# Measuring 3D-reconstruction quality in probabilistic volumetric maps with the Wasserstein Distance

Stéphanie Aravecchia, Antoine Richard, Marianne Clausel, Cedric Pradalier

# Measuring 3D-reconstruction quality in probabilistic volumetric maps with the Wasserstein Distance

Stéphanie Aravecchia[1], Antoine Richard[2], Marianne Clausel[3], Cédric Pradalier[1]

***Abstract*— In this study, we address the challenge of measuring 3D-reconstruction quality in large unstructured environments, when the map is built with uncertainty in the robot localization. The challenge lies in measuring the quality of a reconstruction against the ground-truth when the data is extremely sparse and where traditional methods, such as surface distance metrics, fail. We propose a complete methodology to measure the quality of the reconstruction, at a local level, in both structured and unstructured environments. Building upon the fact that a common map representation in robotics is the probabilistic volumetric map, we propose, along this methodology, to use a novel metric to measure the map quality based directly on the voxels' occupancy likelihood: the Wasserstein Distance. Finally, we evaluate this Wasserstein Distance metric in simulation, under different level of noise in the robot localization, and in a real world experiment, demonstrating the robustness of our method.**

## I. Introduction

In this work, we are interested in large scale natural environments, such as parks or forests, sometimes called "unstructured" ( [1], [2]), in opposition to environments containing large geometrical shapes, such as buildings, called "structured". In a large scale unstructured environment, not only computing an accurate 3D map, with a 3D-Lidar, is particularly difficult but also measuring the quality of the produced map is a challenging task in itself.

When building the map in unstructured environments, among the main challenges, we can specifically highlight the sparsity of the data and the robot localization ( [1], [2]). To begin with the sparsity, the output from an affordable 3D-Lidar is sparse. For example, the Ouster-16 used in this study provides only 8,192 points at 10 Hz, whereas the number of points is generally 100 times higher and the frame rate at least 3 times higher when considering an RGBD camera or the point-cloud derived from visual odometry algorithms. In addition to the sparsity of the laser output, the environment itself leads to a sparse reconstruction. For example, 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 [3]. 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.

In a robotic monitoring or inspection task, an important question to consider is "how accurate is the map locally?". With sparse 3D-reconstructions, in unstructured environments, answering this question is ambiguous. First, because the definition of ground-truth is not straightforward: how to acquire the real ground-truth of a park-like environment? Furthermore, because most classical methods, based on statistics on surface distances, require a reconstructed surface to measure the distance between the reconstruction and the ground-truth. We will show that for natural environments, attempting to reconstruct surfaces that match the actual surfaces of trees, trunks, small branches and leaves is doomed to fail. We circumvent this issue by measuring the quality of the reconstruction not on the surfaces, but directly on the probabilistic map. We propose to measure the "distance" between the occupancy likelihood in the 3D-reconstruction and the occupancy likelihood in the ground-truth. This measure is given by the Wasserstein Distance.

Finally, to obtain local information about the quality of the 3D-reconstruction, we propose a methodology to compare reconstruction to ground-truth at a local level. The code to reproduce the method and experiments is open-source [1].

To summarize, our contributions are:

- a methodology to compare locally 3D-reconstruction and ground-truth, applicable to both structured or unstructured environments;
- a metric to measure the quality of a sparse 3D-reconstruction, also applicable to both structured or unstructured environments: the Wasserstein Distance.

## II. Related Work

### A. Map building

The most common ways to represent maps are volumetric maps, i.e. 3D-grids, and meshes, i.e. 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.

[1] are with IRL2958 GT-CNRS, 2 rue Marconi, 57070 Metz, France first.last@georgiatech-metz.fr

[2] is with University of Luxembourg, Luxembourg

[3] is with Université de Lorraine, Nancy, France

[1] https://github.com/stephanie-aravecchia/3d-reconstruction-metrics.git

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 [4], Poisson Surface Reconstruction [5] or Delaunay triangulations [6]. Although results on the meshing of a single object is widely shown in the literature, meshing large scale natural environments is still a problem which is not solved. Some meshes can be derived, but they are generally a rough approximation of trees and bushes.

In this paper, we focus on a more common map representation for outdoor robotics: volumetric maps. Such a map is a discretization of the space in voxels, each voxel containing some information, such as its occupancy likelihood. [7] presented Octomap, a method to build and store an octree instead of a 3D-grid, saving memory and computation. The open-source Octomap library 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. We use Octomap to build the probabilistic volumetric map denoted as "reconstruction" in this paper.

It is also important to note that the quality of 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 [1]. For those reasons, we evaluate the robustness of our method with respect to the localization precision, with different level of noise in simulation, and with a real experiment.

### B. 3D-reconstruction metrics

In the context of 3D-reconstruction, several metrics exists to measure the quality of the 3D-reconstruction.

Generally, when considering 3D-reconstruction, the objective is to compare two surfaces: the mesh generated from the 3D-point cloud, and the ground-truth mesh. The 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. Statistical metrics are then extracted, among which we typically see the Hausdorff Distance, the Root Mean Square Error (RMSE) or the MAE (Mean Average Error) ( [8]–[10]).

In the context of robotics, the surface coverage is derived from those surface distance errors, and applied either on the meshes or on the 3D-grids. In that case, 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 observed. The metric is the proportion of such points ( [11], [12]).

The fact that those metrics aim at measuring the quality of a dense reconstruction, i.e. reconstructing dense objects with high density point-clouds, is their main limitation. In the context of unstructured environments, the 3D-reconstruction tends to remain sparse, because only a few voxels will reach the threshold probability for them to be considered occupied. Therefore, in a sparse reconstruction, of an unstructured environment, the level of information is not sufficient to recover a surface, as we will also show later. This means that none of the previously seen metrics are adequate to evaluate sparse 3D-reconstructions of unstructured environments. Furthermore, those methods solely focus on the reconstruction of occupied voxels, and not on empty space. In the context of large scale environments with sparse objects, a large portion of the volume corresponds to empty space. Therefore, it is also interesting to measure how well the empty space has been reconstructed.

### C. 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 [11]. However, they do not propose a mean to compare the reconstructed volume to a reference one. In this paper, we propose a methodology to compare two volumetric maps with probabilistic values. Something that, to the best of our knowledge, has not been done before.

Different methods allow comparing probabilities. The most common is probably the Kullback-Leibler divergence (DKL): a measure of how different two probability distributions are. 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 calculate the Optimal Transport plan, linking one probability distribution to another one, with a cost function depending on the geometry [13]. 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). To bypass computational issues, [14] regularizes the optimal transport problem by adding an entropic term, and solves it using a Sinkhorn's fixed point iteration. [15] has implemented an open source Python library providing several solvers for Optimal Transport problems, including [14]'s algorithm. With the Wasserstein Distance, or its regularized variants, 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.

## III. Method

The objective of our method is to compare 3D-reconstruction to ground-truth, and to measure the quality of the reconstruction at a local level. To do so, we discretize the space into cuboid regions and calculate a reconstruction metric for each cuboid.

### A. Cuboid region comparison

This section explains how we compare the reconstruction to the ground-truth locally, on cuboid regions. To enable this local comparison, we discretize the volume the reconstruction and the ground-truth represent 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 metrics between intersecting cuboids. Alg. 1 summarizes this section.

The 3D-reconstruction is obtained from the full probability map constructed by Octomap, the ground-truth is either the mesh of the scene used in simulation, or the point-cloud obtained when scanning the area with a Total Station (Sec. IV-A). For the ground-truth, in the simulated environment, 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 and a value of 0.0. In the real experiment, we construct a 3D matrix of voxel size *res* containing zeros. We iterate on the point-cloud from the Total Station, and for each point, we set the value of the corresponding voxel to 1.0. Our ground-truth dataset, $\mathcal{D}_{\text{gt}}$, 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 $B_{\text{box}_1}$. 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, $\mathcal{D}_{\text{rec}}$, a 3D matrix of size ($h_2$,$w_2$,$n_2$), with a voxel resolution *res*, with its associated bounding-box, $B_{\text{box}_2}$, in the same reference frame, $R_f$.

Finally, we do the comparison in $B_{\text{box}_1} \cap B_{\text{box}_2}$. 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 $B_{\text{box}_1} \cap B_{\text{box}_2}$, in $R_f$, and each element in $C_{rec}$ and $C_{gt}$ stores the occupancy likelihood of its corresponding voxel of size $n$ in $R_f$.

### B. Comparison metrics

This section explains how, for each cuboid region, we compute different metrics to indicate the quality of the reconstruction (Alg. 2).

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 quality of reconstruction of the cuboids with a different metric in each set: the Wasserstein Distance for the cuboids in $\mathcal{U}_{occ}$, the $L_1$ norm for the cuboids in $\mathcal{U}_{empty}$.

Furthermore, because measuring the quality of reconstruction 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 the maximum values $WD_{occ}^{max}$ and $L_1^{max}$.

*1) 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.

First, we 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 vectors of probability distributions $P_{gt}^{occ}$ 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.

Finally, we calculate $WD_{occ}$, the Wasserstein Distance between $P_{rec}^{occ}$ and $P_{gt}^{occ}$, using the Sinkhorn algorithm described in [14], following the implementation of [15]. The cost matrix is set to contain the squared Euclidean distance between the voxels associated to the element of each vector.

*2) $L_1$ 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). Calculating the optimal transport plan makes little sense, because there is no mass distribution on the reference vector. Hence, the Euclidean Distance the mass has to be moved by is not relevant, as long as the mass is evenly distributed to a uniform distribution at the end. Furthermore, what is interesting is only how different the emptiness likelihood is from actual 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.

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

```
// D_gt and D_rec are the datasets
D_gt(B_box1, (h1, w1, n1), R_f), D_rec(B_box2, (h2, w2, n2), R_f)
B_box ← B_box1 ∩ B_box2
M_gt ← D_gt(B_box), M_rec ← D_rec(B_box)
for all cuboid ∈ B_box do:
    C_gt ← M_gt(cuboid), C_rec ← M_rec(cuboid)
    if isObserved(C_rec) then:
        if cuboid ∈ U_occ then:
            cuboid.metrics ← computeWD(C_rec, C_gt)
        else:
            cuboid.metrics ← computeL1(C_rec)
        end if
    else:
        cuboid.metrics ← maxMetrics
    end if
end for
```

**Algorithm 2** Metric Computation

```
V_gt, V_rec ← getVectorsFromCubes(C_gt, C_rec)
if cuboid ∈ U_occ then:
    V_rec^occ, V_rec^free ← getFreeAndOccVectors(V_rec)
    V_gt^occ, V_gt^free ← getFreeAndOccVectors(V_gt)
    P_rec^occ, P_gt^occ ← normalizeToDistribution(V_rec^occ, V_gt^occ)
    cuboid.metrics ← WassersteinDistance(P_rec^occ, P_gt^occ)
else
    cuboid.metrics ← L1(V_rec)
end if
```

### C. Evaluation Methodology

In order to evaluate our metric $WD_{occ}$, we first assess that the metric is meaningful. Then, we compare $WD_{occ}$ against a traditional metric: the surface coverage.

To calculate the surface coverage, we apply the classical methodology in a 3D-grid: all the points in the 3D-grid are considered belonging to the surface if their occupancy likelihood is above a threshold. We calculate the Euclidean distance between a ground-truth point $S$ and the closest reconstructed point $P$. If the distance $SP$ is below a registration distance, the point is considered reconstructed. The total number of ground-truth points is $n_S$, the total number of considered reconstructed points is $n_{rec}$. The surface coverage is then $cov = n_{rec}/n_S$.

First, to assess that $WD_{occ}$ is meaningful and is a distance that actually measures the 3D-reconstruction quality, we select randomly $n$ cuboids $C_{gt}$ from the ground-truth containing at least one occupied point. We then simulate several "ideal" reconstructions: we apply to $C_{gt}$ a 3D Gaussian convolution, with different kernel size and sigmas, and then add a uniform noise. We also simulate a random reconstruction. Finally, we calculate $WD_{occ}$ on those cuboids. If the value of $WD_{occ}$ is significantly smaller for an ideal reconstruction than from a random reconstruction, we can conclude that $WD_{occ}$ is a meaningful metric of 3D-reconstruction quality.

Second, to compare $WD_{occ}$ and $cov$, we compare their ability to measure 3D-reconstruction quality. To assess if a metric is able to measure the reconstruction quality, we simply consider if the metric contains or does not contain information. Here, we do not consider the quality of the reconstruction in itself, but the ability of the metric to measure this quality. For $cov$, we simply consider that if $cov > 0$, it contains information. For $WD_{occ}$, we consider that if $WD_{occ} < WD^*_{occ}$, then it contains information. We set the limit $WD^*_{occ}$ to the mean of the values calculated before with random reconstructions, in the same environment. Then, we calculate, for each metric, the proportion of cuboids for which the metric contains information in $\mathcal{U}_{occ}$. The higher the proportion, the more informative the metric.

Finally, we want to compare $WD_{occ}$ and $cov$ on their ability to discriminate different reconstruction qualities, but also on their robustness. For each "ideal" reconstruction introduced before, we calculate the median of the quality of reconstruction with $WD_{occ}$ and $cov$ on all the cuboids. Let $m$ be this median. We now consider $\mu_b = (1/n)\sum_{i=1}^{i=n} m_i$, with $n$ the number of experiments, an $b$ the level of blur and noise applied to the ground truth to obtain the "ideal" reconstruction. The ability of the metric to discriminate the reconstruction is then shown by $\Delta_k = (\mu_{b_k} - \mu_{b_1})/\mu_{b_1}$, with $k$ the number of different level of blur and noise considered. When we increase the level of noise and blur in the reconstruction, we expect that $\Delta_k$ increases, but not too significantly because even blurred and with some noise, all the considered reconstructions are very good.

## IV. Experiments and Results

Our experiments are done in simulation and in the field. In both cases, we use the ROS framework and the robot is a Clearpath Husky, equipped with a 3D Lidar Ouster OS1-16 (16 planes of 512 points).

### A. Experimental Setup in Simulation

The simulator is Gazebo. A mixture of noise is applied to the Lidar to simulate better the behavior of a Lidar in outdoor natural environments.

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 [2], to

[2] https://pointpats.readthedocs.io/

reproduce the natural implantation of trees. The assets can be either rectangular cuboids or randomly selected in our 15-trees library, where simulated trees are created with a space colonization algorithm [3]. Due to computational considerations in Gazebo, we create only trees without leaves. A random scaling factor is applied on the x, y and z dimensions of the assets, as well as on their orientation.

With the same point process generation, we create two different synthetic environments. The first one contains only rectangular cuboids: the structured environment. The second one contains only trees: the unstructured environment. With the open-source Blender software, we create a single shape-file (.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.

Next, we run a simulation for each environment where the robot is following the same list of waypoints, set manually.

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 presented here are (with $\sigma_p$, standard deviation on position, in m, $\sigma_q$, on orientation, in rad):

- noise 0: perfect localization;
- noise 1: $\sigma_p = 0.005$, $\sigma_q = 0.005$;
- noise 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 0.25m error in the position of a 25m distant point in the Lidar point-cloud).

We generate 6 environments (3 different asset distribution with 1 structured, 1 unstructured each). For each environment, and for each of the 3 noise levels, we build two 3D-grids with Octomap (max-range $25m$), one with resolution $res = 0.05m$, the other $res = 0.1m$. In total, we have 36 reconstructions to compare to 6 ground-truths.

### B. Experimental Setup in the field

The experimental field is a small park containing trees, of an approximate area of $1500m^2$. Although it is not possible to acquire the ground-truth of such an environment, we consider that the 3D point-cloud obtained from the scan of the environment with a Total Station Leica TS60 is precise and dense enough to be considered as ground truth. The horizontal and vertical angular resolution of the scanning is set to 0.05 degrees. To scan the area, the Total Station is placed on three different locations, to have different point of views on the trees. Each scan takes 40 to 60 minutes. At the end, we obtain a single consistent point-cloud from the area. To localize the robot in this area, we track it with the Total Station. The robot is equipped with a prism, that is localized by the Total Station with centimeter accuracy. In a post-processing phase, we recover the orientation by fusing, in an Extended Kalman Filter, the localization from the Total Station, the wheel odometry of the robot, and the IMU. To avoid large errors due to large uncertainty in the orientation, we filter out the Lidar scans associated to turning motions of the robot when we build the map with Octomap. With this setup, we can consider we have a "good" localization for an outdoor robotics application. For this experiment, we obtain two reconstructions with Octomap (max-range $10m$): one with resolution $res = 0.05m$ and one with $res = 0.1m$: we have two reconstructions to compare to one ground-truth.

Ultimately, the volumetric maps are compared to the ground-truth as described in III, with a cuboid resolution of $RES = 10 \times res$. In the experiments, we are interested in comparing the reconstruction of salient objects, not the ground. To enforce it, we apply a threshold on the Z coordinates of the bounding-box on which we perform the comparison.

### C. Results

After processing the data as explained in III, we obtain for each cuboid region in $\mathcal{U}_{occ}$ the reconstruction quality metric $WD_{occ}$ and other metrics we will compare our method to.

Fig. 1: Challenges with meshing: (a) the real tree, (b) the point-cloud from the Total Station (c) an example of mesh failing to describe the actual surface of the fine elements of the tree.

*1) Surface Distance Metrics:* The underlying idea behind our method stems from the issues we encountered when trying to use surface distance metrics. Fig. 1 illustrates the difficulties when trying to reconstruct the surface of a tree. The point-cloud used here is the point-cloud from the Total Station (i.e our ground-truth). As we can see, it is dense on the trunk of the tree and on the larger branches, but really sparse on the small branches and on the leaves, where the sampling is not sufficient to recover the surface. As a consequence, classic meshing methods are able to reconstruct the surface of the trunk, but fail when it comes to the rest of the tree (i.e. the unstructured part). The example shown in Fig. 1 is the mesh reconstructed with the pivoting-ball algorithm with the open-source software MeshLab. We tried all the meshing algorithms available in MeshLab, with default parameters, and that mesh is the best result. We do not claim that there is no solution to compute a better mesh, but we illustrate the difficulties with off-the-shelf algorithms. This mesh, although computed on a single tree with the point-cloud from a Total Station, is also representative of the challenges involved when reconstructing meshes with robot-mounted Lidars in a natural environment. For these reasons, we cannot apply directly any surface distance metric (such as the RMSE) on the reconstruction of that type of environment.

[3] https://github.com/dsforza96/tree-gen

In the context of 3D-grids, a common solution to circumvent the meshing is to calculate the distances not between surfaces but between points. We calculate the surface coverage *cov*, as described in III-C, with the following parameters ($p$, the occupancy likelihood; $d$, the registration distance):

- cov 1: $p = 0.8$, $d = 0.05m$ (as in [11], [12]);
- cov 2: $p = 0.8$, $d = 0.1m$ ;
- cov 3: $p = 0.7$, $d = 0.1m$;
- cov 4: $p = 0.7$, $d = 0.15m$.

Doing so, we have surface coverages with different levels of permissiveness to compare our metric to.

*2) Our metric:* First, we demonstrate, as described in Sec. III-C, that our metric is meaningful.

From each ground truth, we generate three "ideal" reconstructions, where we first blur the ground truth, and then add a uniform noise:

- $b_1$: blurred with kernel size 5, $\sigma$ 0.07, no uniform noise,
- $b_2$: blurred with kernel size 7, $\sigma$ 0.08, uniform noise 0.05,
- $b_3$: blurred with kernel size 11, $\sigma$ 0.2, uniform noise 0.1.

Then, we compute $WD_{occ}$ from those "ideal" reconstructions to the ground truth, and from a random reconstruction to the ground truth. Fig. 2 shows the distribution of the calculated $WD_{occ}$. For visualization purposes on the "ideal" reconstruction, we only display the results for $b_3$. The other results are similar. This figure shows that that $WD_{occ}$ is close to zero when the reconstruction is close to an ideal reconstruction. Conversely, $WD_{occ}$ is large when the reconstruction is random. It shows that our proposed metric is relevant to measure the 3D-reconstruction quality: it is small when the reconstruction is similar to the ground-truth, large if not. Additionally, the figure also shows that the distribution of $WD_{occ}$ when a random reconstruction is compared to the ground-truth is dependent on the environment. However, this dependence is not significant with regard to the 3D-grid resolution.

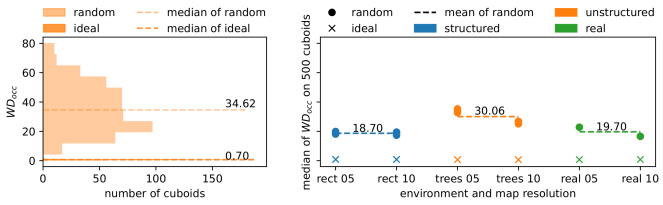


Fig. 2: Left: distribution of $WD_{occ}$ when random and ideal reconstructions are compared to ground truth, for an unstructured environment in simulation, on 500 cuboids, with the values of the medians. Right: summary of the medians of $WD_{occ}$ on random and ideal reconstructions on all our environments. The dashed-lines are the limits for each environment ($WD^*_{occ}$) (used later).

Next, we demonstrate that our metric $WD_{occ}$ is able to measure 3D-reconstruction quality in both structured and unstructured environments, whereas the more classic surface coverage *cov* tends to fail in unstructured environments, specifically when the robot localization is uncertain.

Before going further, we would like to illustrate the previous statement with a more concrete example, where we show at the same time the ability of $WD_{occ}$ to measure and discriminate different reconstructions, and the lower ability of *cov* at doing so.

Fig. 3 shows three cuboids: on the first line, the ground-truth, on the second and third lines, the 3D-reconstructions obtained from two different driving motions. On the second line (*rec 1*), the robot drives in a straight line towards the object. On the third line (*rec 2*), the robot drives in a straight line with the object on the side. The reconstructions are obtained with a perfect localization. The object of interest is a different geometric shape than what is presented so far: a cross extruded shape. The idea of this experiment is to demonstrate the impact of the occlusions on the quality of reconstruction, and on its measure. In both reconstruction cuboids, two slices (6 and 8) remain unknown: the Lidar used in this study is a 16-plane Lidar, and those slices remain situated between two of those planes during the two experiments. Also, in this particular experiment, for illustration purposes, the cuboid contains the ground-plane (this is not the case in the other cuboids in this section, where we are interested only in salient objects). We can see in the figure that the reconstruction *rec 2* is better than *rec 1*. Due to the occlusion, in *rec 1* half of the shape is hidden from the robot. The measure of the quality of reconstruction with $WD_{occ}$ and *cov* are shown in Table I. The variation of $WD_{occ}$ between the two reconstructions is close to 80%, whereas the variation of *cov* remains under 50%. This shows, on this example, that $WD_{occ}$ is better than *cov* at discriminating between the two reconstructions.

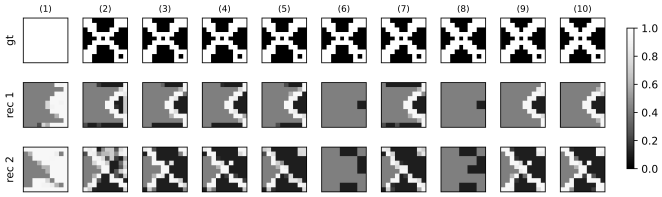


Fig. 3: Example of cuboids (10x10x10 voxels). Each line correspond to a different cuboid: (1) the ground-truth, (2) and (3) two different reconstructions. Each row corresponds to a slice of the cuboid. The figure displays 10 slices, each a 10x10 pixels image. The first row corresponds to the ground-plane, the height increases with the rows. The color depends on the occupancy likelihood.

| | $rec_1$ | $rec_2$ | $\delta$ |
|---|---|---|---|
| $WD_{occ}$ | 14.39 | 3.01 | -79.10% |
| $cov_1$ | 0.25 | 0.44 | -44.35% |
| $cov_2$ | 0.44 | 0.80 | -45.18% |
| $cov_3$ | 0.44 | 0.81 | -45.31% |
| $cov_4$ | 0.52 | 0.95 | -45.01% |

TABLE I: Quality of reconstruction of the two cuboids shown Fig. 3, measured with different metrics. The last column is the variation of the metric from its maximum.

To demonstrate the behavior shown in this single example, we compare $WD_{occ}$ and *cov*, following Sec. III-C, that is, we compare the proportion of cuboids for which the metric is informative. $WD^*_{occ}$ is set to the values shown in Fig. 2.

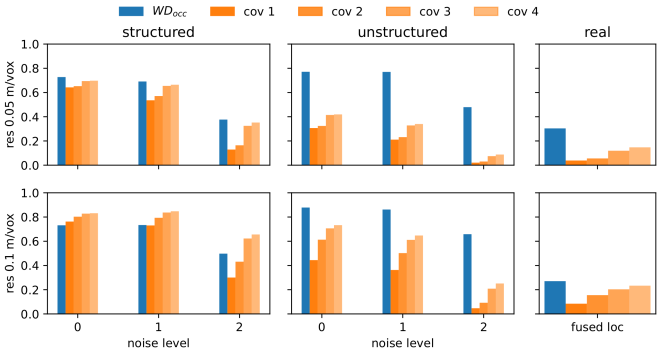


Fig. 4: Proportion of cuboids for which the metric is informative. Columns 1,2: results for the same point process simulation environment (1: structured, 2: unstructured), Column 3: real experiment. One line per map resolution (1: 0.05m/vox, 2: 0.10m/vox). Blue is the proportion of informative cuboids with our proposed metric ($WD_{occ}$), orange with different surface coverage ($cov$). The results are grouped by noise level.

The results are shown in Fig. 4 for a single environment in simulation (structured and unstructured), and for the real experiment. The results for all the other simulated environments are similar. We can see that, as expected, $cov$ is generally working in structured environments, although it is sensitive to uncertainty in the robot localization and to the map resolution. Additionally, in unstructured environments, with noise in the localization, the proportion of cuboids for which $cov$ is informative is low. On the contrary, our proposed metric $WD_{occ}$ is working both in structured and unstructured environments and is less sensitive than $cov$ to map resolution or to noise in the localization. If we consider the map resolution of 0.05m/vox, that seems suitable for mapping trees in large scale natural environments, we can see that, in the case of noisy localization in simulation, $WD_{occ}$ is 5 times more informative than the most permissive $cov$ ($p = 0.7$, $d = 0.15m$). In the real experiment, even with what we can consider a good localization, $WD_{occ}$ is still twice more informative than the most permissive $cov$.

Finally, we further demonstrate that $WD_{occ}$ is an interesting metric. Not only it is discriminant of the reconstruction quality, but it is also robust. To do so, we show how $WD_{occ}$ behaves when we increase the blur and the uniform noise in the "ideal" reconstructions, following Sec. III-C. For each $k$ level of blur and noise in the "ideal" reconstruction $[b_1, b_2, b_3]$, we compute $\Delta_k$. Table II shows the results grouped by type of world, with grid resolution $res = 0.1m$. When we increase the level of blur and noise (i.e increase the index $k$), $\Delta_k$ increases, demonstrating that $WD_{occ}$ is able to discriminate between those good reconstructions. Nonetheless, the value of $\Delta_k$ does not change significantly, demonstrating that $WD_{occ}$ is also a robust metric. The same measure done with $cov$ is always the same: 1.00. $cov$ is not able to discriminate those ideal reconstructions.

## V. Conclusion

We have shown that our proposed metric $WD_{occ}$ is capable of measuring 3D-reconstruction quality both in structured and unstructured environments, even with noise in the robot localization, when traditional metrics such as surface coverage fail. Additionally, we have proposed a methodology to calculate the quality of the 3D-reconstruction at a local level, where the space is divided into cuboid regions. Based on these results, we can now foresee the integration of this metric into inspection and monitoring tasks, either by giving feedback to a human operator or by using them in an autonomous exploration framework. Looking even further, we intend to use this metric within the computation of a reward function for reinforcement learning algorithms.

| | $\Delta_2$ | $\Delta_3$ |
|---|---|---|
| structured | 1.21% | 3.93% |
| unstructured | 1.65% | 6.20% |
| real | 0.39% | 3.46% |

TABLE II: Variation of the measure of the reconstruction quality with $WD_{occ}$ when we incrementally add blur and noise.

## References


[1] G. Chahine, M. Vaidis, F. Pomerleau, and C. Pradalier, "Mapping in unstructured natural environment: a sensor fusion framework for wearable sensor suites," *SN Applied Sciences*, vol. 3, no. 5, pp. 1–14, 2021.

[2] P. Babin, P. Dandurand, V. Kubelka, P. Giguère, and F. Pomerleau, "Large-Scale 3D Mapping of Subarctic Forests," *Springer Proceedings in Advanced Robotics*, vol. 16, pp. 261–275, 2021.

[3] J. Laconte, S. P. Deschênes, M. Labussière, and F. Pomerleau, "Lidar measurement bias estimation via return waveform modelling in a context of 3D mapping," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May, pp. 8100–8106, 2019.

[4] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 4, pp. 349–359, 1999.

[5] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Eurographics symposium on Geometry processing*, vol. 7, 2006.

[6] S.-W. Cheng, T. K. Dey, J. Shewchuk, and S. Sahni, *Delaunay mesh generation*. CRC Press Boca Raton, 2013.

[7] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[8] N. Aspert, D. Santa-Cruz, and T. Ebrahimi, "MESH: Measuring errors between surfaces using the Hausdorff distance," in *Proceedings - 2002 IEEE International Conference on Multimedia and Expo, ICME 2002*.

[9] J. Zhou, X. Fu, L. Schumacher, and J. Zhou, "Evaluating geometric measurement accuracy based on 3d reconstruction of automated imagery in a greenhouse," *Sensors (Switzerland)*, vol. 18, no. 7, pp. 1–16, 2018.

[10] H. Zhu, J. J. Chung, N. R. J. Lawrance, R. Siegwart, and J. Alonso-Mora, "Online Informative Path Planning for Active Information Gathering of a 3D Surface," in *IEEE International Conference on Robotics and Automation*, 2021.

[11] S. Isler, R. Sabzevari, J. Delmerico, and D. Scaramuzza, "An information gain formulation for active volumetric 3D reconstruction," *Proceedings - IEEE International Conference on Robotics and Automation*, 2016.

[12] S. Song and S. Jo, "Surface-Based Exploration for Autonomous 3D Modeling," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4319–4326, 2018.

[13] C. Villani, *Optimal transport: old and new*. Springer Verlag., 2009.

[14] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," *Advances in Neural Information Processing Systems*, pp. 1–9, 2013.

[15] R. Flamary *et al.*, "POT python optimal transport library," *Journal of Machine Learning Research*, vol. 22(78), pp. 1–8, 2021.