On the Role of Representations for Reasoning in Large-Scale Urban Scenes

Supplemental Material

Randi Cabezas[†,1] Maroš Bláha[†,2] Sue Zheng[1] Guy Rosman[1]
Konrad Schindler[2] John W. Fisher III[1]
[1] Massachusetts Institute of Technology [2] ETH Zurich

Contents

1. Representations

In this section we discuss representation specific details which were omitted from the paper for space constraints. We begin by providing a few additional properties of the taxonomy presented in §3.1 of the paper. We continue by discussing the methodology for learning representation, as presented in §3.2. We conclude the discussion by providing scene descriptions for the Enschede and the SynthCity scenes.

[†] shared first authorship

	Element	Metric Attribute	Scaling Property	Connectivity Element
0D	Point	Visual	Surface	-
1D	Contour	Length	Surface	Point
2D	Polygon	Area	Surface	Line
3D	Polyhedron	Volume	Volume	Polygon

Table 1: Summary of taxonomy properties.

1.1. Representation Taxonomy

We proposed a categorization of various 3D representations to better understand and quantify their effect on a reasoning task, see §3.1 of the paper. The proposed taxonomy contains two complementary characteristics: *metric* and *attribute*. The metric characteristic can be broken down into the inherent dimension of the representation element, *i.e.*, the highest metric property that can be measured from it. As a result we categorized representations as 0D, 1D, 2D, or 3D, with metric attribute: visual, length, area, or volume, see Tab. 1. As stated in the paper, points, contours, polygons, and polyhedra are the canonical examples for each dimension.

An additional component of the metric characteristic is connectivity. In principle any element can be connected, introducing added structure into the representations. In general the choice of arbitrary connectivity defines separate representations within the same group in the taxonomy (*e.g.*, triangle soup versus closed triangular mesh). Under *special connectivity*, the representation can in principle move to a different group in the taxonomy, *e.g.*, imagine all lines in a 1D representation connected in such a way that polygons can be constructed from them, in this case 1D-metric representation became a 2D-metric representation. In other words, under very specific connectivity representations of a lower-dimension can implicitly assume a higher-metric dimension. As noted in the paper, we observe that connectivity must occur using elements of one dimension less than the representation dimension (*e.g.*, lines connect polygons).

1.2. Learning Representations

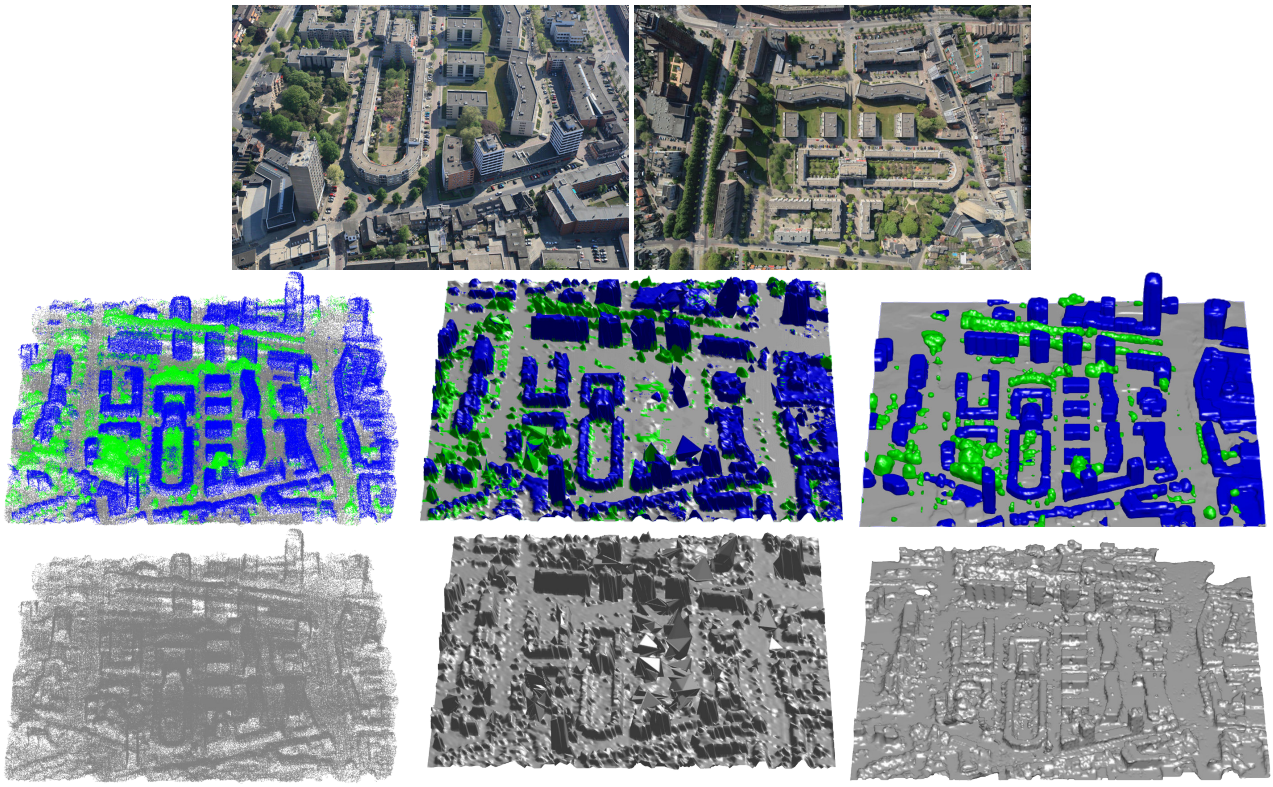
Section 3.2 of the paper provided a brief description of how the six representations used in this work are constructed. In this section we expand that description by discussing additional details.

Pre-Processing. As a first step of processing the input images images are aligned using VisualSFM [11] to produce a sparse 3D point cloud which provides the basis for depthmap creation and image classification. The depthmaps are generated using plane sweep stereo together with the semi-global matching implementation from OpenCV as matching cost [2]. The semantic information is derived from the images using a multi-class version of Adaboost [1] trained on manually labeled ground truth images. The features utilized in the classifier are extracted directly from the images (RGB values) and the depthmaps, where the latter corresponds to local geometry features based on [5]. The classifier provides per-class probabilities for each pixel in each image. As stated in the paper, we restrict ourselves to three mutually exclusive classes: *buildings*, *ground*, *vegetation* (with voxel representations explicitly modeling the additional class *free-space*).

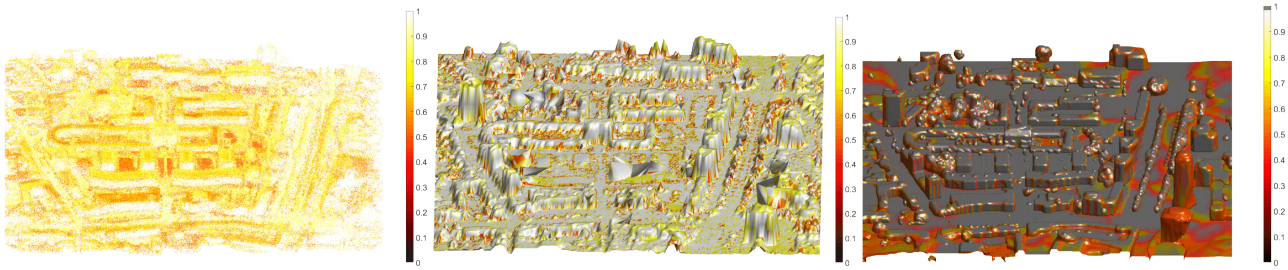
Point Cloud. The non-semantic point cloud representation is computed solely from the depthmaps. Outlier removal is accomplished via filtering [9]. The semantic point cloud is obtained from the non-semantic one, and attributed semantic information by assigning the per-pixel probabilities of the classified input images to the particular points.

Mesh. The non-semantic mesh representation used in this work is based on [3]. This probabilistic method uses a set of noisy images, camera positions, and LIDAR point cloud to reconstruct the geometry and appearance of the 3D scene. The mesh model utilizes the images directly as input, we do not use any LIDAR measurements in this model. The semantic mesh representation used is based on [4], which expands on [3] by modeling 3D semantic labels for each mesh element. Semantic attribution is accomplished in this model by incorporating the semantic point cloud as observations.

Voxels. To obtain the non-semantic voxel model, we apply a state-of-the-art volumetric fusion method [12] to the depthmaps. This leads to a binary space partitioning of equally sized voxels, *i.e.*, each voxel is assigned either to free-space or to a solid object. Our semantic voxel representation is generated using [7]. The depths and semantic class likelihoods are used to compute the data term, a signed distance function, in a 3D grid with equally sized voxels. Next, inference is done via convex optimization, leading to either a semantically-attributed voxel model.



(a) *Top:* two input images. *Bottom:* semantic and non-semantic models. *L-R*: PC, mesh, and voxels.



(b) Semantic uncertainty. Color shows the most likely class probability; higher values imply more certainty. *L-R*: PC, mesh, voxels.

Figure 1: Enschede input images, scene models and semantic uncertainty.

1.3. Scene Descriptions

The results presented in the paper and this document rely on the reconstruction of two scenes: a real data reconstruction of Enschede Netherlands [10], and a synthetic example *SynthCity3* [4]. In this section we show some details about the scene that were omitted from the paper due to space constraints.

1.3.1 Enschede

For the reconstruction of the Enschede scene, 39 images were utilized. The images were in a *Maltese cross* configuration, with four oblique aerial images and one nadir aerial image per imaging point; two of the input images (an oblique image and a nadir image) are shown in Fig. 1(a). The six models used in this work were constructed solely from these images as described earlier, Fig. 1(a). The size of the representations are: 1M points (point cloud), 242K triangles (mesh), 300x300x64 grid (voxel). These values result in similar element sizes across the representations. Fig. 1(b) shows a complementary view of the semantic uncertainty of the models.

1.3.2 SynthCity3

For the reconstruction of the SynthCity3 scene we used 12 images. The images were in a circular configuration around a synthetic downtown scene Fig. 2(a). The six models used in this work were constructed solely from these image, Fig. 2(a). The size of the representations are: 1M points (point cloud), 220K triangles (mesh), 256x256x100 grid (voxel). Fig. 2(b) shows a view of the semantic uncertainty of the models. The value of the synthetic city is that it contains ground-truth semantic and geometric information. We relied on both of these aspects in the paper in order to demonstrate the accuracy of the proposed methodology.

2. Query Uncertainty

Here we derive the query computation and uncertainty results for special cases of queries. For the subsequent derivations, we recall the following notation and relationships. Representations are treated as a collection of random variables denoted by $\mathbf{X} = \{X_i\}_{i=1}^n$, where n is the number of representation elements and X_i is the i-th element. A query is a deterministic function $f(\cdot)$ with parameters θ that operates on the representation $\mathbf{X}$; the query answer is given by $Q = f(\mathbf{X}; \theta)$.

2.1. Derivation of Special Case: Query Dependent on a Subset

Here we show that for a query that operates only on a subset of representation elements (*i.e.*, a "local" query), the entropy of the query answer will simplify to depend only on the subset. Let $\mathbf{X}_s$ denote the subset of elements that the query operates on and $\mathbf{X}_{\setminus s}$ denote the remaining elements; *i.e.*, $Q = f(\mathbf{X}_s; \theta)$ and $\mathbf{X}_{\setminus s} = \{X_i | X_i \notin \mathbf{X}_s\}$. The dependence of the query answer on only the subset results in the conditional independence of Q and $\mathbf{X}_{\setminus s}$ when conditioned on $\mathbf{X}_s$; this conditional independence is the key property we exploit in our derivations.

The joint entropy of the representation and the query answer can be decomposed in the following way:

$$\begin{aligned} H(Q, \mathbf{X}) &= H(Q) + H(\mathbf{X}_s|Q) + H(\mathbf{X}_{\setminus s}|\mathbf{X}_s, Q) \\ &= H(Q) + H(\mathbf{X}_s|Q) + H(\mathbf{X}_{\setminus s}|\mathbf{X}_s) \ . \end{aligned} \tag{1}$$

We use the chain rule for entropy for the first equality and the conditional independence property for the second. We can use the chain rule for entropy to decompose the joint entropy in an alternative fashion then use conditional independence again to obtain a different expression:

$$\begin{aligned} H(Q, \mathbf{X}) &= H(\mathbf{X}_s) + H(\mathbf{X}_{\setminus s}|\mathbf{X}_s) + H(Q|\mathbf{X}_{\setminus s}\mathbf{X}_s) \\ &= H(\mathbf{X}_s) + H(\mathbf{X}_{\setminus s}|\mathbf{X}_s) + H(Q|\mathbf{X}_s) \ . \end{aligned} \tag{2}$$

After equating the RHS of 1 and 2 and performing some algebraic manipulations we arrive at:

$$H(Q) = H(\mathbf{X}_s) + H(Q|\mathbf{X}_s) - H(\mathbf{X}_s|Q) \ . \tag{3}$$

It is evident from 3 that the $H(Q)$ is only a function of the subset $\mathbf{X}_s$.

2.2. Derivation of Special Case: Independent Representation Elements

Here we show how the distribution of the query answer $p_Q(Q)$ simplifies to a convolution when the elements X_i's are independent and the query is linear (*i.e.*, takes the form $Q = \sum_i^n \alpha_i X_i$).

We begin by defining the set of random variables $Y_i = \alpha_i X_i$. Note that the Y_i's are independent and the query answer can be written as $Q = \sum_i^n Y_i$. It is well known that if random variables are independent, the distribution of their sum is equal to the convolution of their distributions [8]. Thus,

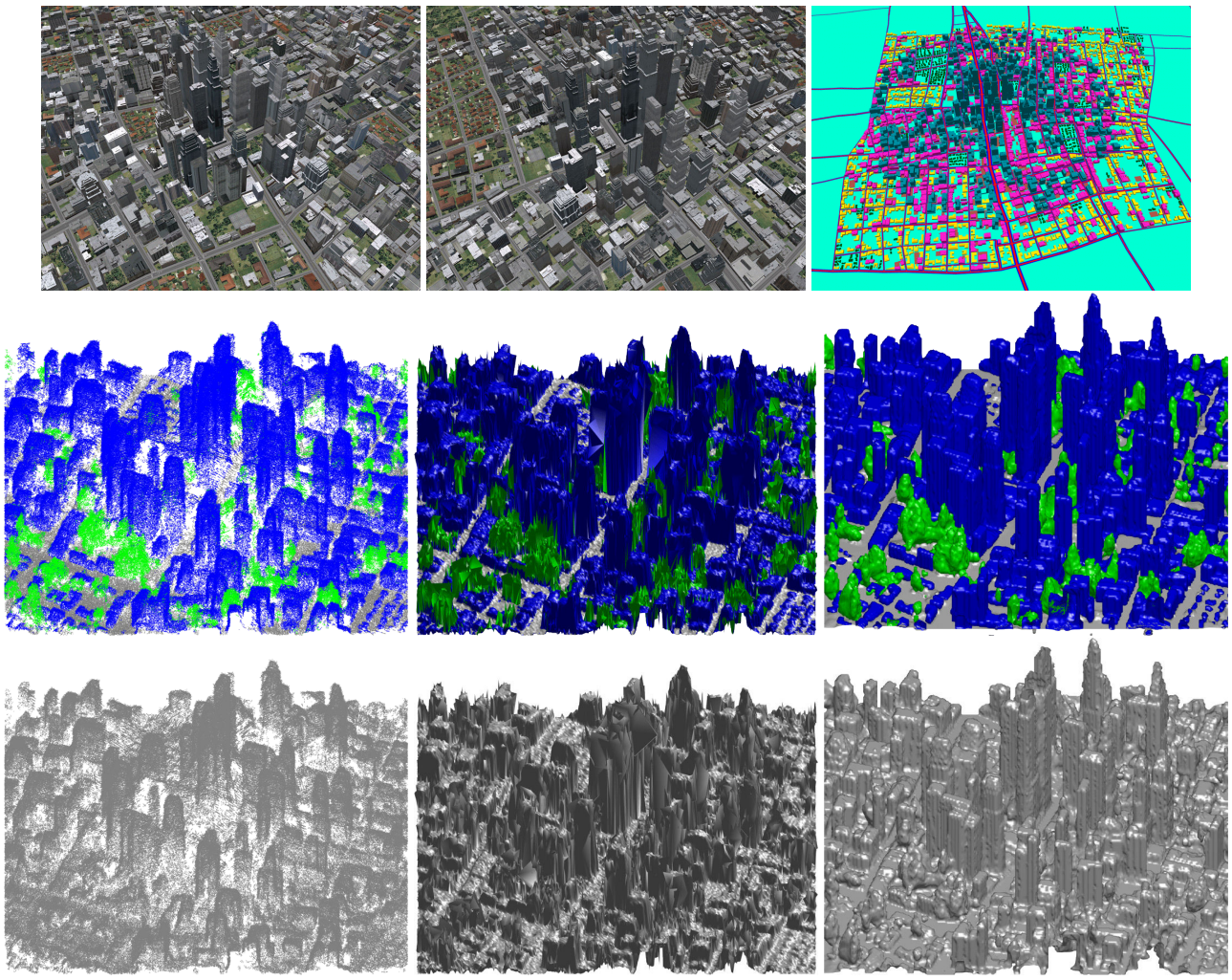
$$p_Q(Q) = (p_{Y_1} * p_{Y_2} * \cdots * p_{Y_n})(Q) \tag{4}$$

where the distribution of Y_i, for continuous X_i, is given by

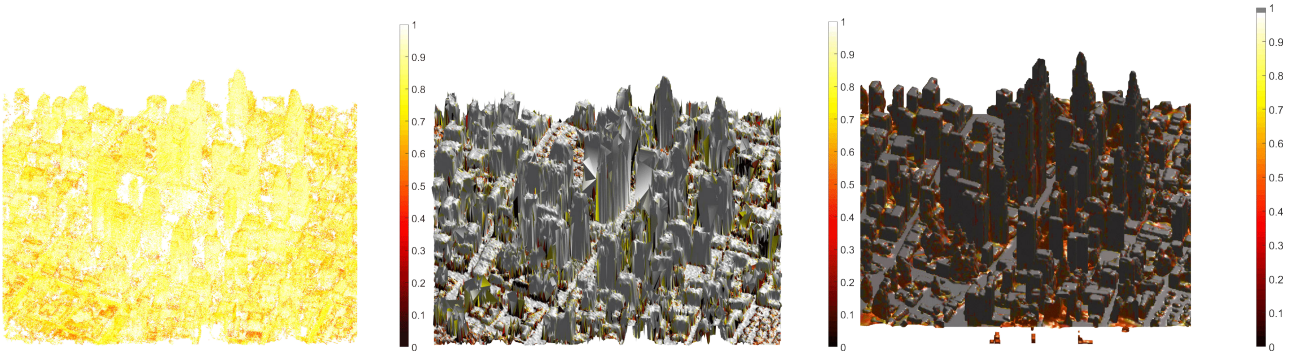
$$p_{Y_i}(Y_i) = \frac{1}{\alpha_i} p_{X_i}(\frac{Y_i}{\alpha_i}) \tag{5}$$

and, for discrete X_i, is given by

$$p_{Y_i}(Y_i) = p_{X_i}(\frac{Y_i}{\alpha_i}) \ . \tag{6}$$



(a) *Top:* two input images and ground-truth categories. *Bottom:* semantic and non-semantic models. *L-R*: PC, mesh, and voxels.



(b) Semantic uncertainty. Color shows the most likely class probability; higher values imply more certainty. *L-R*: PC, mesh, voxels.

Figure 2: SynthCity3 input images, scene models and semantic uncertainty.

2.3. Linear Query with Independent Elements Example

In the special case of linear queries, we can compute the desired statistics in closed-form. Furthermore, if the representation elements are independent, this special case facilitates the exposition of uncertainty propagation from the representation to the query answer. We now walk through an example of a linear query with independent elements to demonstrate how to

compute the desired quantities in closed-form.

The example linear query we consider is the scene category analysis query with counts as the metric quantification. This query for class c can be written as

$$Q^c = \frac{\sum_i Z_i^c}{n} \triangleq \frac{b}{n} \tag{7}$$

where $Z_i = \mathbb{1}[C(X_i) = c]$, $C(X_i)$ is the class of element X_i and $b \triangleq \sum_i Z_i^c$. This is a linear query although it may not appear so; since the representation element comprises geometry, semantics, and appearance parameters, $C(\cdot)$ extracts the component pertaining to semantics while $Z_i = \mathbb{1}[C(X_i) = c]$ is a simple reparameterization of the semantics class into is/is-not class c. Because the representation elements are independent, b has a Poisson binomial distribution which is a generalization of the binomial distribution to the sum of n independent *non-identically* distributed Bernoulli random variables; consequently, the query answer Q^c follows a scaled Poisson binomial distribution with mean

$$\mu_Q = \frac{\sum_i p_i^c}{n} \tag{8}$$

and variance

$$\sigma_Q^2 = \frac{\sum_i p_i^c(1 - p_i^c)}{n^2} \tag{9}$$

where p_i^c is the probability of class c at element X_i. From Eq. 9, it is evident that the variance of the query answer directly depends on the semantic uncertainty of each element (encoded in p_i^c); semantic *certainty* would correspond to p_i^c taking a value close to 1 for a particular class and 0 for the other classes and would result in a low contribution to variance from this element: $p_i^c(1 - p_i^c) \approx 0$. This illustrates how semantic uncertainty propagates to uncertainty in the query answer.

Importantly, we note that the variance *does not* reflect the error in the query answer; only when the query answer is *unbiased* will the variance reflect the mean squared error. As an extreme example, consider the case where there is only one semantic observation for the entire scene. That observation may reflect the class composition in the vicinity of the observation but does not reflect the class composition of the entire scene. In this case, the query answer will be biased and does not reflect the true class composition of the scene. We observe such biases in our query answers in our scene category analysis experiments, where there are significant errors in the query answers but low variances.

2.4. Empirical Entropy Bound Check

In this section we show an empirical result regarding the tightness of Eq. 2 from the paper. We note that the inequality is in general very loose when the function $f(\cdot)$ is not one-to-one. To highlight this, we create a synthetic example by simulating a collection of random variables. For simplicity we assume the random variables are binary. Then assume our function is $Q = \sum_i^N X_i$. As we showed, we can compute the entropy of Q in closed-form, and what we are after is how different is the value provided by Eq. 2 from the closed-form solution.

We being by obtaining a set of random variables $\mathbf{X} = \{X_i\}_{i=1}^N$, where $N = 200$ and X_i is a Bernoulli random variables. That is $X_i \sim \text{Beta}(\alpha, \beta)$, where α and β are the parameters of the Beta distribution that control the mean and variance. The top row of Fig. 3 shows three different values of these parameters; these values were chosen to produce a mean of 0.5, and a variance that shapes the spread, from concentrated around the mean, to evenly distributed in the range $[0, 1]$, to concentrated in the tails. For the reconstruction models employed in this work we expect that the semantic attributes follow more closely the latter case, in other words, most of the elements have semantic attributes with clear values, either on or off.

The bottom row of Fig. 3 shows the exact probability density of Q as computed by following the approach of [6]. From the figure we can see that the distribution closely resembles a Gaussian distribution for all three cases considered (see red outline of the estimated Gaussian distribution as expected from central limit theorem (CLT)). The figure also shows the nominal values of entropy obtained from closed-form calculation as well as several methods for approximating the entropy. We can see that the estimated bound is very far for the exact computation for all three configurations. In addition, the figure shows that the entropy computation from samples (labeled "H_b") produces a much closer approximation to the true value than the bound. It should be noted that this value is an underestimate of the true value (*i.e.*, it is an overconfident estimate). Importantly we note that the estimate from CLT is very close to the true value; this is to be expected as N grows.

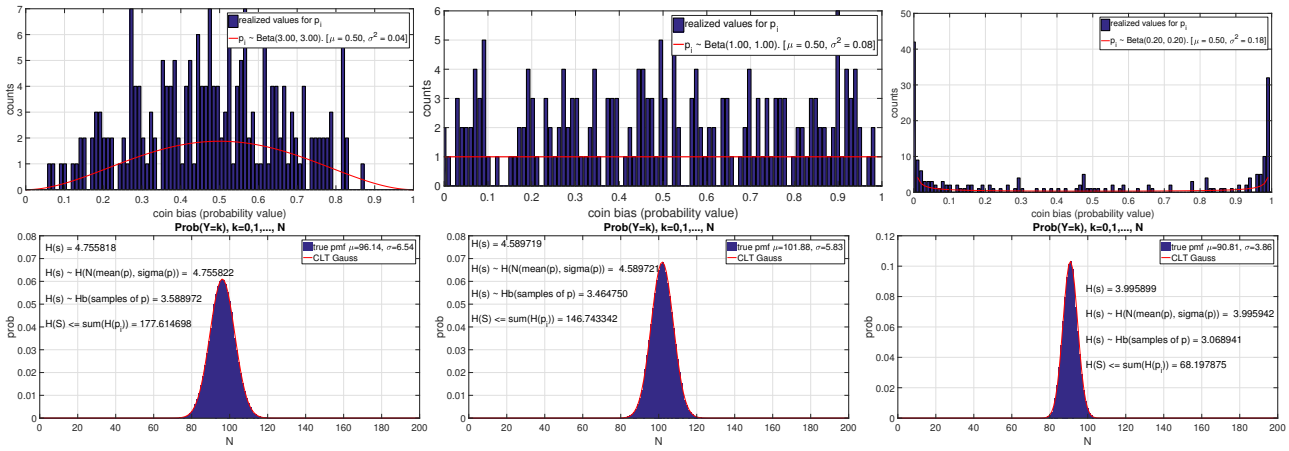


Figure 3: Empirical entropy bound check. *Top:* three random draws of a Beta distribution with mean 0.5 and varying variance, from concentrated around the mean, evenly spread and concentrated in the tails. *Bottom:* Two hundred samples from each distribution where used to estimate entropy and compare it to closed-form calculation.

Model	Number Elements	Non-Semantic		Semantic	
		Total (ms)	Per Element (μs)	Total (s)	Per Element (μs)
PC	1,011,695	117.9	0.1165	4.874	4.818
Mesh	122,525	7.5	0.0612	0.601	4.907
Voxel (semantic)	5,760,000	-	-	27.5699	4.786
Voxel (non-semantic)	5,760,000	297.3	0.0052	-	-

Table 2: Sampling time for the specified representations. The sampling time for non-semantic models includes geometry only; for semantic models, it includes the time required for sampling both geometry and semantics. The voxel model is an exception; there, we only sample once: either geometry (occupied or free), or semantics (free, building, vegetation, or ground). All samplers use *MATLAB's* built-in implementations (version 2015b).

3. Computation Time

We ran our experiments on a *Windows 7* 64-bit PC with an *Intel Xeon* 12 core CPU at 2.4 GHz, and 48 GB of RAM. For the sampler software, we used *MATLAB's* built-in implementations (version 2015b).

The following analysis complements the theoretical analysis of complexity cost in §3.2 and §4.1 of the paper. First, we present the runtime of drawing one sample for each of the different representations in Tab. 2. Since these representations differ in number of elements, we focus on the per-element time rather than the total time. The timing numbers in Tab. 2 show that the semantic representations have similar computational costs and this computation is dominated by semantics sampling. Second, we present the runtimes of performing one query computation on one realization of the world in Tab. 3. Here, we observe high runtimes for the point cloud and mesh in the aerial case of the Path Planning query; this arises from the costly RRT (Rapidly-Exploring Random Trees) method applied in these cases.

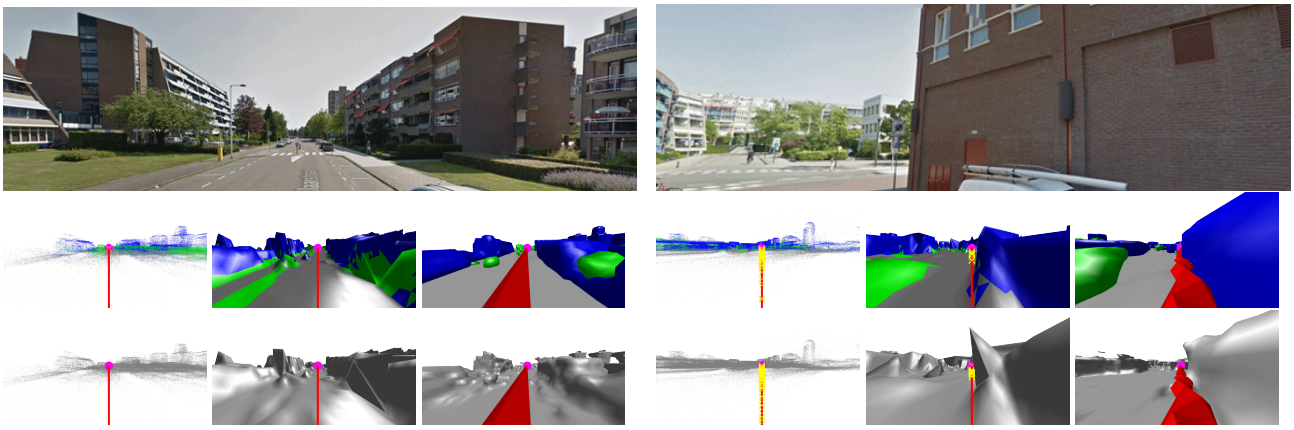
4. Clear Line of Sight

In the following, we present complementary investigations of the clear line of sight (CLOS) query, which were omitted in the paper due to lack of space. In all experiments, 100 samples were drawn for each representation unless stated differently. The first experiment shows three canonical examples and demonstrates the variation of the query's outcome, depending on the underlying representation. The remaining experiments are based on CLOS corner example provided in § 5.1 of the paper. Recall that in these experiments, the sightline is close to a building corner and the target moves from clearly visible to occluded.

The query is tested by perturbing the geometry of the scene according to each representations' uncertainty. In these experiments, the uncertainty was fixed but in reality, the noise level can vary and depends on various factors, *e.g.* the amount and quality of the input data. To address this fact, we vary the noise level artificially in the point cloud and mesh representa-

	Models	Clear Line of Sight		Category Analysis						Path Planning			
				Count		Area		Volume		Ground		Aerial	
		T. (s)	E. (μs)	T. (s)	E. (μs)	T. (s)	E. (μs)	T. (s)	E. (μs)	T. (s)	E. (μs)	T. (s)	E. (μs)
No-Sem.	PC	0.182	0.180	0.073	0.073	2.444	2.448	10.659	10.676	11.19	11.06	222.7	220.2
	Mesh	0.052	0.423	0.097	0.443	0.093	0.422	0.099	0.453	0.316	2.579	69.68	568.7
	Voxel	0.233	0.042	0.157	0.715	0.627	2.857	0.157	0.714	8.319	1.444	31.21	5.419
Sem.	PC	0.328	0.321	0.549	0.552	1.836	1.844	0.697	0.700	2.769	2.736	238.1	235.4
	Mesh	0.077	0.635	0.041	0.187	0.040	0.182	0.040	0.181	0.291	2.402	90.62	749.3
	Voxel	0.726	0.137	0.058	0.265	0.437	1.993	0.060	0.273	7.555	1.311	36.85	6.398

Table 3: Total runtime for each query for a single sample realization (abbreviated "T."). Per element column divides the total time by the number of elements in the representation (abbreviated "E."). Reported times include sampling time. We use *MATLAB's* built-in sampler implementations (version 2015b) with the exception of our own implementation of a multinomial sampler to sample semantics.



(a) CLOS example along a clear road

(b) CLOS example near a building boundary

Figure 4: Two Enschede examples of CLOS with 1 sample per representation. *Top:* view from ray start location; *Middle:* semantic representations; *Bottom:* non-semantic representations. *Left-to-Right:* point cloud, mesh, voxel. Line of sight in red, intersections in yellow, target in magenta.

tions. Since we cannot vary the noise level for voxels in an equivalent manner, we forgo the volumetric representation in this analysis. Since we expect little difference between the semantic and non-semantic cases, we focus on the semantic case as a representative subset. Note that while the absence or presence of semantics attributes can influence geometric aspects of the representations, we observe little impact on the resulting query computation.

4.1. Simple CLOS examples

The CLOS analysis starts with a simple example by drawing one sample per representation. We use a crosswalk in a residential area - a desirable place for clear visibility - as the testbed. As Fig. 4(a) shows, all representations find a clear view between the specified viewer and target. This scenario could incline one to naively trust these query answers; however, urban environments often feature more complex situations, *e.g.* two merging roads in a street canyon. A clear view in this case is crucial for traffic, pedestrians, *etc*. A CLOS passing by the corner of a building may become obstructed under a small perturbation to the geometry of the scene representation Fig. 4(b); this leads us to examine the confidence in the query answers.

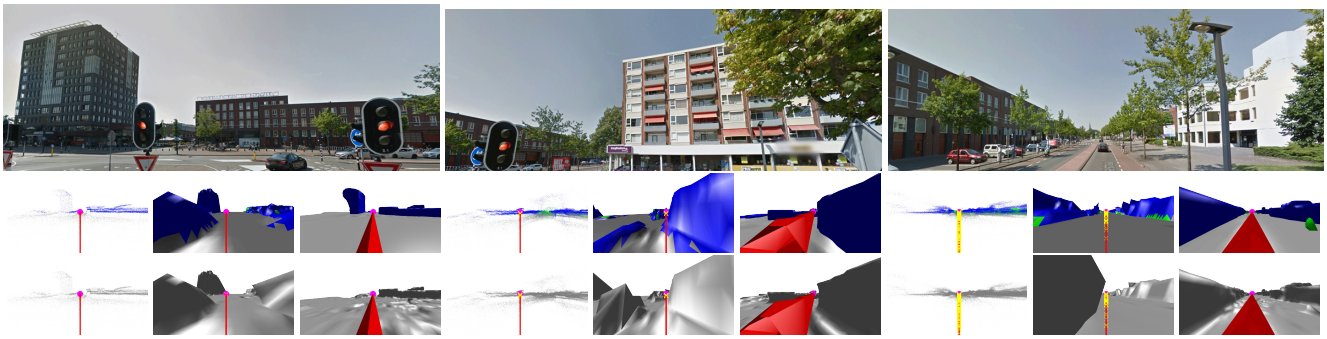


Figure 5: *Left-to-Right*: three canonical examples for an *easy*, *medium* and a *hard* CLOS based on 100 samples per representation. *Top*: view from ray start location. *Middle*: semantic point cloud, mesh and voxel model (*left-to-right*). *Bottom*: non-semantic models in the same order. Line of sight in red, intersections in yellow, target in magenta.

	Models	Easy		Medium		Hard	
		Answer	Confidence	Answer	Confidence	Answer	Confidence
Sem.	Point Cloud	yes	1.00	yes	0.87	no	1.00
	Mesh	yes	1.00	no	0.51	no	0.51
	Voxel	yes	1.00	yes	1.00	yes	1.00
No-Sem.	Point Cloud	yes	1.00	yes	0.82	no	1.00
	Mesh	yes	0.99	yes	0.58	yes	0.58
	Voxel	yes	1.00	yes	1.00	yes	1.00

Table 4: Outcomes and confidences of three canonical CLOS examples (Fig. 5) based on 100 samples per representation.

4.2. Three Canonical Examples

We analyze the visibility in three different scenarios of our urban scene and categorize them according to the concordance between the outcomes of the different representations:

- *Easy*: all representations agree in their answers with high confidence. The test object for this case is a short line in an open space area.
- *Medium*: all representations but one outlier (here, the non-semantic mesh) have the same outcome, however, the confidence level varies. As test object, we use a sightline passing close to a building corner.
- *Hard*: the visibility and the confidences are different in all world states. A long line in a street canyon and alley respectively is used as testbed.

The above categories underline the fact that, in complex environments such as our urban habitat, the outcome of this reasoning task depends on the underlying representation of the world. The qualitative and quantitative results are presented in Fig. 5 and Tab. 4.

4.3. Variable Noise Level

First, the (geometric) noise level is varied for the case when a static line of sight passes near a building corner (Fig. 4(b)). That is Fig. 6 depicts the point cloud and mesh representation with various artificial noise levels. From the figure we can see that as the noise decreases the uncertainty in the query answer decreases as expected. Furthermore, as the noise level decreases we observe that the transition point, when the object goes from visible to not visible, becomes substantially sharper for both representations. In this context, the point cloud is more noisy. At high noise level, both representations report very low chances in visibility. The entropy associated with the query answer also sharpens and becomes more concentrated around the estimated building boundary.

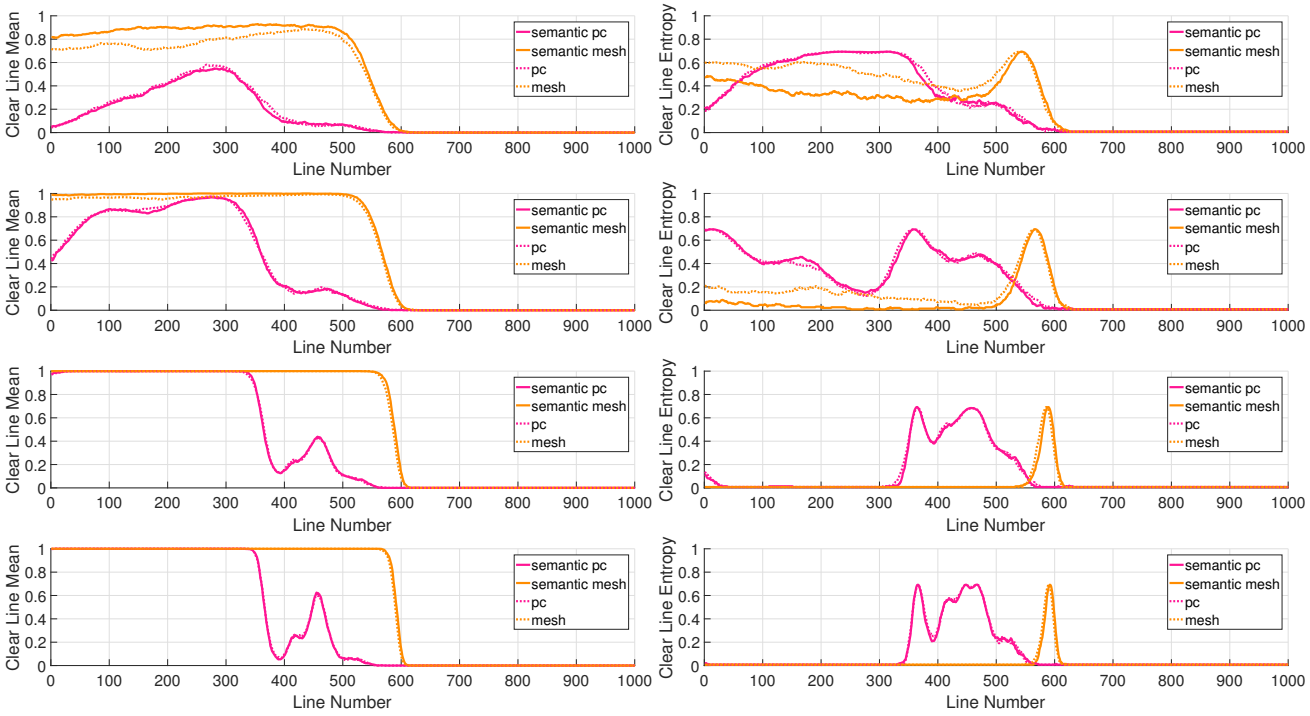


Figure 6: CLOS mean (*left*) and entropy (*right*) when varying the noise level and moving the target around a building corner, based on 1000 samples per representation. *Top-to-Bottom*: four different noise levels, gradually decreasing.

4.4. Variable Sphere-Radius Size

Similar to the noise level variation, the sphere-diameter in the point cloud represents another uncertainty factor. One option (used in the paper) is to choose the average ground sampling distance of a pixel from the input images. In the following, we artificially vary this parameter for the same situations as in § 4.3, and check the query for visibility. In terms of the static sightline, we observe a comparable behavior: the smaller the point size, the better the visibility (Fig. 7). When the target is moved around the corner, almost no visibility results for large sphere diameter. In contrast, small points show potential visibility for the whole target trajectory. In this case, we converge to the naive point cloud implementation (infinite small points) which would always declare a CLOS. We note that the entropy of the query under this noise scenario is highly unstable. Both entropy and mean query answer suggest that picking the right value for this parameter is important, too low and CLOS will always be visible, too high and there won't be much visibility.

5. Scene Category Analysis

Fig. 8 shows the different query formulation computation for the Enschede scene (similar to Fig. 9 of the paper, Fig. 10 of the paper is a selection of columns from Fig. 8). The figure shows that the formulation of the query has the ability to bias our view of the world. In other words, computing scene category analysis based on counts, area or volume will bias our results depending which representation is used, *e.g.*, building make up anywhere between 35% to 50% of the scene depending on formulation and representation. As shown in the synthetic experiments the accuracy of these values can vary significantly based on the assumptions needed in order to lift the representations for the computation.

6. Path Planning

In this section, we provide further experiments of the path planning query which were omitted in the paper due to the limited space. In particular, we show more paths, draw more samples per representation, and analyze additional characteristics such as *path length*, the *time per sample*.

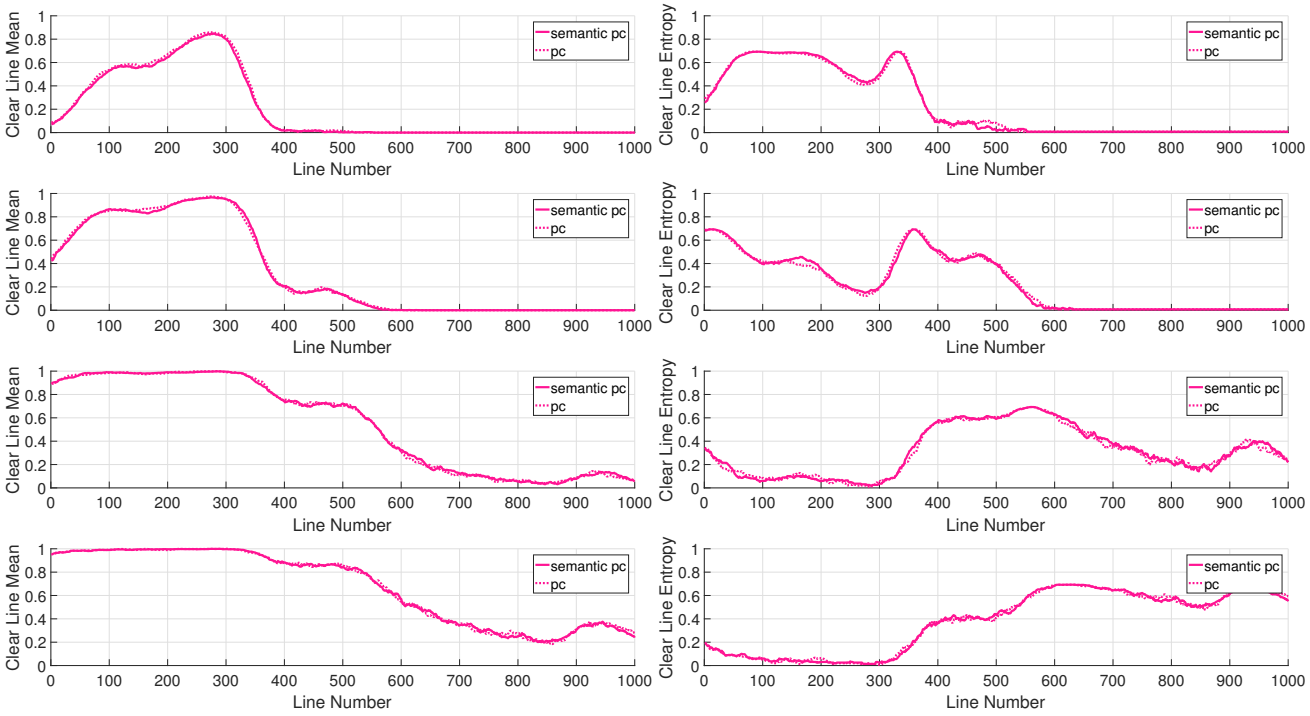


Figure 7: CLOS mean (*left*) and entropy (*right*) when varying the point-radius and moving the target around a building corner for PC representation, based on 1000 samples per representation. *Top-to-Bottom*: four different point sizes, gradually decreasing.

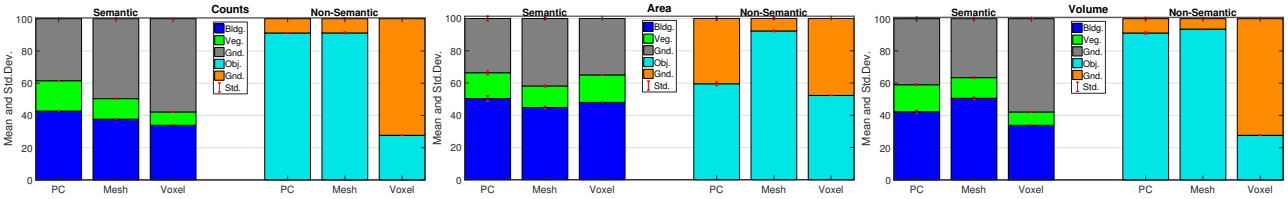


Figure 8: Enschede scene category analysis, for all query formulation, *e.g.*, counts, area and volume (100 samples per reprentation).

6.1. Ground Paths

Here, we present additional results for the path planning query. We consider five long pedestrian paths in the Enschede scene for all representations. For each path in each representation we draw 20,000 samples. The statistics of path existence for each representation is shown in Tab. 5. In addition, Fig. 9 and Fig. 10 presents the spatial distribution of found paths. Importantly, we note that pedestrian paths 3, 4 and 5 were presented in § 5.3 of the paper.

From both table and figure we observe the greatest consistency in paths from the semantic point cloud representation, in contrast to the largest variability from the semantic mesh representation. Interestingly, this characteristic flips in the non-semantic case, where the paths in the mesh are the most consistent. In both semantic and non-semantic cases, the representation with the largest path variability also has the lowest success rate in finding a valid path and lowest entropy. The voxel representations are highly consistent in both semantic and non-semantic case.

As a final example, Fig. 11 shows the *path length* and *time per sample* for the "Pedestrian 1" path. The path length example, Fig. 11(b), shows that paths tend to be shorter in the non-semantic models. This suggests that these paths move along non-traversable surfaces such as vegetation since semantic information is not available in these models; in the semantics models, such paths would be restricted. We find this is the case in all the paths we computed.

In terms of *time per sample*, Fig. 11(a), all representations are consistent in itself, showing similar distributions for the non-

		Pedestrian 1			Pedestrian 2			Pedestrian 3			Pedestrian 4			Pedestrian 5		
		M	S	E	M	S	E	M	S	E	M	S	E	M	S	E
Sem	PC	94.4	0.2	0.22	91.8	0.2	0.28	93.9	0.2	0.23	93.3	0.2	0.25	93.9	0.2	0.23
	Mesh	81.4	0.3	0.48	82.3	0.3	0.47	99.6	0.0	0.02	55.9	0.4	0.69	96.7	0.1	0.15
	Voxel	100	0.0	0.00	100	0.0	0.00	100	0.0	0.00	100	0.0	0.00	100	0.0	0.00
No Sem	PC	74.1	0.3	0.57	73.1	0.3	0.58	95.7	0.1	0.18	74.8	0.3	0.57	90.4	0.2	0.32
	Mesh	100	0.0	0.00	100	0.0	0.00	100	0.0	0.00	100	0.0	0.00	100	0.0	0.00
	Voxel	100	0.0	0.00	100	0.0	0.00	100	0.0	0.00	99.9	0.0	0.01	100	0.0	0.00

Table 5: Confidence of finding a path for five ground examples. Values are based on 20000 samples. Legend: M:Mean (%), S:Standard Deviation (%), E:Entroy (unit-less).

		UAV 1			UAV 2			UAV 3			UAV 4			UAV 5		
		M	S	E	M	S	E	M	S	E	M	S	E	M	S	E
Sem	PC	94.0	0.4	0.23	100	0.0	0.00	98.6	0.2	0.08	95.3	0.3	0.19	99.9	0.0	0.01
	Mesh	97.8	0.2	0.11	92.9	0.4	0.26	99.7	0.1	0.02	94.7	0.3	0.21	92.0	0.4	0.28
	Voxel	100	0.0	0.00	100	0.0	0.00	100	0.0	0.00	100	0.0	0.00	100	0.0	0.00
No Sem	PC	93.7	0.4	0.27	100	0.0	0.00	98.8	0.2	0.07	95.5	0.3	0.18	100	0.0	0.00
	Mesh	97.3	0.2	0.13	93.5	0.4	0.24	99.8	0.1	0.02	98.1	0.2	0.01	95.2	0.3	0.19
	Voxel	100	0.0	0.00	100	0.0	0.00	100	0.0	0.00	100	0.0	0.00	100	0.0	0.00

Table 6: Confidence of finding a path for five aerial examples. Values are based on 5000 samples. Legend: M:Mean (%), S:Standard Deviation (%), E:Entroy (unit-less).

semantic and semantic case. Due to its higher complexity (*i.e*. sampling from geometry and semantics), the semantic cases demand more time in total per sample than the non-semantic cases. Comparing the representations against each other, the point clouds and mesh show a similar behavior and narrow distribution around a clear maximum respectively. The histogram of timing for the voxel representation is more spread.

6.2. Aerial Paths

As in the above ground path analysis, this section represents an extension of the aerial (UAV) example in the last experiment of the paper (in § 5.3). We consider five long aerial paths in the Enschede scene for all representations. For each path in each representation we draw 5,000 samples. The statistics of path existence for each representation is shown in Tab. 6. In addition, Fig. 12 and Fig. 13 presents the spatial distribution of found paths. Importantly, we note that uav path 5 was presented in § 5.3 of the paper.

The mesh and point cloud representations exhibit a larger variance in their path distribution (Fig. 13). The source of this variability is the underlying RRT method, which is used in these representations to determine a valid path. However, despite the spatial variability, the RRT algorithm is successful in finding a path, with reported path existence above the 90% in all representations. Voxel representations are all consistent, with a 100% existence, in addition, their spatial variability is very small. This can be interpreted as high model confidence on the class free-space.

The histograms of path length for aerial paths, Fig. 14(b) confirm the spread observation noted earlier. It reveal a greater spread in the distribution for the point cloud and mesh representations, while the voxel paths always have the same length. The *time per sample*, Fig. 14(a), shows a similar spread for all representations. However, we see that the point cloud and mesh representations are substantially slower, *i.e*., an order of magnitude, than the voxel representations. This difference follows from using RRT as opposed to Dijkstra's algorithm.

References

[1] D. Bonbouzid, R. Busa-Fekete, N. Casagrande, F.-D. Collin, and B. Kégl. MULTIBOOST: A Multiple-purpose Boosting Package. *JMLR*, 2012.

[2] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[3] R. Cabezas, O. Freifeld, G. Rosman, and J. W. Fisher III. Aerial Reconstructions via Probabilistic Data Fusion. *CVPR*, 2014.

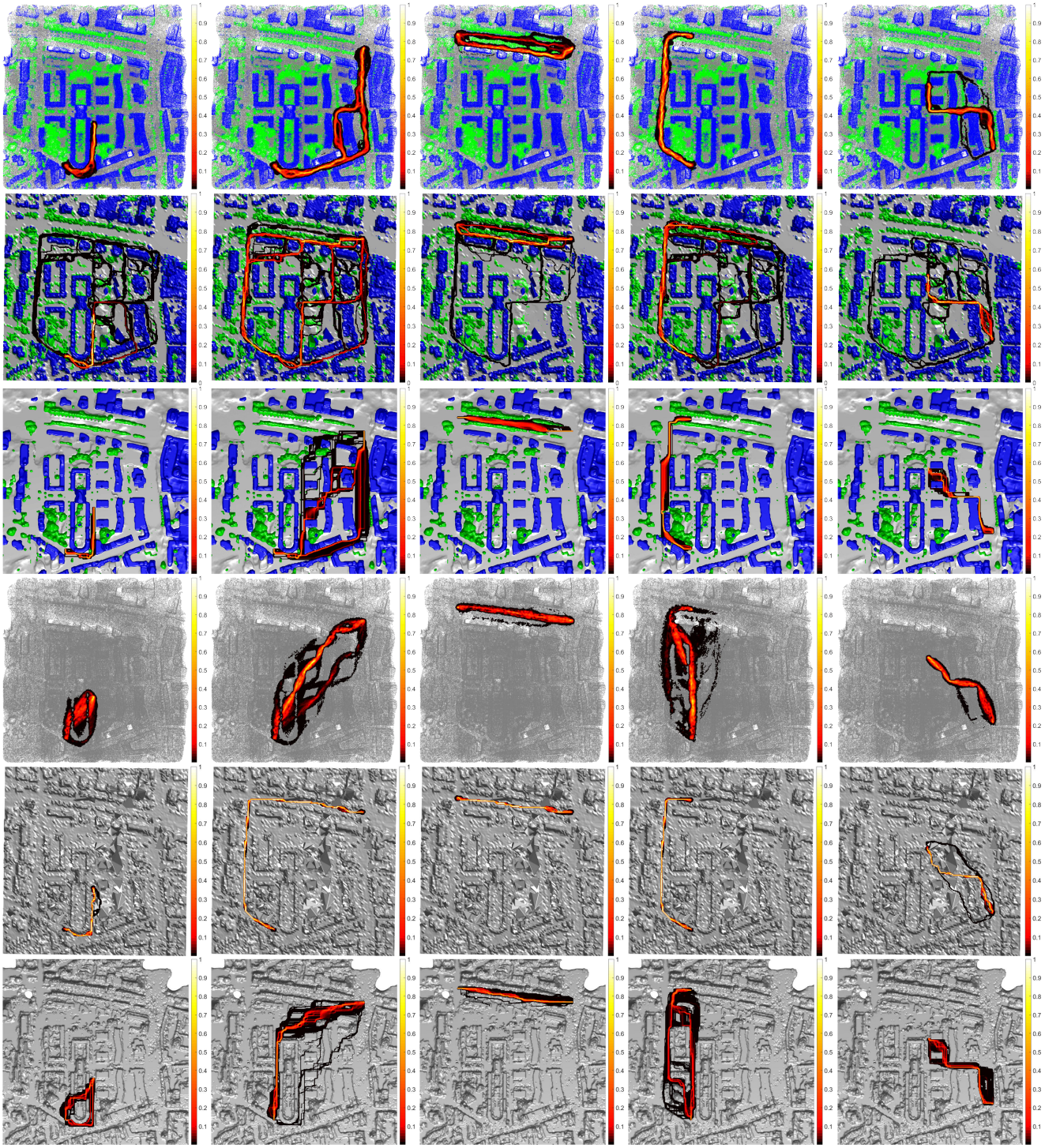


Figure 9: *Columns*: ground paths for five different starting/ending locations for each representation. *Rows:* representations: pc, mesh, voxel, semantic and non-semantic respectively. Yellow indicates high while black indicates low path density.

[4] R. Cabezas, J. Straub, and J. W. Fisher III. Semantically-Aware Aerial Reconstruction from Multi-Modal Data. *ICCV*, 2015.

[5] N. Chehata, L. Guo, and C. Mallet. Airborne lidar feature selection for urban classification using random forests.

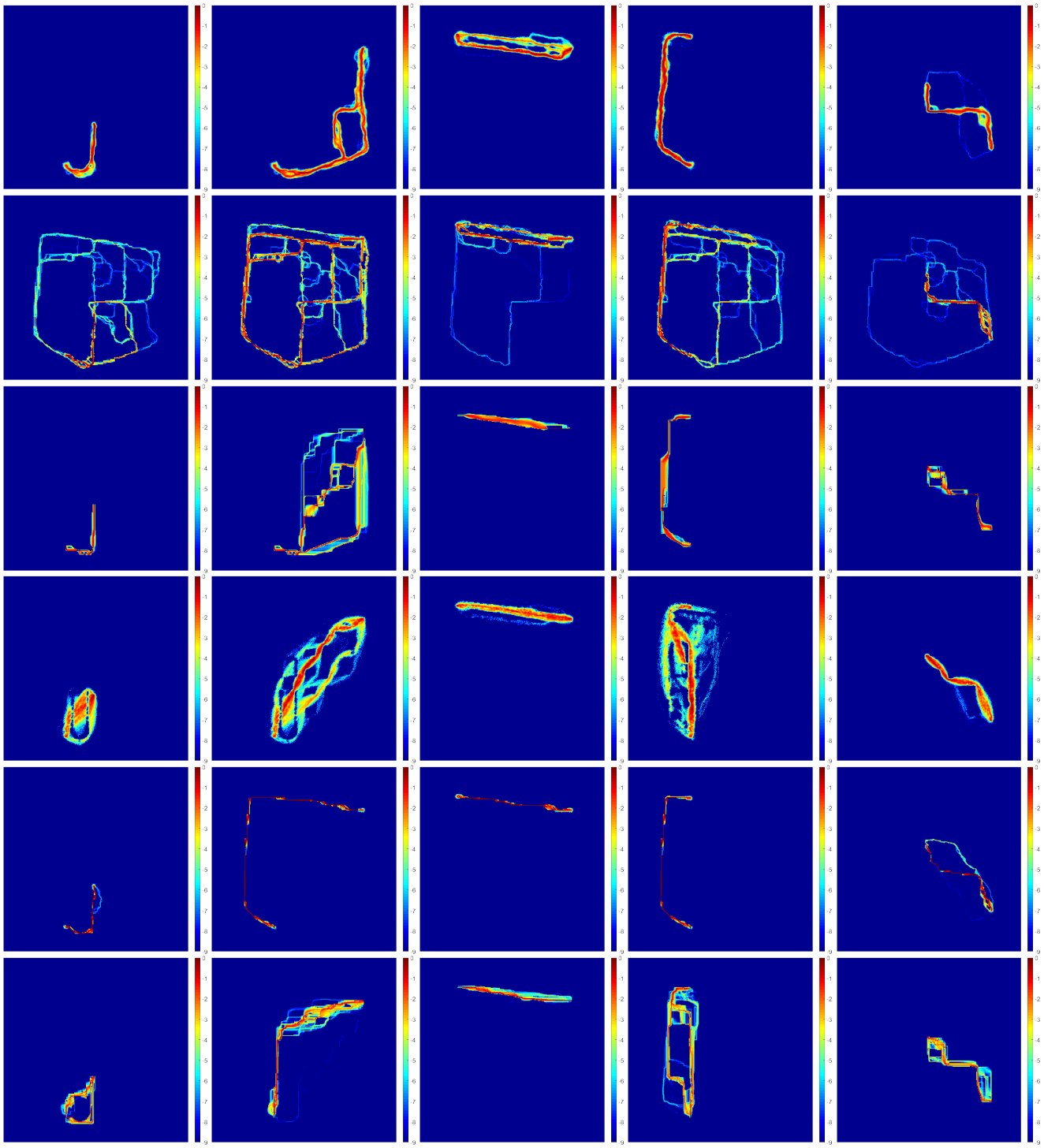
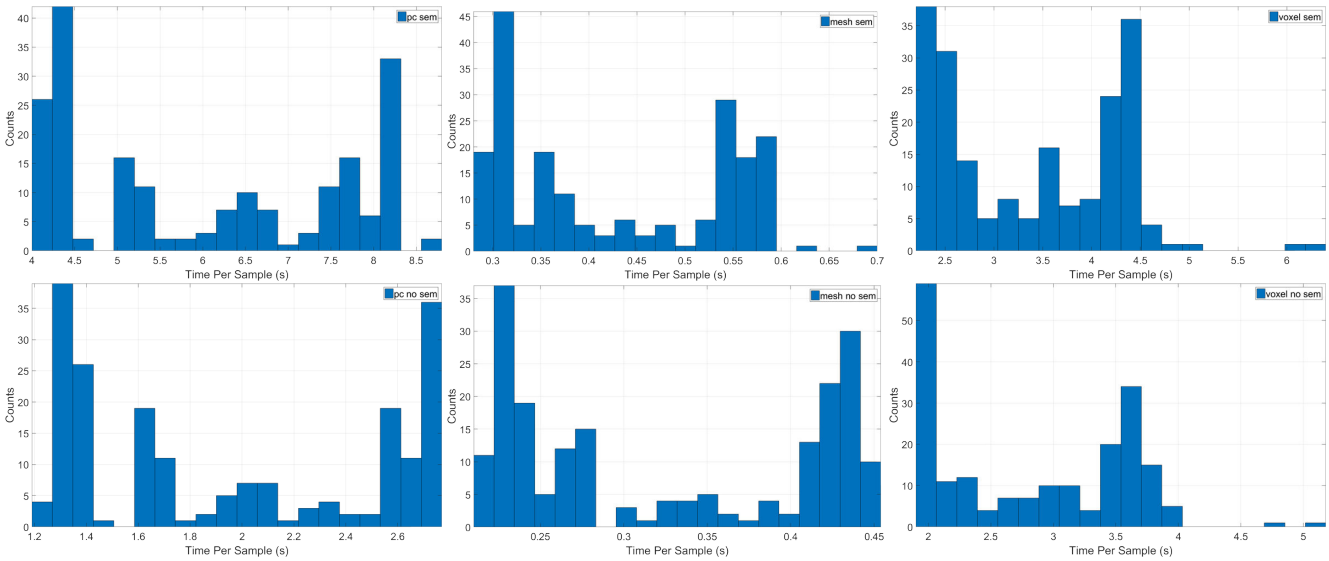


Figure 10: Alternative view to results of Fig. 9.

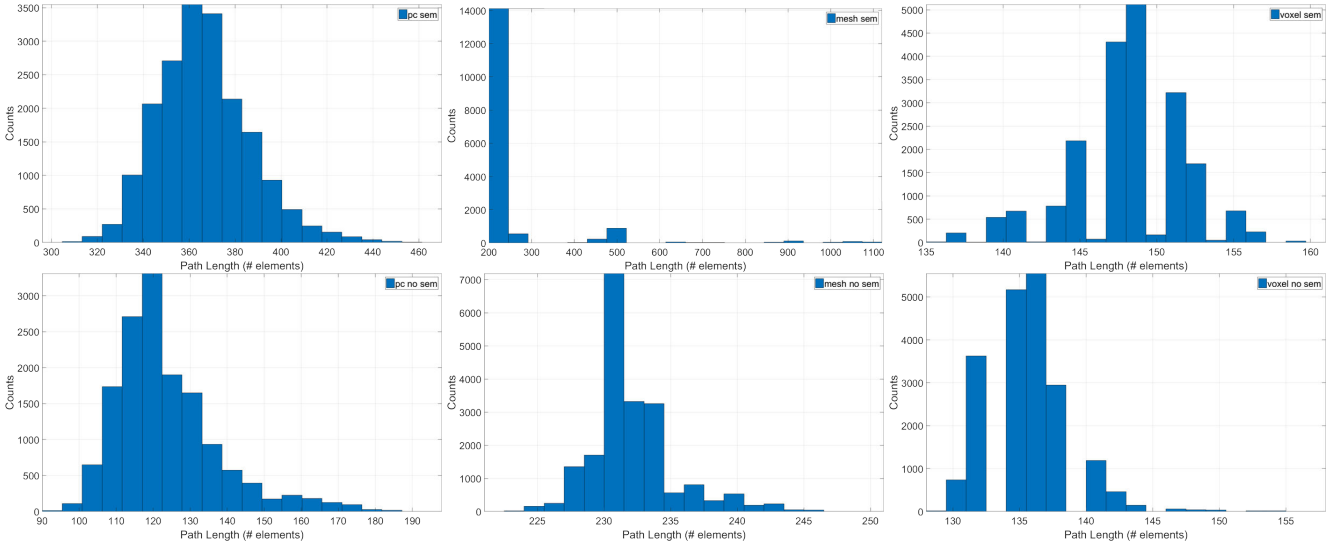
IntArch-PhRS, 2012.

[6] M. Fernandez and S. Williams. Closed-form expression for the poisson-binomial probability density function. *IEEE Transactions on Aerospace and Electronic Systems*, 2010.

[7] C. Hane, C. Zach, A. Cohen, R. Angst, and M. Pollefeys. Joint 3D scene reconstruction and class segmentation. In



(a) Time Per Sample. *L-R:* PC, mesh and voxel, semantic and non-semantic respectively.



(b) Path length. *L-R:* PC, mesh and voxel, semantic and non-semantic respectively.

Figure 11: Details for the *path length* and *time per sample* for "Pedestrian 1" path.

CVPR, 2013.

[8] A. Papoulis and S. U. Pillai. *Probability, random variables, and stochastic processes*. Mc-Graw Hill, 1991.

[9] R. B. Rusu, N. Blodow, M. Zoltan, A. Soos, and B. Michael. Towards 3D point cloud based object maps for household environments. *Autonomous Systems Journal*, 2008.

[10] Slagboom en Peeters Aerial Survey. http://www.slagboomenpeeters.com/3d.htm.

[11] C. Wu. VisualSFM: A Visual Structure from Motion System, 2011.

[12] C. Zach. Fast and High Quality Fusion of Depth Maps. *3DV*, 2008.

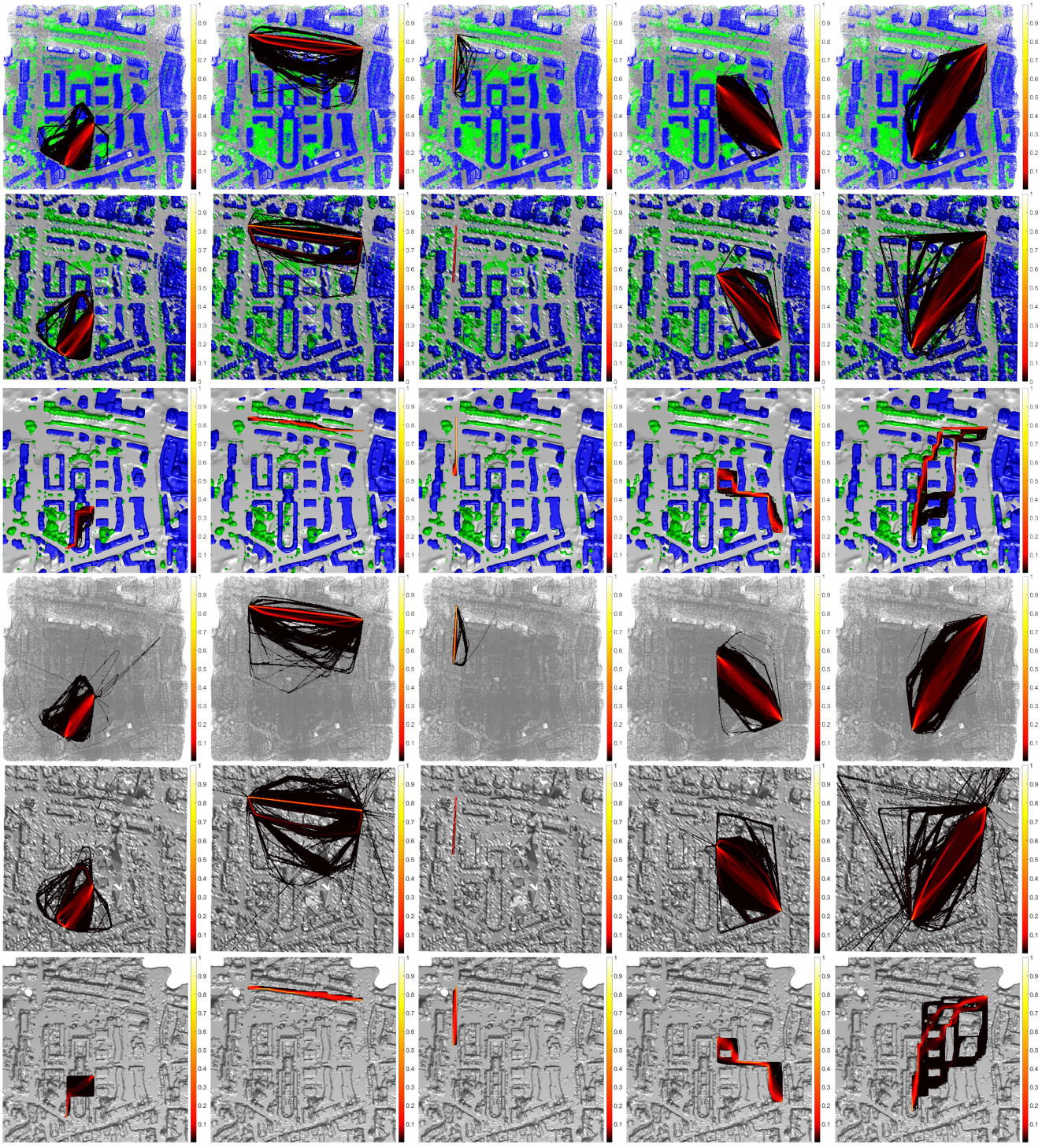


Figure 12: *Columns*: aerial paths for five different starting/ending locations for each representation. *Rows:* representations: pc, mesh, voxel, semantic and non-semantic respectively. Yellow indicates high while black indicates low path density.

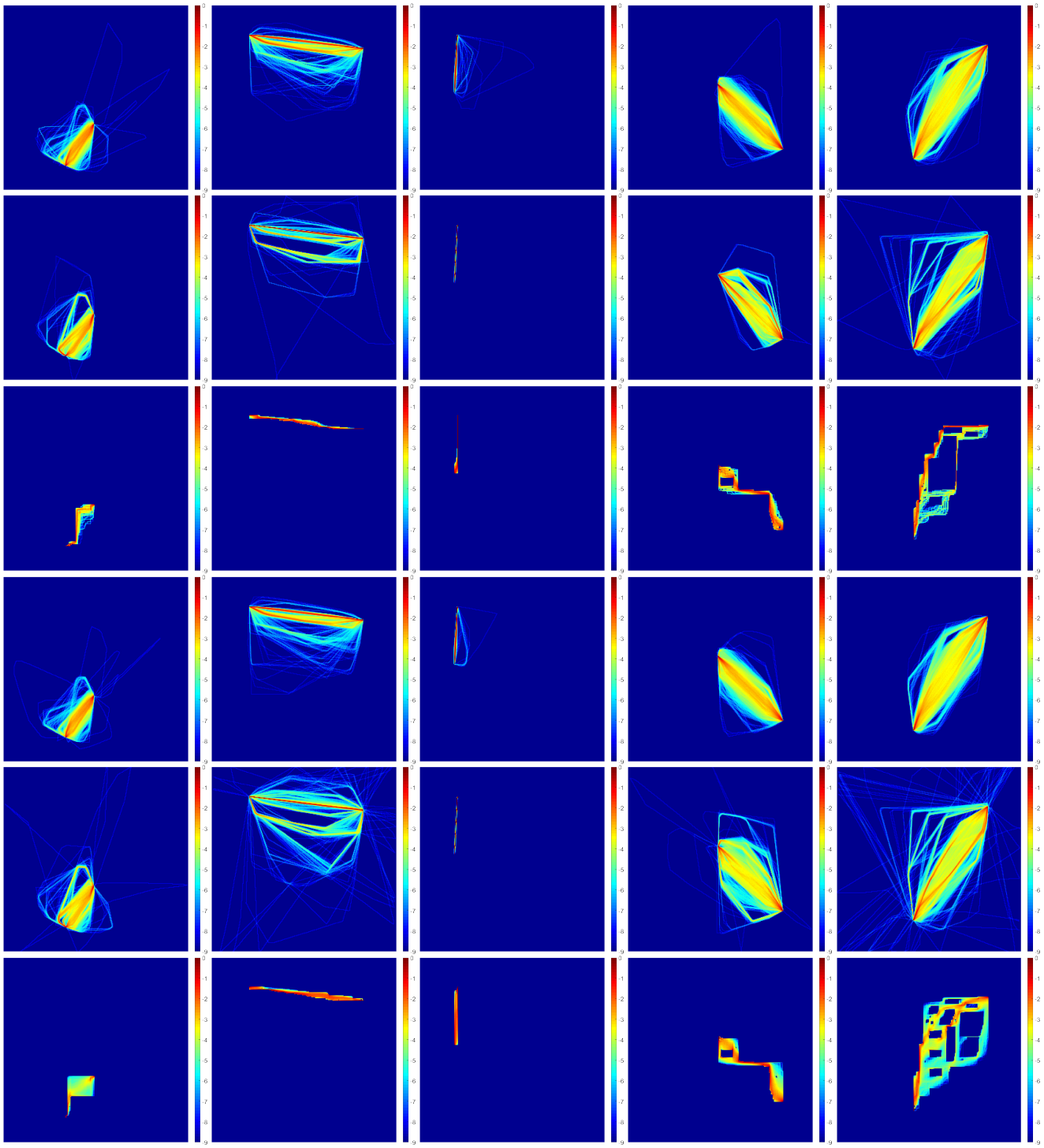
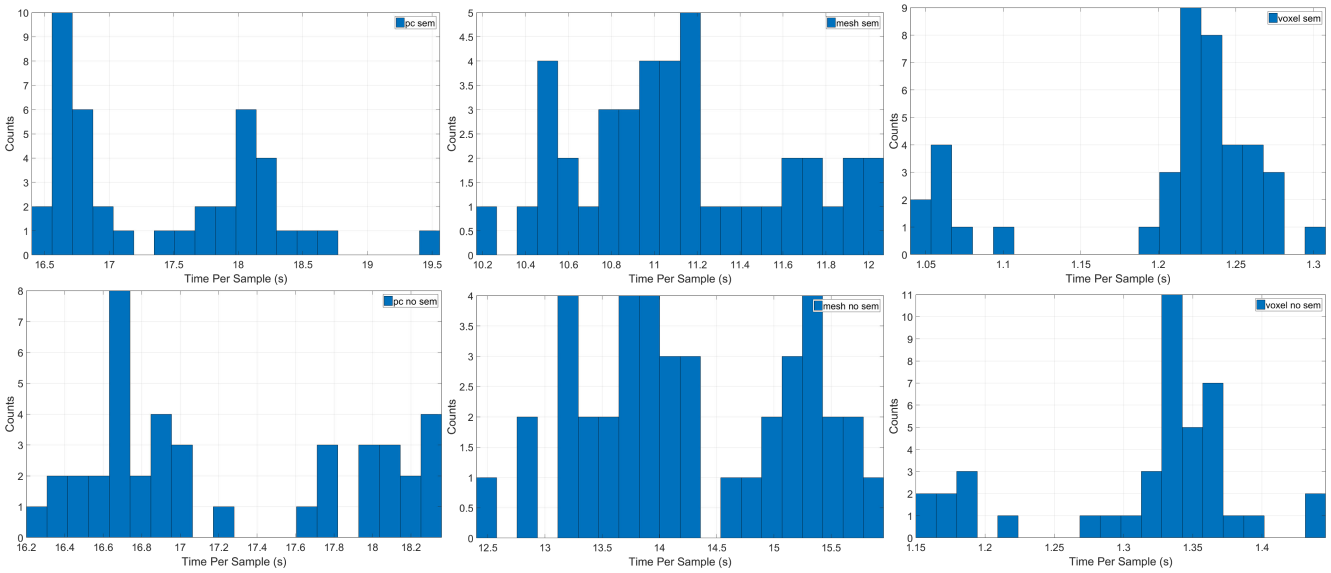
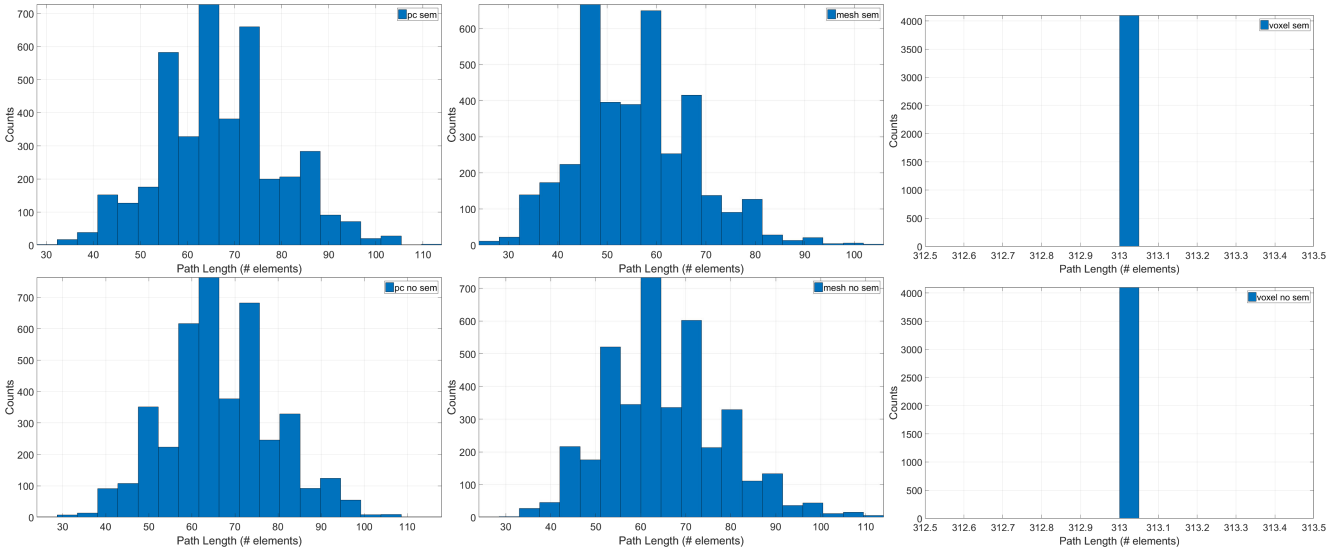


Figure 13: Alternative view to results of Fig. 12.



(a) Time Per Sample. *L-R:* PC, mesh and voxel, semantic and non-semantic respectively.



(b) Path length. *L-R:* PC, mesh and voxel, semantic and non-semantic respectively.

Figure 14: Details for the *path length* and *time per sample* for "UAV 5" path.