Learnable Earth Parser: Discovering 3D Prototypes in Aerial Scans

Romain Loiseau[1, 2] romain.loiseau@enpc.fr
Elliot Vincent[1,3] elliot.vincent@enpc.fr
Mathieu Aubry[1] mathieu.aubry@enpc.fr
Loic Landrieu[1, 2] loic.landrieu@enpc.fr

[1] LIGM, Ecole des Ponts, Univ Gustave Eiffel, CNRS, France
[2] Univ Gustave Eiffel, IGN, ENSG, LASTIG, France
[3] INRIA Paris, France

Abstract

We propose an unsupervised method for parsing large 3D scans of real-world scenes with easily-interpretable shapes. This work aims to provide a practical tool for analyzing 3D scenes in the context of aerial surveying and mapping, without the need for user annotations. Our approach is based on a probabilistic reconstruction model that decomposes an input 3D point cloud into a small set of learned prototypical 3D shapes. The resulting reconstruction is visually interpretable and can be used to perform unsupervised instance and low-shot semantic segmentation of complex scenes. We demonstrate the usefulness of our model on a novel dataset of seven large aerial LiDAR scans from diverse real-world scenarios. Our approach outperforms state-of-the-art unsupervised methods in terms of decomposition accuracy while remaining visually interpretable. Our code and dataset are available at https://romainloiseau.fr/learnable-earth-parser/.

1. Introduction

Modern aerial 3D scanning technologies open up unprecedented opportunities for environmental monitoring and economic intelligence. However, their practical use remains challenging due to the complexity of real-world scenes, the diversity of usage scenarios, and the difficulty of annotation. Therefore, our aim is to develop an approach that could help perform diverse tasks—from counting trees in a forest or identifying the various components of a factory to measuring the surface of greenhouses or monitor urban growth—all without human supervision.

To do so, we address two important limitations of existing 3D deep learning methods. First, they are often primarily designed, trained, and tested on synthetic [7, 46, 59, 72] or highly curated data [2, 30, 39], which fail to capture the endless variability of the real world. Moreover, they often

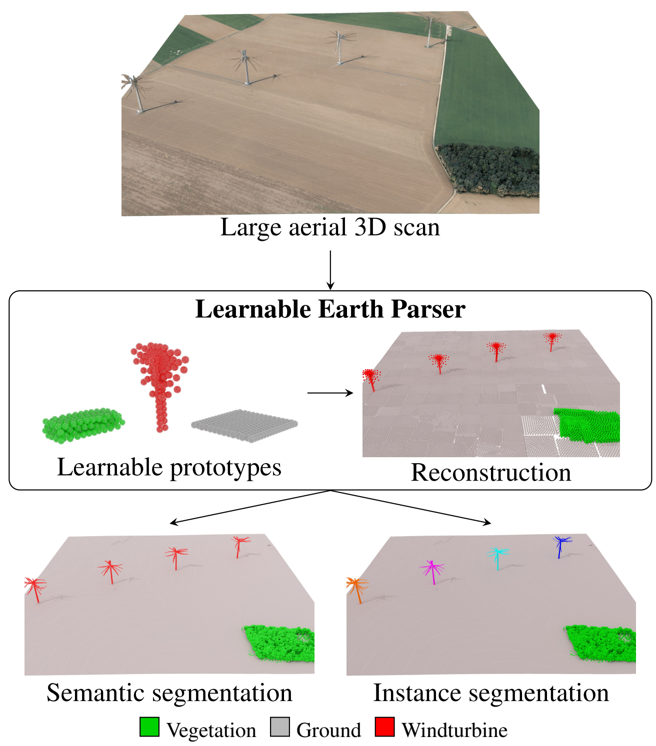


Figure 1. **Learnable Earth Parser.** Our unsupervised method takes large aerial 3D scans as input and model them with a small set of learned 3D prototypes. Our approach is trained without annotation and produce legible decompositions of complex scenes, which can be used for semantic and instance segmentation.

assume that annotations are available for tasks of interest. Second, even unsupervised approaches [1, 77] often rely on learning abstract feature representations, making them difficult to interpret [76]. Although some work has attempted to decompose 3D shapes into meaningful components without supervision [11, 43, 53, 67], they were all designed on simple synthetic shapes and none generalizes to real data.

To overcome these limitations, we present the Learnable Earth Parser, an unsupervised deep learning method

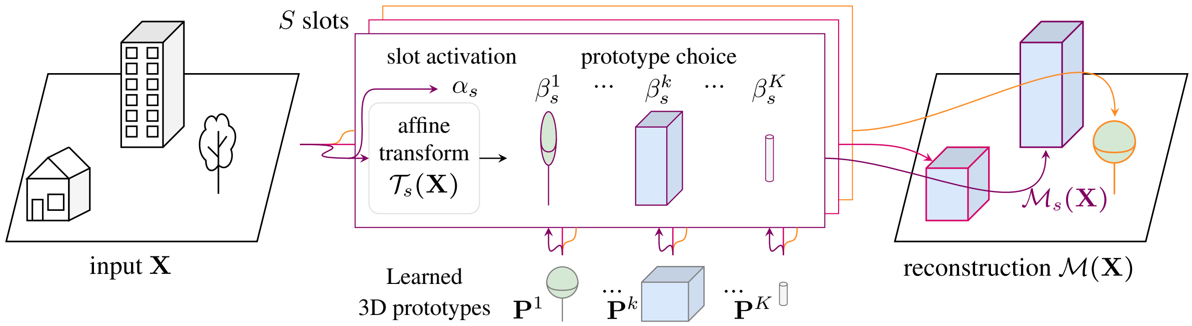


Figure 2. **Method Overview.** Our model approximates an input point cloud $\mathbf{X}$ with S slot models. Each slot maps $\mathbf{X}$ to an affine 3D deformation $\mathcal{T}_s(\mathbf{X})$, a slot activation probability α_s, and the joint probabilities $\beta_s^1, \cdots, \beta_s^K$ of the slot being activated and choosing one of the K learnable prototype point clouds $\mathbf{P}^1, \cdots, \mathbf{P}^K$. The output $\mathcal{M}_s(\mathbf{X})$ of an activated slot s is obtained by applying the transformation $\mathcal{T}_s(\mathbf{X})$ to its most likely prototype. Non-activated slots do not contribute to the output.

designed to decompose large-scale 3D point clouds into interpretable parts. Our model learns a small set of 3D prototypical shapes that are selected, positioned, rotated, and resized to reconstruct an input point cloud. We introduce a novel probabilistic formulation that enables the design of a reconstruction loss for learning jointly the 3D prototypes, but also to select and position them.

To evaluate the effectiveness of our approach, we created a new open-access dataset consisting of 7 aerial LiDAR scans, covering 7.7km^2 and containing 98 million 3D points with annotations in diverse urban and natural environments. Our results demonstrate that the Learnable Earth Parser learns decompositions superior to traditional and deep learning baselines, leading to convincing performance for semantic and instance segmentation, as shown in Figure 1. We believe that our contributions provide researchers and practitioners with new tools and resources to tackle the challenges of real-world 3D data.

2. Related work

Our proposed unsupervised method uses point cloud reconstruction as a proxy to learn to decompose large aerial point clouds and is evaluated on a novel and diverse dataset of 3D scans. In the following, we briefly present related works for primitive-based point cloud decomposition, automatic decomposition of LiDAR data, and an overview of existing aerial LiDAR datasets.

Primitive-based point cloud decomposition. Modeling shapes as a set of primitives such as generalized cylinders [5] or superquadrics [3] has a rich history in vision and graphics [26]. Classical applications include reverse engineering [4], shape completion [61, 66], and shape editing [14]. A variety of methods have been developed to find primitives in unstructured 3D scenes, including seed growing techniques [33, 34], genetic algorithms [8], approaches [17, 38, 56, 60] based on RANSAC [13], and probabilistic methods [40, 71].

For this problem, like for many others in computer vision, deep learning has become the dominant paradigm. However, supervised methods [36, 80] are limited by the availability of annotated datasets. Following the seminal work of Tulsiani et al. [67], unsupervised approaches that simply rely on a reconstruction loss to learn primitive decomposition are the most common and the most closely related to our work [53, 55]. An important challenge for these approaches is to model a variable number of primitives. This has been addressed using recurrent networks [35, 62, 80], capsule networks [78], reinforcement learning strategies [67] or, most similar to our approach, computing the Chamfer distance based on a probabilistic model [53]. Our work is also related to approaches that learn prototypical shapes instead of being restricted to a predefined family of parametric primitives [11, 43].

However, most of these methods are designed, trained and evaluated on well-curated synthetic object datasets, such as ModelNet [72], ShapeNet [7], or D-FAUST [6], and are typically designed to handle single objects from known categories. In contrast, our approach can handle complex scenes composed of many objects with significant variety.

Decomposition of LiDAR scans. Automatically decomposing large LiDAR scans poses unique challenges due to their size and diversity [73]. Some previous approaches use simple shape primitives, such as lines [10, 21], planes [16, 19, 51], or volumes [37], but these may not be flexible enough to capture the complexity of real-world scans. Other approaches are designed for specific object classes, like trees [47] or buildings [24, 32], but they are limited in their ability to represent a wide range of shapes. In contrast, our Learnable Earth Parser overcomes these limitations by learning ad-hoc prototypes for each new scene, ensuring both expressivity and adaptability. LiDAR data are often treated as digital elevation models, *i.e.* images with pixel elevations [20, 23, 42]. Thus, our work is related to image-based primitive prediction [22, 29, 54, 57] and unsupervised multi-object image segmentation [41, 49, 64, 75]. However, 3D point clouds have higher precision and can better represent multi-layered structures such as forest areas.

Aerial LiDAR datasets. The increased availability of aerial LiDAR technology has led to the multiplication of open datasets [15, 52, 74, 79] of varying sizes from 1 to 10 km^2 [63, 68]. However, these scans are limited to dense urban environments and do not capture the challenge of modeling diverse terrains. Some specialized datasets focus on forested areas [27, 70]. Our proposed dataset is of similar scale, spanning 7.7km^2, but covers a variety of urban, natural, and rural scenes, making it more representative of the diversity of possible usage scenarios.

3. Method

Our goal is to learn to break down a point cloud into simpler and more easily understandable components. To achieve this, we propose an *analysis-by-synthesis* approach where we train a highly-constrained model $\mathcal{M}$ to approximate a point cloud as a combination of learned 3D shapes.

3.1. Probabilistic Scene Reconstruction Model

As illustrated in Figure 2, our model first selects up to S shapes from K learnable 3D shapes, then positions and deforms them to best approximate an input point cloud $\mathbf{X}$. We propose a probabilistic formulation of the selection process, which can be seen as an extension of the model of Paschalidou *et al*. [53] to multiple free-form shapes instead of a single parametric family.

Learnable shape prototypes. Following Loiseau *et al*. [43], we define K point clouds $\mathbf{P}^1, \cdots, \mathbf{P}^K$ that we refer to as *prototypes*. Each prototype is meant to represent a single instance of a recurring 3D structure in the considered scene. The points' coordinates are free parameters of the model and learned directly.

Scene reconstruction model. Our full model $\mathcal{M}$ is the combination of S reconstruction models $\mathcal{M}_s$, which we refer to as *slots* in analogy to the Slot Attention approach [41]. Each slot contributes to the final reconstruction only if it is activated. Slot activation is determined by a binary variable a_s: $\mathcal{M}_s$ is activated if and only if $a_s = 1$. The output of $\mathcal{M}(\mathbf{X})$ is the combination of the reconstructions from all activated slots:

$$\mathcal{M}(\mathbf{X}) = \bigcup_{\substack{s=1\cdots S \\ a_s=1}} \mathcal{M}_s(\mathbf{X}) \,. \tag{1}$$

Each slot model outputs a point cloud which is the deformation of one learnable prototype chosen from $\mathbf{P}^1, \cdots, \mathbf{P}^K$.

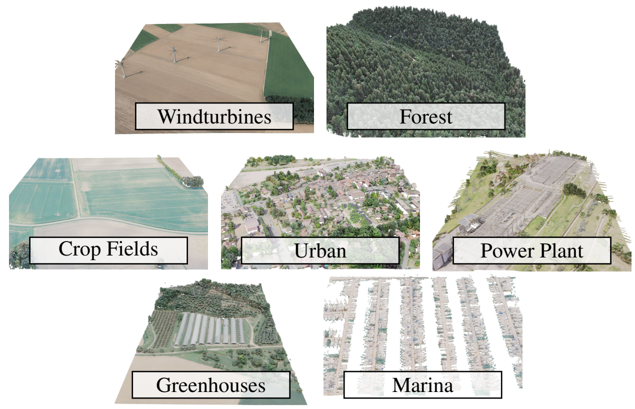


Figure 3. **Earth Parser Dataset.** Our dataset contains 7 scenes representing various urban and natural environments acquired by aerial LiDAR. The illustration of the power plant and the greenhouses display the complete scenes, while other ones display a subset of each scene (between 25 and 50% of the total area).

We associate to each slot s a network $\mathcal{T}_s$ which maps $\mathbf{X}$ to an affine transformation in 3D space $\mathcal{T}_s(\mathbf{X})$. The output $\mathcal{M}_s(\mathbf{X})$ of slot s is determined by a variable $b_s \in \{1, \cdots, K\}$. If $b_s = k$, then the output of $\mathcal{M}_s(\mathbf{X})$ is $\mathbf{Y}_s^k$, the result of applying the transformation $\mathcal{T}_s(\mathbf{X})$ to the prototype $\mathbf{P}^k$:

$$\mathbf{Y}_s^k = \mathcal{T}_s(\mathbf{X})[\mathbf{P}^k] \,. \tag{2}$$

Please note that $\mathbf{Y}_s^k$ is a function of $\mathbf{X}$. However, to keep our notations simple, we omit this dependence.

Probabilistic modeling. We make our reconstruction model probabilistic by modeling a and b as random variables following (multi-)Bernoulli distributions. We call α_s the probability that slot s is activated and β_s^k the probability that it is activated and selects the prototype k:

$$p(a_s = 1) = \alpha_s \quad , \quad p(a_s = 1, b_s = k) = \beta_s^k \,. \tag{3}$$

For each slot s, we predict the vector $(1 - \alpha_s, \beta_s^1, \ldots \beta_s^K)$ with a neural network taking the point cloud $\mathbf{X}$ as input and finishing with a softmax layer. Again, we don't write the dependency of the α_s and β_s on $\mathbf{X}$ explicitly to simplify the notations. The complete model $\mathcal{M}(\mathbf{X})$ and the slots models $\mathcal{M}_s(\mathbf{X})$ can now be seen as random variables, producing different potential reconstructions with probabilities given by α and β. During inference, we consider only slots with $\alpha_s > 0.5$ and select the prototype with highest β_s^k. However, during training, we compute all reconstructions $\mathbf{Y}_s^k$.

3.2. Training Losses

Given a large 3D scene, we train our model by sampling square patches $\mathbf{X}$ from the scene. For each batch of patches, we minimize a loss composed of a reconstruction loss $\mathcal{L}_{\text{rec}}$

and several regularization terms $\mathcal{L}_{\text{reg}}$ implementing different priors:

$$\mathcal{L}(\mathcal{M}) = \mathbb{E}_{\mathbf{X}}\left[\mathcal{L}_{\text{rec}}\left(\mathcal{M},\mathbf{X}\right)\right] + \mathcal{L}_{\text{reg}}\left(\mathcal{M}\right) \ . \tag{4}$$

Reconstruction loss. We define the reconstruction loss $\mathcal{L}_{\text{rec}}$ as the sum of two losses:

$$\mathcal{L}_{\text{rec}}(\mathcal{M},\mathbf{X}) = \mathcal{L}_{\text{acc}}(\mathcal{M},\mathbf{X}) + \mathcal{L}_{\text{cov}}(\mathcal{M},\mathbf{X}) \ . \tag{5}$$

$\mathcal{L}_{\text{acc}}$ encourages likely reconstructions of $\mathcal{M}(\mathbf{X})$ to accurately approximate $\mathbf{X}$, and $\mathcal{L}_{\text{cov}}$ ensures coverage, i.e., that each point of $\mathbf{X}$ is well-reconstructed by at least one activated model. We define each term using the asymmetric Chamfer distance d between two point clouds $\mathbf{X}$ and $\mathbf{Y}$:

$$d(\mathbf{X},\mathbf{Y}) = \frac{1}{|\mathbf{X}|}\sum_{x\in\mathbf{X}}\min_{y\in\mathbf{Y}}\|x-y\|_2^2 \ , \tag{6}$$

with $|.|$ the number of points in a point cloud. We write $d(x,\mathbf{Y})$ the distance between the point x and its closest point in $\mathbf{Y}$.

We define $\mathcal{L}_{\text{acc}}$ as the average over all slots s of the expected distance between $\mathcal{M}_s(\mathbf{X})$ and $\mathbf{X}$:

$$\mathcal{L}_{\text{acc}}(\mathcal{M},\mathbf{X}) = \frac{1}{S}\sum_{s=1}^{S}\mathbb{E}_{a_s,b_s}\left[d\left(\mathcal{M}_s(\mathbf{X}),\mathbf{X}\right)\right] \tag{7}$$

$$= \frac{1}{S}\sum_{s=1}^{S}\sum_{k=1}^{K}\beta_s^k d\left(\mathbf{Y}_s^k,\mathbf{X}\right) \ . \tag{8}$$

Conversely, we define $\mathcal{L}_{\text{cov}}$ as the average over all points x of $\mathbf{X}$ of the expected distance between x and its closest point in the reconstruction:

$$\mathcal{L}_{\text{cov}}(\mathcal{M},\mathbf{X}) = \frac{1}{|\mathbf{X}|}\sum_{x\in\mathbf{X}}\mathbb{E}_{a,b}\left[\min_{s|a_s=1} d\left(x,\mathcal{M}_s(\mathbf{X})\right)\right] \ . \tag{9}$$

Following the ideas of Paschalidou *et al.* [53], we first define $\Delta(x,s)$ as the expected distance between x and $\mathcal{M}_s(\mathbf{X})$ conditionally to the slot s being activated:

$$\Delta(x,s) = \mathbb{E}_{b_s|a_s=1}\left[d\left(x,\mathcal{M}_s(\mathbf{X})\right)\right] \tag{10}$$

$$= \frac{1}{\alpha_s}\sum_{k=1}^{K}\beta_s^k d\left(x,\mathbf{Y}_s^k\right) \ . \tag{11}$$

Next, we compute for each point x a permutation σ_x of $[1,S]$ such that $\Delta(x,\sigma_x(s))$ is non-decreasing, i.e.:

$$\Delta\left(x,\sigma_x\left(1\right)\right) \le \dots \le \Delta\left(x,\sigma_x\left(S\right)\right) \ . \tag{12}$$

If s is the closest activated slot to x, then all the slots closer to x must be deactivated. This observation leads us to rewrite $\mathcal{L}_{\text{cov}}$ as follows:

$$\mathcal{L}_{\text{cov}}(\mathcal{M},\mathbf{X}) = \frac{1}{|\mathbf{X}|}\sum_{x\in\mathbf{X}}\sum_{s=1}^{S}\Delta(x,s)\alpha_s\prod_{r<\sigma_x(s)}\left(1-\alpha_{\sigma_x^{-1}(r)}\right) \ , \tag{13}$$

with σ^{-1} the inverse permutation of σ.

Regularization losses. As is often necessary with fully unsupervised reconstruction methods [43, 48, 49], we define several regularization losses implementing priors on the model output to prevent degenerate local minima:

- To encourage the slot activation to be sparse, we use the following loss penalizing slot activation:

$$\mathcal{L}_{\text{act}}(\mathcal{M}) = \sum_{s=1}^{S}\mathbb{E}_{\mathbf{X}}\left[\alpha_s\right] \ . \tag{14}$$

- To avoid slots and prototypes that are never used, we use the following losses, which we compute batch-wise:

$$\mathcal{L}_{\text{slot}}(\mathcal{M}) = -\sum_{s=1}^{S}\min\left(\frac{\mathbb{E}_{\mathbf{X}}\left[\alpha_s\right]}{\sum_{t=1}^{S}\mathbb{E}_{\mathbf{X}}\left[\alpha_t\right]},\epsilon_S\right) \ , \tag{15}$$

$$\mathcal{L}_{\text{proto}}(\mathcal{M}) = -\sum_{k=1}^{K}\min\left(\frac{\mathbb{E}_{\mathbf{X}}\left[\sum_{s=1}^{S}\beta_s^k\right]}{\sum_{s=1}^{S}\mathbb{E}_{\mathbf{X}}\left[\alpha_s\right]},\epsilon_K\right) \ , \tag{16}$$

with ϵ_S and ϵ_K hyperparameters setting the smallest acceptable relative use frequency for a slot or prototype.

The full regularization loss is a weighted sum of the three losses described above:

$$\mathcal{L}_{\text{reg}} = \lambda_{\text{act}}\mathcal{L}_{\text{act}} + \lambda_{\text{slot}}\mathcal{L}_{\text{slot}} + \lambda_{\text{proto}}\mathcal{L}_{\text{proto}} \ , \tag{17}$$

where we use $\lambda_{\text{act}} = 10^{-4}$, $\lambda_{\text{slot}} = \lambda_{\text{proto}} = 0.1$ and $\epsilon_S = \epsilon_K = 0.1$ in all our experiments on the Earth Parser Dataset.

3.3. Training and Implementation Details

Model configuration. We process the input point cloud $\mathbf{X}$ using a similar architecture as in [53]. We first voxelize it with a $64\times64\times64$ grid and map it to a vector using a sequence of 6 3D sparse convolutions [9] and 6 strided convolutions. The resulting representation is then transformed using one linear layer for each slot. These features are decoded by simple 2-layer MLPs: one generates the distribution parameters α_s and β_s^k, and the other ones the parameters of the 3D transformations $\mathcal{T}_s(X)$. The transformations include an anisotropic scaling, a y-axis tilt of $\pm\pi/10$, a rotation around the z-axis, and a translation, in this specific order. We use $S=64$ slots and $K=6$ prototypes as default parameters. See the supplementary material for details.

The intensity of the return signal of each point is available in the LiDAR scans. We associate each prototype with a single learnable intensity parameter and perform the Chamfer distance (Equation 6) in 4 dimensions: spatial coordinates normalized to $[0,1]^3$ and intensity to $[0,0.1]$.

Curriculum learning. The network predicts simultaneously the slot's probability distributions and their deformations. This results in many concurrent degrees of freedom and can make the training process unstable. Therefore, following Monnier *et al.* [49] and Loiseau *et al.* [43], we implement a multi-stage curriculum learning strategy. We first

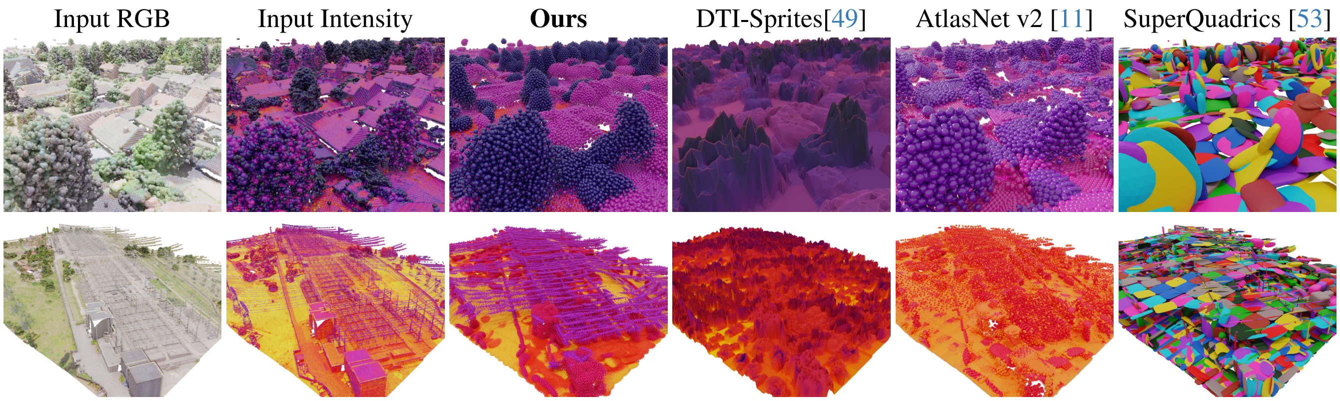


Figure 4. **Reconstruction Quality.** We show two partial scenes with their RGB and intensity values, as well as their reconstruction by our method and competing models. We use the prototypes' intensity to color the points or pixels. As SuperQuadrics does not model the intensity, we use a random colour for each quadric.

initialize the prototypes as point clouds uniformly sampled from a random cuboid and gradually unfreeze the model parameters in the following order: (i) translation, rotation, tilt, slot activation, and choice of prototype; (ii) intensities of the prototypes, when available; (iii) scales of the prototypes; (iv) positions of the prototypes' 3D points; (v) anisotropic scalings of the prototypes. As shown in Section 4, each step of this curriculum scheme improves the performances.

Prototypes selection. We automatically select the number of prototypes for a complete scene using a simple greedy algorithm. We measure the increase of reconstruction loss when preventing the model from selecting each prototype individually. We remove the prototype with the lowest increase if it is lower than 5%, and iterate.

Implementation details. Our model is trained separately for each scene by randomly sampling square patches. During training, the patches are subsampled to a maximum 10^5 points. Each stage of the curriculum is trained until convergence. We use the ADAM optimizer [28] with a learning rate of 10^{-4} and default parameters. See the supplementary material for preprocessing, training and evaluation details.

4. Results

In this section, we assess the ability of our method to parse complex 3D aerial data. We give in Section 4.1 an overview of our proposed dataset of aerial LiDAR scans. In Section 4.2, we then discuss our evaluation metrics and baselines. Finally, we present a quantitative (Section 4.3) and qualitative (Section 4.4) analysis of our results.

4.1. Earth Parser Dataset

We introduce a new dataset to train and evaluate parsing methods on large, uncurated aerial LiDAR scans. We use data from the French Mapping Agency associated to the LiDAR-HD project [45]. Each scan is composed of several airborne LiDAR acquisitions taken at different angles, leading to a minimum resolution of 20 points/m^2. The points are associated with their laser reflectance (intensity), and colorized based on asynchronous aerial photography.

We selected 7 scenes, covering over 7.7km^2 and a total of 98 million 3D points, with diverse content and complexity, such as dense habitations, forests, or complex industrial facilities. We associate most 3D points with a coarse semantic label, such as ground, building, or vegetation. The characteristics of the scenes are detailed in Table 2 and each is visualized in Figure 3.

4.2. Evaluation Metrics and Baselines

We quantitatively evaluate the performances for reconstruction and semantic segmentation of our model and several unsupervised scene decomposition approaches.

Evaluation metrics. As our goal is to summarize a point cloud using few prototypes, the quality of the reconstruction is critical. We measure it with the symmetric Chamfer distance ("Cham." in the Tables) between the input and the output point clouds of our model.

By associating a class to each prototype's points, we can propagate labels from the reconstruction to the input cloud and perform semantic segmentation. We evaluate the quality of this segmentation with the class-averaged Intersection-over-Union (mIoU) metric.

In a practical scenario, an operator can manually annotate the points of the 3D prototypes, allowing for the segmentation of the entirety of $\mathbf{X}$ with minimal effort. To perform automatic evaluation, we follow the standard practice for evaluating clustering and unsupervised segmentation methods [25, 31, 48]: we assign to each prototype's point the most frequent class of its closest point of the input after the reconstruction.

Table 1. **Results on the Earth Parser Dataset.** We report the quality of the reconstruction (Cham.) and semantic segmentation (mIoU) on each of the scenes of our Earth Parser Dataset. While our method does not always provide the most faithful reconstructions, it leads to the most accurate point classification.

	Rec.	Semantic	Crop Fields		Forest		Greenhouses		Marina		Power Plant		Urban		Windturbines	
			Cham.	mIoU	Cham.	mIoU	Cham.	mIoU	Cham.	mIoU	Cham.	mIoU	Cham.	mIoU	Cham.	mIoU
k-means (i,z) [44]	✗	✓	—	93.8	—	71.5	—	39.3	—	41.4	—	42.8	—	56.5	—	87.6
SuperQuadrics [53]	3D	✗	0.86	—	1.04	—	0.60	—	0.93	—	0.58	—	0.40	—	13.5	—
DTI-Sprites [49]	2.5D+i	✓	6.10	83.2	14.59	40.2	5.36	42.0	6.16	41.4	5.36	29.0	2.99	47.3	36.19	25.9
AtlasNet v2 [11]	3D+i	✓	1.07	43.1	1.58	71.4	0.56	49.1	**0.73**	42.1	0.45	41.6	0.63	48.8	9.47	48.1
Ours	3D+i	✓	**0.72**	**96.9**	**0.88**	**83.7**	**0.40**	**91.3**	0.82	**78.7**	**0.44**	**52.2**	**0.29**	**83.2**	**6.65**	**93.4**

Table 2. **Earth Parser Dataset.** Our proposed dataset is composed of 7 diverse scenes acquired by aerial LiDAR.

Name	Surface in km^2	# points $\times 10^6$	annotation ratio in %	num. of classes
Crop Fields	1.1	19.7	77.4	2
Forest	1.1	46.7	97.8	2
Greenhouses	0.1	1.3	95.6	3
Marina	0.1	0.5	92.7	2
Power Plant	0.2	8.6	78.4	4
Urban	1.1	15.7	95.9	3
Windturbines	4.2	5.6	99.8	3
Total	7.7	98.3	91.6	—

Table 3. **Ablation Study.** We evaluate the effect of our prototype selection post-processing, our model's degrees of freedom, and our different regularization losses.

		Urban		Marina	
		Cham.	mIoU	Cham.	mIoU
	Learnable Earth Parser	0.29	83.2	0.82	**78.7**
	↳ w/o post-processing	0.28	**83.7**	0.96	78.3
expressivity	↳ w/o aniso-scale	0.33	82.4	1.04	67.2
expressivity	↳ w/o prototypes	0.36	68.3	1.07	42.8
expressivity	↳ w/o scales	0.55	58.9	1.33	40.8
expressivity	↳ w/o intensities	0.55	58.7	1.09	40.8
losses	↳ w/o $\mathcal{L}_{\text{act}}$	**0.17**	54.1	**0.80**	56.9
losses	↳ w/o $\mathcal{L}_{\text{slot}}$	0.25	77.8	0.81	43.7
losses	↳ w/o $\mathcal{L}_{\text{proto}}$	0.28	57.2	0.97	40.7

Baselines. We adapt several unsupervised approaches for scene reconstruction and/or semantic segmentation tasks to provide comparisons for our approach:

- **k-means.** We cluster the points of the input with the k-means algorithm [44] using as many clusters as we use prototypes. We obtain the best results by using a combination of the point's intensity and elevation as features for clustering. We then assign to each centroid its most frequent class, and propagate this label to the entire cluster, leading to a semantic segmentation. This method does not reconstruct the input, but gives us a simple and surprisingly strong baseline for semantic segmentation.
- **SuperQuadrics revisited.** We use the method of Paschalidou *et al*. [53] to learn to approximate scenes with an adaptive number of superquadrics [3]. It provides a baseline for reconstruction and a qualitative comparison for instance segmentation, shown in supplementary material.
- **DTI-Sprites.** We use the point cloud to construct a digital elevation model, *i.e*. a 2.5D image of resolution 32×32 where each pixel has an elevation and intensity value. We adapt the unsupervised image decomposition approach of Monnier *et al*. [49] to break down this image into a set of 2.5D *sprites*. We evaluate the reconstruction and segmentation by sampling 25 3D point per pixel, transferring the pixel's label to the points, and interpolating their elevations.
- **AtlasNet v2.** This extension [11] of AtlasNet [18] uses a fixed number of learnable prototype point clouds to reconstruct its input. It can be evaluated for both reconstruction and semantic segmentation in a way similar to ours. We extend it to handle intensity in a manner akin to our approach, which improves its segmentation results.

Similar to our method, we train all baselines except k-means by sampling square patches in each scene. Figure 4 shows the output of the reconstruction methods.

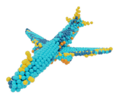
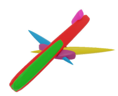
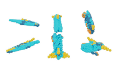
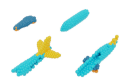
4.3. Quantitative Results

We compare the performance of our approach with the proposed baselines on the Earth Parser Dataset, as well as two publicly available datasets.

Earth Parser Dataset. We provide quantitative reconstruction and semantic segmentation results in Table 1, and illustrations in Figure 4. Despite being highly constrained, our model yields the best reconstruction in 6 out of 7 scenes. Moreover, we significantly outperform the other methods for semantic segmentation across all scenes.

Despite its simplicity, the k-means baseline provides strong semantic segmentation performance, beating the other baselines in 5 of 7 annotated scenes. DTI-Sprites [49] has lower reconstruction and segmentation quality, which

Table 4. **Synthetic Shapes.** We train our method on all planes from ShapeNet-Part [58], with random rotations around z-axis. We show the reconstruction of an input plane and the prototypes learned on the dataset.

	AtlasNet v2 [11]	SuperQuadrics [53]	**Ours**
Recons.			
Protos.		—	
Cham.	1.46	2.91	**1.34**
mIoU	34.5	—	**68.6**

	Cham.	mIoU
k-means (i,z) [44]	—	52.7
Superquadrics [53]	0.33	—
DTI-Sprites [49]	2.25	53.6
AtlasNet v2 [11]	0.31	54.9
Ours	**0.29**	**82.7**

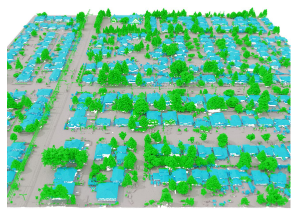


Figure 5. **Results on DALES [68].** We report quantitative and qualitative results for one tile from DALES.

is expected as it models a 3D point cloud in 2.5D. AtlasNet v2 [11] provides good reconstructions but segmentation fails for scenes such as Crop Fields or Urban due its inability to adapt its prototype usage to the input. On the contrary, SuperQuadrics [53] can adjust the number of superquadric it uses and, to some degree, their shape. However, this method uses a single parametric family for all prototypes and fails to reconstruct complex real-world scenes such as Power Plant. Thanks to its probabilistic slot selection, our method can handle inputs with a varying number of objects using only a small set of learned prototypical shapes.

Ablation study. We evaluate the impact of various components of our model and report the results in Table 3. First, we observe that the prototype selection post-processing has limited impact on the quality of the prediction and reconstructions, but allows us to adapt and significantly decrease the number of prototypes used for each scene. Second, we evaluate the impact of reducing the expressivity of our model. We successively remove: (i) the anisotropic scaling of $\mathcal{T}$, and the possibility of learning the prototype's (ii) points position, (iii) scale, and (iv) intensity. As expected, each degree if liberty removed decreases the quality of the 3D reconstruction and segmentation.

We study the impact of the different regularization introduced in Section 3.2. The losses related to slot activation have a marginal effect on reconstruction quality but significantly affect the performance of semantic segmentation. This suggests that using slots sparingly but equally is important for prototypes to specialize for specific objects, but is not necessary to the expressivity of the reconstruction model. The prototype activation loss prevents unused and possibly degenerate prototypes and therefore improves both segmentation and reconstruction.

We evaluate the impact of reducing the number of prototypes or slots. When we set $K = 3$ instead of 6, the average semantic segmentation mIoU drops by 26.3 points, while reconstruction is minimally affected, showing only a 2.3% increase in Chamfer distance. Conversely, setting $S = 32$ instead of 64 leads to a significant +47.3% increase of Chamfer distance, with a modest drop 0.4 of semantic segmentation mIoU. These findings highlight the importance of maintaining sufficient diversity in the prototypes to enable specialization for different object types, whereas the number of slots strongly influences the expressiveness of the reconstruction model.

ShapeNet. We evaluate our model on 2690 planes from ShapeNet-Part [58], whose points are annotated as *wing*, *engine*, *tail*, or *body*. We randomly rotate the shapes around the z-axis during training and evaluation. We report in Table 4 the performances and reconstructions for our approach, AtlasNet v2 [11], and SuperQuadrics revisited [53]. Our method handle rotations better than AtlasNet v2, and manages to successfully locate the tail of the planes. While SuperQuadrics' reconstructions make sense qualitatively, they are worse in terms of accuracy than ours, and do not enable semantic segmentation.

DALES. We also train and evaluate our model on DALES [68], a dataset of aerial LiDAR scans of urban area, with a restricted class set: *ground*, *vegetation*, and *buildings*. Our model significantly outperform competing reconstruction-based approaches, and yields similar performance than for the Urban tile of the Earth Parser Dataset; see Figure 5.

4.4. Qualitative Results

Instance segmentation. We can perform instance segmentation simply by considering each slot as a different instance. This is particularly interesting for parsing natural woodlands, a key endeavor for forest management [50] and biomass estimation [12]. While this task has a long history of handcrafted approaches [69], current deep learning approaches are mostly limited to artificial or low-density forests [65]. As shown qualitatively in Figure 6, our Learnable Earth Parser can learn without any supervision to separate individual trees in dense forests, or boats in a marina. We evaluate quantitatively the performance of our model in terms of instance segmentation, we counted manually 88

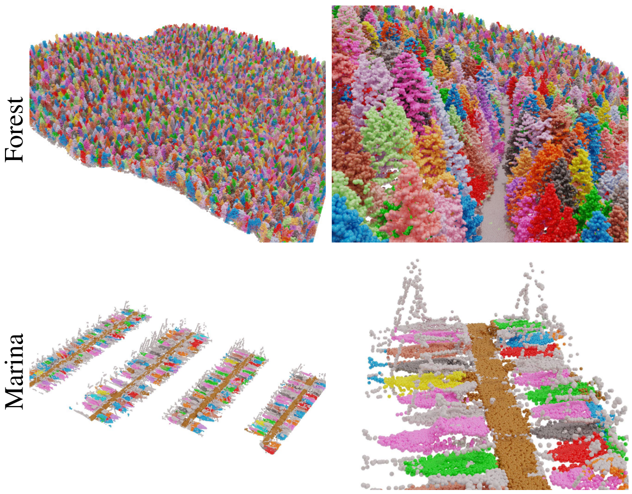


Figure 6. **Instance Segmentation.** We can identify the reconstruction of each slot as a separate instance, allowing us to perform instance segmentation of complex data such as dense natural forests or a marina. For this visualization, we considered the points associated to "trees" or "boat hull" prototypes and color each of their instance randomly.

boat instances in 10 distinct portions of the Marina scene. We then computed the Mean Relative Error (MRE) of the prediction given by the number of boat-like prototypes (prototype #3 in Figure 8) in these zones. Our method yields a MRE of $7,4\%$, 2.6 times lower than DTI-Sprites [49], demonstrating its strong performances.

Semantic segmentation. After annotating the prototypes, our model can perform convincing semantic segmentation of complex scenes, as shown in Figure 7.

Interpretable prototypes. In Figure 8, we show prototypes learned on our Earth Parser Dataset with colors showing the associated semantic label for each point. These prototypes give at a glance insights on the content of these real-world scenes. Our model is able to learn a wide variety of shapes, such as boats' masts, wind turbines, or greenhouses. We also observe that the prototypes are typically associated with a single object type. Empirically, the average class distribution of the nearest neighbors of each prototypes' point exhibits the same normalized entropy (0.22) as a Bernoulli distribution with probability 0.964. The highly constrained nature of the deformations $\mathcal{T}$ prevents slots from repurposing the same prototypes for different objects, making the learned prototypes easy to identify and manually annotate.

5. Conclusion

We introduced a novel unsupervised method for parsing complex real-world aerial scans into simple parts using a small set of learned prototypical shapes. We demonstrate

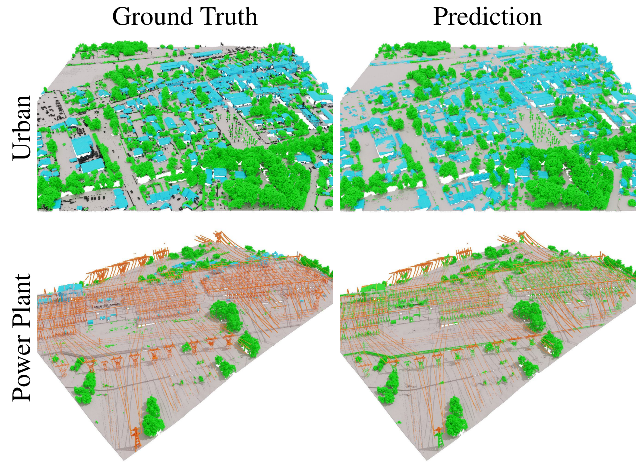


Figure 7. **Semantic Segmentation.** Our model can perform semantic segmentation of large real-world scene based on annotated prototypes. Black points in the ground truth are unannotated.

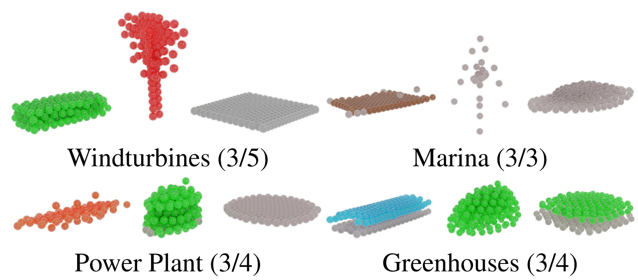


Figure 8. **Learned Prototypes.** We display three of the prototypes chosen during the selection process for various scenes.

the quality and interpretability of our results on a novel dataset of aerial LiDAR scans. To the best of our knowledge, we are the first to demonstrate the possibility of performing deep unsupervised 3D shape analysis on such a challenging real-world dataset. We believe that our results open new perspectives for computer-assisted environment monitoring and economic intelligence.

Acknowledgements This work was supported in part by ANR project READY3D ANR-19-CE23-0007, ANR under the France 2030 program under the reference ANR-23-PEIA-0008, and was granted access to the HPC resources of IDRIS under the allocation 2022-AD011012096R2 made by GENCI. The work of MA was partly supported by the European Research Council (ERC project DISCOVER, number 101076028). The scenes of Earth Parser Dataset were acquired and annotated by the LiDAR-HD project [45]. We thank Zenodo for hosting the dataset. We thank Zeynep Sonat Baltaci, Emile Blettery, Nicolas Dufour, Antoine Guédon, Helen Mair Rawsthorne, Tom Monnier, Damien Robert, Mathis Petrovich and Yannis Siglidis for inspiring discussions and valuable feedback.

References

[1] Antonio Alliegro, Davide Boscaini, and Tatiana Tommasi. Joint supervised and self-supervised learning for 3D real world challenges. In *ICPR*, 2021. 1

[2] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3D semantic parsing of large-scale indoor spaces. In *CVPR*, 2016. 1

[3] Alan H Barr. Superquadrics and angle-preserving transformations. *IEEE Computer graphics and Applications*, 1981. 2, 6

[4] Pál Benkő, Ralph R Martin, and Tamás Várady. Algorithms for reverse engineering boundary representation models. *Computer-Aided Design*, 2001. 2

[5] Thomas Binford. Visual perception by computer. In *Proc. IEEE Conf. on Systems and Control*, 1975. 2

[6] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J Black. Dynamic FAUST: Registering human bodies in motion. In *CVPR*, 2017. 2

[7] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1, 2

[8] YH Chen and CY Liu. Quadric surface extraction using genetic algorithms. *Computer-Aided Design*, 1999. 2

[9] Spconv Contributors. SPConv: Spatially sparse convolution library. https://github.com/traveller59/spconv, 2022. 4

[10] Yang Cui, Qingquan Li, Bisheng Yang, Wen Xiao, Chi Chen, and Zhen Dong. Automatic 3-d reconstruction of indoor environment with mobile laser scanning point clouds. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019. 2

[11] Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir Kim, Bryan Russell, and Mathieu Aubry. Learning elementary structures for 3D shape generation and matching. *NeurIPS*, 2019. 1, 2, 5, 6, 7

[12] António Ferraz, Sassan Saatchi, Clément Mallet, Stéphane Jacquemoud, Gil Gonçalves, Carlos Alberto Silva, Paula Soares, Margarida Tomé, and Luisa Pereira. Airborne LiDAR estimation of aboveground forest biomass in the absence of field inventory. *Remote Sensing*, 2016. 7

[13] Martin A Fischler and Robert C Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. 2

[14] Ran Gal, Olga Sorkine, Niloy J Mitra, and Daniel Cohen-Or. iWIRES: An analyze-and-edit approach to shape manipulation. In *ACM SIGGRAPH*. 2009. 2

[15] Weixiao Gao, Liangliang Nan, Bas Boom, and Hugo Ledoux. SUM: A benchmark dataset of semantic urban meshes. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2021. 3

[16] Xuming Ge, Bo Wu, Yuan Li, and Han Hu. A multi-primitive-based hierarchical optimal approach for semantic labeling of ALS point clouds. *Remote Sensing*, 2019. 2

[17] Paulo FU Gotardo, Olga Regina Pereira Bellon, and Luciano Silva. Range image segmentation by surface extraction using an improved robust estimator. In *CVPR*, 2003. 2

[18] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3D surface generation. In *CVPR*, 2018. 6

[19] Stéphane Guinard, Loıc Landrieu, Laurent Caraffa, and Bruno Vallet. Piecewise-planar approximation of large 3d data as graph-structured optimization. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 2019. 2

[20] Florent Guiotte, Minh-Tan Pham, Romain Dambreville, Thomas Corpetti, and Sébastien Lefèvre. Semantic segmentation of LiDAR points clouds: Rasterization beyond digital elevation models. *IEEE Geoscience and Remote Sensing Letters*, 2020. 2

[21] Bo Guo, Qingquan Li, Xianfeng Huang, and Chisheng Wang. An improved method for power-line reconstruction from point cloud data. *Remote sensing*, 2016. 2

[22] Abhinav Gupta, Alexei Efros, and Martial Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *ECCV*, 2010. 2

[23] Norbert Haala and Claus Brenner. Extraction of buildings and trees in urban environments. *ISPRS journal of photogrammetry and remote sensing*, 1999. 2

[24] Jin Huang, Jantien Stoter, Ravi Peters, and Liangliang Nan. City3D: large-scale building reconstruction from airborne LiDAR point clouds. *Remote Sensing*, 2022. 2

[25] Xu Ji, Joao F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. *CVPR*, 2019. 5

[26] Adrien Kaiser, Jose Alonso Ybanez Zepeda, and Tamy Boubekeur. A survey of simple geometric primitives detection methods for captured 3D data. In *Computer Graphics Forum*. Wiley Online Library, 2019. 2

[27] Ekaterina Kalinicheva, Loic Landrieu, Clément Mallet, and Nesrine Chehata. Multi-layer modeling of dense vegetation from aerial lidar scans. In *CVPR Workshops*, 2022. 3

[28] Diederik P Kingma and Jimmy Ba. ADAM: A method for stochastic optimization. *ICLR*, 2014. 5

[29] Florian Kluger, Hanno Ackermann, Eric Brachmann, Michael Ying Yang, and Bodo Rosenhahn. Cuboids revisited: Learning robust 3D shape fitting to single RGB images. In *CVPR*, 2021. 2

[30] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 2017. 1

[31] Adam Kosiorek, Sara Sabour, Yee Whye Teh, and Geoffrey E Hinton. Stacked capsule autoencoders. *NeurIPS*, 2019. 5

[32] Florent Lafarge, Xavier Descombes, Josiane Zerubia, and Marc Pierrot-Deseilligny. Automatic 3D builfding reconstruction from DEMs: An application to PLEIADES simulations. In *ISPRS*, 2006. 2

[33] Aleš Leonardis, Alok Gupta, and Ruzena Bajcsy. Segmentation of range images as the search for geometric parametric models. *International Journal of Computer Vision*, 1995. 2

[34] Ales Leonardis, Ales Jaklic, and Franc Solina. Superquadrics for segmenting and modeling range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997. 2
[35] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. Grass: Generative recursive autoencoders for shape structures. *ACM Transactions on Graphics*, 2017. 2
[36] Lingxiao Li, Minhyuk Sung, Anastasia Dubrovina, Li Yi, and Leonidas J Guibas. Supervised fitting of geometric primitives to 3D point clouds. In *CVPR*, 2019. 2
[37] Minglei Li, Peter Wonka, and Liangliang Nan. Manhattan-world urban reconstruction from point clouds. In *ECCV*, 2016. 2
[38] Yangyan Li, Xiaokun Wu, Yiorgos Chrysathou, Andrei Sharf, Daniel Cohen-Or, and Niloy J Mitra. Globfit: Consistently fitting primitives by discovering global relations. In *ACM SIGGRAPH*. 2011. 2
[39] Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2D and 3D. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1
[40] Weixiao Liu, Yuwei Wu, Sipu Ruan, and Gregory S Chirikjian. Robust and accurate superquadric recovery: A probabilistic approach. In *CVPR*, 2022. 2
[41] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *NeurIPS*, 2020. 3
[42] Suresh K Lodha, Edward J Kreps, David P Helmbold, and Darren Fitzpatrick. Aerial LiDAR data classification using support vector machines. In *Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*. IEEE, 2006. 2
[43] Romain Loiseau, Tom Monnier, Mathieu Aubry, and Loïc Landrieu. Representing shape collections with alignment-aware linear models. In *3DV*, 2021. 1, 2, 3, 4
[44] J MacQueen. Classification and analysis of multivariate observations. In *5th Berkeley Symp. Math. Statist. Probability*, 1967. 6, 7
[45] IGN (French mapping agency). LiDAR-HD: A 3D mapping of France's soil and subsoil, 2021. 5, 8
[46] John McCormac, Ankur Handa, Stefan Leutenegger, and Andrew J.Davison. SceneNet RGB-D: Can 5M synthetic images beat generic imagenet pre-training on indoor segmentation? *ICCV*, 2017. 1
[47] Jie Mei, Liqiang Zhang, Shihao Wu, Zhen Wang, and Liang Zhang. 3d tree modeling from incomplete point clouds via optimization and l 1-mst. *International Journal of Geographical Information Science*, 2017. 2
[48] Tom Monnier, Thibault Groueix, and Mathieu Aubry. Deep Transformation-Invariant Clustering. *NeurIPS*, 2020. 4, 5
[49] Tom Monnier, Elliot Vincent, Jean Ponce, and Mathieu Aubry. Unsupervised layered image decomposition into object prototypes. In *ICCV*, 2021. 3, 4, 5, 6, 7, 8
[50] Felix Morsdorf, Erich Meier, Benjamin Kötz, Klaus I Itten, Matthias Dobbertin, and Britta Allgöwer. LIDAR-based geometric reconstruction of boreal type forest stands at single tree level for forest and wildland fire management. *Remote sensing of environment*, 2004. 7
[51] Liangliang Nan and Peter Wonka. Polyfit: Polygonal surface reconstruction from point clouds. In *ICCV*, 2017. 2
[52] Joachim Niemeyer, Franz Rottensteiner, and Uwe Soergel. Contextual classification of LiDAR data and building object detection in urban areas. *ISPRS journal of photogrammetry and remote sensing*, 2014. 3
[53] Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. Superquadrics revisited: Learning 3D shape parsing beyond cuboids. In *CVPR*, 2019. 1, 2, 3, 4, 5, 6, 7
[54] Despoina Paschalidou, Luc Van Gool, and Andreas Geiger. Learning unsupervised hierarchical part decomposition of 3D objects from a single RGB image. In *CVPR*, 2020. 2
[55] Despoina Paschalidou, Angelos Katharopoulos, Andreas Geiger, and Sanja Fidler. Neural parts: Learning expressive 3D shape abstractions with invertible neural networks. In *CVPR*, 2021. 2
[56] Michaël Ramamonjisoa, Sinisa Stekovic, and Vincent Lepetit. Monteboxfinder: Detecting and filtering primitives to fit a noisy point cloud. In *ECCV*. Springer, 2022. 2
[57] Lawrence G. Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963. 2
[58] Manolis Savva, Angel X. Chang, and Pat Hanrahan. Semantically-Enriched 3D Models for Common-sense Knowledge. *CVPR Workshop*, 2015. 7
[59] Manolis Savva, Angel X. Chang, and Pat Hanrahan. Semantically-Enriched 3D Models for Common-sense Knowledge. *CVPR 2015 Workshop on Functionality, Physics, Intentionality and Causality*, 2015. 1
[60] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient ransac for point-cloud shape detection. In *Computer Graphics Forum*, 2007. 2
[61] Ruwen Schnabel, Patrick Degener, and Reinhard Klein. Completion and reconstruction with primitive shapes. In *Computer Graphics Forum*. Wiley Online Library, 2009. 2
[62] Gopal Sharma, Rishabh Goyal, Difan Liu, Evangelos Kalogerakis, and Subhransu Maji. CSGNet: Neural shape parser for constructive solid geometry. In *CVPR*, 2018. 2
[63] Nina M Singer and Vijayan K Asari. DALES objects: A large scale benchmark dataset for instance segmentation in aerial LiDAR. *IEEE Access*, 2021. 3
[64] Dmitriy Smirnov, Michael Gharbi, Matthew Fisher, Vitor Guizilini, Alexei Efros, and Justin M Solomon. Marionette: Self-supervised sprite learning. *NeurIPS*, 2021. 3
[65] Chenxin Sun, Chengwei Huang, Huaiqing Zhang, Bangqian Chen, Feng An, Liwen Wang, and Ting Yun. Individual tree crown segmentation and crown width extraction from a heightmap derived from aerial laser scanning data using a deep learning framework. *Frontiers in plant science*, 2022. 7
[66] Minhyuk Sung, Vladimir G Kim, Roland Angst, and Leonidas Guibas. Data-driven structural priors for shape completion. *ACM Transactions on Graphics*, 2015. 2
[67] Shubham Tulsiani, Hao Su, Leonidas J Guibas, Alexei A Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *CVPR*, 2017. 1, 2

[68] Nina Varney, Vijayan K Asari, and Quinn Graehling. DALES: A large-scale aerial LiDAR data set for semantic segmentation. In *CVPR Workshop*, 2020. 3, 7

[69] Jari Vauhkonen, Liviu Ene, Sandeep Gupta, Johannes Heinzel, Johan Holmgren, Juho Pitkänen, Svein Solberg, Yunsheng Wang, Holger Weinacker, K Marius Hauglin, et al. Comparative testing of single-tree detection algorithms under different types of forest. *Forestry*, 2012. 7

[70] Ben G Weinstein, Sergio Marconi, Stephanie Bohlman, Alina Zare, and Ethan White. Individual tree-crown detection in RGB imagery using semi-supervised deep learning neural networks. *Remote Sensing*, 2019. 3

[71] Yuwei Wu, Weixiao Liu, Sipu Ruan, and Gregory S Chirikjian. Primitive-based shape abstraction via nonparametric bayesian inference. In *ECCV*, 2022. 2

[72] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *CVPR*, 2015. 1, 2

[73] Shaobo Xia, Dong Chen, Ruisheng Wang, Jonathan Li, and Xinchang Zhang. Geometric primitives in LiDAR point clouds: A review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2020. 2

[74] Zhen Ye, Yusheng Xu, Rong Huang, Xiaohua Tong, Xin Li, Xiangfeng Liu, Kuifeng Luan, Ludwig Hoegner, and Uwe Stilla. LASDU: A large-scale aerial LiDAR dataset for semantic labeling in dense urban areas. *ISPRS International Journal of Geo-Information*, 2020. 3

[75] Hong-Xing Yu, Leonidas J. Guibas, and Jiajun Wu. Unsupervised Discovery of Object Radiance Fields. *ICLR*, 2022. 3

[76] Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2021. 1

[77] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3d features on any point-cloud. In *ICCV*, 2021. 1

[78] Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 3D point capsule networks. In *CVPR*, 2019. 2

[79] SM Zolanvari, Susana Ruano, Aakanksha Rana, Alan Cummins, Rogerio Eduardo da Silva, Morteza Rahbar, and Aljosa Smolic. DublinCity: annotated LiDAR point cloud and its applications. *BMVC*, 2019. 3

[80] Chuhang Zou, Ersin Yumer, Jimei Yang, Duygu Ceylan, and Derek Hoiem. 3D-PRNN: Generating shape primitives with recurrent neural networks. In *ICCV*, 2017. 2