2024 IEEE International Conference on Robotics and Automation (ICRA)
May 13-17, 2024. Yokohama, Japan

Multi-Model 3D Registration: Finding Multiple Moving Objects in Cluttered Point Clouds

David Jin, Sushrut Karmalkar, Harry Zhang, Luca Carlone

2024 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-8457-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICRA57147.2024.10610926

***Abstract*— We investigate a variation of the 3D registration problem, named *multi-model 3D registration*. In the multi-model registration problem, we are given two point clouds picturing a set of objects at different poses (and possibly including points belonging to the background) and we want to simultaneously reconstruct how all objects moved between the two point clouds. This setup generalizes standard 3D registration where one wants to reconstruct a single pose, *e.g.,* the motion of the sensor picturing a static scene. Moreover, it provides a mathematically grounded formulation for relevant robotics applications, *e.g.,* where a depth sensor onboard a robot perceives a dynamic scene and has the goal of estimating its own motion (from the static portion of the scene) while simultaneously recovering the motion of all dynamic objects. We assume a correspondence-based setup where we have putative matches between the two point clouds and consider the practical case where these correspondences are plagued with outliers. We then propose a simple approach based on Expectation-Maximization (EM) and establish theoretical conditions under which the EM approach converges to the ground truth. We evaluate the approach in simulated and real datasets ranging from table-top scenes to self-driving scenarios and demonstrate its effectiveness when combined with state-of-the-art scene flow methods to establish dense correspondences.**

I. Introduction

3D registration is a foundational problem in robotics and computer vision and arises in several applications, including motion estimation and 3D reconstruction [1], [2], [3], object pose estimation [4], [5], [6], and medical imaging [7], [8]; rotation-only variations of the problem also arise in panorama stitching [9] and satellite attitude determination [10].

3D Registration. In its simplest form, 3D registration looks for the rotation $\boldsymbol{R} \in \mathrm{SO}(3)$ and translation $\boldsymbol{t} \in \mathbb{R}^3$ that align two sets of points $\{\boldsymbol{a}_i\}_{i=1}^n$ and $\{\boldsymbol{b}_i\}_{i=1}^n$. If the correspondences between the two sets of points are known, *i.e.,* we know that point $\boldsymbol{b}_i$ in the second point cloud corresponds to point $\boldsymbol{a}_i$ in the first point cloud after a rigid transformation $(\boldsymbol{R}, \boldsymbol{t})$ is applied, then the problem can be formulated as a nonlinear least squares problem and solved in closed form [11], [12]. More formally, if we assume the following generative model

D. Jin, H. Zhang, and L. Carlone are with the Laboratory for Information & Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, USA, {jindavid,harryz,lcarlone}@mit.edu

S. Karmalkar is with the Department of Computer Science at the University of Wisconsin at Madison, Madison, WI, USA, s.sushrut@gmail.com

This work was partially funded by the Army Research Laboratory Distributed and Collaborative Intelligent Systems and Technology Collaborative Research Alliance, by the Office of Naval Research RAPID project, and the NSF CAREER award "Certifiable Perception for Autonomous Cyber-Physical Systems".

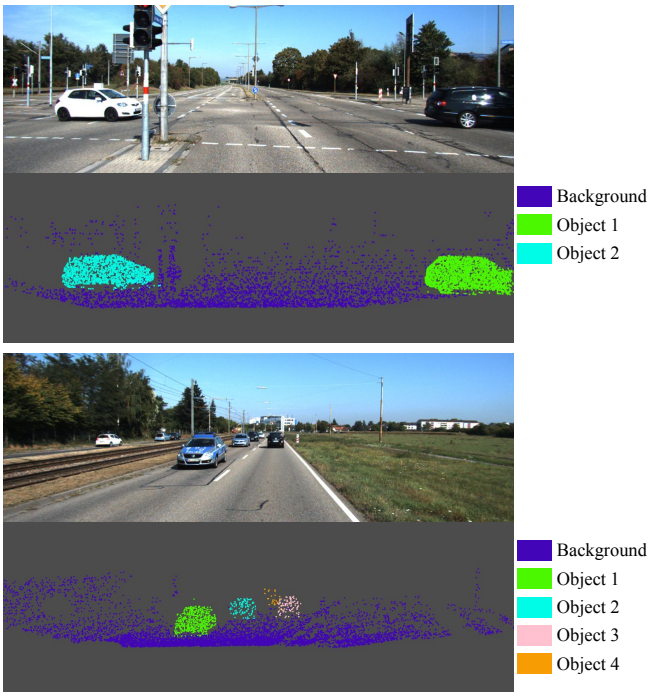


Fig. 1: We propose an Expectation-Maximization approach for *multi-model 3D registration*, which aims to recover the motion of all objects (and background) in a scene from point cloud observations. The figure reports two results produced by our approach on the KITTI dataset. Note that the two cars on the left of the bottom figure are stationary, hence they are correctly deemed to be part of the background.

$$\boldsymbol{b}_i = \boldsymbol{R}\boldsymbol{a}_i + \boldsymbol{t} + \boldsymbol{\epsilon}, \quad i = 1, \dots, n \tag{1}$$

where $\boldsymbol{\epsilon}$ is a noise term distributed according to an isotropic Gaussian, then a maximum likelihood estimate for $(\boldsymbol{R}, \boldsymbol{t})$ can be computed by solving the following nonlinear least squares:

$$\min_{\boldsymbol{R}\in\mathrm{SO}(3), \boldsymbol{t}\in\mathbb{R}^3} \sum_{i=1}^n \|\boldsymbol{b}_i - \boldsymbol{R}\boldsymbol{a}_i - \boldsymbol{t}\|^2 \tag{2}$$

which admits a well-known closed-form solution via singular value decomposition (SVD) [11], [12].

Robust 3D Registration. In practical problems, the measurements contain spurious correspondences. For instance, if the two point clouds represent two RGB-D scans at consecutive time stamps and we are trying to estimate the motion of the sensor between scans, we might attempt to establish correspondences $(\boldsymbol{b}_i, \boldsymbol{a}_i)$, $i = 1, \dots, n$, using descriptor matching [13], optical flow [14], or scene flow [15]. As a result, some of the point pairs $(\boldsymbol{b}_i, \boldsymbol{a}_i)$ may be well-approximated by the measurement model (1), while others

are *outliers* and largely deviate from (1), either because the pairs of points are incorrectly associated by the algorithm that establishes the correspondences, or because they do not lie on a static portion of the scene. In this case, the measurement model becomes:

$$\boldsymbol{b}_i = \theta_i(\boldsymbol{R}\boldsymbol{a}_i + \boldsymbol{t}) + (1-\theta_i)\boldsymbol{o} + \boldsymbol{\epsilon}, \quad i = 1, \dots, n \tag{3}$$

where the (unknown) binary variable $\theta_i \in \{0, 1\}$ decides whether $\boldsymbol{b}_i$ is a rigid transformation of $\boldsymbol{a}_i$ (if $\theta_i = 1$) or is an arbitrary vector $\boldsymbol{o}$, independent of $(\boldsymbol{R}, \boldsymbol{t})$ (if $\theta_i = 0$). A plethora of works has attacked robust registration with outliers. While we refer the reader to Section II and [13] for a more extensive discussion about related work, a popular approach is to resort to M-estimation, which attempts to compute an estimate for $(\boldsymbol{R}, \boldsymbol{t})$ by minimizing a robust loss function. For instance, the work [13] considers a truncated least squares loss:

$$\min_{\substack{\boldsymbol{R}\in\mathrm{SO}(3), \boldsymbol{t}\in\mathbb{R}^3, \\ \theta_i\in\{0;1\}, i=1,\dots,n}} \sum_{i=1}^{n} \theta_i \left\|\boldsymbol{b}_i - \boldsymbol{R}\boldsymbol{a}_i - \boldsymbol{t}\right\|^2 + (1-\theta_i)\bar{c}^2 \tag{4}$$

which computes a least squares estimate for measurements with small residual errors (*i.e.,* whenever $\|\boldsymbol{b}_i - \boldsymbol{R}\boldsymbol{a}_i - \boldsymbol{t}\| < \bar{c}$ the optimization forces $\theta_i = 1$ and the second summand disappears), while discarding measurements with large residuals (when $\theta_i = 0$, the objective becomes constant and the i-th measurement does not contribute to the estimate).

Multi-Model 3D Registration. The robust registration problem (4) looks for a single pose that explains the majority of correspondences, while disregarding the others as outliers. In this paper, we ask: can we instead find further patterns in the outliers? or, in other words, can we *simultaneously recover the motion of all objects present in the point clouds*? More formally, we assume the following generative model:

$$\boldsymbol{b}_i = \sum_{j=1}^{M} \theta_{i,j}(\boldsymbol{R}_j\boldsymbol{a}_i + \boldsymbol{t}_j) + \theta_{i,0}\,\boldsymbol{o} + \boldsymbol{\epsilon}, \quad i = 1, \dots, n \tag{5}$$

where for each measurement i, the vector $\boldsymbol{\theta}_i = [\theta_{i,0} \;\; \theta_{i,1} \;\; \dots \theta_{i,M}] \in \{0;1\}^{M+1}$ is an unknown binary vector with a single entry equal to 1, M it the number of objects (unknown a priori), and $(\boldsymbol{R}_j, \boldsymbol{t}_j)$ is the motion of the j-th object, for $j = 1, \dots, M$. In words, each point $\boldsymbol{b}_i$ in (5) is either generated by an object j (if $\theta_{i,j} = 1$ for a $j \in \{1, \dots, M\}$) or is an outlier (if $\theta_{i,0} = 1$). Clearly, when $M = 1$, eq. (5) falls back to the robust setup in (3).

Contribution. We propose an approach to solve the multi-model registration in eq. (5). The approach is based on an Expectation-Maximization (EM) algorithm that computes the assignments of measurements to objects (*i.e.,* the vectors $\boldsymbol{\theta}_i$ in (5)) and retrieves the pose $(\boldsymbol{R}_j, \boldsymbol{t}_j)$ for each object. The approach does not require prior knowledge of the number of objects M and can also accommodate additional constraints (*e.g.,* that distant objects are distinct, even if they exhibit similar motion). We provide a novel theoretical analysis of the algorithm that suggests that the EM scheme converges to the ground truth as long as the initialization of the vectors $\boldsymbol{\theta}_i$ is sufficient to capture all objects of interest. We evaluate the EM scheme in simulated and real datasets ranging from table-top scenes to large self-driving scenarios (Fig. 1) and demonstrate its effectiveness when combined with state-of-the-art scene flow methods to establish dense correspondences.

II. Related work

Robust Estimation in Robotics and Vision. Robust estimation is an active research area in robotics and vision [16], [17], [18] and has been attacked using different frameworks, including M-estimation [19], [20], [21], [22], consensus maximization [23], [24] (typically solved using sampling-based algorithms, such as RANSAC [25]), or graph-theoretic methods [13], [26], [27], [28]. We refer the reader to [13] for a review of robust 3D registration and to [18], [29] for an overview of robust estimation across robotics and vision.

List-Decodable Regression. While standard robust estimation computes an estimate that agrees with the majority of the measurements, recent work in robust statistics has focused on recovering an estimate from a handful of inliers hidden among an overwhelming amount of outliers, *e.g.,* [30], [31], [32], [33], [34]. In this regime, returning a single accurate hypothesis is information-theoretically impossible, and one has to compute a list of hypotheses to guarantee that at least one of them is accurate. This setup, typically referred to as *list-decodable* regression, was first studied in [31] and [32], which proposed and analyzed algorithms based on semidefinite relaxations. The work [18] observes that the algorithm in [31] can be easily adapted to solve a multi-model rotation-only registration problem; however, the resulting relaxation is impractically slow (*e.g.,* 3 minutes to solve a problem with 50 measurements).

Multi-Model Fitting in Computer Vision. Early work in computer vision has studied the problem of simultaneously recovering multiple models from noisy measurements. The corresponding literature includes clustering-based and optimization-based methods. Clustering-based methods span a variety of techniques, including hierarchical clustering [35], [36], kernel fitting [37], [38], matrix factorization [39], [40] and hypergraph partitioning [41], [42]. Optimization-based methods include generalizations of RANSAC to the multi-model setup, including Sequential RANSAC [43], Multi-RANSAC [44], and RANSACOV [45]. Other methods such as Pearl [46] and Progressive-X [47] take a step further by incorporating additional priors into the objective function.

Mixture of Linear Regression in Applied Mathematics. In the problem of learning a mixture of linear regressions, each measurement is generated from one of several unknown linear regression components, and one has to associate measurements to components and estimate the components. This problem is known to be NP hard in general [48]. However, under certain assumptions on the underlying distribution (*e.g.,* the regressors follow a standard Gaussian distribution, or there are only two mixture components), several approaches have successfully tackled the problem, including algorithms based on the method of moments [49], [50], alternating-minimization [48], [51], and Expectation-Maximization [52], [53], [54]. Contrary to this line of work, in this paper we do not make a Gaussian assumption on the regressors, and we consider a 3D registration problem rather than a linear regression setup.

Learning-based Methods for Motion Tracking. Learning-based methods have recently demonstrated excellent performance in 2D and 3D motion tracking. Optical flow methods [55], [56] estimate the pixel displacement between two frames, which can be used to segment moving objects from a video. Scene flow estimates dense 3D motion for each pixel from a pair of stereo or RGB-D frames. Other approaches, such as DRISF [57] and RigidMask [58], divide scene flow estimation into multiple subtasks and build modular networks to solve each subtask. RAFT-3D [59] computes the scene flow by using feature-level fusion. CamLiRAFT [60], [61] proposes a multi-stage pipeline to better fuse multi-modal information without suffering from accuracy loss due to voxelization.

III. An Expectation-Maximization Approach to Multi-Model Registration

The Expectation-Maximization (EM) algorithm [62] iteratively estimates parameters in statistical models given noisy data, by alternating an Expectation (E) step and a Maximization (M) step, which solves estimation problems in robotics [63], [64], [65], [66]. Here we use a variation of the EM algorithm known as the "Classification Expectation-Maximization" algorithm (*e.g.,* [52]), see Algorithm 1. We start by observing that finding the associations $\boldsymbol{\theta}_i$ can be equivalently thought of as a *clustering* problem, where we try to cluster together measurements corresponding to the same object. We will refer to our clusters with $H_j \subset S$, where S is the given set of correspondences $\{(\boldsymbol{a}_i, \boldsymbol{b}_i)\}_{i=1}^n$ and H_j indicates the correspondences (putatively) associated with object j. Note that this interpretation is consistent with (5), and by definition $H_j := \{(\boldsymbol{a}_i, \boldsymbol{b}_i) \in S \mid \theta_{i,j} = 1\}$. Accordingly, in Algorithm 1, rather than updating the indicator vectors $\boldsymbol{\theta}_i$, we update the clusters H_j for all objects j, at each iteration.

Algorithm 1: Expectation-Maximization (EM)

Input: Point clouds $S := \{(\boldsymbol{a}_i, \boldsymbol{b}_i)\}_{i=1}^n$, Initial clusters $\mathcal{H} := \{H_j \subset S \mid j \in [K]\}$, Distance threshold τ, Number of iterations T, Minimum cluster size $m_{\min}$

Output: $H_j, \boldsymbol{R}_j^{(r)}, \boldsymbol{t}_j^{(r)}, \forall j \in [K]$

1 **for** $r \in [T]$ **do**
2 % Compute a pose, weight, and variance for each cluster **for** $j \in [K]$ **do**
3 $(\boldsymbol{R}_j^{(r)}, \boldsymbol{t}_j^{(r)}) := \text{Horn}(H_j)$. $\pi_j^{(r)} := |H_j|/n$.
4 $E_j := \{\boldsymbol{b} - \boldsymbol{R}_j^{(r)}\boldsymbol{a} - \boldsymbol{t}_j^{(r)} \mid (\boldsymbol{a}, \boldsymbol{b}) \in H_j\}$
5 $\hat{\sigma}_j^{(r)} := \sqrt{\frac{1}{3}\text{tr}\left(\text{cov}(E_j)\right)}$
6 **end**
7 **E-step:** % Compute weighted likelihood:
8 **for** $j \in [K]$ *and* $i \in [n]$ **do**
9 $W_{i,j;\tau}^{(r)}$:= eq. (6)
10 **end**
11 % Remove small clusters H_j from $\mathcal{H}$
12 **for** $j \in [K]$ **do**
13 **if** $|H_j| < m_{\min}$ **then**
14 remove cluster j from $\mathcal{H}, \pi_j^{(r)}, \hat{\sigma}_j^{(r)}, W_{i,j;\tau}^{(r)}$ $K := K - 1$
15 **end**
16 **end**
17 **M-Step:** % Regenerate clusters according to likelihoods
18 **for** $i \in [n]$ **do**
19 **if** $j^\star = \arg\max_{j \in [k]} W_{i,j;\tau}^{(r)}$ **then**
20 add $(\boldsymbol{a}_i, \boldsymbol{b}_i)$ to cluster $H_{j^\star}$
21 **end**
22 **end**
23 **end**

Initialization. The algorithm takes as input, an initial guess for the clusters $\mathcal{H} := \{H_j \subset S \mid j \in [K]\}$ of the correspondences S, where for each object $j \in [K]$, H_j is the set of correspondences associated to j. In the next section, we provide conditions on the initialization under which the EM algorithm converges to the ground truth.

EM Algorithm. Each iteration of Algorithm 1 performs an E-step and M-step. At each iteration r, the algorithm first computes a transform $(\boldsymbol{R}_j^{(r)}, \boldsymbol{t}_j^{(r)})$ for each cluster (line 3); this is done using Horn's method [12] given the measurements in that cluster. The algorithm also computes a weight $\pi_j^{(r)}$ (quantifying the relative size of cluster j) and an intra-cluster variance $\hat{\sigma}_j^{(r)}$ for each cluster (lines 3-5). Then, the E-step estimates the posterior probability that the data point $(\boldsymbol{a}_i, \boldsymbol{b}_i)$ belongs to the cluster j according to the weighted likelihood:

$$W_{i,j;\tau}^{(r)} := \frac{\pi_j^{(r)} \phi^{(r)}(\boldsymbol{b}_i | \boldsymbol{a}_i)}{\sum_{j=1}^k \pi_j^{(r)} \phi^{(r)}(\boldsymbol{b}_i | \boldsymbol{a}_i)} \cdot \mathbf{1}(d_{\text{cluster}}(H_j, (\boldsymbol{a}_i, \boldsymbol{b}_i)) < \tau). \tag{6}$$

Here $\phi_j^{(r)}(\boldsymbol{b}_i | \boldsymbol{a}_i)$ denotes the likelihood of $\boldsymbol{b}_i - \boldsymbol{R}_j^{(r)}\boldsymbol{a}_i - \boldsymbol{t}_j^{(r)}$ with respect to the multivariate Gaussian density with mean $\mathbf{0}$ and covariance $\hat{\sigma}_j^2 \mathbf{I}_3$. The first term of the likelihood essentially quantifies how well the transformation $(\boldsymbol{R}_j^{(r)}, \boldsymbol{t}_j^{(r)})$ agrees with the correspondence $(\boldsymbol{b}_i, \boldsymbol{a}_i)$; the weighted likelihood also accounts for the cluster size (*i.e.,* the weight π_j). The second term $\mathbf{1}(d_{\text{cluster}}(H_j, (\boldsymbol{a}_i, \boldsymbol{b}_i)) < \tau)$ assigns zero likelihood to points that are far away (farther than a distance τ) from cluster j, where $d_{\text{cluster}}(H_j, (\boldsymbol{a}_i, \boldsymbol{b}_i)) := \min_{\boldsymbol{a}' \in H_j} \|\boldsymbol{a}' - \boldsymbol{a}_i\|$. This term avoids to cluster objects that have the same motion, but are far away from each other.

The M-step updates the assignment of samples to the clusters by assigning each $(\boldsymbol{b}_i, \boldsymbol{a}_i)$ to the cluster H_j maximizing $W_{i,j;\tau}^{(r)}$. This particular variation of the M-step is called the "Classification M-step", see, *e.g.,* [52]. Before executing the M-step, the algorithm removes overly small clusters (line 12).

We remark that Algorithm 1 is almost parameter free, and only requires setting the distance τ beyond which we consider two objects to be distinct, the minimum "size" $m_{\min}$ of what we would consider an object, and the number of iterations T. In particular, the weighted likelihood only depends on τ and does not require setting a noise bound, *e.g.,* as done in RANSAC. We also remark that the number of clusters K is estimated during the iterations, and ideally will converge to the true number of objects M, see eq. (5).

In the following section, we derive conditions under which Algorithm 1 converges to the ground truth clusters.

IV. Theoretical Analysis

In this section, we sketch a proof demonstrating that Algorithm 1 recovers the ground truth clusters under suitable condition on the initial clusters. In the following, we say that a set of points P is τ-*connected* if between any pair of points $\boldsymbol{x}, \boldsymbol{y} \in P$ there is a sequence of points in P such that each pair of consecutive points in the sequence are at most at distance τ from each other. Then we say that a set of correspondences $S := \{(\boldsymbol{a}_i, \boldsymbol{b}_i)\}_{i=1}^n$ is τ-connected if $\{\boldsymbol{a}_i\}_{i=1}^n$ is τ-connected.

We make the following assumptions on the ground truth.

Definition 1 (Ground Truth). *We are given a set of correspondences $S := \{(\boldsymbol{a}_i, \boldsymbol{b}_i)\}_{i=1}^n$, which can be partitioned into M τ-connected parts $G_1, \ldots, G_M$ such that there are functions $g_1, \ldots, g_M$ of the form $g_j = \boldsymbol{R}_j \boldsymbol{a} + \boldsymbol{t}_j$ where $(\boldsymbol{R}_j, \boldsymbol{t}_j)$ is a rigid transformation, satisfying:*

1) ***Uniform Bounded Noise:*** *For all $j \in [M]$ and $(\boldsymbol{a}, \boldsymbol{b}) \in G_j$, $g_j(\boldsymbol{a}) + \boldsymbol{\epsilon} = \boldsymbol{b}$, where $\boldsymbol{\epsilon}$ is drawn from the uniform distribution over $[-\sigma, \sigma]^3$.*
2) ***Object Separation:*** *For all distinct $i, j \in [M]$, $\min_{(\boldsymbol{a},\boldsymbol{b}) \in G_i, (\boldsymbol{a}', \boldsymbol{b}') \in G_j} \|\boldsymbol{a} - \boldsymbol{a}'\| > \tau$.*
3) ***Bounded Point Cloud:*** *For all $i \in [n]$, $\|\boldsymbol{a}_i\| \leq B$.*
4) ***Outliers:*** *Some of the samples may be "outliers". We say that a point $(\boldsymbol{a}_o, \boldsymbol{b}_o)$ is an outlier if, for all $j \in [M]$, $d_{cluster}(G_j, (\boldsymbol{a}_o, \boldsymbol{b}_o)) > \tau$.*

For Algorithm 1 to converge, we will require good initial clustering (in Section V, we show that using a simple Euclidean clustering or more modern alternatives, like SegmentAnything (SAM) [67], suffices). Here we formalize what it means to have good initial clustering:

Definition 2 (Good Clustering). *In the setting of Definition 1, we say that the initial clustering $\mathcal{H} := \{H_1, \ldots, H_K\}$ with $K \geq M$ is (τ, α, m_0)-good, if it is a partition of the correspondences S (as defined in Definition 1) satisfying:*

1) ***τ-connected:*** *For all $j \in [K]$, H_j is τ-connected.*
2) ***Large Initial Clusters:*** *For all $j \in [K]$, $|H_j| \geq m_0$.*
3) ***Identifying cluster:*** *For each ground truth cluster G, let $\mathcal{H}_G := \{H \in \mathcal{H} \mid |H \cap G| > 0\}$ and $H^* := \arg\max_{H \in \mathcal{H}_G} |H|$. Then, for some $\alpha > 1$, $|H^*| > \alpha \max_{H \in \mathcal{H}_G \setminus \{H^*\}} |H|$.*

Intuitively, the last condition captures the idea that for any ground truth cluster G, the largest cluster having a nonzero intersection with G, namely H^*, is notably larger than all other initial clusters having a nonzero intersection with G.

Theorem 3 (Expectation-Maximization Guarantee). *In the setting of Definition 1, assume that the initial clustering $\mathcal{H}$ is (τ, α, m_0)-good in the sense of Definition 2, for some sufficiently large m_0. Then, running Algorithm 1, with high probability (dependent on m_0), returns $\mathcal{H}'$, a partition of the set of correspondences S, containing each of the ground truth clusters (*i.e., *the objects and background).*

Intuition: The proof of theorem 3 follows by noting that the initial clustering results in a partition of each ground truth cluster such that one of the partitions is notably larger than the rest. As the algorithm progresses, the M-step assigns more points to the biggest estimated cluster until it exactly matches the ground truth object which contains it. This happens because the likelihood $W_{i,j;\tau}$ is maximized by the largest cluster, due to the presence of the weight π_j (*i.e.,* the cluster size).

Proof Sketch: We make the following observations, which together imply that the final clusters produced by Theorem 3 include the ground truth clusters. First, each ground truth-cluster is partitioned by the initial clustering. This is because the τ-connected subsets of the data either consist of samples that are entirely contained in one of the G_j or (possibly) the set of outliers. Since the initial clustering $\mathcal{H}$ exclusively consists of τ-connected subsets, each element of $\mathcal{H}$ is either a subset of G_j for some j or entirely consists of outliers.

Now suppose the weights for the clusters are given by $\pi_1, \ldots, \pi_K$. Without loss of generality, suppose $\{H_1, \ldots, H_t\}$ form a partition of G_1 with $\pi_1 \geq \ldots \geq \pi_t$. Since $\mathcal{H}$ is (τ, α, m_0)-good, we know that $\pi_1 > \alpha \pi_2$ because of Item 3 in Definition 2. Our second claim is that in each iteration of the M-step, elements of S that are τ-close to the largest cluster H_1 are assigned to H_1. To see this, consider a point $(\boldsymbol{a}_i, \boldsymbol{b}_i) \in G_1 \setminus H_1$ which is τ-close to H_1 (if no such point exists, then $H_1 = G_1$). We show that for the point $(\boldsymbol{a}_i, \boldsymbol{b}_i)$, the likelihood $W_{i,j;\tau}$ is maximized when $j = 1$.

Since Horn's Method is consistent (*i.e.,* for a sufficiently large sample size, the algorithm converges to the true solution in the presence of zero-mean noise) and the domain of the point-cloud is bounded by a ball of radius B, $h_j := \boldsymbol{R}_j^{(r)} \boldsymbol{a}_i + \boldsymbol{t}_j^{(r)}$ and σ_j estimate g_1 and σ up to an additive error of ϵ', according to standard concentration results. Choosing sufficiently large $m_{\min}$ and m_0 ensures that the sample size is large and consequently that ϵ' is small. This is enforced by the algorithm, which deletes candidate clusters of size less than $m_{\min}$. Now note that $\arg\max_j W_{i,j;\tau} = \arg\max_j \pi_j \phi_j(\boldsymbol{b}_i \mid \boldsymbol{a}_i) = \arg\max_j \pi_j \exp(-\|\boldsymbol{b}_i - h_j(\boldsymbol{a}_i)\|^2 / {\sigma_j}^2)/\sigma_j$. Since ϵ' is sufficiently small for large sample size, we get $W_{i,j;\tau} = \pi_j(1 \pm \epsilon) \exp(-\|\boldsymbol{b}_i - g_1(\boldsymbol{a}_i)\|^2 / \sigma^2)/\sigma$, where ϵ is a function of ϵ', σ, and B. Since $\exp(-\|\boldsymbol{b}_i - g_1(\boldsymbol{a}_i)\|^2 / \sigma^2)/\sigma$ is a constant with respect to j, it does not affect the maximization. This implies $\arg\max_j W_{i,j;\tau}$ is essentially determined by $\arg\max_j \pi_j(1 \pm \epsilon)$. Choosing small ϵ to ensure $(1+\epsilon)/(1-\epsilon) < \alpha$ and recalling that $\pi_1 > \alpha \pi_2$, we see $\arg\max_j W_{i,j;\tau} = \arg\max_j \pi_j(1 \pm \epsilon) = 1$. Since π_1 keeps increasing in size at each iteration, eventually all the elements of G_1 will collect into H_1. □

Remark 4 (Novelty). *Similar to theoretical analyses of the EM algorithm in prior work in the context of learning a mixture of linear regressions, we require a good initialization, see,* e.g., *[54]. However, contrary to related work, we do not assume the regressors (roughly speaking, the vectors $\boldsymbol{a}_i$) to follow a Gaussian distribution, which would be too strict in practical multi-modal registration problems.*

V. EXPERIMENTS

We conduct a wide range of experiments on both synthetic and real-world datasets. The synthetic datasets are PASCAL3D+ [68] and FlyingThings3D [69] and the real-world dataset is from KITTI [70]. We test the proposed approach against Sequential RANSAC [43] and T-Linkage [35], and show it dominates these baselines on the multi-model 3D registration problem, and its performance is further improved by using SegmentAnything (SAM) [67] as initialization.

A. Baselines and Initialization

We compare our approach against T-Linkage [35] and Sequential RANSAC (SRANSAC) [43]. We also include a Naive baseline that applies Horn's method [12] to compute a pose estimate for each initial cluster.

T-Linkage [35]. T-Linkage computes the distance between pairs of clusters and iteratively merges the closest pair of clusters. For each cluster j (with associated pose $(\boldsymbol{R}_j, \boldsymbol{t}_j)$), T-Linkage defines a *preference function* for point i as

$$\psi_{j,i} = \mathrm{e}^{-d_{j,i}/\tau_t} \text{ if } d_{j,i} \leq 5\tau_t \text{ or } 0 \text{ otherwise}, \quad (7)$$

where $d_{j,i} := \|\boldsymbol{b}_i - \boldsymbol{R}_j \boldsymbol{a}_i - \boldsymbol{t}_j\|$. It then uses the preference functions to compute distances between pairs of clusters, and terminates when the distance between every pair is large.

Sequential RANSAC [43]. This baseline sequentially applies RANSAC and tries to recover one object at a time. After each RANSAC execution, the correspondences selected as inliers by RANSAC are used to compute a pose estimate and then removed to facilitate the search for other objects.

Initialization. SRANSAC does not require an initial guess for the clusters, while T-Linkage and our approach do. In PASCAL3D+ we use Euclidean clustering on $\{\boldsymbol{a}_i\}_{i=1}^n$ to obtain initialization for both T-Linkage and EM. In KITTI and FlyingThing3D, we ablate the effect of initial clustering, by initializing T-Linkage [35] and our method with SegmentAnything (SAM) [67] or Euclidean clustering. We tune both initialization approaches to generate around 100 clusters.

B. Metrics

We use three main performance metrics for evaluation.

Per-Point Error (↓). This metric evaluates the average mismatch between the ground-truth and estimated point clouds of each object (including the background). It first segments the first point cloud according to the ground-truth clusters to obtain $\boldsymbol{a}^{(i)}, i = 1, \ldots, M$, and then applies the ground-truth transformation to each object to obtain $\boldsymbol{b}^{(i)}, i = 1, \ldots, M$. It repeats the same process using the estimated clusters and transforms to compute $\hat{\boldsymbol{b}}^{(j)}, j = 1, \ldots, K$. Then, each estimated object i is associated with the ground truth object j that has the largest intersection with i (*i.e.,* the largest number of points in common), and the Chamfer distance between i and j is recorded. The point error is defined as the average Chamfer distance across objects.

Rotation and Translation Error (↓). This metric evaluates the distance between the estimated and ground-truth poses. For each estimated object H_j, we find every ground-truth object G_k that has a non-zero intersection with H_j. Then, we compute the *translation error* as the Euclidean distance between the estimated and ground-truth translation; the *rotation error* is the angular distance [71] between the estimated and ground-truth rotation. The final translation and rotation errors are the weighted averages of the errors, where the weights are computed as $|H_j \cap G_k|/|H_j|$.

Intersection over Union (↑). This metric evaluates the quality of the estimated clusters by assigning a ground-truth cluster to each estimated cluster with the largest intersection and calculating the average Intersection over Union (IoU).

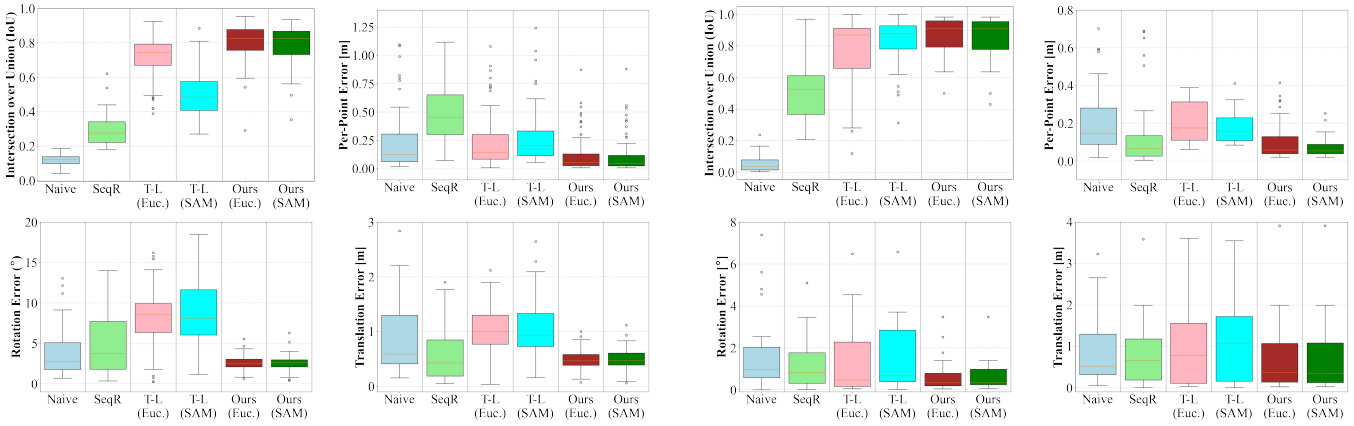
TABLE I: Results for synthetic PASCAL3D+ dataset. Experiment 1: Noiseless. Experiment 2: with additive Gaussian noise. Experiment 3: with additive Gaussian noise and 2 objects with the same motion.

	Metric (Mean)	Method				
		Naive	SRANSAC	T-Linkage	Ours (Vanilla)	Ours
1	Per-point Error [m]	0.0454	**9.56e-15**	0.163	**9.56e-15**	**9.56e-15**
	Rotation error [deg]	46.8	**8.69e-7**	19.6	**8.69e-7**	**8.69e-7**
	Translation error [m]	0.487	**3.81e-15**	0.242	**3.81e-15**	**3.81e-15**
	IoU	0.698	**1.0**	0.723	**1.0**	**1.0**
2	Per-point Error [m]	0.0550	0.332	0.199	0.0135	**0.00516**
	Rotation error [deg]	81.6	74.2	27.6	2.42	**1.53**
	Translation error [m]	0.858	0.796	0.287	0.0286	**0.0165**
	IoU	0.668	0.450	0.785	0.908	**0.964**
3	Per-point Error [m]	0.0104	0.321	0.215	0.00860	**0.00776**
	Rotation error [deg]	75.3	67.3	15.9	1.48	**1.12**
	Translation error [m]	5.83	1.33	0.300	**0.0200**	0.0499
	IoU	0.755	0.398	0.729	0.825	**0.970**

C. PASCAL3D+

Experimental Setup. PASCAL3D+ [68] is a synthetic dataset for 3D object understanding. We choose 7 objects from the dataset and downsample the vertices of their CAD models to form object point clouds (22,395 points overall). To generate point cloud pairs $\{(\boldsymbol{a}_i, \boldsymbol{b}_i)\}_{i=1}^n$, we randomly sample 7 transformation matrices and apply them to each point cloud. In this dataset, there are no outliers, and we want to test the capability of the compared techniques to tell the 7 objects apart and estimate their motion (Experiment 1). We also repeat in two more challenging settings, where we add zero-mean Gaussian noise with standard deviation 0.03m to the point cloud $\{(\boldsymbol{b}_i)\}_{i=1}^n$ (Experiment 2), and where, in addition to the noise, we assume 2 of the 7 objects have the same motion (Experiment 3). We test our method with and without the distance term in (6); we denote the latter as "Ours (Vanilla)". We use Euclidean clustering (with 100 clusters) to obtain the initial clusters for Naive, T-Linkage, and our methods. For our method, we set $\tau = 1.5$m, $m_{\min} = 4$, $T = 10$. We set $\tau_t = 0.2$m in T-Linkage. For SRANSAC, we use 0.01m as the inlier threshold for experiment 1 (noiseless case) and 0.5m for the others and use 1000 max iterations. Results are averaged over 100 runs.

Results. Table I shows the results obtained with the compared techniques in the three settings described above. In the noiseless case, SRANSAC and our methods achieve perfect scores (*i.e.,* errors are numerically zero). However, in the noisy experiments, our method outperforms other methods across metrics. This is because our method adjusts the noise variance at each iteration and computes the likelihood function accordingly. In Experiment 3, where we enforce two objects' motion to be identical, although both variants of our methods still estimate the poses with similar errors, the one with the distance term ("ours") stands out in terms of IoU: with the distance term, our method can identify two objects that are relatively far apart even if they have the same



(a) FlyingThings3D Results

(b) KITTI Results

Fig. 2: Results (IoU, per-point error, rotation error, and translation error) on (a) FlyingThings3D and (b) KITTI. We evaluate two variations of our method with two different initializations (SAM and Euclidean) and four baselines: Naive, Sequential RANSAC (SeqR), T-Linkage with SAM initialization (T-L (SAM)), and T-Linkage with Euclidean initialization (T-L (Euc.)).

motion. Naive reflects the quality of the initial clustering and typically leads to poor IoU. T-Linkage consistently improves over Naive in all experiments, but is not competitive with SRANSAC and our methods. The runtime for SRANSAC is 30ms, while T-Linkage takes about 2s. Our method without the distance term takes about 800ms in Python on a Macbook Pro with M1 Pro chip. Adding the distance term in our method increases the runtime to a couple of seconds, since we have not optimized the distance computation.

D. FlyingThings3D

Experimental Setup. The FlyingThings3D [69] dataset has randomly moving objects from ShapeNet [72]. The dataset provides RGB images, segmentation masks, depth maps, and disparity maps. We construct the point clouds with ground-truth scene flows by back-projecting pixels to 3D points using disparity maps. In our method, to construct correspondences, we use the state-of-the-art scene flow model CamliRAFT [60] on RGB images and downsampled point cloud $\boldsymbol{a}$ with 32,768 points to get predicted scene flow $\boldsymbol{f}$ and add predicted scene flow on the point cloud to get the next frame's point cloud ($\boldsymbol{b} = \boldsymbol{a} + \boldsymbol{f}$). Using the segmentation masks, we obtain ground-truth poses for each object cluster by running Horn's method on each object's point cloud and its counterpart displaced by the ground-truth scene flow.

We set the distance threshold in our method to be 5m, since the diameters of most objects are about 3m in the FlyingThings3D dataset. Then, for T-Linkage we set the constant τ_t to be 1m in (7). For SRANSAC, we use 0.2 as the residual threshold and 1000 (maximum) iterations.

Results. We show the comparison between our method and other baselines on FlyingThings3D in Fig. 2a. As shown in the boxplots, our method outperforms other baselines. In particular, our method recovers the object clusters consistently (highest IoU) and works well with different kinds of initialization methods. SRANSAC achieves a very low IoU score because it recovers over 50 clusters which is a lot more than the ground truth (about 10 clusters). T-Linkage clusters the point cloud better using Euclidean clustering which is considered as a weaker initialization method than SAM. This is because, in the FlyingThings3D dataset, since the objects are relatively far apart, the initial clusters from Euclidean clustering are significantly better than SAM. For our method, IoU scores are almost the same. This shows that our method is less sensitive to the quality of initial clustering.

E. KITTI

Experimental Setup. The KITTI [73], [70] scene flow dataset consists of 400 scenes split into training, validation, and testing datasets with RGB images and depth. We only use the validation set because no ground-truth optical flow is provided in the testing set for us to evaluate.

To run our method, we follow the same steps done for the FlyingThings3D dataset to construct correspondences and obtain initial clustering. Since KITTI only provides semantic masks, to compute ground truth poses and (instance-level) clusters we perform Euclidean clustering only on the point clouds with the car label in the ground-truth semantic segmentation and merge everything else as background, resulting in a masked point cloud with a cluster for each car and another cluster for the background. Then, we calculate the pose for each cluster, similar to FlyingThings3D. We use the same parameters as in Section V-D since the main moving objects here are cars which are also about 3m long.

Results. In Fig. 2b, we compare our method and other baselines on the KITTI dataset, where our method outperforms. SRANSAC still suffers from over-segmenting as in the FlyingThings3D experiment. Since SAM performs better on KITTI (compared to Euclidean clustering), T-Linkage exhibits slightly better performance with SAM.

VI. Conclusion

We investigated a variation of the 3D registration problem, named *multi-model 3D registration*, that simultaneously recovers the motion of multiple objects in point clouds. We proposed a simple approach based on Expectation-Maximization (EM) and established theoretical conditions under which the EM scheme recovers the ground truth. We evaluated the EM scheme in both synthetic and real-world datasets ranging from table-top scenes to large self-driving scenarios and demonstrated its effectiveness.

REFERENCES

[1] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments," *Intl. J. of Robotics Research*, vol. 31, no. 5, pp. 647–663, 2012.

[2] G. Blais and M. D. Levine, "Registering multiview range data to create 3d computer objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, no. 8, pp. 820–824, 1995.

[3] S. Choi, Q. Y. Zhou, and V. Koltun, "Robust reconstruction of indoor scenes," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 5556–5565.

[4] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3D object recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 998–1005.

[5] J. M. Wong, V. Kee, T. Le, S. Wagner, G. L. Mariottini, A. Schneider, L. Hamilton, R. Chipalkatty, M. Hebert, D. M. S. Johnson *et al.*, "Segicp: Integrated deep semantic segmentation and pose estimation," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 5784–5789.

[6] A. Zeng, K. T. Yu, S. Song, D. Suo, E. Walker, A. Rodriguez, and J. Xiao, "Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1386–1383.

[7] M. A. Audette, F. P. Ferrie, and T. M. Peters, "An algorithmic overview of surface registration techniques for medical imaging," *Med. Image Anal.*, vol. 4, no. 3, pp. 201–217, 2000.

[8] G. K. L. Tam, Z. Q. Cheng, Y. K. Lai, F. C. Langbein, Y. Liu, D. Marshall, R. R. Martin, X. F. Sun, and P. L. Rosin, "Registration of 3d point clouds and meshes: a survey from rigid to nonrigid." *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 7, pp. 1199–1217, 2013.

[9] J. Bazin, Y. Seo, R. Hartley, and M. Pollefeys, "Globally optimal inlier set maximization with unknown rotation and focal length," in *European Conf. on Computer Vision (ECCV)*, 2014, pp. 803–817.

[10] G. Wahba, "A least squares estimate of satellite attitude," *SIAM review*, vol. 7, no. 3, pp. 409–409, 1965.

[11] K. Arun, T. Huang, and S. Blostein, "Least-squares fitting of two 3-D point sets," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 9, no. 5, pp. 698–700, sept. 1987.

[12] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Opt. Soc. Amer.*, vol. 4, no. 4, pp. 629–642, Apr 1987.

[13] H. Yang, J. Shi, and L. Carlone, "TEASER: Fast and Certifiable Point Cloud Registration," *IEEE Trans. Robotics*, vol. 37, no. 2, pp. 314–333, 2020, extended arXiv version 2001.07715 (pdf).

[14] J. L. Barron, D. J. Fleet, and S. S. Beuachemin, "Performance of optical flow techniques," *Intl. J. of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.

[15] S. Vedula, P. Rander, R. Collins, and T. Kanade, "Three-dimensional scene flow," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 27, no. 3, pp. 475–480, 2005.

[16] L. Peng, C. Kümmerle, and R. Vidal, "On the convergence of irls and its variants in outlier-robust estimation," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 17 808–17 818.

[17] A. Barik and J. Honorio, "Outlier-robust estimation of a sparse linear model using invexity," 2023.

[18] L. Carlone, "Estimation contracts for outlier-robust geometric perception," *Foundations and Trends (FnT) in Robotics, arXiv preprint: 2208.10521*, 2023, (pdf).

[19] K. M. Tavish and T. D. Barfoot, "At all costs: A comparison of robust cost functions for camera correspondence outliers," in *Conf. Computer and Robot Vision*. IEEE, 2015, pp. 62–69.

[20] M. J. Black and A. Rangarajan, "On the unification of line processes, outlier rejection, and robust statistics with applications in early vision," *Intl. J. of Computer Vision*, vol. 19, no. 1, pp. 57–91, 1996.

[21] H. Yang, P. Antonante, V. Tzoumas, and L. Carlone, "Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection," *IEEE Robotics and Automation Letters (RA-L)*, vol. 5, no. 2, pp. 1127–1134, 2020, arXiv preprint:1909.08605 (with supplemental material), (pdf).

[22] H. Yang and L. Carlone, "Certifiably optimal outlier-robust geometric perception: Semidefinite relaxations and scalable global optimization," *IEEE Trans. Pattern Anal. Machine Intell.*, 2022, (pdf).

[23] T.-J. Chin, Z. Cai, and F. Neumann, "Robust fitting in computer vision: Easy or hard?" in *European Conf. on Computer Vision (ECCV)*, 2018.

[24] P. Antonante, V. Tzoumas, H. Yang, and L. Carlone, "Outlier-robust estimation: Hardness, minimally tuned algorithms, and applications," *IEEE Trans. Robotics*, vol. 38, no. 1, pp. 281–301, 2021, (pdf).

[25] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography," *Commun. ACM*, vol. 24, pp. 381–395, 1981.

[26] J. Shi, H. Yang, and L. Carlone, "ROBIN: a graph-theoretic approach to reject outliers in robust estimation using invariants," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2021, arXiv preprint: 2011.03659, (pdf).

[27] A. P. Bustos, T.-J. Chin, F. Neumann, T. Friedrich, and M. Katzmann, "A practical maximum clique algorithm for matching with pairwise constraints," *arXiv preprint arXiv:1902.01534*, 2019.

[28] O. Enqvist, K. Josephson, and F. Kahl, "Optimal correspondences from pairwise constraints," in *Intl. Conf. on Computer Vision (ICCV)*, 2009, pp. 1295–1302.

[29] M. Bosse, G. Agamennoni, and I. Gilitschenski, "Robust estimation and applications in robotics," *Foundations and Trends in Robotics*, vol. 4, no. 4, pp. 225–269, 2016.

[30] M. Charikar, J. Steinhardt, and G. Valiant, "Learning from untrusted data," in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, ser. STOC 2017, 2017, pp. 47–60.

[31] S. Karmalkar, A. Klivans, and P. Kothari, "List-decodable linear regression," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 32, 2019.

[32] P. Raghavendra and M. Yau, "List decodable learning via sum of squares," in *Proceedings of the Thirty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '20, 2020, p. 161–180.

[33] I. Diakonikolas, D. Kane, and D. Kongsgaard, "List-decodable mean estimation via iterative multi-filtering," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9312–9323, 2020.

[34] Y. Cherapanamjeri, S. Mohanty, and M. Yau, "List decodable mean estimation in nearly linear time," in *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2020, pp. 141–148.

[35] L. Magri and A. Fusiello, "T-linkage: A continuous relaxation of j-linkage for multi-model fitting," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3954–3961.

[36] R. Toldo and A. Fusiello, "Robust multiple structures estimation with j-linkage," in *Computer Vision – ECCV 2008*, D. Forsyth, P. Torr, and A. Zisserman, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 537–547.

[37] T.-J. Chin, H. Wang, and D. Suter, "Robust fitting of multiple structures: The statistical learning approach," in *2009 IEEE 12th International Conference on Computer Vision*, 2009, pp. 413–420.

[38] T.-J. Chin, D. Suter, and H. Wang, "Multi-structure model selection via kernel optimisation," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3586–3593.

[39] L. Magri and A. Fusiello, "Multiple structure recovery via robust preference analysis," *Image and Vision Computing*, vol. 67, pp. 1–15, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S026288561730152X

[40] M. Tepper and G. Sapiro, "Nonnegative matrix underapproximation for robust multiple model fitting," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 655–663.

[41] S. Lin, G. Xiao, Y. Yan, D. Suter, and H. Wang, "Hypergraph optimization for multi-structural geometric model fitting," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 8730–8737, Jul. 2019. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/4897

[42] P. Purkait, T.-J. Chin, A. Sadri, and D. Suter, "Clustering with hypergraphs: The case for large hyperedges," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 9, pp. 1697–1711, 2017.

[43] P. H. Torr, "Geometric motion segmentation and model selection," *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 356, no. 1740, pp. 1321–1340, 1998.

[44] M. Zuliani, C. Kenney, and B. Manjunath, "The multiransac algorithm and its application to detect planar homographies," in *IEEE International Conference on Image Processing 2005*, vol. 3, 2005, pp. III–153.

[45] L. Magri and A. Fusiello, "Multiple models fitting as a set coverage

problem," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3318–3326.

[46] H. Isack and Y. Boykov, "Energy-based geometric multi-model fitting," *International Journal of Computer Vision*, vol. 97, pp. 123–147, 04 2012.

[47] D. Baráth and J. Matas, "Progressive-x: Efficient, anytime, multi-model fitting algorithm," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 3779–3787.

[48] X. Yi, C. Caramanis, and S. Sanghavi, "Alternating minimization for mixed linear regression," in *International Conference on Machine Learning*. PMLR, 2014, pp. 613–621.

[49] H. Sedghi, M. Janzamin, and A. Anandkumar, "Provable tensor methods for learning mixtures of generalized linear models," in *Artificial Intelligence and Statistics*. PMLR, 2016, pp. 1223–1231.

[50] Y. Li and Y. Liang, "Learning mixtures of linear regressions with nearly optimal complexity," in *Conference On Learning Theory*. PMLR, 2018, pp. 1125–1144.

[51] X. Yi, C. Caramanis, and S. Sanghavi, "Solving a mixture of many random linear equations by tensor decomposition and alternating minimization," *CoRR*, vol. abs/1608.05749, 2016. [Online]. Available: http://arxiv.org/abs/1608.05749

[52] S. Faria and G. Soromenho, "Fitting mixtures of linear regressions," *Journal of Statistical Computation and Simulation*, vol. 80, no. 2, pp. 201–225, 2010. [Online]. Available: https://doi.org/10.1080/00949650802590261

[53] J. M. Klusowski, D. Yang, and W. D. Brinda, "Estimating the coefficients of a mixture of two linear regressions by expectation maximization," *IEEE Transactions on Information Theory*, vol. 65, no. 6, pp. 3515–3524, 2019.

[54] J. Kwon and C. Caramanis, "Em converges for a mixture of many linear regressions," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, S. Chiappa and R. Calandra, Eds., vol. 108. PMLR, 26–28 Aug 2020, pp. 1727–1736. [Online]. Available: https://proceedings.mlr.press/v108/kwon20a.html

[55] Z. Teed and J. Deng, "Raft: Recurrent all-pairs field transforms for optical flow," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 402–419.

[56] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," *CoRR*, vol. abs/1612.01925, 2016. [Online]. Available: http://arxiv.org/abs/1612.01925

[57] W.-C. Ma, S. Wang, R. Hu, Y. Xiong, and R. Urtasun, "Deep rigid instance scene flow," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3609–3617.

[58] G. Yang and D. Ramanan, "Learning to segment rigid motions from two frames," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 1266–1275.

[59] Z. Teed and J. Deng, "Raft-3d: Scene flow using rigid-motion embeddings," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 8371–8380.

[60] H. Liu, T. Lu, Y. Xu, J. Liu, W. Li, and L. Chen, "Camliflow: Bidirectional camera-lidar fusion for joint optical flow and scene flow estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5791–5801.

[61] H. Liu, T. Lu, Y. Xu, J. Liu, and L. Wang, "Learning optical flow and scene flow with bidirectional camera-lidar fusion," 2023.

[62] T. K. Moon, "The expectation-maximization algorithm," *Signal processing magazine, IEEE*, vol. 13, no. 6, pp. 47–60, 1996.

[63] B. Eckart, K. Kim, and J. Kautz, "Hgmr: Hierarchical gaussian mixtures for adaptive 3d registration," in *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XV*. Berlin, Heidelberg: Springer-Verlag, 2018, p. 730–746. [Online]. Available: https://doi.org/10.1007/978-3-030-01267-0_43

[64] J. G. Rogers, A. J. Trevor, C. Nieto-Granda, and H. I. Christensen, "Slam with expectation maximization for moveable object tracking," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 2077–2082.

[65] V. Indelman, E. Nelson, N. Michael, and F. Dellaert, "Multi-robot pose graph localization and data association from unknown initial relative poses via expectation maximization," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 593–600.

[66] S. Bowman, N. Atanasov, K. Daniilidis, and G. Pappas, "Probabilistic data association for semantic SLAM," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2017, pp. 1722–1729.

[67] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[68] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond pascal: A benchmark for 3d object detection in the wild," in *IEEE Winter Conf. on Appl. of Computer Vision*. IEEE, 2014, pp. 75–82.

[69] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4040–4048.

[70] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[71] R. Hartley, J. Trumpf, Y. Dai, and H. Li, "Rotation averaging," *IJCV*, vol. 103, no. 3, pp. 267–305, 2013.

[72] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.

[73] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3061–3070.