

EVALUATING SURFACE MESH RECONSTRUCTION OF OPEN SCENES

Yanis Marchand, Bruno Vallet, Laurent Caraffa

LASTIG, Univ Gustave Eiffel, ENSG, IGN, F-94160 Saint-Mandé, France – yanismarchan@gmail.com

KEY WORDS: Computer Graphics, Surface Reconstruction, Evaluation protocol, Benchmark, LiDAR simulation, Open-Source

ABSTRACT:

This paper addresses the evaluation of algorithms reconstructing a watertight surface from a point cloud acquired on an open scene. The objective is to set a rigorous protocol measuring the quality of the reconstruction and to propose a quality metric that is informative with respect to the various qualities that such an algorithm should have, and in particular its capacity to interpolate and extrapolate accurately. Our approach aims at being more informative and rigorous than previous works on this topic. In addition, we use publicly available data and our implementation is open-source. We argue that a rigorous evaluation of surface reconstruction of open scenes needs to be performed on synthetic data where a perfect continuous ground truth surface is available, so we developed our own LiDAR simulator of which we give a description in the present paper.

1. INTRODUCTION

The topic of surface mesh reconstruction from point cloud has been thoroughly studied judging by the number of approaches recently surveyed by (Berger et al., 2017) and (Khatamian and Arabnia, 2016). The goal of this task is to produce a digital hole-free continuous representation (triangle mesh) of the visible surface of individual objects or entire scenes from sensing data (mostly images and LiDAR scans). In the context of remote sensing, this topic has first been neglected in favor of 2.5D approaches where the surface reconstruction problem is merely a question of 2D interpolation of possibly sparse height data sampled on a regular grid, leading to the popular Digital Elevation Models (DEMs) or Digital Surface Models (DSMs) used to represent the geometry of the visible surface of a scene seen from above. This representation is however getting more and more limited: an increasing number of applications require terrestrial data (from Mobile Mapping Systems, fixed stations and hand-held cameras) for which the 2.5D setting is completely inappropriate. Moreover, the spread of aerial oblique imagery aiming at acquiring more data on vertical surfaces, and the fact that some aerial LiDAR can scan up to 40° away from the vertical call for more generic 3D models to produce a continuous geometric representation of the underlying scene. Finally, an increasing number of commercial products such as Sure from NFrames or ContextCapture from Bentley already propose full 3D processing pipelines for surface mesh reconstruction from remote sensing data. For all these reasons, 3D surface reconstruction, once a topic mainly studied in the geometry processing community, is becoming more and more widespread in remote sensing.

In this paper, we focus on watertight surface reconstruction of open scenes. Watertightness (meaning that the surface has no borders or holes) is a desirable property for most applications (simulation, visualization, ...) and most state of the art approaches produce watertight surfaces. We define open scenes as scenes where only a part of the scanned object is seen, such as aerial or (outdoor) terrestrial acquisitions where only a very small part of the object (the earth) is acquired. This is somehow contradictory with watertightness as the scan has a limited extent, thus a border. To lift this contradiction, some approaches

allow the reconstructed surface to intersect the bounding box or convex hull of the points to avoid the necessity to completely close the surface after this border, which we will call *soft* watertightness, by opposition to *hard* watertightness where the surface really has no borders. Obviously, *hard* watertight approaches will compare very defavourably to *soft* watertight approaches as a large surface uncoherent with the real scene needs to be added to close the surface.

As pointed out by (Van Kreveld et al., 2013), there is a lack of ground truth and benchmark in the field of urban reconstruction. Therefore, the first motivation of the work presented in this paper is to propose adequate metrics to evaluate algorithms which reconstruct 3D meshes from point clouds, which faces three challenges. First, Surface mesh reconstruction is a complicated task and few existing open source tools are easy to use, especially on massive remote sensing data. Second, most reconstruction methods aim at reconstructing a watertight mesh. While this is important for many applications, an open scene necessarily has a border that the algorithms need to cope with. Finally, A 3D mesh reconstruction algorithm aims at recovering the continuous nature of the underlying scene, such that evaluation metrics need to be based on a continuous representation of the scene that can only be accessed if the data acquisition is simulated on an existing realistic continuous surface.

We consider that a surface reconstruction algorithm mainly needs to deal with interpolation (recovering the surface where the scene is well seen from the sensor dealing with noise, outliers and varying point density while preserving the (possibly sharp) features of the scene) and extrapolation (filling holes of various sizes to make the surface watertight). Mathematically speaking, these two issues are the same, we want the algorithm to guess where the real surface is, more or less close to the input points. In this paper we will call α -interpolation the capacity of the algorithm to recover the correct surface at distance α , and propose a metric to measure exactly how good are the evaluated algorithms at α -interpolation. We argue that except for extremely simple (thus not interesting) scenes, this is only possible on synthetic data, because we will never be able to scan a complex open scene without any occlusion, such that the absence of data point at a point of space can never ensure that there is no surface there. Thus we propose to base our evalu-

Source code is available at: [GitHub/SurfaceReconEval](https://github.com/SurfaceReconEval)

ation on a virtual scan a realistic surface mesh that will be our perfect continuous ground truth. The main contributions of this paper are:

The definition of a rigorous and informative evaluation protocol to assess the quality of a reconstructed surface based on a perfect continuous ground truth

A LiDAR simulation tool allowing to generate realistic synthetic LiDAR data based on an existing mesh that can serve as the perfect ground truth for our metric

An evaluation of 5 state of the art approaches using the proposed protocol on data simulated with our LiDAR simulator. The paper is organised as follows: Section 2 reviews existing works in surface reconstruction and in particular the papers describing the methods that we will evaluate, but also existing works on the evaluation of such methods. In Section 3, we present our realistic aerial LiDAR simulation tool. Section 4 describes our evaluation protocol. Finally, Section 5 shows the results of our evaluation and Section 6 draws some conclusions and proposes perspectives to this work.

2. RELATED WORKS

2.1 Surface Reconstruction

Here we review existing methods to reconstruct a triangle mesh from point cloud and classify them by the paradigm they use. Methods evaluated in Section 5 are typesetted in bold.

2.1.1 Indicator function: Often used to achieve watertight reconstructions, this class of algorithms proceed by computing a space segmentation. The object itself is defined as the region of space where the labelling equals a certain value. The surface is then computed by finding the changes in the segmentation. A popular approach in this field is **Poisson reconstruction (Kazhdan et al., 2006)**. Their indicator function χ is defined as 1 inside the object and 0 outside. They show that χ convolved with a smoothing filter has to respect Poisson equation (1) where \vec{V} is a vector field depending on point locations and the associated normals.

$$\Delta \tilde{\chi} = \nabla \cdot \vec{V} \quad (1)$$

This differential equation is solved numerically and an adaptation of the *marching cube* algorithm (Lorensen and Cline, 1987) is used to extract a triangle mesh approximating the $\tilde{\chi} = \gamma$ isosurface, γ being the average of $\tilde{\chi}$ at the sample positions. This approach is *screened* in (Kazhdan and Hoppe, 2013) to incorporate additional constraints on sample locations which significantly improves the resulting quality. This implementation also supports two boundary conditions: **Dirichlet** specifies the values that χ needs to take along the boundary of the domain ∂M . Watertightness is then enforced by imposing $\chi = 0$ along ∂M . **Neumann** specifies the values that $\nabla \chi$ needs to take along ∂M . While this boundary condition also allows watertightness, it is less restrictive because it enables the surface to cross ∂M orthogonally.

IM-NET (Chen and Zhang, 2019) is a learning framework which predicts whether any point (x, y, z) is inside or outside the given shape needing to be reconstructed. The input of their network is the 3 coordinates of a point as well as a feature vector that can be computed using *PointNET* (Qi et al., 2017).

Occupancy Networks (Mescheder et al., 2019) presents a similar way of computing the so-called *occupancy function* of the 3D object. However, instead of concatenating a feature vector

to the coordinates of points, they use a batch-sampling strategy. A main advantage of the two latter methods is the arbitrary resolution at which the surface can be extracted.

Recently, (Groueix et al., 2018) proposed a general learning framework dubbed *AtlasNet* to take as input a 3D point cloud or an RGB image. It proceeds by concatenating this data with a sampling of a *patch*, namely the unit square, before passing it to multilayer perceptrons (MLPs) with rectified linear unit (ReLU) nonlinearities producing as output a point cloud of arbitrary resolution. A mesh can be generated by either transferring the connections between vertices of a mesh defined on the patch to their 3D image points or using Poisson Surface Reconstruction (Kazhdan et al., 2006) on a sufficiently dense point cloud. A third solution is to sample a 3D sphere instead of patches.

2.1.2 Volumetric Segmentation: This is a sub-discipline of indicator functions as it consists in giving information about whether a region of space is filled by the object or empty. The data structure can be:

- *the Delaunay Triangulation* of input samples as in (Labatut et al., 2009), (Lafarge and Alliez, 2013), (Caraffa et al., 2016) and (Kolluri et al., 2004).
- *voxels*: (Holenstein et al., 2011) labels them as *free space*, *occupied* or *unknown*. To achieve this, point locations combined with sensor positions enable to compute the ray corresponding to a beam of free space. An interesting feature is that undesirable moving objects such as humans can be erased in the final surface thanks to scans of the same area from different sensor positions.

(Labatut et al., 2009) label as *inside* or *outside* each tetrahedron of the Delaunay triangulation of the point samples. The triangles separating an *empty*-labelled tetrahedron from an *occupied*-labelled one are extracted thanks to a graph-cut optimisation of an energy function defined thanks to the lines of sight (emanating from the vertex and pointing at the laser scanner) and the shape of the triangles.

Similarly, (Caraffa et al., 2016) label as *occupied* or *empty* each tetrahedron t of the Delaunay triangulation T of the point samples. First of all, a set of *mass functions* m_t are computed. Each m_t corresponds to the likelihood of tetrahedron $t \in T$ to be *empty*, *occupied* or if its occupancy is mostly *unknown*. These are computed based visibility priors, making use of sensor positions. Binary labels l_t are attributed to each tetrahedron $t \in T$ minimising an energy function (equation 2). Denoting as $l_T = (l_t)_{t \in T}$ the labelling of each tetrahedron in the triangulation T and by L the labels set, the problem is formulated as:

$$l_T = \arg \min_{l_T \in L^T} \left(E_{data}(l_T) + \lambda E_{prior}(l_T) \right) \quad (2)$$

$$E_{data}(l_T) = \sum_{t \in T} \|l_t - m_t\|^2 \quad (3)$$

$$E_{prior}(l_T) = \sum_{(t_1, t_2) \in T^2} |t_1 \cap t_2| \cdot \|l_{t_1} - l_{t_2}\|^2 \quad (4)$$

Data and *prior* terms (equations 3 and 4) respectively enforce a solution close to the overall mass function and minimise interfaces area. The value of parameter λ balances the two of them.

2.1.3 Signed-distance function: Another way of generating a watertight surface is to compute the signed-distance function f to the surface and to extract its zero-level set. This is the approach chosen in Multi-level Partition of Unity (MPU) (Ohtake et al., 2003) and SSD (Calakli and Taubin, 2011). The latter consists in using a least-square minimisation of an average of several energy functions $\mathcal{E}_{D_i}(f)$, weighted by coefficients λ_i (equation 5).

$$\mathcal{E}_D(f) = \sum_i \lambda_i \mathcal{E}_{D_i}(f) \quad (5)$$

The two main energy functions used are those of equations 6 and 7.

$$\mathcal{E}_{D_0}(f) = \frac{1}{N} \sum_{k=1}^N f(p_k)^2 \quad (6)$$

$$\mathcal{E}_{D_1}(f) = \frac{1}{N} \sum_{k=1}^N \|\nabla f(p_k) - n_k\|^2 \quad (7)$$

Equation 6 is precisely the one enforcing the condition $f(x) = 0$ at point locations (near the surface). Equation 7 comes from the fact that the gradient of the function f represents the normal field where $f(x) = 0$. As a consequence, near the surface, the condition $\nabla f(p_k) = n_k$ should be satisfied. A third energy function involving the Hessian matrix of f enforces that the gradient of f should remain almost constant away from the surface. Following this idea, the surface produced is watertight as long as the function f is continuous.

Recently, *DeepSDF* (Park et al., 2019) has shown how to learn the surface distance field using as input the 3D coordinate of a point as well as a *latent vector* that accounts for the type of shape being at stake.

2.1.4 Unsigned-distance function: (Hornung and Kobbelt, 2006) presents a method for reconstructing large-scale watertight and manifold surfaces using only point locations. They proceed by estimating a confidence map over a pre-defined volumetric grid V . This function $\phi : v \rightarrow c \in [0, 1]$ associates to any voxel, v its confidence c which can be seen as the pseudo-distance to the nearest point location p . The aim is then to minimise the sum of pseudo-distances over a certain set of voxels. An algorithm for extracting the corresponding mesh is also proposed.

2.1.5 Primitive-based: In this field, *PolyFit* (Nan and Wonka, 2017) uses RANSAC (Schnabel et al., 2007) to detect planar segments and refine them. The surface is extracted by combining the optimisation of an objective function which favors data fitting, point coverage and model complexity and the enforcement of watertightness and manifoldness.

(Lafarge and Alliez, 2013) relies on the Delaunay triangulation of input points and the labelling of its tetrahedrons as *empty* or *occupied* but their specificity resides in the extraction of primitives as a pre-processing step, a resampling of the resulting structures and the combination of points from planar regions and unstructured ones in the reconstruction step.

2.1.6 MLS-based: Moving least squares (MLS) was first introduced by Lancaster in (Lancaster, 1979), based on the work conducted by, amongst others, Shepard in (Shepard, 1968). Since then, a tremendous amount of extensions have been added as pointed out by a survey conducted in (Cheng et al., 2008). For instance, (Levin, 2000), (Alexa et al., 2001) and (Levin, 2003) noticeably contributed to the advances in MLS-

based algorithms. As explained in (Cheng et al., 2008), MLS-based algorithms can be roughly classified into two main categories: **Projection MLS** surfaces consists of first computing a projection operator that maps any point of the space onto one belonging to the surface. The surface is then made of the set of stationary points. **Implicit MLS** surfaces requires the computation of a level set function of which the zero isosurface can be extracted.

2.1.7 Refinement: These methods are particularly useful when data is too massive or when it needs to be processed on the fly. As an example, (Allegre et al., 2006) presents an out-of-core algorithm that enables to interactively process point clouds that do not fit into memory. Their way consists in sub-sampling the initial input point cloud P to produce a new, smaller one: P_{rep} . The Delaunay triangulation (DT) of P_{rep} is computed and the *geometric convection* algorithm from (Chaine, 2003) allows to reconstruct a simplified version of the surface implied by P . After dividing P in n regions of equal size such that: $P_1 \cup P_2 \cup \dots \cup P_n = P$, points of each $P_i, i=1, \dots, n$ are inserted in the triangulation and a surface refinement algorithm processes them in order to update the reconstructed surface.

2.2 Surface Reconstruction Evaluation

In order to assess the quality of a reconstruction, there is a need for a ground truth, an input point cloud and a means of calculating the difference between a given output surface and the so-called ground truth. Let us detail the various possibilities that have so far been considered for these three aspects.

2.2.1 Ground Truth: Ground truth could potentially take any surface form, i.e. implicit field, triangle mesh, volumetric segmentation, point set, deformed model, skeleton curve, primitives. However, only two have yet been considered: **triangle mesh** (Ter Haar et al., 2005), (Kazhdan, 2005) and **implicit field** (Berger et al., 2013).

2.2.2 Input Point Cloud: Producing point samples from a surface can be carried out in several ways:

Real laser-based scanning a physical object (or scene) generates a point cloud directly. Such technologies include Time-of-Flight (Lange and Seitz, 2001) and Structured-light (Geng, 2011) devices. Besides, terrestrial or airborne LiDAR (Lohani and Ghosh, 2017) offer the possibility to deal with large areas. The main issue is that no digital ground truth is available.

Image-based technologies like Multi-view stereo (Furukawa and Hernández, 2015) and Structure from motion (Ozyesil et al., 2017) enables to get a 3D model from images which can be the starting point for surface reconstruction. An assessment protocol tackling this particular case has been recently proposed in (Nocerino et al., 2020).

Synthetically sampling a continuous digital model has the advantage of making it possible to fully control the data. In particular, one can generate more realistic data by adding noise, outliers, misalignment, occlusions and setting density. In that field, several procedures have been considered: **random** or **uniform sampling** (Kazhdan, 2005), (Manson et al., 2008), (Süßmuth et al., 2010), **synthetic raytracing** (Hoppe et al., 1996), (Berger et al., 2013) or **z-buffering** (Ter Haar et al., 2005). Of particular interest is the very recent work presented in (Manivasagam et al., 2020). They developed *LiDARsim*: a virtual terrestrial LiDAR platform generating realistic point clouds based on a digital, moving object-free, high quality mesh.

2.2.3 Comparison: As regards to comparing an output reconstruction, three main possibilities have been explored:

Visually: Most of the time, surface reconstruction aims at producing a digital representation as visually similar as possible to a real object. Hence, Poisson (Kazhdan et al., 2006), MPU (Ohtake et al., 2003) and SSD (Calakli and Taubin, 2011) have simply compared models on a visual basis.

Mesh-to-mesh distance computation: This method comes with the advantage of providing a quantitative quality assessment which is independent of any human bias. (Ter Haar et al., 2005), (Kazhdan and Hoppe, 2013) use *Metro* tool (Cignoni et al., 1998) which works as follows: given two meshes (a sampled one \mathcal{M}_s and a target one \mathcal{M}_t), *Metro* samples \mathcal{M}_s and measures the shortest distance from each sample to \mathcal{M}_t . *Metro* then computes the mean distance, the max and the Root Mean Square (RMS) over all samples.

Mesh-to-implicit distance computation: (Berger et al., 2013) chose to use implicit field Ω as ground truth and consequently, they adapted *Metro* methodology in order to compute the distance from a nearly uniform sampling of Ω to the evaluated mesh and vice-versa.

3. AERIAL LIDAR SIMULATOR

Similar to *LiDARsim* (Manivasagam et al., 2020) for terrestrial scan, we developed our own airborne LiDAR simulation platform in order to generate realistic scans of a given environment.

3.1 Virtual environment

We used the open dataset from (ville-eurometropole de Strasbourg, 2018) which was financed by the European Union as part of a FEDER (*Fonds Européen de Développement Régional*). It consists of a 3D mesh of a large area covering the metropole of Strasbourg. It was produced by photogrammetry using high resolution (between 4 and 7cm GSD) oblique imagery acquired by helicopter platform such that it presents details at a higher resolution than typical aerial LiDAR acquisitions. Figure 1 shows a 250m by 250m tile of this mesh.

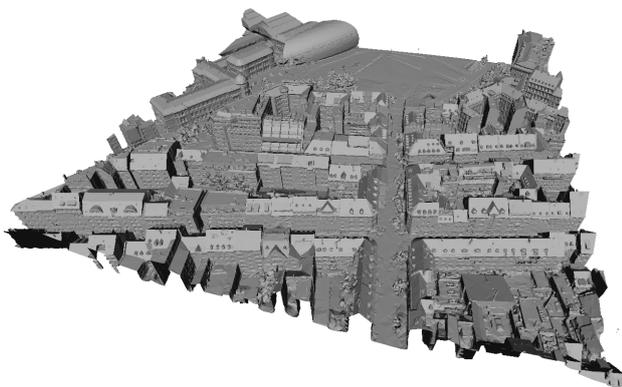


Figure 1. Virtual environment from Strasbourg open dataset

3.2 Scanning process

In this section, we formalize our aerial LiDAR simulator.

3.2.1 Plane trajectory: We use $(O, \vec{e}_x, \vec{e}_y, \vec{e}_z)$ as the global coordinate frame, the one in which mesh vertices coordinates are expressed as detailed in Figure 2. We model the acquisition by a linear trajectory of the LiDAR optical center M moving from $A(x_A, y_A, z_A)$ to $B(x_B, y_B, z_B)$ at constant speed v_0 . We also use a local coordinate frame $(M, \vec{i}, \vec{j}, \vec{k})$ associated to M defined as:

$$\vec{k} = \frac{\overrightarrow{AB}}{\|\overrightarrow{AB}\|}; \quad \vec{j} = \frac{\vec{e}_z \wedge \vec{k}}{\|\vec{e}_z \wedge \vec{k}\|}; \quad \vec{i} = \vec{j} \wedge \vec{k} \quad (8)$$

which ensures that \vec{j} is orthogonal to \overrightarrow{AB} and to the vertical direction \vec{e}_z . Note that \vec{k} is undefined only when \overrightarrow{AB} and \vec{e}_z are colinear which never happens in practice (a plane does not fly vertically). \vec{i} completes the frame so that $(\vec{i}, \vec{j}, \vec{k})$ is right-handed. $M(t)$ moves in a straight line and thus can be defined as:

$$\forall t \in [0, t_B], \overrightarrow{OM}(t) = \overrightarrow{OA} + v_0 t \vec{k} \quad (9)$$

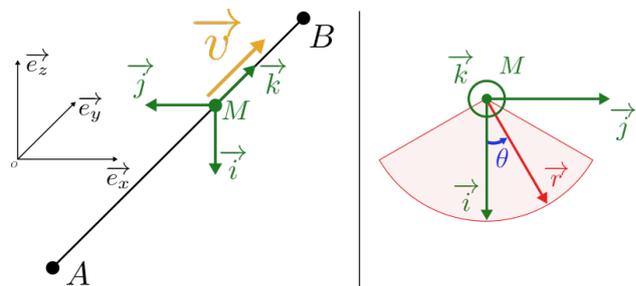


Figure 2. Left: Global frame and flight trajectory. Right: zoom in on the local frame and laser pulse direction. The red-blurred area represents the *field of view*.

3.2.2 Scanning pattern: Among the many possibilities available on the market (zig-zag, elliptical, circular...), we chose to replicate the **parallel line** pattern produced by a rotating mirror mechanism. We denote as \vec{r} the direction of the laser ray. \vec{r} is rotating around \vec{k} at constant angular speed $\omega = \dot{\theta}$. Thus:

$$\vec{r} = \cos(\theta) \vec{i} + \sin(\theta) \vec{j} \quad (10)$$

$$\forall t \in [0, t_B], \theta(t) = \omega t + \theta_0 \quad (11)$$

Such a system is also defined by its *field of view* i.e. the angle range $[\theta_{min}, \theta_{max}]$ to which θ must belong for laser pulses being actually emitted, and by the *pulse rate* (frequency at which pulses are emitted) f_p .

3.2.3 Noise model: The quality of LiDAR data has recently been surveyed by (Warcho\l, 2019). We are especially interested in the accuracy of the point coordinates that a real LiDAR can achieve. Most of studies in this area agree to state that it can be split up in an **altimetric** and a **planimetric** component. Values are obviously influenced by the modernity of the system but they also vary depending on the type of terrain that is considered (bare soil, low grass, forestry) and the flight parameters (altitude, speed). See (Aguilar and Mills, 2008), (Aguilar et al., 2010) for studies on altimetric error and (Toth et al.,

2008), (Vosselman and others, 2008) for planimetric error analysis. As suggested by these contributions, we suppose a **normal distribution of errors**, differentiating them into planimetric $\Delta x, \Delta y$ and altimetric Δz components:

$$\Delta x, \Delta y \sim \mathcal{N}(\mu_{xy}, \sigma_{xy}^2); \quad \Delta z \sim \mathcal{N}(\mu_z, \sigma_z^2) \quad (12)$$

3.2.4 Implementation: We implemented the method described above in C++ using CGAL library (The CGAL Project, 2020). Rays are traced from the virtual aerial station in direction \vec{r} , their exact intersections with the virtual environment are computed and gaussian noise following 12 is added.

4. EVALUATION PROTOCOL

While mesh distances are sufficient to evaluate the reconstruction of closed, completely scanned objects, we consider that for open scenes, this will only assess the quality of α -interpolation for large α 's, or in other terms the filling of larger holes, where the largest errors are expected. The methodology proposed in this paper addresses this issue.

In order to assess surface reconstruction, we first assume the availability of a ground truth triangle mesh \mathcal{M}_{GT} . A point cloud \mathcal{P} representing a realistic sampling of \mathcal{M}_{GT} is then generated using a LiDAR simulator like the one we presented in section 3. Finally, we will denote by \mathcal{M}_E a reconstructed triangle mesh produced with one of the methods to evaluate.

In the general case, **only a limited portion of the ground truth surface \mathcal{M}_{GT} is covered by the acquisition**. If we can expect an algorithm to fill some holes in the data in a reasonable manner, we cannot expect it to recover the shape of the scene far from this covered area. Thus, we define the "reconstructible" part \mathcal{M}_{GT}^α of \mathcal{M}_{GT} at distance α , which can be non connected and have holes, but is still orientable and manifold. To this end, we propose to define an explicit maximum interpolation and extrapolation distance α at which the algorithm will be evaluated. **We compute \mathcal{M}_{GT}^α by removing all triangles of \mathcal{M}_{GT} for which none of its vertices lie closer than α to a point of the sampling \mathcal{P}** (cf algorithm 1). \mathcal{M}_{GT}^α is then the best surface an algorithm can be expected to reconstruct at distance α from the input points. Consequently, this is the surface we will compare every reconstructed mesh with. However, watertight reconstruction methods interpolate surface even where point samples are absent. In order to be impartial with every algorithm prior, we will not evaluate those interpolated mesh parts further away than α from the input data. As a consequence, we also need to compute the sub-mesh \mathcal{M}_E^α of \mathcal{M}_E (applying algorithm 1) containing only triangles closer to \mathcal{P} than a distance α . This is not limiting as in practice, it is an easy post processing step that the user can choose to perform if he does not require a fully watertight mesh.

4.1 Sampling

Once \mathcal{M}_{GT}^α and \mathcal{M}_E^α have been computed, we need a way to measure the error between them. We consider that computing error metrics based on distances between vertices of one mesh and triangles of the other is biased, as different algorithm can produce triangles of very different size. Thus we propose to perform a Poisson-Disk sampling of the triangles, which guarantees an even distribution of samples so that we can consider that each sample represents the same amount of surface area. We choose a Poisson disk radius R significantly smaller than α . We denote by $\mathcal{P}_{\mathcal{M}_{GT}^\alpha}^R$ and $\mathcal{P}_{\mathcal{M}_E^\alpha}^R$ the Poisson-Disk samplings of radius R of respectively \mathcal{M}_{GT}^α and \mathcal{M}_E^α .

```

Data:  $\mathcal{M}, \mathcal{P}, \alpha$ 
// Browse vertices of  $\mathcal{M}$ :
for each vertex  $v \in \text{vertices}(\mathcal{M})$  do
  compute  $d = d(v, \mathcal{P}) = \min_{P_i} d(v, P_i)$ 
  if  $d < \alpha$  then
    // Browse all triangles incident to  $v$ :
    for each triangle  $t \in \{T : v \in T\}$  do
      if  $t \notin \mathcal{M}^\alpha$  then
        // Browse vertices adjacent to  $t$ :
        compute  $(v_0, v_1, v_2) = \text{vertices}(t)$ 
        for each vertex  $v_t \in \{v_0, v_1, v_2\}$  do
          if  $v_t \notin \mathcal{M}^\alpha$  then
            | add  $v_t$  to  $\mathcal{M}^\alpha$ 
          end
        end
        add  $t$  to  $\mathcal{M}^\alpha$ 
      end
    end
  end

```

Algorithm 1: Reconstructible part of mesh \mathcal{M} computation

4.2 Quality measure

Thanks to notations introduced in Section 4.1 and using the point-to-mesh distance $d(p, \mathcal{M})$ defined in equation 13, bi-directional distances can be computed (equations 14, 15).

$$\forall p \in \mathbb{R}^3, d(p, \mathcal{M}) = \min_{q \in \mathcal{M}} d(p, q) \quad (13)$$

$$\text{Mean Precision: } \frac{1}{|\mathcal{P}_{\mathcal{M}_E^\alpha}^R|} \sum_{p \in \mathcal{P}_{\mathcal{M}_E^\alpha}^R} d(p, \mathcal{M}_{GT}^\alpha) \quad (14)$$

$$\text{Mean Recall: } \frac{1}{|\mathcal{P}_{\mathcal{M}_{GT}^\alpha}^R|} \sum_{p \in \mathcal{P}_{\mathcal{M}_{GT}^\alpha}^R} d(p, \mathcal{M}_E^\alpha) \quad (15)$$

Precision measures how close points from the reconstructed mesh are to the ground truth and *Recall* indicates how well the ground truth surface is recovered by the reconstruction. While these terms are named in analogy with the machine learning community metrics, it is important to note that they do not measure a ratio of relevant information but a distance and thus the lower, the better.

Finally, our proposed metric is composed of the two curves of *Precision* and *Recall* for a range of α values. Note that the case where $\alpha \rightarrow \infty$ corresponds to computing the distances on samplings of the raw meshes \mathcal{M}_{GT} and \mathcal{M}_E , without applying algorithm 1. The resulting curves will both indicate the interpolation quality of the algorithm for small α values and the extrapolation/hole filling quality for larger α values.

5. EVALUATION

Algorithms presented in Section 2.1 have been run on a point cloud we generated using the LiDAR simulator described in section 3 and the virtual environment displayed in Figure 1. Resulting surfaces are shown on Figure 3.

5.1 Experimental parameters and methodology

The values used to generate the point set on which we evaluated the various algorithms are to be found in Table 1. As regards to the evaluation part, we set to $R = 0.3m$ the Poisson-Disk sampling radius. This parameter is chosen by considering the trade-off between sampling density and computational time. As every algorithm we assessed can be tuned, we first studied the performance of each of them for a small number of α 's,

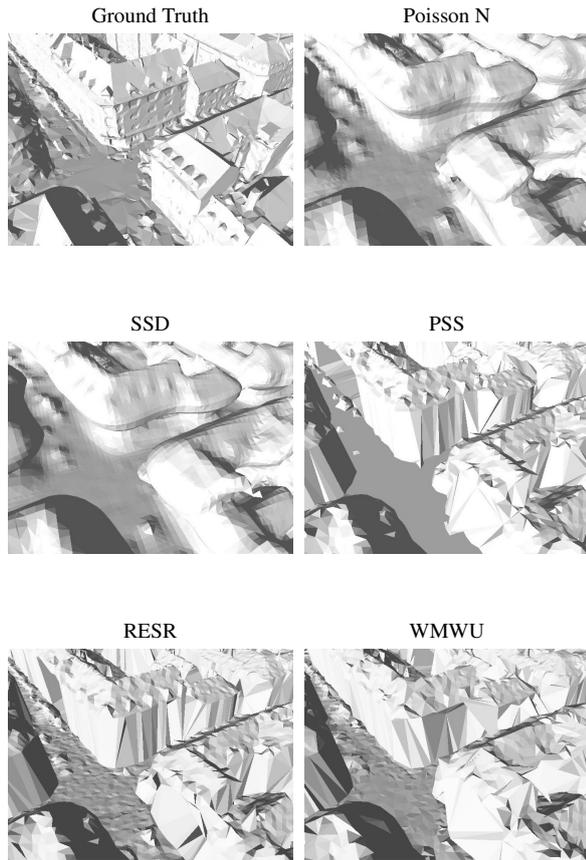


Figure 3. Result of the different algorithms evaluated on a crossroad of the scene

Symbol	Value	Unit	Description
h	1 000	m	Flying altitude
v_0	60	$m.s^{-1}$	Flying speed
ω	150	Hz	Angular speed
$\Delta\theta$	40	$(^\circ)$	Field of view
f_p	400 000	Hz	Pulse frequency
σ_{xy}	0.13	m	Planimetric error
σ_z	0.05	m	Altimetric error

Table 1. Values of experimental parameters we used.

changing some parameter values. We then selected the best version of each algorithm and carried out the full evaluation for which the results are presented in section 5.2.

5.2 Quantitative results

We plotted the mean precision on Figure 4 and mean recall on Figure 5 (following equations 14 and 15) as a function of α .

(Labatut et al., 2009) (RESR for *Robust and efficient surface reconstruction*) achieves the best performance both in terms of precision and recall and for any value of α . Additional information provided by sensor positions is certainly helping but we can assume they are making the best use of it, in comparison to (Caraffa et al., 2016).

Poisson (Kazhdan and Hoppe, 2013) has been evaluated using both boundary conditions (see Section 2.1). Results confirm that *Neumann* (Poisson N) better reconstruct open scenes as it enables the surface to extend out to the boundary of the domain. In contrast, *Dirichlet* (Poisson D) enforces that every edge should be shared by two triangles. Consequently, an un-

desirable closure of the surface from the bottom leads to a poor precision, but only for high α 's. We now see how important it is to differentiate between global and local evaluation.

(Caraffa et al., 2016) (WMWU for *Watertight Mesh With Uncertainties*) evaluation shows a comparable result as those provided in the original article. This method provides good results regarding the recall metrics while precision quickly worsens when α increases. The main explanation is the same as for Poisson with Dirichlet boundary condition: the surface is artificially closed to ensure *hard* watertightness, leading a great amount of surface to lie away from the ground truth.

(Calakli and Taubin, 2011) (SSD for *Smooth signed distance*) curve logically follows the one of Poisson N, as both methods use smooth basis functions to solve their equation.

(Lafarge and Alliez, 2013) (PSS for *Point set structuring*) could have been expected to perform better since it is based on primitive detection and an urban scene is full of planar regions. However, it is quite sensitive to missing data and facades of the buildings are mostly out of reach for a parallel line scanning pattern LiDAR.

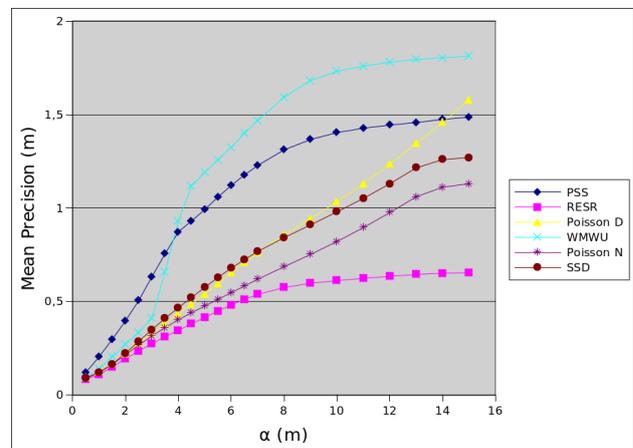


Figure 4. Mean Precision

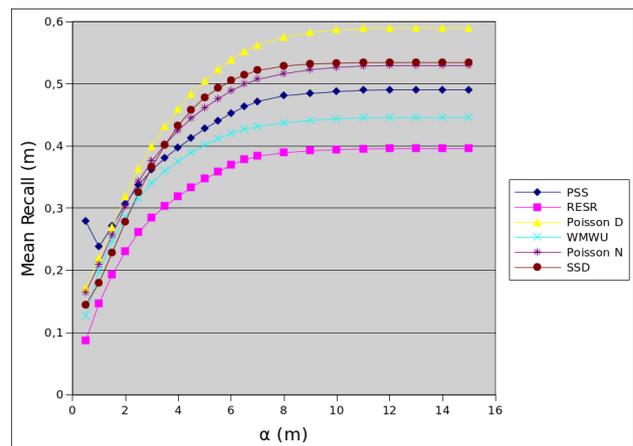


Figure 5. Mean Recall

6. CONCLUSION AND PERSPECTIVES

6.1 Conclusion

Surface mesh reconstruction from remote sensing point clouds is a challenging task that becomes more and more important

as data acquisition starts from a purely vertical perspective. Our evaluation shows important differences between the various state of the art approaches. In this paper, we argue that the most important characteristic of a surface mesh reconstruction is the quality of the interpolation that it performs between the input points, and that this should be measured as a function of the distance at which we expect the algorithm to interpolate the surface. The resulting metric, taking the form of curves indicating the evolution of several quality criteria as a function of maximum interpolation distance are very informative on the qualities of the reconstruction algorithm. Our study shows that (Labatut et al., 2007), while quite old, remains a very good choice. The very popular Poisson method (Kazhdan et al., 2006) is efficient and scales well but does not preserve sharp features that are very present in our urban evaluation scene.

6.2 Perspectives

From the starting point of this work, many perspectives can be of interest. We will consider adding to our evaluation Deep Learning based surface mesh reconstruction methods, which problem is becoming popular in the computer vision community. An obvious continuation of this work would be to turn it into an open benchmark by extending to more complex scanning geometries (zig-zag, elliptic,...) but also other perspectives, in particular terrestrial, drone, or satellite platforms. An important extension would be to also handle multiple echos by intersecting a thin cone instead of a single ray with the virtual scene. An ambitious perspective would be to extend our evaluation protocol in order to address real data with a controlled repercussion on our evaluation metric, assuming that we have a denser/more accurate point cloud to serve as ground truth than the one used for reconstruction. This might prove very difficult because as advocated in this paper, a major aspect of surface reconstruction is its ability to interpolate a surface where data is missing, and any real acquisition of a real scene will have missing data making it impractical to evaluate this important aspect. Finally, applying surface mesh reconstruction to very large data sizes (typically billions of 3D points) has become a major challenge in remote sensing with the rise of sensor and storage capabilities. In this context, assessing the capacity of the methods to scale up (both in terms of computing time and memory footprint) would be an interesting avenue.

REFERENCES

- Aguilar, F. J., Mills, J. P., Delgado, J., Aguilar, M. A., Negreiros, J. G., Pérez, J. L., 2010. Modelling vertical error in LiDAR-derived digital elevation models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(1), 103 – 110.
- Aguilar, F., Mills, J., 2008. Accuracy assessment of LiDAR-derived digital elevation models. *The Photogrammetric Record*, 23, 148 – 169.
- Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D., Silva, C. T., 2001. Point set surfaces | Proceedings of the conference on Visualization '01.
- Allegre, R., Chaine, R., Akkouche, S., 2006. A Dynamic Surface Reconstruction Framework for Large Unstructured Point Sets. 11.
- Berger, M., Levine, J. A., Nonato, L. G., Taubin, G., Silva, C. T., 2013. A benchmark for surface reconstruction. *ACM Transactions on Graphics*, 32(2), 1–17. <https://dl.acm.org/citation.cfm?doid=2451236.2451246>.
- Berger, M., Tagliasacchi, A., Seversky, L. M., Alliez, P., Guennebaud, G., Levine, J. A., Sharf, A., Silva, C. T., 2017. A Survey of Surface Reconstruction from Point Clouds. *Comput. Graph. Forum*, 36(1), 301–329. <https://doi.org/10.1111/cgf.12802>.
- Calakli, F., Taubin, G., 2011. SSD: Smooth signed distance surface reconstruction. *Computer Graphics Forum*, 30, Wiley Online Library, 1993–2002. Issue: 7.
- Caraffa, L., Brédif, M., Vallet, B., 2016. 3D Watertight Mesh Generation with Uncertainties from Ubiquitous Data. S.-H. Lai, V. Lepetit, K. Nishino, Y. Sato (eds), *Computer Vision – ACCV 2016*, Lecture Notes in Computer Science, Springer International Publishing, 377–391.
- Chaine, R., 2003. A geometric convection approach of 3-D reconstruction. *Proceedings of the 2003 Eurographics/ACM SIG-GRAPH symposium on Geometry processing*, SGP '03, Eurographics Association, Aachen, Germany, 218–229.
- Chen, Z., Zhang, H., 2019. Learning implicit fields for generative shape modeling. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5939–5948.
- Cheng, Z.-Q., Wang, Y.-Z., Li, B., Xu, K., Dang, G., Jin, S.-Y., 2008. A survey of methods for moving least squares surfaces. *Proceedings of the Fifth Eurographics / IEEE VGTC conference on Point-Based Graphics*, SPBG'08, Eurographics Association, Los Angeles, CA, 9–23.
- Cignoni, P., Rocchini, C., Scopigno, R., 1998. Metro: measuring error on simplified surfaces. *Computer graphics forum*, 17, Wiley Online Library, 167–174. Issue: 2.
- Furukawa, Y., Hernández, C., 2015. Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2), 1–148. Publisher: Now Publishers Inc. Hanover, MA, USA.
- Geng, J., 2011. Structured-light 3D surface imaging: a tutorial. *Advances in Optics and Photonics*, 3(2), 128–160.
- Groueix, T., Fisher, M., Kim, V. G., Russell, B. C., Aubry, M., 2018. A Papier-Mâché Approach to Learning 3D Surface Generation. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Holenstein, C., Zlot, R., Bosse, M., 2011. Watertight surface reconstruction of caves from 3D laser data. *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 3830–3837.
- Hoppe, H., Deroose, T., Duchamp, T., McDonald, J., Stuetzle, W., 1996. Surface reconstruction from unorganized points. *SIG-GRAPH Computer Graphics*, 26(2), 71-78. *Computer Graphics (Proc. SIGGRAPH 86)*, 20.
- Hornung, A., Kobbelt, L., 2006. Robust reconstruction of watertight 3 d models from non-uniformly sampled point clouds without normal information. *Symposium on geometry processing*, Citeseer, 41–50.
- Kazhdan, M., 2005. Reconstruction of solid models from oriented point sets. *Proceedings of the third Eurographics symposium on Geometry processing*, Eurographics Association, 73.

- Kazhdan, M., Bolitho, M., Hoppe, H., 2006. Poisson surface reconstruction. *Proceedings of the fourth Eurographics symposium on Geometry processing*, Eurographics Association, 61–70.
- Kazhdan, M., Hoppe, H., 2013. Screened poisson surface reconstruction. *ACM Transactions on Graphics*, 32(3), 1–13.
- Khatamian, A., Arabnia, H., 2016. Survey on 3D Surface Reconstruction. *Journal of Information Processing Systems*, 12, 338–357.
- Kolluri, R., Shewchuk, J. R., O'Brien, J. F., 2004. Spectral Surface Reconstruction from Noisy Point Clouds. *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '04, ACM, New York, NY, USA, 11–21. event-place: Nice, France.
- Labatut, P., Pons, J.-P., Keriven, R., 2007. Efficient Multi-View Reconstruction of Large-Scale Scenes using Interest Points, Delaunay Triangulation and Graph Cuts. *2007 IEEE 11th International Conference on Computer Vision*, 1–8. ISSN: 2380-7504.
- Labatut, P., Pons, J.-P., Keriven, R., 2009. Robust and efficient surface reconstruction from range data. *Computer graphics forum*, 28, Wiley Online Library, 2275–2290. Issue: 8.
- Lafarge, F., Alliez, P., 2013. Surface reconstruction through point set structuring. *Proc. of Eurographics*, Girona, Spain.
- Lancaster, P., 1979. Moving Weighted Least-Squares Methods. B. N. Sahney (ed.), *Polynomial and Spline Approximation: Theory and Applications*, NATO Advanced Study Institutes Series, Springer Netherlands, Dordrecht, 103–120.
- Lange, R., Seitz, P., 2001. Solid-state time-of-flight range camera. *IEEE Journal of Quantum Electronics*, 37(3), 390–397.
- Levin, D., 2000. The Approximation Power Of Moving Least-Squares. *Mathematics of Computation*, 67.
- Levin, D., 2003. Mesh-Independent Surface Interpolation. *Geometric Modeling for Scientific Visualization*, 3.
- Lohani, B., Ghosh, S., 2017. Airborne LiDAR technology: a review of data collection and processing systems. *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences*, 87(4), 567–579. Publisher: Springer.
- Lorensen, W. E., Cline, H. E., 1987. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, ACM, New York, NY, USA, 163–169.
- Manivasagam, S., Wang, S., Wong, K., Zeng, W., Sazanovich, M., Tan, S., Yang, B., Ma, W.-C., Urtasun, R., 2020. LiDARsim: Realistic LiDAR Simulation by Leveraging the Real World. *arXiv:2006.09348 [cs]*. arXiv: 2006.09348.
- Manson, J., Petrova, G., Schaefer, S., 2008. Streaming Surface Reconstruction Using Wavelets. *Proceedings of the Symposium on Geometry Processing*, SGP '08, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 1411–1420. event-place: Copenhagen, Denmark.
- Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A., 2019. Occupancy networks: Learning 3d reconstruction in function space. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4460–4470.
- Nan, L., Wonka, P., 2017. Polyfit: Polygonal surface reconstruction from point clouds. *Proceedings of the IEEE International Conference on Computer Vision*, 2353–2361.
- Nocerino, E., Stathopoulou, E. K., Rigon, S., Remondino, F., 2020. Surface Reconstruction Assessment in Photogrammetric Applications. *Sensors*, 20(20).
- Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., Seidel, H.-P., 2003. Multi-level partition of unity implicits. *ACM Transactions on Graphics*, 22(3), 463–470.
- Ozyesil, O., Voroninski, V., Basri, R., Singer, A., 2017. A survey of structure from motion. *arXiv preprint arXiv:1701.08493*.
- Park, J. J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S., 2019. DeepSDF: Learning continuous signed distance functions for shape representation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 165–174.
- Qi, C. R., Su, H., Mo, K., Guibas, L. J., 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. 652–660.
- Schnabel, R., Wahl, R., Klein, R., 2007. Efficient RANSAC for point-cloud shape detection. *Computer graphics forum*, 26, Wiley Online Library, 214–226. Issue: 2.
- Shepard, D., 1968. A two-dimensional interpolation function for irregularly-spaced data. *Proceedings of the 1968 23rd ACM national conference*, ACM '68, Association for Computing Machinery, New York, NY, USA, 517–524.
- Süßmuth, J., Meyer, Q., Greiner, G., 2010. Surface reconstruction based on hierarchical floating radial basis functions. *Computer Graphics Forum*, 29, Wiley Online Library, 1854–1864. Issue: 6.
- Ter Haar, F., Cignoni, P., Min, P., Veltkamp, R., 2005. A comparison of systems and tools for 3D scanning. *3D Digital Imaging and Modeling: Applications of Heritage, Industry, Medicine and Land*.
- The CGAL Project, 2020. *CGAL User and Reference Manual*. 5.2 edn, CGAL Editorial Board.
- Toth, C., Paska, E., Brzezinska, D., 2008. Quality assessment of lidar data by using pavement markings. *Proceedings of the ASPRS Annual Conference, Portland, Oregon, USA*, 28.
- Van Kreveland, M., Van Lankveld, T., Veltkamp, R. C., 2013. Watertight scenes from urban lidar and planar surfaces. *Computer Graphics Forum*, 32, Wiley Online Library, 217–228. Issue: 5.
- ville-eurometropole de Strasbourg, 2018. odata3d_pm3d dataset.
- Vosselman, G., others, 2008. Analysis of planimetric accuracy of airborne laser scanning surveys. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37(3a), 99–104.
- Warchoł, A., 2019. The concept of LiDAR data quality assessment in the context of BIM modeling. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*.