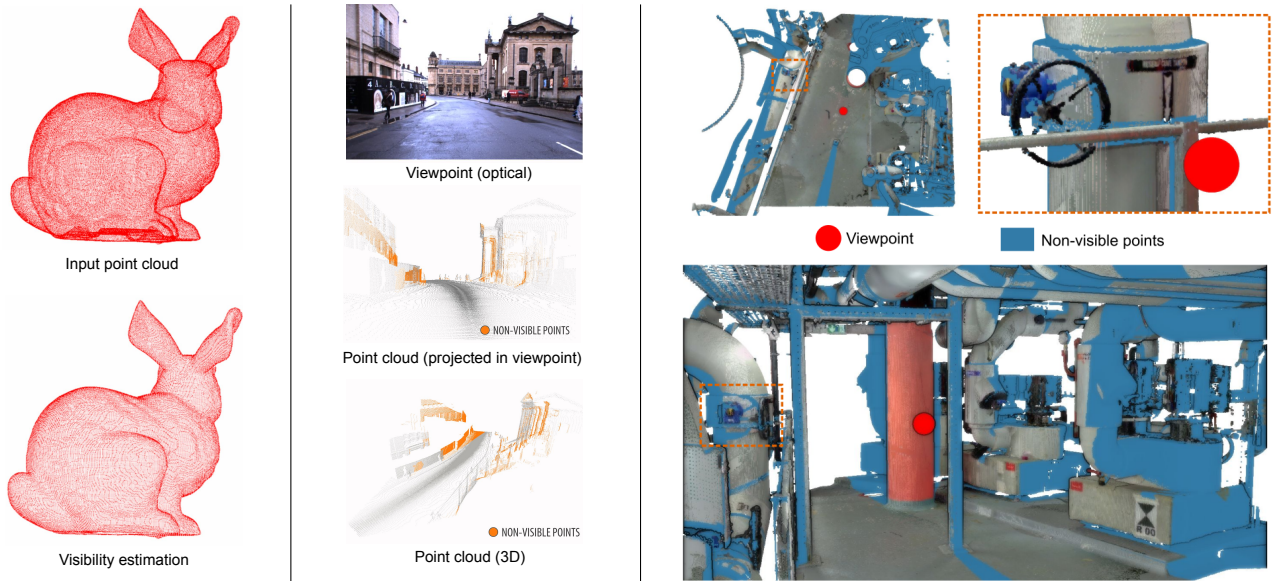
IRIS-VIS: a New Dataset for Visibility Estimation in an Industrial Environment

Flavien Armangeon[1,2] Thibaud Ehret[1] Enric Meinhardt-Llopis[1]
Rafael Grompone von Gioi[1] Guillaume Thibault[2] Marc Petit[2] Gabriele Facciolo[1]
[1] Université Paris-Saclay, CNRS, ENS Paris-Saclay, Centre Borelli, France
[2] EDF R&D (Electricité de France), France

https://centreborelli.github.io/iris-vis



Katz *et al*. [9] PCVD [2] IRIS-VIS (ours)

Figure 1. Katz *et al*. [9] as well as its follow-up work [13] introduce methods for visibility estimation on synthetic object scenes without quantitative analysis due to the lack of ground truth. Biasutti *et al*. [2] focus on outdoor scenes and provide PCVD, the first dataset for point visibility estimation manually annotated on real data (by comparing the RGB image with the projected point cloud) for three different viewpoints. In this work, we introduce IRIS-VIS, a dataset for visibility estimation on real complex indoor data. This indoor scene enables challenging visibility estimation due to the presence of small and detailed industrial equipment. A ground truth can be generated from any viewpoint inside the scene thanks to the provided CAD model.

Abstract

Point cloud visibility estimation is fundamental as it is useful for many computer vision applications including surface reconstruction, 3D segmentation from paired images and point densification. Previous works showed outstanding results on simple object and outdoor datasets. However, unlike the previously studied scenes, the most challenging environments are those providing a high amount of object points in the same direction, typically in complex indoor scenes. In this kind of environments, due to the lack of real data ground truth, quantitative analysis are either missing or based on simulated data. In this work, we present IRIS-VIS (Industrial Room In Saclay - VISibility), a new dataset for point visibility estimation in an indoor environment. It is a high complexity scene due to the large variety in the shape, size and orientation of the objects. To our knowledge, this is the first dataset on real indoor data providing a dense LiDAR station-based point cloud along with a well-fitted CAD model. The latter is useful to compute automatically, quickly and accurately the visibility from any given viewpoint, enabling evaluations under infinite conditions. We propose new metrics for the visibility estimation task and evaluate state-of-the-art methods in both sparse and dense conditions with the proposed dataset.

1. Introduction

Linking the visible objects to a given viewpoint is a crucial step for 3D tasks. For example, it is required for cloud rendering and 2D-to-3D segmentation masks projection.

Due to the discrete nature of a point cloud, all points can be seen from a given point of view. As illustrated in Figure 2, the objective of point visibility estimation is to remove the points that are not visible in the real scene from this point of view. More explicitly, the problem can be formulated as a binary classification where every point receives a predicted label "visible" or "hidden".

The particularity of complex indoor environments in the visibility estimation context, compared to the available outdoor datasets, is the higher quantity of objects behind each other. This is why some methods, designed specifically for simple outdoor scenes, tend to perform poorly indoor [2]. Moreover, the main difference between classic indoor scenes or object scenes with the so aptly named "complex" indoor environments, is the diversity in the shapes, size and orientations of the objects. In particular, we introduce IRIS-VIS (Figure 1), the first dataset in industrial environment providing a dense LiDAR station-based point cloud and a handmade CAD model. This challenging scene provides many thin and large pipes, masking partial or total parts of the equipment in multiple directions. It also contains many small and detailed objects such as valves, gauges, railings, that can be difficult to manage. We think that this dataset will be useful for many computer vision tasks including visibility estimation.

The specificity of our dataset, compared to the previous ones is: 1) The real input point cloud including the specific noise from acquisitions that cannot be fully reproduced on synthetic data. 2) The complexity of the scene induced by the industrial equipment and piping. 3) The well-paired CAD model which we can use as the reference for the visibility. 4) The automatic and fast ground truth computation from any given viewpoint.

Experiments on our dataset show that the state-of-the-art methods struggle to estimate the visibility on some challenging locations, specifically on objects of non-regular shapes, at the borders of the visibility range.

Our contributions can be summarized as follows:

- IRIS-VIS: a new dataset for visibility estimation in a complex indoor industrial environment on real data, coupled with a handmade CAD model.
- The first quantitative evaluations for point visibility estimation on real indoor data. New metrics, more challenging for this task, are also designed.
- A framework to generate a ground truth from any viewpoint and to reproduce the results.

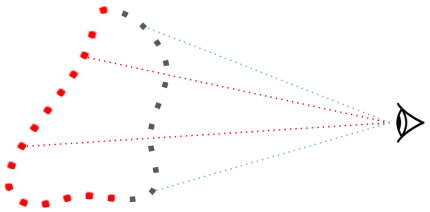


Figure 2. Visibility estimation problem: because of the discrete nature of point clouds, Red points can be seen from the viewpoint while they should be hidden when considering the structure of the object.

2. Related Work

Some previous works address the problem of point visibility estimation. However, most of them focus on simple objects or outdoor datasets, which are not as challenging due to the low variability in the object-viewpoint distance. In particular, it is important to remark that, as far as we know, no quantitative evaluations are given on real complex indoor scenes.

Nature of the datasets. First, we should point out that a ground truth for point visibility can be deduced from any surface by simple raycasting or rasterization strategies. Datasets for visibility estimation can be separated into two main categories depending on whether the input point cloud is simulated or real. Synthetic clouds usually come from point sampling on 3D models or are generated from RGB, RGB-D images [12, 18] with photogrammetric and alignment tools, such as SfM [17], which output a sparse cloud, and can be coupled with a point densification method [11]. In both cases, the ground truth is inherent, obtained by construction. However, the main drawback of these data is that the acquisition noise, useful to evaluate the methods in real conditions, cannot be reproduced. On the other hand, for datasets with real point clouds, coming from acquisitions, the ground truth needs more work to obtain. It can be human annotated by comparing the projected cloud with RGB images such as in [2], but this process is very time-consuming. The visibility can also be deduced from a handmade 3D model or with automatic surface reconstruction on a denser point cloud depending on the tolerance. In our case, the point cloud comes from real station-based LiDAR acquisitions and the visibility is given by the CAD model.

Scene. The structure of the scene is very important for challenging visibility estimation. Some methods [8, 9, 13] work on simple object scenes without ground truth. In outdoor, Biasutti *et al*. published PCVD [2], the first dataset on real data with human annotated visibility. However, due to the structure of these street scenes and the LiDAR acquisition located on a car, this dataset is quite simple because the points-to-view distance is not very variable. In real indoor scenes, no quantitative evaluation is given. Many data com-

ing from 3D models or acquisitions could be processed as previously described. However, most of the available data is located in classic and usual rooms such as in [1, 3, 4, 22]. Due to the industrial environment containing various and large piping as well as small and detailed equipment, IRIS-VIS stands out for its complexity. Table 1 compares our dataset with PCVD, the only previously available dataset specifically designed for visibility estimation.

	PCVD [2]	IRIS-VIS (ours)
# Views	3	Unlimited
# Points	10^6	2×10^9
Scene	Streets (outdoor)	Industrial (indoor)

Table 1. Quantitative information on visibility estimation datasets.

Methods. Two categories emerge from the literature: methods based on surface reconstruction and those that are not. Both categories contain traditional and deep learning based approaches. Some methods [2, 8, 9, 13] present simple and intuitive ideas without mesh computation. Other methods like [15, 19] estimate the visibility in classic indoor scenes using convolutional neural networks on depth maps provided by RGB-D images datasets [12, 18]. Surface reconstruction approaches include geometric triangulation methods [5, 10], continuous volumetric radiance fields (NeRF [14] and variants) and implicit function based on kernels estimation such as NKF [21] and NKSR [6]. Note that some of these methods require the normals as input in addition to the point cloud. Such normals can be estimated using classic tools. NKSR also outputs the refined estimated normals of the points in addition to the surface.

3. The IRIS-VIS dataset

IRIS-VIS provides a dense LiDAR point cloud coupled with a handmade CAD model that could be useful for many different tasks including 3D visibility estimation. We first present the data and their challenging environment, specifying the way they were obtained. Then, we discuss their reliability for our task and describe the automatic processing that allows us to compute the ground truth from any viewpoint.

3.1. Scene and raw data

Data were acquired in the Thermofrigopumps and Energy Storage Tanks (TEST) scene, a large environment of 530 square meters (Figures 3 and 4). It contains very detailed objects with non-regular shapes and piping of various sizes in many different directions. Figure 5 presents the given point cloud and CAD model illustrating their fine pairing. Acquisition and modeling were done following the processing steps discussed by Hullo *et al*. [7].

Figure 3. Thermofrigopumps and Energy Storage Tanks (TEST) scene. It includes large and thin rotating pipes with different kinds of junctions, such as T, U and reductions, as well as small and complex equipment with detailed shapes such as valves and gauges.

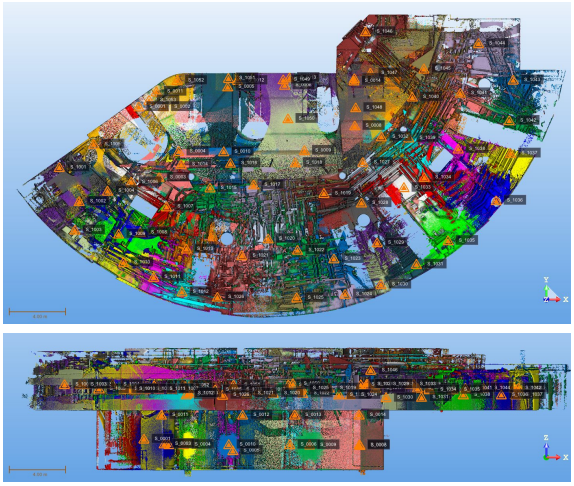


Figure 4. LiDAR point clouds acquired from 67 stations represented by triangles. The merged point cloud contains more than 2.1 billion points and covers the whole TEST scene ($530 m^2$).

Point cloud. The cloud comes from multiple station-based LiDAR acquisitions, whose positions are illustrated in Figure 4. Classic preprocessing steps have been applied to align and merge the point clouds, and remove the outliers. The resulting cloud is of top-quality regarding its very high density, low noise and accuracy of $3\,\sigma = \pm 2\,cm$. The cloud contains line patterns that are specific to LiDAR acquisitions. Also, the density is variable according to the number of stations from where the points are visible and their distances, leading to holes where objects cannot be seen. These observations can be seen in Figure 6.

CAD model. The CAD model is reconstructed close to the point cloud following a classic segmentation and adjustment procedure in a semi-automatic way. The tolerance for the reconstruction is $2.56\,\sigma = \pm 2\,cm$, which we consider to be of good quality given the complexity of the TEST scene. Every type of equipment is represented by a standard 3D model, adjusted to be as close as possible to the reality. Particular attention has been given to the pipes to represent the delicate junctions, elbows of different sizes with respect to the point cloud. The CAD model and the underlying mesh are illustrated in Figure 5.

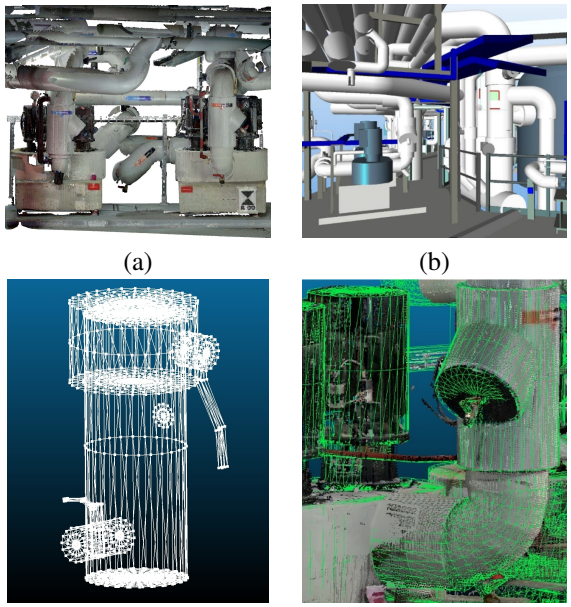


Figure 5. Raw data. (a) Point cloud. (b) CAD model. (c) Example of the mesh given by the CAD model containing pipes and valves. (d) Mesh and point cloud.

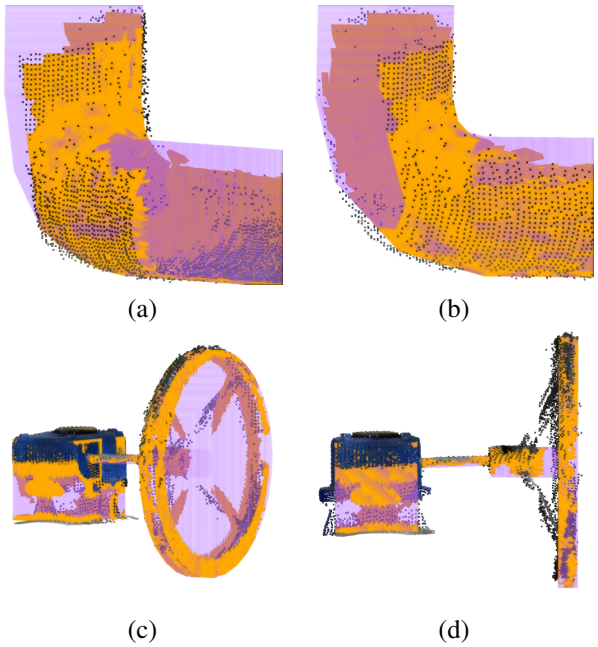


Figure 6. Initial mesh (violet), corrected mesh (orange) and points on an elbow (a, b) and a valve (c, d). LiDAR acquisition patterns and noise can be seen in (a) and (b). Note that the points located at the angle of the elbow are well covered by the mesh despite the appearance. The pink color is the combination of violet and orange. See Section 3.2 for more details.

Points-mesh consistency. Despite its quality, the CAD model still presents inconsistencies with the point cloud through several aspects. First, some minor objects, most of them temporarily present in the scene during acquisitions, are visible in the point cloud but not included in the CAD model. Secondly, the 3D templates of the equipment do not represent completely the complex shapes of the objects while they are well reflected in the point cloud (e.g. Figure 6 the handwheel is represented by a disk). Third, many holes in the point cloud, due to the lack of visibility from the stations, have been filled by the CAD modeling. Also, some points can be slightly inside or outside the associated mesh.

3.2. Data processing for point visibility

As the CAD model is reconstructed close to the point cloud, we can use the derived mesh as the reference for the visibility. However, as mentioned previously, the model presents inconsistencies with the point cloud. Using it directly for visibility estimation, would lead to miss-classification in the ground truth. To mitigate this problem, we process the data automatically in three steps:

1. Points-to-mesh pairing. We remove the points that are too far from the mesh (namely 3cm). Given the precision of the CAD model, this step removes objects, or parts, that are either not modeled or not well-aligned with the model. The resulting point cloud becomes the input for visibility estimation.

2. Mesh-to-points pairing. Triangles too far from the point cloud are discarded (3cm). Note that, as shown in Figure 5, the mesh triangles are large compared to the size of the objects, leading to a bad pairing because most of the triangles are kept. To handle this, we divide them into smaller triangles before elimination, with care so as not changing the global shape of the mesh. All edges that are longer than a threshold (here 2cm) are iteratively split into two equals parts. Figure 6 compares the mesh before and after mesh-to-points pairing.

3. Raycasting. The ground truth is deduced from the mesh by raycasting from the viewpoint to every point in the input point cloud. As shown in Figure 7, each point is set as visible or hidden depending on the distance to the first hit-point along the ray (namely 3cm).

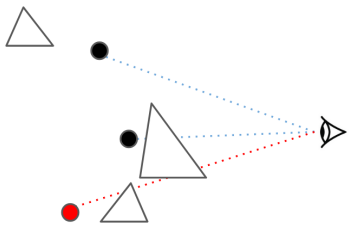


Figure 7. Visibility condition for the ground truth. The points and triangles are respectively part of the input point cloud and the mesh. Rays are cast from the viewpoint to every points. The points that are close enough to the first hit-point are set as visible (black), the others are set as hidden (red).

Threshold values were chosen in accordance with the tolerances on the raw data and qualitative visualizations of the ground truth. We see in Figure 6 that the pairing correction fixes the discrepancies between the point cloud and the CAD model to the previously given confidences. In particular, the holes in the cloud due to the occlusions from the acquisition stations and the fine structures, such as valves, are matched with the corrected mesh.

Density	Sparse - 2%	Dense - 10%
# Points	13 534 572	67 672 868
% Visible	34.99	35.00

Table 2. Quantitative information on the experiment clouds. We use three clouds, covering in total 30% of the whole TEST scene ($\sim 180m^2$), and nine viewpoints whose ground truth has been computed following Section 3.2. Two point densities are sampled from the extremely dense IRIS-VIS point cloud.

4. Methods for point visibility estimation

In this section, we present the state-of-the-art methods for point visibility evaluated on the proposed dataset.

4.1. Point visibility using mesh reconstruction

The most intuitive solution to visibility estimation is to render the point cloud, similar to how the ground truth is computed for our dataset in Section 3.2. This however requires knowing the surface associated to the points, something that is usually not available in practice.

Song *et al*. proposed Vis2Mesh [19], a surface reconstruction method based on the visibility estimation from multiple camera views and graph-cut merging to compute the mesh. In the same way as VisibNet [15], the visibility is predicted using deep convolutional neural networks taking as input the projected depth maps over the 2D views. These networks, variants of the U-net [16], are trained with the supervision of visibility maps obtained from a 3D model. Vis2Mesh can be used in two different ways to mitigate our task: using the neural networks to estimate the visibility in multiple directions around the viewpoint or by simple raycasting over the resulting mesh. In this work, we chose the more practical second option.

Neural Kernel Surface Reconstruction (NKSR), proposed by Huang *et al*. [6], is one of the state-of-the-art methods for local mesh estimation that focuses on generalization capabilities and noise robustness. Similar to [21], the surface is encoded as the zero level set of an implicit function defined as the weighted sum of kernels, i.e. positive-definite basis functions. To improve the scalability while estimating precise models, Huang *et al*. use a sparse voxel hierarchy to support the kernels. This voxel hierarchy is predicted, from the input point cloud P and the set of normals N associated to the point cloud, at L levels using a sparse convolutional neural network. The encoder of this network is based on [20]. Such an implicit function is thus defined as

$$f\left(x|P,N\right) = \sum_{i,l} \alpha_i^{(l)} K_\theta^{(l)} \left(x, x_i^{(l)}|P,N\right), \qquad (1)$$

where each kernel $K_\theta^{(l)} : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}$, $l \in \{1, ..., L\}$ is derived from a convolutional neural network trained for this specific level. The coefficients $\alpha_i^{(l)} \in \mathbb{R}$, at the i voxel with center $x_i^{(l)}$, are optimized via a ridge regression during the forward pass. In addition to the voxel hierarchy, a second contribution is a new gradient-based kernel formulation that handle noise via predicted normal constraints.

4.2. Point visibility without mesh reconstruction

Because of the computational cost of mesh reconstruction, other more efficient options were proposed. In [9], Katz *et al*. define an hidden point removal operator. Knowing the position of the point of view $\mathbf{v}$, the first step is to *invert* all the points $\mathbf{p} \in \mathbb{R}^3$ of the point cloud P. This *inversion*, named *linear* or *spherical inversion*, is defined as

$$\hat{\mathbf{p}} = (\gamma - \|\mathbf{p} - \mathbf{v}\|_2) \frac{\mathbf{p} - \mathbf{v}}{\|\mathbf{p} - \mathbf{v}\|_2} \qquad (2)$$

with $\gamma > \max_{\mathbf{p} \in P} (\|\mathbf{p}\|_2)$ a parameter of the function. It corresponds to computing the symmetric of $\mathbf{p}$ with respect to the surface of a sphere of radius $\frac{\gamma}{2}$ centered at $\mathbf{v}$. Other *inversions* are also possible. In their second paper [8], Katz *et al*. give the necessary and sufficient conditions for such a function. They define two other *inversions*, the *exponential inversion*

$$\hat{\mathbf{p}} = \|\mathbf{p} - \mathbf{v}\|_2^\gamma \frac{\mathbf{p} - \mathbf{v}}{\|\mathbf{p} - \mathbf{v}\|_2} \qquad (3)$$

with $\gamma < 0$, and the *natural exponential inversion*

$$\hat{\mathbf{p}} = e^{-\gamma \|\mathbf{p} - \mathbf{v}\|_2} \frac{\mathbf{p} - \mathbf{v}}{\|\mathbf{p} - \mathbf{v}\|_2} \qquad (4)$$

with $\gamma > 0$.

We define as $\hat{P}$ the set of *inverted* point $\hat{\mathbf{p}}$. The second step is to compute the convex hull of the inverted set of points $\hat{P}$. A point $\mathbf{p}$ is said to be visible when its *inverse* $\hat{\mathbf{p}}$ lies on the convex hull derived during the second step. In the following, we refer to this method as DVPS. This approach is then extended in [13] to work on noisy point clouds. Note that the hull provides a local viewpoint-dependant mesh and is not used to render the point cloud.

Biasutti *et al*. [2] focus on the problem of point density when estimating the visibility. They propose a new method, that we refer to as VEVD, based on local neighboring points to avoid the problem of parameter selection in [9] based on the density of the studied point cloud. For that, the point cloud is projected onto the view camera and the k nearest

Density	Sparse					Dense				
Method	DVPS	VEVD	VEVD-I	Vis2Mesh	NKSR	DVPS	VEVD	VEVD-I	Vis2Mesh	NKSR
t(s)	10^1	10^2	10^2	10^1	10^1	10^1	10^2	10^2	10^2	10^1
TP	27.27	**31.59**	22.51	27.74	28.75	22.10	**31.90**	24.23	26.49	28.44
FP	**1.87**	18.86	9.38	7.02	3.53	**0.53**	18.49	9.49	4.90	3.14
FN	7.72	**3.40**	12.48	7.25	6.24	12.90	**3.10**	10.77	8.51	6.55
TN	**63.14**	46.16	55.63	57.99	61.49	**64.47**	46.52	55.51	60.11	61.87
Precision	**93.58**	62.62	70.59	79.81	89.07	**97.64**	63.31	71.86	84.40	90.07
Recall	77.95	**90.28**	64.33	79.29	82.16	63.14	**91.16**	69.24	75.69	81.27
Accuracy	**90.41**	77.74	78.14	85.74	90.23	86.57	78.42	79.75	86.60	**90.31**
F1-score	85.05	73.95	67.32	79.55	**85.48**	76.69	74.73	70.53	79.81	**85.45**
TP-c	17.39	**24.13**	13.44	23.64	18.22	12.94	**27.58**	15.91	22.81	18.77
FP-c	**1.92**	15.54	4.81	15.33	4.34	**0.48**	14.82	4.60	9.66	3.97
FN-c	14.98	**8.24**	18.93	8.73	14.15	22.95	**8.31**	19.98	13.08	17.12
TN-c	**65.71**	52.09	62.82	52.30	63.29	**63.63**	49.29	59.51	54.45	60.14
Precision-c	**90.05**	60.82	73.63	60.65	80.77	**96.40**	65.05	77.58	70.25	82.54
Recall-c	53.73	**74.55**	41.52	73.02	56.27	36.06	**76.84**	44.32	63.55	52.31
Accuracy-c	**83.10**	76.22	76.26	75.93	81.51	76.57	76.87	75.42	77.26	**78.91**
F1-score-c	**67.30**	66.99	53.10	66.27	66.33	52.48	**70.46**	56.41	66.73	64.03

Table 3. Quantitative results. Information on the clouds are given in Table 2. Positive predictions are the visible points in the outputs. The "-c" metrics are the complex visibility estimation metrics defined in Section 5. Vis2Mesh and NKSR were run on a GPU, the others on CPU. VEVD-I is described in Section 4.2. VEVD [2], DVPS [9], Vis2Mesh [19] and NKSR [6] are state-of-the-art methods.

neighbors are estimated for each point in the 2D space. The visibility of a given point $\mathbf{p}$ is given by

$$\alpha(\mathbf{p}) = \exp\left(-\frac{(d(\mathbf{p}) - d_{\min}(\mathbf{p}))^2}{(d_{\max}(\mathbf{p}) - d_{\min}(\mathbf{p}))^2}\right), \qquad (5)$$

where the depth $d(\mathbf{p})$ is the distance from the camera center to the point $\mathbf{p}$, namely $d(\mathbf{p}) = \|\mathbf{p} - \mathbf{v}\|_2$, $d_{\min}(\mathbf{p})$ is the distance to the closest point from $\mathcal{N}(\mathbf{p})$ the set of nearest neighbors of $\mathbf{p}$, namely $d_{\min}(\mathbf{p}) = \min_{\mathbf{p}' \in \mathcal{N}(\mathbf{p})} d(\mathbf{p}')$, and $d_{\max}(\mathbf{p})$ the distance to the farthest element, *i.e.* $d_{\max}(\mathbf{p}) = \max_{\mathbf{p}' \in \mathcal{N}(\mathbf{p})} d(\mathbf{p}')$. The αs are then thresholded to obtain a binary visibility classification.

VEVD is designed for urban scenes acquired with a LiDAR located on a car. In an indoor scene where many objects can be behind each others, the definition of the visibility α, in Eq. (5), can cause points to be classified as visible when they are not. Indeed, a hidden object can have a relatively small depth compared to the maximum depth of the scene. Formally, this corresponds to the case where a hidden point $\mathbf{p}$ is such that $d(\mathbf{p})$ is much smaller than $d_{\max}(\mathbf{p})$ thus leading to $\alpha(\mathbf{p})$ being close to 1. To mitigate this problem, we introduce a variant, named VEVD-I, that discards a neighbor when its depth is too different from $d(\mathbf{p})$ in the 3D space. A neighbor $\mathbf{p}'$ is kept if and only if

$$d(\mathbf{p}') - d(\mathbf{p}) < t(\mathbf{p}), \qquad (6)$$

where $\mathbf{t}(\mathbf{p}) > 0$ is a point-dependent threshold value function. We use $t(\mathbf{p}) = median_{\mathbf{p}' \in \mathcal{N}(\mathbf{p})} |d(\mathbf{p}') - d(\mathbf{p})|$ in the following. Compared to a global constant threshold, a local threshold function should help to generalize across datasets.

5. Experiments

We evaluate the state-of-the-art visibility estimation methods presented in Section 4 on nine viewpoints arbitrarily chosen inside the IRIS-VIS point cloud (Figure 4 and Table 2). The ground truth is computed using our given code as described in Section 3.2. In order to evaluate robustness towards density, experiments are performed on a sparse and a dense cloud generated via random uniform sampling by a factor two and ten percent respectively from the raw point cloud. We also evaluate the impact of the LiDAR acquisition noise in the supplementary material by performing the same evaluation as in this section but with a synthetic point cloud sampled from the CAD model.

Visibility estimation methods classify points in two categories, visible or hidden. This is why we consider classification metrics to measure the performance. In that case, positive and negative predictions correspond respectively to visible and hidden points from the considered viewpoint. On top of the true positives (TP), false positives (FP), false negatives (FN) and true negatives (TN), we also consider the precision $\left(\frac{TP}{TP+FP}\right)$, the recall $\left(\frac{TP}{TP+FN}\right)$, the accuracy $\left(\frac{TP+TN}{TP+FP+TN+FN}\right)$ and the f1-score $\left(\frac{2TP}{2TP+FP+FN}\right)$.

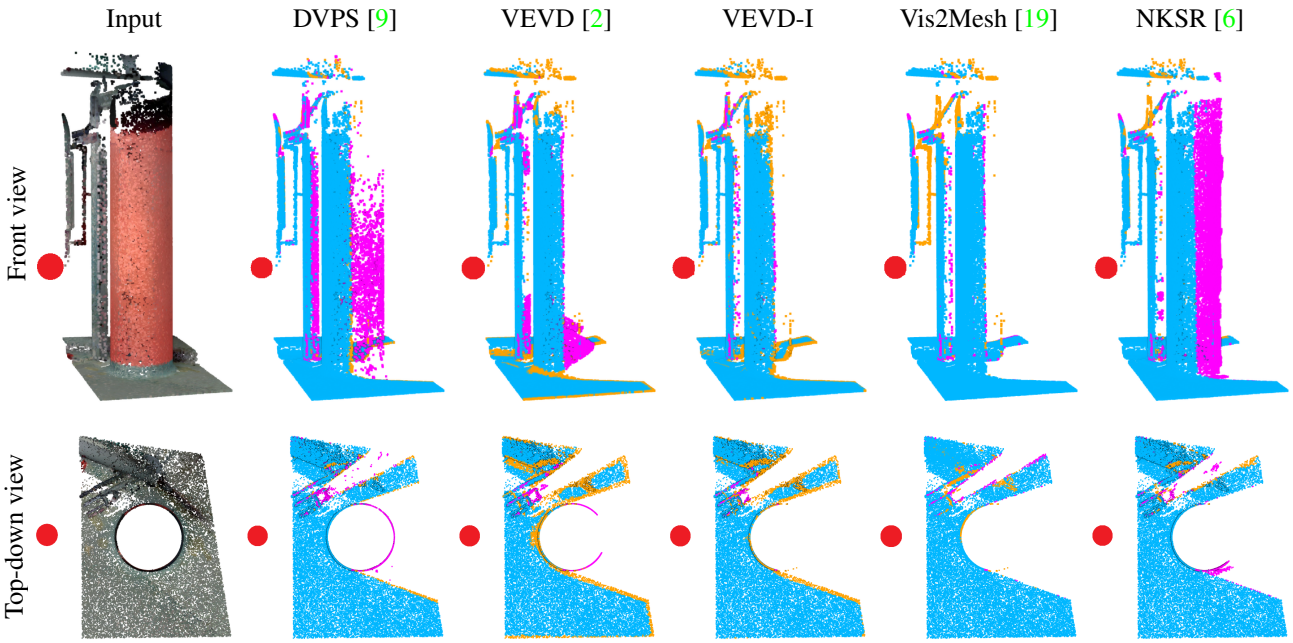


Figure 8. Qualitative results on Scene Quali-1 in the sparse point cloud. TP (blue), FP (purple), FN (orange). Positive predictions are the visible points in the outputs.

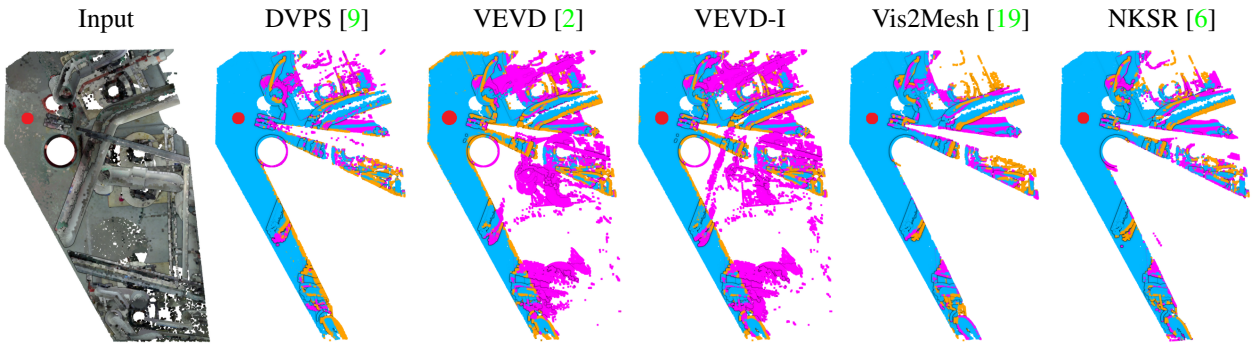


Figure 9. Qualitative results on Scene Quali-2 in the sparse point cloud. TP (blue), FP (purple), FN (orange). Positive predictions are the visible points in the outputs.

We also believe that, in most of the 3D tasks involving visibility estimation, false positives are more disruptive than false negatives and this is why we make the emphasis on the false positives related metrics during the analysis.

In practice, we observed that the visibility is harder to predict where the variation in depth of the visible points is high. Thus, we define new metrics, referred to as "complex visibility estimation metrics", to focus on these challenging areas that are often located at the border of the visible objects. To automatically compute these interesting locations, we first project the visible cloud on a sphere centered at the viewpoint and select the points whose variation in depth in the local neighborhood (50 nearest neighbors) is higher than a specific value. We chose the percentile 90 of the depth variances in the neighborhoods as threshold. Secondly, we project the non-visible cloud on the same sphere and select the points belonging to the new neighborhood of a previously selected visible point. Figure 10 shows that the selected points are mostly located at the visibility boundaries and Table 3 demonstrates that methods struggle to find the visible points in these areas as the complex recall provides significantly worse results than the standard recall.

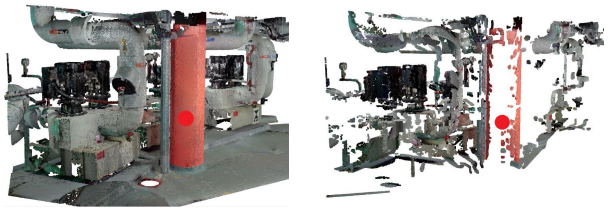


Figure 10. Point selection for the complex visibility estimation metrics. Left: the input point cloud. Right: the selected points.

We discuss implementation details and parameters choice in the supplementary material.

5.1. Quantitative evaluation

In Table 3, we present the quantitative experiments performed on both dense and sparse version of the cloud.

Looking at the accuracy and f1-score, NKSR [6] gives the best results for both densities. Regarding the precision, all methods perform significantly different. DVPS is the best by far. The poor ranking of VEVD confirms that it is not suitable for indoor environments. VEVD-I shows significant improvements for both densities but still trails the other methods. VEVD provides the best recall, giving less FN than the other methods.

NKSR and VEVD seem to be robust to low point density. Indeed, their performances are similar across all metrics for both sparse and dense point clouds.

Points located in complex areas are often predicted as non-visible, leading to a significant increase in FN-c and a decrease of the recall-c. As a result, the accuracy-c and f1-score-c also decrease for all the methods and densities. Similar to the standard metrics, for both densities DVPS provides the best precision-c and VEVD is the best in terms of recall-c.

In the supplementary, the performances on the synthetic point cloud are different than on the LiDAR derived cloud and often better in accuracy and f1-score. This experiment demonstrates the impact of the noise for this task and the advantage of providing a real point cloud in our dataset.

5.2. Qualitative evaluation

Qualitative results are shown on two scenes, named Quali-1 and Quali-2, included in the clouds used for the quantitative experiments. Figures 8 and 9 present the results in the sparse point cloud, whereas Figure 11 compares the performances of DVPS [9] in sparse and dense configuration, the only method that shows significantly different visualizations. More results are given in the supplementary, including the quantitative evaluation on these scenes.

Scene Quali-1 is simple: a pole with a convex and regular shape with no object behind (Figure 8). In these conditions, VEVD-I and Vis2Mesh, both sparse and dense, and DVPS [9] dense give the fewer amount of FPs. This is particularly visible behind the pole. We see also that VEVD gives worse results than the proposed variant VEVD-I. The methods not based on surface reconstruction (DVPS, VEVD and VEVD-I) present FNs at the visibility boundaries. The smaller amount of FNs given by Vis2Mesh and NKSR is the result of a precise mesh estimation and a raycasting step similar to the ground truth generation. We show the impact of point density for DVPS in Figure 11.

Scene Quali-2 includes scene Quali-1 and add objects behind the pole (Figure 9). DVPS, Vis2Mesh and NKSR

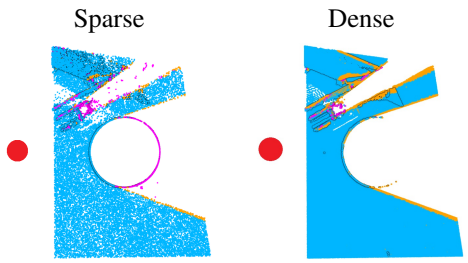


Figure 11. Difference between the sparse and the dense DVPS [9] predictions on scene Quali-1. TP (blue), FP (purple), FN (orange). Positive predictions are the visible points in the outputs.

are clearly robust to this scene modification as these methods compute a mesh hiding the objects behind. On the contrary, VEVD [2] struggles in removing the hidden points in scene Quali-2 contrary to scene Quali-1. The objects behind the pole increases $d_{\max}$, thus leading to a higher visibility in Eq. (5). VEVD-I shows fewer FPs but still more than DVPS, Vis2mesh and NKSR. DVPS dense gives fewer FPs than the other methods, removing both hidden points on the pole and the objects behind it. On the sparse cloud, only Vis2Mesh is able to remove all the points behind the pole.

DVPS and NKSR can estimate the visibility on small and thin objects such as pipes, valves or gauges. On the other hand, Vis2Mesh has difficulties with thin pipes as most of them are missing from the resulting mesh. VEVD and VEVD-I show FNs on some valves but perform well overall on most of the other thin objects.

6. Discussion and Perspectives

IRIS-VIS provides a dense real indoor point cloud coupled with a well-fitted CAD model of an industrial scene. Thanks to the presence of very detailed and small objects as well as large piping in many directions, it is suitable for challenging evaluations in many computer vision tasks. In particular, we designed new metrics for point visibility estimation and saw that VEVD-I outperforms VEVD [2] qualitatively. However, these two methods still trail significantly DVPS, Vis2Mesh and NKSR in computation time and task performance. DVPS seems to be a good compromise between performance (especially precision) and computation time but we also saw that it is not robust to low point density and very dependent on the choice of parameters. If the computational cost is not a limitation, NKSR [6] provides the best accuracy and f1-score while being robust to the density. Nevertheless, the performance of these methods is not yet sufficient for automatic industrial applications thus showing that more work is required to develop more appropriate methods for this case.

References

[1] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. ARKitscenes - a diverse real-world dataset for 3d indoor scene understanding using mobile RGB-d data. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. 3

[2] Pierre Biasutti, Aurélie Bugeau, Jean-François Aujol, and Mathieu Brédif. Visibility estimation in point clouds with variable density. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2019. 1, 2, 3, 5, 6, 7, 8

[3] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017. 3

[4] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 3

[5] Boris Delaunay. Sur la sphere vide. *Bulletin of the Academy of Sciences of the USSR Classe des Sciences Mathematiques et Naturelles*, 7:793–800, 1934. 3

[6] Jiahui Huang, Zan Gojcic, Matan Atzmon, Or Litany, Sanja Fidler, and Francis Williams. Neural kernel surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4369–4379, 2023. 3, 5, 6, 7, 8

[7] Jean-François Hullo, Guillaume Thibault, and Christian Boucheny. Advances in multi-sensor scanning and visualization of complex plants: The utmost case of a reactor building. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-5/W4:163–169, 2015. 3

[8] Sagi Katz and Ayellet Tal. On the Visibility of Point Clouds. In *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE. 2, 3, 5

[9] Sagi Katz, Ayellet Tal, and Ronen Basri. Direct visibility of point sets. In *ACM SIGGRAPH 2007 papers*, pages 24–es. 2007. 1, 2, 3, 5, 6, 7, 8

[10] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006. 3

[11] Patrick Labatut, Jean-Philippe Pons, and Renaud Keriven. Efficient multi-view reconstruction of large-scale scenes using interest points, delaunay triangulation and graph cuts. pages 1–8, 11 2007. 2

[12] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 3

[13] Ravish Mehra, Pushkar Tripathi, Alla Sheffer, and Niloy J Mitra. Visibility of noisy point cloud data. *Computers & Graphics*, 34(3):219–230, 2010. 1, 2, 3, 5

[14] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 3

[15] Francesco Pittaluga, Sanjeev J Koppal, Sing Bing Kang, and Sudipta N Sinha. Revealing scenes by inverting structure from motion reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 145–154, 2019. 3, 5

[16] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. 5

[17] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[18] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision, ECCV 2012 - 12th European Conference on Computer Vision, Proceedings*, number PART 5 in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pages 746–760, 2012. 12th European Conference on Computer Vision, ECCV 2012 ; Conference date: 07-10-2012 Through 13-10-2012. 2, 3

[19] Shuang Song, Zhaopeng Cui, and Rongjun Qin. Vis2mesh: Efficient mesh reconstruction from unstructured point clouds of large scenes with learned virtual view visibility. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6514–6524, 2021. 3, 5, 6, 7

[20] Peng-Shuai Wang, Chun-Yu Sun, Yang Liu, and Xin Tong. Adaptive o-cnn: a patch-based deep representation of 3d shapes. *ACM Transactions on Graphics*, 37(6):1–11, Dec. 2018. 5

[21] Francis Williams, Zan Gojcic, Sameh Khamis, Denis Zorin, Joan Bruna, Sanja Fidler, and Or Litany. Neural fields as learnable kernels for 3d reconstruction, 2021. 3, 5

[22] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 3