Linewise Non-Rigid Point Cloud Registration

Linewise Non-Rigid Point Cloud Registration

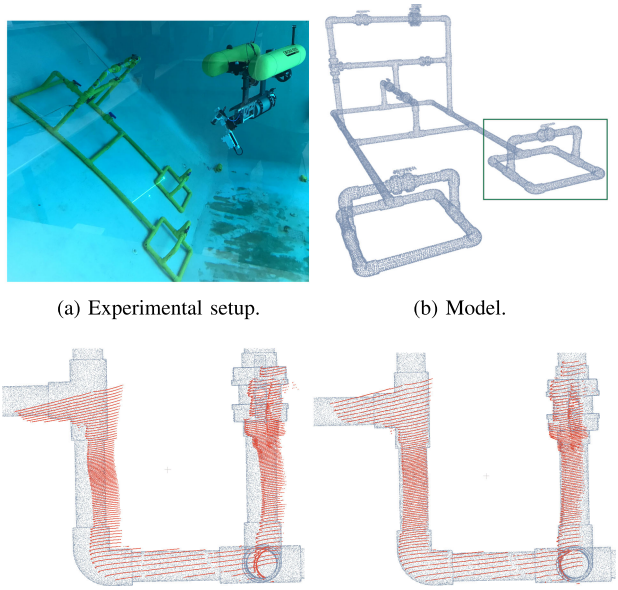
Miguel Castillón, Pere Ridao, *Member, IEEE*, Roland Siegwart, *Fellow, IEEE*, and César Cadena, *Member, IEEE*

***Abstract*—Robots are usually equipped with 3D range sensors such as laser line scanners (LLSs) or lidars. These sensors acquire a full 3D scan in a line by line manner while the robot is in motion. All the lines can be referred to a common coordinate frame using data from inertial sensors. However, errors from noisy inertial measurements and inaccuracies in the extrinsic parameters between the scanner and the robot frame are also projected onto the shared frame. This causes a deformation in the final scan containing all the lines, which is known as *motion distortion*. Rigid point cloud registration with methods like ICP is therefore not well suited for such distorted scans. In this paper we present a non-rigid registration method that finds the rigid transformation to be applied to *each line* in the scan in order to match an existing model. We fully leverage the continuous and relatively smooth robot motion with respect to the scanning time to formulate our method reducing the computational complexity while improving accuracy. We use synthetic and real data to benchmark our method against a state-of-the-art non-rigid registration method. Finally, the source code for the algorithm is made publicly available.[1]**

***Index Terms*—Autonomous vehicle navigation, non-rigid registration, point set registration, range sensing.**

I. Introduction

POINT set registration is typically used to process range data coming from sensors mounted on robotic platforms, such as laser line scanners (LLSs) or lidars. These sensors do not acquire all the 3D points at once, but rather line by line. All the lines can be referred to a common coordinate frame using data from inertial sensors. However, errors from noisy inertial measurements and inaccuracies in the extrinsic calibration are also projected onto the shared frame. This causes a deformation in the final scan containing all the lines, which is known as *motion distortion*. This distortion varies smoothly through the scan lines: each scan line tends to have suffered a similar deformation to its neighbouring lines (see Fig. 1(c)). Motion distortion is especially relevant for sensors with low refresh frequency or for robots with relatively high dynamics. In our case, we use an underwater LLS [1], [2] with a typical scan density of 50 lines per scan running at 1 Hz (50 lines per second). Since it only takes around 20 ms to acquire all the points in each line, we can safely assume in-line rigidity.



(a) Experimental setup. (b) Model. (c) Original distorted scan. (d) Registered scan.

Fig. 1. Linewise non-rigid registration of real scans. The experiments (a) were carried out with our underwater LLS [1], [2] mounted on the AUV Girona1000 [3]. The 3D CAD model of the structure (b) was used as reference model. Errors in navigation data result in scans affected by motion distortion (c). Our method successfully finds the set of transformations to be applied to the scan in order to fit the model (d).

In order to undistort the scan, we can model the spatial transformation $\mathcal{T}$ that needs to be applied to the scan point cloud $\boldsymbol{Y}$ as a set of rigid transformations, one for each scan line: $\mathcal{T} = \{\mathcal{T}_1 \dots \mathcal{T}_L\}$ for a scan made up of L lines. Thus, a 3D point $\boldsymbol{y} \in \mathbb{R}^3$ in the l-th line of $\boldsymbol{Y}$ will be transformed by the rigid transform $\mathcal{T}_l \in \mathbb{SE}(3)$, with rotation $\boldsymbol{R}_l \in \mathbb{SO}(3)$ and translation $\boldsymbol{t}_l \in \mathbb{R}^3$, as $\mathcal{T}_l(\boldsymbol{y}) = \boldsymbol{R}_l\boldsymbol{y} + \boldsymbol{t}_l$.

In this letter, we present a novel non-rigid registration algorithm that fully leverages a priori knowledge on range sensors to allow smooth inter-line deformation while keeping in-line rigidity. This notably limits the space of feasible solutions and results in improved accuracy and reduced computational complexity when compared to a state-of-the-art non-rigid registration method. Furthermore, we offer an open-source implementation of the method.

Manuscript received February 23, 2022; accepted May 27, 2022. Date of publication June 3, 2022; date of current version June 13, 2022. This letter was recommended for publication by Associate Editor A. Nuechter and Editor S. Behnke upon evaluation of the reviewers' comments. This work was supported in part by the European Commission through ATLANTIS Project under Grant H2020-ICT-2019-2-871571, in part by the Spanish Ministry of Science through Projects PER2IAUV and OPTIHROV under Grants PID2020-115332RB-C32 and PDC2021-120791-C21, in part by the Doctoral Grant of the University of Girona under Grant IFUdG2019, and in part by the mobility Grant IFMobUdG2019. *(Corresponding author: Miguel Castillón.)*

Miguel Castillón and Pere Ridao are with the Computer Vision and Robotics Research Institute (VICOROB), University of Girona, 17003 Girona, Spain (e-mail: miguel.castillon@udg.edu; pere@eia.udg.edu).

Roland Siegwart and César Cadena are with the Autonomous Systems Lab (ASL), ETH Zurich, 8092 Zurich, Switzerland (e-mail: rsiegwart@ethz.ch; cesarc@ethz.ch).

Digital Object Identifier 10.1109/LRA.2022.3180038

[1]The source code of the algorithm can be found in https://github.com/miguelcastillon/lnrr

Contextualization: A great effort has been done in mobile robotics in the last decade to obtain accurate and consistent 3D data from range sensors mounted on moving platforms [4]–[6]. These works typically parameterize the robot trajectory using a continuous representation, such as B-splines, rather than discrete poses, which they aim to optimize by fitting observations from the range sensor. This continuous-time SLAM formulation presents the advantage of allowing an easy fusion of inertial and range sensors, even if they are not perfectly synchronized, while minimizing the motion distortion present in the scans.

However, these methods are based on the implicit assumption that there is enough overlap between consecutive scans. This assumption is met by terrestrial lidars but not by underwater LLSs. State-of-the-art underwater LLSs typically have a maximum range of around 3 to 5 meters and a field of view (FoV) of around $40^\circ \times 40^\circ$ in realistic visibility conditions, due to the high attenuation rate of visible light when travelling through water [7]. Furthermore, autonomous underwater vehicles (AUVs) need to keep a safe distance of at least 1 m during real operations in presence of water currents to avoid crashing into the scanned structure. As a consequence of these factors, it is very common that a relatively high percentage of scans fail at capturing enough informative points and can effectively be considered as empty point clouds. Nonetheless, in the context of missions performed by AUVs such as inspection of an industrial underwater structure, the CAD model of the target to be scanned is usually available. For all these reasons, our method considers the non-rigid registration of one scan at a time against the point cloud sampled from the model of the structure. The result of our method is an undistorted point cloud, which may later be used in downstream applications, such as structural damage assessment. Please note that this approach is motivated by our application but its applicability is not limited to the underwater inspection case, since very few assumptions have been made during the design of the algorithm.

II. Related Work

Many point set registration approaches have been proposed in the literature. They can be classified according to how they i) find correspondences between point clouds and ii) model the transformation $\mathcal{T}$. This section provides a brief overview of point set registration algorithms that are closely related to ours: more precisely, only feature-less methods are considered.

Regarding the choice of correspondences, iterative closest point (ICP) [8], [9] follows the very simple yet effective approach of assigning the closest point at each iteration. This is known as *hard-assignment*. Thanks to its simplicity and low computational complexity, ICP is probably the most popular registration method. One of the factors that severely limits the performance of ICP is the existence of noise in the data points. Probabilistic methods overcome this limitation by *soft-assigning* a correspondence probability between each pair of points. The first method to use soft assignment in point set registration was robust point matching (RPM) [10]. The alternating soft-assignment of correspondences and transformation in RPM is an equivalent to the expectation maximization (EM) algorithm [11] for a Gaussian mixture model (GMM), if we consider one of the sets as GMM centroids and the other as data points [12]. As a matter of fact, point set registration is modelled as a GMM likelihood maximization problem in several methods [13]–[15]. These methods normally add an extra distribution to the GMM in order to account for outliers.

Regarding the choice of the transformation $\mathcal{T}$, registration methods are typically classified into rigid and non-rigid. Rigid methods (like ICP) assume that a rigid transformation is enough to relate both point sets, whereas non-rigid methods allow deformation. Non-rigid registration is naturally more challenging due to the increased number of degrees of freedoms (DoFs). Therefore, it is common for non-rigid techniques to include a regularization term in their cost functions to avoid overfitting. Non-rigid methods usually parameterize the transformation with local displacement fields, which typically use either thin plate spline (TPS) [16] or Gaussian kernels. An example of the former is [13], which combined them with RPM resulting in TPS-RPM. Another example is the correlation-based approach proposed in [17] and extended in [14], which models both point sets as GMMs and estimates the TPS parameters by minimizing the L_2 norm between the distributions. One of the arguably most popular non-rigid methods, coherent point drift (CPD) [18], uses a Gaussian kernel to define the radius of the local displacement field. CPD is related to motion coherence theory [19], [20], which imposes the assumption that neighbouring points tend to move similarly. Among its strengths, its Gaussian kernel provides a free parameter to control the locality of deformations. Moreover, the algorithm is designed to estimate the Gaussian width within the minimization framework instead of using deterministic annealing, which results in reportedly shorter runtimes and better performance [18]. When compared to the correlation-based method in [14], CPD effectively minimizes the KL divergence between two distributions rather than the L_2 norm, which yields better results because it weighs the error according to its probability [18]. Several variants of CPD have been presented which focus on improving different aspects of the algorithm. For instance, [21] improves registration quality and robustness in applications where correspondence priors are available by integrating them in closed form, [22] uses efficient Gaussian filtering methods to achieve substantially faster computational performance while maintaining robustness, and [23], [24] propose a Bayesian approach that accelerates the registration process and guarantees convergence.

Our method draws inspiration from CPD in modelling the problem as a GMM and in parameterizing the regularization term as a local displacement field ruled by a Gaussian kernel. However, it differs from it in the parameterization of the transformations $\mathcal{T}$: we exploit the a priori knowledge of the working principles of LLSs to reduce the dimensionality of the non-rigid displacement field by imposing in-line rigidity.

Within the family of non-rigid methods that characterize the deformation as a displacement field, CPD is the most common method to compare against. Like ICP for rigid methods, CPD is a simple and general method that can be set as a fair baseline for comparison. Moreover, CPD has an open-source implementation that can be readily used. For these reasons, we selected it as the non-rigid method against which we benchmark our algorithm in Section IV.

III. Method

The goal of our method is to register two 3D point sets. The model $\boldsymbol{X}$ is composed of N points, $\boldsymbol{X} = [\boldsymbol{x}_1 \dots \boldsymbol{x}_n \dots \boldsymbol{x}_N]^T$, and the deformable point set $\boldsymbol{Y}$ contains M points, $\boldsymbol{Y} = [\boldsymbol{y}_1 \dots \boldsymbol{y}_m \dots \boldsymbol{y}_M]^T$. In fact, $\boldsymbol{Y}$ is assumed to be made up of L lines, where line l has P_l points, with $\sum_l P_l = M$. The variables

TABLE I
VARIABLES IN THE PROPOSED METHOD

Name	Meaning	Size
$\boldsymbol{G}$	Gaussian kernel	$L \times L$
$\boldsymbol{g}_l$	l-th row in $\boldsymbol{G}$	$1 \times L$
$\boldsymbol{J}_R^T$	Jacobian of the rotation matrix	$3L \times 9L$
L	Number of lines in the scan $\boldsymbol{Y}$	1×1
M	Number of points in the scan $\boldsymbol{Y}$	1×1
N	Number of points in the model $\boldsymbol{X}$	1×1
P_l	Number of points in line l of the scan $\boldsymbol{Y}$	1×1
p_{mn}	Correspondence probability between $\boldsymbol{x}_n$ and $\boldsymbol{y}_m$	1×1
$\boldsymbol{P}$	Probability matrix made up of elements p_{mn}	$M \times N$
$\boldsymbol{R}_l$	Rotation applied to line l in the scan $\boldsymbol{Y}$	3×3
$\boldsymbol{R}$	Matrix of rotation matrices $\boldsymbol{R}_l$	$3 \times 3L$
$\boldsymbol{t}_l$	Translation applied to line l in the scan $\boldsymbol{Y}$	1×3
$\boldsymbol{t}$	Matrix of translation vectors $\boldsymbol{t}_l$	$L \times 3$
$\boldsymbol{X}$	Model point set	$N \times 3$
$\boldsymbol{x}_n$	n-th point (row) in the model $\boldsymbol{X}$	1×3
$\boldsymbol{Y}$	Scan point set	$M \times 3$
$\boldsymbol{y}_m$	m-th point (row) in the scan $\boldsymbol{Y}$	1×3
β	Gaussian width	1×1
λ	Regularization parameter	1×1
w	Weight of the outlier distribution	1×1

TABLE II
SIZES OF THE AUXILIARY MATRICES

Name	Size	Name	Size	Name	Size
$\boldsymbol{A}$	$L \times 3$	$\boldsymbol{F}$	$M \times L$	$\boldsymbol{W}$	$L \times 6$
$\boldsymbol{B}$	$L \times 3L$	$\boldsymbol{S}$	$3L \times 3L$	$\boldsymbol{T}$	$3 \times 3L$
$\boldsymbol{C}$	$M \times 3$	$\boldsymbol{U}$	$L \times 3$	$\boldsymbol{Y}_D$	$M \times 3L$
$\boldsymbol{D}$	$M \times 3L$	$\boldsymbol{V}$	$L \times 3$	$\boldsymbol{Z}$	$L \times L$

defined in this section are gathered in Tables I and II. Please note that $M \gg L$.

The points in $\boldsymbol{Y}$ represent the centroids of a GMM, and the points $\boldsymbol{X}$ represent data points generated by the GMM. In order to account for outliers, we add an extra uniform distribution to the model. The probability density function of the resulting mixture is:

$$p(\boldsymbol{x}_n) = w\frac{1}{N} + (1-w)\sum_{m=1}^{M} P(m)p(\boldsymbol{x}_n|m), \tag{1}$$

with w, $0 \leq w \leq 1$ weighting the outliers contribution. We assign equal weights to all the components in the GMM: $P(m) = \frac{1}{M}$. All the normal distributions in the GMM ($p(\boldsymbol{x}_n|m) = \mathcal{N}(\boldsymbol{x}_n|\mathcal{T}(\boldsymbol{y}_m, \xi), \boldsymbol{\Sigma}_m)$) are assumed to have equal, isotropic variances σ^2:

$$p(\boldsymbol{x}_n|m) = \frac{1}{(2\pi\sigma^2)^{3/2}} \exp\left(-\frac{||\boldsymbol{x}_n - \mathcal{T}(\boldsymbol{y}_m, \xi)||^2}{2\sigma^2}\right), \tag{2}$$

where ξ are the updated centroids of the GMM. Given all the observations $\boldsymbol{X}$, the combined negative log-likelihood to minimize is:

$$E(\xi, \sigma^2) = -\sum_{n=1}^{N} \log \sum_{m=1}^{M} P(m)p(\boldsymbol{x}_n|m) \tag{3}$$

Due to the difficulty in directly minimizing E, we use the expectation maximization (EM) technique instead. This method consists on iterating over two steps. First, in the E-step we fix the parameters (ξ, σ^2) and compute the probability of every possible correspondence. Then, in the M-step we fix these correspondence probabilities and minimize a cost function. In the E-step, we compute the probability that data point $\boldsymbol{x}_n$ was generated by the component m as its posterior probability:

$$p_{mn} = P(m|\boldsymbol{x}_n) = \frac{\exp\left(-\frac{||\boldsymbol{x}_n - \mathcal{T}(\boldsymbol{y}_m, \xi)||^2}{2\sigma^2}\right)}{\sum_{m'=1}^{M} \exp\left(-\frac{||\boldsymbol{x}_n - \mathcal{T}(\boldsymbol{y}_{m'}, \xi)||^2}{2\sigma^2}\right) + c}, \tag{4}$$

where $c = \frac{w}{1-w}\frac{M(2\pi\sigma^2)^{3/2}}{N}$. Then, for the M-step we define the expectation of the complete negative log-likelihood function E [18]:

$$Q(\xi, \sigma^2) = \frac{1}{2\sigma^2}\sum_n \sum_m p_{mn}||\boldsymbol{x}_n - \mathcal{T}(\boldsymbol{y}_m, \xi)||^2 + \frac{3}{2}N_P \log \sigma^2, \tag{5}$$

where $N_P = \sum_n \sum_m p_{mn}$. Jensen's inequality states that Q is an upper bound for E [25], so minimizing Q also minimizes E. Up to this point, the derivation is the same as in [18]. However, our method proposes to define $\mathcal{T}$ as a set of L rigid transformations. We still need to introduce a regularization term in the objective function in order to apply the a priori knowledge that transformations of neighbouring lines tend to be similar. We can define this field as $\boldsymbol{p} : \mathbb{N} \mapsto \mathbb{SE}(3)$, which relates each line index with a 6-DoF rigid transformation.[2] In order to transform centroid $\boldsymbol{y}$, $\boldsymbol{p}$ is first evaluated at the line number of $\boldsymbol{y}$, l:

$$\mathcal{T}(\boldsymbol{y}) = \boldsymbol{p}(l) \oplus \boldsymbol{y}, \tag{6}$$

where the operator $\oplus$ denotes 3D composition [26]. Then, we can introduce the regularization term $\phi(\boldsymbol{p})$ weighted with a regularization parameter $\lambda > 0$:

$$Q(\xi, \sigma^2) = \frac{1}{2\sigma^2}\sum_n \sum_m p_{mn}||\boldsymbol{x}_n - \boldsymbol{p}(l) \oplus \boldsymbol{y}||^2 + \frac{3}{2}N_P \log \sigma^2 + \frac{\lambda}{2}\phi(\boldsymbol{p}) \tag{7}$$

The term $\phi(\boldsymbol{p})$ is introduced to ensure *smoothness* in the field of rigid transformations. In our case, smoothness refers to a measure of the oscillatory nature of the field $\boldsymbol{p}$. In the frequency domain, the field $\boldsymbol{p}$ can be said smooth if it has most of its energy at low frequency (small bandwidth). Therefore, we can define $\phi(\boldsymbol{p})$ as a measure of the remaining energy in $\boldsymbol{p}$ after applying a high-pass filter to it [27]:

$$\phi(\boldsymbol{p}) = \int \frac{|\tilde{\boldsymbol{p}}(\boldsymbol{s})|}{\tilde{G}(\boldsymbol{s})} d\boldsymbol{s}, \tag{8}$$

where the symbol $\tilde{}$ indicates the Fourier transform, and $\tilde{G}$ is some positive function that tends to zero as $||\boldsymbol{s}|| \to \infty$ (so that $\frac{1}{\tilde{G}}$ is a high-pass filter), where $\boldsymbol{s}$ belongs to the complex frequency domain resulting from the Laplace transform. The field $\boldsymbol{p}$ that minimizes this energy in (8) has the form [28]:

$$\boldsymbol{p}(l) = \sum_{k=1}^{L} \boldsymbol{w}_k\, G(l,k) + \psi(l), \tag{9}$$

where $\psi(l)$ is a term in the nullspace of ϕ. From the theory of reproducing kernels [29] it is known that if G is a positive definite function, then (8) is a norm in its corresponding reproducing kernel Hilbert space (RKHS). The key intuition behind this

[2]Please note that this field is different from the field $\boldsymbol{v} : \mathbb{R}^3 \mapsto \mathbb{R}^3$ proposed in [18].

definition is that this squared norm of function ϕ can be thought of as a generalization to functions of the quadratic form of an n-vector [30]. Therefore, its nullspace contains only the zero element: $\psi(l) = 0$. In order to meet this requirement, we choose the radially symmetric Gaussian function:

$$G(l,k) = \exp\left(-\frac{(l-k)^2}{2\beta^2}\right), \tag{10}$$

where the Gaussian width β controls the *locality* of the area in which smoothness is applied. As explained in [18], this choice of kernel agrees with motion coherence theory [19]. There, the authors chose it because it has second order derivatives and it generates analytic solutions.

If we stack $\boldsymbol{w}_k$ and $G(l,k)$ in (9) into matrices $\boldsymbol{W}$ and $\boldsymbol{G}$, respectively, we obtain: $\boldsymbol{p}(l) = \boldsymbol{g}_l\boldsymbol{W}$ and $\phi(\boldsymbol{p}) = ||\boldsymbol{p}||^2_{\mathcal{H}} = \text{tr}(\boldsymbol{W}^T\boldsymbol{G}\boldsymbol{W})$, where $\boldsymbol{g}_l$ is the l-th row of $\boldsymbol{G}$. Substituting in the cost function:

$$\begin{aligned} Q = &\frac{1}{2\sigma^2}\sum_n\sum_m p_{mn}||\boldsymbol{x}_n - \boldsymbol{g}_l\boldsymbol{W} \oplus \boldsymbol{y}_m||^2 \\ &+ \frac{3}{2}N_P\log\sigma^2 + \frac{\lambda}{2}\text{tr}(\boldsymbol{W}^T\boldsymbol{G}\boldsymbol{W}) \end{aligned} \tag{11}$$

Please note that index l changes according to the line in the scan to which the sum index m belongs. Now, our goal is to find the weights $\boldsymbol{W}$ that minimize Q. In order to ease our task, we separate the rotational from the translational components: $\boldsymbol{U}$ and $\boldsymbol{V}$, respectively. $\boldsymbol{U}$ are the first three columns of $\boldsymbol{W}$, and $\boldsymbol{V}$ the last three. We can then rewrite the distance function in (11):

$$\boldsymbol{g}_l\boldsymbol{W} \oplus \boldsymbol{y}_m = \boldsymbol{y}_m\mathbf{Rot}(\boldsymbol{g}_l\boldsymbol{U})^T + \boldsymbol{g}_l\boldsymbol{V} \tag{12}$$

Please note that the rotation matrix is transposed because $\boldsymbol{y}_m$ is a row vector. Rotations are parameterized using the Euler angles roll around x, pitch around y, and yaw around z, in ZYX order. Moreover, we can also use the properties of the trace to rewrite the regularization term in (11), $\text{tr}(\boldsymbol{W}^T\boldsymbol{G}\boldsymbol{W}) = \text{tr}(\boldsymbol{U}^T\boldsymbol{G}\boldsymbol{U}) + \text{tr}(\boldsymbol{V}^T\boldsymbol{G}\boldsymbol{V})$, and we can define $\boldsymbol{t}_l = \boldsymbol{g}_l\boldsymbol{V}$ and $\boldsymbol{R}_l = \mathbf{Rot}(\boldsymbol{g}_l\boldsymbol{U})$, so that the final cost function can be rewritten as:

$$\boxed{\begin{aligned} Q = &\frac{1}{2\sigma^2}\sum_n\sum_m p_{mn}||\boldsymbol{x}_n - (\boldsymbol{y}_m\boldsymbol{R}_l^T + \boldsymbol{t}_l)||^2 \\ &+\frac{3}{2}N_P\log\sigma^2 + \frac{\lambda}{2}\text{tr}(\boldsymbol{U}^T\boldsymbol{G}\boldsymbol{U}) + \frac{\lambda}{2}\text{tr}(\boldsymbol{V}^T\boldsymbol{G}\boldsymbol{V}) \end{aligned}} \tag{13}$$

subject to $\boldsymbol{R}_l^T\boldsymbol{R}_l = I$ and $\det(\boldsymbol{R}_l) = 1,\ \forall\ l \leq L$. Please note that these two constraints are naturally observed in the construction of each rotation matrix $\boldsymbol{R}_l$ out of Euler angles. In the remaining of the section we show how to compute the weights $\boldsymbol{U}$ and $\boldsymbol{V}$ that minimize the cost function in (13).

A. Solving for $\boldsymbol{V}$

First, we solve for $\boldsymbol{V}$ by setting the corresponding partial derivative of the cost function in (13) to zero. The resulting expression is:

$$\begin{aligned} &\left(\sigma^2\lambda\boldsymbol{I}_{L\times L} + \boldsymbol{F}^T\text{diag}\left(\boldsymbol{P}\mathbf{1}_N\right)\boldsymbol{F}\boldsymbol{G}\right)\boldsymbol{V} \\ &\quad = \boldsymbol{F}^T\boldsymbol{P}\boldsymbol{X} - \boldsymbol{F}^T\text{diag}\left(\boldsymbol{P}\mathbf{1}_N\right)\boldsymbol{Y}_D\boldsymbol{R}^T, \end{aligned} \tag{14}$$

where $\boldsymbol{P}$ is the probability matrix made up of elements p_{mn}, $\boldsymbol{I}_{L\times L}$ is the $L \times L$ identity matrix, $\mathbf{1}_N$ is an N-vector of all ones, $\boldsymbol{R} = [\,\boldsymbol{R}_1\ \cdots\ \boldsymbol{R}_l\ \cdots\ \boldsymbol{R}_L\,]$, and

$$\boldsymbol{F} = \left[\text{diag}\left(\mathbf{1}_{P_1},\ldots,\mathbf{1}_{P_L}\right)\right], \tag{15}$$

$$\boldsymbol{Y}_D = \left[\text{diag}\left(\begin{array}{c}\boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_{P_1}\end{array}, \begin{array}{c}\boldsymbol{y}_{P_1+1} \\ \vdots \\ \boldsymbol{y}_{P_1+P_2}\end{array}, \cdots\right)\right]. \tag{16}$$

The optimal $\boldsymbol{V}$ ((14)) can be rewritten as:

$$\boldsymbol{V} = \boldsymbol{A} - \boldsymbol{B}\boldsymbol{R}^T, \tag{17}$$

with $\boldsymbol{A} = \boldsymbol{Z}\boldsymbol{F}^T\boldsymbol{P}\boldsymbol{X}$, $\boldsymbol{B} = \boldsymbol{Z}\ \ \boldsymbol{F}^T\text{diag}(\boldsymbol{P}\mathbf{1}_N)\boldsymbol{Y}_D$ and $\boldsymbol{Z} = (\sigma^2\lambda\boldsymbol{I}_{L\times L} + \boldsymbol{F}^T\text{diag}(\boldsymbol{P}\mathbf{1}_N)\boldsymbol{F}\boldsymbol{G})^{-1}$.

B. Solving for $\boldsymbol{U}$

The cost function ((13)) can be written in matrix form:

$$\begin{aligned} Q = &\frac{1}{2\sigma^2}\left[\text{tr}\left(\boldsymbol{X}^T\text{diag}\left(\boldsymbol{P}^T\mathbf{1}_M\right)\boldsymbol{X}\right) - 2\text{tr}\left(\boldsymbol{X}^T\boldsymbol{P}^T\boldsymbol{Y}_R\right)\right. \\ &\left. + \text{tr}\left((\boldsymbol{Y}_R)^T\text{diag}\left(\boldsymbol{P}\mathbf{1}_N\right)\boldsymbol{Y}_R\right)\right] + \frac{3}{2}N_P\log\sigma^2 \\ &+ \frac{\lambda}{2}\text{tr}(\boldsymbol{U}^T\boldsymbol{G}\boldsymbol{U}) + \frac{\lambda}{2}\text{tr}(\boldsymbol{V}^T\boldsymbol{G}\boldsymbol{V}), \end{aligned} \tag{18}$$

where $\boldsymbol{Y}_R$ is the registered point set: $\boldsymbol{Y}_R = \mathcal{T}(\boldsymbol{Y}) = \boldsymbol{Y}_D\boldsymbol{R}^T + \boldsymbol{F}\boldsymbol{t}$, with $\boldsymbol{t} = [\boldsymbol{t}_1\cdots\boldsymbol{t}_l\cdots\boldsymbol{t}_L]^T$. In fact, $\boldsymbol{Y}_R$ can be expressed as a function of $\boldsymbol{R}$:

$$\boldsymbol{Y}_R = \boldsymbol{C} + \boldsymbol{D}\boldsymbol{R}^T, \tag{19}$$

where $\boldsymbol{C} = \boldsymbol{F}\boldsymbol{G}\boldsymbol{A}$ and $\boldsymbol{D} = \boldsymbol{Y}_D - \boldsymbol{F}\boldsymbol{G}\boldsymbol{B}$. Using (17) and (19) and dropping out the terms that do not depend on $\boldsymbol{U}$, (18) can be rewritten as:

$$Q = \frac{1}{2}\text{tr}\left(\boldsymbol{R}\boldsymbol{S}\boldsymbol{R}^T\right) + \text{tr}\left(\boldsymbol{T}\boldsymbol{R}^T\right) + \frac{\lambda}{2}\text{tr}(\boldsymbol{U}^T\boldsymbol{G}\boldsymbol{U}), \tag{20}$$

where $\boldsymbol{S} = \lambda\boldsymbol{B}^T\boldsymbol{G}\boldsymbol{B} + \frac{1}{\sigma^2}\boldsymbol{D}^T\text{diag}(\boldsymbol{P}\mathbf{1}_N)\boldsymbol{D}$ and $\boldsymbol{T} = -\lambda\boldsymbol{A}^T\boldsymbol{G}\boldsymbol{B} + \frac{1}{\sigma^2}(\text{diag}(\boldsymbol{P}\mathbf{1}_N)\boldsymbol{C} - \boldsymbol{P}\boldsymbol{X})^T\boldsymbol{D}$. Please note that $\boldsymbol{S}$ is symmetric. The optimal $\boldsymbol{U}$ that minimizes the cost function is found by setting the corresponding partial derivative to zero:

$$\boxed{\frac{\partial Q}{\partial \boldsymbol{U}} = \boldsymbol{J}_R^T\,\text{vec}\left(\boldsymbol{R}\boldsymbol{S} + \boldsymbol{T}\right) + \text{vec}(\lambda\boldsymbol{G}\boldsymbol{U}) = \mathbf{0}} \tag{21}$$

The operator $\text{vec}(\cdot)$ flattens the matrix in column-major order. The transpose of the jacobian $\boldsymbol{J}_R$ is a sparse matrix computed as:

$$\boldsymbol{J}_R^T = \left[\begin{array}{c}\text{diag}\left(\text{vec}(\frac{\partial \boldsymbol{R}_1}{\partial\phi}),\ldots,\text{vec}(\frac{\partial \boldsymbol{R}_L}{\partial\phi})\right) \\ \hline \text{diag}\left(\text{vec}(\frac{\partial \boldsymbol{R}_1}{\partial\theta}),\ldots,\text{vec}(\frac{\partial \boldsymbol{R}_L}{\partial\theta})\right) \\ \hline \text{diag}\left(\text{vec}(\frac{\partial \boldsymbol{R}_1}{\partial\psi}),\ldots,\text{vec}(\frac{\partial \boldsymbol{R}_L}{\partial\psi})\right)\end{array}\right] \tag{22}$$

The elements in $\boldsymbol{J}_R$ are the jacobians of $\boldsymbol{R}$ with respect to roll, pitch and yaw, respectively. Finally, (21) is solved numerically.

The optimal σ^2 is found by setting the corresponding partial derivative of (18) equal to 0:

$$\begin{aligned} \sigma^2 = &\frac{1}{3N_P}\left[\text{tr}\left(\boldsymbol{X}^T\text{diag}\left(\boldsymbol{P}^T\mathbf{1}_M\right)\boldsymbol{X}\right) - 2\text{tr}\left(\boldsymbol{X}^T\boldsymbol{P}^T\boldsymbol{Y}_R\right)\right. \\ &\left. + \text{tr}\left((\boldsymbol{Y}_R)^T\text{diag}\left(\boldsymbol{P}\mathbf{1}_N\right)\boldsymbol{Y}_R\right)\right] \end{aligned} \tag{23}$$

Algorithm 1: Linewise Non-Rigid Registration.

Inputs: $\boldsymbol{X}, \boldsymbol{Y}, \lambda > 0, \beta > 0, 0 \leq w \leq 1$
Initialize: $\boldsymbol{Y}_R = \boldsymbol{Y}$, $\sigma^2 = \frac{1}{3NM}\sum_n \sum_m ||\boldsymbol{x}_n - \boldsymbol{y}_m||^2$
Build $\boldsymbol{F}$, $\boldsymbol{Y}_D$, $\boldsymbol{G} : g_{ij} = \exp(-\frac{(i-j)^2}{2\beta^2})$
while not converged **do**
  // E-Step:
  $p_{mn} = \frac{\exp(-\frac{||\boldsymbol{x}_n - \boldsymbol{y}_{Rm}||^2}{2\sigma^2})}{\frac{w}{1-w}\frac{M(2\pi\sigma^2)^{3/2}}{N} + \sum_{m'=1}^{M}\exp(-\frac{||\boldsymbol{x}_n - \boldsymbol{y}_{Rm'}||^2}{2\sigma^2})}$
  // M-Step:
  $\boldsymbol{A} = \boldsymbol{Z}\boldsymbol{F}^T\boldsymbol{P}\boldsymbol{X}, \quad \boldsymbol{B} = \boldsymbol{Z}\,\boldsymbol{F}^T\mathrm{diag}(\boldsymbol{P}\boldsymbol{1}_N)\boldsymbol{Y}_D$
  $\boldsymbol{C} = \boldsymbol{F}\boldsymbol{G}\boldsymbol{A}, \qquad \boldsymbol{D} = \boldsymbol{Y}_D - \boldsymbol{F}\boldsymbol{G}\boldsymbol{B}$
  $\boldsymbol{S} = \lambda\boldsymbol{B}^T\boldsymbol{G}\boldsymbol{B} + \frac{1}{\sigma^2}\boldsymbol{D}^T\mathrm{diag}(\boldsymbol{P}\boldsymbol{1}_N)\boldsymbol{D}$
  $\boldsymbol{T} = -\lambda\boldsymbol{A}^T\boldsymbol{G}\boldsymbol{B} + \frac{1}{\sigma^2}(\mathrm{diag}(\boldsymbol{P}\boldsymbol{1}_N)\boldsymbol{C} - \boldsymbol{P}\boldsymbol{X})^T\boldsymbol{D}$
  Solve $\boldsymbol{U}$ from $\boldsymbol{J}_R^T \cdot \mathrm{vec}(\boldsymbol{R}\boldsymbol{S} + \boldsymbol{T}) + \mathrm{vec}(\lambda\boldsymbol{G}\boldsymbol{U}) = \boldsymbol{0}$
  Build $\boldsymbol{R}$
  $\boldsymbol{Y}_R = \boldsymbol{C} + \boldsymbol{D}\boldsymbol{R}^T$
  $\sigma^2 = \frac{1}{3N_P}[\mathrm{tr}(\boldsymbol{X}^T\mathrm{diag}(\boldsymbol{P}^T\boldsymbol{1}_M)\boldsymbol{X}) - -2\mathrm{tr}(\boldsymbol{X}^T\boldsymbol{P}^T\boldsymbol{Y}_R) + \mathrm{tr}((\boldsymbol{Y}_R)^T\mathrm{diag}(\boldsymbol{P}\boldsymbol{1}_N)\boldsymbol{Y}_R)]$
end while
Return: $\boldsymbol{Y}_R, \mathcal{T}$

The method presented in this section is summarized in Algorithm 1. In comparison with CPD, both methods share the same E-step (building the $\boldsymbol{P}$ matrix). However, the different parameterization of the transformation $\mathcal{T}$ applied to the moving point set $\boldsymbol{Y}$ makes the M-step different. It is worth noting that our method does not optimize all the parameters ($\boldsymbol{U}$, $\boldsymbol{V}$, σ^2) at once but rather iteratively by means of partial derivatives. Such iterations decrease the Q function but not to an exact minimum, which is known as the generalized EM algorithm [11].

We followed the proposal in [18] to alleviate the computational burden of the E-step by using fast Gauss transform (FGT) [31][3] to evaluate the Gaussian kernel and to compute the matrix-vector products $\boldsymbol{P}\boldsymbol{X}$, $\boldsymbol{P}\boldsymbol{1}_N$, and $\boldsymbol{P}^T\boldsymbol{1}_M$.

IV. Results

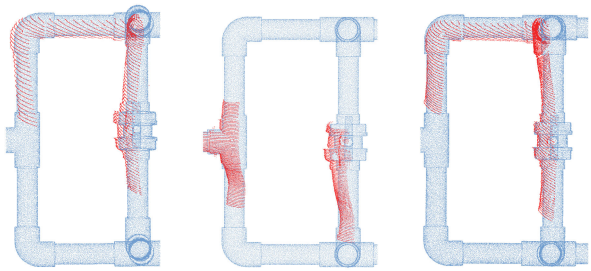
This section presents the registration results of applying our method to synthetic and real scans (Section IV-A and IV-B, respectively). The tests were conducted using an Intel i7-9750H CPU at 2.60 GHz with 16 GB of RAM. The C++ implementation exploited parallelization on the 12 cores whenever possible. Results of the proposed method were benchmarked against a rigid and a non-rigid method. The chosen rigid method was generalized ICP (G-ICP) [32][4]. The chosen non-rigid method was CPD[5]. Different performance aspects were quantitatively evaluated, such as registration accuracy, robustness to noise, robustness to different types and amplitudes of deformations, and runtimes. Throughout the experiments, all methods were given identical, reasonably good initial guesses. In all the boxplots presented in this section, the length of the whiskers represent the 1.5 interquartile range of the data.

As already explained in Section I, our method has been developed in the context of mobile underwater scanning. AUVs

[3]Implementation from https://github.com/gadomski/fgt

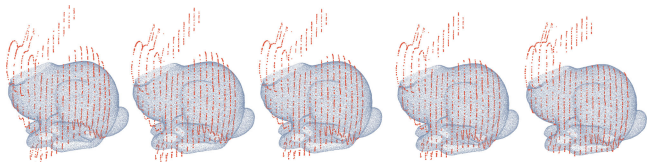
[4]Implemented in PCL https://pointclouds.org/documentation/classpcl_1_1_generalized_iterative_closest_point.html

[5]Implementation from https://github.com/gadomski/cpd



(a) (b) (c)

Fig. 2. Three different examples of real distorted scans (before registration). Each scan only covers a rather small area of the structure due to the limited FoV of the LLS. The distortions present a smooth distribution.



(a) Initial (b) Iter. 5 (c) Iter. 8 (d) Iter. 10 (e) Iter. 30

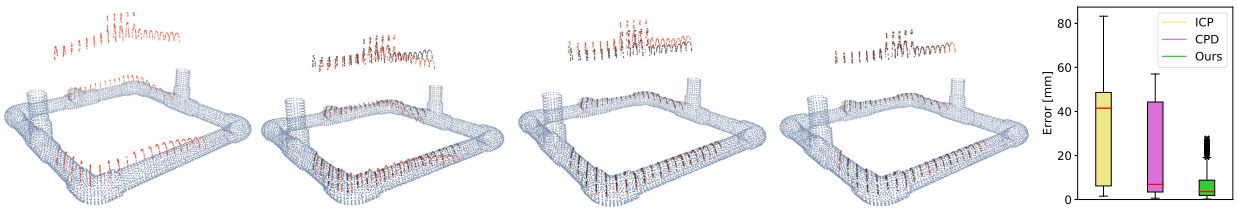
Fig. 3. Registration process of simulated vertical scan lines (red) with an incomplete version of the Stanford bunny model (blue) using the proposed method.

are usually equipped with modern inertial navigation systems (INSs), which are more accurate than inertial sensors mounted on terrestrial robots, especially for measuring orientation. For example, the INS mounted on Girona1000 has orientation errors smaller than 0.10° around all three axes[6] thanks to its high-end fiber optic gyroscopes (FOGs). By examining the experimental scans (such as Fig. 2) and the navigation data, it can be concluded that the translational component of the distortion is predominant over the rotational one. The synthetic dataset used in Section IV-A was designed to be consistent with the experimental dataset used in Section IV-B but with higher deformations: we applied errors of tens of millimeters in translation and of up to 2° in rotation (which is 20x higher than expected in reality). The applicability of our method to realistic scenarios is proved by the low registration errors in Section IV-B (for a visual example, see the registration process from Fig. 1(c) to Fig. 1(d)). Please note that the fact that we test our method on datasets relevant to our use case does not shrink its applicability, since no assumption was made in the design of the algorithm other than in-line rigidity.

A. Synthetic Data

The proposed method was first validated using simulated data from two different datasets: the Stanford bunny and an underwater industrial site. The Stanford bunny was used to visually verify the performance of the proposed method on a well-known model. We simulated line scans and applied different, smoothly-changing rigid transformations to each line (see Fig. 3(a)), following a sinusoidal function. As shown through the iterations in the figure, the proposed method achieved good results (see Fig. 3(e)). Despite the upper part

[6]https://www.ixblue.com/wp-content/uploads/2022/01/Phins%20Compact%20Series%20-%20Datasheet.pdf



(a) Initial (b) ICP. (c) CPD. (d) Ours. (e) Errors

Fig. 4. Challenging example of a synthetically generated scan registered against an incomplete model of the underwater pipe structure. The upper pipe and valve are missing in the model but present in the scan. Figures (b)–(d) compare visually the registration results of ICP, CPD and our method. Black dots in these figures show the ground truth.

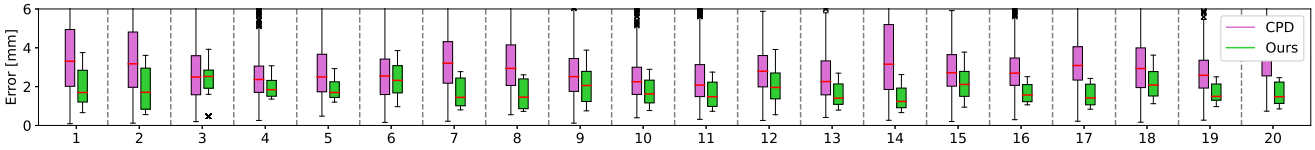


Fig. 5. Synthetic benchmark based on 20 different simulated scans of the underwater pipe structure.

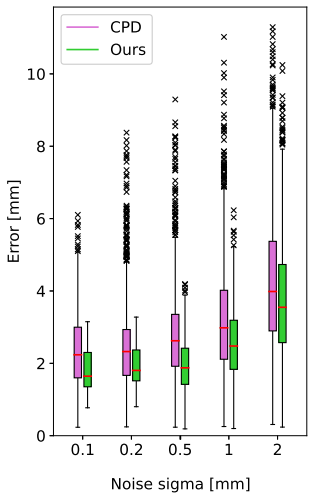


Fig. 6. Accuracy comparison for increasing levels of noise in the scan.

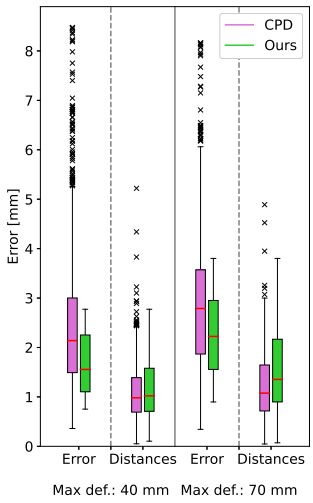


Fig. 7. Accuracy comparison for different deformation levels using two different metrics.

of the model being missing, the in-line rigidity enforced by our method pulls the points without correspondences towards their true position.

The presented method was also synthetically validated using a mock-up model of an underwater industrial site (see Fig. 1(b)). The dimensions of the whole structure are approximately $3500 \times 2100 \times 1300$ mm and it is made up of different types of pipes and valves. The diameter of the pipes is of 60 mm. The points for $\boldsymbol{X}$ were sampled from the model using a voxel grid of 5 mm. This scenario was designed to be more realistic and therefore challenging than the Stanford bunny. First, as it can be seen in Fig. 4, the scan does not contain points that would be occluded from a given point of view of the scanner: in this case, the scanner is assumed to view the structure from above. Second, points in the scan and in the model are subsampled differently and therefore there are no direct correspondences. Unless otherwise stated, the metric used in all the boxplots of synthetic data is the 3D distance of each point to its ground truth after registration. We use this metric because the main goal of our work is to minimize the distortion present in the scans. The red lines in the boxplots refer to the median.

A visual example of how the three different methods compare in this dataset can be found in Fig. 4. Note that in this case only an incomplete subset of the model was considered in order to test the robustness of the method: the upper pipe and its valve are present in the scan but missing in the model. A different rigid transformation was applied to each scan line. The distribution of these transformations followed a sinusoidal pattern, with a maximum distortions of 40 mm in translation and 2° in rotation. The boxplot in Fig. 4(e) shows how the proposed method achieves a more accurate registration than ICP and CPD. ICP naturally fails at registering the distorted scan because it incorrectly assumes that all the points in it can be treated as a rigid body (see Fig. 4(b)). CPD is able to accurately register the points that are close to the visible parts of the structure. However, it fails at registering the points corresponding to the missing upper pipe and valve (see Fig. 4(c)). The reason for it is that coherence in CPD applies to 3D distances between points,

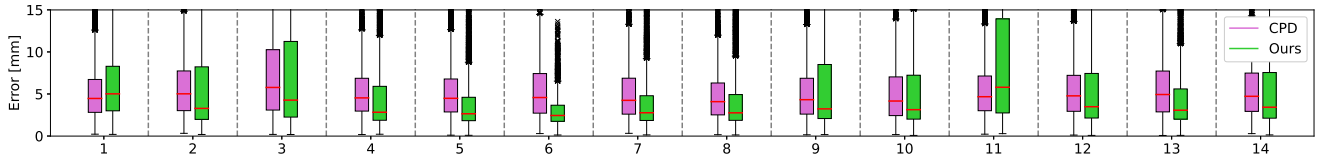
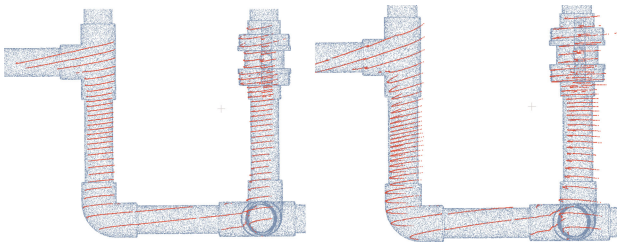


Fig. 8. Experimental benchmark based on 14 different scans of the underwater pipe structure.



(a) Our method. (b) CPD.

Fig. 9. Comparison of a registration result of a real scan using both methods (number 4 in Fig. 8). In this case, both methods achieve similar errors when using the distance-to-model. However, our method maintains the structure of the scan much better than CPD thanks to the in-line rigidity constraint.

TABLE III
RUNTIME. FASTEST METHOD IS MARKED IN BOLD

L	P	N	**CPD [s]**	**Ours [s]**
20	50	5.6k	1.44	**0.7**
20	100	5.6k	4.02	**1.2**
20	150	5.6k	10.7	**1.8**
20	200	5.6k	30.7	**2.0**
20	250	5.6k	57.6	**2.5**
20	200	12.6k	29.6	**2.3**
30	200	12.6k	93.7	**6.1**
40	200	12.6k	232.4	**20.6**

irrespective of which line they correspond to. On the other hand, our method exploits the fact that all the points in each line share the same rigid transform, even if they are distant in 3D space. This makes that the final registration result is closer to the ground truth (see Fig. 4(e)).

1) Benchmark Ideal Conditions: Our method was benchmarked against CPD on 20 simulated scans of the underwater pipe structure (see Fig. 5). ICP was no longer considered because it consistently achieved much worse results than the non-rigid methods (see Fig. 4(e)). All the scans were noise-free and made up of 20 lines. The structure did not have any missing parts. The different scans featured different deformation shapes and directions (always smooth), and different number of points per line. The maximum deformation level was 40 mm in translation and 2° in rotation, following a sinusoidal pattern. It can be seen in Fig. 5 that the proposed method systematically outperforms CPD in this diverse set of scans. The parameter values chosen for our method were $(\beta, \lambda, w) = (60, 80, 0.1)$, and equivalent values for CPD. These values were chosen empirically on a subset of the simulated data and it was observed that slight variations of up to around 40% did not have a very noticeable impact on the results. However, an automatic procedure to estimate these parameters should be further studied, as proposed in Section V.

2) Robustness to Noise: The performance of our method in presence of noise was compared to CPD in Fig. 6. The noise was only applied to the 3D position of the points in scan and it followed a normal distribution centered around 0 with increasing values of standard deviation, as shown in the x axis. Our method consistently yielded lower errors than CPD for all levels of noise.

3) Deformation Amplitude: The performance of both methods on relatively large deformations was studied in Fig. 7. Considering for now only the columns with the label "Error" in the figure, it can be seen that our method can successfully register scans with deformations of 70 mm achieving errors in the order of a few mm (please note that the diameter of the pipes is 60 mm). Our method once again outperforms CPD.

4) Metric Comparison: Until now we have only worked with synthetic data. The registration metric that we have used was the final error with respect to the ground truth. However, this ground truth is not available when working with experimental data. One of the most commonly used metrics in that case is the distance to the closest point in the model. This metric, however, tends to underestimate the registration error because it assigns low errors to points converging towards *any* point in the model, even if far from their ground truth. This effect is depicted in Fig. 7. For both levels of deformation, our method outperforms CPD when we consider the error with respect to the ground truth, as explained in Section IV-A3. If we use the distance-to-model metric instead, both methods are assigned lower errors, which can lead to false interpretations of the results.

5) Runtimes: A comparison of runtimes for different sizes of the scan and of the model can be found in Table III. It can be seen how an increasing number of points per line affects CPD much more than it does our method: a 5x increment in the number of points per line (from 50 to 250) results in a 40x increment in CPD runtime but only 3.5x increment for our method. Increasing the number of points in the model has little effect in both of them, mainly thanks to the use of FGT in the E-step of both. Finally, increasing the number of lines in the scan affects both CPD and our method: doubling the number of lines (from 20 to 40) results in a similar increment in runtime for both methods.

B. Real Data

The proposed method was also validated on real data. The data was gathered in a set of experiments carried out in the water tank at the CIRS lab using our recently developed underwater scanner [2] mounted on Girona1000 [3] (see Fig. 1(a)). The goal of the experiments was to navigate around the structure in the figure while acquiring 3D data. Each scan line was projected to the world reference frame using data from the INS in the robot.

Out of these experiments, a set of 14 scans was established to benchmark our method against CPD (see Fig. 8). The metric

used in this figure is the distance to the closest point in the model. The magnitude of the errors of both methods is larger than in the synthetic dataset, mainly due to experimental errors such as small geometric discrepancies between the model and the real structure. In any case, our method consistently outperforms CPD, achieving final reconstruction errors in the order of a few mm. The parameter values chosen for our method were $(\beta, \lambda, w) = (40, 100, 0.1)$, and equivalent values for CPD. These values were chosen empirically on a subset of the simulated data. A more robust method to determine these values according to the actual deformation of the scans should be further investigated (see experiments 1 and 11 in Fig. 8).

In order to better understand these results, we present a visual comparison of both registration methods in Fig. 9. We see that both methods can fit the model relatively well. However, our method successfully maintains the global topology of the scan by enforcing rigidity of points belonging to the same line. On the other hand, CPD achieves low distance to model but fails at maintaining straight scan lines (please bear in mind the effect of the distance metric, as explained in Section IV-A4).

V. Conclusion

In this work we propose a novel non-rigid registration algorithm that fully leverages on knowledge of the working principles of 3D scanners used by robots. We proved using synthetic and experimental data that our method is able to outperform state-of-the-art non-rigid registration methods, achieving higher accuracy with a lower computational complexity. Moreover, we have made the source code publicly available. Future work in this line of research may involve making our method more robust to relatively bad initial guesses, and including a way of automatically estimating the needed parameters. Finally, the results of applying this algorithm in real robotic scenarios can be used as observations in a SLAM framework or to refine the extrinsic calibration parameters of the scanner.

References

[1] M. Castillón, A. Palomer, J. Forest, and P. Ridao, "Underwater 3D scanner model using a biaxial MEMS mirror," *IEEE Access*, vol. 9, pp. 50231–50243, 2021.

[2] M. Castillón, J. Forest, and P. Ridao, "Underwater 3D scanner to counteract refraction: Calibration and experimental results," *IEEE/ASME Trans. Mechatronics*, 2022, doi: 10.1109/TMECH.2022.3170504.

[3] D. Ribas, N. Palomeras, P. Ridao, M. Carreras, and A. Mallios, "Girona 500 AUV: From survey to intervention," *IEEE/ASME Trans. Mechatronics*, vol. 17, no. 1, pp. 46–53, Feb. 2012.

[4] H. Alismail, L. D. Baker, and B. Browning, "Continuous trajectory estimation for 3D SLAM from actuated lidar," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 6096–6101.

[5] D. Droeschel and S. Behnke, "Efficient continuous-time SLAM for 3D lidar-based online mapping," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 5000–5007.

[6] K. Daun, M. Schnaubelt, S. Kohlbrecher, and O. Von Stryk, "HectorGrapher: Continuous-time lidar SLAM with multi-resolution signed distance function registration for challenging terrain," in *Proc. IEEE Int. Symp. Safety, Secur., Rescue Robot.*, 2021, pp. 152–159.

[7] M. Castillón, A. Palomer, J. Forest, and P. Ridao, "State of the art of underwater active optical 3D scanners," *Sensors*, vol. 19, no. 23, 2019, Art. no. 5161.

[8] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," in *Sensor Fusion IV: Control Paradigms and Data Structures*, vol. 1611. Bellingham, WA, USA: SPIE, Apr. 1992, pp. 586–606.

[9] Z. Zhang, "Point matching for registration of free-form surfaces (Lecture Notes in Computer Science)," in *Comput. Anal. of Images and Patterns*, vol. 719. Berlin, Germany: Springer, 1993, no. 2, pp. 460–467.

[10] S. Gold, A. Rangarajan, C. P. Lu, S. Pappu, and E. Mjolsness, "New algorithms for 2D and 3D point matching: Pose estimation and correspondence," *Pattern Recognit.*, vol. 31, no. 8, pp. 1019–1031, 1998.

[11] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Society: Ser. B*, vol. 39, no. 1, pp. 1–22, 1977.

[12] H. Chui and A. Rangarajan, "Feature registration framework using mixture models," in *Proc. Workshop Math. Methods Biomed. Image Anal.*, 2000, pp. 190–197.

[13] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Comput. Vis. Image Understanding*, vol. 89, no. 2/3, pp. 114–141, 2003.

[14] B. Jian and B. C. Vemuri, "Robust point set registration using Gaussian mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1633–1645, Aug. 2011.

[15] J. Ma, J. Wu, J. Zhao, J. Jiang, H. Zhou, and Q. Z. Sheng, "Nonrigid point set registration with robust transformation learning under manifold regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 12, pp. 3584–3597, Dec. 2019.

[16] F. L. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 6, pp. 567–585, Jun. 1989.

[17] Y. Tsin and T. Kanade, "A correlation-based approach to robust point set registration," in *Proc. Eur. Conf. Comput. Vis.*, vol. 3023, 2004, pp. 558–569.

[18] A. Myronenko and X. Song, "Point set registration: Coherent point drifts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2262–2275, Dec. 2010.

[19] A. L. Yuille and N. M. Grzywacz, "Motion coherence theory," in *Proc. 2nd Int. Conf. Comput. Vis.*, 1988, pp. 344–353.

[20] A. L. Yuille and N. M. Grzywacz, "A mathematical analysis of the motion coherence theory," *Int. J. Comput. Vis.*, vol. 3, no. 2, pp. 155–175, 1989.

[21] V. Golyanik, B. Taetz, G. Reis, and D. Stricker, "Extended coherent point drift algorithm with correspondence priors and optimal subsampling," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2016, pp. 1–9.

[22] W. Gao and R. Tedrake, "Filterreg: Robust and efficient probabilistic point-set registration using gaussian filter and twist parameterization," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019, 2019, pp. 11087–11096.

[23] O. Hirose, "A Bayesian formulation of coherent point drift," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 7, pp. 2269–2286, Jul. 2021.

[24] O. Hirose, "Acceleration of non-rigid point set registration with downsampling and Gaussian process regression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2858–2865, Aug. 2021.

[25] R. Durrett, *Probability: Theory and Examples*, Cambridge, U.K.: Cambridge Univ. Press 2019.

[26] J. Blanco, "A tutorial on SE (3) transformation parameterizations and on-manifold optimization," Univ. Malaga, Málaga, Spain, Tech. Rep. 3, 2013.

[27] F. Girosi, M. Jones, and T. Poggio, "Regularization theory and neural networks architectures," *Neural Computation*, vol. 7, no. 2, pp. 219–269, Mar. 1995.

[28] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proc. IEEE*, vol. 78, no. 9, pp. 1481–1497, Sep. 1990.

[29] N. Aronszajn, "Theory of reproducing kernels," *Trans. Amer. Math. Soc.*, vol. 68, no. 3, pp. 337–404, 1950.

[30] C. E. Rasmussen and C. Williams, "Gaussian processes for machine learning," in *Adaptive Computation and Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.

[31] L. Greengard and J. Strain, "The fast Gauss transform," *SIAM J. Sci. Stat. Comput.*, vol. 12, no. 1, pp. 79–94, 1991.

[32] A. V. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Proc. Robot.: Sci. Syst.*, vol. 2, 2009, Art. no. 435.