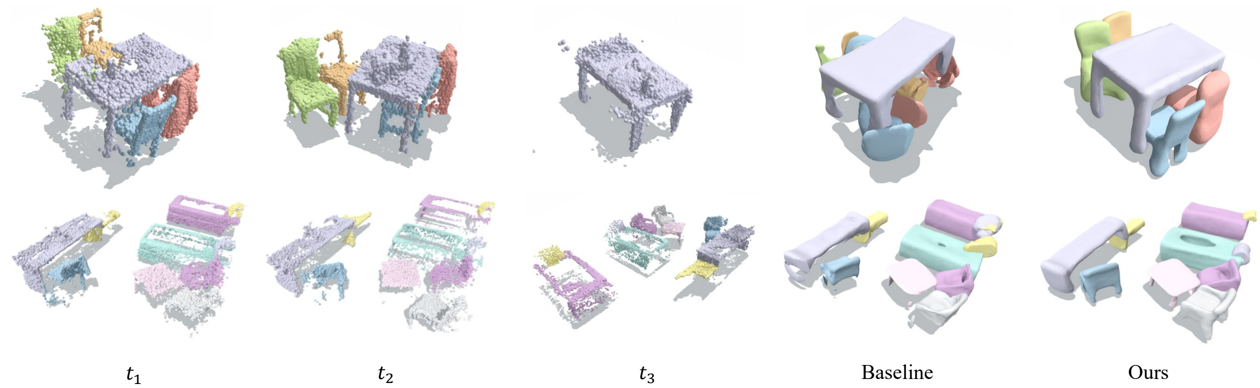

Figure 3. **End-to-end cumulative reconstruction with multiple scans.** t_1, t_2, and t_3 denote the same scene captured at three times. Point clouds from t_2 and t_3 are accumulated to t_1. Interestingly, chairs in t_3 (top) are removed from the scene, but MoRE^2 is able to handle it.

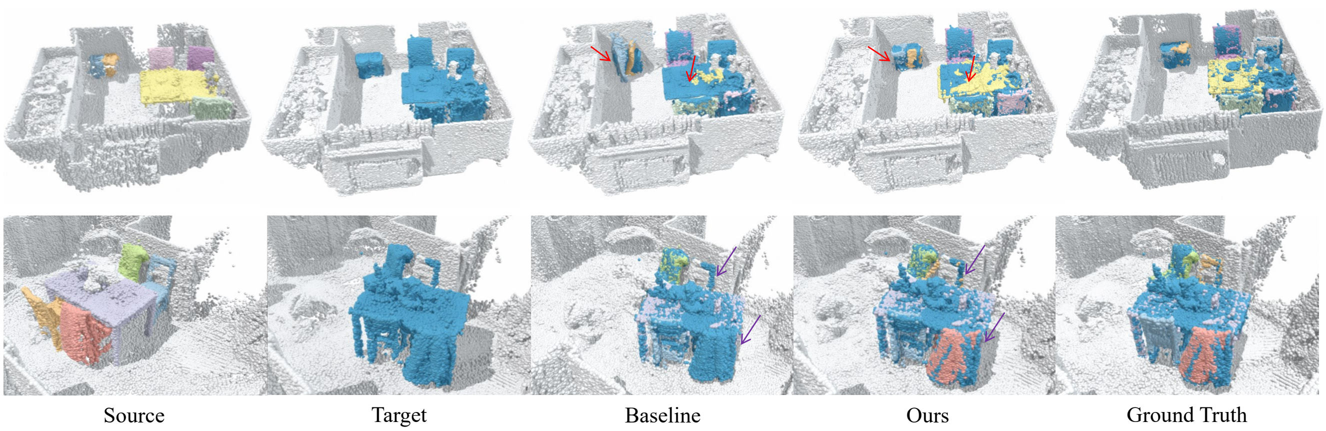

Figure 4. **Multi-object relocalization on *3RScan* [62].** Instances, uniquely colored in source scan, are matched and registered to their corresponding instances in target scan, as per ground truth. ↘ highlights differences between methods on registration and ↘ on matching.

***3RScan* [62].** *3RScan* is a real-world dataset for benchmarking object instance relocalization, consisting of 1428 RGB-D sequences of 478 indoor scenes that include rescans of them. It provides annotated instance segmentation, associations, and transformations between temporal scans. We use the validation set of *3RScan* for evaluation[3]. To evaluate our comprehensive tasks, we extend *3RScan* to an instance matching and reconstruction benchmark.

4.2. Evaluation Metrics

For **instance matching**, we get inspiration from the evaluation of image feature matching [34, 51] and calculate the instance-level matching recall, which measures the proportion of correct matches.

We also calculate the scene-level matching recall and use as thresholds 25%, 50%, 75%, and 100% to denote the minimum acceptable ratio of number of correct matches over the total number of matches between two temporal scans. As **point cloud registration** is a standardized task, we follow [17, 21, 47] and report registration recall (RR), median rotation error (MedRE), transformation error (RMSE), and median Chamfer Distance (MedCD). The rotation error threshold for registration recall is $5°$ for *FlyingShape*, and $10°$ for *3RScan* due to its low accuracy in ground truth annotations.

For **instance reconstruction**, we follow [36, 41, 43] and report Chamfer Distance and volumetric IoU. Moreover, we introduce signed distance function (SDF) recall to describe the successful ratio of reconstruction at instance level.

End-to-end Metrics. To assess the end-to-end performance on multi-object relocalization and reconstruction, we propose two joint metrics, namely *MR* recall and *MRR* recall. *MR* stands for the end-to-end recall of relocalization

$$\mathrm{P}(R_1, M) = \mathrm{P}(R_1|M)\mathrm{P}(M). \tag{9}$$

Here M and R_1 denote the event of an instance being correctly matched and registered, respectively. $\mathrm{P}(\cdot)$ denotes the probability of an event to happen. We pass the output of instance matching to registration and calculate registration recall (RR) as *MR* recall, to include both the errors in

[3]The ground truth information of the *3RScan* test set is hidden on a private server that is no longer maintained.

	FlyingShape		*3RScan*	
Method	***MR* Recall ↑**	***MRR* Recall ↑**	***MR* Recall ↑**	***MRR* Recall ↑**
Baseline†	67.32	54.30	44.02	30.77
Ours	74.39	62.00	49.07	40.74

Table 1. **End-to-end performance.** *MR* evaluates joint matching and registration, while *MRR* measures all tasks.

	Instance-level	**Scene-level Recall**		
Method	**Recall ↑**	**R@50%↑**	**R@75%↑**	**R@100%↑**
MendNet [13]	83.69	96.75	68.25	60.75
VN-DGCNN$_{\text{cls}}$ [11]	61.37	73.50	32.25	27.75
VN-ONet$_{\text{recon}}$ [11]	86.63	96.00	74.50	67.75
Ours	88.75	97.50	78.00	72.00

Table 2. **Instance matching results on *FlyingShape*.**

	Instance-level Recall ↑			**Scene-level Recall ↑**		
Method	**Static**	**Dynamic**	**All**	**R@25%**	**R@50%**	**R@75%**
MendNet [13]	60.32	63.76	62.20	80.68	64.77	37.50
VN-DGCNN$_{\text{cls}}$ [11]	43.39	49.34	46.65	72.32	53.41	29.55
VN-ONet$_{\text{recon}}$ [11]	56.08	72.05	64.83	86.36	71.59	44.32
Ours	60.32	87.50	71.77	87.50	78.41	50.00

Table 3. **Instance matching results on *3RScan* [62].**

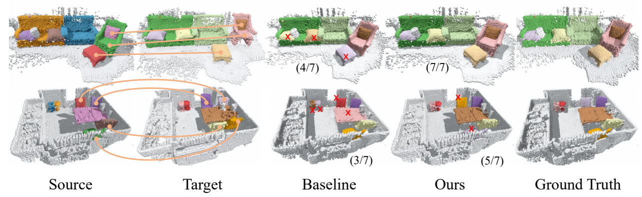

Figure 5. **Multi-object matching on *3RScan* [62].** We repaint the instances in the source scan using the same colors as those of matched instances in the target scan. X denotes the wrongly matched instances. Curves depict the associations of moving objects. *(5/7)* denoting 5 correct matches out of 7 pairs in the scene.

Dataset	**Method**	**RR ↑**	**MedRE ↓**	**RMSE ↓**	**MedCD ↓**
FlyingShape	RPMNet [71]	23.17	2.37	31.77	0.0249
	FreeReg [77]	47.50	2.44	33.84	0.0760
	GeoTransformer [47]	77.67	1.36	16.66	0.0271
	Ours w/o optim	83.00	0.86	20.83	0.0171
	Ours full	83.83	0.74	18.47	0.0168
3RScan [62]	RPMNet [71]	9.40	3.78	15.91	0.0248
	FreeReg [77]	26.06	5.76	11.05	0.0082
	GeoTransformer [47]	51.71	3.46	6.51	0.0141
	Ours w/o optim	58.12	3.77	5.49	0.0032
	Ours full	61.11	3.77	4.74	0.0030

Table 4. **Point cloud registration results on both datasets.**

matching and registration. Similarly, *MRR* is formulated as

$$\mathrm{P}(M, R_1, R_2) = \mathrm{P}(R_2|R_1, M)\mathrm{P}(R_1|M)\mathrm{P}(M), \quad (10)$$

where R_2 denotes the event of an instance being correctly reconstructed. Here, we pass the predicted matches and the resulting registration to the reconstruction task and use the SDF recall as *MRR*. As such, the performance of all three tasks is evaluated in a single metric.

We train our model and all the baselines on the training set of the ShapeNet [69] *subset*, and evaluate on *FlyingShape* and *3RScan*. In the following quantitative evaluations, we highlight results being best and second-best.

4.3. End-to-end Performance

We combine the best-performing baseline methods in each task as the **end-to-end baseline** (Baseline†), which comprises VN-ONet [11], GeoTransformer [47], and ConvONet [43]. As shown in Tab. 1, our method consistently outperforms the baseline method across all metrics. Notably, there is a similar performance decrease for both methods when comparing numbers on *FlyingShape* to those on *3RScan*, as anticipated due to the inherent domain gap. Also, there is an increased gap between our method and the baseline in end-to-end evaluation vs. the results on individual tasks (Sec. 4.4 to 4.6), for both relocalization[4] and reconstruction. This disparity is attributed to our unified approach, utilizing a single network and representation that retains shape and pose information. In contrast, the combined baseline lacks coherence between tasks, employing three distinct networks and representations. Qualitative results on *3RScan* [62] are in Fig. 3 for end-to-end performance (MRR) and in Fig. 4 for relocalization (MR).

[4] 3RScan [62] provides a baseline for instance relocalization. We do not compare with it because it takes as input TSDF patches, not point clouds, and it is not reproducible due to missing codebase and inadequate details.

4.4. Instance Matching

We compare MoRE^2 with three baselines, namely MendNet [13], VN-ONet [11, 36], and VN-DGCNN [11, 64], of which the first two are point cloud reconstruction networks, and the last is a point cloud classification network. We present the results in Tab. 2 and Tab. 3 for *FlyingShape* and *3RScan*, respectively. MoRE^2 outperforms the baseline methods on all metrics. This can be attributed to the representation power of its embeddings: the encoder can output expressive global invariant features to handle large pose variations, and the equivariant features model the high-frequency details of the input point cloud. In Fig. 5, we showcase that MoRE^2 can handle in-category object matching by capturing minor geometric variations.

4.5. Point Cloud Registration

We compare MoRE^2 with three baselines: RPMNet [71], which is a learning-based method that only targets object-level registration; FreeReg [77], which uses equivariant embeddings to solve rotation; and GeoTransformer [47], which is the state-of-the-art method on 3DMatch [75] and 3DLo-

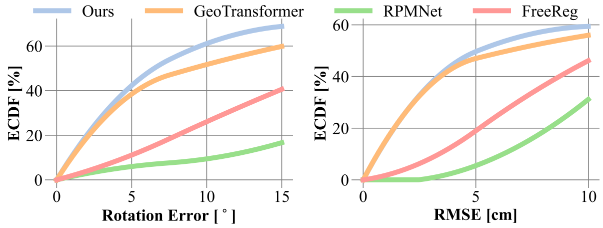

Figure 6. **ECDF curves of registration results on *3RScan* [62].**

Dataset	*FlyingShape*			*3RScan*	
Method	$L1$-Chamfer$_2$ ↓	IoU ↑	SDF Rec.↑	$L1$-Chamfer$_1$ ↓	SDF Rec. ↑
MendNet [13]	25.27	47.79	6.17	17.73	20.99
VN-ONet [11]	8.55	34.47	65.00	10.65	51.91
ConvONet [43]	6.64	36.99	80.67	7.61	64.89
Ours w/o optim	6.27	49.98	78.00	9.28	56.87
Ours full	6.11	66.73	83.33	6.16	64.12

Table 5. **Instance reconstruction results**. $L1$-Chamfer $\times 10^{-3}$.

Method	Ins. Seg.	Mat. Rec. ↑	Reg. Rec. ↑	*MR* Rec. ↑	*MRR* Rec. ↑
Baseline†	GT	64.83	51.71	44.02	30.77
Ours		71.77	61.11	49.07	40.74
Baseline†	Mask3D [52]	43.43	47.74	27.86	20.89
Ours		45.76	51.27	40.14	33.80

Table 6. **Results with Mask3D [52] on *3RScan* [62].**

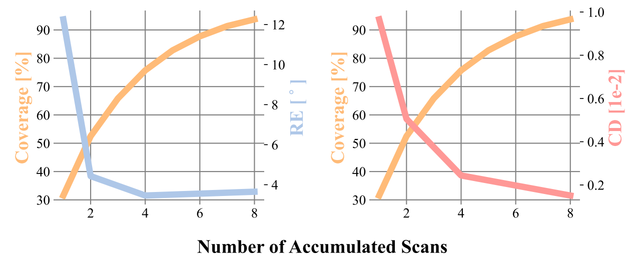

Figure 7. **Ablation study on point cloud accumulation.** The change of point cloud coverage, rotation error and chamfer distance w.r.t the number of accumulated scans.

Match [21] datasets. Results on *FlyingShape* and *3RScan* are in Tab. 4. [77] can only provide coarse registration and does not work well under large partiality changes. In contrast to [71] and [47], MoRE^2 does not rely on discrete point-wise correspondences but represents the geometry with continuous signed distance field and aligns the point cloud with the field via optimization. Our analysis is further corroborated by the highest Empirical Cumulative Distribution Function (ECDF) curve of MoRE^2 in Fig. 6.

4.6. Instance Reconstruction

We compare MoRE^2 with MendNet [13], VN-ONet [11] and ConvONet [43]; results are in Tab. 5. We report the 2-way chamfer on *FlyingShape* and only the 1-way chamfer on *3RScan* as it only provides incomplete object meshes (non-watertight). With joint optimization, our full method surpasses the baselines on most metrics across the two datasets. Without, it is on par with ConvONet on *FlyingShape*. This demonstrates the adaptation power of our optimization algorithm on noisy and randomly oriented point clouds. In contrast to baseline methods that only perform surface reconstruction, our pose graph $\mathbf{G}$ and shared embedding $\mathbf{F}$ enable optimization message passing and fusion between accumulated point clouds, improving both registration and reconstruction performance (*c.f*. Tab. 4 and Tab. 5).

5. Ablation Study

Predicted instance segmentation. Noisy and incomplete instance segmentation masks from Mask3D [52] are provided to MoRE^2. Results in Tab. 6 across all tasks and combinations, when compared to GT masks, show our method outperforming the combined baseline. More importantly, MR and MRR recall is substantially lower for the baseline, despite a similar matching and registration recall for both methods. Also, the drop of the baseline in MR and MRR recall between GT and Mask3D [52] is $\approx$ 33%, vs $\approx 18\%$ in ours. These findings demonstrate the efficacy of MoRE^2's shared representation across tasks and joint optimization, even in noisy settings.

Benefit of accumulation. The results in Tab. 4 are computed for pairs of point clouds, *i.e*., between two points in time. Here, we experiment with increasing the number of multi-temporal scans used for accumulation (*c.f*. Fig. 7) and report the performance of registration (RE) and reconstruction (CD) on *FlyingShape*, to showcase an increasing geometric accuracy and completeness over time. We see significant improvement on both metrics in the range of no accumulation (one point cloud) to four point clouds, after which the performance starts to saturate. The saturation is explained when compared to the coverage ratio of accumulated point cloud w.r.t. the complete shape. By the 4th scan, completeness is close to 75%, hence any additional scan will affect less registration and reconstruction.

6. Conclusion

We propose MoRE^2, a novel approach to parse long-term dynamic scenes (living scenes) involving three consecutive tasks. MoRE^2 solves the three tasks by flexible adaptation of equivariant embeddings and a joint optimization that enables multi-temporal accumulation. Our approach exhibits superior performance across both synthetic and real-world datasets. It empowers the cumulative comprehension of 3D assets in the scene. Future research directions involve addressing challenges posed by the presence of elastic deformations and multiple identical objects in the scene and comprehending large-scale spatiotemporal changes [56].

References

[1] Aikaterini Adam, Torsten Sattler, Konstantinos Karantzalos, and Tomas Pajdla. Objects can move: 3d change detection by geometric transformation consistency. In *ECCV*, 2022.

[2] Serge Assaad, Carlton Downey, Rami Al-Rfou, Nigamaa Nayakanti, and Ben Sapp. Vn-transformer: Rotation-equivariant attention for vector neurons. *Transactions on Machine Learning Research*, 2023.

[3] Stefan Andreas Baur, David Josef Emmerichs, Frank Moosmann, Peter Pinggera, Björn Ommer, and Andreas Geiger. Slim: Self-supervised lidar scene flow and motion segmentation. In *ICCV*, pages 13126–13136, 2021.

[4] Jerome Berclaz, Francois Fleuret, Engin Turetken, and Pascal Fua. Multiple object tracking using k-shortest paths optimization. *TPAMI*, 2011.

[5] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, pages 586–606. Spie, 1992.

[6] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.

[7] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv:1512.03012*, 2015.

[8] Yunlu Chen, Basura Fernando, Hakan Bilen, Matthias Nießner, and Efstratios Gavves. 3d equivariant graph implicit functions. In *ECCV*, 2022.

[9] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *CVPR*, 2019.

[10] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *ICML*, 2016.

[11] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J Guibas. Vector neurons: A general framework for so (3)-equivariant networks. In *CVPR*, 2021.

[12] Congyue Deng, Jiahui Lei, Bokui Shen, Kostas Daniilidis, and Leonidas Guibas. Banana: Banach fixed-point network for pointcloud segmentation with inter-part equivariance. In *NeurIPS*, 2023.

[13] Shivam Duggal, Zihao Wang, Wei-Chiu Ma, Sivabalan Manivasagam, Justin Liang, Shenlong Wang, and Raquel Urtasun. Mending neural implicit modeling for 3d vehicle reconstruction in the wild. In *WACV*, 2022.

[14] Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc Van Gool. A mobile vision system for robust multi-person tracking. In *CVPR*, 2008.

[15] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. SE(3)-transformers: 3d roto-translation equivariant attention networks. In *NeurIPS*, 2020.

[16] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013.

[17] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The perfect match: 3d point cloud matching with smoothed densities. In *CVPR*, 2019.

[18] Zan Gojcic, Or Litany, Andreas Wieser, Leonidas J Guibas, and Tolga Birdal. Weakly supervised learning of rigid 3d scene flow. In *CVPR*, 2021.

[19] Maciej Halber, Yifei Shi, Kai Xu, and Thomas Funkhouser. Rescan: Inductive instance segmentation for indoor rgbd scans. In *ICCV*, pages 2541–2550, 2019.

[20] Jiahui Huang, He Wang, Tolga Birdal, Minhyuk Sung, Federica Arrigoni, Shi-Min Hu, and Leonidas J Guibas. MultiBodySync: Multi-body segmentation and motion estimation via 3d scan synchronization. In *CVPR*, 2021.

[21] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *CVPR*, 2021.

[22] Shengyu Huang, Zan Gojcic, Jiahui Huang, Andreas Wieser, and Konrad Schindler. Dynamic 3d scene analysis by point cloud accumulation. In *ECCV*, 2022.

[23] Muhammad Zubair Irshad, Thomas Kollar, Michael Laskey, Kevin Stone, and Zsolt Kira. Centersnap: Single-shot multi-object 3d shape reconstruction and categorical 6d pose and size estimation. 2022.

[24] Muhammad Zubair Irshad, Sergey Zakharov, Rares Ambrus, Thomas Kollar, Zsolt Kira, and Adrien Gaidon. Shapo: Implicit representations for multi-object shape, appearance, and pose optimization. In *ECCV*, 2022.

[25] Shengze Jin, Daniel Barath, Marc Pollefeys, and Iro Armeni. Q-reg: End-to-end trainable point cloud registration with surface curvature. *3DV*, 2023.

[26] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976.

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014.

[28] Jiahui Lei, Congyue Deng, Karl Schmeckpeper, Leonidas Guibas, and Kostas Daniilidis. EFEM: Equivariant neural field expectation maximization for 3d object segmentation without scene supervision. In *CVPR*, 2023.

[29] Fernanda Leite. Buildings as living things. https://cockrell.utexas.edu/news/archive/9042-buildings-as-living-things-6-questions-with-fernanda-leite, 2020.

[30] Kejie Li, Daniel DeTone, Yu Fan Steven Chen, Minh Vo, Ian Reid, Hamid Rezatofighi, Chris Sweeney, Julian Straub, and Richard Newcombe. Odam: Object detection, association, and mapping using posed rgb video. In *ICCV*, 2021.

[31] Kejie Li, Hamid Rezatofighi, and Ian Reid. Moltr: Multiple object localization, tracking and reconstruction from monocular rgb videos. *Robotics and Automation Letters*, 2021.

[32] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, 2021.

[33] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. Flownet3d: Learning scene flow in 3d point clouds. In *CVPR*, 2019.

[34] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[35] Mayank Lunayach, Sergey Zakharov, Dian Chen, Rares Ambrus, Zsolt Kira, and Muhammad Zubair Irshad. Fsd: Fast self-supervised single rgb-d to categorical 3d objects. 2023.

[36] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 2019.

[37] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv:1603.00831*, 2016.

[38] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.

[39] Yue Pan, Federico Magistri, Thomas Läbe, Elias Marks, Claus Smitt, Chris McCool, Jens Behley, and Cyrill Stachniss. Panoptic mapping with fruit completion and pose estimation for horticultural robots. *IROS*, 2023.

[40] Ziqi Pang, Zhichao Li, and Naiyan Wang. Simpletrack: Understanding and rethinking 3d multi-object tracking. In *ECCV*, 2022.

[41] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019.

[42] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[43] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *ECCV*, 2020.

[44] Luis Pineda, Taosha Fan, Maurizio Monge, Shobha Venkataraman, Paloma Sodhi, Ricky TQ Chen, Joseph Ortiz, Daniel DeTone, Austin Wang, Stuart Anderson, Jing Dong, Brandon Amos, and Mustafa Mukadam. Theseus: A Library for Differentiable Nonlinear Optimization. In *NeurIPS*, 2022.

[45] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017.

[46] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017.

[47] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric transformer for fast and robust point cloud registration. In *CVPR*, 2022.

[48] Martin Runz, Kejie Li, Meng Tang, Lingni Ma, Chen Kong, Tanner Schmidt, Ian Reid, Lourdes Agapito, Julian Straub, Steven Lovegrove, et al. Frodo: From detections to 3d objects. In *CVPR*, 2020.

[49] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (FPFH) for 3d registration. In *ICRA*, 2009.

[50] Sayan Deb Sarkar, Ondrej Miksik, Marc Pollefeys, Daniel Barath, and Iro Armeni. Sgaligner: 3d scene alignment with scene graphs. *arXiv:2304.14880*, 2023.

[51] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020.

[52] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3D: Mask transformer for 3d semantic instance segmentation. In *ICRA*, 2023.

[53] Mo Shan, Qiaojun Feng, You-Yi Jau, and Nikolay Atanasov. Ellipsdf: joint object pose and shape optimization with a bi-level ellipsoid and signed distance function description. In *ICCV*, 2021.

[54] Ziyang Song and Bo Yang. Ogc: Unsupervised 3d object segmentation from rigid dynamics of point clouds. In *NeurIPS*, 2022.

[55] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020.

[56] Tao Sun, Yan Hao, Shengyu Huang, Silvio Savarese, Konrad Schindler, Marc Pollefeys, and Iro Armeni. Nothing stands still: A spatiotemporal benchmark on 3d point cloud registration under large geometric and temporal change. *arXiv:2311.09346*, 2023.

[57] Zachary Teed and Jia Deng. Tangent space backpropagation for 3d transformation groups. In *CVPR*, 2021.

[58] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv:1802.08219*, 2018.

[59] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *ECCV*, 2010.

[60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[61] Christoph Vogel, Konrad Schindler, and Stefan Roth. 3d scene flow estimation with a rigid motion prior. In *ICCV*. IEEE, 2011.

[62] Johanna Wald, Armen Avetisyan, Nassir Navab, Federico Tombari, and Matthias Nießner. Rio: 3d object instance re-localization in changing indoor environments. In *CVPR*, 2019.

[63] Haiping Wang, Yuan Liu, Zhen Dong, and Wenping Wang. You only hypothesize once: Point cloud registration with rotation-equivariant descriptors. In *ACM Multimedia*, 2022.

[64] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (ToG)*, 38(5):1–12, 2019.

[65] Bowen Wen, Jonathan Tremblay, Valts Blukis, Stephen Tyree, Thomas Müller, Alex Evans, Dieter Fox, Jan Kautz, and Stan Birchfield. Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects. In *CVPR*, 2023.

[66] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 3d multi-object tracking: A baseline and new evaluation metrics. In *IROS*, 2020.

[67] Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, and Junsong Yuan. Track to detect and segment: An online multi-object tracker. In *CVPR*, 2021.

[68] Wenxuan Wu, Zhi Yuan Wang, Zhuwen Li, Wei Liu, and Li Fuxin. Pointpwc-net: Cost volume on point clouds for (self-) supervised scene flow estimation. In *ECCV*, pages 88–107. Springer, 2020.

[69] Jin Xie, Yi Fang, Fan Zhu, and Edward Wong. Deepshape: Deep learned shape descriptor for 3d shape matching and retrieval. In *CVPR*, 2015.

[70] Jingyun Yang, Congyue Deng, Jimmy Wu, Rika Antonova, Leonidas Guibas, and Jeannette Bohg. EquivAct: SIM(3)-equivariant visuomotor policies beyond rigid object manipulation, 2023.

[71] Zi Jian Yew and Gim Hee Lee. Rpm-net: Robust point matching using learned features. In *CVPR*, 2020.

[72] Zi Jian Yew and Gim Hee Lee. REGTR: End-to-end point cloud correspondences with transformers. In *CVPR*, 2022.

[73] Hao Yu, Zheng Qin, Ji Hou, Mahdi Saleh, Dongsheng Li, Benjamin Busam, and Slobodan Ilic. Rotation-invariant transformer for point cloud matching. In *CVPR*, 2023.

[74] Junle Yu, Luwei Ren, Yu Zhang, Wenhui Zhou, Lili Lin, and Guojun Dai. PEAL: Prior-embedded explicit attention learning for low-overlap point cloud registration. In *CVPR*, 2023.

[75] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *CVPR*, 2017.

[76] Changqing Zhou, Zhipeng Luo, Yueru Luo, Tianrui Liu, Liang Pan, Zhongang Cai, Haiyu Zhao, and Shijian Lu. Pttr: Relational 3d point cloud object tracking with transformer. In *CVPR*, 2022.

[77] Minghan Zhu, Maani Ghaffari, and Huei Peng. Correspondence-free point cloud registration with SO(3)-equivariant implicit shape representations. In *CoRL*, 2022.