Joint Alignment of Multiple Point Sets with Batch and Incremental Expectation-Maximization

Georgios Evangelidis, Radu Horaud

Joint Alignment of Multiple Point Sets with Batch and Incremental Expectation-Maximization

Georgios D. Evangelidis and Radu Horaud

***Abstract*—This paper addresses the problem of registering multiple point sets. Solutions to this problem are often approximated by repeatedly solving for pairwise registration, which results in an uneven treatment of the sets forming a pair: a model set and a data set. The main drawback of this strategy is that the model set may contain noise and outliers, which negatively affects the estimation of the registration parameters. In contrast, the proposed formulation treats all the point sets on an equal footing. Indeed, all the points are drawn from a central Gaussian mixture, hence the registration is cast into a clustering problem. We formally derive batch and incremental EM algorithms that robustly estimate both the GMM parameters and the rotations and translations that optimally align the sets. Moreover, the mixture's means play the role of the registered set of points while the variances provide rich information about the contribution of each component to the alignment. We thoroughly test the proposed algorithms on simulated data and on challenging real data collected with range sensors. We compare them with several state-of-the-art algorithms, and we show their potential for surface reconstruction from depth data.**

***Index Terms*—Point registration, expectation maximization, mixture models, joint alignment**

I. Introduction

The registration of point sets is an essential methodology in computer vision, computer graphics, robotics, and medical image analysis. The vast majority of existing techniques solve the pairwise (two sets) registration problem [1]–[6], while the multiple-set registration problem has comparatively received less attention [7]–[9]. Solutions to this problem are often approximated by repeatedly solving for pairwise registration, either sequentially [10]–[12], or via a *one-versus-all* strategy [13]–[15].

Independently of the particular two-set registration algorithm that is used, the above mentioned approximate solutions have their own limitations. On the one hand, sequential strategies suffer from the well known drift accumulation owing to the chain-based optimization, i.e. sequential registration between pairs of point sets. On the other hand, one-versus-all strategies lead to a biased estimator since the registration is governed by a single reference set. In addition, both strategies lack closed-loop information and one needs to further consider this constraint. Therefore, an unbiased solution that treats all the point sets on an equal footing and that implicitly enforces a loop constraint is particularly desirable.

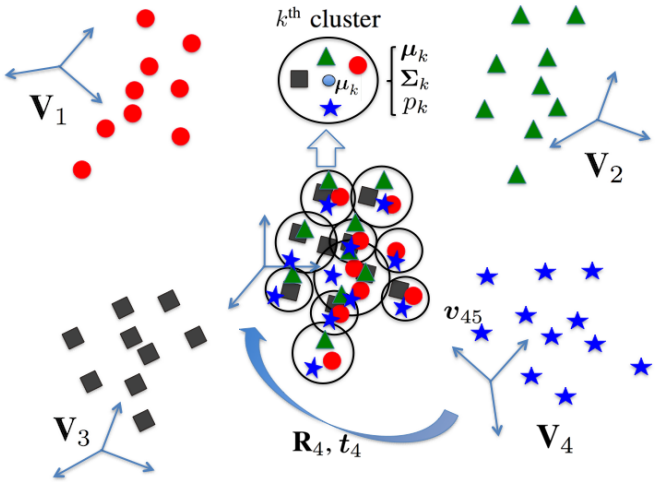


Fig. 1: The proposed *joint registration* method assumes that all points from all sets, e.g. $\mathbf{V}_1$ to $\mathbf{V}_4$ are realizations of the same mixture (shown in the center). An observed point, e.g. $\boldsymbol{v}_{45} \in \mathbf{V}_4$, once rotated and translated from the set-centered coordinate frame to the mixture-centered coordinate frame ($\mathbf{R}_4$ and $\boldsymbol{t}_4$) is assigned to the k^{th} mixture component defined by $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ and p_k. As shown on the figure, the estimated mixture is not associated to any of the point sets, as is the case with *pairwise registration* methods.

Such an unbiased solution is targeted by motion averaging approaches that build on pairwise registration schemes and aim to evenly distribute the total error across the network of point sets, either as a post-processing step [16] or as an over-successive registration between pairs of point sets [9]. We rather aim to *jointly* register all the point sets and not re-distribute the error from a pairwise registration. To this end, we propose a generative approach to the joint registration of multiple point sets. An arbitrary number of point sets, observed from different sensor locations, are assumed to be generated from a *single* Gaussian mixture model (GMM). The problem is cast into a data clustering problem which, in turn, is solved via maximum likelihood and leads to an *expectation maximization* (EM) algorithm, whereby both the mixture and registration parameters are optimally estimated. We present batch and incremental EM algorithms: both can deal with point sets of

Georgios D. Evangelidis, DAQRI International, Dublin, Ireland, E-mail: georgios.evangelidis@daqri.com (This work was done while the author was with INRIA, Grenoble Rhône-Alpes, France)

Radu Horaud, INRIA Grenoble Rhône-Alpes, Montbonnot Saint-Martin, France, E-mail: radu.horaud@inria.fr

Funding from Agence Nationale de la Recherche (ANR) MIXCAM project #ANR-13-BS02-0010-01 and from the European Union FP7 ERC Advanced Grant VHIA #340113 is greatly acknowledged.

different cardinalities and contaminated by noise and outliers.

Pairwise probabilistic registration methods constrain the GMM means to coincide with the points of one set, e.g. [4], [5]. Note that such a coincidence is inherently problematic, as long as both point sets are noisy and may include outliers. Even if one includes a uniform component in the mixture to deal with outliers [17], one of the sets is supposed to be "perfect". Instead, the means of the proposed formulation are not tight to a particular set: they result from fitting a mixture model to the data sets that are appropriately rotated and translated. In addition to registration, this also achieves scene reconstruction, since the cluster means may be viewed as the scene model. The proposed formulation implicitly enforces a closed-loop constraint. In other words, the proposed model assumes a star network topology, while the pairwise registration schemes assume a ring topology or a fully connected network

This article is an extended version of [18]. Several aspects of the proposed model are discussed into more detail, namely initialization, behavior, complexity and advantages over existing methods. In addition to the batch EM described in [18], we introduce an incremental version of EM, which solves the parameter estimation problem more efficiently at the cost of less accurate results. Experiments with novel datasets and benchmarks with several recent methods are included as well.

The remainder of this paper is organized as follows: Section II discusses the related work. Section III formulates the problem in a generative probabilistic framework. Section IV describes the batch EM together with an algorithm analysis while the incremental version is presented in Section V. Section VI describes in detail various initialization procedures. Section VII presents the experimental results and Section VIII concludes the paper.[1]

II. Related work

The two-set registration problem is usually solved by ICP [1], [19] or by one of its variants [2], [3], [20]–[22]. While ICP alternates between hard assignments and transformation estimation, more sophisticated registration approaches replace the binary assignments with probabilities [4], [5], [23]–[25]. Nevertheless, whether based on ICP or on soft assignments, these methods consider one set as the "model" and the other set as the "data", thus leading to solutions that are biased as long as both sets may contain noise and outliers. Alternatively, [6], [26] consider two Gaussian mixtures, one per point set, and the rigid transformation is applied to one of these mixtures. This leads to a non-linear optimization problem.

Multiple point set registration is often addressed by a sequential pairwise registration strategy [10], [11], [19], in particular when an online solution is required. Whenever an additional set is available, the model set is updated using either an ICP-like or a probabilistic scheme. Apart from the drawbacks associated with pairwise registration, this incremental mode of operation is subject to error propagation, while it fails to close any existing loop. As for offline applications, several approaches have been proposed, being mostly based on the underlying network, a.k.a. viewgraph, defined by the sets (represented as nodes) and their relative overlap (represented as edges). The majority of these methods initialize the poses via a pairwise registration.

The first solution for the problem in question was proposed in [13], where the sets are organized in a star-shaped network with one of the sets in the center, and such that any two sets are linked via two edges, hence by combining two rigid transformations. An algorithm computes the transformations incrementally based on a point-to-plane ICP algorithm [19]. [27] proposed to accelerate this algorithm by allowing incremental updates once pairwise registration within the loop has been performed. [15] starts with pairwise registrations to build the set graph, while a global registration step eliminates inconsistent matches and leads to the model graph whereby poses are provided. All these methods, however, consider in practice one set as reference, thus favoring a biased and non-symmetric solution.

Alternatively, [28] proposes a method to register multiple range images based on shape modeling: a point-to-surface distance is defined, the signed distance field, and the algorithm alternates between alignment and registration. Measurement errors and wrong correspondences are handled by a robust loss function. This method is well suited for dense range data since a surface representation is necessary.

Other methods consider known and fixed correspondences across multiple sets, thus updating only the transformations to balance the global error over the viewgraph [7], [14], [16], [29]–[31]. The main principle of these methods is that transformations along a network cycle ideally compose to the identity transformation. The cycles may refer to either minor loops between two adjacent sets or a larger cycle over the network.[2] Provided an approximate alignment, the goal is to minimize the on-cycle accumulated error from registering pairs of relevant (nearby) views. However, when data are ignored, a low inconsistency between coordinate frames does not necessarily mean better surface registration, in particular when good initialization is not available. As a consequence, these methods just "spread" any existing bias across the network without any correspondence refinement.

An alternative approach consists of considering a dense sequence of depth images and of estimating slight misalignments between these images. If the images are linearly correlated, the image alignment can be obtained via low-rank decomposition of a large matrix which has as columns the misaligned images. This formulation has been successfully applied to 2D image alignment [32] and extended to align images gathered with a RGB-D sensor [33]. The method is however limited to small camera motions such as to preserve the necessary condition that the images are linearly correlated. Our method addresses a different scenario, because it can handle large camera

[1]Matlab code, datasets and videos are available at https://team.inria.fr/perception/research/jrmpc/.

[2]When a spanning tree is used, an unused edge is added to obtain a cycle.

displacements and it does not necessitate dense RGB-D data. We conclude that our method and [33] are complementary.

Several recent methods built on the motion averaging principle introduced in [34] and based on rotation averaging [35]. Provided the view network, [9] suggests a motion averaged ICP algorithm. This algorithm alternates between the correspondence step and a double motion update. Any edge of the network implies an ICP run that updates a relative motion, and the redundancy information from *all* the relative motions in turn lead to a new global motion (one transformation per set) through the Lie-algebraic motion averaging principle. Then, the global motion information is back propagated in order to re-update the relative motions in a globally consistent manner. Again, the main assumption behind averaging is that traversing a cycle on the view network implies no motion. However, point correspondences are also updated here. [36] adopts the same technique but it employs trimmed-ICP [37] to compute pairwise motions. Note that an existing closed-loop may need to be pre-defined or pre-detected.

Probabilistic methods have been also proposed. As in [6], [8] represents each point set as a GMM and the non-rigid transformations are applied to cluster centers rather than to raw points. The model parameters are estimated by minimizing the Jensen-Shannon divergence of multiple densities and a probabilistic mean shape is built (as a by-product) from the convex combination of the aligned sets. This method vitally depends on each set's clusters, thus requiring highly and well structured point sets with no outliers. More closely to our method, [38] proposed an EM algorithm that alternates between the reconstruction of the object's mean shape and the registration between the sets and this shape. Despite the same principle, i.e. an emerging mean shape generates the sample sets, [38] considers *given correspondences* as well as several simplifications. KinectFusion [12] would roughly fall into this category owing to its model-to-frame registration strategy. Unlike these approaches, [39] and [40] build on pairwise registrations. The former generalizes [4] to align multiple super-resolved depth images by jointly optimizing many pairwise alignments along with compensating for pixel-dependent systematic bias. The latter extends the objective function of [7] and [30] by considering correspondences as missing data that are inferred along with the pairwise transformations in an EM fashion. Recently, [41] proposed an extension of [18] that integrates RGB information which enables better initial matches. In a large-scale outdoor context, [42] exploits positioning and map data to a pre-detect closed loop, while it proposes a multiple point set extension of [21] to simultaneously refine the intra-loop poses of the range sensor.

III. Problem formulation

Let $\mathbf{V}_j = [\boldsymbol{v}_{j1} \dots \boldsymbol{v}_{ji} \dots \boldsymbol{v}_{jN_j}] \in \mathbb{R}^{3\times N_j}$ be N_j data points that belong to point set j and let M be the number of point sets. We denote with $\mathbf{V} = \{\mathbf{V}_j\}_{j=1}^{M}$ the union of all these sets. A rigid transformation $\phi_j : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, i.e. a rotation matrix and a translation vector, maps $\boldsymbol{v}_{ji}$ from a *set-centered* frame to a *model-centered* frame, such that all the points form all the sets are expressed in the same coordinate frame. The objective is to estimate the M data-set-to-model-set transformations under the assumption that the observed points are generated from the same mixture model

$$P(\boldsymbol{v}_{ji}) = \sum_{k=1}^{K} p_k \mathcal{N}\big(\mathbf{R}_j\boldsymbol{v}_{ji} + \boldsymbol{t}_j; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big) + p_{K+1}\mathcal{U}(h), \quad (1)$$

where $\mathbf{R}_j \in \mathbb{R}^{3\times3}$ is a rotation matrix and $\boldsymbol{t}_j \in \mathbb{R}^3$ is a translation vector such that $\phi_j(\boldsymbol{v}_{ji}) = \mathbf{R}_j\boldsymbol{v}_{ji} + \boldsymbol{t}_j$, p_k are the mixing coefficients with $\sum_{k=1}^{K+1} p_k = 1$, $\boldsymbol{\mu}_k \in \mathbb{R}^3$ and $\boldsymbol{\Sigma}_k \in \mathbb{R}^{3\times3}$ are the mean vectors and covariance matrices respectively, and $\mathcal{U}(h)$ is the uniform distribution parameterized by the volume h of the 3D convex hull encompassing the data [5]. We now define γ as the ratio between outliers and inliers

$$\gamma = \frac{p_{K+1}}{\sum_{k=1}^{K} p_k}. \quad (2)$$

This allows to balance the outlier/inlier proportion via a judicious choice of γ. To summarize, the model parameters are

$$\Theta = \big\{\{p_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}, \{\mathbf{R}_j, \boldsymbol{t}_j\}_{j=1}^{M}\big\}. \quad (3)$$

This problem can be solved using an EM algorithm. We define hidden variables $\mathcal{Z} = \{Z_{ji} | j \in [1 \dots M], i \in [1 \dots N_j]\}$ such that $Z_{ji} = k$ means that observation $\boldsymbol{v}_{ji}$ is assigned to the k-th component of the mixture, and we seek to estimate the parameters Θ by maximizing the expected complete-data log-likelihood given the observed data

$$\begin{aligned}\mathcal{E}(\Theta|\mathbf{V}, \mathcal{Z}) &= \mathbb{E}_{\mathcal{Z}}[\log P(\mathbf{V}, \mathcal{Z}|\mathbf{V}; \Theta)] \\ &= \sum_{\mathcal{Z}} P(\mathcal{Z}|\mathbf{V}; \Theta) \log(P(\mathbf{V}, \mathcal{Z}; \Theta)). \end{aligned} \quad (4)$$

IV. Batch Registration

Assuming that the observed data are independent and identically distributed, it is straightforward to write (4) as

$$\mathcal{E}(\Theta|\mathbf{V}, \mathcal{Z}) = \sum_{j,i,k} \alpha_{jik}\Big(\log p_k + \log P(\boldsymbol{v}_{ji}|Z_{ji} = k; \Theta)\Big), \quad (5)$$

where $\alpha_{jik} = P(Z_{ji} = k|\boldsymbol{v}_{ji}; \Theta)$ are the posteriors. By replacing the standard expressions of the likelihoods [43] and by ignoring constant terms, (5) can be written as an objective function of the form

$$\begin{aligned} f(\Theta) = -\frac{1}{2}\sum_{j,i,k} \alpha_{jik}\big(&\|\phi_j(\boldsymbol{v}_{ji}) - \boldsymbol{\mu}_k)\|^2_{\boldsymbol{\Sigma}_k} + \log|\boldsymbol{\Sigma}_k| \\ &- 2\log p_k\big) + \log p_{K+1}\sum_{j,i}\alpha_{ji(K+1)}, \end{aligned} \quad (6)$$

where $|\cdot|$ denotes the determinant and $\|\boldsymbol{y}\|^2_{\mathbf{A}} = \boldsymbol{y}^\top\mathbf{A}^{-1}\boldsymbol{y}$. The model is farther restricted to isotropic covariances, i.e. $\boldsymbol{\Sigma}_k = \sigma_k\mathbf{I}_3$, since this leads to closed-form maximization solutions for all the model parameters (3), while non-isotropic covariances lead to a more complex convex optimization problem with no significant gain in accuracy [5]. Particular

care must be given to the estimation of the rotation matrices, namely a constrained optimization problem

$$\begin{cases} \max_{\Theta} f(\Theta) \\ \text{s.t.} \quad \mathbf{R}_j^\top \mathbf{R}_j = \mathbf{I}_3 \text{ and } |\mathbf{R}_j| = 1, \forall j \in [1 \dots M], \end{cases} \tag{7}$$

which can be solved via EM. Notice that the standard M steps for Gaussian mixtures are augmented with a step that estimates the rigid transformation parameters. We will refer to this algorithm as *joint registration of multiple point clouds* (JRMPC). The batch version will be referred to as JRMPC-B and it is outlined in Algorithm 1. This leads to a conditional maximization procedure [44]. Each M-step first estimates the transformation parameters, given the current responsibilities and Gaussian mixture parameters, and then estimates the new mixture parameters, given the new transformation parameters. It is of course possible to adopt a reverse order, in particular when rough rigid transformations are available. However, the proposed order does not assume such prior information.

Algorithm 1 Batch Joint Registration of Multiple Point Clouds (JRMPC-B)

Require: Initial parameter set Θ^0, number of components K, number of iterations Q.
$q \leftarrow 1$
repeat
E-step:
Use Θ^{q-1} to estimate posterior probabilities $\alpha_{jik}^q = P(Z_{ji} = k | \boldsymbol{v}_{ji}; \Theta^{q-1})$, i.e. (8).
M-rigid-step:
Use α_{jik}^q, $\boldsymbol{\mu}_k^{q-1}$ and $\boldsymbol{\Sigma}_k^{q-1}$ to estimate $\mathbf{R}_j^q$ and $\boldsymbol{t}_j^q$, i.e. (12) and (14).
M-GMM-step:
Use α_{jik}^q, $\mathbf{R}_j^q$ and $\boldsymbol{t}_j^q$ to estimate the means $\boldsymbol{\mu}_k^q$, i.e. (15).
Use α_{jik}^q, $\mathbf{R}_j^q$, $\boldsymbol{t}_j^q$ and $\boldsymbol{\mu}_k^q$ to estimate the covariances $\boldsymbol{\Sigma}_k^q$, i.e. (16).
Use α_{jik}^q to estimate the priors p_k^q, i.e. (18).
$q \leftarrow q + 1$
until $q > Q$ (or Θ's update is negligible)
return Θ^q

A. E-step

The posterior probability of point $\boldsymbol{v}_{ji}$ to be associated with cluster k, e.g. an inlier, is

$$\alpha_{jik} = \frac{\beta_{jik}}{\sum_{s=1}^{K} \left(\beta_{jis}\right) + \frac{\gamma}{h(\gamma+1)}}, \tag{8}$$

where $\gamma / h(\gamma + 1)$ accounts for the uniform component in the mixture, and with the notation:

$$\beta_{jik} = \frac{p_k}{\sigma_k^{3/2}} \exp\left(\frac{-\|\mathbf{R}_j \boldsymbol{v}_{ji} + \boldsymbol{t}_j - \boldsymbol{\mu}_k\|^2}{2\sigma_k}\right). \tag{9}$$

Therefore, the posterior probability of being an *outlier* is simply given by $\alpha_{ji\,K+1} = 1 - \sum_{k=1}^{K} \alpha_{jik}$. As shown in Algorithm 1, the posterior probability at the q-th iteration, α_{jik}^q, is computed from (8) using the parameter set Θ^{q-1}.

B. M-rigid-step

This step estimates the rotations $\mathbf{R}_j$ and translations $\boldsymbol{t}_j$ that maximize $f(\Theta)$, given current values for α_{jik}, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, and p_k. Notice that this estimation can be carried out independently for each set $\mathbf{V}_j$. By setting the GMM parameters to their current values, we reformulate the problem of estimating the rotations and translations. The rigid transformation parameters that maximize $f(\Theta)$ can be estimated from the following constrained minimization

$$\begin{cases} \min\limits_{\mathbf{R}_j, \boldsymbol{t}_j} \|(\mathbf{R}_j \mathbf{W}_j + \boldsymbol{t}_j \boldsymbol{e}^\top - \mathbf{M}) \boldsymbol{\Lambda}_j\|_F^2 \\ \text{s.t.} \quad \mathbf{R}_j^\top \mathbf{R}_j = \mathbf{I}_3 \text{ and } |\mathbf{R}_j| = 1, \end{cases} \tag{10}$$

where $\boldsymbol{\Lambda}_j \in \mathbb{R}^{K \times K}$ is a diagonal matrix with entries $\lambda_{jkk} = (\sum_{i=1}^{N_j} \alpha_{jik} / \sigma_k)^{1/2}$, $\mathbf{M} = [\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K] \in \mathbb{R}^{3 \times K}$, $\boldsymbol{e} \in \mathbb{R}^K$ is a vector of ones, $\|\cdot\|_F$ denotes the Frobenius norm, and $\mathbf{W}_j = [\boldsymbol{w}_{j1}, \dots, \boldsymbol{w}_{jK}] \in \mathbb{R}^{3 \times K}$, where $\boldsymbol{w}_{jk}$ is the weighted average of the j-th point set assigned to the k-th mixture component

$$\boldsymbol{w}_{jk} = \frac{\sum_{i=1}^{N_j} \alpha_{jik} \boldsymbol{v}_{ji}}{\sum_{i=1}^{N_j} \alpha_{jik}}, \tag{11}$$

The minimization (10) can be solved in closed-form and is a weighted version of the solution [45]. The optimal rotation matrices are

$$\mathbf{R}_j = \mathbf{U}_j^l \mathbf{S}_j \mathbf{U}_j^{r\top}, \ \forall j \in [1 \dots M], \tag{12}$$

where $\mathbf{U}_j^l$ and $\mathbf{U}_j^r$ are the left and right matrices respectively, obtained from the singular value decomposition of matrix $\mathbf{M} \boldsymbol{\Lambda}_j \mathbf{P}_j \boldsymbol{\Lambda}_j \mathbf{W}_j^\top$, with

$$\mathbf{P}_j = \mathbf{I}_3 - \frac{\boldsymbol{\Lambda}_j \boldsymbol{e} \boldsymbol{e}^\top \boldsymbol{\Lambda}_j}{\boldsymbol{e}^\top \boldsymbol{\Lambda}_j^2 \boldsymbol{e}} \tag{13}$$

is a projection matrix and $\mathbf{S}_j = \text{diag}(1, 1, |\mathbf{U}_j^l||\mathbf{U}_j^r|)$. Once the optimal rotation matrices are estimated, the optimal translation vectors are easily computed with

$$\boldsymbol{t}_j = \frac{1}{\text{trace}(\boldsymbol{\Lambda}_j^2)} (\mathbf{M} - \mathbf{R}_j \mathbf{W}_j) \boldsymbol{\Lambda}_j^2 \boldsymbol{e}, \ \forall j \in [1 \dots M]. \tag{14}$$

Note that each rigid transform ϕ_j aligns the GMM means with K virtual points $\{\boldsymbol{w}_{jk}\}_{k=1}^K$ (one virtual point per component). Therefore, the proposed method can deal with point sets of different cardinalities and the number of components in the mixture, K, can be chosen independently of these cardinalities. This is an important advantage over pairwise registration methods that assume that the cardinalities of the two point sets must be similar.

C. M-GMM-step

Given rigid transformation estimates and posterior probabilities, one can use standard optimization techniques to compute

the optimal means and covariances:

$$\boldsymbol{\mu}_k = \frac{\sum_{j=1}^{M}\sum_{i=1}^{N_j}\alpha_{jik}(\mathbf{R}_j\boldsymbol{v}_{ji}+\boldsymbol{t}_j)}{\sum_{j=1}^{M}\sum_{i=1}^{N_j}\alpha_{jik}}, \tag{15}$$

$$\sigma_k = \frac{\sum_{j=1}^{M}\sum_{i=1}^{N_j}\alpha_{jik}\|\mathbf{R}_j\boldsymbol{v}_{ji}+\boldsymbol{t}_j-\boldsymbol{\mu}_k\|_2^2}{3\sum_{j=1}^{M}\sum_{i=1}^{N_j}\alpha_{jik}} + \epsilon^2, \tag{16}$$

where ϵ is a small scalar to avoid singularities. As for the priors, we introduce a Lagrange multiplier to take into account the constraint $\sum_{k=1}^{K+1} p_k = 1$. This leads to the following dual function

$$g(p_1,\dots,p_K,\eta) = \sum_{k=1}^{K}\left(\log p_k \sum_{j=1}^{M}\sum_{i=1}^{N_j}\alpha_{jik}\right) + \eta\left(\sum_{k=1}^{K} p_k - \frac{1}{1+\gamma}\right). \tag{17}$$

and its optimization yields

$$p_k = \frac{1}{\eta}\sum_{j=1}^{M}\sum_{i=1}^{N_j}\alpha_{jik}, \quad \forall k \in \{1,\dots,K\} \tag{18}$$

$$p_{K+1} = 1 - \sum_{k=1}^{K} p_k,, \tag{19}$$

with $\eta = (\gamma+1)(N - \sum_{j=1}^{M}\sum_{i=1}^{N_j}\alpha_{ji\,K+1})$ and $N = \sum_{j=1}^{M} N_j$. Note that if $\gamma \to 0$, which means that there is no uniform component in the mixture, then $\eta \to N$, which is in agreement with [43].

D. Algorithm Analysis

The leading complexity of JRMPC-B is $O(NK)$ owing to E-step and equation (9). If $\bar{N}$ is the average cardinality of a point set, the complexity can be written as $O(\bar{N}MK)$. Typically, $K < \bar{N}$ owing to underlying clustering while K could be close to or even greater than $\bar{N}$ when many non-ovelapping sets cover a large volume.

The proposed algorithm has a number of advantages over pairwise registration methods. Such methods, e.g. [7], [9], [36], are intrinsically more time-consuming than joint registration because one has to consider all point-set-to-point-set combinations. Either using EM or ICP when registering each pair of sets, the evaluation of all point-to-point distances is needed. Such a strategy requires $O(\bar{N}^2M^2)$ operations in principle. This complexity can be decreased by structuring the data, e.g. using KD-trees, at the cost of data structure building. Approximate solutions, e.g. sequential or one-versus-all approaches, consider $M-1$ pairs of sets, thus requiring $O(\bar{N}^2M)$ operations at the expense of performance.

Another important difference between joint and pairwise registrations is that the former puts all the point sets on an equal footing and registration is truly cast into clustering, i.e. (1), whereas the latter performs an unbalanced treatment of the point sets, i.e. one set constitutes the data and the other set constitutes the model. More precisely, when EM is used for registering pairs [4], [5], the generative model $\mathcal{N}\big(\mathbf{R}_j\boldsymbol{v}_{ji}+\boldsymbol{t}_j;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\big)$ in (1) is replaced with $\mathcal{N}\big(\boldsymbol{v}_{aj};\mathbf{R}_{ab}\boldsymbol{v}_{bi}+\boldsymbol{t}_{ab},\boldsymbol{\Sigma}_{bi}\big)$, where $\boldsymbol{v}_{ai}$ belongs to point set a (the data), $\boldsymbol{v}_{bj}$ belongs to point set b (the model) and $(\mathbf{R}_{ab},\boldsymbol{t}_{ab})$ is the rigid alignment that maps b onto a. Hence, in the joint case, the mixture is modeled by a set of free parameters, while in the pairwise case, the means directly depend on the rigid parameters. The immediate consequence is that noisy points or outliers that may be present in the data will propagate via this dependency and will give rise to bad means in the mixture. When ICP is used, again one of the sets is identified with the model.

The minimal configuration required by the proposed method consists of two sets with at least three overlapping points. The algorithm can be applied to a large number of point sets. However in this case, the computation time increases linearly with N on the premise that K does not depend on the number of point sets to be aligned. For this reason, it is interesting to provide an incremental version of the algorithm, on the following ground: once M point sets ($M \geq 2$) are aligned with the JRMPC-B algorithm described above, new sets can be added incrementally (one at a time) and aligned with the current model, at a lower computational cost than the batch algorithm, using update formulae for mixture parameters. The incremental version of the algorithm is described in detail in the next section.

V. Incremental Registration

The incremental version of the proposed registration method, referred to as JRMPC-I, is outlined in Algorithm 2. This algorithm considers the new m-th point set to be aligned with $m-1$ already registered sets. The latter and the corresponding $m-1$ transformations are not updated and therefore are not used as input and output arguments. The JRMPC-I algorithm starts with computing the responsibilities α_{mik}, it then estimates the rigid transformation that aligns the set with the already aligned sets, $\mathbf{R}_m$ and $\boldsymbol{t}_m$, and finally updates the mixture parameters. This process can be optionally repeated for a few iterations (Q). While the mixture parameters are initialized with those previously calculated, the integration of the new set requires initialization of $\mathbf{R}_m$ and $\boldsymbol{t}_m$, referred to as $\mathbf{R}_m^0$ and $\boldsymbol{t}_m^0$ in Algorithm 2. One possible strategy that has been successfully used with a moving RGB-D camera, e.g. [12], is to initialize the rigid transformation with $\mathbf{R}_m^0 = \mathbf{R}_{m-1}$ and $\boldsymbol{t}_m^0 = \boldsymbol{t}_{m-1}$. In the more general case, one can use the initialization strategy discussed in Section VI.

We denote with $\{p_k^{1:m-1}, \boldsymbol{\mu}_k^{1:m-1}, \sigma_k^{1:m-1}\}_{k=1}^K$ the GMM parameters estimated with JRMPC-B, where the notation $1:m$ denotes the sets from 1 to m. The incremental registration algorithm proceeds iteratively. The E-step computes the re-

Algorithm 2 Incremental Joint Registration of Multiple Point Clouds (JRMPC-I)

Require: GMM parameters estimated with JRMPC-B, $\{p_k^{1:m-1}, \boldsymbol{\mu}_k^{1:m-1}, \sigma_k^{1:m-1}\}_{k=1}^K$, m-th point set, rigid transformation $\mathbf{R}_m^0, \boldsymbol{t}_m^0$, number of iterations Q.

$q \leftarrow 1$
repeat
E-step:
  Use (20) to estimate α_{mik}^q, $1 \leq i \leq N_m, 1 \leq k \leq K$.
M-rigid-step:
  Use (12) and (14) to estimate $\mathbf{R}_m^q, \boldsymbol{t}_m^q$.
M-GMM-step:
  Use (21) to update $\boldsymbol{\mu}_k^{1:m\,q}, 1 \leq k \leq K$.
  Use (22) to update $\sigma_k^{1:m\,q}, 1 \leq k \leq K$.
  Use (23) to update $p_k^{1:m\,q}, 1 \leq k \leq K$.
  $q \leftarrow q + 1$
until $q > Q$.
return $\mathbf{R}_m^q, \boldsymbol{t}_m^q, \{p_k^{1:m\,q}, \boldsymbol{\mu}_k^{1:m\,q}, \sigma_k^{1:m\,q}\}_{k=1}^K$

sponsibilities associated with the m-th set:

$$\alpha_{mik} = \frac{\beta_{mik}}{\sum_{s=1}^{K} \left(\beta_{mis}\right) + \frac{\gamma}{h(\gamma+1)}}, \tag{20}$$

$$\beta_{mik} = \frac{p_k^{1:m-1}}{(\sigma_k^{1:m-1})^{\frac{3}{2}}} \exp\left(\frac{-\|\mathbf{R}_m \boldsymbol{v}_{mi} - \boldsymbol{t}_m - \boldsymbol{\mu}_k^{1:m-1}\|^2}{2\sigma_k^{1:m-1}}\right).$$

The M-rigid-step uses equations (11)-(14) to calculate $\mathbf{R}_m$ and $\boldsymbol{t}_m$ in closed-form. This rigid transformation aligns the m-th set with the GMM means that explain the previously aligned sets, hence the joint alignment of all the sets. The M-GMM-step updates the means, covariances and priors:

$$\boldsymbol{\mu}_k^{1:m} = \frac{\zeta_k \boldsymbol{\mu}_k^{1:m-1} + \boldsymbol{u}_{mk}}{\zeta_k + 1}, \tag{21}$$

$$\sigma_k^{1:m} = \frac{\zeta_k \sigma_k^{1:m-1} + \|\Delta\boldsymbol{\mu}_k\|^2 - \Delta\boldsymbol{\mu}_k^\top (\boldsymbol{u}_{mk} - \frac{1}{\alpha_{mk}} \boldsymbol{\mu}_k^{1:m-1})}{\zeta_k + 1}, \tag{22}$$

$$p_k^{1:m} = \frac{\alpha_{mk}\zeta_k + 1}{\eta^{1:m}}, \tag{23}$$

with

$$\boldsymbol{u}_{mk} = \mathbf{R}_m \boldsymbol{w}_{mk} + \boldsymbol{t}_m, \quad \Delta\boldsymbol{\mu}_k = \boldsymbol{\mu}_k^{1:m} - \boldsymbol{\mu}_k^{1:m-1},$$
$$\zeta_k = \frac{\eta^{1:m-1} p_k^{1:m-1}}{\alpha_{mk}}, \quad \alpha_{mk} = \sum_{i=1}^{N_m} \alpha_{mik},$$
$$\eta^{1:m} = \eta^{1:m-1} + (\gamma+1)(N_m + 1 - \alpha_{mk}).$$

The number of iterations that JRMPC-I needs to converge depends on its initialization. It was noticed that a small number of iterations are sufficient when data are gathered from a smoothly moving camera. Once a few sets have been integrated with JRMPC-I, it may be useful to run JRMPC-B in order to obtain a globally optimal alignment and to reject the outliers. Also, it is worthwhile to remark that JRMPC-I is not meant to grow the model, i.e. it is not designed to increase the number of components of the Gaussian mixture as point sets, possibly with no overlap, are incrementally added. JRMPC-I should be merely used when an efficient algorithm is needed. In the particular case of a large number of sets, e.g. depth sequences, a temporally hierarchical scheme that benefits from both versions is recommended to cope with the large memory requirements.

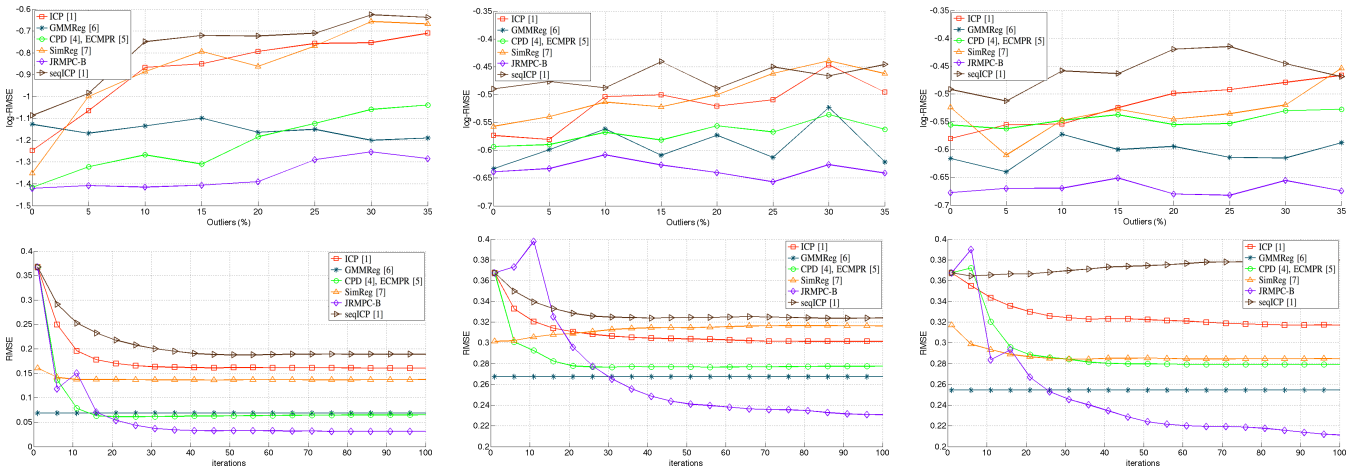
VI. Initialization

It is well known that initialization plays a crucial role in EM procedures. Therefore, we discuss here initialization options well suited for point set registration. We assume no prior information about the position and orientation of the camera(s) with respect to the scene. However, information such as the calibration parameters of a network of static cameras, or transformations between pairs of point sets, could be used if available. We also assume that there is sufficient overlap between pairs of point sets. The sensitivity of our method to the amount of overlap is tested an analyzed in Sec. VII.

When the point sets have a sufficient joint overlap, the translation vectors can be initialized by centroid differences, i.e. $\boldsymbol{t}_j^0 = \bar{\boldsymbol{\mu}} - \bar{\boldsymbol{v}}_j$, where $\bar{\boldsymbol{\mu}}$ is the centroid of the cluster means and $\bar{\boldsymbol{v}}_j$ the centroid of the $j-$th set. If the point sets suffer from strong artifacts, e.g. flying pixels, the difference of medians can be used instead. Rotation matrices can be simply initialized with $\mathbf{R}_j^0 = \mathbf{I}_3$. Instead, when many non-overlapping pairs exist, a pairwise registration is preferred, i.e. the minimum number of pairs that leads to a rough global alignment can be registered beforehand.

Several strategies may be adopted for initializing the mixture parameters. One way to do it is to initialize the means with the points of one set. Another way is to distribute the means on the surface of a sphere that encompasses the convex hull of the point sets already centered at $\bar{\boldsymbol{\mu}}$. Concerning the variance, we found that starting with a high value yields very good results and that the variances quickly converge to the final values. Our algorithm converges much faster than EM algorithms that adopt a deterministic annealing behavior, i.e. the variance is decreased according to an annealing schedule. Finally the priors are initialized with $1/(K+1)$, where we remind that K is the number of Gaussian components. While update formulae for the priors are provided with both our algorithms, in practice it was found that keeping the priors constant affect neither the convergence nor the quality of the registrations. Notice that any rough pre-alignment of the sets results in a very good initilazation of the mixture parameters, that is, the means can be intitialized from re-sampling the registered set while the variances can be intiialized such that each cluster encompasses a sufficient number of points.

In order to choose the number of components, we propose the following empirical strategy. If the cardinalities of the point sets are similar, one can use $K = \bar{N}$ (recall that $\bar{N}$ is the average number of points in a set). However, K may be chosen to be smaller than $\bar{N}$ if the sets highly overlap, or larger if there are many non-overlapping sets. One should notice that



(a) (b) (c)

Fig. 2: *Top:* log-RMSE as a function of outlier percentage when SNR=$10dB$. *Bottom:* The learning curve of algorithms for a range of 100 iterations when the models are disturbed by SNR=$10dB$ and 20% outliers. (a) "Lucy", (b) "Bunny" (c) "Armadillo".

the number of components in the mixture merely depends on the data and on the application at hand. Experimentally we found that $K \ll \bar{N}$ yields excellent alignment results in the presence of dense depth data, as provided by depth sensors.

VII. Experiments

In this section, we test and benchmark the proposed algorithms with widely used and publicly available 3D data, as well as with time-of-flight (TOF) and structured-light data. First, we compare the proposed algorithm with pairwise registration methods, which illustrates the behavior of the algorithm in comparison with other algorithms and in particular its robustness to noise and to outliers. Second, we compare our algorithm with recently proposed joint-registration algorithms. Third, we test and evaluate the best performing algorithms with challenging TOF data and with structured-light data captured with a moving Kinect camera.

A. Simulated Data

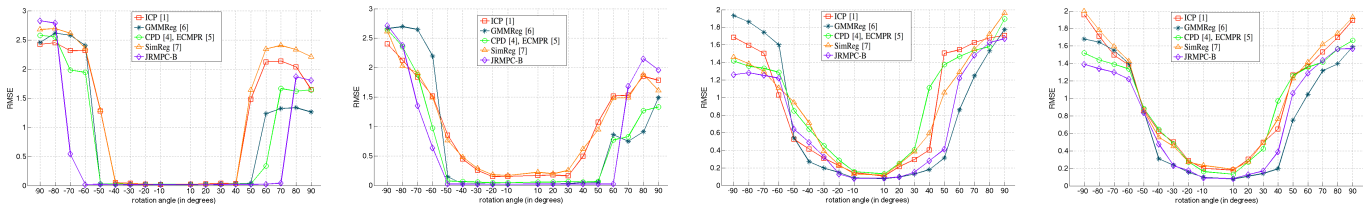
1) Comparison with pairwise registration algorithms: We use 3D models from Stanford's 3D scanning repository, namely "Bunny", "Lucy" and "Armadillo", and we proceed as follows in order to synthesize multiple point sets from different viewpoints. The model is shifted around the origin, the points are downsampled and then rotated in the xz-plane; points with negative z coordinates are rejected. This way, only a part of the object is viewed in each set, the point sets do not fully overlap, and the extent of the overlap depends on the rotation angle, as in real scenarios. It is important to note that downsampling differs over the sets, such that different points are present in each set as well as different cardinalities (from the range $[1000, 2000]$) are obtained. We add Gaussian noise to point coordinates based on a predefined signal-to-noise ratio (SNR), and more importantly, we add outliers to each set which are uniformly distributed around five randomly chosen points of the set. A tractable case of registering four point sets ($M = 4$) is considered here, the angle between the first set and the other sets being 10^o, 20^o and 30^o respectively. We include JRMPC-I in the latter experiments where more point-sets are registered.

For comparison, we consider the following baselines that follow the one-vs-all approach: ICP [1], CPD [4], ECMPR [5], GMMReg [6]. In addition, we include a sequential version of ICP (seqICP) and a modification of [7], abbreviated here as SimReg. Unlike the original version, the latter allows updating the matches at each iteration. Recall that CPD is exactly equivalent to ECMPR when it comes to rigid registration.[3] As showed in [6], Levenberg-Marquardt ICP [2] performs similarly with GMMReg, while [8] shows that GMMReg is superior to Kernel Correlation [3]. As a consequence, we implicitly assume a variety of baselines. All the competitors employ $M - 1$ registrations between the first and rest sets, while SimReg considers all the pairs of (overlapping) sets.

To evaluate the performance. we use the root of the mean squared error (RMSE) of the rotation parameters averaged by the number of sets. For all algorithms, we implicitly initialise the translations by transferring the centroids of the point clouds into the same point, while identity matrices initialize the rotations. GMMReg and SimReg are kind of favored in the comparison, since the former benefits from a two-level optimization (the first level initializes the second one) while the latter starts from the point where the pairwise ICP ends. Notice that the proposed method provides a transformation for *every* point set, while ground rotations are typically expressed in terms of the first set. Hence, the product of estimations $\hat{\mathbf{R}}_1^\top \hat{\mathbf{R}}_j$ is compared with rotation $\mathbf{R}_j$, i.e. the error for the j-th set is $\|\hat{\mathbf{R}}_1^\top \hat{\mathbf{R}}_j - \mathbf{R}_j\|_F$.

JRMPC starts here from a completely unknown GMM

[3]CPD uses a common variance for all components, while ECMPR uses a different variance for each comonent. The latter is used here.



(a) noise (b) noise+outliers (c) noise (d) noise+outliers

Fig. 3: RMSE as a function of the overlap (rotation angle) when two point sets are registered (SNR=$20dB$, 30% outliers) (a),(b) "Lucy" (c), (d) "Armadillo".

TABLE I: Registration error of indirect mappings. For each model, the two first columns show the rotation error of $V_2 \rightarrow V_3$ and $V_3 \rightarrow V_4$ respectively, while the third column shows the standard deviation of the two errors ($SNR = 10db$, 30% outliers).

	Bunny			Lucy			Armadillo		
ICP [1]	0.329	0.423	0.047	0.315	0.297	0.009	0.263	0.373	0.055
GMMReg [6]	0.364	0.303	0.030	0.129	0.110	0.009	0.228	0.167	0.031
CPD [4], ECMPR [5]	0.214	0.242	0.014	0.144	0.109	0.017	0.222	0.204	0.009
SimReg [7]	0.333	0.415	0.041	0.354	0.245	0.055	0.269	0.301	0.016
JRMPC-B	0.181	0.165	**0.008**	0.068	0.060	**0.004**	0.147	0.147	**0.000**

where the initial means $\boldsymbol{\mu}_k$ are distributed on a sphere that spans the convex hull of the sets. The variances σ_k are here initialized with the median distance between $\boldsymbol{\mu}_k$ and all the points in $\mathbf{V}$. We found that updating the priors does not drastically improve the registration. We therefore keep them constant and equal to $1/(K+1)$ during EM, while h is chosen to be the volume of a sphere whose radius is 0.5; the latter is not an arbitrary choice because the point coordinates are normalized by the maximum distance between points of the convex hull of $\mathbf{V}$. The number of the components, K, is here equal to 60% of the mean cardinality. We use 100 iterations for all algorithms while GMMReg performs 10 and 100 function evaluations for the first and second optimization levels respectively. However, the current authors' implementation allows to extract the parameters after the latest evaluation.

Fig. 2 shows the final log-RMSE, averaged over 100 realisations and all views, as a function of outlier percentage for each 3D model. Apparently, ICP and SimReg are more affected by the presence of outliers owing to one-to-one correspondences. CPD and GMMReg are affected in the sense that the former assigns outliers to any of the GMM components, while the latter may merge outliers into clusters. The proposed method is more robust to outliers and the registration is successful even with densely present outliers. The behavior of the proposed algorithm in terms of the outliers is discussed in detail below and showed on Fig. 4. To visualize the convergence rate of the algorithms, we show curves for a challenging setting ($SNR = 10dB$ and 20% outliers). Regarding GMMReg, we just plot a line that shows the error in steady state, since the author's implementation allow to extract the final parameters only. There is a performance variation as the model's surface changes. "Lucy" is more asymmetric than "Bunny" and "Armadillo", thus a lower floor is achieved. Unlike the competitors, JRMPC-B may show a minor perturbation in the first iterations owing to the joint solution and the initialization of the means and the variances.

It is also important to show the estimation error between sets whose geometric relation is not directly estimated. This also shows how biased each algorithm is. Based on the above experiment (SNR=$10db$, 20% outliers), Table I reports the average rotation error for the pairs (V_2, V_3) and (V_3, V_4), as well as the standard deviation of these two errors as a measure of bias. All but seqICP do not estimate these individual mappings alone. The proposed scheme, not only provides the lowest error, but it also offers the most symmetric solution.

A second experiment evaluates the robustness of the algorithms in terms of the rotation angle between two point sets, that is, the extent of their overlap. This also allows us to show how the proposed algorithm deals with the simple case of two point sets. Recall that JRMPC-B does not reduce to CPD/ECMPR in the two-set case, but it still computes the poses of the two sets with respect to the "central" GMM. Fig. 3 plots the average RMSE over 50 realizations of "Lucy" and "Armadillo", when the relative rotation angle varies from -90^o to 90^o. As for an acceptable registration error, the proposed scheme achieves the widest and shallowest basin for "Lucy", and competes GMMReg for "Armadillo". Since "Armadillo" consists of smooth and concave surfaces, the performance of the proposed scheme is better with multiple point sets than the two-set case, hence the difference with GMMReg. The wide basin of GMMReg is also due to its sophisticated initialization.

As mentioned, a by-product of the proposed method is the reconstruction of an outlier-free model. In addition, we are able to detect the majority of the outlying points based on the variance of the component they most likely belong to. To show this effect, we use the results of one realization of the first experiment with 30% outliers. Fig. 4 shows in (a) and (b) two out of four point sets, thereby one verifies the distortion of the point sets, as well as how different the sets may be, e.g. the right hand is missing in the first set. The progress of $\boldsymbol{\mu}_k$ estimation is shown in (d-f). Apparently, the algorithm starts by reconstructing the scene model (observe the presence of the right hand). Notice the size increment of the hull of the points $\boldsymbol{\mu}_k$, during the progress. This is

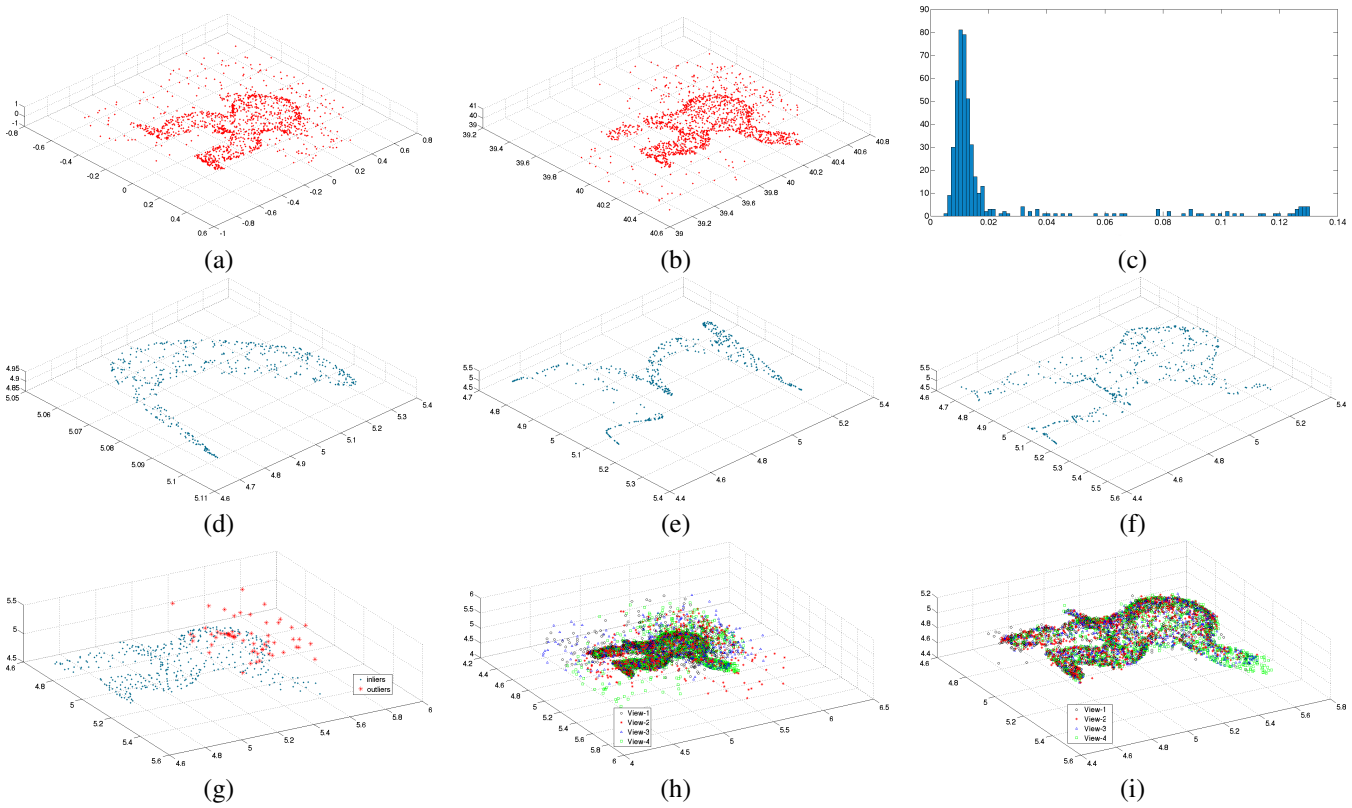


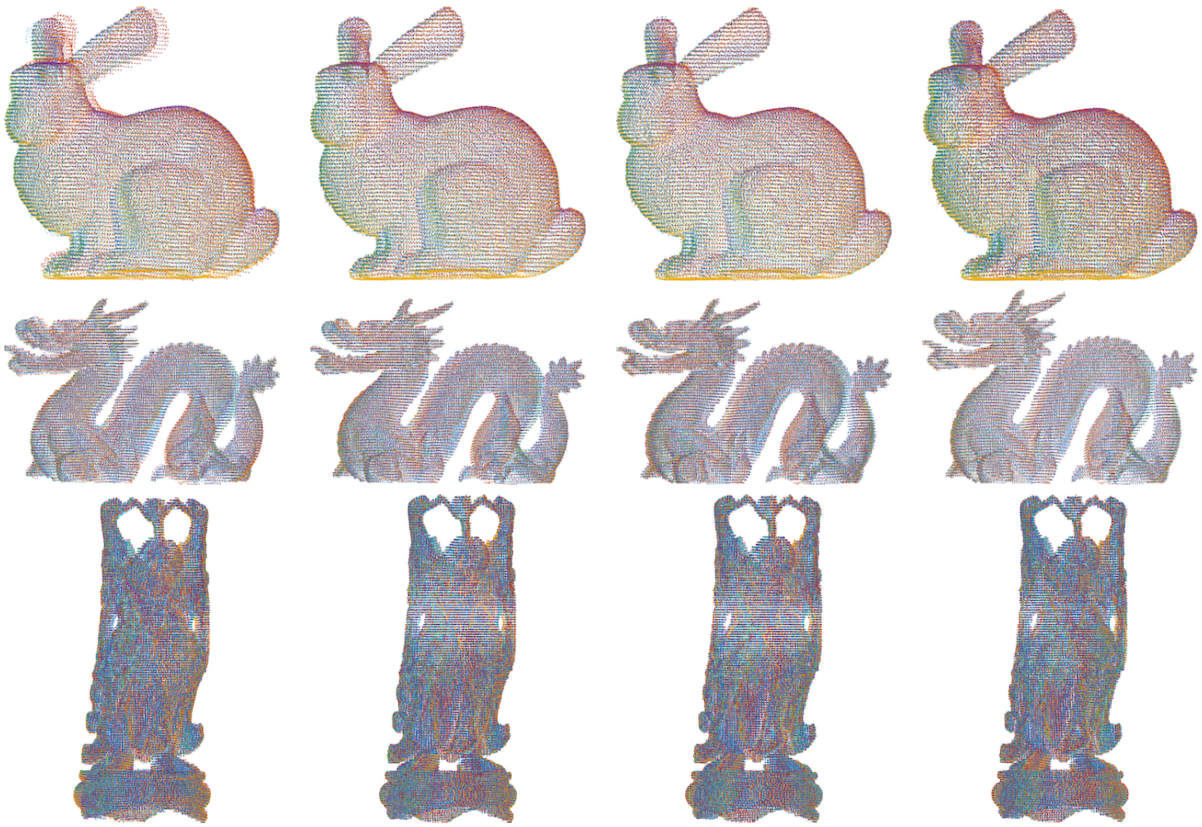
Fig. 4: (a),(b) Two point sets (out of four) with outliers; (c) distribution of estimated variances; instances of GMM means after (d) 5, (e) 15, and (f) 30 iterations; (g) the splitting of model points into inliers and outliers; joint-registration of four point sets (h) before and (i) after removing "bad" points (*best viewed on-screen*).

because the posteriors in the first iteration are very low and make the means $\boldsymbol{\mu}_k$ shrink into a very small cell. While the two point sets are around the points $(0,0,0)$ and $(40,40,40)$, we build the scene model around the point $(5,5,5)$. The distribution of the final deviations σ_k is shown in (c). We get the same distribution with any model and any outlier percentage, as well as when registering real data. Although one can fit a pdf, e.g. Rayleigh, here it is convenient to split the components using the threshold $T_\sigma = 2 \times median(\mathcal{S})$, where $\mathcal{S} = \{\sigma_k | k = 1, \dots K\}$. Accordingly, we build the scene model and we visualize the binary classification of points $\boldsymbol{\mu}_k$. Apparently, whenever components attract outliers, even not far from the object surface, they tend to spread their hull by increasing their scale. Based on the above thresholding, we can detect such components and reject points that are assigned with high probability to these components, as shown in (g). Despite the introduction of the uniform component that prevents the algorithm from building clusters away from the object surface, locally dense outliers are likely to create components outside the surface. In this example, most of the point sets contain outliers above the shoulders, and the algorithm builds components with outliers only, that are post-detected by their variance. The integrated surface is shown in (h) and (i) when "bad" points are automatically removed. Of course, the surface can be post-processed, e.g. smoothing, for a more accurate representation, but this is beyond of our goal.

2) Comparison with joint registration algorithms: We here compare our method with the joint registration algorithms of [9] and [36].[4] Recall that both rely on the motion averaging strategy using the ICP and the trimmed-ICP algorithm, respectively, hence abbreviated as MAICP and MATrICP. According to the literature, MAICP and MATrICP seem to outperform the methods of [7], [13], [27], [29]. The method of [7] is also included here as a baseline that considers fixed matches between the sets, and is referred to as multi-view ICP (MV-ICP). As mentioned above, MV-ICP considers all the pairs of overlapping views. While [8] generalizes GMMReg [6] for multiple point-sets, the authors provide the code for two-set case only.

For consistency reasons, the experimental setup of [9] is adopted, i.e. the point-sets have been roughly pre-aligned using a standard pair-wise ICP scheme. The error metric is the angle (in degrees) obtained from the composition of true and inverse estimation averaged over all point-sets, that is, it should ideally vanish. As with [9], we use the "Bunny", "Dragon" and "Happy Buddha" models from Stanford scanning repository owing to the availability of the ground truth motions. While "Bunny" is asymmetrically captured from 10 viewpoints, the last two sets contain 15 scans from evenly spaced view angles (every 24 degrees). To get the point-sets, true transformations first apply to the sets and then, we deform each set by a

[4]The code was kindly provided by the authors.

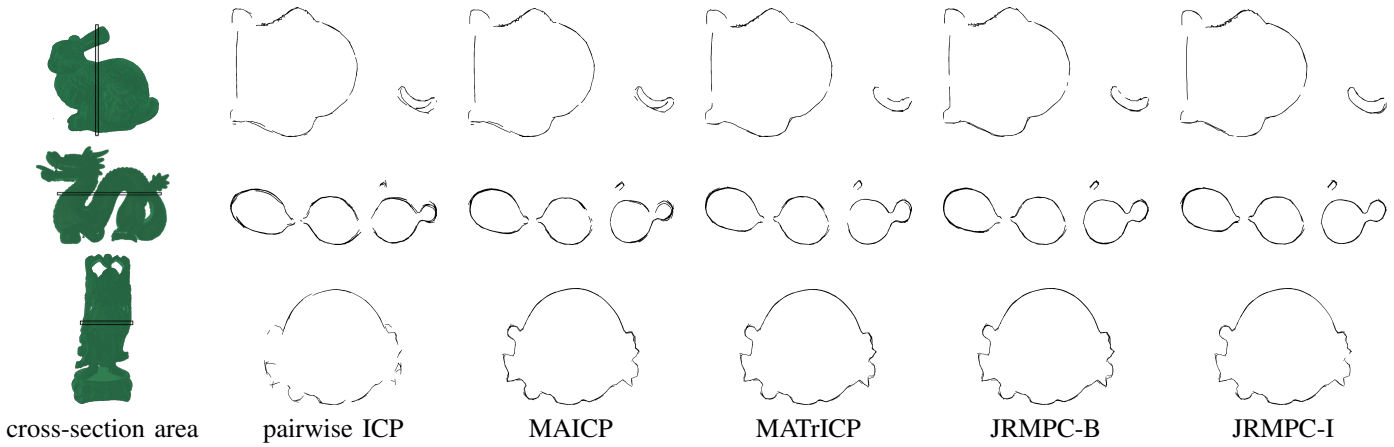


(a) MAICP (b) MATrICP (c) JRMPC-B (d) JRMPC-I

Fig. 5: Integrated models of Bunny (first row), Dragon (second row) and Happy Buddha (third row) based on four joint-wise registration methods (*best viewed on-screen*).

TABLE II: Comparison of multi-view registration methods without adding noise.

	Raw-data	Initialization	MV-ICP [7]	MAICP [9]	MATrICP [36]	JRMPC-B	JRMPC-I
Bunny	3.45	2.10	1.54	0.95	**0.27**	0.37	0.69
Dragon	7.28	4.37	3.75	1.95	0.62	**0.47**	0.73
Happy Buddha	10.77	3.18	2.45	0.64	0.43	**0.36**	0.77



cross-section area pairwise ICP MAICP MATrICP JRMPC-B JRMPC-I

Fig. 6: Cross-section of Bunny (top), Dragon (middle) and Happy Buddha (bottom) obtained from several algorithms (*best viewed on-screen*).

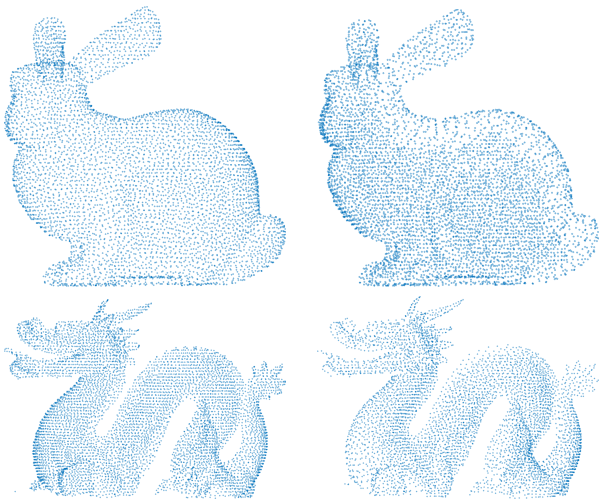


Fig. 7: The GMM means obtained from JRMPC-based algorithms for (top) "Bunny" and (bottom) "Dragon" Unlike JRMPC-B, JRMPC-I leads to non-uniformly distributed mixture components (biased towards the initial sets) since "old" means cannot be freely re-distributed (*best viewed on-screen*).

random yet known transformation. Finally, we down-sample the point-sets so that the cardinalities vary from 2000 to 5000 points. Unlike [9] and [36], we also evaluate the registration performance, when the point-sets have been further perturbed by noise. We deliberately avoid adding outliers since any mis-registration in the initialization step would make the motion averaging methods completely fail.

Both JRMPC-B and JRMPC-I consider the same number of components ($K \simeq 4000$) while the initial centers are randomly selected points from roughly aligned sets. For JRMPC-I, when a new set appears, K/M components are rejected and re-initialized with points from the new set. This is to enforce the displacement of some GMM means towards the new data, as long as model growing is not considered here. Several conditions may apply to this rejection stage. Here, we first reject degenerate clusters ($\sigma^2 = \epsilon^2$) (if any) and we randomly select old components to replace. One iteration of integration step and 30 refinement iterations with JRMPC-B are allowed owing to the different viewpoints, while we let the algorithm run 50 cycles to register the two first sets. Note that the current implementations of MV-ICP and MAICP consider the closed-loop known, that is, pairing the last with the first set, while MAICP also considers the scan boundaries known and rejects such points for potential matching. Instead, both versions of our algorithm as well as MATrICP make no use of any prior knowledge about the loop and the overlap.

Table II shows the registration error of the methods. As expected, MV-ICP fails to provide accurate registration owing to fixed matches. The proposed algorithm along with MATrICP achieve the most accurate registration, while JRMPC-I provides results of sufficient quality. Indeed, as claimed in [36], it seems that motion averaging benefits from more robust versions of ICP. The corresponding integrated models of the best performing algorithms are shown in Fig. 5. Likewise, MAICP is less accurate while JRMPC schemes and MATrICP provide very good reconstructions.

Fig. 6 shows cross-sections of the reconstructions obtained by the proposed and motion-averaging algorithms (best viewed on-screen). The more "clean" and solid the sketch, the more accurate the alignment. The algorithms achieve to correct the initial sketch of the pair-wise ICP method. A detailed look verifies the superiority of the proposed batch method and the potential of the incremental version. Note that down-sampling makes short lines intersect in the cross-sections, even when using the ground truth motions.

Despite its incremental nature, JRMPC-I achieves comparable reconstructions and closes the loop successfully. However, the components are not distributed in the same way. Fig. 7 shows the distribution of means after running both versions of JRMPC. Despite the refinement step and the rejection stage, the means seem to remain a little biased towards initial sets, which might be problematic with long data sequences. In such a scenario, one should enforce a constraint so that new components that replace the rejected ones entirely belong to new scene surface. Note that detecting the points that may belong to the new part of the scene/object when the depth sensor is moving is easy with today hybrid sensors that deliver visual and inertial data.

Table III provides a quantitative comparison between the methods when the point-sets are further perturbed with noise of SNR=25dB. As seen, the motion averaging methods seem to be more sensitive than the proposed ones. This is mainly because the GMM means get cleaned over time and the registration module in JRMPC is more robust to noise. As a consequence, even JRMPC-I outperforms the motion averaging methods. The presence of noise make the illustration of cross-sections and integrated models meaningless.

Remarkably, we experimentally found that fixing the variance for the initial iterations make JRMPC-B converge at a lower level. When the sets are roughly aligned, a fixed and reasonable value of the variance (that make each cluster include a few points) leads to better distributed means in terms of the object skeleton, which in turn lead to more accurate transformations. This is because the skeleton carries more informative points than the surface itself. Then, the update of the variance leads to better reconstruction of the object and to "safe" refinements of the rotations. From a mathematical point of view, this strategy helps avoiding local minima in the variance-rotations subspace.

B. Real Data

In [18], we tested JRMPC-B along with pairwise strategies on EXBI dataset, that contains depth data captured with a time-of-flight camera rigidly attached to two color cameras. Once calibrated [46], [47], this TOF-stereo sensor provides RGB-D data. The EXBI data consist of ten point clouds gathered

TABLE III: Performance of multi-view registration methods when points are perturbed by gaussian noise (SNR: 25dB).

	Raw-data	Initialization	MV-ICP [7]	MAICP [9]	MATrICP [36]	JRMPC-B	JRMPC-I
Bunny	3.45	2.42	2.66	2.87	2.37	**1.07**	1.41
Dragon	7.28	7.34	7.37	3.28	1.55	**0.64**	0.89
Happy Buddha	10.77	6.88	6.86	4.13	1.92	**1.18**	1.69

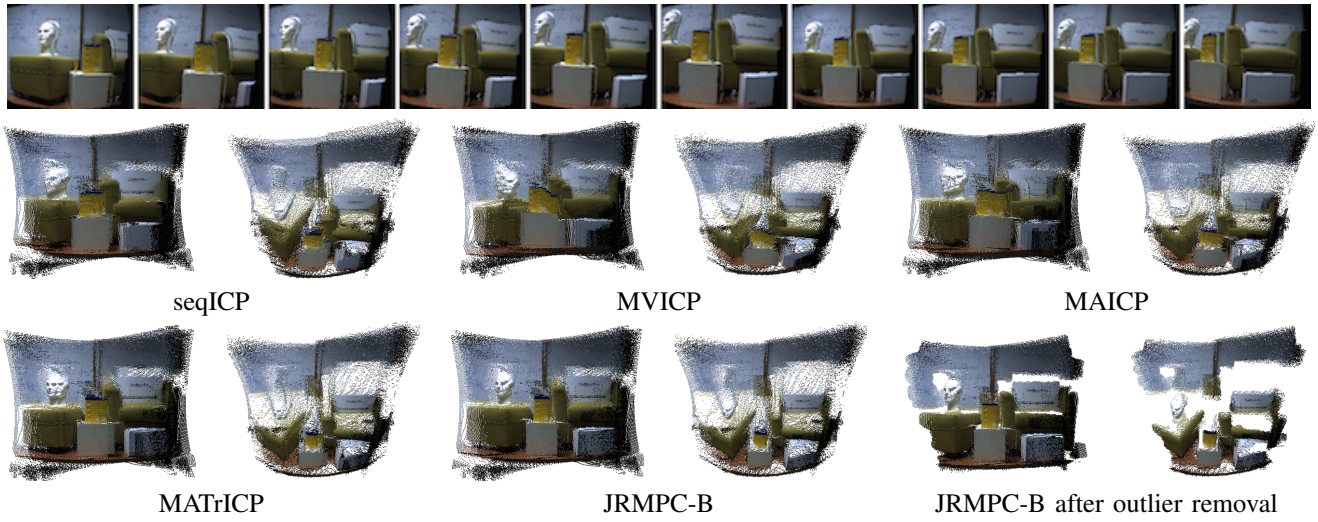


Fig. 8: Integrated point clouds from the joint registration of 10 TOF images that record a static scene (EXBI data-set). *Top*: color images that roughly show the scene content of each range image (occlusions due to cameras baseline may cause texture artefacts). *Bottom:* front-view and top-view of integrated sets after joint registration.

by manually moving the TOF-stereo sensor in front of a scene, e.g. Fig. 8. Each point cloud contains approximatively $25,000$ points. While JRMPC-B only uses the depth data, color information is used the final assessment and also shows the potential for fusing RGB-D data.

The comparison in [18] showed that, unlike JRMPC, all the pairwise strategies suffer from misalignments and need further processing, e.g. motion averaging.[5] Therefore, we test the performance of MAICP, MATrICP and MVICP on EXBI data-set and compare with JRMPC. SeqICP is used to roughly initialize the transformations of the point clouds.

Fig. 8 shows the front and top view of the integrated sets obtained by seqICP (initialization) as well as by MVICP, MAICP, MATrICP and JRMPC algorithms. Both versions of the proposed algorithm provide visually similar results. As verified, the motion averaging method cannot fully compensate for the misalignments of the initialization. This is shown even in front views, e.g. on the dummy head area. Again, MATrICP is more robust than MAICP, while MVICP clearly underperforms. The proposed scheme, however, achieves to register the point clouds accurately. Despite the large number of outliers, we are also able to get an outlier-free reconstruction of the scene based on the above thresholding principle. Of particular note, finally, is that JRMPC obtains these results with only 450 components, a fact that further validates its potential.

Finally, we evaluate the performance JRMPC-I with a large number of point clouds collected with a moving sensor, namely the TUM dataset [48]. In particular, we converted the depth sequences *fr1/desk* and *fr2/desk* from this dataset into two sequences of 570 and 2880 point sets, respectively. The first sequence includes several sweeps (local loops) over four desks in a typical office environment while the second sequence includes a full loop around a desk. The sequences are captured with a Kinect and ground-truth camera poses are provided with the help of a motion-capture system. As in the previous experiment, color data are not used by the registration algorithm.

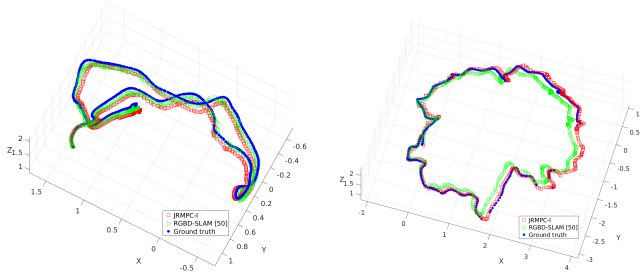
To enable the algorithm to deal with a large number of sets, we considered a hierarchical scheme with two modules, a front-end and a back-end module. The front-end registers groups of N_f successive point-sets with JRMPC-I and provides an outlier-free GMM, whose means are referred to as the *mean set*. The back-end module uses JRMPC-I on a temporal window of N_b mean sets, that is, a new mean set is integrated at every N_f times stamps and the local model instance of the window is refined with the batch method. We noticed that applying JRMPC-B on a small number of temporally integrated sets, e.g. one down-sampled registered set per 100 point sets, further improves the alignment.

All the initial point sets have been downsampled by a factor of 50 before running the algorithm. We used $N_b = 10$, while $N_f = 3$ and $N_f = 10$ for *fr1/desk* and *fr2/desk*, respectively owing to differences in motion patterns. The overlap between successive groups in the front-end module is one set. The number of components is 3000 and 6000 for the back-end and front-end modules, respectively. The batch method refines the window model for 50 iterations owing to its relatively small

[5] we also refer the reader to the supplementary material of [18]

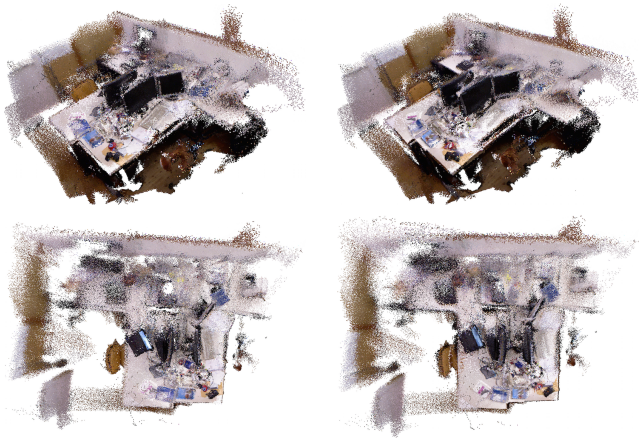
TABLE IV: RMSE (m) of translation for SLAM methods and for the proposed method for the TUM dataset.

	[49]	[50]	[51]	JRMPC-I
fr1/desk	0.016	0.020	0.026	0.047
fr2/desk	0.009	0.071	0.057	0.034



fr1/desk *fr2/desk*

Fig. 9: Camera trajectories obtained with JRMPC-I and with RGBD-SLAM [51].



JRMPC-I Ground-truth

Fig. 10: Dense point-cloud reconstruction obtained with JRMPC-I for the *fr1/desk* sequence.

window. An optimized implementation of this procedure may lead to real-time performance.

Table IV shows the performance of the proposed algorithm based on the protocol of [48]. Typically, the RMSE of the translation is used for evaluating SLAM methods. The error of JRMPC is computed for all frames while the other algorithms use only keyframes. We also provide the error of state-of-the-art RGB-D SLAM methods [49]–[51] as a reference (a direct comparison is not fair), as reported in [49]. Although SLAM methods use both modalities (RGB and depth) and invoke several modules to achieve accurate camera localization, the performance of the proposed algorithm is quite close to theirs.

Fig. 9 shows the camera trajectories obtained with the proposed algorithm and with RGBD-SLAM [51]. SLAM methods generally provide smooth trajectories owing to their internal tracking module and pose-graph optimization. Instead, our algorithm simply registers depth data in a model-to-frame manner and one may observe local perturbations. Fig. 10 shows the final alignment for the *fr1/desk* sequence obtained with JRMPC-I and the corresponding ground truth. Interestingly, the proposed scheme delivers promising reconstructions despite the fact that it only uses only depth data.

VIII. Conclusions

We presented a generative model and associated algorithms to jointly register multiple point sets. The vast majority of existing techniques select one of the sets as the model and attempt to align the other sets with this model. Instead, the proposed method treats all the point sets on an equal footing: points are realizations of a unique GMM and the registration is cast into a clustering problem. We formally derived an expectation-maximization algorithm that estimates the GMM parameters as well as the rotations and translations between each individual set and the initially unknown GMM means. An incremental version of the algorithm that efficiently integrates new point sets into the registration pipeline was also derived. We thoroughly validated the proposed algorithms on challenging data sets gathered with depth cameras, we compared them with several state-of-the-art methods, and we showed their potential for effectively fusing depth data. In the future we plan to investigate the use of more efficient representations of generative models, e.g. [52] and an incremental registration method allowing the number of clusters to grow.

References

[1] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 239–256, 1992.
[2] A. W. Fitzgibbon, "Robust registration of 2D and 3D point sets," *Image and Vision Computing*, vol. 21, no. 12, pp. 1145–1153, 2001.
[3] Y. Tsin and T. Kanade, "A correlation-based approach to robust point set registration," in *European Conference on Computer Vision*, 2004.
[4] A. Myronenko and X. Song, "Point-set registration: Coherent point drift," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010.
[5] R. Horaud, F. Forbes, M. Yguel, G. Dewaele, and J. Zhang, "Rigid and articulated point registration with expectation conditional maximization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 587–602, 2011.
[6] B. Jian and B. C. Vemuri, "Robust point set registration using gaussian mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1633–1645, 2011.
[7] J. Williams and M. Bennamoun, "Simultaneous registration of multiple corresponding point sets," *Computer Vision and Image Understanding*, vol. 81, no. 1, pp. 117–142, 2001.
[8] F. Wang, B. C. Vemuri, A. Rangarajan, and S. J. Eisenschenk, "Simultaneous nonrigid registration of multiple point sets and atlas construction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 2011–2022, 2008.
[9] V. M. Govindu and A. Pooja, "On averaging multiview relations for 3d scan registration," *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1289–1302, 2014.
[10] G. Blais and M. D. Levine, "Registering multiview range data to create 3d computer objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 820–824, 1995.
[11] T. Masuda and N. Yokoya, "A robust method for registration and segmentation of multiple range images," *Computer Vision and Image Understanding*, vol. 61, no. 3, pp. 295–307, 1995.
[12] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera," in *ACM Symposium on UIST*, 2011.

[13] R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau, "Towards a general multi-view registration technique," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 5, pp. 540–547, 1996.
[14] U. Castellani, A. Fusiello, and V. Murino, "Registration of multiple acoustic range views for underwater scene reconstruction," *Computer Vision and Image Understanding*, vol. 87, no. 1-3, pp. 78–89, 2002.
[15] D. F. Huber and M. Hebert, "Fully automatic registration of multiple 3d data sets," *IVC*, vol. 21, no. 7, pp. 637–650, 2003.
[16] S.-W. Shih, Y.-T. Chuang, and T.-Y. Yu, "An efficient and accurate method for the relaxation of multiview registration error," *IEEE Transactions on Image Processing*, vol. 17, no. 6, pp. 968–981, 2008.
[17] J. D. Banfield and A. E. Raftery, "Model-based Gaussian and non-Gaussian clustering," *Biometrics*, vol. 49, no. 3, pp. 803–821, 1993.
[18] G. Evangelidis, D. Kounades-Bastian, R. Horaud, and E. Psarakis, "A generative model for the joint registration of multiple point sets," in *European Conference on Computer Vision*, 2014.
[19] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and Vision Computing*, vol. 10, no. 3, pp. 145–155, 1992.
[20] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek, "The trimmed iterative closest point algorithm," in *IEEE International Conference on Pattern Recognition*, 2002.
[21] A. V. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Robotics: Science and Systems*, 2009.
[22] J. Yang, H. Li, and Y. Jia, "Go-ICP: solving 3D registration efficiently and globally optimally," in *IEEE International Conference on Computer Vision*, 2013.
[23] W. Wells III, "Statistical approaches to feature-based object recognition," *International Journal of Computer Vision*, vol. 28, no. 1/2, pp. 63–98, 1997.
[24] S. Granger and X. Pennec, "Multi-scale EM-ICP: A fast and robust approach for surface registration," in *European Conference on Computer Vision*, 2002.
[25] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Computer Vision and Image Understanding*, vol. 89, no. 2-3, pp. 114–141, 2003.
[26] J. Hermans, D. Smeets, D. Vandermeulen, and P. Suetens, "Robust point set registration using EM-ICP with information-theoretically optimal outlier handling," in *IEEE Computer Vision and Pattern Recognition*, 2011.
[27] R. Benjemaa and F. Schmitt, "A solution for the registration of multiple 3d point sets using unit quaternions," in *European Conference on Computer Vision*, 1998, pp. 34–50.
[28] T. Masuda, "Registration and integration of multiple range images by matching signed distance fields for object shape modeling," *Computer Vision and Image Understanding*, vol. 87, no. 1, pp. 51–65, 2002.
[29] G. C. Sharp, S. W. Lee, and D. K. Wehe, "Multiview registration of 3d scenes by minimizing error between coordinate frames," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1037–1050, August 2004.
[30] S. Krishnan, P. Y. Lee, and J. B. Moore, "Optimisation-on-a-manifold for global registration of multiple 3d point sets," *Int. J. Intelligent Systems Technologies and Applications*, vol. 3, no. 3/4, pp. 319–340, 2007.
[31] A. Torsello, E. Rodol, and A. A, "Multiview registration via graph diffusion of dual quaternions," in *IEEE Computer Vision and Pattern Recognition*, 2011.
[32] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2233–2246, 2012.
[33] D. Thomas, Y. Matsushita, and A. Sugimoto, "Robust simultaneous 3D registration via rank minimization," in *IEEE 3DIMPVT*, 2012, pp. 33–40.
[34] V. M. Govindu, "Combining two-view constraints for motion estimation," in *IEEE Computer Vision and Pattern Recognition*, 2001.
[35] R. Hartley, J. Trumpf, Y. Dai, and H. Li, "Rotation averaging," *International Journal of Computer Vision*, vol. 103, no. 3, pp. 267–305, 2013.
[36] Z. Li, J. Zhu, K. Lan, C. Li, and C. Fang, "Improved techniques for multi-view registration with motion averaging," in *IEEE Conference on 3D Vision*, 2014.
[37] D. Chetverikov, D. Stepanov, and P. Krsek, "Robust Euclidean alignment of 3D point sets: the trimmed iterative closest point algorithm," *Image and Vision Computing*, vol. 23, no. 3, pp. 299–309, 2005.
[38] G. Jacob, "Registration of multiple point sets using the EM algorithm," in *IEEE International Conference on Computer Vision*, 1999.
[39] Y. Cui, S. Schuon, S. Thrun, D. Stricker, and C. Theobalt, "Algorithms for 3D shape scanning with a depth camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 5, pp. 1039–1050, 2013.
[40] X. Mateo, X. Orriols, and X. Binefa, "Bayesian perspective for the registration of multiple 3d views," *Computer Vision and Image Understanding*, vol. 118, pp. 84 – 96, 2014.
[41] M. Danelljan, G. Meneghetti, F. Khan, and M. Felsberg, "A probabilistic framework for color-based point set registration," in *IEEE Computer Vision and Pattern Recognition*, 2016.
[42] T. Shiratori, J. Berclaz, M. Harville, C. Shah, T. Li, Y. Matsushita, and S. Shiller, "Efficient large-scale point cloud registration using loop closures," in *IEEE Conference on 3D Vision*, 2015.
[43] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
[44] X.-L. Meng and D. B. Rubin, "Maximum likelihood estimation via the ECM algorithm: a general framework," *Biometrika*, vol. 80, pp. 267–278, 1993.
[45] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 376–380, 1991.
[46] M. Hansard, R. Horaud, M. Amat, and G. Evangelidis, "Automatic detection of calibration grids in time-of-flight images," *Computer Vision and Image Understanding*, vol. 121, pp. 108–118, 2014.
[47] M. Hansard, G. Evangelidis, Q. Pelorson, and R. Horaud, "Cross-Calibration of Time-of-flight and Colour Cameras," *Computer Vision and Image Understanding*, vol. 134, pp. 105–115, 2015.
[48] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Intelligent Robots and Systems*, 2012.
[49] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An open-source SLAM system for monocular, stereo and RGB-D cameras," *arXiv*, 2016.
[50] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "Elasticfusion: Real-time dense SLAM and light source estimation," *International Journal of Robotics Research*, vol. 35, no. 14, pp. 1697–1716, 2016.
[51] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-D mapping with an RGB-D camera," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 177–187, 2014.
[52] B. Eckart, K. Kim, A. Troccoli, and A. Kelly, "Accelerated generative models," in *IEEE Computer Vision and Pattern Recognition*, 2016.

Georgios D. Evangelidis received his BSc, MSc and PhD degree in computer science in 2001, 2003 and 2008 respectively from the University of Patras, Greece. From 2007 to 2009 he was an adjunct lecturer of the Technological Institute of Larissa, Greece. During 2009-2010, he was an ERCIM (Alain Bensoussan) Fellow and joined the Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS) in Sankt Augustin, Germany, as a postdoctoral researcher. From 2012 to 2015, he was a researcher at the Perception Team of INRIA Grenoble, France. His research interests are in the area of computer vision and machine learning and include 3D reconstruction, Depth sensors, Action recognition and Visual-Inertial Odometry. Since January 2016, he is a senior computer vision scientist of Daqri Research Center in Vienna, Austria.

Radu Horaud received the B.Sc. degree in Electrical Engineering, the M.Sc. degree in Control Engineering, and the Ph.D. degree in Computer Science from the Institut National Polytechnique de Grenoble, France. In 1982-1984 he was a post-doctoral fellow with the Artificial Intelligence Center, SRI International, Menlo Park, CA. Currently he holds a position of director of research with INRIA Grenoble, where he is the founder and head of the PERCEPTION team. His research interests include computer vision, machine learning, audio signal processing, audiovisual analysis, and robotics. R. Horaud is area editor of *Computer Vision and Image Understanding*, advisory board member of *International Journal of Robotics Research*, and associate editor of *International Journal of Computer Vision*. He was program co-chair of IEEE ICCV'01 and of ACM ICMI'15. In 2013 Radu Horaud was awarded an ERC Advanced Grant for his project *Vision and Hearing in Action* (VHIA) and in 2017 he was awarded an ERC Proof of Concept (PoC) grant for the VHIALab project.