



**HAL**  
open science

# Comparing Metrics for Evaluating 3D Reconstruction Quality in Large-Scale Environment Mapping

Stéphanie Aravecchia, Marianne Clausel, Cedric Pradalier

► **To cite this version:**

Stéphanie Aravecchia, Marianne Clausel, Cedric Pradalier. Comparing Metrics for Evaluating 3D Reconstruction Quality in Large-Scale Environment Mapping. 2023. hal-04128242v1

**HAL Id: hal-04128242**

**<https://hal.science/hal-04128242v1>**

Preprint submitted on 14 Jun 2023 (v1), last revised 8 Jan 2024 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Comparing Metrics for Evaluating 3D Reconstruction Quality in Large-Scale Environment Mapping

Stéphanie Aravecchia\*      Marianne Clausel†      Cédric Pradalier\*

June 14, 2023

## Abstract

**In this study, we focus on addressing the challenge of measuring the quality of 3D-reconstruction in large scale environments. Specifically, we consider scenarios where the map is built using a robot’s 3D-Lidar point cloud observations, with potential uncertainty in the robot localization. When considering a large scale environment that is unstructured, such as a park or a forest, another difficulty arises: the data becomes extremely sparse. As a result, measuring the quality of a reconstruction becomes even more challenging. This study aims to compare the effectiveness of various metrics in measuring the quality of 3D reconstruction. Firstly, we evaluate these metrics in a controlled experimental setup, where the reconstruction is created by progressively degrading the ground-truth using different degradation models. Secondly, we compare their ability to measure reconstruction quality at a local level, across various simulated environments, ranging from structured to unstructured. Finally, we conduct a qualitative comparison to demonstrate the robustness of certain metrics to noise in the robot localization. Ultimately, we synthesize the properties of these metrics and provide practical recommendations for their selection.**

---

\*are with IRL2958 Georgia Tech - CNRS

†is with University of Lorraine

## 1 Introduction

The motivation behind this study arises from the difficulties faced in evaluating the quality of 3D reconstruction in large-scale environments. Typically, the evaluation of 3D reconstruction quality aims to provide a single metric that reflects the overall quality of the reconstruction. Nonetheless, in the context of a robot autonomously mapping a large-scale environment for inspection or monitoring purposes, it becomes interesting to obtain a localized measure of map quality. Indeed, the map quality may not necessarily be homogeneous throughout a large-scale environment.

This study specifically focuses on this scenario, where the robot’s 3D-lidar observations construct the map. In this case, the prevailing map representation is the 3D grid, where each voxel encodes information. Traditionally, this 3D grid encodes the occupancy likelihood for each voxel. However, in this common scenario, the conventional measures of reconstruction quality, namely surface coverage and reconstruction accuracy, may not always hold significant meaning, especially when dealing with large-scale environments that are not only sparse but also unstructured. We will demonstrate it in this study, by emphasizing the specific case of mapping a large-scale, sparse, unstructured environment, such as a natural environment.

In the case of “unstructured” environments, distinct challenges arise compared to those encountered when mapping “structured” environments. The literature often focuses on large-scale 3D-reconstructions in “structured” environments like urban areas, as

demonstrated by extensive research conducted on the Kitti Dataset [1]. However, when it comes to work in natural environments, such as those explored in [2], [3], [4] or [5], the differentiation between structured and unstructured, dense and sparse environments becomes more prominent due to the unique challenges involved. The 3D-reconstruction in a large-scale natural environment is both unstructured and sparse. It consists predominantly of empty space, with only a few points where the 3D-lidar actually hits an object, further complicated by the increased noise level inherent to natural environments. For instance, trees, by their very structure, are sparse because they offer only an aggregation of small surfaces to the Lidar. Although the sampling of a trunk from a Lidar could be dense, the sampling of small branches or leaves, will generally not be high enough to recover the underlying structure. Moreover, from a laser’s perspective, trees can behave as semi-transparent structures, thereby inducing errors in the measurements. The reason is that when laser-rays – light cones in practice – reach a small object, like a branch, often only a portion of the energy is reflected, causing inaccurate distance readings. Two other causes of errors in laser measurements, that are particularly abundant in large-scale natural environments, are linked to the distance to the objects and the incidence angle: the error increases with each of them [6]. Finally, the localization of the robot in the map is also a source of errors because the 3D-reconstruction is a probabilistic accumulation of all the point-clouds transformed in the localization frame. This study does not directly evaluate various mapping algorithms applied to large-scale natural environments. Instead, our main emphasis lies in addressing the challenges associated with assessing the quality of 3D reconstruction in such environments.

In a large-scale environment, which could also be both sparse and unstructured, the question arises: How can we evaluate the local quality of the reconstruction? Are the conventional metrics capable of delivering meaningful measurements? This study specifically centers around evaluating various 3D-reconstruction metrics, including both conventional and less conventional ones, with a specific focus on assessing their effectiveness in accurately measuring

the reconstruction quality. First, we begin by selecting six relevant metrics: surface coverage, reconstruction accuracy, average Hausdorff Distance, Cohen’s Kappa coefficient, Kullback-Leibler Divergence, and Wasserstein Distance. Then, we introduce a methodology that enables the comparison of reconstruction to ground-truth at a local level. This is accomplished by extracting cuboid regions from both the ground-truth and the reconstruction, and by assessing the reconstruction quality with the previously mentioned metrics. Later on, we propose an additional methodology to evaluate the capability of the selected metrics in measuring various degradation models, when the 3D reconstruction is directly derived from the ground-truth. Finally, we empirically compare the metrics in situations where the 3D reconstruction is built from point clouds obtained from the robot’s observations. Ultimately, we present a comparison of the selected metrics, highlighting their properties, along with guidelines towards the choice of the metric depending on the application.

## 2 Related Workd

### 2.1 Map building

Building a map consists in building a representation of the perceived environment that aligns with the intended purpose of the map. In robotics, the maps are commonly built with the primary objective of optimal navigation planning, such as [7]. However, there are cases where the purpose is to reconstruct a scene to enable its monitoring, such as [8]. When the purpose of mapping is to reconstruct a scene, the most common maps are volumetric maps, or 3D-grids, and meshes, or 3D-surface maps. Meshes are generally obtained from either a Lidar point-cloud or from the point-cloud derived from visual odometry. In a mesh, the surface is described by connected triangles. An optimization algorithm is applied on the point-cloud to build the mesh. Among the most widely used techniques, we can cite Ball-Pivoting Algorithm [9], Poisson Surface Reconstruction [10] or Delaunay triangulations [11], mainly applied to mesh a single object. Other methods tackle large-scale meshing, such as [7]

who includes texture in the mesh to improve its visual aspect, or [8] who build a semantic mesh representation of a large-scale environment with the Las Vegas Reconstruction Toolkit [12]. Recently, other methods have been presented to build meshes with deep learning techniques, such as Voxel2Mesh [13].

In this paper, we focus on the prevailing map representation for outdoor robotics: volumetric maps. Such a map is the discretization of the space into voxels, each voxel containing some information, such as its occupancy likelihood. Several methods provide means to build 3D-grids from 3D-Lidar point-clouds. For instance, Voxblox [14] builds 3D grids where the truncated signed distance field (TSDF) is stored for each voxel. Although that method is developed to produce high quality maps, it tends to struggle when the environment size increases. Other methods have been proven efficient to build large 3D-grids, such as Octomap. Even though Octomap might not be the most efficient, as shown in [15], it is still one of the most widely used methods.

Octomap [16] is a method to build and store an octree instead of a 3D-grid, saving memory and computation. The open-source ROS Octomap library <sup>1</sup> implements the complete probabilistic map building process. The map is constructed in a given map frame, and every point-cloud in the Lidar frame updates the map. For each point in the point-cloud, a ray-casting operation is performed. Between the robot and the returned point, the probability of occupancy of the leaves in the octree is decreased. The returned points are updated as occupied, their probability of occupancy is increased. A thorough update process has been implemented to be robust to noise in the Lidar’s point-cloud. Moreover, the leaves in the octree encode the likelihood of occupancy, or emptiness, only if a point has been observed. Therefore, the octree encodes the unknown volume, i.e. the volume represented by absent leaves, a feature that could be useful in different robotics applications, such as exploration. We use Octomap to build the probabilistic volumetric map denoted as “reconstruction” in this study.

It is also important to note that the quality of

<sup>1</sup><http://wiki.ros.org/Octomap>

the map is intrinsically linked to the precision of the robot localization and that solving the localization in large-scale natural environment is still challenging. Since the point-clouds are expressed in the Lidar frame (which is based on the robot’s localization), any error in the transformation from the Lidar frame to the map frame will lead to errors in the map [4]. For those reasons, in this study, we evaluate the different metrics through the scope of the localization precision with different level of noise in simulation.

## 2.2 3D-reconstruction metrics

### 2.2.1 Classic 3D reconstruction metrics in robotics

Generally, when considering 3D-reconstruction, whether it be for robotics applications or not, measuring 3D reconstruction quality consists in comparing two surfaces. Often, those are the mesh generated from the 3D-point cloud, and the ground-truth mesh. Traditional metrics in that case are based on surface distance errors. They consist in calculating, for all surface points of one surface, the distance to the closest on the other surface, and then extract some statistics. Common surface distance errors are the Hausdorff Distance, that we will detail later, the Root Mean Square Error (RMSE) or the MAE (Mean Average Error) (used respectively in [17], [18], [19] for instance).

In the context of robotics, the prevailing metrics of reconstruction quality are the surface coverage and the reconstruction accuracy. Both are derived from those surface distance errors, and applied either on the meshes or on the 3D-grids. In the latter case, two sets of points are compared. Provided we want to compute the surface coverage, we are interested on the proportion of the set of points from the ground-truth accurately reconstructed. To do so, if the distance between a ground-truth point and its closest reconstructed point is less than a registration distance, the ground-truth point is considered as reconstructed (i.e valid). The metric is the proportion of such points ([20, 21]). Similarly, the reconstruction accuracy corresponds to the proportion of accurately reconstructed points in the set of points from the re-

construction, that is, points whose distance to the closest ground-truth point is below a registration distance.

### 2.2.2 Other reconstruction metrics

Some metrics have been proved efficient when it comes to evaluate the quality of the semantic segmentation predicted by deep-learning models, although most of them have been used outside the field of computer vision for decades. The idea with those metrics is to evaluate at the same time the classification accuracy and the correctness of the localization. Among them, we can cite the accuracy, the precision, the recall, or even the Intersection Over Union (IoU, Jaccard Index), or the F1 score. [22] provides a thorough review and comparison of the existing segmentation metrics for 3D-medical image segmentation tasks. Building upon [22], [23] compares some of those metrics depending on the size of the regions of interest to segment. As [22], building on the fact that we are working on 3D-grids, we can consider our problem a 3D-segmentation problem. We can then consider each voxel is assigned the class "empty" or "occupied" based on its occupancy likelihood. In large-scale environments, the volume is mostly empty space, with sparse objects whose contours represent the occupied space. [23] shows that two metrics are particularly sensitive when it comes to measuring segmentation quality of small objects in an image: the Average Hausdorff Distance and Cohen's Kappa coefficient.

The Hausdorff Distance (HD) is, as the other metrics seen before, a spatial distance metric, widely used to evaluate 3D-reconstruction [17, 24]. HD is a common measure of distance between two point sets, but it is sensitive to outliers. [22] proposes to use instead the Average Hausdorff Distance (*AHD*), introduced in [25]. The *AHD* averages the HD over all the points, becoming more stable and less sensitive to outliers than HD.

The other interesting metric pointed out by [23], is the Cohen's Kappa coefficient (*KAP*). Unlike the metrics seen previously, *KAP* is not a spatial distance metric but a probabilistic metric. It was first proposed in [26]: it provides a score measuring the agree-

ment between two samples. As an advantage over other measures, *KAP* takes into account the agreement caused by chance, which makes it more robust. That is, *KAP* is in  $[-1,1]$ , where 1 corresponds to complete agreement, -1 to complete opposition, and 0 to random.

### 2.3 Comparing probabilities

Since we are building a probabilistic volumetric map with Octomap, we could take advantage of that framework to measure the quality of the map. Each voxel in the probabilistic map has a probability of occupancy between 0%, the absolute certainty that the voxel is empty, and 100%, the absolute certainty that the voxel is occupied. Leveraging the probabilities inside a 3D-grid is not a novelty, and has been explored in [20]. However, they do not propose a mean to compare the reconstructed volume to a reference one. They calculate the entropy of the voxels, which represents their distance to the unknown, thereby indicating the quantity of observation for each voxel, but not a measure of the reconstruction quality.

In this paper, we propose a methodology to compare two volumetric maps with probabilistic values. Different methods allow comparing probabilities. The most common is probably the Kullback-Leibler Divergence (*DKL*). The *DKL* [27] is a measure of how different a probability distribution is from another probability distribution. With the *DKL*, we can measure how the probability distribution of the reconstruction is different from the probability distribution of the ground-truth.

Nonetheless, since we are considering a grid of probabilistic voxels, we have access to another information: the Euclidean distance between voxels. As an example, if a point is erroneously reconstructed 5 cm away from an actual object, the reconstruction is better than if the erroneous point is 50 cm away. The *DKL* is not sensitive to this difference. An alternative solution is then to find the Optimal Transport plan, linking one probability distribution to another one, with a cost function depending on the geometry [28]. From this Optimal Transport plan, we can calculate a distance, the Wasserstein Distance which is a generalization of the concept of Earth Mover's

Distance (EMD). Computing the Optimal Transport Plan may be cumbersome, because it is an optimization problem that is not necessarily convex. To bypass computational issues, [29] regularizes the optimal transport problem by adding an entropic term, and solves it using a Sinkhorn’s fixed point iteration. [30] provides an open source Python library implementing several solvers for Optimal Transport problems, including [29]’s algorithm. With this regularized Wasserstein Distance, we can measure the quality of the 3D-reconstruction, comparing not only the ground-truth and reconstructed values of probabilistic maps but also taking into account the Euclidean distances in the errors.

### 3 Comparison Methodology

The objective of this study is to evaluate various 3D-reconstruction metrics with the expectation that these metrics will provide a meaningful measure of the reconstruction quality at a local level. Firstly, we present our methodology for extracting cuboid regions that enable the comparison of 3D-reconstruction to ground truth. Secondly, we describe the computation process for each considered 3D-reconstruction metric within these cuboid regions. Thirdly, we outline our methodology for evaluating the effectiveness of these metrics in measuring the reconstruction quality. This evaluation is conducted using controlled 3D-reconstruction obtained through the controlled degradation of the ground truth. All the code is available open-source<sup>2</sup>.

#### 3.1 Cuboid region extraction

To enable the comparison of the local reconstruction and the ground-truth, we extract cuboids regions from both the reconstruction and the ground-truth. To do so, we discretize both volumes into two 3D-grids, of the same resolution  $RES$  and in the same reference frame  $R_f$ . A local region is then a cuboid  $C$ . To compare the two local regions, we compute

the metrics described in Sec. 3.2 between intersecting cuboids. Alg. 1 summarizes this section.

The ground-truth is the mesh of the scene used in simulation, the 3D-reconstruction is the full probability map constructed by Octomap from the robot’s observations.

For the ground-truth, we first slice the ground-truth mesh with horizontal planes, with a vertical space of  $res$ , our spatial resolution (with  $res < RES$ ). In each plane, we then calculate the intersection of the mesh and the plane, and we store it in a 3D matrix of voxel size  $res$ . Each voxel on the intersection is occupied and has a value of 1.0, all the remaining voxels are empty with a value of 0.0. Our ground-truth dataset,  $\mathcal{D}_{gt}$ , is then a 3D matrix of size  $(h_1, w_1, n_1)$ . Its associated volume in space, in the reference frame  $R_f$ , is the bounding-box  $B_{box_1}$ .

For the reconstruction, we first create a full probability map with Octomap in the same reference frame  $R_f$ , and with the same resolution  $res$ . From this octree, and its bounding-box in  $R_f$ , we initialize a 3D matrix as unknown space, i.e. values of 0.5, corresponding to the equal probability of the voxel to be occupied or empty. We then iterate on all the leaves in the octree. For each leaf, we set the probability of its associated voxels in the 3D matrix to the probability of the leaf (the unknown space is implicitly described in Octomap with absent leaves). We finally obtain the reconstruction dataset,  $\mathcal{D}_{rec}$ , a 3D matrix of size  $(h_2, w_2, n_2)$ , with a voxel resolution  $res$ , with its associated bounding-box,  $B_{box_2}$ , in the same reference frame,  $R_f$ .

Finally, we do the comparison in  $B_{box_1} \cap B_{box_2}$ . First, we load the intersection of both datasets in two 3D matrices,  $M_{gt}$  and  $M_{rec}$ : a voxel  $v_{gt}^{ijk}$  from  $M_{gt}$  corresponds to the same volume in  $R_f$  than  $v_{rec}^{ijk}$  from  $M_{rec}$ . Second, we go through both 3D matrices, and compare, cuboid region by cuboid region, the reconstruction and the ground truth. A cuboid region is a group of  $n \times n \times n$  voxels in each 3D matrix, and is noted  $C_{gt}$  or  $C_{rec}$ . Those cuboid regions are in fact large voxels of size  $RES = n \times res$ , in the 3D-grid  $B_{box_1} \cap B_{box_2}$ , in  $R_f$ , and each element in  $C_{rec}$  and  $C_{gt}$  stores the occupancy likelihood of its corresponding voxel of size  $n$  in  $R_f$ . Fig. 1 shows an

<sup>2</sup><https://github.com/stephanie-aravecchia/3d-reconstruction-metrics>

example of a cuboid where  $n = 10$ .

### 3.2 Comparison metrics

This section explains how, for each cuboid region, we compute different metrics to indicate the quality of the reconstruction.

Since this method is developed with the objective to work also on large-scale environments with sparse objects, the reconstructed volume may contain more empty space than actual objects to reconstruct. A measure that would give information on occupied space only may not be representative of the complete volume. For this reason, we found it interesting to measure not only how well objects have been reconstructed, but also how well the empty space has been reconstructed. To do so, we first define two sets:  $\mathcal{U}_{occ}$ , the set of the cuboids regions containing at least one occupied voxel in  $C_{gt}$  and  $\mathcal{U}_{empty}$ , its complement. Then, we measure the reconstruction quality of the cuboids with a different metric in each set: one of the considered reconstruction metrics for the cuboids in  $\mathcal{U}_{occ}$ , the  $L_1$  norm for the cuboids in  $\mathcal{U}_{empty}$ . This study focuses on the evaluation of metrics in  $\mathcal{U}_{occ}$ .

Furthermore, because measuring reconstruction quality of unknown space is pointless, we consider a threshold before calculating our metrics. If all the probabilities in  $C_{rec}$  are close to the unknown ( $0.5 \pm 0.1$ ), we do not calculate the metrics, but set them to default values.

#### 3.2.1 Surface Distance Metrics

The three surface distance metrics we consider here (surface coverage, reconstruction accuracy and Average Hausdorff Distance) are based on spatial distances between sets of points. To compute them, we define the following constants:

- $\hat{p}$  is an occupancy likelihood threshold,
- $d_r$  is a registration distance.

We also define the following variables:

- $\mathcal{U}_{rec}$  is the set of points  $P$  from  $C_{rec}$  whose occupancy likelihood is above  $\hat{p}$ ,

- $\mathcal{U}_{gt}$  is the set of occupied points from  $C_{gt}$ ,
- $n_{rec}$  is the number of points in  $\mathcal{U}_{rec}$ ,
- $n_{gt}$  is the number of points in  $\mathcal{U}_{gt}$ ,
- $\|PS\|$  is the Euclidean Distance between a point  $P$  from  $\mathcal{U}_{rec}$  to the closest  $S$  in  $\mathcal{U}_{gt}$ ,
- $\|SP\|$  is the Euclidean Distance between a point  $S$  from  $\mathcal{U}_{gt}$  to the closest  $P$  in  $\mathcal{U}_{rec}$ .

As an example, in Fig. 1, the points from  $\mathcal{U}_{gt}$  and  $\mathcal{U}_{rec}$  correspond respectively to the white voxels in  $C_{gt}$  and  $\hat{C}_{rec}$ .

#### 3.2.2 Surface Coverage

To compute  $COV$ , we apply the classical methodology in a 3D-grid, as [20], to our cuboids. We first compute the number of reconstructed points,  $k_{rec}$ , that is, points in  $\mathcal{U}_{gt}$  we consider correctly reconstructed, and then we compute the surface coverage,  $COV$ , the proportion of correctly reconstructed points.

$$k_{rec} = \sum_{S \in \mathcal{U}_{gt}} \mathbf{1}_B(\|SP\| \leq d_r) \quad (1)$$

where  $\mathbf{1}_B(b) = 1$  if  $b$ , 0 otherwise.

$$COV = k_{rec}/n_{gt} \quad (2)$$

#### 3.2.3 Reconstruction Accuracy

To compute  $ACC$ , we proceed similarly: we first compute the number of accurate points,  $k_{acc}$ , that is, points in  $\mathcal{U}_{rec}$  we considered valid, and then we compute the reconstruction accuracy,  $ACC$ , the proportion of valid points.

$$k_{acc} = \sum_{P \in \mathcal{U}_{rec}} \mathbf{1}_B(\|PS\| \leq d_r) \quad (3)$$

$$ACC = k_{acc}/n_{rec} \quad (4)$$

#### 3.2.4 Mean Average Distance

To compute  $AHD$ , we follow [22]. The Hausdorff Distance measures the distance between two sets of points. We compute two Hausdorff Distances: the

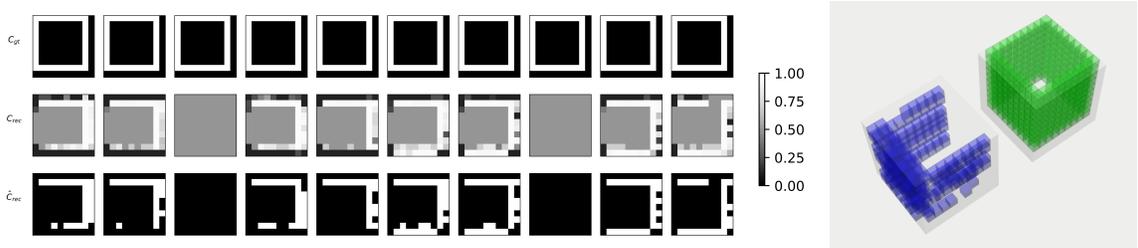


Figure 1: Visualization of a cuboid (10x10x10 voxels). On the left part of the figure, each row of images correspond to a single cuboid, each column in the row to a slice of the cuboid. The color encodes the occupancy likelihood, from free space in black, to occupied space in white, as shown in the colorbar. The grey corresponds to the unknown. The first row is the ground-truth cuboid,  $C_{gt}$ . The second row is the reconstruction cuboid,  $C_{rec}$ , encoding the occupancy likelihood. The last row is the binary version of  $C_{rec}$ :  $\hat{C}_{rec}$ , where the voxels whose occupancy likelihood is above 0.8 are set to occupied, the others to empty.  $DKL$  and  $WD_{occ}$  are computed directly  $C_{rec}$ , whereas  $COV$ ,  $ACC$ ,  $AHD$  and  $KAP$  are computed on  $\hat{C}_{rec}$ . The right part of the figure displays in green  $C_{gt}$ , in blue the thresholded  $\hat{C}_{rec}$ .

distance from reconstruction to ground-truth  $d_{PS}$ , and the distance from ground-truth to reconstruction  $d_{SP}$ :

$$d_{PS} = \frac{1}{n_p} \sum_{P \in \mathcal{U}_{rec}} \|PS\| \quad (5)$$

$$d_{SP} = \frac{1}{n_s} \sum_{S \in \mathcal{U}_{rec}} \|SP\| \quad (6)$$

Then, we compute the Average Hausdorff Distance, which consists in the maximum between the two distances:

$$AHD = \max(d_{PS}, d_{SP}) \quad (7)$$

### 3.2.5 Cohen's Kappa

The metric Cohen's Kappa,  $KAP$ , provides a score in  $[-1, 1]$  ([worst, best]). This score provides a measure of agreement between two sets of classification. Unlike surface distance metrics seen before, we compare here directly the elements of  $C_{rec}$  and  $C_{gt}$ . In the cuboids, we consider our problem as a binary classification problem: each voxel is assigned either the class occupied or empty. In  $C_{gt}$ , each voxel already has a value 0 (class empty) or 1 (class occupied). In  $C_{rec}$ , we consider a voxel occupied if its occupancy

likelihood  $p > \hat{p}$ , else, we consider it empty and call the resulting binary cuboid  $\hat{C}_{rec}$ . Fig. 1 provides an example. We iterate on all the voxels of  $\hat{C}_{rec}$ , compare them to their corresponding voxel in  $C_{gt}$ , and we count  $FP$  (occurrence of false positive),  $FN$  (false negative),  $TP$  (true positive),  $TN$  (true negative).

Then, to compute  $KAP$ , we follow [22]. Let  $N$  be the number of voxels in a cuboid:

$$f_c = \frac{(TN + FN)(TN + FP) + (FP + TP)(FN + TP)}{N} \quad (8)$$

$$KAP = \frac{(TP + TN) - f_c}{N - f_c} \quad (9)$$

### 3.2.6 Kullback-Leibler Divergence

The DKL provides a measure of how a probability distribution is different from a reference probability distribution. The  $DKL$  metric we use in this work is the sum of the DKL between the probability distributions derived from the elements of the cuboids. Let  $p^0$  be the occupancy likelihood of a voxel in  $C_{rec}$ ,  $g^0$  the occupancy likelihood of the same voxel in  $C_{gt}$ . For numerical reasons, we saturate  $p^0$  and  $g^0$  in  $[m, 1-m]$ , where  $m$  is a small number. The saturated values are  $p$  and  $g$ . Then, we iterate on the  $N = n \times n \times n$  ele-

ments of the cuboids, and we compute the *DKL* metric as follows:

$$DKL = \sum_{k=1}^{k=N} \left[ (1 - p_k) \cdot \log \frac{(1 - p_k)}{(1 - g_k)} + p_k \cdot \log \frac{p_k}{g_k} \right] \quad (10)$$

### 3.2.7 Wasserstein Distance

The Wasserstein Distance is derived from the optimal transport plan to “move” the mass distribution from a query vector to match the mass distribution of a reference vector. The cost of moving the mass being a function of the Euclidean distance it has to be moved by. Here, we calculate the Wasserstein Distance between two cuboid regions.

As this Wasserstein Distance is defined on histograms, that is, vectors that sum to 1, we first need to remap all the elements of  $C_{gt}$  and  $C_{rec}$  into two vectors of doubles,  $V_{gt}$  and  $V_{rec}$ , such that  $\forall(i, j, k) \in 0, n, V_*(i \cdot n^2 + j \cdot n + k) = C_*(i, j, k)$ . Second, we derive from each vector, two different vectors. From  $V_{gt}$ , we derive  $V_{gt}^{occ} = \max(2V_{gt} - 1, 0)$  and  $V_{gt}^{free} = \max(1 - 2V_{gt}, 0)$ . They contain respectively the probability of an element to correspond to an occupied voxel, and the probability of an element to correspond to an empty voxel. We then normalize each vector by their sum and obtain two histograms  $P_{gt}^{occ}$  and  $P_{gt}^{free}$ . Similarly, from  $V_{rec}$ , we obtain  $P_{rec}^{occ}$  and  $P_{rec}^{free}$ . We do this partition between occupancy and emptiness because we observed that the Wasserstein Distance between  $P_{rec}^{occ}$  and  $P_{gt}^{occ}$ , which embeds in each element the distance from its probability of occupancy to the unknown, contains more signal. Moreover, this corresponds better to what we intend to measure with this metric, that is how well the occupied space has been reconstructed. Using the same mapping between voxels and elements of the vector, we set the cost matrix  $M$  to contain the squared Euclidean distance between the voxels associated to the elements of the vector.

Finally, we calculate  $WD_{occ}$  the regularized Wasserstein Distance between  $P_{rec}^{occ}$  and  $P_{gt}^{occ}$ , computed using the Sinkhorn algorithm described in [29], following the implementation of [30]. This algorithm

is an optimization that seeks an optimal coupling which minimizes the displacement cost of a discrete measure,  $P_{rec}^{occ}$  in our case, to a discrete measure,  $P_{gt}^{occ}$ , with respect to a cost, a transport matrix,  $M$ , under an entropic constraint. The optimal value of the optimal transport problem is the Wasserstein Distance, defined as follows, with  $\gamma \mathbf{1} = P_{rec}^{occ}$ ,  $\gamma^T \mathbf{1} = P_{gt}^{occ}$  and  $\gamma \geq 0$ :

$$WD_{occ} = \min_{\gamma} \langle \gamma, M \rangle + \alpha \cdot \Omega(\gamma) \quad (11)$$

Where  $M$  is the previously defined cost matrix,  $\Omega$  is the entropic regularization term:  $\Omega(\gamma) = \sum_{i,j} \gamma_{i,j} \log(\gamma_{i,j})$ ,  $\alpha$  is an entropic regularization factor and  $\langle \cdot, \cdot \rangle$  is the Frobenius dot-product. In this study, we set the regularization factor  $\alpha = 1.0$ .

### 3.2.8 $L_1$ norm

In a cuboid region of the ground-truth in  $\mathcal{U}_{empty}$ , all the voxels have a probability of occupancy of 0. Therefore, when comparing the reconstruction to the ground-truth, we are comparing a vector containing some probabilities of occupancy  $V_{rec}$  to a vector of the same size containing only zeros (absolute certainty of emptiness). Such a measure is given by the  $L_1$  norm of  $V_{rec}$ : the distance between  $V_{rec}$  and the vector of zeros that represent the empty space. We do not evaluate  $L_1$  in this study, but simply provide it to the reader, because we found it convenient in other works.

## 3.3 Evaluation Methodology

### 3.3.1 Controlled 3D-reconstruction

This section describes the evaluation of the metrics, with a specific focus on examining their behavior in the context of decreasing reconstruction quality. We present the methodology using a single ground-truth cuboid. The key approach involves initially measuring the reconstruction quality when it is perfect, indicated by  $C_{rec} = C_{gt}$ . Then, we introduce various degradation models that are iteratively applied to the reconstructed cuboid, progressively degrading the quality of the reconstruction. Let us assume  $v$  is a voxel randomly sampled in the cuboid  $C_{rec}$ ,  $p(v)$

---

**Algorithm 1** Reconstruction and ground-truth comparison
 

---

```

//  $\mathcal{D}_{gt}$  and  $\mathcal{D}_{rec}$  are the datasets
 $\mathcal{D}_{gt}(B_{\text{box}1}, (h_1, w_1, n_1), R_f)$ ,  $\mathcal{D}_{rec}(B_{\text{box}2}, (h_2, w_2, n_2), R_f)$ 
 $B_{\text{box}} \leftarrow B_{\text{box}1} \cap B_{\text{box}2}$ 
 $M_{gt} \leftarrow \mathcal{D}_{gt}(B_{\text{box}})$ ,  $M_{rec} \leftarrow \mathcal{D}_{rec}(B_{\text{box}})$ 
for all cuboid  $\in B_{\text{box}}$  do:
   $C_{gt} \leftarrow M_{gt}(\text{cuboid})$ ,  $C_{rec} \leftarrow M_{rec}(\text{cuboid})$ 
  if isObserved( $C_{rec}$ ) then:
    if cuboid  $\in \mathcal{U}_{occ}$  then:
      cuboid.metrics  $\leftarrow$  computeMetrics( $C_{rec}, C_{gt}$ )
    else:
      cuboid.metrics  $\leftarrow$  computeL1( $C_{rec}$ )
    end if
  else:
    cuboid.metrics  $\leftarrow$  maxMetrics
  end if
end for

```

---

the occupancy likelihood of  $v$ , and  $v'$  a direct neighbor of  $v$ . The degradation models considered in this study are:

1. random\_occ:  $p(v) = 1$ ;
2. random\_free:  $p(v) = 0$ ;
3. random\_unknown:  $p(v) = 0.5$ ;
4. random\_random:  $p(v) \sim [0, 1]$
5. random\_shift:  $p(v') = p(v)$ ,  $p(v) = p(v')$
6. random\_flip:  $p(v) = 1 - p(v)$

### 3.3.2 Metric Evaluation and Comparison

We evaluate the different metrics with a consistent perspective, focusing on their ability to discriminate reliably “good” from “less good” reconstructions. Before going further, it is important to note that some metrics are distance metrics (the lower, the better):  $DKL$ ,  $AHD$ ,  $WD_{occ}$ , whereas the others are score metrics (the higher, the better):  $COV$ ,  $ACC$ ,  $KAP$ . First, we define the threshold  $\hat{n}$  that divides the population of  $C_{rec}$  in two, the “good” ones, and the “less good” ones:  $\hat{n}$  is simply a fixed level of degradation of the cuboid (i.e 20%). Second, we define the threshold  $\hat{\theta}$  that divides the population of  $C_{rec}$  in two: the cuboids measured as “good” and those “less good”. In the case of distance metrics, a measure  $\theta$  smaller than  $\hat{\theta}$  indicates a “good” measured reconstruction (conversely for score metrics). Finally, we consider

our problem as a classification problem, and we populate a confusion matrix by evaluating all the cuboids, at all the iterations as explained in Tab.1. It should be noted that the definition of true / false positive / negative described here are different from the quantities used to compute Cohen’s Kappa, where the classification was made on the voxels.

	<i>score metric</i>	<i>distance metric</i>
true positive	$(n \leq \hat{n}) \ \& \ (\theta > \hat{\theta})$	$(n \leq \hat{n}) \ \& \ (\theta \leq \hat{\theta})$
true negative	$(n > \hat{n}) \ \& \ (\theta \leq \hat{\theta})$	$(n > \hat{n}) \ \& \ (\theta > \hat{\theta})$
false positive	$(n > \hat{n}) \ \& \ (\theta > \hat{\theta})$	$(n > \hat{n}) \ \& \ (\theta \leq \hat{\theta})$
false negative	$(n \leq \hat{n}) \ \& \ (\theta \leq \hat{\theta})$	$(n \leq \hat{n}) \ \& \ (\theta > \hat{\theta})$

Table 1: Condition tested on a cuboid to populate the Confusion Matrix

From the confusion matrix, where we count the occurrence of true positive  $TP$ , true negative  $TN$ , false positive  $FP$ , and false negative  $FN$ , we compute the precision and recall of the metric:

$$precision = TP / (TP + FP) \quad (12)$$

$$recall = TP / (TP + FN) \quad (13)$$

We repeat the process for different thresholds  $\hat{\theta}$  for each metric. From this data, we are able to compare the precision recall curves for the different metrics.

## 4 Experimental Setting

This section explains how we generate the simulated worlds to conduct the metric evaluation with the controlled 3D-reconstruction (3.3.1). Then, with the same worlds, we conduct experiments, in simulation. The simulations are run within the ROS framework with Gazebo and a Clearpath Husky robot equipped with a 3D Lidar Ouster OS1-16 (16 planes of 512 points).

### 4.1 World Generation

We generate randomly several environments. Each environment is a plane of dimension  $60m \times 60m$  on which we place assets with a Poisson Cluster Point Process, to reproduce the natural spatial distribution

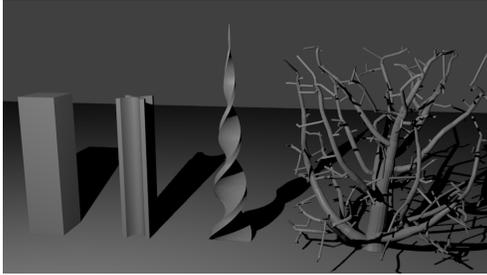


Figure 2: Illustration of the different assets used in simulation: rectangular cuboid, cross-extruded shape, helicoidal cone, simulated tree.

of trees. The assets (illustrated in Fig. 2) can be, with an increasing level of difficulty with respect to the reconstruction, either rectangular cuboids, cross-extruded shapes, helicoidal cones, or randomly selected in our 15-trees library mimicking winter bare trees. Those trees are created with a space colonization algorithm<sup>3</sup>. Due to computational considerations in Gazebo, we create only trees without leaves. Random factors are applied on the assets’ dimensions and orientation.

With the same point process generation, we create four different synthetic environments, one with each type of assets. With the open-source Blender software<sup>4</sup>, we create a single shapefile (.stl) for each environment. This ensures that the mesh we slice to obtain the 3D-grid ground-truth is the same used in Gazebo to infer the collisions of the Lidar, hence to construct the map.

We generate 12 environments (3 different spatial distributions with 4 different types of asset).

From those environments, we either build 3D-reconstructions from the robot’s observations in simulation, as explained thereafter (Sec. 4.2), or we sample randomly 1500 cuboids to apply the controlled 3D-reconstruction evaluation (Sec. 3.3.1).

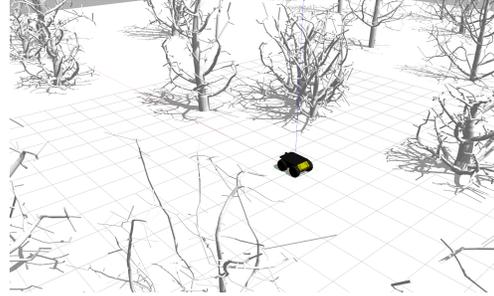


Figure 3: Illustration of an experiment, with the Husky robot at the center in an environment containing only trees.

## 4.2 Experimental Simulation

An experiment, illustrated in Fig.3, consists in loading one of the environments described previously in Gazebo, teleporting the robot randomly in the environment, accumulating the point-cloud, and teleporting it again, for 100 iterations. A mixture of noise is applied to the Lidar to simulate better the behavior of a Lidar in outdoor natural environments. Since the quality of the reconstruction is directly linked to the localization of the robot, we incrementally add noise in this localization. For each noise level, we build a probabilistic map with Octomap. To obtain the different noise levels, we use the perfect localization from Gazebo on which we apply a Gaussian noise on the position and on the orientation. The noise levels considered in this study are (with the standard deviations  $\sigma_p$ , on position, in meter,  $\sigma_q$ , on orientation, in radian): 0. perfect localization; 1.  $\sigma_p = 0.005$ ,  $\sigma_q = 0.005$ ; 2.  $\sigma_p = 0.05$ ,  $\sigma_q = 0.01$ .

Errors in the orientation estimation lead to large errors in the position of far away points (i.e a 0.01 radian error in the orientation leads to a 0.20m error in the position of a 20m distant point in the Lidar point-cloud). In this study, the sensor range is set to 20m, the map resolution to  $res = 0.1m$ , and the cuboid resolution to  $RES = 1m$ .

For each of the 12 environments, for each of the 3 noise levels, we run 28 simulations, for a total of 1008

<sup>3</sup><https://github.com/dsforza96/tree-gen>

<sup>4</sup><https://www.blender.org/>

experiments in simulation.

## 5 Results

### 5.1 Theoretical Metrics Comparison

We compare the metrics when the reconstruction moves further from the ground-truth, following the methodology in Sec. 3.3.1, and we distinguish the experiments by type of world.

As described in Sec. 3.2, some metrics depends on constants (occupancy likelihood threshold, registration distance). In this study, we run the experiments with the following values, but display the results only for the values in bold, because even though the value of the metric is slightly influenced by the value of the constants, the general behavior remains the same:

- $\hat{p}$ : 0.7, **0.8** (*COV*, *ACC*, *KAP*, *AHD*),
- $d_r$ : **0.05**, 0.1, 0.15 meters (*COV*, *ACC*).

Fig. 4 displays the results for the 1500 sampled cuboids from Sec. 4.2. The figure display one line per metric, one column per type of degradation applied to the ground-truth, and one color per type of world (that is, per type of asset in the world). Keeping in mind that *DKL*, *AHD* and *WD<sub>occ</sub>* are distance metrics, whereas *COV*, *ACC* and *KAP* are score metrics, we can see on this graph that the different metrics we consider are differently sensitive to the type of transformations we apply on the ground-truth,

A "good" metric is expected to: be sensitive to all the types of transformations, vary monotonically when the reconstruction moves further from the ground-truth, be independent of the type of world, and give measurements in a range that does not depend too drastically on the transformations. Fig. 4 shows that no metric satisfies all those conditions.

From each graph in Fig. 4 corresponding to a couple (metric, type of degradation), we can distinguish three tendencies, depending on how  $\theta$  vary when  $n$  increases.

- no variation, the curve is flat: the metric is not sensitive to the type of degradation;
- a small variation (gentle slope): the metric is slightly sensitive to the type of degradation;

- a huge variation: the metric is highly sensitive to the type of degradation.

*DKL*, *WD<sub>occ</sub>*, *AHD*, *ACC* and *KAP* are highly sensitive to additional points (random occ). The high sensitivity to random points (random random) and random flip may only be linked to the sensitivity to additional points, since a random point may have a high occupancy likelihood, and flip tends to add new points in the cuboids we consider (containing more empty voxels than occupied voxels). Unlike the other four, *KAP* seems to remain slightly sensitive to the other types of degradation. Not surprisingly, *COV* and *ACC* are totally insensitive to some transformations: *COV* is not sensitive to additional reconstructed points, as long as all the ground-truth points are covered. Similarly, *ACC* is not sensitive to missing points in the ground-truth, as long as all the reconstructed points are indeed accurate. Similarly, *WD<sub>occ</sub>* is not sensitive to missing points, or to diffusion (random shift). *AHD* is less sensitive to missing points or diffusion, than to other types of transformations, but still remains a bit sensitive.

Additionally, we can see in Fig. 4 that the metrics are generally dependent on the type of world. The areas corresponding to 80% of the measures in TREES environments (red) are larger and the medians are different from the areas corresponding to 80% of RECT environments for instance (blue). That tends to indicate that not only the measure provided by the metrics are noisier in challenging environments, but also the very value supposedly dependent only on the reconstruction quality also depends highly on the type of environment.

In real world applications, the noise model is probably a combination of all the degradation models considered here, and a central question is: is there a threshold to discriminate reliably "good" and "less good" reconstructions for the chosen metric ?

Fig. 5 shows the very challenge of setting thresholds to discriminate "good" and "less good" reconstructions. This figure show the precision-recall curves obtained following Sec. 3.3.2, where the threshold  $\hat{\theta}$  for each metric vary in a range corresponding to the min and max values of the y-axis of Fig.4 for the considered metric. For each metric, the

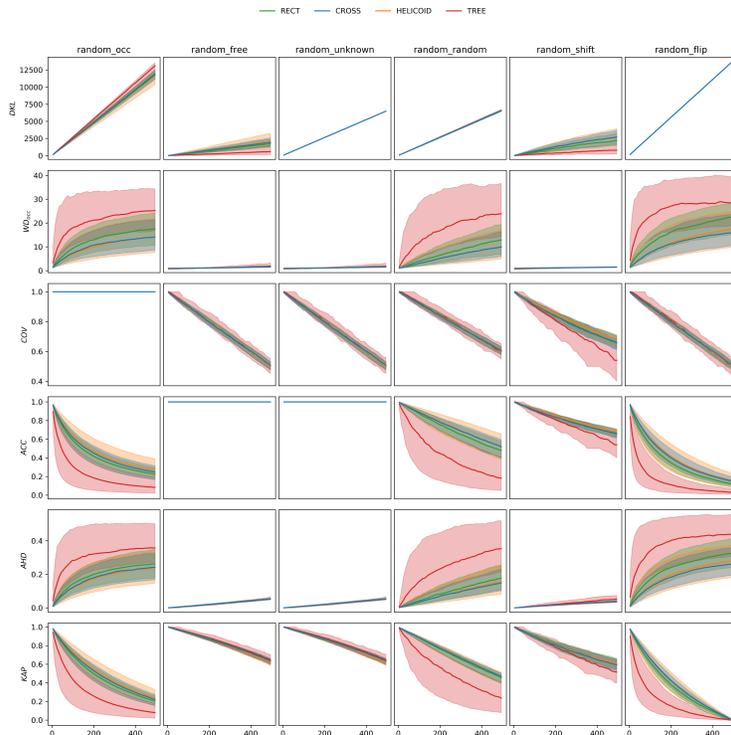


Figure 4: Illustration of the different metrics behavior when the reconstruction moves further from the ground-truth. One line per metric, one column per type of degradation applied to the gt. For each sub-figure, the x-axis is the number of iteration  $n$ , the y-axis the value of the metric  $\theta$ . The results are grouped by type of asset in the world. In each sub-figure, the line corresponds to the median value from all cuboids in the type of world and the filled area shows the spread of 80% of the population.

curves are drawn with 10 values of  $\hat{\theta}$ , and we highlight three particular values with the circle, diamond and star markers. Fig. 5 shows that, with the same  $\hat{\theta}$  (one of the markers), a metric can be very good when it comes to some degradation models, but very bad at others. This figure also shows that, even though with a different threshold, *AHD* and *KAP* seem able to deal with all the degradation models, whereas the other metrics always struggle with at least one degradation model.

## 5.2 Metrics Comparison on experimental reconstructions

In this section, we compare the metrics when the map is built from the robot’s observation.

Fig. 6 shows the distribution of the different metrics per type of world. We consider only the cuboids in  $\mathcal{U}_{occ}$ , and we display also the distribution of  $n_{gt}$  and  $n_{rec}$ , the occurrence of occupied voxels in the ground-truth and reconstruction cuboids  $C_{gt}$  and  $C_{rec}$  (Sec. 3.2). The general trend of the histograms remain the same when we split the population of cuboids per noise level in the localization. Therefore, we present only here the synthesis. In the large-scale environments populated with sparse objects we con-

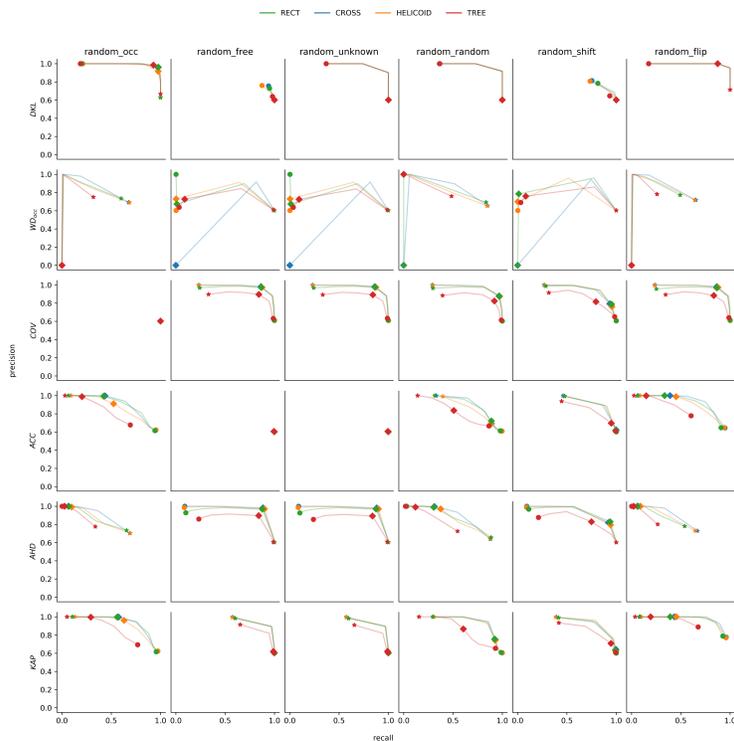


Figure 5: Precision-Recall curves when we vary the value of the threshold  $\hat{\theta}$  for each metric. One line per metric, one column per type of degradation, one color per type of world. The three markers display precision-recall points for three values of  $\hat{\theta}$  for each metric. The results are displayed for a level of degradation of the cuboid of 20%. The points in (0,0) corresponds to points where it is not possible to compute precision and recall (division by 0 in Eq. 12 and 13)

sider in this study, not only  $n_{gt}$  is generally small, but  $n_{rec}$  is even smaller. The histograms embed also in blue a particular case: when  $n_{rec} = 0$ , we denote such cuboids  $C_{rec}^0$ . Those cuboids, where not a single point have been reconstructed, are likely to be “not good” reconstructions. Fig. 6 also shows that not only the choice of the metric becomes difficult when the world becomes more challenging with respect to the reconstruction, but also that it is not straightforward to set a meaningful threshold above (or below) which the reconstruction is deemed good. The graphs are arranged from top to down in increasing level of difficulty in the reconstruction, from RECT (basic geometrical shape), to TREES (unstructured

objects).

Fig. 6 shows two trends:

- The metric is likely to be noisy: the cuboids of  $C_{rec}^0$  are spread on all the range of values.
- The range of the values the metric provides shrinks when the complexity of the world increases.

From those two trends, we can hypothesize that  $DKL$  and  $WD_{occ}$  are likely to be noisy. We can also hypothesize that  $AHD$  is the most capable of providing measures when the difficulty in the world increases: the proportion of cuboids where the value is not in the first bar is the largest. Additionally,  $AHD$

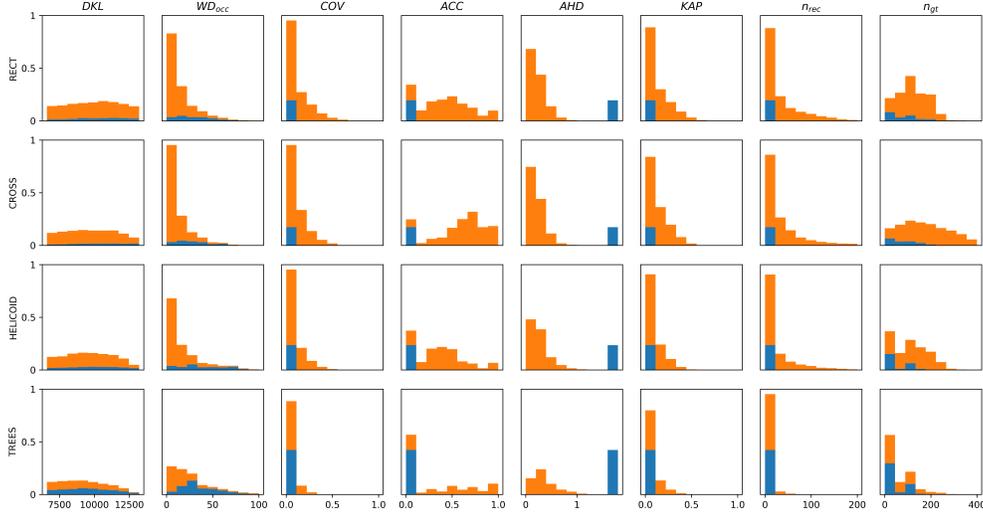


Figure 6: Distribution of the values of the metrics among the cuboids from  $\mathcal{U}_{occ}$  in the experiments. The blue corresponds to the values of the cuboids where  $n_{rec} = 0$ . The orange to the other cuboids. The values are computed only for “observed” cuboids (at least one voxel in  $C_{rec}$  has  $p < 0.4$  or  $p > 0.6$ , explained in Sec. 3.3). The figure display one line per type of world and one column per metric, with two extra-columns showing  $n_{rec}$  and  $n_{gt}$ .

provides measures only when there is at least a reconstructed point, making it easier to identify  $C_{rec}^0$  cuboids.

We have shown in Sec. 5.1 that all the metrics are not sensitive to the same type of errors, with a controlled reconstruction. What happens when the reconstruction is built from the robot’s observations? To answer this question, Figure 7 shows the correlation matrix of the different metrics computed on the cuboids containing at least one reconstructed point, in all the experiments. We computed this same correlation matrix by type of world, and by noise level in the localization, and although the level of correlation may change, the trend remains the same.

From that matrix, we can see that  $KAP$  is highly correlated with  $COV$ , and moderately with  $ACC$ , but also slightly correlated with all the other metrics. We can also see that  $AHD$  is moderately correlated to  $WD_{occ}$ , but also slightly with  $COV$  and  $KAP$ .  $DKL$  is the metric the least correlated with the others. We can hypothesize that the metrics the least corre-

lated are sensitive to different information and may be combined for a more reliable estimate of the reconstruction quality. Such combination is not straightforward. It would require normalizing the combined metrics in a range in which they all are meaningful, and add weighting factors. We leave that to future work.

Finally, Table 2 shows the computation time for each metric. It should be noted that no effort was made to optimize any of the implementations, and that the computation of  $WD_{occ}$  would benefit from running on GPU. What is interesting here, is that some metrics that appear to be moderately correlated have drastically different computation time.  $KAP$  is faster to compute than both  $COV$  and  $ACC$ , whereas  $AHD$  is faster to compute than  $WD_{occ}$ . It is worth noting that  $ACC$  is faster to compute than  $COV$  because there are generally fewer points in the reconstruction than in the ground-truth. That computation time could directly influence the choice of the metric, or, if different metrics are to be combined,

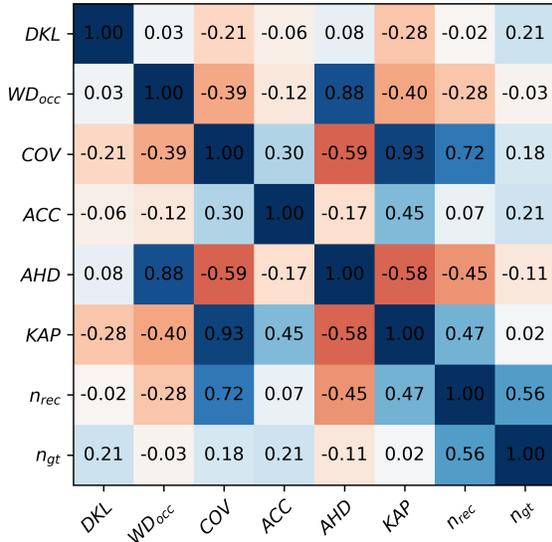


Figure 7: Correlation Matrix of the metrics. The correlation can be positive or negative depending on the type of metric (score or distance).

their selection.

	time ( $\mu s$ )	$\pm std$
DKL	12	$\pm 1$
WD <sub>occ</sub>	$137 \times 10^3$	$\pm 63 \times 10^3$
COV	230	$\pm 53$
ACC	11	$\pm 17$
AHD	241	$\pm 57$
KAP	5	$\pm 1$

Table 2: Comparison of the computation time

### 5.3 Qualitative comparison

In this section, we intend to compare qualitatively the metrics on different reconstructions of the same ground-truth cuboid. To enable this comparison, we design another experiment in simulation, in a world with either a rectangular cuboid or a simulated tree. We create two sets of experiments. In the first set, FRONT, the robot drives in a straight line towards the object. In the second set, LAT, the robot drives in a straight line with the object on the side. We

then build the reconstructions from the robot’s observations, with different noise level in the localization (Sec.4.2).

Fig. 8 and Fig. 9 shows the results for the same cuboid, in the RECT and TREES environments, and the three reconstructions built from the two respective driving behaviors. We focus first on Fig. 8. In the context of an autonomous robot building a map, based on our expectations, we can presume that the maps from LAT are better than the maps from FRONT. Indeed, the group of reconstructions at the bottom of the figure appear ”better” than the group at the top. We then expect the metrics to measure better reconstructions in LAT compared to FRONT. Nonetheless, the metrics yield divergent results when assessing the reconstruction quality, as they are not equally sensitive to all types of errors. For instance, COV measures a better reconstruction in LAT compared to FRONT, as a larger portion of the object has been reconstructed. Conversely, ACC measures a better map in FRONT compared to LAT, as the few points in FRONT are reconstructed more accurately. KAP is highly sensitive to noise in the localization: it penalizes erroneous points, regardless of the Euclidean error distance. Due to errors in the localization, discretization errors, or aliasing, an offset of one voxel, or pixel, is very likely, and a metric that penalizes such errors is very strict. We can see such errors in Fig.8. As a consequence, the variation of KAP’s measures in LAT includes the range of values of those in FRONT. In other words, with KAP, one LAT reconstruction is measured as better than the all FRONT reconstructions, whereas another one is measured as worse. On the contrary, AHD and WD<sub>occ</sub> are robust to that type of errors, as they are to noise in the localization. They provide a measure that is consistent with what one would expect in the context of autonomous robot mapping, that is, LAT are better than FRONT. DKL is dominated by the unknown volume of the map. It does not appear useful in this example, but it might provide information complementary to the other metrics, for instance in an exploration task, where reducing the unknown is a central feature.

Fig. 9 shows the results in the most challenging simulated environment: TREES. First, when we an-

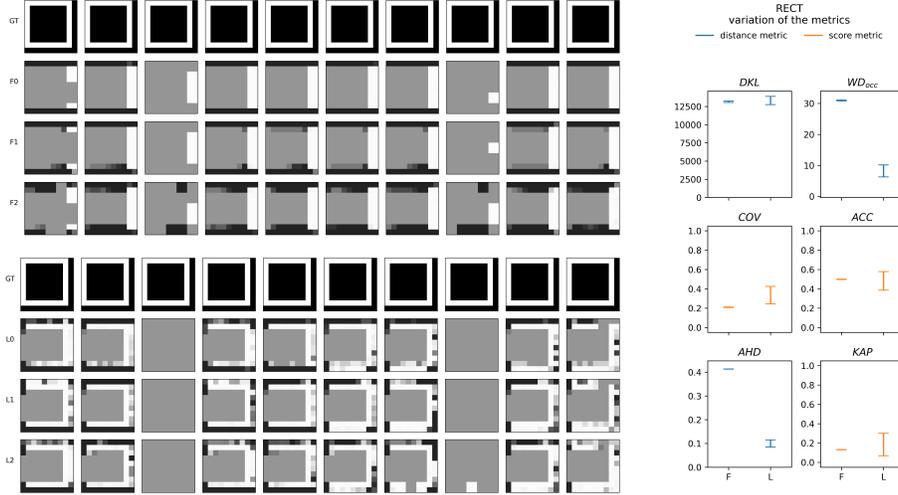


Figure 8: Example of a cuboid in a RECT environment. In the left, the first group of rows corresponds to the cuboids reconstructed with the FRONT driving behavior. GT is the ground-truth cuboid, F0, F1, F2 correspond to the reconstructions obtained with three noise level in the robot localization. The second group of rows corresponds to the LAT driving behavior. L0, L1, L2 to the reconstructions with the three noise levels. Two slices of the cuboids (3 and 8) remain mostly unknown: the Lidar used in this study is a 16-plane Lidar, and those slices remain situated between two of those planes during the experiments. The right part of the figure displays the values of the metrics, with the two different driving behaviors. Distance and score metrics are displayed with different colors to facilitate the interpretation (distance: lower is better, score: higher is better). The errorbars display the variation between the min and the max of each metric, for each driving behavior, when the reconstruction is built with the three noise levels in the robot localization.

analyze this figure, we cannot reach an obvious conclusion, as with the previous example. Visually, the reconstructions from LAT experiments do not appear significantly better compared to FRONT experiments. However, this trend is still what is indicated by all the metrics but  $WD_{occ}$ . This illustrates our claim that measuring 3D-reconstruction quality in such an environment is a challenging task in itself.

Also, we can point out that the metrics are generally sensitive to the density of points, either in the ground-truth or in the reconstruction. When one or both is really low, we reach the limit of all those metrics.

## 5.4 Summary

To summarize, we have seen that no metric is able to provide a meaningful measure in all the situations. Mainly, it depends on the intent of the 3D-reconstruction measurement. Table 3 shows a summary of the metrics properties. In the context of autonomous robot mapping in large scale environment, we believe a key feature is noise-robustness. When the environment is unstructured,  $AHD$  is probably a good choice of metric.  $AHD$  may be combined to  $DKL$  in the context of autonomous exploration, where it would also be interesting to have information on the remaining unknown volume. When the environment is structured,  $ACC$  or  $COV$  might still be good choices too, if the intent of the measure is to assess either the extent of the ground-truth surface

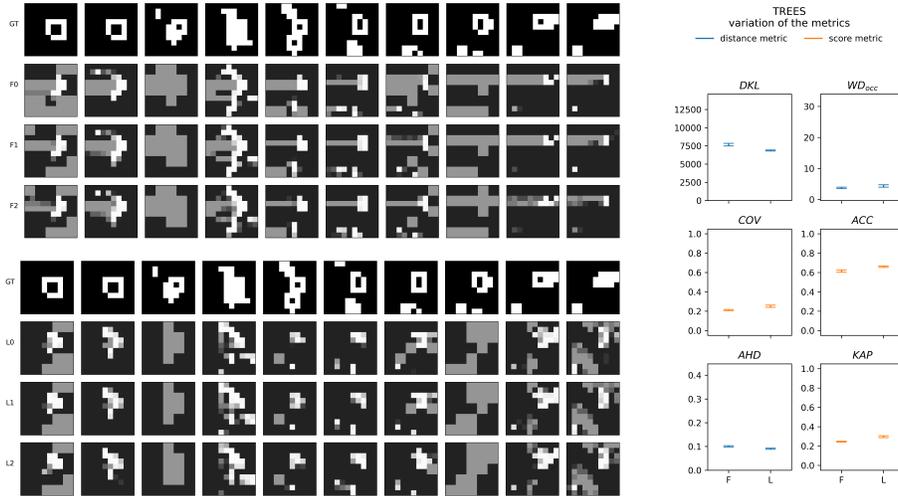


Figure 9: Example of a cuboid in a TREE environment. The figure display the same information that Fig. 8.

	DKL	WD <sub>occ</sub>	COV	ACC	AHD	KAP
sensitive to additional points	2	2	0	2	2	2
sensitive to missing points	1	0	2	0	1	2
informative wrt unknown volume	2	0	0	0	0	0
fast to compute	2	0	1	2	1	2
sensitive to Euclidean distance error	0	2	2	2	2	0
robust to noise	1	2	1	1	2	0
robust to point density	0	0	0	0	1	0
already normalized	0	0	1	1	0	1

Table 3: Summary of the metric properties. 0: the metric does not have this property, 1: it has it, but it is not a significant feature, 2: the property is significant

observed ( $COV$ ), or the precision of the reconstructed points ( $ACC$ ), but not both at the same time. In a structured environment, their lack of robustness to point-density is counterbalanced with the underlying density of the ground-truth. Nonetheless, if the aim of the metric is to provide a robust measure of the reconstruction quality, then  $AHD$  remains a good choice, both in structured and unstructured environments.

## 6 Conclusion

In this study, we have highlighted the challenges associated with assessing the 3D reconstruction quality in large scale environments. We have also shown that these challenges are amplified when dealing with unstructured environments. We have proposed a methodology for comparing the 3D-reconstruction built from the robot’s observations to the ground-truth, at a local level. While our method specifically uses Octomap’s octree, it can be adapted to other mapping techniques, as long as the produced map can be converted to a 3D-grid representation. Furthermore, we have demonstrated that the different

metrics used for evaluating 3D reconstruction quality exhibit distinct properties and sensitivities to specific types of errors. Therefore, the choice of the metric should be carefully considered, based on the specific error modes we want to quantify. We have also observed that comparing the local quality of 3D reconstructions is generally feasible. However, determining an absolute threshold that reliably indicates a good reconstruction is not feasible for this set of metrics in a general case. In conclusion, our study provides practical recommendations for future research focused on evaluating the quality of 3D reconstruction in large-scale environments.

## References

- [1] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: The KITTI dataset, *International Journal of Robotics Research* 32 (11) (2013) 1231–1237. doi:10.1177/0278364913491297.
- [2] F. Pomerleau, F. Colas, R. Siegwart, S. Magnat, Comparing ICP variants on real-world data sets, *Autonomous Robots* 34 (3) (2013) 133–148. doi:10.1007/s10514-013-9327-2.
- [3] P. F. Alcantarilla, C. Beall, F. Dellaert, Large-Scale Dense 3D Reconstruction from Stereo Imagery, 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (2013).
- [4] G. Chahine, M. Vaidis, F. Pomerleau, C. Pradalier, Mapping in unstructured natural environment: a sensor fusion framework for wearable sensor suites, *SN Applied Sciences* 3 (5) (2021) 1–14. doi:10.1007/s42452-021-04555-y.  
URL <https://doi.org/10.1007/s42452-021-04555-y>
- [5] P. Babin, P. Dandurand, V. Kubelka, P. Giguère, F. Pomerleau, Large-Scale 3D Mapping of Subarctic Forests, *Springer Proceedings in Advanced Robotics* 16 (2021) 261–275. arXiv:1904.07814, doi:10.1007/978-981-15-9460-1\_19.
- [6] J. Laconte, S. P. Deschênes, M. Labussière, F. Pomerleau, Lidar measurement bias estimation via return waveform modelling in a context of 3D mapping, *Proceedings - IEEE International Conference on Robotics and Automation 2019-May* (2019) 8100–8106. arXiv:1810.01619, doi:10.1109/ICRA.2019.8793671.
- [7] I. Ben Salah, S. Kramm, C. Demonceaux, P. Vasseur, Summarizing large scale 3D mesh for urban navigation, *Robotics and Autonomous Systems* 152 (2022) 104037. doi:10.1016/j.robot.2022.104037.  
URL <https://doi.org/10.1016/j.robot.2022.104037>
- [8] G. Chahine, C. Pradalier, G. Chahine, C. Pradalier, S.-a. Alignment, N. Outdoor, Semantic-aware spatio-temporal Alignment of Natural Outdoor Surveys To cite this version : HAL Id : hal-03738518 Semantic-aware spatio-temporal Alignment of Natural Outdoor Surveys (2022).
- [9] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, G. Taubin, The ball-pivoting algorithm for surface reconstruction, *IEEE Transactions on Visualization and Computer Graphics* 5 (4) (1999) 349–359. doi:10.1109/2945.817351.
- [10] M. Kazhdan, M. Bolitho, H. Hoppe, Poisson surface reconstruction, in: *Eurographics symposium on Geometry processing*, Vol. 7, 2006.
- [11] S. W. Cheng, T. K. Dey, J. R. Shewchuk, Delaunay mesh generation, *Delaunay Mesh Generation* (2012) 1–386doi:10.1201/b12987.
- [12] T. Wiemann, F. Igelbrink, S. Pütz, J. Hertzberg, A File Structure and Reference Data Set for High Resolution Hyperspectral 3D Point Clouds, *IFAC-PapersOnLine* 52 (8) (2019) 93–98. doi:10.1016/j.ifacol.2019.08.101.
- [13] U. Wickramasinghe, E. Remelli, G. Knott, P. Fua, Voxel2Mesh: 3D Mesh Model Generation from Volumetric Data, *Lecture Notes in Computer Science* (including subseries Lecture

- Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 12264 LNCS (2020) 299–308. arXiv:1912.03681, doi:10.1007/978-3-030-59719-1\_30.
- [14] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, J. Nieto, Voxblox: Incremental 3D Euclidean Signed Distance Fields for on-board MAV planning, IEEE International Conference on Intelligent Robots and Systems 2017-September (2017) 1366–1373. arXiv:1611.03631, doi:10.1109/IROS.2017.8202315.
- [15] A. Khoche, M. K. Wozniak, D. Duberg, P. Jensfelt, Semantic 3D Grid Maps for Autonomous Driving, IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC 2022-October (2022) 2681–2688. arXiv:2211.01700, doi:10.1109/ITSC55140.2022.9922537.
- [16] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, W. Burgard, OctoMap: An efficient probabilistic 3D mapping framework based on octrees, Autonomous Robots 34 (3) (2013) 189–206. doi:10.1007/s10514-012-9321-0.
- [17] N. Aspert, D. Santa-Cruz, T. Ebrahimi, MESH: Measuring errors between surfaces using the Hausdorff distance, in: Proceedings - 2002 IEEE International Conference on Multimedia and Expo, ICME 2002, Vol. 1, 2002. doi:10.1109/ICME.2002.1035879.
- [18] J. Zhou, X. Fu, L. Schumacher, J. Zhou, Evaluating geometric measurement accuracy based on 3d reconstruction of automated imagery in a greenhouse, Sensors (Switzerland) 18 (7) (2018) 1–16. doi:10.3390/s18072270.
- [19] H. Zhu, J. J. Chung, N. R. J. Lawrance, R. Siegwart, J. Alonso-Mora, Online Informative Path Planning for Active Information Gathering of a 3D Surface, in: IEEE International Conference on Robotics and Automation, 2021. arXiv:2103.09556. URL <http://arxiv.org/abs/2103.09556>
- [20] S. Isler, R. Sabzevari, J. Delmerico, D. Scaramuzza, An information gain formulation for active volumetric 3D reconstruction, Proceedings - IEEE International Conference on Robotics and Automation 2016-June (2016) 3477–3484. doi:10.1109/ICRA.2016.7487527.
- [21] S. Song, S. Jo, Surface-Based Exploration for Autonomous 3D Modeling, Proceedings - IEEE International Conference on Robotics and Automation (2018) 4319–4326doi:10.1109/ICRA.2018.8460862.
- [22] A. A. Taha, A. Hanbury, Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool, BMC Medical Imaging 15 (1) (2015). doi:10.1186/s12880-015-0068-x. URL <http://dx.doi.org/10.1186/s12880-015-0068-x>
- [23] D. Müller, I. Soto-Rey, F. Kramer, Towards a guideline for evaluation metrics in medical image segmentation, BMC Research Notes 15 (1) (2022) 1–7. arXiv:2202.05273, doi:10.1186/s13104-022-06096-y.
- [24] A. Bulbul, T. Capin, G. Lavoue, M. Preda, Assessing visual quality of 3-D polygonal models, IEEE Signal Processing Magazine 28 (6) (2011) 80–90. doi:10.1109/MSP.2011.942466.
- [25] W. J. Rucklidge, Efficiently Locating Objects Using the Hausdorff Distance, International Journal of Computer Vision 24 (3) (1997) 251–270. doi:10.1023/A:1007975324482.
- [26] J. Cohen, A Coefficient of Agreement for Nominal Scales, Educational and Psychological Measurement 20 (1) (1960) 37–46. doi:10.1177/001316446002000104.
- [27] S. Kullback, R. A. Leibler, On Information and Sufficiency, The Annals of Mathematical Statistics 22 (1) (1951). doi:10.1214/aoms/1177729694.
- [28] C. Villani, Optimal transport: old and new, Springer Verlag., 2009.

- [29] M. Cuturi, Sinkhorn distances: Lightspeed computation of optimal transport, *Advances in Neural Information Processing Systems* (2013) 1–9.
- [30] R. Flamary, N. Courty, A. Gramfort, M. Z. Alaya, A. Boisbunon, S. Chambon, L. Chapel, A. Corenflos, K. Fatras, N. Fournier, L. Gautheron, N. T. Gayraud, H. Janati, A. Rakotomamonjy, I. Redko, A. Rolet, A. Schutz, V. Seguy, D. J. Sutherland, R. Tavenard, A. Tong, T. Vayer, POT python optimal transport library, *Journal of Machine Learning Research* 22(78) (2021) 1–8.  
URL `github:https://github.com/rflamary/POT`