sensors

MDPI

Article

Elevation Angle Estimations of Wide-Beam Acoustic Sonar Measurements for Autonomous Underwater Karst Exploration

Yohan Breux * and Lionel Lapierre

Laboratory of Informatics, Robotics and MicroElectronics (LIRMM) (UMR 5506 CNRS—UM), Université Montpellier, 161 rue Ada, CEDEX 5, 34392 Montpellier, France; lapierre@lirmm.fr

* Correspondence: breux@lirmm.fr

Received: 30 June 2020; Accepted: 17 July 2020; Published: 20 July 2020

Abstract: This paper proposes a solution for merging the measurements from two perpendicular profiling sonars with different beam-widths, in the context of underwater karst (cave) exploration and mapping. This work is a key step towards the development of a full 6D pose SLAM framework adapted to karst aquifer, where potential water turbidity disqualifies vision-based methods, hence relying on acoustic sonar measurements. Those environments have complex geometries which require 3D sensing. Wide-beam sonars are mandatory to cover previously seen surfaces but do not provide 3D measurements as the elevation angles are unknown. The approach proposed in this paper leverages the narrow-beam sonar measurements to estimate local karst surface with Gaussian process regression. The estimated surface is then further exploited to infer scaled-beta distributions of elevation angles from a wide-beam sonar. The pertinence of the method was validated through experiments on simulated environments. As a result, this approach allows one to benefit from the high coverage provided by wide-beam sonars without the drawback of loosing 3D information.

Keywords: karst; underwater; sonar; SLAM; AUV; 3D

1. Introduction

The context of this paper is the development of robotic systems to explore karstic environments. Karsts are networks of underground natural conduits resulting from the dissolution of soluble rocks, limestone, dolomite and gypsum that drain groundwater on large scales. This is illustrated in Figure 1. Karst aquifers are of upmost importance as they are the main sources of drinking water for millions of people worldwide. They are gigantic tanks of renewable fresh water, but their exploitation requires deeper investigation of their characteristics: location, geomorphology and dynamics. Active groundwater management is a major issue for public authorities involved in the prospection, protection and management of the groundwater resource and hydrogeologic risk prevention. In order to have a better understanding of karst aquifers, it is essential to know the geometry of the flow paths, which may be difficult to acquire. Currently, this is done by cave divers with a limited range due to physiological constraints. An autonomous robotic solution would provide the capacity to go further and deeper in the karst conduit while acquiring dense information on the environment.

However, there are still many scientific and technical challenges to overcome before reaching this goal. This paper is related to the robot localization problem in a SLAM framework. Recently, a lot of studies have focused on vision-based localization approaches. Although they can provide accurate and dense measurements, they are subject to the water conditions and the appearance of the environment. Thus, for extensive exploration of an unknown environment, we need to complement it

with robust sensors. For terrestrial robots, one can use a scan matching technique such as ICP (iterative closest point) using LIDAR [1]. However, in underwater robotics, LIDAR requires water transparency, which is not guaranteed in a karstic environment. Then, the solution relies on acoustic sonars. This has two major drawbacks:

- As sonar deals with acoustic waves, the time-of-flight is always higher than for a laser. Moreover, mechanical pencil-beam sonars are slow, and unlike LIDAR, for which full scan are instantaneous, they take several seconds.
- A sonar beam can also have non-negligible beam width which leads to high uncertainty on the 3D positions of the observations.

Our final objective was then to develop a variant of ICP adapted to those constraints. In particular, that requires dealing with the 3D estimation of sonar measures with non-negligible beam width. In our setup, similarly to [2–4], the robot was mounted with two mechanically scanning sonar systems (MSIS). A narrow-beam sonar is mounted vertically while the another wide-beam sonar is mounted horizontally. In [4], the vertical sonar is only used for 3D reconstruction of the environment while the horizontal sonar is used to estimate the 2D robot motion using a probabilistic ICP approach. We introduce in this paper a new approach for merging both sonars' information to estimate the 3D positions of points in the environment measured with the wide-beam sonar. Indeed, the advantage of a wide acoustic beam is to obtain a large coverage of the environment, increasing the surface covered by two successive full 360° scans. Obviously, it comes at the price of having a high uncertainty in the true observed 3D point. The idea was to use the narrow vertical sonar to learn a local probabilistic model of the surface with Gaussian process regression. This model was then leveraged to estimate the probabilistic distribution of each horizontal sonar measurement.

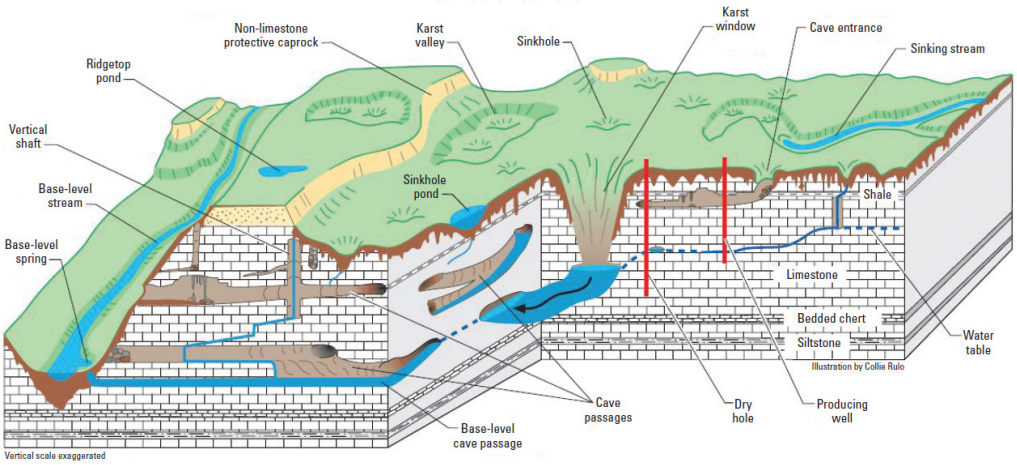


Figure 1. Karstic environment (extracted from [5]).

The first contribution of this paper is to propose a 3D Gaussian process regression adapted to karst environment and more generally applicable to approximately elliptic cylindrical shaped surfaces. We analyzed the results obtained with different kernels to better comprehend their effects on the estimated surfaces. The second contribution is a principled approach to estimating the scaled-beta distribution of elevation angles from a wide-beam sonar by exploiting the previously estimated environment. The pertinence of the method was quantitatively assessed through simulations wherein the estimated distributions could be evaluated against the environment ground truth.

The structure of the paper is as follows. Section 2 gives an overview on related works in underwater robotics. Our robotic system and related definitions used throughout the paper are described in Section 3. Section 4 is the main part of this paper in which we develop our approach. Finally, Section 5 describes evaluations of our work in simulated environments. The conclusion and future working directions wrap up in Section 6.

2. Related Works

In the last few decades, a wide range of simultaneous localization and mapping (SLAM) techniques have been developed [1,6]. In particular, a large focus has been put in the robotic field on terrestrial and aerial applications based on camera sensors [7,8]. Some efforts were also made to use optical sensors for underwater applications [9–11], but were rather limited as they were highly dependent on water conditions (turbidity, luminosity). Hence we cannot guarantee the robot localization in an unknown environment based solely on vision sensors. However, those approaches can be used when applicable to complement more robust sonar-based methods.

Thus, a large part of the underwater SLAM literature is based on acoustic sonars. References [12] and [13] proposed a pose-graph SLAM approach using a dual-frequency identification sonar (DIDSON) for the inspection of underwater structures. The advantage of their approach is to compensate for the low coverage of their sonar by sweeping it in a stationary position. However, this is not applicable in the context of karst exploration where the robot moves while sensing the environment. Closer to our application, Fairfield et al. [14,15] first developed a SLAM approach for cenote (underwater vertical tunnel) exploration. It is based on a Rao-Blackwellized particule filter (RBPF) with a mapping is represented by a 3D evidence grid in an octree structure. They used an array of 54 pencil-beam sonars placed around their vehicle in order to map the environment and recognize previously mapped region for SLAM purposes. Rahman et al. [10] proposed a system for underwater cave exploration. They notably leverage profiling sonar, and inertial and depth sensors to strengthen their approach based on stereo camera vision. However, the system is not designed to work in degraded mode when vision is not available.

Palomer et al. [3] used an extended Kalman filter (EKF) SLAM framework for seafloor mapping. They used a multi-beam echosounder and the resulting scans were matched using probabilistic ICP [16], as proposed by [2] in a similar context. While this is pertinent for bathymetry applications, we cannot fix a multibeam echosounder in a fixed direction for the exploration of a karst aquifer. Similarly, Burguera [4] developed a method for registering data from mechanically scanned imaging sonar (MSIS). Elevation angles from beam measurements and thus 3D data are obtained through the assumption that the robot moves parallel to the seafloor. Obviously we cannot make the same assumption in our context. Chen et al. [17] also recently proposed a Rao-Blackwellized particle filter (RBPF) SLAM framework using MSIS for the same application. They considered a beam enpoint model to compute the likelihood of an observation which is classical in the occupancy grid approach. They focus on the 2D localization of the robot and the lost of 3D information from the wide-beam sonar was ignored.

Unlike previous works which focused either on inspection or bathymetry in open space, we are interested in the exploration and mapping of confined underwater environments (karstic caves). Mallios et al. [18,19] presented experimental results of the exploration of an underwater cave based on their earlier work [2]. They also provided a dataset from this experiment. Thanks to the MSIS vertical beam width and by supposing a quasi-translational displacement between two full 360° scans, they could restrict the estimation to a 2D problem. Depth, pitch and roll absolute values were directly read from the precision and the IMU sensor. Our work was inspired by this approach and aimed at considering the full 3D problem. For instance, this approach cannot deal with L-shaped conduits where measured points cannot be consider coplanar. Furthermore, even in coplanar cases, it is not theoretically sound to separate the rigid transformation estimation to $SE(2)$ and then set the other values (depth, pitch and roll) with the sensor's data. The problem should be cast as a constrained optimization on $SE(3)$. Finally, Martins et al. [20] developed a robotic system called UX-1 under the European project UNEXMIN designed to explore flooded, abandoned mines. It is equipped with a camera, a four-customized-laser-based structured light system and multibeam profiler sonar. They extract natural landmarks from the different sensors which are then integrated into a graph-based SLAM framework. While their context was similar to ours, they mainly relied on the fusion of vision and a structured light system. Note that while we focus solely on sonar sensors in this paper, we ultimately aim at complementing already existing methods.

To the best of our knowledge, karstic exploration by AUV remains rather unexplored territory. The main difficulty is that the karst can be composed with narrow galleries or large caves. In particular, we need MSIS sonar with a wide beam to increase the chance that a previously mapped region will be encountered when revisited. Meanwhile, we also need 3D data from the sonar as we cannot restrict the problem to 2D here. Indeed, in a karstic environment, the robot would have to follow non-planar trajectories. Furthermore, it gives the possibility to further improve the relative displacement estimation by considering a 3D ICP constrained in depth, pitch and roll. This means that we have to estimate the elevation angle from the sonar measurements. It is the same problem expressed recently by Wang et al. [21]. Their work was similar to [3] but improved on it by searching to estimate the sonar elevation angles which were assumed null in the previous works. They did so by directly tracking features in the multibeam sonar images. They also estimated the terrain height by modeling it with Gaussian processes directly integrated in their factor graph SLAM. Note that in their case, the elevation angle was indirectly obtained through the terrain height estimate. In our case, we used MSIS (single beam) sonar. Besides, due to the karstic environment, we had to fully model the environment. The assumptions, such as elevation angle with null mean, are not applicable in our case.

3. Definitions

3.1. Notation

We define here the notation used throughout this paper. Vectors are represented in bold lower case (e.g., $\mathbf{x}$) and matrices in upper case (e.g., M). Integer intervals are denoted $[\![a, b]\!]$. If X is a random variable, $X \sim q(\boldsymbol{\lambda})$ means that X follows a law defined by the probability density function (pdf) q with parameters $\boldsymbol{\lambda}$. With an abuse of notation, we denote as $P(X = x) = P(x)$ the probability of X taking the value x (in the continuous case , $P(X = x)$ is always zero (null set). In this case, $P(X = x) \widehat{=} q(x|\lambda)$ where q is its pdf.). We also denote $\hat{X}$ the estimator of a random variable X.

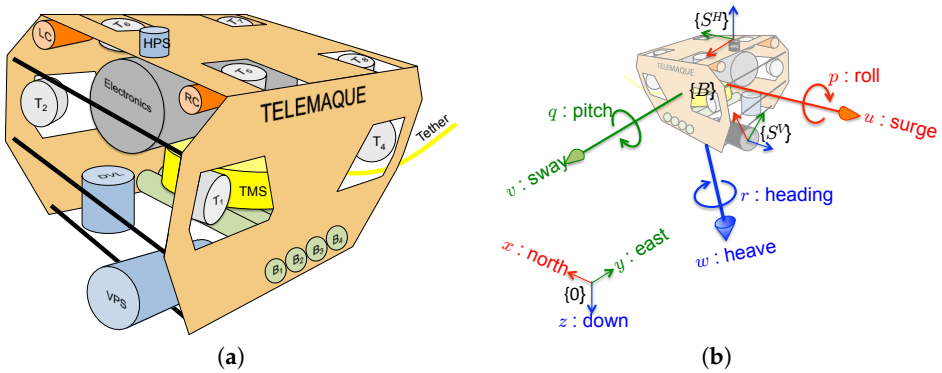
The operator $\oplus$ (resp. $\ominus$) represents the pose-pose and pose-point composition (resp. inverse composition) operator [22]. We use the notation $\mathbf{x}_{|\{R\}}$ to express that a vector $\mathbf{x}$ is expressed in the basis (frame) $\{R\}$.

Let M be a d-manifold embedded in $\mathbb{R}^n$. We denote $T_{\mathbf{x}}M$ the tangent plane (d-dimensional vector space) at $\mathbf{x} \in M$. In particular, if M is a Riemaniann manifold, we denote $< ., . >_{\mathbf{x}}$ the inner product on $T_{\mathbf{x}}M$.

$SE(3)$ is the special Euclidean group representing the rigid transformations of space. It is a Lie group and we denote $\mathfrak{se}(3)$ its Lie algebra. We denote Euler angles $[\psi, \chi, \phi]^T$ in yaw–pitch–roll convention. $R(\mathbf{q})$ (resp. $R(\psi, \chi, \phi)$) is the rotation matrix corresponding to a quaternion $\mathbf{q}$ (resp. the Euler angles $[\psi, \chi, \phi]^T$). When needed, we also use the Matlab notation $\mathbf{x}(i : j)$.

3.2. System Description

In the sequel we will consider the underwater robotic system Telemaque, depicted in Figure 2a. Telemaque is equipped with 8 thrusters, affording the robot with omni-directional capability. A complete sensor suite (9-axis IMU and DVL) allows for computing dead-reckoning navigation. Two profiling sonars horizontally and vertically scan the environment and will be used for SLAM. As a preliminary step, the merging between the 2 sonar measurements is performed. This is the subject of this paper.



(**a**) (**b**)

Figure 2. (**a**) Model of the underwater robot Telemaque, equipped for karstic exploration, and (**b**) frames definition.

According to Figure 2b, we define several frames:

- Let $\{0\}$ be the universal fixed frame;
- Let $\{B\}$ be the body-fixed frame, attached to the robot;
- Let $\{S^H\}$ be the frame attached to the horizontal profiling sonar;
- Let $\{S^V\}$ be the frame attached to the vertical profiling sonar.

Equipped with those frames definitions, let:

- $\boldsymbol{\tau}$ be the (3×1) vector denoting the position of $\{B\}$ origin with respect to $\{0\}$;
- $\mathbf{q}$ be the (4×1) normalized quaternion denoting the attitude of $\{B\}$ with respect to $\{0\}$;
- $\boldsymbol{\eta} = \left[\boldsymbol{\tau}^T \ \mathbf{q}^T\right]^T$ be the (7×1) vector denoting the system pose with respect to $\{0\}$;
- $\mathbf{p}^H = [\boldsymbol{\tau}_H^T \ \mathbf{q}_H^T]^T$ be the (7×1) vector denoting the pose of the horizontal profiling sonar, with respect to $\{B\}$, i.e., position and attitude of $\{S^H\}$ with respect to $\{B\}$;
- $\mathbf{p}^V = [\boldsymbol{\tau}_V^T \ \mathbf{q}_V^T]^T$ be the (7×1) vector denoting the pose of the vertical profiling sonar, with respect to $\{B\}$, i.e., position and attitude of $\{S^V\}$ with respect to $\{B\}$.

The robot is equipped with navigation sensors (IMU and DVL) which allow for performing dead-reckoning navigation and provide, regularly, an estimation of the global system pose $\boldsymbol{\eta}$. It is assumed to follow a normal Gaussian distribution, i.e., $\boldsymbol{\eta} \sim \mathcal{N}(\overline{\boldsymbol{\eta}}, \Sigma_{\boldsymbol{\eta}})$, where $\mathcal{N}(\overline{\boldsymbol{\eta}}, \Sigma_{\boldsymbol{\eta}})$ stands for a Gaussian distribution of mean $\overline{\boldsymbol{\eta}}$ and covariance matrix $\Sigma_{\boldsymbol{\eta}}$. We call discretized trajectory the set of system pose $H = \{\boldsymbol{\eta}_i\}, i \in [\![1, n^r]\!]$ regularly obtained by dead-reckoning along its n^r first poses.

3.3. Sonars and Measurements

The two sonars embed an acoustic rotating head and are mounted on the robot according to Figure 3 with poses in $\{B\}$ denoted $\mathbf{p}^H$ and $\mathbf{p}^V$. We denote Θ^H (resp. Θ^V) the vertical aperture and Ψ^H (resp. Ψ^V) the horizontal aperture for the horizontal (resp. vertical) sonar.

Due to those apertures, a single beam from those sonars can correspond to several echos with different ranges, as illustrated in Figure 4. In our case, according to the specification of the sonar used, we consider $\Psi^H = \Psi^V = \Theta^V = 0$. In other words, only the vertical aperture of the horizontal sonar is taken in account. As a corollary, a single beam from the vertical sonar provides a unique measure $\mathbf{m}_i^v = [\rho_i^v \ 0 \ \psi_i^v]$ expressed as spherical coordinates in its local frame $\{S^V\}$ and acquired from the robot pose $\boldsymbol{\eta}_i$ (in fact, it is slightly different from the classic spherical coordinates in the θ definition. Here it is defined as the oriented angle between the XY-plane and the measured ray. See Figure 3.

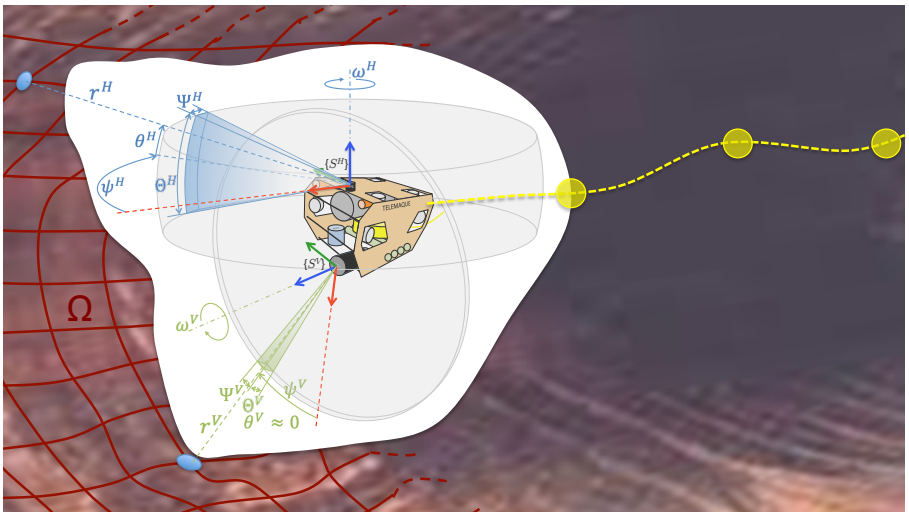


Figure 3. The vertical and horizontal profiling sonars mounted on Telemaque while scanning the environment's surface Ω.

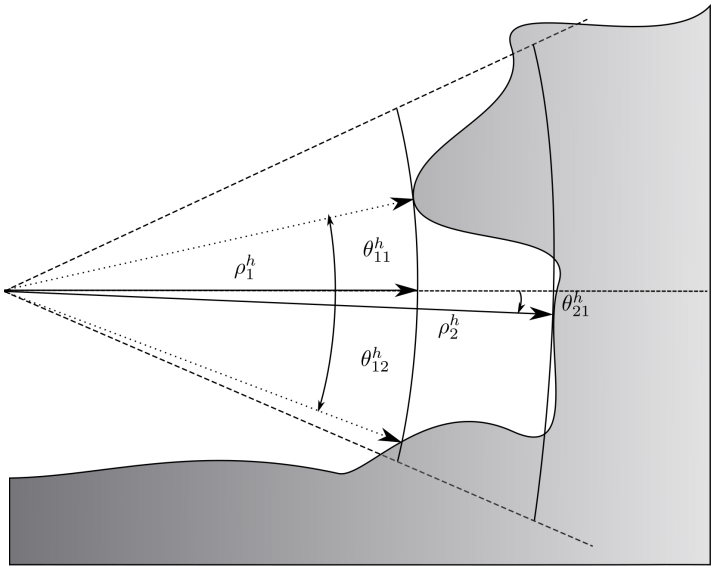


Figure 4. A single measure from the horizontal sonar can correspond to several 3D points and in consequence to several elevation angles.

For the horizontal sonar, due to the vertical aperture, a single insonification can provide n different echos corresponding to n different ranges ρ_j with unknown elevation angles θ_j, $j \in [\![1, n]\!]$. Hence the i-th measurement of the horizontal sonar is represented by a $3 \times n$ matrix $M_i^h = [\mathbf{m}_{ij}^h]$ where each $\mathbf{m}_{ij}^h = [\rho_{ij}^h \; \theta_{ij}^h \; \psi_i^h]^T$ corresponds to an observed point on the environment surface. Note that the angle ψ_i^h corresponding to the sonar head rotation angle is the same for the n points coming from the i-th beam.

Similarly to [4], only the horizontal sonar can be used to estimate the relative poses using a variant of ICP. Thus we define a complete scan $S = \{\mathcal{M}^h, \mathcal{M}^v\}$ when the horizontal scan has completed a full 360° measurement. $\mathcal{M}^h = \left\{M_i^h\right\}$, $i \in [\![1, n^h]\!]$ is the set of measurements from the horizontal sonar during a complete scan. Similarly, we define $\mathcal{M}^v = \left\{\mathbf{m}_k^v\right\}$, $k \in [\![1, n^v]\!]$ the set of measurements from the vertical sonar obtained during the same interval of time.

Note that in [4] the vertical sonar is only used to add points to the reconstructed 3D environment. In this work, the vertical sonar is also used to estimate the elevation angles from the horizontal sonar.

As the dead-reckoning navigation and both sonar measurements are asynchronous, the i-th sonar measurements M_i^h, M_i^v are not acquired simultaneously at the i-th robot pose $\boldsymbol{\eta}_i$. We then define the

following mapping linking the sonar measurements to the robot pose at which they have been acquired. We denote $\sigma^h : [\![1, n^h]\!] \to [\![1, n^r]\!]$ (resp. $\sigma^v : [\![1, n^v]\!] \to [\![1, n^r]\!]$) the function which returns the pose index at which a horizontal (resp. vertical) sonar measurement has been made. This means for example that the i-th horizontal measurement M_i^h has been acquired at the robot pose $\boldsymbol{\eta}_{\sigma^h(i)}$.

We also define the functions g^h and g^v which maps local spherical coordinates in the horizontal (resp. vertical) sonar frame to the Cartesian coordinates in the global frame $\{0\}$:

$$g^h : \begin{cases} \mathbb{R}^+ \times \Theta^H \times [-\pi, \pi] \times [\![1, n^h]\!] & \to \Omega \\ \begin{bmatrix} r \\ \theta \\ \psi \end{bmatrix}, i & \mapsto \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \boldsymbol{\eta}_{\sigma^h(i)} \oplus \mathbf{p}^H \oplus \begin{bmatrix} r\cos(\theta)\cos(\psi) \\ r\cos(\theta)\sin(\psi) \\ r\sin(\theta) \end{bmatrix} \end{cases} \tag{1}$$

$$g^v : \begin{cases} \mathbb{R}^+ \times \Theta^V \times [-\pi, \pi] \times [\![1, n^v]\!] & \to \Omega \\ \begin{bmatrix} r \\ \theta \\ \psi \end{bmatrix}, i & \mapsto \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \boldsymbol{\eta}_{\sigma^v(i)} \oplus \mathbf{p}^V \oplus \begin{bmatrix} r\cos(\theta)\cos(\psi) \\ r\cos(\theta)\sin(\psi) \\ r\sin(\theta) \end{bmatrix} \end{cases} \tag{2}$$

where Ω is the environment surface. All the definitions are illustrated in Figure 2.

3.4. Gaussian Process Regression for Surface Estimation

A Gaussian process (GP) [23] is a set of random variables such that every finite subset of those variables are normally distributed. A GP regression is thus a non-parametric method for function regression. In our case, it is used to model the environment surface through the relation between the vertical sonar output (range ρ^v) and inputs (rotation angle ψ^v and curvilinear abscissa s along the robot's trajectory as explained in the Section 4.1):

$$f(s, \psi^v) + \epsilon = \rho^v \sim \mathcal{N}(\bar{\rho}, \sigma_\rho^2) \tag{3}$$

$$f \sim \mathcal{GP},\ \epsilon \sim \mathcal{N}(0, \sigma_n^2) \tag{4}$$

where ϵ is the observation noise. Let f_p denote the prior function which models our prior knowledge of the environment. The definition of this function in our case will be discussed in Section 4.2.1.

Gaussian processes involve kernel functions which measure the similarity between data points. A kernel is simply defined as a function mapping a pair of inputs $\boldsymbol{x}, \boldsymbol{x'}$ into $\mathbb{R}$. In the case of Gaussian processes, the notion of similarity is defined by the covariance. Hence we are interested in symmetric positive definite kernels. More details on Gaussian processes and kernel properties can be found in [23]. The choice of the kernel has a great incidence on the training result. The radial basis function (RBF) kernel is generally the default kernel used in GP:

$$K^{RBF}(\mathbf{x}, \mathbf{x}') = \sigma_p^2 e^{\left(-\frac{||\mathbf{x}-\mathbf{x}'||^2}{2l^2}\right)} \tag{5}$$

where σ_p^2 is the output length scale and l^2 is the input length scale. It gives smooth results which are appropriate for smooth surface but not for rougher one. This kernel is member of the broader Matern class of covariance functions K_ν^M [24]. Those kernels depend on a positive parameter ν which controls the smoothness of the function. Only values of the form $\nu = p + \frac{1}{2}, p \in \mathbb{N}$ are considered, as it allows one to considerably simplify the general expression of Matern kernel. Besides, values above $\frac{7}{2}$ are

ignored as they give results almost identical to the case $\nu = +\infty$ (see [23], pp. 84–85). In particular, the case $\nu = \frac{1}{2}$ corresponds to the squared exponential kernel

$$K^{exp}(\mathbf{x}, \mathbf{x}') = K^M_{\frac{1}{2}}(\mathbf{x}, \mathbf{x}') = \sigma_p^2 e^{\left(-\frac{||\mathbf{x}-\mathbf{x}'||}{l}\right)}$$

and the case $\nu = +\infty$ gives the RBF as in Equation (5), e.g., $K^{RBF} = K^M_{+\infty}$. A trade-off between those two values is obtained for $\nu = \frac{3}{2}$ as also explained in [25]. The expression of the Matern kernel for $\nu = \frac{3}{2}$ is given by

$$K^M_{\frac{3}{2}}(\mathbf{x}, \mathbf{x}') = \sigma_p^2 \left(1 + \frac{||\mathbf{x} - \mathbf{x}'||\sqrt{3}}{l}\right) e^{\left(-\frac{||\mathbf{x}-\mathbf{x}'||\sqrt{3}}{l}\right)} \tag{6}$$

All the hyperparameters l, σ_p and the observation noise variance σ_n (Equation (4)) can be set manually and/or obtained by maximum likelihood estimation (MLE) on a training data set. Note that products and sums of kernels are also kernels. This allows one to build customized kernels for a specific problem based on basic kernels such as those presented in this section.

The original GP regression problem considers the full set of training input and considers them as noise-free. Some recent work [26–28] proposes extensions for sparse GP regression and/or taking into account input uncertainties. Here, as we are only locally modeling the surface through the vertical sonar measurements, our training data are relatively few; thus, a sparse formulation is not mandatory. However, as our robot accumulates position uncertainties (due to dead-reckoning drift) during the scan acquisition, taking in account input uncertainties could significantly improve our estimation. In a first step, we currently assume noise-free inputs.

4. MSIS Scan Merging

4.1. Environment Model

We model the surface of the environment based on virtual vertical sonar measurements. More precisely, a 3D point on the surface is parameterized by the 6D pose of the robot and the vertical sonar rotation angle at which the point could have been seen by the vertical sonar. Formally, the surface is defined as a function which maps a 6D pose $\boldsymbol{\eta}$ and a scan rotation angle ψ^v to an estimated distance ρ^v:

$$f : \begin{cases} SE(3) \times [-\pi, \pi] & \rightarrow \mathbb{R}^+ \\ (\boldsymbol{\eta},\ \psi^v) & \mapsto \rho^v \end{cases} \tag{7}$$

This means that ρ^v is the distance expressed in the vertical sonar plan (XY-plan in $\{S^V\}$) that would have been acquired for a measurement at the robot pose $\boldsymbol{\eta}$ and with the sonar rotation angle ψ^v. We cannot use here a terrain model of the form $z = f(x, y)$ like in [21,29] as it is a non-functional relation in our karstic context.

As the robot trajectory is a 1D submanifold of $SE(3)$, we can parameterize it with a simple scalar $u \in \mathbb{R}$. We recall that kernels measure similarity between two inputs (cf. Section 3.4). If we consider a kernel $K(.,.)$, we would like to have

$$K(\boldsymbol{\eta}_1, \boldsymbol{\eta}_2) = K(u_1, u_2). \tag{8}$$

The first idea is to use the time t to describe the position of the trajectory manifold, as done in [25]. However, in this case Equation (8) is not verified. Indeed, the kernel length scale l as in (5) in the Gaussian process regression will be depending on time and not on space. The problem can be illustrated by considering an immobile robot at two instants t_1 and t_2 such that $t_2 \gg t_1$ and $\boldsymbol{\eta}(t_1) = \boldsymbol{\eta}(t_2)$. We then have $K(t_1, t_2) \approx 0$ whereas $K(\boldsymbol{\eta}_1, \boldsymbol{\eta}_2) = \sigma_p^2$.

To avoid this, we use the curvilinear abscissa as an input parameter based on the length of the trajectory in $SE(3)$. We use the left-invariant Riemannian metric induced by the following point-independant Riemannian metric on $GA(3)$ (general affine group) ([30], Section 3.2)

$$\begin{aligned} \forall A \in SE(3),\ \forall\, \mathbf{u},\mathbf{v} \in T_A SE(3),\ \langle u,v\rangle_A &= Tr(\mathbf{u}^T\mathbf{v}) \\ &= \langle A^{-1}\mathbf{u}, A^{-1}\mathbf{v}\rangle_e \\ &= \langle \mathbf{s}_u, \mathbf{s}_v\rangle_e \\ &= \mathbf{s}_u^T G \mathbf{s}_v \end{aligned} \tag{9}$$

$$G = \begin{bmatrix} I_3 & 0 \\ 0 & 2I_3 \end{bmatrix} \tag{10}$$

where e is the identity element of $SE(3)$, $T_A SE(3)$ the tangent plane at A and $\mathbf{s}_u, \mathbf{s}_v \in \mathfrak{se}(3)$. The length of a curve $\boldsymbol{\eta}(t) : \mathbb{R} \to SE(3)$ between two instants t_1 and t_2 is then given by

$$L_{\boldsymbol{\eta}(t1)}^{\boldsymbol{\eta}(t2)} = \int_{t_1}^{t_2} \left\langle \frac{d\boldsymbol{\eta}}{dt}, \frac{d\boldsymbol{\eta}}{dt} \right\rangle^{\frac{1}{2}} dt \tag{11}$$

The curvilinear abscissa s is simply given by

$$\forall t,\ s(t) = L_{\boldsymbol{\eta}(0)}^{\boldsymbol{\eta}(t)} \tag{12}$$

In our work, we consider a discrete trajectory regularly updated from dead-reckoning fusion algorithms, fed with DVL and IMU measurements. Note that we only consider the trajectory during one full scan acquisition. The full trajectory estimation in a graph SLAM framework is out of the scope of this paper. Given the discrete trajectory $H = \{\boldsymbol{\eta}_i\}$, we approximate the length as follows.

$$\begin{aligned} \forall i \in [\![2, n^r]\!],\ s_i = L_{\boldsymbol{\eta}_1}^{\boldsymbol{\eta}_i} &= \sum_{k=1}^{i} \langle \Delta\boldsymbol{\eta}_k, \Delta\boldsymbol{\eta}_k\rangle^{\frac{1}{2}} \\ &= \sum_{k=1}^{i} \langle \boldsymbol{\eta}_k \ominus \boldsymbol{\eta}_{k-1}, \boldsymbol{\eta}_k \ominus \boldsymbol{\eta}_{k-1}\rangle^{\frac{1}{2}} \\ &= \sum_{k=1}^{i} \sqrt{log(\boldsymbol{\eta}_k \ominus \boldsymbol{\eta}_{k-1})^T\ G\ log(\boldsymbol{\eta}_k \ominus \boldsymbol{\eta}_{k-1})} \\ &= \sum_{k=1}^{i} \sqrt{log(T_{k-1}^{-1}T_k)^T\ G\ log(T_{k-1}^{-1}T_k)} \end{aligned} \tag{13}$$

where log is the logarithm map of $SE(3)$ and $T_k \in GA(3)$ is the transformation matrix equivalent to the pose $\boldsymbol{\eta}_k$.

Similarly to [25,29], our objective is to estimate f by training a Gaussian process on the vertical sonar measurements. The learned function is then used to estimate the most likely angles θ_{ij}^h inside the horizontal sonar beams measurements.

4.2. Algorithm Description

In this section, we provide a detailed description of our algorithm which is summarized in Algorithm 1.

First, we compute the curvilinear abscissa for each pose of the trajectory based on Equation (13) (line 1). The surface prior f_p is then estimated by fitting an elliptic cylinder to the vertical sonar points (line 2). The Gaussian process (GP) modeling the environment surface is trained using the vertical sonar measurements and their corresponding curvilinear abscissa (line 3). Once done, for

each horizontal sonar data, we estimate the most likely elevation angles θ_{ij}^{h} (line 5) and compute their corresponding uncertainties (line 7). In the following, we detail the process for each step.

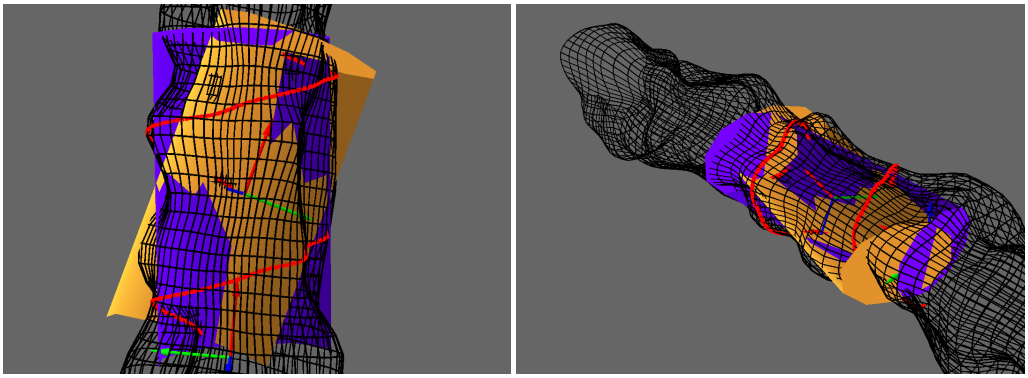
4.2.1. Environment Prior f_p

In this section, we explain how our environment prior f_p is defined. Based on typical karstic environment, the natural choice is to use an elliptic cylinder as a prior shape. For a given pose $\boldsymbol{\eta}$ on the trajectory (equivalently its curvilinear abscissa) and a given sonar rotation angle ψ^v, $f_p(\boldsymbol{\eta}, \psi^v)$ returns the distance to the prior shape.

In order to fit this primitive on the vertical sonar measurements, we follow an approach similar to [31]. First, as for any non-linear optimization approach, we need a good initial shape. It is obtained by computing robust principal component analysis (RPCA) [32] on the vertical sonar point cloud. It provides three principals vectors which, in the case of a perfect elliptic cylinder, should correspond to the cylinder axis, the major and the minor elliptic axis. If the length of the estimated cylinder is higher than its width, then the eigenvector with the highest eigenvalue is the cylinder axis. However, we cannot make this assumption in our case. To find the cylinder axis among those principal vectors, we used the fact that every tangent on a cylinder surface is orthogonal to its axis. This means that by approximating tangent vectors by the vector formed by two successive points, their projections on the PCA bases should be minimal along the cylinder axis.

Then, the elliptic section is estimated by fitting the projections of all the points in the plane spanned by the two other principals axis [33]. The PCA base is then rotated around the cylinder axis to be aligned with the ellipse axis.

As can be seen with the orange elliptic cylinder in Figure 5, the dissymetric distribution of the vertical sonar points has a huge influence on the RPCA result. To further improve our estimation, we use the Levendberg–Marquardt (LM) algorithm to obtain the shape which minimizes orthogonal distances to the elliptic cylinder. The estimation vector $x \in \mathbb{R}^8$ representing our elliptic cylinder is composed of a 6D pose in $SE(3)$ and the scales along the minor and major ellipse axis. Here, we use on-manifold optimization on $SE(3)$, as described in [22]. The final prior can be seen as the purple cylinder in Figure 5.



(**a**) Top view. (**b**) Side view.

Figure 5. Elliptic cylinder fitting of vertical sonar points for the prior shape illustrated from Top (**a**) and Side (**b**) views. The RPCA results are represented in orange and the final results after Levendberg–Marquardt optimization in purple. The RPCA base is represented by a frame in the center of the cylinder.

4.2.2. Gaussian Process Regression and Kernel Choices

In this section, we explain the choice of kernels for the following Gaussian process regression. Based on our formulation of the surface through the functional f (cf. Section 4.1), we exploit its

cylindrical structure by combining a kernel on the trajectory abscissa and a kernel on the sonar rotation angle:

$$K_{prod}\left([s\ \psi^v]^T,[s'\ \psi'^v]^T\right) = K_s(s,s')K_\psi(\psi^v,\psi'^v) \tag{14}$$

K_s can be any kernel as proposed in Section 3.4. For K_ψ, as we deal with angles, we cannot use a simple euclidean distance. When choosing the metric to use, we have to ensure that the resulting kernel is positive definite [34,35]. We can define two distance d_{chord} and d_{geo} based respectively on the chordal distance and the geodesic distance on $\mathbb{S}^1$:

$$d_{chord}(\psi,\psi') = 2\sin\left(\frac{\psi-\psi'}{2}\right) \tag{15}$$

$$d_{geo}(\psi,\psi') = acos\left(\cos\left(\psi-\psi'\right)\right) \tag{16}$$

In [35], it is shown that for geodesic distance d_{geo} a C^2-Wendland function [36] can be used to define a valid kernel

$$\forall 0 \le t \le \pi,\ W_\pi(t) = \left(1+\tau\frac{t}{\pi}\right)\left(1-\frac{t}{\pi}\right)_+^\tau,\ \tau \ge 4 \tag{17}$$

For $d_{chordal}$, any kernel can be used.

In this paper, we only consider exponential K^{exp} and Matern52 $K^M_{\frac{5}{2}}$ kernels with chordal distance for angles. In the experimental section, we investigate the effects of those different kernels and distances on the surface estimation.

4.2.3. Maximum Likelihood Estimation of θ

Originally, we cannot obtain the 3D points observed by the horizontal sonar as any point in its beam may return the same measurement. We exploit the estimated surface by the vertical sonar to infer the elevation angles θ for each horizontal sonar measurements. The idea is that elevation angles corresponding to 3D points near the estimated surface are good estimate. This is illustrated in Figure 6. Hence, for each horizontal sonar measurement ψ^h_k, ρ^h_k, we search for the elevation angles solutions of:

$$\begin{aligned}\hat\theta &= arg\max_{\theta\in\Theta^H} p\left(\hat\rho^v | f,\psi^h_k,\rho^h_k,\boldsymbol{\eta}_{\sigma_h(k)},\theta\right)\\ &= arg\max_{\theta\in\Theta^H} ln\left(p\left(\hat\rho^v | f,\psi^h_k,\rho^h_k,\boldsymbol{\eta}_{\sigma_h(k)},\theta\right)\right)\\ &= arg\max_{\theta\in\Theta^H} J(\theta;\psi^h_k,\rho^h_k)\end{aligned} \tag{18}$$

where $\hat\rho^v$ is the estimated range we would have obtained if this point was observed by the vertical sonar. In fact, depending on the environment configuration, several angles can correspond to one measure as previously illustrated in the Figure 4. Thus, instead of solving Equation (18), we search for local log-maximum likelihood values $\hat{\boldsymbol{\Theta}}$:

$$\hat{\boldsymbol{\Theta}} = \left\{\hat\theta \in \Theta^H \mid \frac{\partial J}{\partial\theta}(\hat\theta) = 0 \text{ and } \frac{\partial^2 J}{\partial\theta^2}(\hat\theta) < 0\right\} \tag{19}$$

The corresponding algorithm is described in Algorithm 2 and detailed in the following.

As it is difficult to obtain a closed-form expression of $p\left(\hat\rho^v | f,\rho^h_k,\psi^h_k,\boldsymbol{\eta}_{\sigma^h(k)},\theta\right)$ relative to θ, we uniformly sample θ values in the range Θ^H. In practice, $\Theta^H = \left[-\frac{b}{2},\ \frac{b}{2}\right]$ where b is the beam vertical aperture. We denote those samples $\theta_q, q \in [\![1, n_q]\!]$. For each sample θ_q, we compute the corresponding virtual 3D points $\mathbf{x}(\theta_q)$

$$\mathbf{x}(\theta_q) = g^h\left(\begin{bmatrix}\rho^h & \theta_q & \psi^h\end{bmatrix}, k\right) \tag{20}$$

with g^h defined by Equation (1). To assess the likelihood of having measured this point on the surface, we need to use the previously trained GP model f. We therefore have to express this point in terms of f inputs and output. In other words, we need the curvilinear abscissa $\hat{s}_q$, the rotation angle $\hat{\psi}_q^v$ and range $\hat{\rho}_q^v$ as if the point was virtually seen by the vertical sonar. This is illustrated in Figure 7.

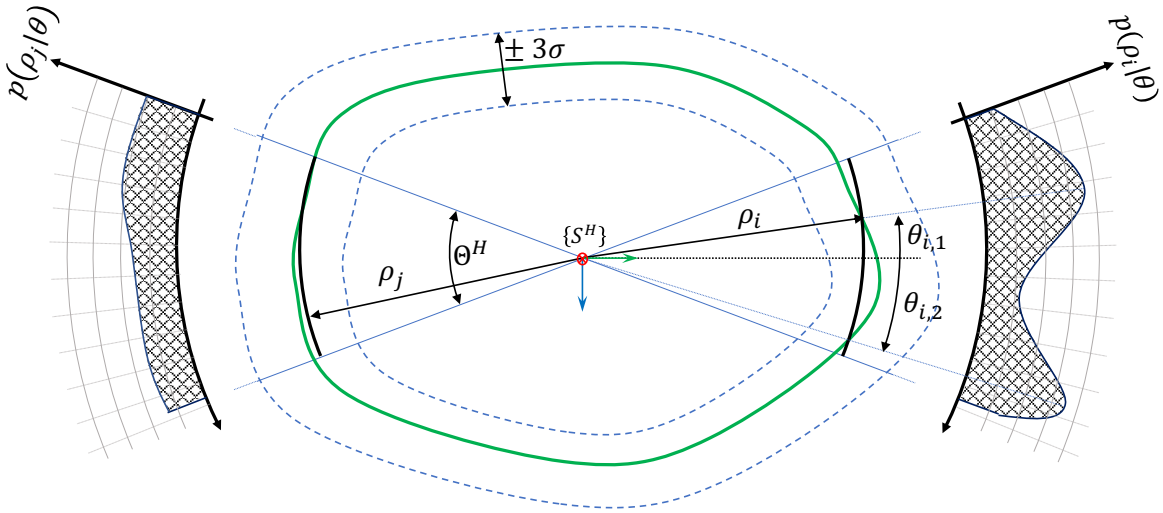


Figure 6. Illustration of elevation angle maximum likelihood estimation on a vertical slice. The green curve is the mean surface (cofounded here with the true surface for visibility) and blue curves are the confidence interval at $\pm 3\sigma$. Note that in this particular configuration $\rho = \rho^h = \hat{\rho}^v$.

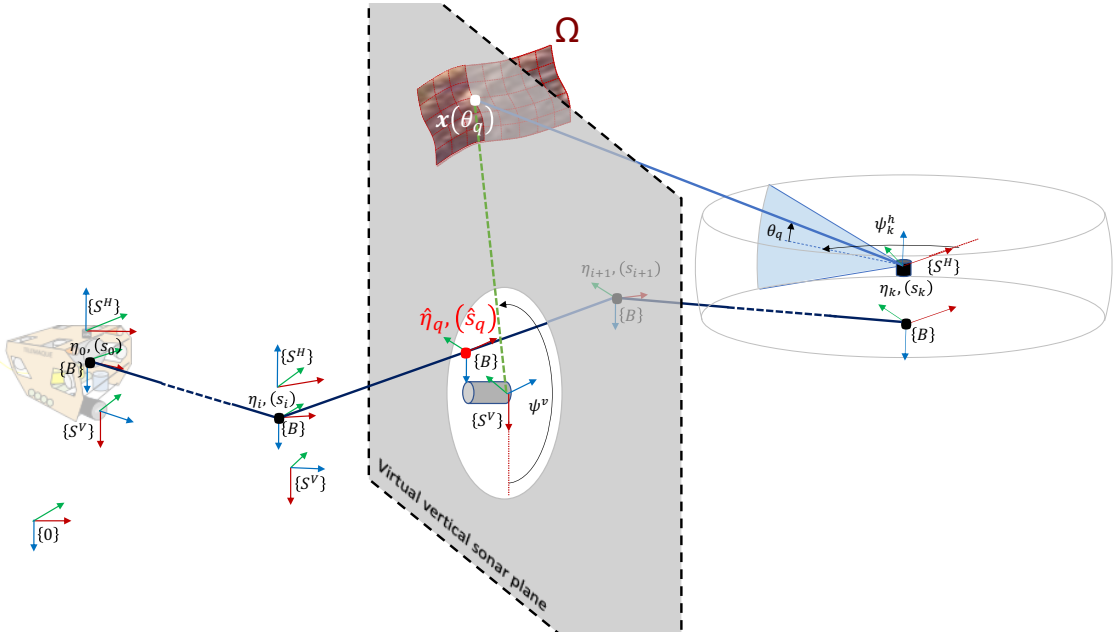


Figure 7. Elevation angle estimation illustration where the vertical sonar plane is parallel to the ZY plane of the robot.

To obtain those parameters, we proceed as follows. First, let us denote $\mathbf{n}^v_{|\{S^V\}}$ the rotation axis of the vertical sonar expressed in its local frame. It is also the normal vector of the sonar plane. We are searching a pose $\hat{\boldsymbol{\eta}}_q$ along the trajectory such that $\mathbf{x}(\theta_q)$ belongs to the vertical sonar plane. To do so, we need to interpolate the robot trajectory. The first step is to determine in which segments of the trajectory $\hat{\boldsymbol{\eta}}_q$ could belong to (several segments can be found). We recall that we note the global robot position by $\boldsymbol{\tau}_i$ and its orientation by the quaternion $\mathbf{q}_i$. The vertical sonar orientation quaternion is also defined by $\mathbf{q}_V$. The rotation axis of the vertical sonar expressed in the global frame $\mathbf{n}^v_{|\{0\},i}$ when the robot is in its i-th pose $\boldsymbol{\eta}_i$ is given by

$$\mathbf{n}^v_{|\{0\},i} = R\left(\mathbf{q}_i\right) R\left(\mathbf{q}_V\right) \mathbf{n}^v_{|\{S^V\}} \tag{21}$$

A segment (represented by indexes $[i, i+1]$) is considered if the projection of the vector $\mathbf{x}(\theta_q) - \boldsymbol{\tau}_i$ on the vertical rotation axis $\mathbf{n}^v_{|\{0\},i}$ has a change of sign between the beginning and end of the segment (Figure 8). Indeed, thanks to the intermediate value theorem, such a segment contains the pose for which the projection is null. Formally, we consider the set I_q of starting index such that

$$I_q = \left\{ i \mid \left\langle \mathbf{n}^v_{|\{0\},i}, \mathbf{x}(\theta_q) - \boldsymbol{\tau}_i \right\rangle \left\langle \mathbf{n}^v_{|\{0\},i+1}, \mathbf{x}(\theta_q) - \boldsymbol{\tau}_{i+1} \right\rangle < 0 \right\} \tag{22}$$

For each selected segment $[i, i+1], i \in I_q$, we denote $\boldsymbol{\eta}_{[i,i+1]} : [0,1] \to SE(3)$ the linear interpolation between the poses $\boldsymbol{\eta}_i$ and $\boldsymbol{\eta}_{i+1}$. $\boldsymbol{\tau}_{[i,i+1]}$ is defined by the first three coordinates of $\boldsymbol{\eta}_{[i,i+1]}$ and is simply given by

$$\boldsymbol{\tau}_{[i,i+1]}(t) = (\boldsymbol{\tau}_{i+1} - \boldsymbol{\tau}_i)t + \boldsymbol{\tau}_i \tag{23}$$

Let define the relative angular pose express in yaw–pitch–roll Euler angles between $\boldsymbol{\eta}_i$ and $\boldsymbol{\eta}_{i+1}$:

$$\begin{bmatrix} \delta\psi & \delta\chi & \delta\phi \end{bmatrix}^T = \text{Euler}\left((\boldsymbol{\eta}_{i+1} \ominus \boldsymbol{\eta}_i)(3:7)\right) \tag{24}$$

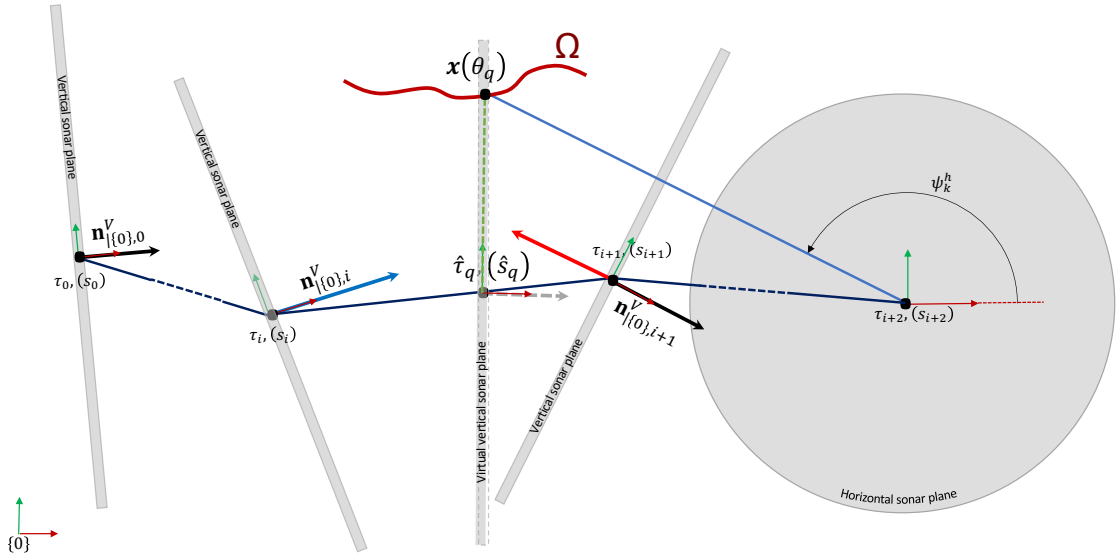


Figure 8. Segment selection for the k-th horizontal sonar measurement and q-th sample $\mathbf{x}(\theta_q)$. We have represented the vertical sonar planes and rotation axis $\mathbf{n}^v_{|\{0\},i}$ along the trajectory from a top view. The blue (resp. red) vector corresponds to $\left\langle \mathbf{n}^v_{|\{0\},i}, \mathbf{x}(\theta_q) - \boldsymbol{\tau}_i \right\rangle \mathbf{n}^v_{|\{0\},i}$ (resp. $\left\langle \mathbf{n}^v_{|\{0\},i+1}, \mathbf{x}(\theta_q) - \boldsymbol{\tau}_{i+1} \right\rangle \mathbf{n}^v_{|\{0\},i+1}$).

We then have the associated interpolated rotation matrix $\delta R(t)$:

$$\delta R(t) = R\left(\delta\psi t, \delta\chi t, \delta\phi t\right) \tag{25}$$

The global rotation matrix $R(t)$ is given by

$$R(t) = R_i \delta R(t) \tag{26}$$

We can now define the linear interpolation of the vertical sonar rotation axis $\mathbf{n}^v_{|\{0\},[i,i+1]}$:

$$\mathbf{n}^v_{|\{0\},[i,i+1]}(t) = R(t) R\left(\mathbf{q}_V\right) \mathbf{n}^v_{|\{S^V\}} \tag{27}$$

We search for the interpolation value $\hat{t}$ such that $\mathbf{x}(\theta_q)$ belongs to the sonar plane with normal $\mathbf{n}^v_{|\{0\},[i,i+1]}(\hat{t})$, that is

$$\left\langle \mathbf{n}^v_{|\{0\},[i,i+1]}, \mathbf{x}(\theta_q) - \boldsymbol{\tau}_{[i,i+1]} \right\rangle (\hat{t}) = 0 \tag{28}$$

We assume small angle variation so that we can approximate δR to the first order

$$\delta R(t) = \begin{bmatrix} 1 & -\delta\phi t & \delta\chi t \\ \delta\phi t & 1 & -\delta\psi t \\ -\delta\chi t & \delta\psi t & 1 \end{bmatrix} \tag{29}$$

$\hat{t}$ is the root in $[0, 1]$ of the following order 2 polynomial expression

$$\mathbf{n}^{v^T}_{|\{S^V\}} R\left(\mathbf{q}_V\right)^T R(\hat{t})^T \left(\left(\boldsymbol{\tau}_i - \boldsymbol{\tau}_{i+1}\right)\hat{t} + \mathbf{x}(\theta_q) - \boldsymbol{\tau}_i\right) = 0 \tag{30}$$

We then have $\hat{\boldsymbol{\eta}}_q = \boldsymbol{\eta}_{[i,i+1]}(\hat{t})$ the robot pose corresponding to the computed $\hat{t}$. We can now compute the curvilinear abscissa $\hat{s}$ corresponding to $\hat{t}$

$$\hat{s}_q = s_i + L^{\hat{\boldsymbol{\eta}}_q}_{\boldsymbol{\eta}_i} \tag{31}$$

To compute $\hat{\psi}^v_q$, we express $\mathbf{x}(\theta_q)$ in the vertical sonar frame coordinates corresponding to the robot pose $\hat{\boldsymbol{\eta}}_{q,i}$

$$\hat{\mathbf{x}}(\theta_q)_{|\{S^V\}} = \mathbf{x}(\theta_q) \ominus \left(\hat{\boldsymbol{\eta}}_q \oplus \mathbf{p}^V\right) = \begin{bmatrix} \hat{x}^v & \hat{y}^v & 0 \end{bmatrix}^T \tag{32}$$

$$\hat{\psi}^v_q = atan2(\hat{y}^v, \hat{x}^v) \tag{33}$$

Finally, the estimated range $\hat{\rho}^v_q$ is simply the L2 norm of $\hat{\mathbf{x}}(\theta_q)_{|\{S^V\}}$

$$\hat{\rho}^v_q = ||\hat{\mathbf{x}}(\theta_q)_{|\{S^V\}}||_2 \tag{34}$$

We can now compute the log-likelihood $J(\theta_q; \psi^h_k, \rho^h_k) = ln\left(p\left(\hat{\rho}^v_q | f, \psi^h_k, \rho^h_k, \boldsymbol{\eta}_{\sigma_h(k)}, \theta_q\right)\right)$

$$\rho^{*,v}_q \sim \mathcal{N}(\bar{\rho}^{*,v}_q, \sigma^2_{\rho^{*,v}_q}) \tag{35}$$

$$J(\theta_q; \psi^h_k, \rho^h_k) = -\left(\frac{1}{2\sigma^2_{\rho^{*,v}_q}}(\bar{\rho}^{*,v}_q - \hat{\rho}^v_q)^2 + ln(\sigma_{\rho^{*,v}_q}) + \frac{ln(2\pi)}{2}\right) \tag{36}$$

The expressions for Equation (35) can be found in ([23], p17 Equation (2.25/2.26)) and are obtained using the GP regression trained to estimate f (Section 3.4).

4.2.4. Estimation Uncertainty

In the previous section, we explained how the set $\hat{\Theta}$ is obtained. We are also interested in its related uncertainties. In other words, we want to define the probability distributions $p(\hat{\theta})$. A normal distribution is not adapted as its support is infinite while the width of our horizontal sonar beam is restricted to its aperture. A good choice is a scaled beta distribution [37] defined by two parameters α, β

$$\hat{\theta} \sim Beta\,(\alpha, \beta, \theta_{min}, \theta_{max}) \tag{37}$$

In our case, we consider $\theta_{max} = -\theta_{min} = \frac{b}{2}$ where b is the horizontal sonar beam width. We then have the following probability density function

$$f(\theta; \alpha, \beta, b) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \frac{(\theta + \frac{b}{2})^{\alpha-1}(\frac{b}{2} - \theta)^{\beta-1}}{b^{\alpha+\beta-1}} \tag{38}$$

where Γ is the gamma function which extends the factorial function to complex numbers and is defined by

$$\forall z \in \mathbb{C},\ \Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx,\ Re(z) > 0 \tag{39}$$

For each estimated $\hat{\theta}$ from the previous section, the parameters α, β can be computed from a given mean $\bar{\theta}$ and variance σ_θ^2:

$$\alpha = \left(b + 2\bar{\theta}\right) \frac{b^2 - 4(\bar{\theta}^2 + \sigma_\theta^2)}{8b\sigma_\theta^2} \tag{40}$$

$$\beta = \left(b - 2\bar{\theta}\right) \frac{b^2 - 4(\bar{\theta}^2 + \sigma_\theta^2)}{8b\sigma_\theta^2} \tag{41}$$

In our case, the mean is simply the estimated θ ie $\bar{\theta} = \hat{\theta}$. The variance σ_θ^2 requires more computation and is estimated as the inverse of the Fisher information $\mathcal{I}(\bar{\theta})$:

$$\sigma_\theta^2 = \frac{1}{\mathcal{I}(\bar{\theta})} \tag{42}$$

$$\begin{aligned} \mathcal{I}(\theta) &= -E\left(\frac{\partial^2 ln\left(p(\rho^v|\theta)\right)}{\partial\theta^2}|\theta\right) \\ &= -\int \frac{\partial^2 ln(p(\rho^v|\theta)}{\partial\theta^2} p(\rho^v|\theta) dr \end{aligned} \tag{43}$$

Direct computation of Equation (43) gives us (see Appendix A for the derivation):

$$\sigma_\theta^2 = \frac{\sigma_{\rho^v}^2(\bar{\theta})}{2\left(\frac{\partial\sigma_{\rho^v}}{\partial\theta}|_{\bar{\theta}}\right)^2 + \left(\frac{\partial\bar{\rho^v}}{\partial\theta}|_{\bar{\theta}}\right)^2} \tag{44}$$

Note that a uniform distribution can still be represented by a beta distribution with shape parameters $\alpha = \beta = 1$. Indeed, from Equations (39) and (38) we have

$$f(\theta; 1, 1, b) = \frac{\Gamma(2)}{b\Gamma(1)\Gamma(1)} = \frac{1}{b} \tag{45}$$

which is the density probability for a uniform distribution on $\left[-\frac{b}{2}, \frac{b}{2}\right]$.

We now define a threshold σ_l to filter out variances σ_θ leading to invalid Beta distribution parameters . First, the parameters α, β of a Beta distribution must be strictly positive:

$$\alpha \geq 0, \beta \geq 0 \iff \sigma_\theta^2 \leq \left(\frac{b}{2}\right)^2 - \bar{\theta}^2 \tag{46}$$

Furthermore, as we consider each distribution as unimodal, we must have $\alpha \geq 1$ or $\beta \geq 1$. (In the case $\alpha, \beta < 1$, the beta distribution is bimodal with modes at $\{-\frac{b}{2}, \frac{b}{2}\}$. A mode corresponds to a local maxima of a probability density function (pdf).) From the expressions (40) and (41), we have

$$\alpha \geq 1 \iff \sigma_\theta^2 \leq A(\bar{\theta}) = \frac{\left(b + 2\bar{\theta}\right)^2 \left(b - 2\bar{\theta}\right)}{4\left(3b + 2\bar{\theta}\right)} \tag{47}$$

$$\beta \geq 1 \iff \sigma_\theta^2 \leq B(\bar{\theta}) = A(-\bar{\theta}) \tag{48}$$

This implies

$$\sigma_\theta^2 \leq \sigma_l^2\left(\bar{\theta}\right) = max\left(A(\bar{\theta}), A(-\bar{\theta})\right) = A(|\bar{\theta}|) \tag{49}$$

It is also easy to verify that the first constraint $\alpha, \beta \geq 0$ is included in the second as we have

$$\forall \bar{\theta} \in \left[-\frac{b}{2}, \frac{b}{2}\right], \sigma_l^2\left(\bar{\theta}\right) = A(|\bar{\theta}|) \leq \left(\frac{b}{2}\right)^2 - \bar{\theta}^2 \tag{50}$$

We also add another threshold σ_m to filter out valid but too uncertain estimations. It is defined by

$$k\sigma_m = \frac{b}{2} \tag{51}$$

The parameter k controls the filter sensibility, with higher values being more restrictive. In our implementation, we used a default value $k = 3$. The final threshold σ_T is then

$$\sigma_T^2(\bar{\theta}) = min\left(\sigma_l^2\left(\bar{\theta}\right), \sigma_m^2\right) \tag{52}$$

Figure 9 shows the graphs of σ_l, σ_m and σ_T.

Figure 10 illustrates the uncertainty estimation process. The colors used to represent the beta distributions directly on the measurement arc are obtained through the following mapping

$$m : \begin{cases} \mathbb{R}^+ & \to [0, 1] \\ x & \mapsto \frac{2}{\pi} arctan\left(bx\right) \end{cases} \tag{53}$$

Note that the central color (turquoise) is obtained for $x = \frac{1}{b}$, which is the pdf for a uniform distribution.

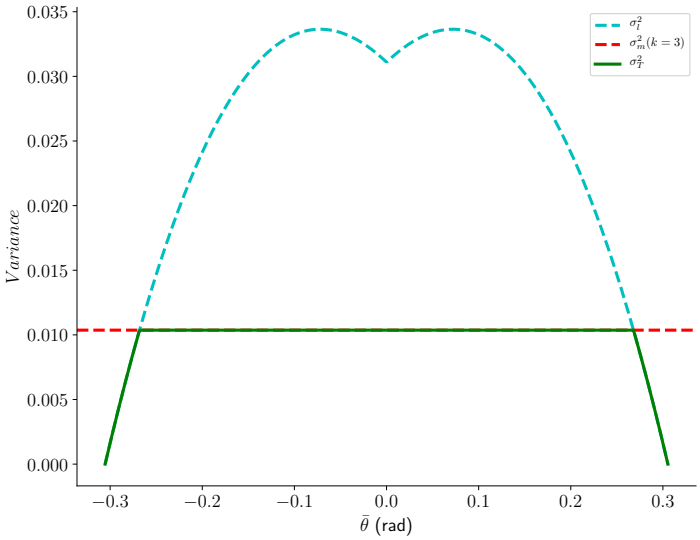


Figure 9. Thresholds σ_l, σ_m and σ_T relative to $\bar{\theta}$ with $b = 0.61$ radians (35 deg).

Algorithm 1 Scan merging.

Input: Sonar measurements $\mathcal{M}^h$, $\mathcal{M}^v$, Robot trajectory H

Output: Estimated horizontal measurements $p(\hat{\theta})$

$\mathbf{s} = \begin{bmatrix} s_1 & \dots & s_{n_r} \end{bmatrix}^T \leftarrow s(H)$ ▷ Curvilinear abscissa, Equation (12)

$f_p \leftarrow fitEllipticCylinder(\mathcal{M}^v)$ ▷ Surface prior

$f \leftarrow \mathcal{GP}(\mathbf{s},\ \mathcal{M}^v, f_p)$ ▷ Surface Estimation by Gaussian Process

for each ψ_i^h, ρ_{ij}^h **do** ▷ Estimation

  $\hat{\mathbf{\Theta}} =$ **ElevationAngleMLE** $\left(f, \mathbf{s}, \psi_i^h, \rho_{ij}^h\right)$ ▷ See Algorithm 2

  for $\hat{\theta} \in \hat{\mathbf{\Theta}}$ **do**

    $p(\hat{\theta}) \leftarrow$ **UncertaintyEstimate** $\left(p(\rho^v|\theta), \hat{\theta}\right)$

  end for

end for

Algorithm 2 Elevation angle MLE.

Input: Trained surface GP f, a horizontal sonar measure (ψ_k^h, ρ_k^h), robot trajectory H and the corresponding curvilinear abscissa $\mathbf{s}$, Rotation axis of vertical scan expressed in the vertical sonar frame $\mathbf{n}^v_{|\{S^V\}}$, Number of θ samples n_q

Output: Estimated local maximum elevation angles $\hat{\boldsymbol{\Theta}}$

$\hat{\boldsymbol{\Theta}} \leftarrow \emptyset$, $\Delta_\theta \leftarrow \frac{\theta_{max} - \theta_{min}}{n_q}$
for q **do**
  $\theta_q \leftarrow \theta_{q-1} + \Delta_\theta$
  $\mathbf{x} \leftarrow g^h\left(\begin{bmatrix}\rho_k^h & \theta_q & \psi_k^h\end{bmatrix}, k\right)$ ▷ Equation (20)
  $I \leftarrow \emptyset$
  for $i \in [1, n^r - 1]$ **do**
    $\mathbf{n}^v_{|\{0\},i}, \mathbf{n}^v_{|\{0\},i+1} \leftarrow R(\mathbf{q}_i) R(\mathbf{q}_V) \mathbf{n}^v_{|\{S^V\}}, R(\mathbf{q}_{i+1}) R(\mathbf{q}_V) \mathbf{n}^v_{|\{S^V\}}$ ▷ Equation (21)
    if $\left\langle \mathbf{n}^v_{|\{0\},i}, \mathbf{x} - \boldsymbol{\tau}_i \right\rangle \left\langle \mathbf{n}^v_{|\{0\},i+1}, \mathbf{x} - \boldsymbol{\tau}_{i+1} \right\rangle < 0$ **then** ▷ Equation (22)
      $I \leftarrow i$
    end if
  end for
  $\hat{\theta}_q \leftarrow 0$
  $l_q^{max} \leftarrow 0$ ▷ Maximum likelihood value
  for $i \in I$ **do**
    $\mathbf{n}^v_{|\{0\},[i,i+1]}(t) \leftarrow linearInterp\left(\mathbf{n}^v_{|\{0\},i}, \mathbf{n}^v_{|\{0\},i+1}\right)$
    $\boldsymbol{\tau}_{[i,i+1]}(t) \leftarrow linearInterp(\boldsymbol{\tau}_i, \boldsymbol{\tau}_{i+1})$
    $\boldsymbol{\eta}_{[i,i+1]}(t) \leftarrow linearInterp(\boldsymbol{\eta}_i, \boldsymbol{\eta}_{i+1})$
    $\hat{t} \leftarrow \text{root}\left(\left\langle \mathbf{n}^v_{|\{0\},[i,i+1]}, \mathbf{x} - \boldsymbol{\tau}_{[i,i+1]} \right\rangle (t)\right)$ ▷ Equations (28) and (30)
    $\hat{\boldsymbol{\eta}}_q \leftarrow \boldsymbol{\eta}_{[i,i+1]}(\hat{t})$
    $\hat{\mathbf{x}}_{|\{S^V\}} = \begin{bmatrix}\hat{x}_i^v & \hat{y}_i^v & 0\end{bmatrix}^T \leftarrow \mathbf{x} \ominus (\hat{\boldsymbol{\eta}}_q \oplus \mathbf{p}^V)$ ▷ Equation (32)
    $\hat{s}_q \leftarrow s_i + L_{\boldsymbol{\eta}_i}^{\hat{\boldsymbol{\eta}}_q}$ ▷ Equation (31)
    $\hat{\psi}_q^v \leftarrow atan2(\hat{y}^v, \hat{x}^v)$ ▷ Equation (33)
    $\hat{\rho}_q^v \leftarrow ||\hat{\mathbf{x}}_{|\{S^V\}}||_2$ ▷ Equation (34)
    $l_q \leftarrow ln(p(\hat{\rho}_i^v | f, \hat{s}_i, \hat{\psi}_i, \theta_q))$ ▷ Equation (36)
    if $l_q > l_q^{max}$ **then**
      $l_q^{max} \leftarrow l_q$
      $\hat{\theta}_q \leftarrow \theta_q$
    end if
  end for
  if $(l_q^{max} < l_{q-1}^{max}) \wedge (l_{q-1}^{max} > l_{q-2}^{max})$ **then** ▷ $\frac{\partial^2 J}{\partial \theta^2} < 0$, Equation (19)
    $\hat{\boldsymbol{\Theta}} \leftarrow \hat{\theta}_q$
  end if
end for

Algorithm 3 Uncertainty estimate.

Input: Likelihood $p(\rho^v|\theta)$, local maximum likelihood $\bar{\theta}$, beam width b

Output: Posterior $p(\hat{\theta})$

1: $I(\bar{\theta}) \leftarrow \frac{2\left(\frac{\partial \sigma_\rho}{\partial \theta}\right)^2 + \left(\frac{\partial \hat{\rho}}{\partial \theta}\right)^2}{\sigma_\rho^2}$ ▷ Fisher information Equation (43)

2: $\sigma_\theta^2 \leftarrow \frac{1}{I(\bar{\theta})}$

3: $\alpha, \beta \leftarrow 1$ ▷ Beta distr. parameters

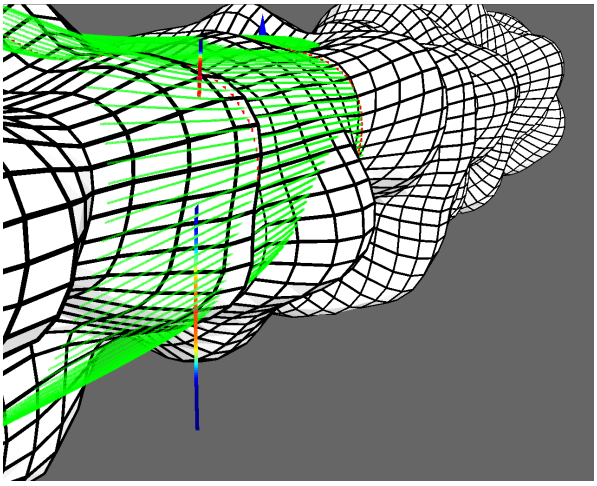
4: **if** $\sigma_\theta \leq \sigma_T$ **then** ▷ Equation (52)

5: $\quad \alpha \leftarrow \left(b + 2\bar{\theta}\right) \frac{b^2 - 4(\bar{\theta}^2 + \sigma_\theta^2)}{8b\sigma_\theta^2}$

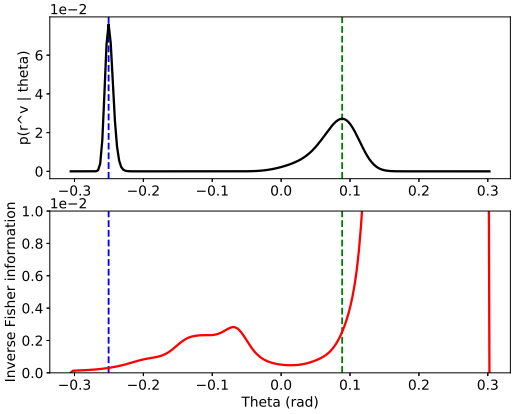
6: $\quad \beta \leftarrow \left(b - 2\bar{\theta}\right) \frac{b^2 - 4(\bar{\theta}^2 + \sigma_\theta^2)}{8b\sigma_\theta^2}$

7: **end if**

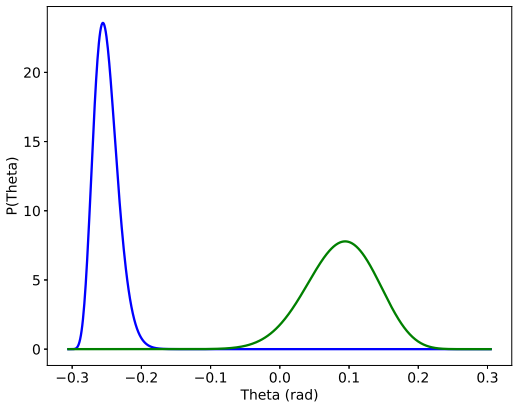
8: $p(\hat{\theta}) \leftarrow Beta\left(\alpha, \beta, -\frac{b}{2}, \frac{b}{2}\right)$



(**a**)



(**b**)



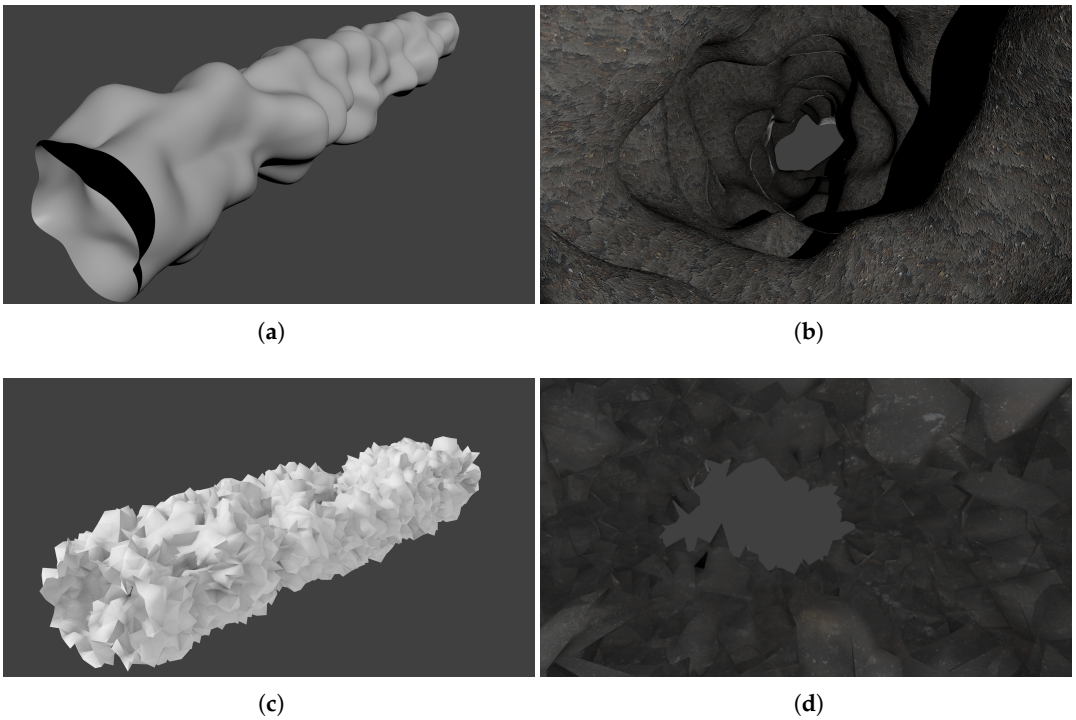
(**c**)

Figure 10. Uncertainty estimation. (**a**) Corresponding horizontal arc measurement with colors related to the beta pdf values. The top (resp. bottom) distribution correspond to the blue (resp. green) curve in the figure below. (**b**) Graphics for θ likelihood (top) and inverse fisher information (bottom). (**c**) Beta distributions corresponding to the two local maxima.

5. Experimental Results

To validate the proposed method described previously, we made experiments on simulated environments shown in Figure 11. Simulation allows to modify the experimental configuration such as the measurements density or the smoothness of the environment surface. It also provides a ground truth so that quantitative analysis can be carried out.

We are currently working on a novel version of pICP adapted to the presented methodology. It will be integrated in a pose-graph SLAM framework and implemented on the Telemaque AUV shown in Section 3.2. Real experiments will then be performed in the Lez karst aquifer in the South of France [38,39].



(**a**) (**b**)

(**c**) (**d**)

Figure 11. Models used in our simulations. Note that for generating sparse measurements, we used a scaled down version of the smooth karst model. (**a**) Outside view of the smooth karst model. (**b**) Inside view of the smooth karst model. (**c**) Outside view of the chaotic karst model. (**d**) Inside view of the chaotic karst model.

5.1. Implementation

Our current implementation consists of a simulation engine implemented in C++ based on the MRPT library [40]. The Gaussian process regression is done in python using the GPy library [41]. In this paper, we do not focus on the algorithm complexity and computation time. It will be part of our future work as a mandatory step towards its use in real applications. Note that final implementation is planned to be fully C++ using the Limbo library [42] for the Gaussian process regression.

5.2. Settings

We present here the settings for which we test our algorithm. We consider three variables which are the surface smoothness, the measure density and the kernel choices.

We can also include the robot trajectory as a fourth variable. If not explicitly stated, we consider a straight linear robot trajectory as it results in clearer display easier to understand, especially for qualitative analysis. Quantitative results are also provided for random trajectories.

To apprehend the effect of environment surface smoothness on the surface estimation, we also evaluate our approach on a chaotic model where we have added some random noise to a cylindrical karst model. We also consider two measurement densities. While fixing the robot movement at each simulation step, we generate dense measurements by considering a large karst (radius of 22m) and sparse measurements with a scaled down karst (radius of 1m). Note that the dense case is the similar to the general density of measurements expected in real karst exploration. The sonar measurements, used as training data for the surface estimation, are computed using ray-tracing on the environment mesh.

For the kernels, we consider the products between kernels on trajectory abscissa $K_s \in \{K^{exp}, K^M_{\frac{3}{2}}, K^M_{\frac{5}{2}}, K^{RBF}\}$ and kernels on the sonar rotation angle $K_\psi \in \{K^{exp} \circ d_{chordal}, K^M_{\frac{5}{2}} \circ d_{chordal}\}$. This is summarized in Table 1.

Table 1. Different settings used in our experiments.

Variable	Values
Environment	Smooth, Chaotic
Measure density	Sparse, Dense
K_s	$K^{exp}, K^M_{\frac{3}{2}}, K^M_{\frac{5}{2}}, K^{RBF}$
K_ψ	$K^{exp} \circ d_{chordal}, K^M_{\frac{5}{2}} \circ d_{chordal}$

5.3. Surface Estimation

5.3.1. Effect of Kernel Choices

In order to quantitatively assess the performance of our algorithm, we have modeled our karst environment using the 3D modeling software Blender (http://www.blender.org). Unless otherwise stated, we optimize the GP with all the kernel parameters presented in Section 3.4. We discuss when needed the benefit and/or effect from fixing some of those parameters. As we are dealing with surface in 3D, it is hard to have a clear representation of the GP results. Figure 12 shows some examples of estimated surface with the mean (green) and the lower/upper bound at $-/+3\sigma$ (dark blue/blue). To assess the effect of kernels on the surface estimation, we represent the surface estimation by considering upper contours in the XY, XZ and ZY planes (The selected ZY plane is set at the middle of the trajectory).

One should choose kernels which provides the best trade-off between confidence (variance) in the estimation and accuracy. Ideally, the variance in the surface estimation should be high enough to contain the real surface in its confidence interval. However, too high variance will lead to uniform distribution for the elevation angle estimation.

Note that as our surface estimation operates per full scan and thus is local, one can adapt dynamically the kernels used during operation depending on a priori information on the robot environment.

Kernel K_s

We fix here $K_\psi = K^M_{\frac{5}{2}} \circ d_{chordal}$ and consider the cases where the lengthscale l_ψ is learned or fixed to 0.5. The objective here is to apprehend the behavior of each kernel K_s. Figure 13a (resp. Figure 13b) shows the Gaussian process regression results in the sparse and dense configuration for $l_\psi = 0.5$ (resp. learned l_ψ in Table 2b). From those graphics, we see that when $l_\psi = 0.5$, the uncertainties in the surface estimations are different in the sparse and dense cases. In the former, we have by increasing order of uncertainty the Matern32, Matern52, rbf and exponential kernels. In the latter, we have Exponential, Matern32, Matern52 and RBF. Based solely on kernels smoothness, we expect the results seen in the dense case. This shows that the kernels behavior depends heavily on the environment configuration. We explain this difference by the bigger variations in measured ranges in the sparse

case compared to the dense case (Figure 12). Obviously, the accuracy of the estimated surface is worse in the sparse case as large part of the surface are unseen by the vertical sonar.

When the l_ψ is learned, we see results similar between the sparse and dense case. The variability of the results obtained shows the importance to consider both kernels K_s and K_ψ simultaneously.

Table 2. Lengthscale l_ψ values in different configurations.

(**a**) Values for the two K_ψ with K_s fixed.

$\mathbf{K_\psi}$	Dense	Sparse
$K^{exp} \circ d_{chordal}$	9.621	75.815
$K^M_{\frac{5}{2}} \circ d_{chordal}$	1.029	0.699

(**b**) Values for $K_\psi = K^M_{\frac{5}{2}} \circ d_{chordal}$.

$\mathbf{K_s}$	Dense	Sparse
Exponential	0.614	0.491
*Matern*32	1.005	0.701
*Matern*52	1.016	0.709
RBF	1.018	0.770

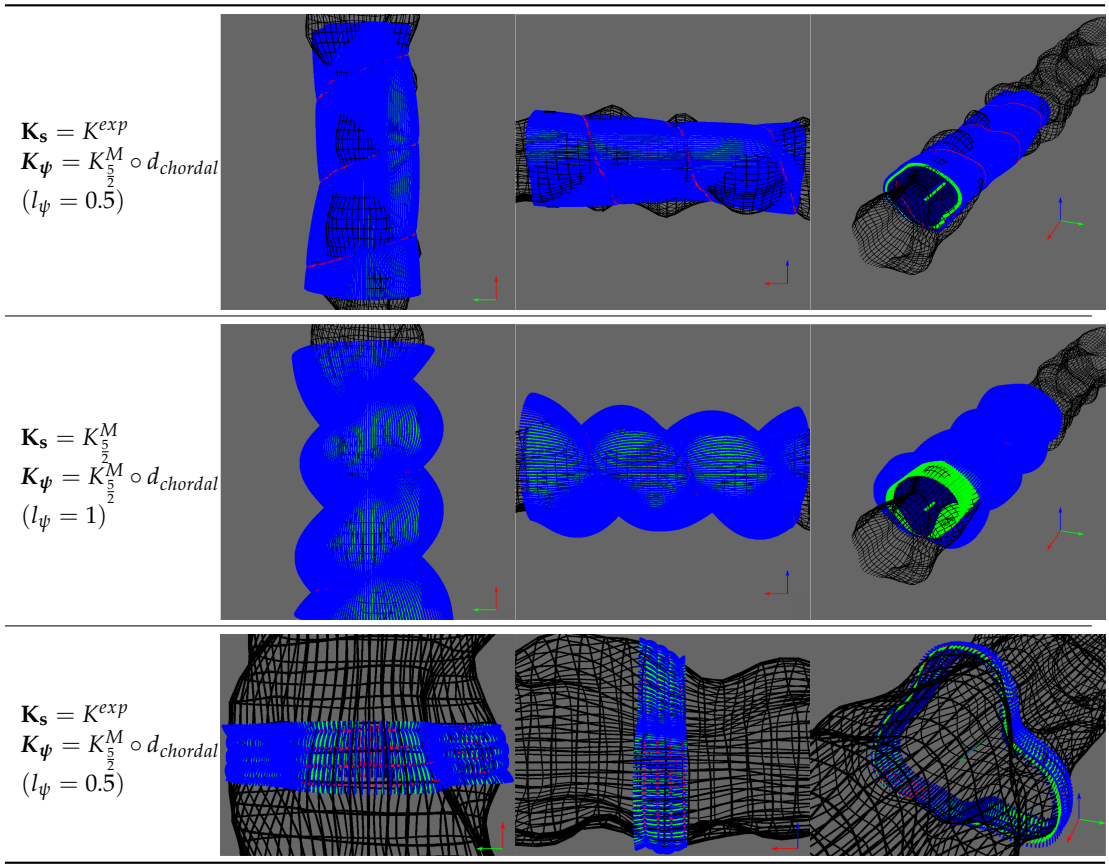
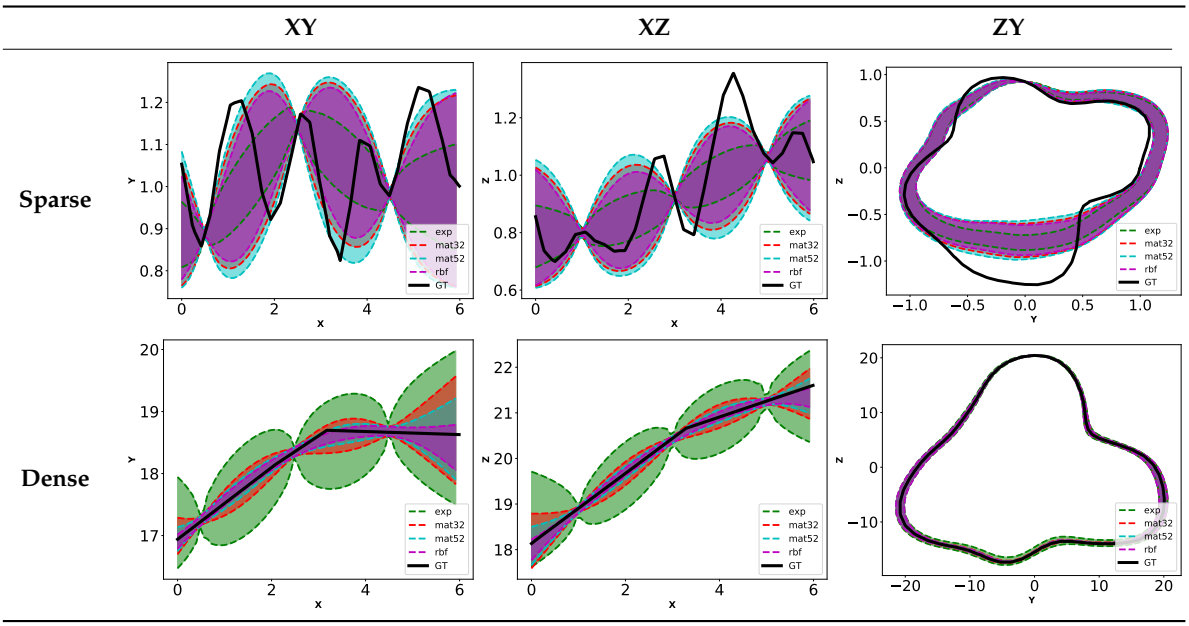
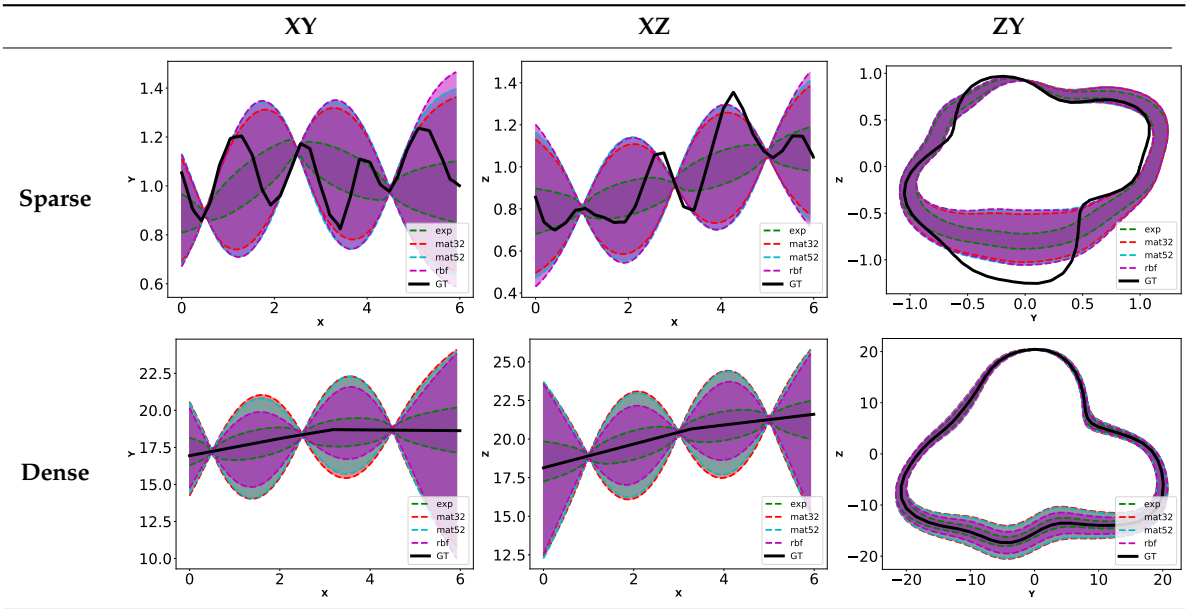


Figure 12. Some 3D views of the resulting GP regressions. The first two lines correspond to the sparse case and the last one to the dense case. The mean surface is represented in green and the lower/upper bound at $+/-3\sigma$ is in dark blue/blue. Red points correspond to the simulated vertical sonar measurements.



(**a**) Fixed Lengthscale $l_\psi = 0.5$.



(**b**) Learned lengthscale l_ψ (values in Table 2).

Figure 13. Contours of estimated surface with $K_\psi = K^M_{\frac{5}{2}} \circ d_{chordal}$ and several K_s with different lengthscale l_ψ.

Kernel K_ψ

We now fix $K_s = K^M_{\frac{5}{2}}$. Figure 14a,b show the results obtained when choosing the $K^{exp} \circ d_{chordal}$ or $K^M_{\frac{5}{2}} \circ d_{chordal}$ kernel for different fixed lengthscales. Table 2 contains the values of lengthscale l_ψ obtained when learning them. We observe that the exponential kernel gives lower variance than required to take into account for the unknown environment. The first row of Table 2 shows that we obtain very high values for learned lengthscale. It shows that the GP failed to find appropriate value for this parameter. On the other hand, the kernel based on Matern52 gives more appropriate uncertainties controlled by the lengthscale parameter. The learned lengthscales are in this case more consistent.

The main conclusion of those results is that the choice of K_s is not obvious. In other works dealing with surface estimation [21,25], $K^M_{\frac{5}{2}}$ and $K^M_{\frac{5}{2}}$ were chosen as being a compromise between K^{exp} and K^{RBF}. However, depending on the configuration, it could be interesting to consider other kernels.

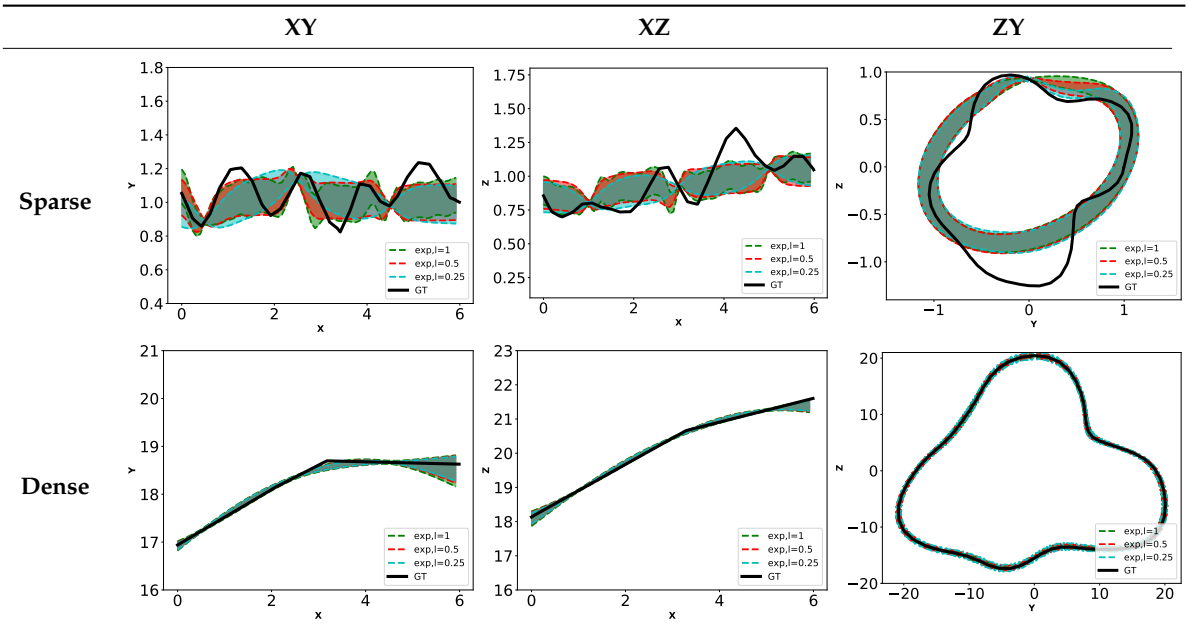
Concerning K_ψ, $K^M_{\frac{5}{2}} \circ d_{chordal}$ seems to be the best choice. Note that we are not exhaustive on the tested kernels. We use here generic kernel used extensively in the machine learning field. It would be interesting to further investigate on other kernels or creation of customized kernels adapted to the task at hand.

From the previous results, the pair of kernels $K_s = K^M_{\frac{5}{2}}, K_\psi = K^M_{\frac{5}{2}} \circ d_{chordal}$ seems to be a reasonable choice and we use them in the following experiments.

We also tested our surface estimation approach on a more chaotic karst environment as shown in Figure 15.

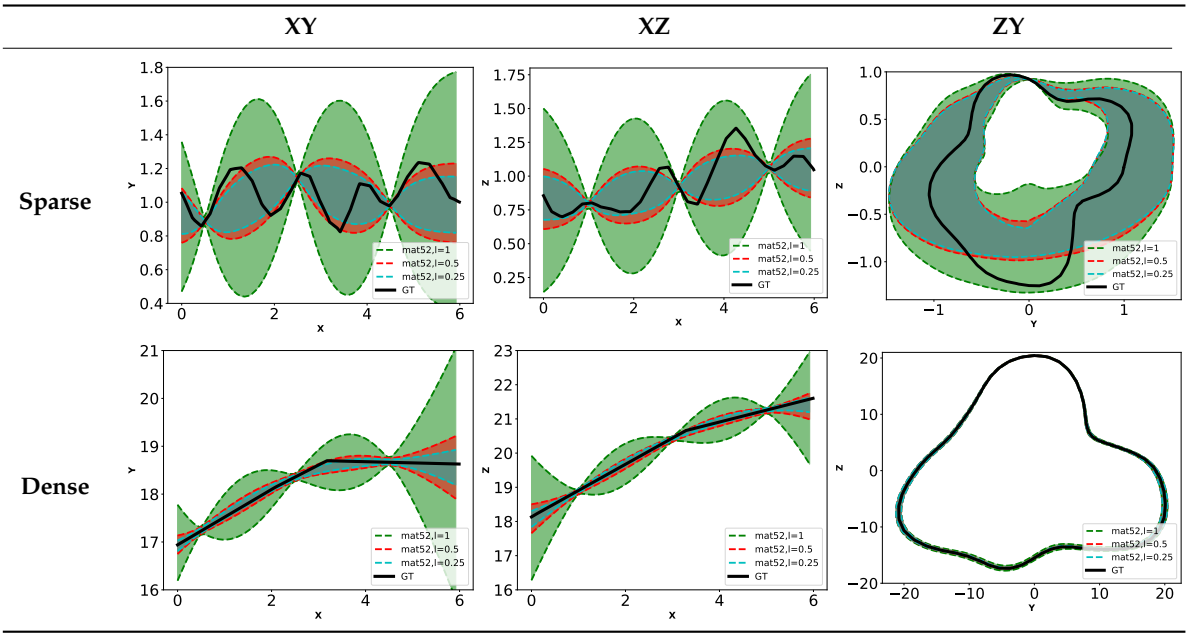
5.3.2. Random Trajectories

In the previous sections, we only considered straight line trajectory to facilitate the visualization. We present here the surface estimation for a robot with a randomly generated trajectory. This trajectory is a simple translation along x with an added Gaussian noise. It is illustrated in Figure 16a. Figure 16 shows the corresponding surface estimation. Note that the estimated surface cannot be directly compared with results in the previous section: the trajectories being differents, the points observed by the vertical sonar are different.



(**a**) $K_\psi = K^{exp} \circ d_{chordal}$

Figure 14. *Cont.*



(**b**) $K_\psi = K^M_{\frac{5}{2}} \circ d_{chordal}$

Figure 14. Contours of estimated surface with fixed $K_s = K^M_{\frac{5}{2}}$.

5.4. Evaluation of Elevation Angle Estimation

In this section, we aim at quantitatively assessing the proposed approach. We simulated random trajectories in modeled karsts in the following configurations: smooth karst with sparse/dense measurements and chaotic karst with dense measurements. The surfaces are estimated using the method described previously. Each simulated horizontal sonar beams is sampled by considering the ranges measured at 5 equally spaced elevation angles. Note that each of those samples corresponds to an arc of measure for which we estimate the elevation angle distribution as represented in Figure 17. For the robot trajectories, we used a RNG generator with a fixed seed for simulation in a given environment. It means that for the sparse, dense and chaotic cases each trial with a different kernel is based on the same randomly generated trajectory. However, note that as the estimated surface depends on the kernel used in the GP regression, the number of non-uniform estimated distributions will vary.

The performance of our algorithm is difficult to measure directly. One possible metric would be to compute the likelihood of the true elevation angle relative to the estimated distribution. However, as discussed in Section 3.3, one range can correspond to different real points and hence to different distributions. We choose to measure the error of an estimated non-uniform distribution (A non-uniform distribution arises when the beam measurement concerns an uncertain part of the estimated surface.) as the distance expectation from the measurement beam to the environment mesh. We denote $d(\theta)$ the distance from a point at elevation angle θ to the surface mesh and call this expected error the posterior error $E_{post}(d)$. We compare it to the prior error $E_{prior}(d)$ which is simply the distance expectation of the same measurement beam with a uniform distribution. Formally those errors are defined as follows

$$E_{prior}(d) = \int_{-\theta_m}^{\theta_m} d(\theta)p(\theta)d\theta = \frac{1}{2\theta_m}\int_{-\theta_m}^{\theta_m} d(\theta)d\theta \quad (54)$$

$$E_{post}(d) = \int_{-\theta_m}^{\theta_m} d(\theta)p(\theta|f)d\theta \quad (55)$$

In practice, those expectations are numerically approximated using the same n_q samples θ_q defined in Section 4.2.3 so that

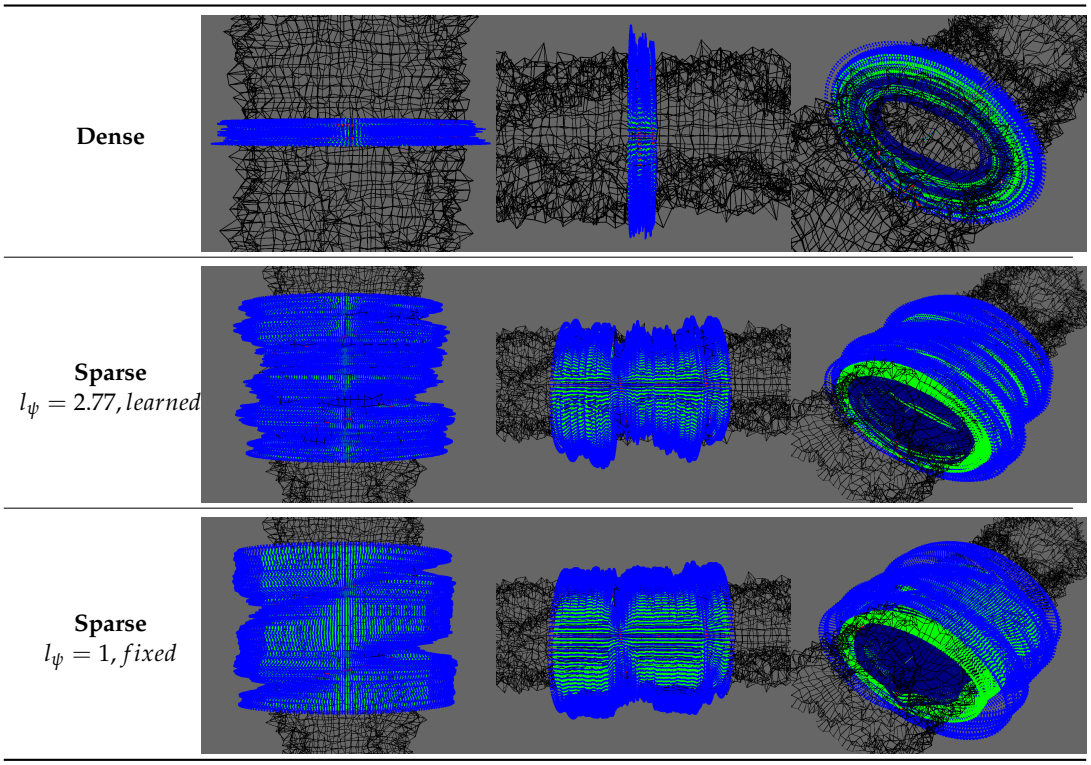
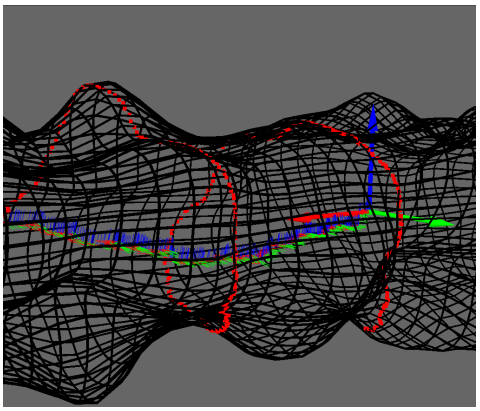


Figure 15. Surface estimation for a chaotic environment.

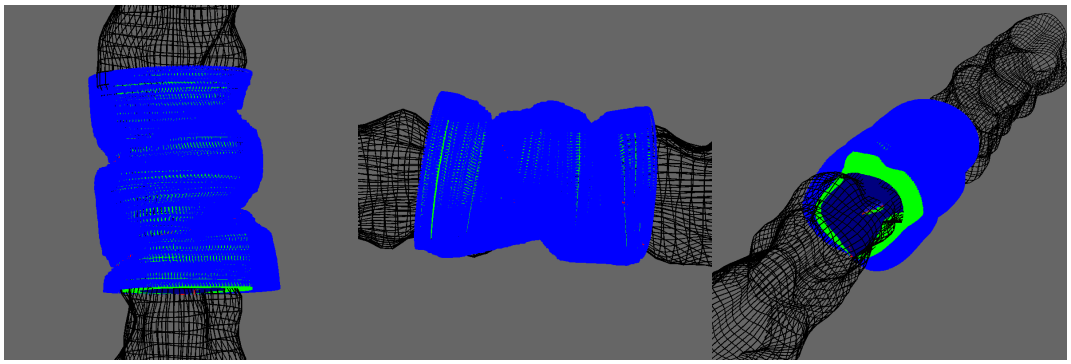
$$E_{prior}(d) \approx \frac{1}{n_q}\sum_{q=1}^{n_q} d(\theta_q) \tag{56}$$

$$E_{post}(d) \approx \frac{\sum_{q=1}^{n_q} d(\theta_q)p(\theta_q|f)}{\sum_{q=1}^{n_q} p(\theta_q|f)} \tag{57}$$

In the following experiments, we use $n_q = 200$ and beam width $b = 35$ deg. For each fixed settings (smooth karst with sparse and dense measurements, chaotic karst), we compute the errors for several kernels K_s while fixing $K_\psi = K_{\frac{5}{2}}^M \circ d_{chordal}$. Error distributions are represented using box-and-whisker diagram. The rectangular box extends from the lower $Q1$ to the upper quartile $Q3$ of the distribution. The filled (resp. dot) line corresponds to the median (resp. mean). The upper whisker is drawn above the upper quartile up to the largest value at a distance below 1.5 times the interquartile range IQR $Q3 - Q1$ (It corresponds to the rectangular box vertical length.). Similarly, the lower whisker is drawn at the lowest value value above 1.5 times the IQR. Values above and below the whiskers are drawn as circles. For each kernel K_s, the non-uniformly distributed beams considered for error computation are different. Indeed, as the surface estimation varies, some beam are estimated as uniform distribution for some kernel but not for others. Hence the need to compute the prior error for each kernel. The results are represented by pair of corresponding diagrams in Figure 18. We also compare the number of non-uniform distributions estimated with each kernel in Figure 18d. Table 3 provides the values of medians and means for each case.



(**a**) Robot trajectory.



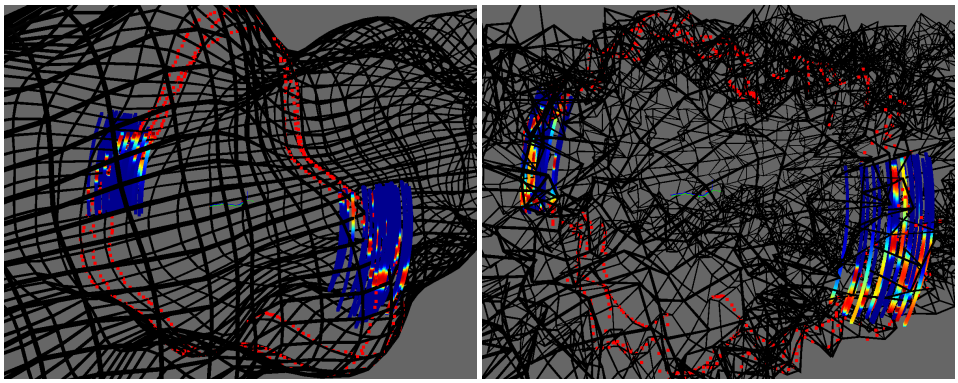
(**b**) Surface estimation.

Figure 16. Surface estimation with a random trajectory.

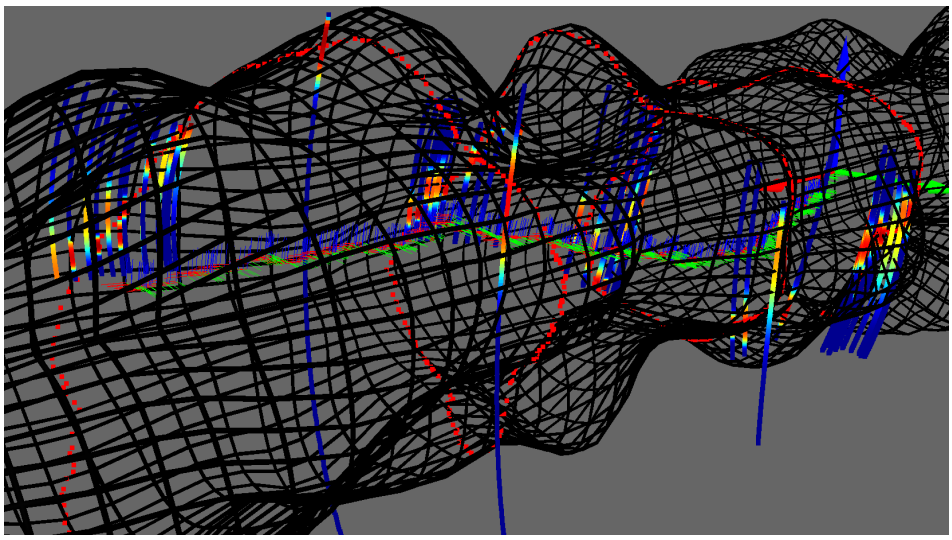
We globally observe that in all cases regardless to the kernel considered our approach improves significantly upon the prior configuration. In particular, we have a relatively high percentage of the distribution with low errors. Besides, the number of estimated non-uniform distributions can vary depending on the kernel used. In our context, the estimated distributions are expected to be further processed by higher level algorithms to reject outliers.

In the sparse case (Figure 18a), the intermediate kernels Matern32 and Matern52 give the best results. We notice some discrepancy in the number of relevant estimations (non-uniform distribution) as can be seen in Figure 18d. This relates to results observed in Table 2a,b where the surface estimated with the exponential Kernel has less uncertainty. In this particular case, this kernel gives an overconfident surface estimation.

In the dense case (Figure 18b), the Matern32 and Matern52 kernels also give the best results with the RBF kernel being more erroneous. There is also a drop in the number of non-uniform distributions obtained for the RBF kernel.



(**a**) Dense case. (**b**) Chaotic.

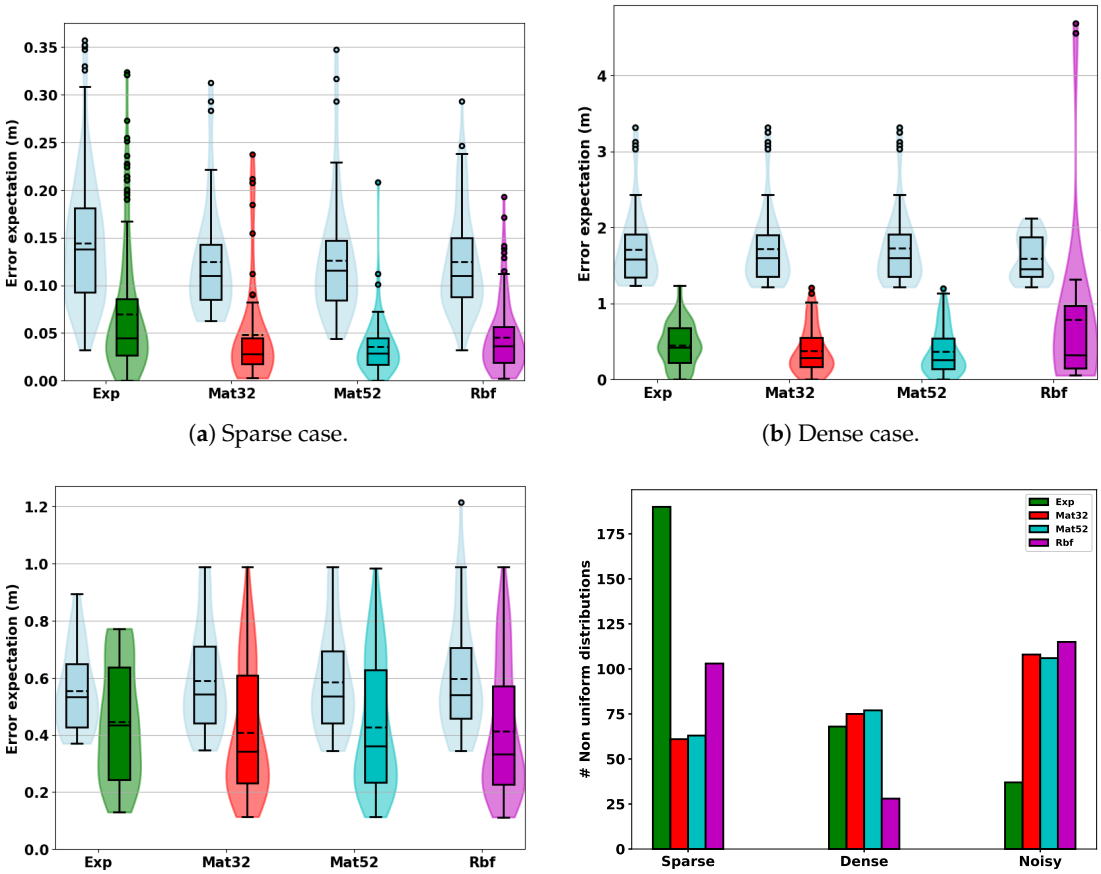


(**c**) Sparse case.

Figure 17. Distributions obtained in the experiment with $K_s = K_{\frac{5}{2}}^{M}$. For better visibility, uniform distribution are not displayed.

Finally, Figure 18c shows the results obtained for a chaotic karst. Note that in this case, the difference between the prior and posterior are relatively small compared to the previous results (beam measurements are generally intersecting the environment several times). Even in this more difficult configuration, an important proportion of the estimated distribution gives low distance expectation error.

Overall those different observations validates what was theoretically expected. In particular, it also justifies the use of Matern52 kernel for K_s as the best default choice. Our experiments is however limited to non exhaustive simulation results. Those results are thus a guideline to better understand the effect of kernel choices and this will be refined based on futur experiments on real data.



(**a**) Sparse case.

(**b**) Dense case.

(**c**) Chaotic case.

(**d**) Number of non-uniform estimated distributions.

Figure 18. Diagrams of error distributions. The box-plot center box represents the interquartile range, IQR = $Q1 - Q3$. The filled (resp. dot) line corresponds to the median (resp. mean). The upper (resp. lower) whisker corresponds to the largest (resp. lowest) value at a distance below (resp. above) 1.5 times the IQR. Values above and below the whiskers are outliers drawn as circles.

Table 3. Medians/means of error distributions. Bold values corresponds to minima.

		Exp	Mat32	Mat52	RBF
Sparse	Prior	0.138/0.144	0.110/0.124	0.115/0.126	0.110/0.125
	Posterior	0.045/0.070	0.028/0.048	**0.028**/**0.035**	0.037/0.046
Dense	Prior	1.579/1.710	1.597/1.718	1.600/1.727	1.454/1.591
	Posterior	0.422/0.452	0.283/0.378	**0.262**/**0.368**	0.323/0.787
Chaotic	Prior	0.534/0.554	0.543/0.591	0.536/0.585	0.541/0.597
	Posterior	0.435/0.447	0.342/0.409	0.361/0.427	**0.334**/**0.414**

6. Conclusions

In this paper, we presented a method for probabilistic sonar scan merging in the context of underwater karstic exploration. It consists of using the measurements of a high resolution sonar (narrow beam) to estimate the measurement of another lower resolution sonar (wide beam). This is done using the cofounding variable linking both sonar measurements which is the environment. The first sonar measurements are used to obtain a probabilistic approximation of the surrounding surface through Gaussian process regression. It was then leveraged to estimate the probability distribution of the second sonar measurements.

In this work, we validated our approach to obtaining a probabilistic estimate of elliptic cylindrical shaped surface such as a karst aquifer. We analyzed the behavior of different kernel choices on the surface estimation. The results obtained were generally in agreement with the smoothness of considered kernels, but some interesting differences were also noted, notably between the sparse and dense measurements case. We also showed quantitative results based on the expected distances between measurement beams and the simulated mesh. We compared the distances obtained with our estimated elevation angles and a uniformly distributed prior distribution. On our simulated smooth karst, we improved the median expected distance by a ratio of 6.1 in the dense measurement case which corresponds to the measurements expected in real experiments and similar to those obtained in the dataset provided by [19]. We also show significant improvements in more difficult settings with a ratio of 4.1 for sparse measurement in a smooth karst and a ratio of 1.62 on a difficult chaotic environment.

This paper aimed at validate our approach and we will focus on improving it in our future works. There are still areas of improvement to explore. We are currently not taking in account uncertainties on the robot trajectory during a full scan acquisition and the noise on the sonar measurements. Besides, we can also improve the GP regression computation time using sparse GP based on inducing points. Another direction is to develop non-stationary kernel (Kernel which does not solely depends on the distance between its two inputs. Informally, it can modeled with variable lengthscale.) as proposed in [29]. It could be particularly interesting to deal with an environment containing strong discontinuities.

The proposed methodology is only a first step towards a real application for underwater karstic exploration. We are currently working on its usage by adapting probabilistic ICP [16] to our estimated point distributions inside a pose graph SLAM framework.

Author Contributions: The work presented in this paper has been done as a collaboration by all the authors. L.L. was the project leader in charge of the supervision. Y.B. developed, implemented and tested the presented method. Both authors have read and agreed to the published version of the manuscript.

Funding: This research is a LabEx NUMEV and MUSE institution project publicly funded through ANR (French National Research Agency) under the "Investissements d'avenir" program (PIA) (references ANR-16-IDEX-0006, ANR-10-LABX-20).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SLAM	Simulataneous Localization and Mapping
(p)ICP	(Probabilistic) Iterative Closest Point
LIDAR	Light Detection Additionally, Ranging
DIDSON	Dual Frequency Identification Sonar
EKF	Extended Kalman Filter
RBPF	Rao-Blackwellized Particle Filter
MSIS	Mechanically Scanning Sonar Systems
IMU	Internial Measurement Unit
DVL	Doppler Velocity Log
PDF	Probability Distribution Function
GP	Gaussian Process
RBF	Radial Basis Function
MLE	Maximum Likelihood Estimation
(R)PCA	(Robust) Principal Component Analysis
LM	Levenberg–Marquardt

Appendix A. Derivation of $\mathcal{I}(\theta)$

The negative log-likelihood of $p(\rho|\theta)$ is simply given by

$$-ln\left(p(\rho|\theta)\right) = ln(\sigma_\rho) + \frac{ln(2\pi)}{2} + \frac{(\rho-\bar{\rho})^2}{2\sigma_\rho^2} \tag{A1}$$

By derivating once, it follows that

$$-\frac{\partial ln\left(p(\rho|\theta)\right)}{\partial\theta} = \frac{1}{\sigma_\rho}\frac{\partial\sigma_\rho}{\partial\theta} - \frac{\rho-\bar{\rho}}{\sigma_\rho^3}\left(\frac{\partial\bar{\rho}}{\partial\theta}\sigma_\rho + (\rho-\bar{\rho})\frac{\partial\sigma_\rho}{\partial\theta}\right) \tag{A2}$$

Further derivating this expression and refactoring it in a polynom in ρ gives

$$\begin{aligned}
-\frac{\partial^2 ln\left(p(\rho|\theta)\right)}{\partial\theta^2} &= a\rho^2 + b\rho + c \\
&= \frac{1}{\sigma^4}\left[\left(3\left(\frac{\partial\sigma_\rho}{\partial\theta}\right)^2 - \frac{\partial^2\sigma_\rho}{\partial\theta^2}\sigma_\rho\right)\rho^2 + \right. \\
&\quad \left(2\frac{\partial\sigma_\rho}{\partial\theta}\left(2\frac{\partial\bar{\rho}}{\partial\theta}\sigma_\rho - 3\frac{\partial\sigma_\rho}{\partial\theta}\bar{\rho}\right) + \sigma_\rho\left(2\frac{\partial^2\sigma_\rho}{\partial\theta^2}\bar{\rho} - \frac{\partial^2\bar{\rho}}{\partial\theta^2}\sigma_\rho\right)\right)\rho + \\
&\quad \left(\frac{\partial\bar{\rho}}{\partial\theta}\sigma_\rho - 3\frac{\partial\sigma_\rho}{\partial\theta}\bar{\rho}\right)\left(\frac{\partial\bar{\rho}}{\partial\theta}\sigma_\rho - \frac{\partial\sigma_\rho}{\partial\theta}\bar{\rho}\right) + \sigma_\rho\left(\frac{\partial^2\bar{\rho}}{\partial\theta^2}\sigma_\rho\bar{\rho} - \frac{\partial^2\sigma_\rho}{\partial\theta^2}\bar{\rho}^2\right) + \\
&\quad \left.\sigma_\rho^2\left(\frac{\partial^2\sigma_\rho}{\partial\theta^2}\sigma_\rho - \left(\frac{\partial\sigma_\rho}{\partial\theta}\right)^2\right)\right]
\end{aligned} \tag{A3}$$

By integrating we have

$$\begin{aligned}
\mathcal{I}(\theta) &= aE(\rho) + bE(\rho^2) + c \\
&= a\bar{\rho} + b(\sigma_\rho^2 + \bar{\rho}^2) + c
\end{aligned} \tag{A4}$$

By developing the term a, b, c from Equation (A3) into Equation (A4), all the terms in $\bar{\rho}$ and $\bar{\rho}^2$ are canceled and we finally obtain

$$\mathcal{I}(\theta) = \frac{2\left(\frac{\partial\sigma_\rho}{\partial\theta}\right)^2 + \left(\frac{\partial\bar{\rho}}{\partial\theta}\right)^2}{\sigma_\rho^2} \tag{A5}$$

References

1. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotics*; Massachusetts Institute of Technology: Cambridge, MA, USA, 2005.
2. Mallios, A.; Ridao, P.; Ribas, D.; Hernández, E. Scan matching SLAM in underwater environments. *Auton. Robot.* **2014**, *36*, 181–198.
3. Palomer, A.; Ridao, P.; Ribas, D. Multibeam 3D underwater SLAM with probabilistic registration. *Sensors* **2016**, *16*, 560. [CrossRef] [PubMed]
4. Burguera, A. A novel approach to register sonar data for underwater robot localization. In Proceedings of the 2017 IEEE Intelligent Systems Conference (IntelliSys), London, UK, 7–8 September 2017; pp. 1034–1043.
5. Taylor, C.J.; Greene, E.A. Hydrogeologic characterization and methods used in the investigation of karst hydrology. *Field Techniques for Estimating Water Fluxes between Surface Water and Ground Water*; Rosenberry, D.O., LaBaugh, J.W., Eds.; US Geological Survey: Reston, VA, USA, 2008; pp. 71–114.
6. Fernández-Madrigal, J.A. *Simultaneous Localization and Mapping for Mobile Robots: Introduction and Methods: Introduction and Methods*; IGI Global: Hershey, PA, USA, 2012.
7. Taketomi, T.; Uchiyama, H.; Ikeda, S. Visual SLAM algorithms: A survey from 2010 to 2016. *IPSJ Trans. Comput. Vis. Appl.* **2017**, *9*, 16. [CrossRef]

8. Mur-Artal, R.; Tardós, J.D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]
9. Hong, S.; Kim, J.; Pyo, J.; Yu, S.C. A robust loop-closure method for visual SLAM in unstructured seafloor environments. *Auton. Robot.* **2016**, *40*, 1095–1109. [CrossRef]
10. Rahman, S.; Li, A.Q.; Rekleitis, I. SVIn2: An Underwater SLAM System using Sonar, Visual, Inertial, and Depth Sensor. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 4–8 November 2019; pp. 1861–1868.
11. Hong, S.; Kim, J. Three-dimensional Visual Mapping of Underwater Ship Hull Surface Using Piecewise-planar SLAM. *Int. J. Control. Autom. Syst.* **2020**, *18*, 564–574. [CrossRef]
12. VanMiddlesworth, M.; Kaess, M.; Hover, F.; Leonard, J.J. Mapping 3d underwater environments with smoothed submaps. In *Field and Service Robotics*; Springer: New York, NY, USA, 2015; pp. 17–30.
13. Teixeira, P.V.; Kaess, M.; Hover, F.S.; Leonard, J.J. Underwater inspection using sonar-based volumetric submaps. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 4288–4295.
14. Kantor, G.; Fairfield, N.; Jonak, D.; Wettergreen, D. Experiments in navigation and mapping with a hovering AUV. In *Field and Service Robotics*; Springer: New York, NY, USA, 2008; pp. 115–124.
15. Fairfield, N.; Kantor, G.; Wettergreen, D. Real-time SLAM with octree evidence grids for exploration in underwater tunnels. *J. Field Robot.* **2007**, *24*, 3–21. [CrossRef]
16. Montesano, L.; Minguez, J.; Montano, L. Probabilistic scan matching for motion estimation in unstructured environments. In Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Hamburg, Germany, 28 September–2 October 2005; pp. 3499–3504.
17. Chen, L.; Yang, A.; Hu, H.; Naeem, W. RBPF-MSIS: Toward Rao-Blackwellized Particle Filter SLAM for Autonomous Underwater Vehicle With Slow Mechanical Scanning Imaging Sonar. *IEEE Syst. J.* **2019**. [CrossRef]
18. Mallios, A.; Ridao, P.; Ribas, D.; Carreras, M.; Camilli, R. Toward autonomous exploration in confined underwater environments. *J. Field Robot.* **2016**, *33*, 994–1012. [CrossRef]
19. Mallios, A.; Vidal, E.; Campos, R.; Carreras, M. Underwater caves sonar data set. *Int. J. Robot. Res.* **2017**, *36*, 1247–1251. [CrossRef]
20. Martins, A.; Almeida, J.; Almeida, C.; Dias, A.; Dias, N.; Aaltonen, J.; Heininen, A.; Koskinen, K.T.; Rossi, C.; Dominguez, S.; et al. UX 1 system design-A robotic system for underwater mining exploration. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 1494–1500.
21. Wang, J.; Shan, T.; Englot, B. Underwater Terrain Reconstruction from Forward-Looking Sonar Imagery. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 3471–3477.
22. Blanco, J.L. *A Tutorial on se (3) Transformation Parameterizations and On-Manifold Optimization*; Tech. Rep.; University of Malaga: Malaga, Spain, 2010; Volume 3.
23. Williams, C.; Rasmussen, C.E. Gaussian processes for machine learning. In *Summer School on Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006; Volume 2.
24. Matérn, B. *Spatial Variation*; Springer Science & Business Media: New York, NY, USA, 2013; Volume 36.
25. Smith, M.; Posner, I.; Newman, P. Generating implicit surfaces from lidar data. In Proceedings of the TAROS 2010, Plymouth, Devon, UK, 31 August–2 September 2010.
26. Deisenroth, M.; Rasmussen, C.E. PILCO: A model-based and data-efficient approach to policy search. In Proceedings of the 28th International Conference on Machine Learning (ICML-11), Washington, DC, USA, 28 June–2 July 2011; pp. 465–472.
27. Bijl, H.; Schön, T.B.; van Wingerden, J.W.; Verhaegen, M. Online Sparse Gaussian Process Training with Input Noise. Available online: http://user.it.uu.se/~thosc112/bijlsvwv21016sonig.pdf (accessed on 20 July 2020).
28. Johnson, J.E.; Laparra, V.; Camps-Valls, G. Accounting for input noise in Gaussian process parameter retrieval. *IEEE Geosci. Remote. Sens. Lett.* **2019**, *17*, 391–395.. [CrossRef]
29. Plagemann, C.; Mischke, S.; Prentice, S.; Kersting, K.; Roy, N.; Burgard, W. Learning predictive terrain models for legged robot locomotion. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 3545–3552.

30. Belta, C.; Kumar, V. Euclidean metrics for motion generation on SE (3). *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2002**, *216*, 47–60. [CrossRef]
31. Nurunnabi, A.; Sadahiro, Y.; Lindenbergh, R. Robust cylinder fitting in three-dimensional point cloud data. *Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2017**, *42*. [CrossRef]
32. Hauberg, S.; Feragen, A.; Enficiaud, R.; Black, M.J. Scalable robust principal component analysis using grassmann averages. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 2298–2311. [CrossRef] [PubMed]
33. Fitzgibbon, A.; Pilu, M.; Fisher, R.B. Direct least square fitting of ellipses. *IEEE Trans. Pattern Anal. Mach. Intell.* **1999**, *21*, 476–480. [CrossRef]
34. Gneiting, T.; others. Strictly and non-strictly positive definite functions on spheres. *Bernoulli* **2013**, *19*, 1327–1349. [CrossRef]
35. Padonou, E.; Roustant, O. Polar Gaussian processes and experimental designs in circular domains. *Siam/Asa J. Uncertain. Quantif.* **2016**, *4*, 1014–1033. [CrossRef]
36. Wendland, H. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Adv. Comput. Math.* **1995**, *4*, 389–396. [CrossRef]
37. Johnson, N.L.; Kotz, S.; Balakrishnan, N. *Continuous Univariate Distributions*; John Wiley & Sons, Ltd.: New York, NY, USA, 1994.
38. Fleury, P.; Ladouche, B.; Conroux, Y.; Jourde, H.; Dörfliger, N. Modelling the hydrologic functions of a karst aquifer under active water management–the Lez spring. *J. Hydrol.* **2009**, *365*, 235–243. [CrossRef]
39. Bicalho, C.C.; Batiot-Guilhe, C.; Seidel, J.; Van Exter, S.; Jourde, H. Geochemical evidence of water source characterization and hydrodynamic responses in a karst aquifer. *J. Hydrol.* **2012**, *450*, 206–218. [CrossRef]
40. Claraco, J.L.B. Development of scientific applications with the mobile robot programming toolkit. *The MRPT Reference Book. Machine Perception and Intelligent Robotics Laboratory*; University of Málaga: Málaga, Spain, 2008.
41. GPy. GPy: A Gaussian Process Framework in Python. Available online: http://github.com/SheffieldML/GPy (accessed on 20 July 2020).
42. Cully, A.; Chatzilygeroudis, K.; Allocati, F.; Mouret, J.B. Limbo: A flexible high-performance library for gaussian processes modeling and data-efficient optimization. *J. Open Source Softw.* **2018**, *3*, 545. [CrossRef]