Start from Minimum Labeling: Learning of 3D Object Models and Point Labeling from a Large and Complex Environment

Quanshi Zhang†, Xuan Song†, Xiaowei Shao†, Huijing Zhao‡ and Ryosuke Shibasaki†

***Abstract*— A large category model base can provide object-level knowledge for various perception tasks of the intelligent vehicle system. The automatic and efficient construction of such a model base is highly desirable but challenging. This paper presents a novel semi-supervised approach to discover possible prototype models of 3D object structures from the point cloud of a large and complex environment, given a limited number of seeds in an object category. Our method incrementally trains the models while simultaneously collecting object samples. Considering the bias problem of model learning caused by bias accumulation in a sample collection, we propose to gradually differentiate the standard category model into several sub-category models to represent different intra-category structural styles. Thus, new sub-categories are discovered and modeled, old models are improved, and redundant models for similar structures are deleted iteratively during the learning process. This multiple-model strategy provides several interactive options for the category boundary to deal with the bias problem. Experimental results demonstrate the effectiveness and high efficiency of our approach to model mining from "big point cloud data".**

I. Introduction

An all-embracing category model base can directly provide high-level knowledge for AI tasks, such as object detection, segmentation, and tracking. Mining such bases from "big data" with a minimum human labeling has become an important AI area, which provides a continuous challenge to state-of-the-art algorithms. [38], [39] have recently been proposed to use a single labeled object to mine category model bases from cluttered RGB or RGB-D images.

Therefore, in this paper, we propose a semi-supervised method to learn category models from unlabeled "big point cloud data". The algorithm only requires to label a small number of object seeds in each object category to start the model learning, as shown in Fig. 1. Such design saves both the manual labeling and computation cost to satisfy the model-mining efficiency requirement.

We propose an iterative framework for collecting object samples and learning models, as shown in Fig. 2. Considering a large intra-category variation, we differentiate an entire object category into different structural styles as sub-categories, and use multiple models, each representing the structural distribution of the object samples within a sub-category. These sub-category models are incrementally learned along with sample collection. During the model learning, new sub-categories are discovered, similar sub-categories are merged, and models for the old sub-categories are gradually improved.

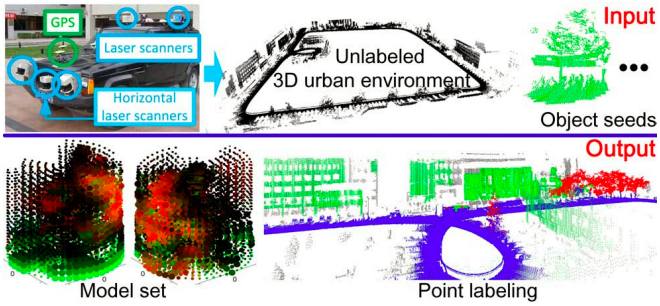

Fig. 1. Given an unlabeled point cloud of a large urban environment and a small number of object seeds in a category, we aim to mine a set of models to provide object-level knowledge for point labeling. The point cloud of a large environment can be considered as a kind of "big data", and thus we propose to limit both the manual labeling and the computation cost to ensure a high efficiency for model mining.

Unlike semi-supervised model learning based on image search engine results [16], the bias[1] problem is still intractable without the sifting of the search engine in large and complex environments. Owing to large intra-category shape variations and the prevalence of occlusions, object collection bias during the initial learning steps will affect further model learning, and errors will be accumulated into a significant model bias. The multiple-model strategy is a plausible way of dealing with the bias problem. Newly collected samples only affect models of their own categories, and the growth of the sample collection bias is therefore limited within certain biased sub-categories. We manually eliminate the biased models after the learning process is complete, to provide the correct category boundary.

The main contributions of this research can be summarized as follows. 1) To the best of our knowledge, this is the first proposal for an efficient mining of category models from "big point cloud data". With limited computation and human labeling, the method is oriented toward an efficient construction of a category model base. 2) A multiple-model strategy is proposed as a solution to the bias problem, and provides several discrete and selective category boundaries.

II. Related work

Point cloud processing has developed rapidly in recent years. In this section, we discuss a wide range of related

† Quanshi Zhang, Xuan Song, Xiaowei Shao, and Ryosuke Shibasaki are with Center for Spatial Information Science, University of Tokyo. {zqs1022,songxuan,shaoxw,shiba} at csis.u-tokyo.ac.jp

‡ Huijing Zhao is with Key Laboratory of Machine Perception (MoE), Peking University. zhaohj at cis.pku.edu.cn

[1] Without sufficient manual labeling, the model may either over fit to some special sub-categories within the target category, or shift to other categories in the incremental learning process.

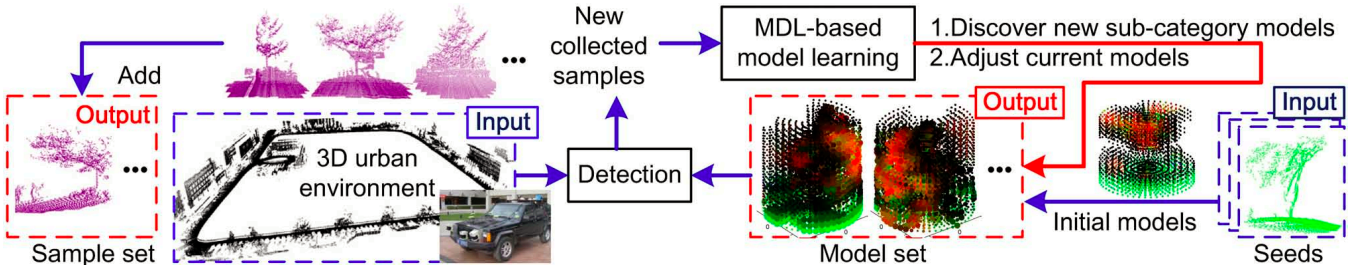

Fig. 2. What is a feasible and efficient way to mine category models? Given a limited number of 3D point clouds of the object seeds in an object category and a real 3D environment, our algorithm is able to automatically collect object samples with a large intra-category variation and learn the category model. In this iterative framework, blue and red rectangles indicate the inputs and outputs, respectively.

work to provide a better understanding of our method of category model mining.

Knowledge mining: The segmentation and classification (point labeling) of 3D point clouds are two kinds of 3D environment understanding [25], [26], [27], [28], [29], [30], [31], [32], [33], [36], [37]. [34] focus on unsupervised segmentation of 3D point clouds based on local features. Other studies have contributed to the learning of common structures of different categories from well-segmented 3D objects [13]. Munoz *et al.* [25], [33], Triebel *et al.* [26], and Anguelov *et al.* [32] made a breakthrough when they employed associative Markov networks (AMNs) with a max-margin strategy for supervised point cloud classification and segmentation.

However, knowledge mining is mainly required to be applied to unlabeled data in an unsupervised or semi-supervised manner. In addition, knowledge mining should focus on the learning of a general model for objects within an entire category, rather than just collecting a number of individual object samples based on local segmentation criteria.

Even so, we used the trained models to achieve point labeling (although the models can also be applied to other tasks, such as object retrieval and recognition), and performed experiments to compare its performance with the classical supervised AMN-based point classification.

Object-level global structure: A number of pioneering studies have contributed to the extraction of high and middle level structural knowledge. Hebert *et al.* [10] used some high-level shape assumptions to discover various structures in the environment, while Ruhnke *et al.* [19] learned a compact representation of a 3D environment based on Bayesian information criteria. Endres *et al.* [17] used latent Dirichlet allocation to discover 3D objects. These methods focus on patterns at the part level, whereas we expected to extract global structures with the correct object-level semantemes.

Category modeling: Closer to our field of category model mining, some approaches for the collection of 3D object samples have been proposed. Herbst *et al.* [11], [12] detected which objects had been moved across multiple depth images of the same scene, and Somanath *et al.* [22] detected the same objects appearing in different 3D scenes. Detry *et al.* [18] learned a general hierarchical object model from stereo data with clear edges.

Category model mining is not limited to the segmentation of objects with a specific shape or to recurrent objects, but also includes category discovery with large intra-category shape variations. From this viewpoint, the most closely related work involves unsupervised repetitive shape extraction [20], [21] and unsupervised 3D category discovery [1], where object samples are extracted automatically and classified into different categories.

By contrast, we aim to learn category models for point labeling, rather than simply collecting object samples or label points in a point cloud. The proposed model encodes the structural knowledge for a whole category with considerable intra-category variations. Meanwhile, our approach satisfies the efficiency requirement, *i.e.* a relatively small amount of computation and manual labeling, which is important for model mining from "big point cloud data". Nevertheless, we compare our approach with [20], [21] and [1] in experiments from the perspective of point labeling.

III. Algorithm

Our algorithm operates in a bootstrapping framework. We use current models to collect new samples and estimate their reliability. Strangely shaped or largely occluded samples are considered unreliable. Thus, *samples are weighted by their reliability, and this weighting is used as feedback to refine the current sub-category models and discover new sub-categories.*

A. Preliminaries: category modeling

We extend the cell-based object representation proposed in our previous work [1] to represent the structure distribution of samples within a sub-category. We divide the object into cells, and extract the local shape features from each cell. The model encodes the point-occupying probability of each cell and its local feature distributions among the object samples.

We use a cylinder template to describe the cell division, as shown in Fig. 3. Objects are placed into the 3D space of a vertical cylinder. The cylinder size is determined by the actual size of the object seeds. The cylinder is divided into F floors and L layers ($F = 16, L = 8$). It is then further divided into $N = 50$ parts using radial planes. We therefore obtain a total of $FLN = 6400$ cells.

Sample: The sample is placed at the center of the cylinder. We calculate the local feature for each cell to represent the shape of the corresponding local point cloud. Inspired by the spectral analysis of point clouds [23], we use a cuboid to fit the point cloud in each cell. The edge length of the cuboid is the square root of the eigenvalues of the point covariance matrix. We estimate the cuboid volume

Algorithm 1 Learn object models and collect object samples simultaneously

Input: The point cloud of a large and complex environment and k pre-labeled object seeds in a category.
Output: A set of models and collected object samples.
Initialization: Generate an initial model from each object seed to form the initial model set. Initialize the sample set consisting of the object seeds and their top-ranked matched samples in the environment.
repeat
1. Use the current models to estimate the reliability of all samples in the sample set (see Section III-B).
2. Produce new candidate models from the pure breeding of new samples and the hybridization between new samples and current models (see Section III-C.1).
3. MDL-based model selection using reliability-weighted samples (see Section III-C.2).
4. Search the top-ranked samples corresponding to the current models in the environment, and add them to the sample set (see Section III-B).

until no new samples can be well matched to the models.

using its edge length, and take the volume as the local feature. Obviously, *line-shape* cells and *surface-shape* cells have small feature values, whereas *cloud-shape* cells have large values. The local features of cells that do not contain any 3D points are defined as *none*. Thus, an object sample can be represented as a vector of local features $S = <s_i>$, where i is the cell index.

Model: The model encodes the point-occupying probability and local feature distributions in each cell, as $m = \{P, \mu, Var\}$, where $P = <p_i>$, $\mu = <\mu_i>$, $Var = <var_i>$. For each cell i, p_i indicates its probability of containing points, and μ_i and Var_i denote the mean value and variance of its local feature when it contains points, respectively.

Matching: We use object matching to compute similarities between models and samples. As objects are brute-force searched in the environment (details follow in Section III-B), they can be simply sampled at the center of the cylinder. Thus, we only consider horizontal rotations in matching.

For cell i in the sample, $\theta(i)$ denotes the corresponding cell in the model with horizontal rotation θ. The matching value between model $m = \{P, \mu, Var\}$ and sample S is calculated as follows:

$$match(S,m) = \max_{\theta} \frac{\sum_{i:s_i \neq none, p_{\theta(i)} \neq 0} Cell_i}{\sqrt{\sum_i \delta(s_i) \sum_i p_{\theta(i)}}} \quad (1)$$

$match(S,m)$ is normalized based on both the sample and model sizes. $\delta(s_i)$ is 1 if $s_i \neq none$, and 0 otherwise. $Cell_i$ is the local matching value of cell i in the sample with rotation angle θ, and it is assumed to follow a Gaussian distribution:

$$Cell_i = p_{\theta(i)} \mathscr{G}(s_i | \mu_{\theta(i)}, var_{\theta(i)}) \quad (2)$$

where $\mathscr{G}(\cdot)$ indicates a Gaussian distribution. $Cell_i$ is weighted using the point-occupying probability. Similar to [1], the maximization problem in (1) can be solved via gradient descent methods with the initial pose estimation, or simply through an exhaustive search.

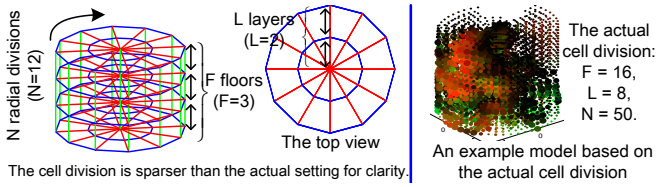

Fig. 3. Illustration of cell division in the cylinder. The central cells are denser, as we assume the object is located in the center, and thus the central cells are more reliable for object representation. Please see texts and Fig. 10 for detailed explanations of the model.

Model-based point labeling: Given the object samples collected by the trained models, we can further apply model-based object segmentation to the collected samples, thereby achieving point labeling.

Let M denote the model set. We collect object samples using these models via a brute-force search in the environment. Given a threshold τ, each sample S satisfying $\max_{m \in M} match(S,m) > \tau$ is collected. For the segmentation, let C_i be a cell in sample S. If the local matching value of C_i—$Cell_i$—is less than $1/3$, C_i is removed from S.

B. Incremental sample collection and evaluation

According to Algorithm 1, the initial models are generated using the object seeds. We then iteratively use the current models to collect new samples, and use the new samples to incrementally train the models.

Sample collection: New samples are brute-force searched and collected from the 3D environment using the current models $m \in M$. Each model inserts three of its best matched samples into the sample set, and a threshold $\max_{m \in M} match(S,m) < \varepsilon$ is set as the stopping condition for the sample collection of model m. Actually, we can use the initial models to prune the search range to not-so-badly matched samples during preprocessing, thus greatly reducing computation of sample collection in further iterations.

Sample evaluation: Not all of the collected samples have equally *reliable* shapes. Unreliable samples either do not contain the target object in their center, or have strange, largely occluded, or mixed shapes. We use the current models to estimate the reliability of each sample. As different models indicate different sub-categories, each sample only needs to be well matched to its target sub-category. The reliability of sample S can therefore be formulated as

$$\mathscr{R}(S) = \max_{m \in M} match(S,m) \quad (3)$$

.

C. Model learning

Sample collection and model learning are operated alternatively, as shown in Algorithm 1. Given the new samples collected by the current models, model learning has two steps, *i.e.* candidate model generation and model selection. In the first step, new candidate models are incrementally generated by combining the current models and new samples.

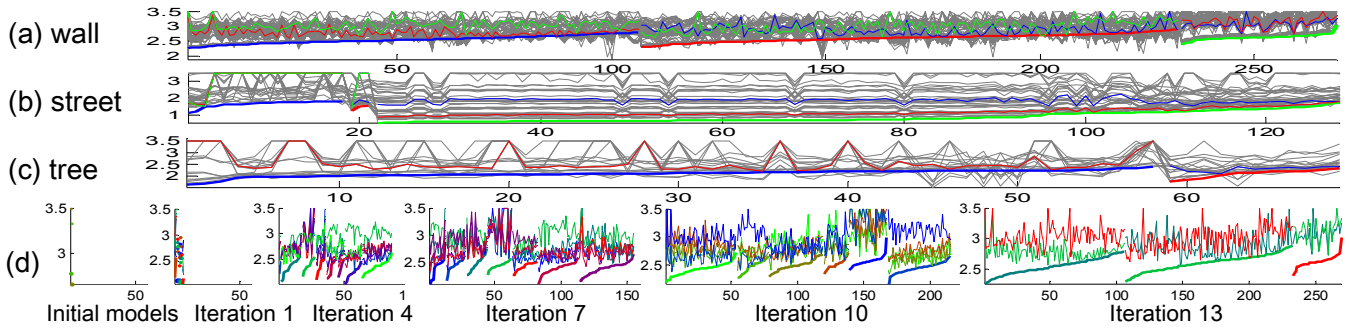

Fig. 4. Model selection results in the final iteration (a–c) and evolution of the wall models (d). The horizontal axis shows different samples, and the vertical axis shows the matching uncertainty as defined in (8a). Different curves show the matching uncertainty produced by different candidate models. The colored curves are for the selected models. The thick lines indicate the best-matched models of the samples after model selection. The samples are sorted to ensure a monotonic increase in the thick lines for clarity. The selected models an approximate minimization of matching uncertainty.

This incremental model generation ensures that the new model number is independent of the growth of the sample number. In the second step, the minimum description length (MDL) principle is used to select a subset of these candidate models that best describe the sub-categories of the samples. In this way, models of new sub-categories may be discovered, and current sub-categories may be assigned better models. The model selection results in the final learning iteration are illustrated in Fig. 4(a–c), and the evolution of the wall models after different iterations is shown in (d).

1) **Candidate model generation:** The candidate model set consists of 1) current models, 2) models generated through the pure breeding of new samples, and 3) models generated by the hybridization between new samples and current models.

Pure breeding of a new sample: Pure breading is used to produce a new model directly from a sample S. For cell i in this model, let its (spatially) nearest *not-none* cell in sample S be cell j of feature s_j. The point-occupying probability is assumed to follow the Gaussian distribution *w.r.t.* its distance $dist_{ij}$, and the local feature distribution is initialized with a constant variation as

$$\mu_i = s_j,\ var_i = \nu,\ p_i = \mathscr{G}(dist_{ij}|\mu' = 0, \delta' = \eta R_j) \tag{4}$$

The local feature variance ν can be either pre-defined or estimated from a seed. R_j denotes the distance between the sample center and cell j.

Hybridization between a new sample and model: This operation first generates a pure-breeding model m^s from sample S, and then merges m^s and the current model $m \in M$ into a new model m^{new}. Let the current model m be generated on the basis of Γ samples. Thus, m^{new} merged by m and m^s is a $(\Gamma+1)$-sample model. Let m and S be matched with horizontal rotation angle θ, and let cell i in m_s correspond to cell $\theta(i)$ in m. μ_i^{new}, var_i^{new}, and p_i^{new} in m^{new} are computed as

$$p_i^{new} = \frac{p_i^s + p_{\theta(i)}\Gamma}{\Gamma+1} \tag{5a}$$

$$\mu_i^{new} = \begin{cases} \frac{\mu_{\theta(i)}p_{\theta(i)}\Gamma + \mu_i^s p_i^s}{\Gamma p_{\theta(i)} + p_i^s} & p_{\theta(i)} \neq 0, p_i^s \neq 0 \\ \mu_{\theta(i)} & p_{\theta(i)} \neq 0, p_i^s = 0 \\ \mu_i^s & p_{\theta(i)} = 0, p_i^s \neq 0 \\ none & p_{\theta(i)} = 0, p_i^s = 0 \end{cases} \tag{5b}$$

$$var_i^{new} = \begin{cases} \max\{\frac{((\mu_{\theta(i)} - \mu_i^{new})^2 + var_{\theta(i)})\Gamma p_{\theta(i)}}{\Gamma p_{\theta(i)} + p_i^s} \\ \quad + \frac{(\mu_i^s - \mu_i^{new})^2 p_i^s}{\Gamma p_{\theta(i)} + p_i^s}, 0.1\nu\} & p_{\theta(i)} \neq 0, p_i^s \neq 0 \\ var_{\theta(i)} & p_{\theta(i)} \neq 0, p_i^s = 0 \\ var_i^s & p_{\theta(i)} = 0, p_i^s \neq 0 \\ none & p_{\theta(i)} = 0, p_i^s = 0 \end{cases} \tag{5c}$$

where the minimum var_i^{new} is set to 0.1ν to avoid an over-fitting between the model and samples.

2) **MDL-based model selection:** We use the MDL principle [35] to select a model subset M from the candidate model set $\mathscr{M}$ in order to describe the different sub-categories.

$$\underset{M \subseteq \mathscr{M}}{\operatorname{argmin}} \mathscr{L}(S,M), \quad \mathscr{L}(S,M) = \mathscr{L}(\mathscr{S}|M) + \mathscr{L}(M) \tag{6}$$

$$\mathscr{L}(M) = -\sum_{m \in M} p_m \log p_m, \quad \mathscr{L}(\mathscr{S}|M) = -\sum_{m \in M} p_m U_m \tag{7}$$

where the total description length $\mathscr{L}(S,M)$ consists of the inter-model description length $\mathscr{L}(M)$ and the intra-model sample variation given the models $\mathscr{L}(\mathscr{S}|M)$. p_m is the probability of a sample being best-matched to model m among all models in M. U_m is the average matching uncertainty between model m and its best-matched samples. Let $u(S,m)$ denote the matching uncertainty between sample S and model m. The calculation of U_m is weighted by the sample reliability $\mathscr{R}(S)$:

$$u(S,m) = -\log\ match(S,m) \tag{8a}$$

$$U_m = \sum_{\Phi(S)=m} \mathscr{R}(S)u(S,m) \Big/ \sum_{\Phi(S)=m} \mathscr{R}(S) \tag{8b}$$

$$\text{where,} \quad \Phi(S) = \underset{m \in M}{\operatorname{argmax}}\ match(S,m)$$

We use the greedy strategy to find an approximate solution to the minimization problem, as illustrated in Fig. 4. In each step, we remove the model from the current model set M, which minimizes the total description length, as follows:

$$\underset{m \in M}{\operatorname{argmin}} \mathscr{L}(\mathscr{S}, M \setminus \{m\}) \tag{9}$$

Encouragement of sub-category diversity: In the early iterations, the models are seed-like, and thus prone to collecting seed-like samples. Therefore, seed-like samples make up a larger proportion than they should in early iterations. If we

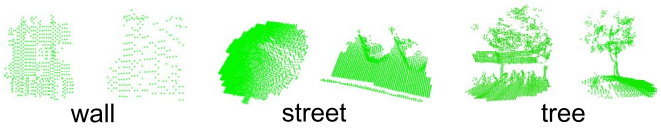

Fig. 5. Object seeds

estimate p_m on the basis of the collected samples, the large p_m of seed-like models may prevent the selection of models of other shape styles. As a result, the learning process will be biased toward seed-like sub-categories. Hence, we set p_m to a constant c for all models to ensure sub-category diversity. Thus, based on (6) and (7), we get $\mathscr{L}(M) = -\sum_{m\in M} c\log c$ and

$$\underset{M\subseteq\mathscr{M}}{\operatorname{argmin}}\mathscr{L}(\mathscr{S},M) = \underset{M\subseteq\mathscr{M}}{\operatorname{argmin}}\{-\lambda\|M\| - \sum_{m\in M} U_m\}, \qquad (10)$$

where $\lambda = \log c$ is a constant.

IV. Experiments

A. Intelligent vehicle and 3D point cloud

We develop an intelligent vehicle system to collect 3D data. The vehicle is equipped with five single-row laser scanners to profile its surroundings in different directions, as shown in Fig. 1. A global positioning system (GPS) and an inertial measurement unit (IMU) are mounted onto the vehicle. A localization module is developed by fusing the GPS/IMU navigation unit with a horizontal laser scanner. In this module, the localization problem is formulated as a simultaneous localization and mapping system with moving object detection and tracking [24], thereby ensuring both the global and local accuracy of the vehicle's pose estimation. A 3D representation of the environment can be obtained by geo-referencing the local range data from four slant laser scanners in a global coordinate system, given the vehicle's pose and the geometric parameters of the slanted laser scanners.

Our intelligent vehicle collects 3D point cloud data in an urban environment. Unlike RGB-D image data, large-scale 3D point cloud data do not contain color information. The proposed category model mining requires that each category contains enough objects to form a shape pattern, and thus the environment must be quite large and contain various objects. The size of the entire point cloud in our experiment is $300m \times 400m$. We choose the wall, street, and tree as the three target categories in our experiments. Within the environment, a large number of cars park on the street sides. The smooth street has a small variation in shape, whereas the wall has a larger shape variation. The wall has a high noise level owing to its long distance from the intelligent vehicle. The wall also has various shapes, as it may be occluded by tree branches. Trees have the largest shape variation, due to their varied structures.

B. Model learning

Object samples for the wall, street, and tree are randomly selected from the environment as seeds (as shown in Fig. 5). We set the minimum matching values for sample collection as $\varepsilon = 0.1$, and the model probability for model selection as $\lambda = 0.03$ for the *street* and *tree* categories; we set lower values for parameters $\varepsilon = 0.001$ and $\lambda = 0.01$ for the *wall* category, as the wall is far from the vehicle. Occlusion and measurement noise thus cause large structure variations of the wall category.

C. Results and evaluation

Fig. 10 shows the trained models and collected samples. All collected samples and the division of their sub-categories are shown in Fig. 4. The three wall models represent three sub-categories: a normal shape (*Wall 1*), noise shape (*Wall 2*), and incomplete shape (*Wall 3*). The two tree models represent isolated trees (*Tree 1*) and Δ-shaped trees (*Tree 2*). The street with cars to the side is represented by *Street 1*, and the flat street is represented by *Street 3*. The model for *Street 2* appears somewhat ambiguous, being between two other street sub-categories. Only three samples are described as *Street 2* in the sample set (see the thick red line in Fig. 4), and there appears to be a strange shape pattern, *i.e. a fence in the middle of the street*, in this environment (Fig. 10). Therefore, we consider *Street 2* as a biased model, which is typically an incorrect model in semi-supervised learning.

We use point labeling to evaluate the learned models. We compare the proposed method with pure seed-based point labeling, conventional unsupervised methods, and the widely-used supervised method of point cloud classification, which is based on AMNs. Each competing method is typical in 3D point labeling in a large urban environment, although none are as close to the task of efficient model base construction as our method. We therefore only compare them from the perspective of point labeling. Besides, as point cloud data do not have color information, many RGBD-image-based methods [2], [4], [5], [7], [8], [9] are not comparable.

To construct the ground truth, we manually label 3D points as *wall*, *flat street*, *tree* and *car*. Points labeled *car* and *flat street* are considered as positives in the street detection. Both the tree seeds and tree models contain a small area of flat streets, so the *flat street* points are not considered as either positives or negatives in the tree detection. Different segmentation thresholds (τ) are used to draw the performance curves for point labeling.

1) **Comparison with seed-based point labeling:** Seed-based point labeling directly generates the initial models from seeds, and uses them to retrieve objects in the environment. Fig. 6 compares model- and seed-based point labeling. The results demonstrate the accuracy of the learned models. Point labeling uses the global structure of objects, so small object fragments from large occlusions are not prone to being detected, although they are labeled as different categories in the ground truth. As a result, the curve does not continuously increase toward a recall of 100%.

2) **Comparison with conventional unsupervised methods:** The first unsupervised method for comparison is unsupervised 3D category discovery and the point labeling proposed in [1]. Considering the requirements for the inputs and outputs in Section II, this method is the most comparable to our own. Along with [1], we also compare our method

TABLE I

COMPARISON OF THE TIME COST

	Unsupervised 3D category discovery [1]	Unsupervised repetitive shape extraction [20], [21], [1]	Ours approach: Semi-supervised		
			Wall	Street	Tree
Time (s)	6280	3366	200	47	52

with unsupervised repetitive shape extraction. This method was originally proposed by [20], [21], and was applied to indoor environment. Zhang *et al.* extended the core idea of this method to a large urban environment, as one of its competing methods in the experiments of [1]. Thus, we choose this extended version of repetitive shape extraction for comparison. Note that repetitive shape extraction is mainly based on the hierarchical clustering of object samples (see [1] for details), and as is widely known, the cluster number (or cluster size) greatly affects the cluster purity. Fortunately, one of the final cluster-merging steps produces two clusters in the wall, street, and tree categories, as well as one chaotic cluster (shown in Fig. 7), which is similar to the final number of sub-categories in our method. Thus, we use these clusters for comparison.

Result: The results of the two unsupervised competing methods are shown in Fig. 6. The unsupervised 3D category discovery [1] shows relatively low error rates, while unsupervised repetitive shape extraction [20], [21], [1] exhibits relatively high detection rates.

Without shape guidance from object seeds or any mechanism to limit the bias problem, unsupervised repetitive shape extraction provides biased results, *i.e.* object samples that are not correctly localized (see Fig. 7). One category consists of wall fragments, and another category takes a combination of two trees as a single object. Actually, the chaotic cluster is a wall category, but is greatly biased. Its proportion of *wall*, *flat street*, *tree*, and *car* points is $1 : 0.4 : 0.6 : 0.2$.

Time cost: Table. I shows the time costs of the two unsupervised methods and our approach. Compared to the unsupervised methods, the biggest advantage of our semi-supervised approach is its low computational cost, as it only deals with some semantically meaningful structure patterns belonging to certain target categories, rather than all possible (probably meaningless) repetitive patterns in a large environment. The high efficiency is important for knowledge mining from "big point cloud data".

3) **Comparison with AMN-based classification:** AMNs [14] demonstrated a superior performance in the multiple-class classification of point clouds in recent years. Although not designed for category model mining, and despite their requirement for the manual labeling of a large amount of training data, we compare AMNs with our system from the perspective of point labeling. AMNs are trained to classify the *wall*, *flat street*, *tree*, *car*, and *unlabeled* categories. The *unlabeled* category mainly consists of small fragments resulting from data sparsity and other objects, such as buses. They are unclear at the object level, and thus are not used in the previous evaluation. The evaluation follows the same criteria above: *car* and *flat street* are considered as positives in the street detection, and *flat street* points are not used in the tree detection.

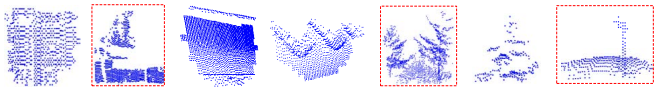

Fig. 7. Central samples of the clusters learned by unsupervised repetitive shape extraction [20], [21], [1]. The object samples describe, from left to right, the shape patterns of two wall categories, two street categories, two tree categories, and one chaotic category. The red rectangles indicate biased categories.

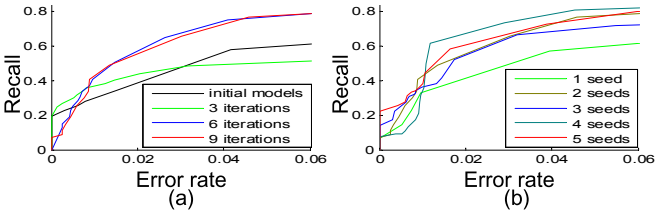

Fig. 8. Comparison of *tree* models (a) after different numbers of iterations, and (b) with different seed numbers. The curves with more than four seeds converge, showing that a limited number of seeds are sufficient for model learning, as seed bias is overcome through sample collection.

The max-margin strategy allows the AMN to operate as a powerful multiple-class classifier, but the algorithm does not use the global shapes of objects efficiently. In some cases, local features are discriminative enough for classification, thus leading to relatively low recalls and high error rates. Fig. 6 shows the classification results with different numbers of training samples.

4) **Other evaluations:** We evaluate the models trained after a different number of iterations, or with different seed numbers. The *tree* category is selected for the tests. We do not manually remove biased models from the results to avoid bringing subjective judgments into the evaluation; in fact, there are no extremely biased models in the *tree* results.

Performance after a different number of iterations: Fig. 8(a) shows a performance comparison. The initial models are directly generated from seeds. In early iterations, the system tends to collect seed-like samples, which enlarges the bias problem in the initial models. Thus, both the initial models and those after three iterations are highly accurate for seed-like objects (the recall is relatively high whereas the error rate is low), but they have a low accuracy for non-seed-like objects (their recalls converge at a low value). However, our MDL-based model selection ensures model diversity (Section. III-C.2) and enables the evolution from seed-like to well developed models. As a result, the curves after six iterations converge, showing a similarly good performance.

Performance with different seed numbers: Our algorithm does not require a large number of seeds for high accuracy, as the seeds are just the starting points of the learning process. The models are mainly trained using automatically collected samples. As shown in Fig. 8(b), except the 1-seed curve, the other four curves converge, showing that a limited number of seeds are sufficient for model learning.

V. CONCLUSION AND DISCUSSION

This paper proposed a semi-supervised approach for training object models in a large and complex 3D environment,

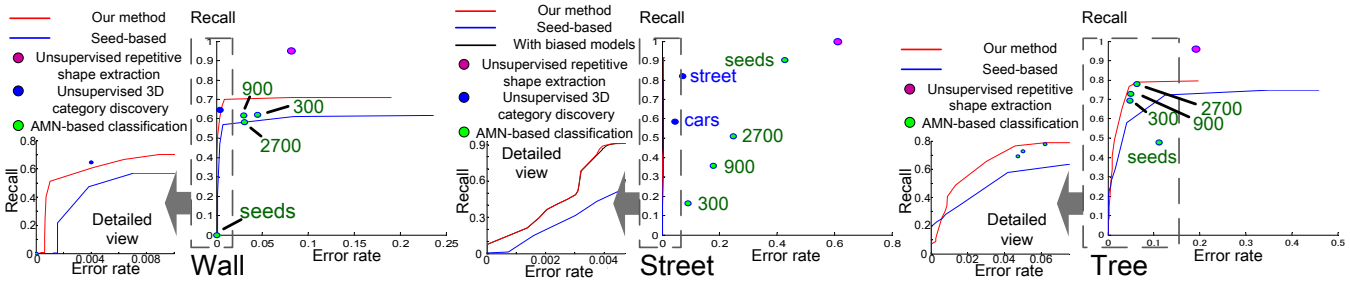

Fig. 6. Comparison of competing methods, including seed-based point labeling, AMN-based point cloud classification [14], unsupervised 3D category discovery [1] and unsupervised repetitive shape extraction [20], [21], [1]. Performance curves based on both our learned models and seeds are shown. The biased model—*Street 2*—is removed and thus not used in our street detection; in addition, the black curve illustrates the performance without subtracting the biased models. Recalls and error rates of the unsupervised 3D category discovery [1] are indicated by the blue dots. As [1] regards the "flat street" and "the cars on the street sides" as two individual categories, we use two blue dots to show their performance for the street category. Besides, the evaluation criterion for the tree category in [1] is different from ours, so its tree category performance is not shown. The purple and green dots indicate the unsupervised repetitive shape extraction [20], [21], [1] and AMN-based multiple-class classification [14], respectively. In the learning of the AMNs, We label the seeds, and then further randomly label 300, 900, and 2700 point cloud cliques as different settings of the training samples (see the text next to the green dots).

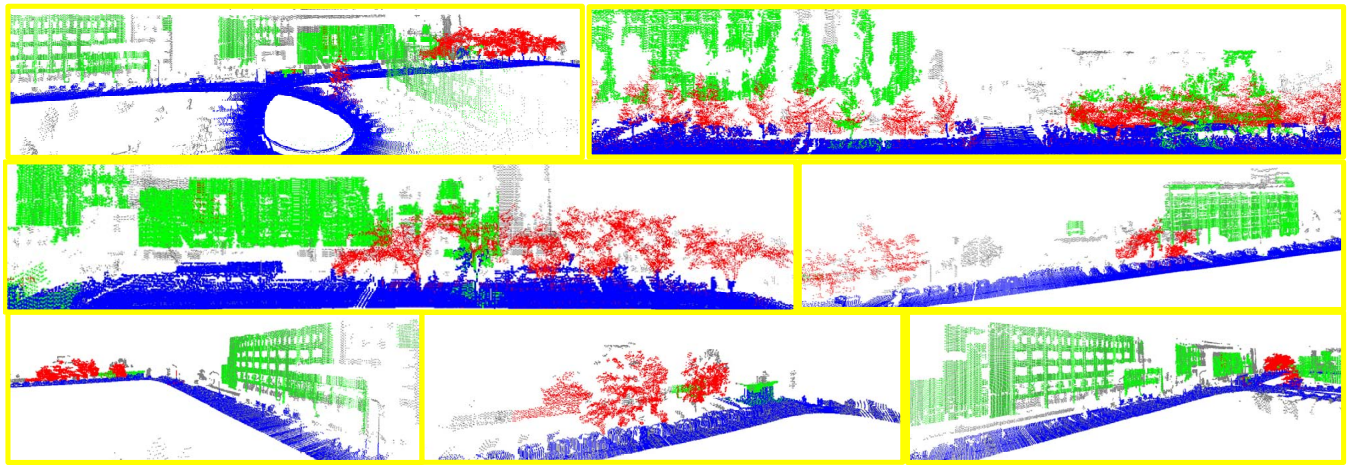

Fig. 9. Model-based point labeling results. Different colors indicate different categories, *i.e.* wall (green), tree (red), and street (blue).

and described various experiments that demonstrated its effectiveness and high efficiency.

The proposed approach is a plausible way for model mining from "big point cloud data", as it requires much less human labeling than supervised methods and exhibits less computational cost than unsupervised methods.

Color information is not used, proving that the mining of category models can be performed using 3D-point-cloud data alone. In our experiment, the biased street model did not greatly reduce the point labeling accuracy of the entire street category (Fig. 6), as it only detected a limited number of strange shape patterns. We can consider the biased model as an abnormal detection from the environment.

ACKNOWLEDGMENT

This work was supported by Microsoft Research, a Grant-in-Aid for Young Scientists (23700192) of Japans Ministry of Education, Culture, Sports, Science, and Technology (MEXT), and Grant of Japans Ministry of Land, Infrastructure, Transport and Tourism (MLIT).

REFERENCES

[1] Q. Zhang, X. Song, X. Shao, H. Zhao, and R. Shibasaki, "Unsupervised 3D Category Discovery and Point Labeling from a Large Urban Environment", In *ICRA*, 2013.

[2] A. Anand, H. S. Koppula, T. Joachims, and A. Saxena "Contextually guided semantic labeling and search for three-dimensional point clouds", In *IJRR* vol. 32, no. 1, 19–34, 2013.

[3] A. Aldoma, F. Tombari, L. D. Stefano, and M. Vincze, "A global hypotheses verification method for 3D object recognition", *In* ECCV, 2012.

[4] K. Lai, L. Bo, X. Ren, and D. Fox, "Detection-based object labeling in 3D scenes", *In* ICRA, 2012.

[5] K. Lai, L. Bo, X. Ren, and D. Fox, "Sparse distance learning for object recognition combining rgb and depth information", *In* ICRA, 2011.

[6] W. Susanto, M. Rohrbach, and B. Schiele, "3D object detection with multiple kinects", *In* ECCV, 2012.

[7] X. Ren, L. Bo, and D. Fox, "Rgb-(d) scene labeling: Features and algorithms", *In* CVPR, 2012.

[8] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images", *In* ECCV, 2012.

[9] H. S. Koppula, A. Anand, T. Joachims, and A. Saxena, "Semantic labeling of 3D point clouds for indoor scenes", *In* NIPS, 2011.

[10] A. Collet, S. S. Srinivasay, and M. Hebert, "Structure discovery in multi-modal data: a region-based approach", In *ICRA*, 2011.

[11] E. Herbst, P. Henry, X. Ren, and D. Fox, "Toward object discovery and modeling via 3D scene comparison", In *ICRA*, 2011.

[12] E. Herbst, X. Ren, and D. Fox, "Rgb-d object discovery via multi-scene analysis", *In* IROS, 2011.

[13] K. Lai, L. Bo, X. Ren, and D. Fox A Large-Scale Hierarchical Multi-View RGB-D Object Dataset. In ICRA, 2011.

[14] D. Munoz, J. A. Bagnell, N. Vandapel, and M. Hebert, "Contextual classification with functional max-margin Markov networks", *In* CVPR, 2009.

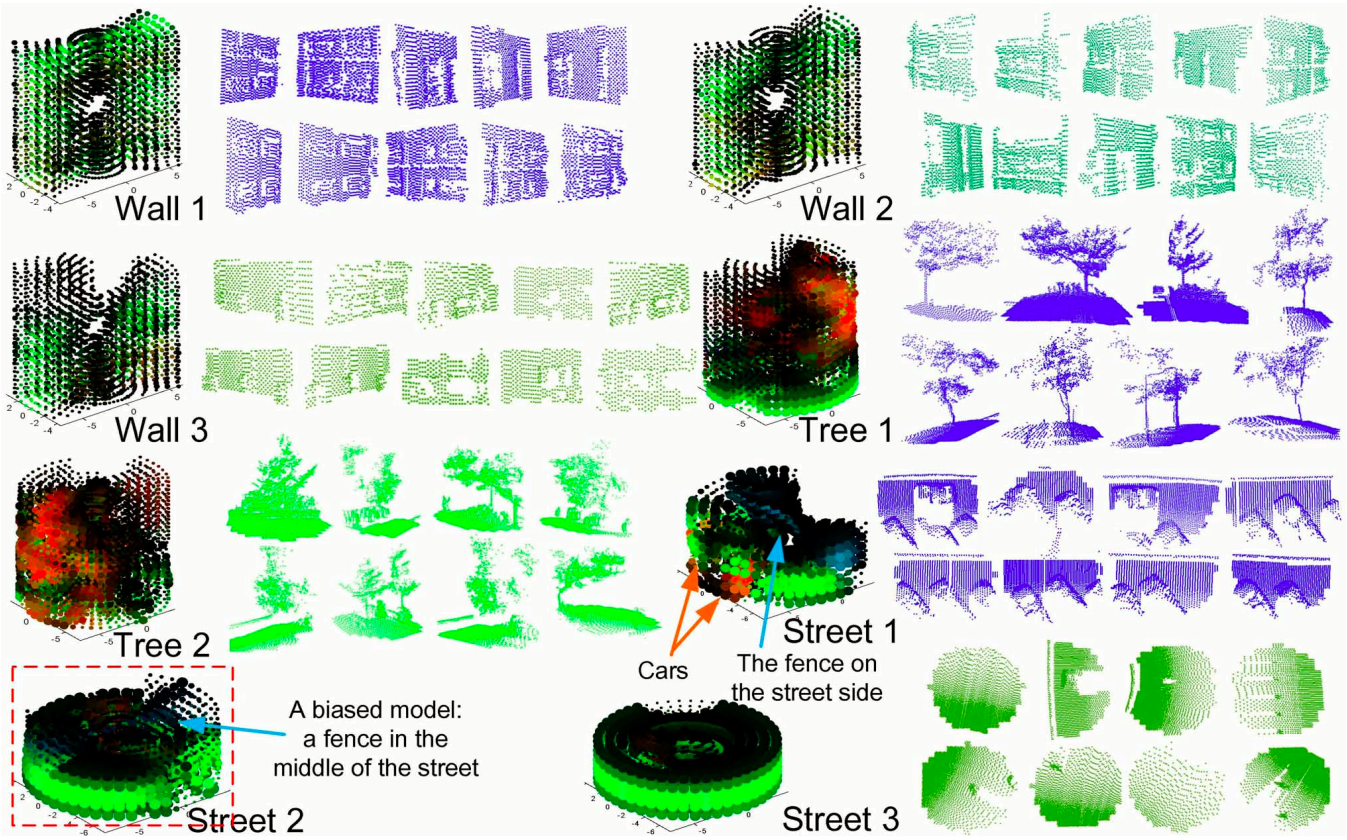

Fig. 10. Models and their corresponding samples. Each circle in the model represents a *not-none* cell, showing its point-occupying probability (intensity), mean value ([low] blue→green→red [high]) and variance (the reciprocal of the circle area) of the local feature.

[15] D. Anguelov, B. Taskary, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng, "Discriminative learning of Markov random fields for segmentation of 3D scan data", *In* CVPR, 2005.

[16] L.-J. Li, G. Wang, and Li F.-F.. OPTIMOL: automatic Online Picture collecTion via Incremental MOdel Learning. In *IJCV*, vol. 88, no. 2, pp. 147–154, 2010.

[17] F. Endres, C. Plagemann, C. Stachniss, and W. Burgard, "Scene analysis using latent dirichlet allocation", In *RSS*, 2009.

[18] R. Detry, N. Pugeault, and J. H. Piater, "A probabilistic framework for 3D visual object representation", In *PAMI*, vol. 31, no. 10, pp. 1790–1803, 2009.

[19] M. Ruhnke, B. Steder, G. Grisetti, and W. Burgard, "Unsupervised learning of compact 3D models based on the detection of recurrent structures", In *IROS*, 2010.

[20] J. Shin, R. Triebel, and R. Siegwart, "Unsupervised discovery of repetitive objects", In *ICRA*, 2010.

[21] M. Ruhnke, B. Steder, G. Grisetti, and W. Burgard, "Unsupervised learning of 3D object models from partial views", In *ICRA*, 2009.

[22] G. Somanath, R. MV, D. Metaxas, and C. Kambhamettu, "D-clutter: Building object model library from unsupervised segmentation of cluttered scenes", In *CVPR*, 2009.

[23] J.-F. Lalonde, N. Vandapel, D. F. Huber, and M. Hebert, "Natural terrain classification using three-dimensional ladar data for ground robot mobility", In *JFR*, vol. 23, no. 1, pp. 839–861, 2006.

[24] H. Zhao, M. Chiba, R. Shibasaki, X. Shao, J. Cui, and H. Zha, "Slam in a dynamic large outdoor environment using a laser scanner", In *ICRA*, 2008.

[25] D. Munoz, N. Vandapel, and M. Hebert. Onboard Contextual Classification of 3-D Point Clouds with Learned High-order Markov Random Fields. In *ICRA*, 2009.

[26] R. Triebel, K. Kersting, and W. Burgard. Robust 3D Scan Point Classification using Associative Markov Networks. In *ICRA*, 2006.

[27] H. Zhao, Y. Liu, X. Zhu, Y. Zhao, and H. Zha. Scene Understanding in a Large Dynamic Environment through a Laser-based Sensing. In *ICRA*, 2010.

[28] A. Golovinskiy, V. G. Kim, and T. Funkhouser. Shape-based Recognition of 3D Point Clouds in Urban Environments. In *ICCV*, 2009.

[29] C. Wojek, S. Roth, K. Schindler, and B. Schiele. Monocular 3D Scene Modeling and Inference: Understanding Multi-Object Traffic Scenes. In *ECCV*, 2010.

[30] I. Posner, M. Cummins, and P. Newman. A Generative Framework for Fast Urban Labeling Using Spatial And Temporal Context. In *Autonomous Robots*, vol. 26, no. 2, pp. 153–170, 2009.

[31] K. Klasing, D. Wollherr, and M. Buss. Realtime segmentation of range data using continuous nearest neighbors. In *ICRA*, 2009.

[32] D. Anguelov, B. Taskary, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng. Discriminative learning of Markov random fields for segmentation of 3D scan data. In *CVPR*, 2005.

[33] D. Munoz, J. A. Bagnell, N. Vandapel, and M. Hebert. Contextual Classification with Functional Max-Margin Markov Networks. In *CVPR*, 2009.

[34] K. Klasing, D. Wollherr, and M. Buss. A Clustering Method for Efficient Segmentation of 3D Laser Data. In *ICRA*, 2008.

[35] M. H. Hansen and B. Yu, "Model selection and the principle of minimum description length", In *JASA*, vol. 96, no. 454, pp. 746–774, 2001.

[36] R. Triebel, R. Paul, D. Rus, and P. M. Newman, "Parsing outdoor scenes from streamed 3D laser data using online clustering and incremental belief updates", In *AAAI*, 2012.

[37] R. Paul, R. Triebel, D. Rus, and P. M. Newman, "Semantic categorization of outdoor scenes with uncertainty estimates using multi-class Gaussian process classification", In *IROS*, 2012.

[38] Q. Zhang, X. Song, X. Shao, H. Zhao, and R. Shibasaki, "Category Modeling from just a Single Labeling: Use Depth Information to Guide the Learning of 2D Models", In *CVPR*, 2013

[39] Q. Zhang, X. Song, X. Shao, H. Zhao, and R. Shibasaki, "Learning Graph Matching for Category Modeling from Cluttered Scenes", In *ICCV*, 2013