2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Daejeon Convention Center
October 9-14, 2016, Daejeon, Korea

Point Clouds Registration with Probabilistic Data Association

Gabriel Agamennoni[1], Simone Fontana[2], Roland Y. Siegwart[1] and Domenico G. Sorrenti[2]

Abstract— **Although Point Clouds Registration is a very well studied problem, with many different solutions, most of the approaches in the literature aims at aligning two dense point clouds. Instead, we tackle the problem of aligning a dense point cloud with a sparse one: a problem that has to be solved, for example, to merge maps produced by different sensors, such as a vision-based sensor and laser scanner or two different laser-based sensors. The most used approach to point clouds registration, Iterative Closest Point (ICP), is also applicable to this sub-problem. We propose an improvement over the standard ICP data association policy and we called it Probabilistic Data Association. It was derived applying statistical inference techniques on a fully probabilistic model. In our proposal, each point in the source point cloud is associated with a set of points in the target point cloud; each association is then weighted so that the weights form a probability distribution. The result is an algorithm similar to ICP but more robust w.r.t. noise and outliers. While we designed our approach to deal with the problem of dense-sparse registration, it can be successfully applied also to standard point clouds registration.**

I. INTRODUCTION

A. Related work

The problem of aligning a sparse point cloud with a dense one is, to our knowledge, little explored. Most of the works, indeed, tackle the problem of aligning two generic point clouds, without any reference to their density or to the density difference between the two. Most of these techniques, however, may be used to solve our problem too. One large category of point clouds registration techniques are those exploiting some kind of geometric features, which are, basically, a representation of the underlining surface. Thus they usually require, in order to compute the descriptors, the surface normals. Examples of this kind of features are PFH [1] and their faster variant FPFH [2], or angular-invariant features, [3]. Unfortunately, geometric features cannot be used to solve the proposed problem because a sparse cloud may not have enough information to produce meaningful features; i.e., it could be locally too sparse to represent the surface in an informative way. This is the case, for example, of sparse keypoints maps produced by vision-based systems. On the other hand, while a keypoint map usually has some kind of visual features associated with the keypoints, this is not necessary true for a dense point cloud, that may include only a set of 3D points.

Sehgal et al. developed an approach for point clouds registration that uses SIFT features extracted from a bi-dimensional image generated from the point cloud [4]. In a similar way also NARF features extracted from a range image [5] can be used for point clouds registration. Also these techniques are not suitable to our case, because a range image, like those considered in such contributions, cannot be generated starting from a very sparse map.

Our goal was to design a general technique that could work using only two sets of points, even when features are not available. For the same reason we do not make any reference to RGB-D based registration techniques: while this kind of sensors is nowadays common, LiDARs that do not provide also a RGB image are still widespread and still have advantages over most RGB-D sensors, regarding precision, accuracy, range and field of view.

Another large category of point clouds registration algorithms are those related, in some way, to the Iterative Closest Point algorithm (ICP). ICP was developed independently by Besl and McKay [6], Chen and Medioni [7], and Zhang [8]. Although its first introduction was in 1991, it is still the *de facto* standard for point clouds registration. ICP assumes that the two point clouds are already roughly aligned and aims at finding the rigid transformation, i.e., a rototranslation, that best refines the alignment. One point cloud is usually called the source cloud, while the other is the target cloud. The aim of ICP is to align the source cloud with the target cloud and, to do so, it repeats the following steps until convergence:

1) For each point x_j in the source cloud find the closest point y_k in the target cloud
2) Find the best values for R and T (rotation and translation) that minimize
$$\sum_j \|Rx_j + T - y_k\|^2 \tag{1}$$
3) Transform the source cloud using R and T

The algorithm may end, for example, after a predefined number of iterations, when the sum of the residuals defined by Equation 1 becomes smaller than a certain threshold or when the difference of the rototranslations between two consecutive steps becomes smaller than a threshold.

Many different variants of ICP have been proposed; usually they aim at speeding up the algorithm or at improving the quality of the result. For an extensive review and comparison of ICP variants, see [9]. In this paper we will refer to the basic ICP algorithm (also called Point-to-Point ICP) as implemented in the pcl library [10], because it is still the most used version and it is easily available for comparisons.

Although ICP was not specifically designed to deal with sparse point clouds, it is still suitable for solving the problem. However, in such conditions, it has some disadvantages. Two point clouds will never align perfectly and a point in a point

[1] ETH-Zürich gabriel.agamennoni@mavt.ethz.ch, rsiegwart@ethz.ch
[2] Università degli Studi di Milano - Bicocca [simone.fontana, sorrenti]@disco.unimib.it

978-1-5090-3761-2/16/$31.00 ©2016 IEEE

cloud will never exactly correspond to another point in the other; this is particularly true in our case, both because dense and sparse point clouds are often produced with different kind of sensors (thus, with different resolutions and scanning patterns) and also because usually a large amount of noise is present. The problem becomes even harder when a big amount of distortion is present. Generalized ICP [11], a version of ICP using Point-to-Plane associations, is aimed at facing these problems. However, as we will show with our experiments, our approach slightly outperforms it. Moreover, as shown in [9], Generalized ICP does not perform better than standard ICP in outdoor, unstructured environments.

Another important technique used for point clouds registration is called Normal Distribution Transform (NDT), [12]. This technique was originally developed to register 2D laser scans, but has been successfully applied also to 3D point clouds, [13]. Unlike ICP, it does not establish any explicit correspondence between points. Instead, the source point cloud or laser scan is subdivided into cells and a normal distribution is assigned to each cell, so that the points are represented by a probability distribution. The matching problem is then solved as a maximization problem, using Newton's algorithm.

The problem of aligning two point clouds with different densities is very relevant to the robotics community. For example it has to be solved when aligning two maps produced by different robots with different sensors, such as a Kinect and a LiDAR. The former, indeed, produces much denser point clouds than the latter. Another scenario that can be described as a problem of dense-sparse registration is when a robot needs to localize in a map produced with a different sensor (thus, maybe, with a different density). Eventually also the problem of calibrating two different sensors on the same robot can be reconducted to a dense-sparse registration problem.

B. Our contribution

In this paper we present a novel approach for introducing robustness in a point clouds registration algorithm. Although our approach is motivated by the problem of aligning a dense point cloud with a sparse one, it can also be used as a substitute of ICP for generic point clouds registration. For this reason, in the following sections, we won't make any reference to sparse and dense clouds, but we will use the usual ICP notation (target and source point clouds). Our main contribution is a new data association policy, which we called Probabilistic Data Association, because it was derived by applying statistical inference techniques on a fully probabilistic model. The final result is an algorithm similar to ICP, but more robust w.r.t. noise and outliers.

C. Outline

In Section II we introduce the probabilistic model from which we derived our algorithm. In Section III we explain how we reformulate the point clouds registration problem as an Expectation Maximization problem. In Section IV we give a detailed description of our new data association policy and of the resulting optimization problem, while in Section V we compare our approach with other point clouds registration techniques and give quantitative results on datasets.

II. MODEL DEFINITION

A. Problem statement

Suppose that we have two point clouds representing the same scene, X and Y, and with their points $x_1, ..., x_n$ and $y_1, ..., y_m$. These may have been acquired with different sensors (e.g., a camera and a laser scanner) or with the same sensor at different times; they may also have very different densities, such as a dense, laser scanner-produced, point cloud and a sparse keypoints map. Our approach is completely agnostic w.r.t. these characteristics. We want to recover the rigid transformation between the two point clouds, i.e., the roto-translation that best aligns X with Y. The limitation to only rigid transformations has, of course, some effects on the quality of the result; indeed it is a simplification often used in the literature and usually, as long as the two point clouds are not heavily distorted, leads to good results.

For each point y_k in Y, we may define

$$y_k = Rx_j + T \tag{2}$$

where x_j is the point in the point cloud X corresponding to point y_k in the point cloud Y and R and T are, respectively, the rotation and translation that align X with Y. In practice, the true point associations are unknown and Equation 2 is never exactly holding, because of noise. Uncertainty due to sensor noise is often treated as a random variable, e.g., an additive white Gaussian noise combined with a deterministic value. We argue that association ambiguity is also a source of uncertainty and therefore it should also be treated as a random variable.

B. Probabilistic model

In order to reason about sensor noise and data association uncertainty simultaneously, we define a probabilistic model. Probabilistic models are attractive because:

1) they are interpretable and extensible;
2) they allow well-behaving optimization criteria, i.e. the negative log-likelihood;

A probabilistic model is defined by a series of statements about the distribution of the variables involved in the model. For example, we define:

$$p(y_k | x_j, a_{jk} = 1) \sim \mathcal{N}(Rx_j + T, \Sigma) \tag{3}$$

where $a_{jk} = 1$ if point x_j corresponds to point y_k. This means that, if we were certain that point x_j in X corresponds to point y_k in Y, then y_k would follow a multi-variate normal distribution with mean $Rx_j + T$ and covariance Σ. To complete the model we must place a prior distribution over a_{jk}. Suppose we have a set C_k of candidate points from X corresponding to y_k. Then, we may define

$$P(a_{jk}) = \left\{ \begin{array}{cc} |C_k|^{-1} & j \in C_k \\ 0 & j \notin C_k \end{array} \right\} \tag{4}$$

This means that, a priori, all the points in C_k are equally likely to be associated to y_k.

In practice we prefer a slightly more sophisticated model that should account for outliers produced by sensor noise. For this reason, instead of a Gaussian, a t-distribution is more appropriate. We redefine the model as follows

$$p(y_k|x_j, a_{jk} = 1) \sim \mathcal{T}(Rx_j + T, \Sigma, \nu) \tag{5}$$

where $\mathcal{T}$ denotes the family of multi-variate t-distributions and ν represents its degree of freedoms. Equation 5 states that y_k conditioned to being the correspondent of x_j is t-distributed. A t-distribution is a heavy-tailed distribution and ν controls the weight of the tails. For $\nu \to \infty$ the t-distribution reduces to a Gaussian. For finite ν it assigns a non-negligible probability to the tails, thus implicitly taking into account for outliers, without the need to pre-filter them or to treat them as a special case.

It is convenient to re-parametrize the model. Namely, the t-distribution in Equation 5 is equivalent to

$$p(y_k|x_j, a_{jk} = 1, w_k) \sim \mathcal{N}(Rx + T, \frac{\Sigma}{w_k}) \tag{6a}$$

$$w_k \sim \mathcal{G}(\frac{\nu}{2}, \frac{\nu}{2}) \tag{6b}$$

where $\mathcal{G}(a, b)$ denotes a Gamma distribution with shape a and rate b [14]. The convolution of 6a and 6b produces Equation 5. The weight w_k is an auxiliary variable that arises from the parametrization. This particular parametrization is convenient because, if we knew a_{jk} and w_k, then the negative log-likelihood would be a quadratic function of x_j and we could run a non-linear least-squares solver.

III. POINT CLOUD REGISTRATION AS AN EM PROBLEM

The model defined in Section II-B can be used to recover the rigid transformation between the two point clouds using an optimization algorithm. It needs a rough initial guess and it iteratively improves it, simultaneously estimating the transformation and the point associations.

Expectation-maximization (EM), [15], is a procedure that can be used to iteratively estimate the marginal log-likelihood of the data, given a set of parameters. Each EM iteration consists of two steps: the E-step and the M-step. The E-step effectively estimates the values of the hidden variables by evaluating expectations, while the M-step updates the parameters in order to decrease the expected negative log-likelihood. EM is well-suited for models containing latent variables, such as ours. In our case the latent variable is the auxiliary weight w_k. The negative log-likelihood we want to minimize with EM is given by

$$l(x) = -\ln \sum_{k=1}^{m} \sum_{j=1}^{n} \int_{w_k} p(y_k|x_{C_k}, a_{jk}, w_k) p(a_{jk}) p(w_k) dw_k \tag{7}$$

where x_{C_k} is the set of all the points x_j such that $j \in C_k$ (see Section II-B) and p denotes the probability density function implied by 6a and 6b. Minimizing the negative log-likelihood directly is difficult due to non-convexity. It is much easier to minimize a convex upper bound on Equation 7. Let q_k be an arbitrary probability density function of a_{jk} and w_k. Then, applying Jensen's inequality leads to

$$l(x) \leq \sum_{k=1}^{m} b_k(x_{C_k}, q_k) \tag{8}$$

where

$$b_k(x_{C_k}, q_k) = \\ -\sum_{j=1}^{n} \int_{w_k} q_k(a_{jk}, w_k) ln(p(y_k|x_{C_k}, a_{jk}, w_k)) d_{w_k} + \\ \sum_{j=1}^{n} \int_{w_k} q_k(a_{jk}, w_k) ln(\frac{q_k(a_{jk}, w_k)}{p(a_{jk}) p(w_k)}) dw_k \tag{9}$$

The inequality in 8 defines an upper bound on $l(x)$. If $q_k(a_{jk}, w_k) = p(a_{jk}, w_k|x_{C_k}, y_k)$, then the bound becomes tight and the inequality becomes an equality. If we evaluate the expectations in Equation 9 and retain only the terms involving the points we obtain

$$b_k(x_{C_k}, q_k) = \frac{1}{2} \sum_{j \in C_k} \rho_{jk} r_k^2(x_j) \tag{10}$$

where

$$r_k^2(x_j) = \|y_k - (Rx_j + T)\|^2 \tag{11}$$

is the squared Euclidean norm of the residual, and

$$\rho_{jk} = \sum_{j} \int_{w_k} \delta(a_{jk}, j) w_k q(a_{jk}, w_k) dw_k \tag{12}$$

is the residual weight. Hence for a fixed q_k, b_k is a quadratic-composite function of the points positions in the source cloud. It is also a local function, depending only on the candidate corresponding points for the *k*th point in the target cloud.

EM minimizes the upper bound coordinate-wise. The E-step minimizes the bound with respect to q and computes the residual weights, hence it is equivalent to the re-weighting step. The M-step updates the objective function along a descent direction. Therefore, applying EM is equivalent to solving an iteratively re-weighted, non-linear least-squares problem.

1) The E-step: For fixed x_j, the minimum of b_k occurs when

$$q_k(a_k, w_k) = p(a_{jk}, w_k|x_{C_k}, y_k) \tag{13}$$

In other words, minimizing the upper bound with respect to q_k is the same as computing the joint posterior distribution over all the $a_j k$ and w_k , given x_j. Due to conjugacy, the posterior has the same mathematical form as the prior, i.e. multinomial Gamma. Specifically,

$$P(a_{jk}|x_{C_k}, y_k) \propto t(y_k, Rx_j + T, \Sigma, \nu) \tag{14a}$$

$$w_k|a_{jk}, x_{C_k}, y_k \sim \mathcal{G}\left(\frac{\nu + d}{2}, \frac{\nu + r_k^2(x_j)}{2}\right) \tag{14b}$$

where t is the density function of the multivariate t-distribution, and the proportionality sign implies a normalization constant such that $\sum_{j \in C_k} P(a_{jk}|x_{C_k}, y_k) = 1$. Evaluating the expectation in Equation 12 yields

$$\rho_{jk} = P(a_{jk}|X_{C_k}, y_k) E[w_k|x_{C_k}, y_k, a_{jk}] \tag{15}$$

where

$$E[w_k|x_{C_k}, y_k, a_{jk}] = \frac{\nu + d}{\nu + r_k^2(x_j)} \tag{16}$$

follows from the properties of the Gamma distribution.

2) The M-step: There are several different ways of updating the rotation and translation in our model. A trust-region method such as Levenberg-Marquardt [16] solves a sequence of quadratic sub-problems of the form

$$\min_{\Delta x_j} \sum_{k,j \in C_k} \rho_{jk} \, ||y_k - (Rx_j + T)||^2 \tag{17}$$

The solution to a sub-problem is an update step which is applied to the current estimate of the rotation and translation between the clouds.

IV. IMPLEMENTATION

As can be seen from Section III, our approach differs from ICP in the data association. In ICP each point in the source point cloud is associated only with a point in the target point cloud, while the proposed algorithm associates a point in the source point cloud with a set of points in the target cloud. Moreover, the candidate associations are not changed at every iteration, but remain the same for the whole duration of the algorithm. What is changed, instead, are the weights of the associations, that are updated during the *expectation* phase of the EM algorithm. The two different data association methods are depicted in figure 1.

The candidate points to be associated may be found in different ways, for example nearest neighbours search, feature matching, etc... We found that, for the problem of sparse-dense registration and given a reasonable initial hypothesis on the transformation, the nearest neighbours search proved to be good enough, while remaining very fast to compute, in contrast with feature extraction and matching, which is usually a slow process. Feature-based data association is not an option for our application, since sparse point clouds usually do not contain enough information to extract discriminative geometric features. However, our approach can potentially accommodate feature matching as prior information. For instance, in equation 4, the prior association probabilities can be scaled according to a non-negative feature similarity metric.

For each point x_j in the source point cloud, we look for the n nearest points, $y_0, ..., y_n$, in the target. For each of these points y_k, with $0 \leq k \leq n$, we define an error term given by

$$||y_k - (Rx_j + T)||^2 \tag{18}$$

Equation 18 represents the squared error between the point y_k in the target point cloud and the associated point x_j from the source point cloud, transformed using the current estimate of the rototranslation.

Our point cloud registration algorithm is formed by an optimization problem, whose error terms are calculated according to Equation 18 and that is then solved using a suitable method (such as Levenberg-Marquardt). However, given a set of points associated to x_j, not all the corresponding error terms should have the same weight. Intuitively we want to give more importance to the associations that are in accordance with the current estimate of the transformation and lower importance to the others. Thus, using the model described in section III, the weight of the error term $\|y_k - (Rx_j + T)\|^2$ is given by

$$w_{kj} \propto e^{-\frac{\|y_k - (Rx_j + T)\|^2}{2}} \tag{19}$$

where the proportionality implies a normalization among all the error terms associated with x_j so that their weights represents a probability distribution. Equation 19 is derived from the EM algorithm, with an additive Gaussian noise model.

The Gaussian in Equation 19 works well, provided there are no outliers and all points in the source point cloud have a corresponding point in the target point cloud. However, as already described in section II-B, a t-distribution is a better choice in presence of outliers, especially when there is lot of distortion in one of the maps that, thus, cannot be aligned perfectly. Consequently, a more robust equation for the weights, basing on the t-distribution, is given by

$$p_{kj} \propto \left(1 + \frac{\|y_k - (Rx_j + T)\|^2}{\nu}\right)^{-\frac{\nu + d}{2}} \tag{20}$$

$$w_{kj} = p_{kj} \frac{\nu + d}{\nu + \|y_k - (Rx_j + T)\|^2} \tag{21}$$

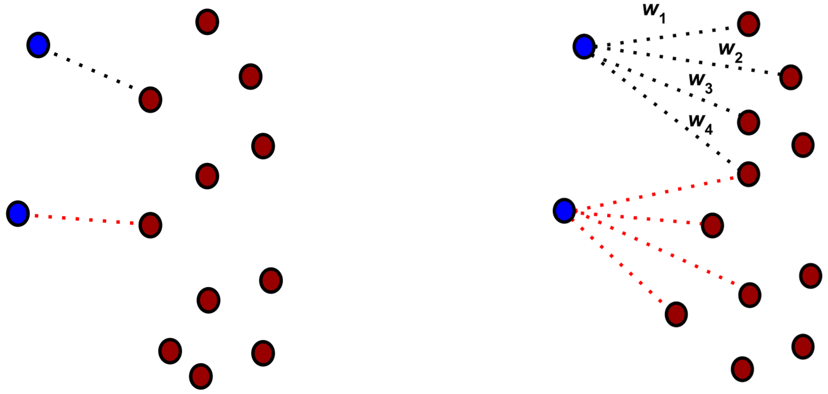
where ν is the degree of freedom of the t-distribution and d the dimension of the error terms (in our case 3, since we are operating with points in the 3D space).

However, in order to calculate the weights, we need an estimate of the rotation and translation, but these are estimated by solving the optimization problem whose error terms are weighted with the weights we want to calculate. Hence our problem cannot be formulated as a simple least-square error problem, but it has to be reformulated as an Expectation Maximization problem. During the Expectation phase the latent variables, in our case the weights, are estimated using the previous iteration estimate of the target variables (the rotation and translation), while during the Maximization phase, the problem becomes a least-square error optimization problem, with the latent variables assuming the values estimated during the Expectation phase.

V. EXPERIMENTAL RESULTS

A. Experimental Setup

We tested the proposed approach and compared it against other techniques. Here we will show results on three pairs



(a) ICP Data Association

(b) Probabilistic Data Association

Fig. 1. The two different data association policies

of point clouds, acquired with different kind of sensors. The first two datasets, called “office” and “corridor”, represent a typical office environment, with walls, desks, chairs and, in a case, a robot (Figures 2a and 2b). Each dataset is composed of two point clouds, one acquired with a Kinect 2, the other one with a Velodyne VLP-16. The Kinect 2 produces very dense, although noisy, point clouds and has a relatively narrow field of view (43° vertically and 57° horizontally). The Velodyne VLP-16 is a laser scanner that has sixteen scanning planes and an horizontal field of view of 360°. The produced point clouds are definitely less dense than those produced with a Kinect 2, but are also less affected by noise. Thus, the point clouds produced with the VLP-16 are the sparse point clouds and represent a wider view of the environment, while those produced by the Kinect 2 are the dense point clouds and represent a more detailed and narrower view.

The third dataset contains data provided to us by the UASTech Laboratory of the Linköping University, in the context of the SHERPA European project (http://www.sherpa-project.eu). The dataset has been acquired with an aerial robot and represents a big rural area, with trees and some buildings, (Figure 3). It is composed of two point clouds: the first one is relatively sparse and has been produced with a LiDAR, while the second one is much denser and has been produced using photogrammetry with images from a camera. The second point cloud is very noisy and is heavily distorted at the edges. Both sensors were mounted on the same robot, however, due to noise and distortion in the second cloud, the two point clouds cannot be aligned perfectly applying a rigid transformation. Therefore, this third dataset represents a very challenging test bench.

For all the datasets, we precisely aligned the two point clouds, in order to have a ground truth to use to quantitatively compare the various techniques.

Since our approach, like the other we compare to, can be used only for fine-registration (that is, it needs a rough initial guess), the transformation between the two point clouds in the same dataset is relatively small (Figure 4). Although, the datasets are very challenging because the two point clouds present very different scanning patterns, different densities and, in some cases, have a large amount of noise and distortion.

B. Results

Since our objective was to compare our probabilistic approach to other point clouds registration techniques, we used various algorithms on the same data and compared the results. The techniques we compared are:

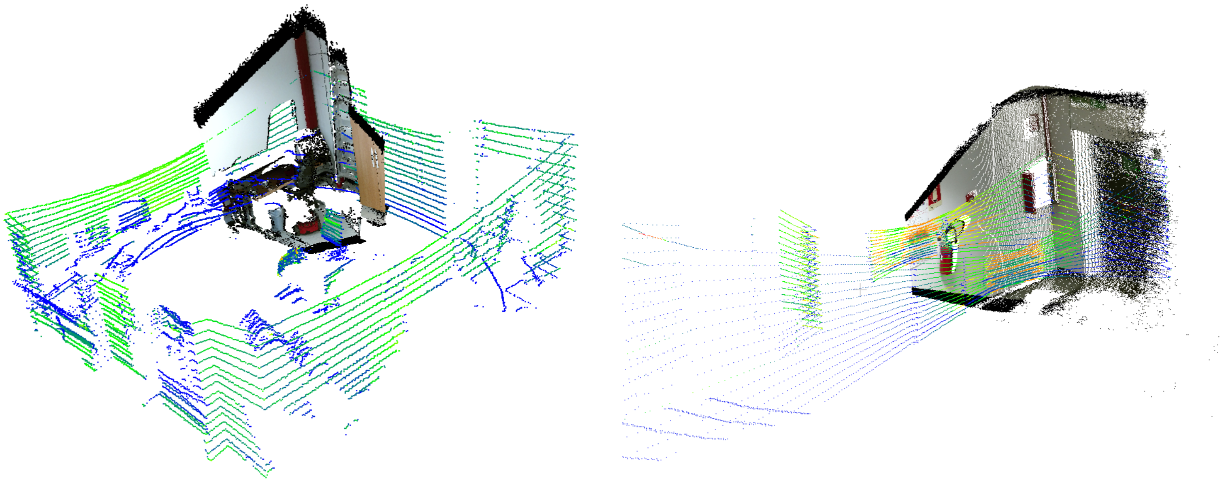
- Iterative Closest Point (ICP)
- Generalized Iterative Closest Point (G-ICP)
- Normal Distribution Trasform (NDT)
- The proposed “probabilistic” approach

For each of these algorithms we used the implementation available in the PCL Library, [10]. The metric used to compare the results is the mean distance between the points in the source point clouds, as aligned by an algorithm, and the ground truth. That is:

$$\frac{\sum_{i=0}^{N} ||x_i - g_i||}{N} \tag{22}$$

Where x_i is a point in the registered source point cloud, g_i is the corresponding point in the ground truth and N is the cardinality of the point clouds. The smaller the value of Equation 22 the better the result.

The results of our experiments are shown in Tables I to III. One problem with our proposal is that, since the point associations do not change at every iteration, it is very sensible to the initial data association. To choose the points in the target cloud to be associated to a particular point in the source cloud, we simply look at those within a certain distance (the *radius* parameter of the algorithm) or, in a way similar to ICP, to the k-nearest neighbours. If among all the points in the target cloud associated to a particular point in the source cloud the right data association is not present, whenever this happens for a large number of points, the algorithm will not converge to the right solution. One way to avoid this problem consists in applying our probabilistic algorithm several times each time using as input



(a) Office

(b) Corridor

Fig. 2. The point clouds from the “office” and “corridor” datasets. The RGB-D point cloud has been produced with a Kinect 2, the other one with a Velodyne VLP-16.

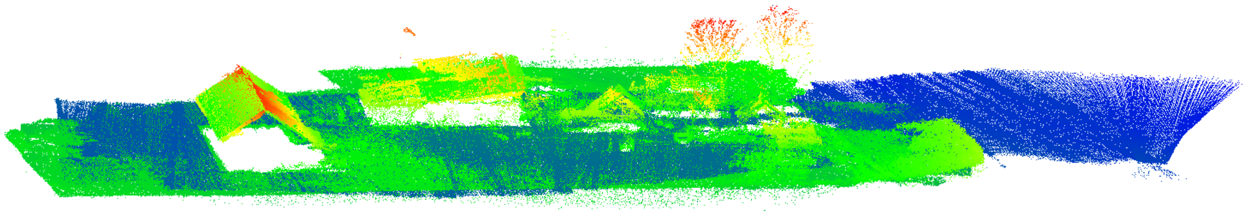
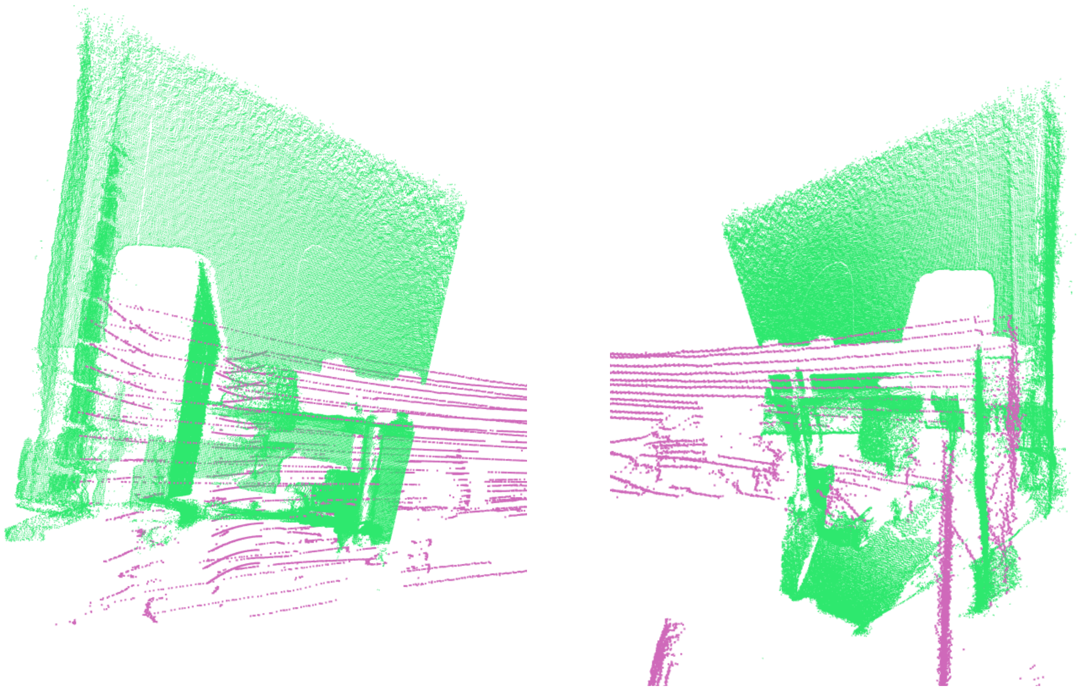


Fig. 3. The point clouds from the “Linköping” dataset. The blue point cloud has been acquired with a LiDAR, the other with a camera using a photogrammetry technique.



(a)

(b)

Fig. 4. Two different views of the misalignment between the two point clouds in the “office” dataset. Flat colours have been used for clarity.

TABLE I

Tests with the office dataset

Algorithm	Residual Mean Distance
Probabilistic	0.386401
Prob. (multiple runs)	0.040809
ICP	0.553467
G-ICP	0.074794
NDT	0.783207

TABLE II

Tests with the corridor dataset

Algorithm	Residual Mean Distance
Probabilistic	0.315502
Prob. (multiple runs)	0.073592
ICP	0.286967
G-ICP	0.100754
NDT	0.129964

TABLE III

Tests with the Linköping dataset

Algorithm	Residual Mean Distance
Probabilistic	1.842854
Prob. (multiple runs)	0.43761
ICP	1.680124
G-ICP	0.6527864
NDT	3.564592

the solution of the previous run and thus re-estimating the correspondences. In this way we use a technique very similar to ICP, where each iteration is a run of our algorithm. In this case, the difference w.r.t. ICP is that, instead of using a single nearest neighbour, we use a set of points, and each association is weighted as described. In the tables this technique is described with the phrase ”Multiple Runs”. When used in this way, the execution time of the algorithm increases greatly. As an example, for the corridor dataset, on a machine with an Intel i5-760 processor and 8 GB of RAM, the execution times are:

- $7741ms$ for the “Multiple Runs” mode
- $252ms$ in single run mode
- $70ms$ for ICP
- $2140ms$ for G-ICP
- $1902ms$ for NDT

These times are purely indicative, since they depends strongly from the problem that has to be solved.

The experiments show that the proposed probabilistic approach is not necessary better than the other point clouds registration techniques when used with a single iteration. However it provides much better results than all the other techniques when used in the “Multiple Runs” mode.

VI. CONCLUSIONS

We developed a novel algorithm aimed at solving point clouds registration problems, with particular attention to dense-sparse registration. Our solution differs from ICP basically in the data association: each point in the source point cloud is associated with a set of points in the target cloud. The association with each point is then weighted so that the weights in a given data association set form a probability distribution.

With the associations and the relative weights we build and solve an optimization problem.

The experiments show that the best way to use our approach is running it several times, updating the data association, thus using it as an iteration of an ICP-like technique. When used in this way our approach outperformed all the other point clouds registration techniques we compared.

References

[1] R. Rusu, N. Blodow, Z. Marton, and M. Beetz, “Aligning point cloud views using persistent feature histograms,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008. IROS 2008*, Sep. 2008, pp. 3384–3391.

[2] R. Rusu, N. Blodow, and M. Beetz, “Fast Point Feature Histograms (FPFH) for 3d registration,” in *IEEE International Conference on Robotics and Automation, 2009. ICRA ’09*, May 2009, pp. 3212–3217.

[3] J. Jiang, J. Cheng, and X. Chen, “Registration for 3-D point cloud using angular-invariant feature,” *Neurocomputing*, vol. 72, no. 1618,5-05-29, pp. 3839–3844, Oct. 2009.

[4] A. Sehgal, D. Cernea, and M. Makaveeva, “Real-Time Scale Invariant 3d Range Point Cloud Registration,” in *Image Analysis and Recognition*, ser. Lecture Notes in Computer Science, A. Campilho and M. Kamel, Eds. Springer Berlin Heidelberg, 2010, no. 6111, pp. 220–229.

[5] B. Steder, R. B. Rusu, K. Konolige, and W. Burgard, “Narf: 3d range image features for object recognition,” in *Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, vol. 44, 2010.

[6] P. Besl and N. D. McKay, “A method for registration of 3-d shapes,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 14, no. 2, pp. 239–256, Feb 1992.

[7] Y. Chen and G. Medioni, “Object modeling by registration of multiple range images,” in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*. IEEE, 1991, pp. 2724–2729.

[8] Z. Zhang, “Iterative point matching for registration of free-form curves and surfaces,” *International journal of computer vision*, vol. 13, no. 2, pp. 119–152, 1994.

[9] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, “Comparing ICP variants on real-world data sets,” *Autonomous Robots*, vol. 34, no. 3, pp. 133–148, Apr. 2013.

[10] R. B. Rusu and S. Cousins, “3d is here: Point cloud library (pcl),” in *International Conference on Robotics and Automation*, Shanghai, China, 2011 2011.

[11] A. Segal, D. Haehnel, and S. Thrun, “Generalized-ICP.” in *Robotics: Science and Systems*, vol. 2, 2009.

[12] P. Biber and W. Straßer, “The normal distributions transform: A new approach to laser scan matching,” in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 3. IEEE, 2003, pp. 2743–2748.

[13] H. Merten, “The three-dimensional normal-distributions transform,” *threshold*, vol. 10, p. 3, 2008.

[14] S. Kotz and S. Nadarajah, *Multivariate t-distributions and their applications*. Cambridge University Press, 2004.

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[16] J. J. Moré, “The levenberg-marquardt algorithm: implementation and theory,” in *Numerical analysis*. Springer, 1978, pp. 105–116.