Contextual Labeling 3D Point Clouds with Conditional Random Fields

Anh Nguyen and Bac Le

University of Science, Vietnam
{nqanh,lhbac}@fit.hcmus.edu.vn

Abstract. In this paper we present a new approach for labeling 3D point clouds. We use Conditional Random Fields (CRFs) as an objective function, with unary energy term assessing the consistency of points with labels, and pairwise energy term between points and its neighbors. We propose a new method to learn this function from a collection of trained labels using JointBoost classifier formalism. By using CRFs with different geometric and contextual features, we show that our method enables the combination of semantic relations and achieves higher accuracy. We validate and demonstrate the efficiency of our method on complex urban laser scans and compare it with several alternative approaches.

Keywords: 3D point cloud labeling; conditional random fields; JointBoost.

1 Introduction

With the development of scanning technologies, 3D point clouds are now widely available. Professional scanners such as Light Detection and Ranging (LIDAR) and Microsoft Kinect provide enormous amount of data need to be analyzed. In 2011, Point Cloud Library (PCL) [10] which contains state of the art algorithms for 3D understanding was introduced. As scanning technologies advance and the development of PCL, processing point clouds gains more and more attraction in computer vision and robotics community. 3D perception is an important problem and has many applications in autonomous systems such as intelligent vehicles, autonomous mapping and navigation.

3D point cloud labeling is a process that assigns a label to all objects in an observed scene. This is an important task because the its results could be helpful in many autonomous robot navigation tasks such as locating and recognizing objects, obstacle avoidance, and and environment modeling. Point cloud data contain useful geometric information which is ambiguous and unreliable if we reconstruct from 2D images or stereo images. However, labeling objects from point clouds is complicated and challenging because the point cloud data are noisy, unorganized and sparse. Moreover, the sampling density of point clouds is typically uneven due to varying linear and angular rates of the scanner. In addition, the surface shape can be arbitrary with sharp features and there is no statistical distribution pattern in the point cloud data [9].

N.T. Nguyen et al. (Eds.): ACIIDS 2014, Part I, LNAI 8397, pp. 581–590, 2014.
© Springer International Publishing Switzerland 2014

A labeling algorithm should always consider the trade off between accuracy and speed. Because the over or under segmentation when labeling is inevitable, it is helpful for an algorithm to accept additional intuitive parameters describing assumptions or can include contextual information. In general, there are two basic approaches for segmenting and labeling 3D point clouds:

Geometric Reasoning: This approach uses purely mathematical model and geometric reasoning techniques such as region growing or model fitting methods; and a robust estimators to fit linear and nonlinear models to point cloud data. This approach achieves good results in simple scenario and not time consuming. However, its limitations are it cannot deal with incomplete or uneven distribution data, difficult to choose the size of model when fitting objects, and not working well in complex scenes.

Machine Learning: This approach uses feature descriptors to extract 3D features from point cloud data, then machine learning techniques are used to build a classifier to learn different classes of objects. Afterwards, different points are labeled by applying this classifier. Points can be treated independently or contextually with their neighbors.

The machine learning techniques usually outperform techniques purely based on geometric reasoning. The reason is due to uneven density, occlusions in point cloud data, it is very difficult to find and fit complicated geometric primitives to objects. Although machine learning techniques give better results, they are usually slow and rely on the result of feature extraction step [9].

We propose a graph based method to simultaneous segmentation and labeling 3D point clouds. Our method works by extracting geometric and contextual features from the point clouds. We use CRFs as the combination of a set of unary terms, defined for each point individually, and a set of pairwise terms, defined as a function of neighbor points. Instead of solving the model as a discrete energy minimization task over a graph containing nodes corresponding to individual points, we use JointBoost [15] classifier formalism to learn CRFs.

The remainder of the paper is organized as follows: In the next section, we summarize related work on similar initiatives. We review the formulation of CRFs used for labeling 3D point clouds in section 3. Section 4 describes the learning problems and techniques using JointBoost classifier algorithm. Section 4 reports the experimental results followed by a discussion and future work suggestions.

2 Previous Work

Several methods have been proposed to segment and label 3D point clouds. In [9], authors provided a good survey for segmenting and labeling point cloud data. In this section, we only summarize previous works that use machine learning techniques.

Graphical models such as Markov Random Field (MRF) and Conditional Random Field (CRF) are commonly used in computer vision tasks (e.g. image denoising, optical flow, segmentation, etc). Many work on labeling point clouds field use these models. Rusu et al. [1] proposed an approach based on surface segmentation for labeling points with different geometric surface primitives using CRF and feature descriptor called Fast Point Feature Histograms (FPFH) [6]. By defining classes of 3D geometric surfaces, and making use of contextual information using CRF, this method successfully segment and label 3D points based on their surfaces even with noisy data. Golovinskiy [8] used k-nearest neighbours (KNN) to build a 3D graph on the point clouds. This method introduces a penalty function to encourage smooth segmentation where the foreground is weakly connected to the background.

The MRF framework provides a natural way of incorporating contextual information. Schoenberg et al. [3] used MRF to segment 3D point clouds achieved by the combination of an optical camera and a laser scanner. Texture of point clouds are generated from interpolating from laser range and aligned optical image. The weights of MRF model are computed as a fusion of Euclidean distances, pixel intensity differences and angles between surface normals estimated at each point. This method showed good results in urban environment but requires a complex camera system.

Shapovalov [7] introduced a cutting-plane training of Non-associative Markov Network for 3D point cloud segmentation. This method applied kernel trick as the non-linear method to train non-associative Markov networks in a principled manner using the structured Support Vector Machine (SVM) formalism. The work of Munoz et al. [4] use Max-Margin Markov Networks and adapt a functional gradient approach for learning high dimensional parameters in order to perform discrete, multi-label classification. This method showed how the model can be incorporated with robust potentials to preserve less dominant labels.

Anguelov et al. [5] used Associative Markov Networks (AMN) for segmentating 3D point clouds. The Associative Markov Networks encourage neighboring points to have same class labels. This method uses maximum-margin framework to train the model from a set of labeled scans. The energy is a sum of potential functions of the features corresponding to individual nodes extracted from the scan. The inference is performed using graph cut and the parameters of the energy are learned by reducing the training problem to quadratic programming.

Munoz [11] introduced an efficient learning algorithm of random field with higher-order cliques by using subgradient optimization. A context approximation is proposed to make the model usable on a mobile vehicle for environment modeling. This work is not time consuming and can be run onboard of a mobile robot.

Koppula [13] proposed a new learning algorithm for scene understanding that exploits rich relational information derived from 3D point cloud for object labeling. In particular, this work used graphical model that naturally captures the geometric relationships of a 3D scene. Each 3D segment is associated with a node, and pairwise potentials model the relationships between segments. The model is trained using a maximum-margin approach that globally minimizes an upper bound on the training loss.

3 CRFs for Labeling 3D Point Clouds

Generative graphical models represent a joint probability distribution $p(x, y)$, where x expresses the observations and y expresses the label. These approaches require to model the observations which lead to erroneous independence assumptions among features. A solution is to use discriminative models such as CRFs, which represent a conditional probability distribution $p(y|x)$. The distribution is defined by the dependencies of the random variables represented in an undirected graph where each vertex represents a random variable and the edges represent a dependency between two variables. In general, we can formulate a CRF model as:

$$p(y|x) = \frac{1}{Z(x)} \exp \psi(x, y) \tag{1}$$

where $Z(x)$ is the normalizing partition function.

In this work, we consider CRF as a pairwise potentials. Our model includes the unary term E_1 measures consistency between the unary features x_i, which includes descriptors such as curvatures, FPFH, etc, of point i and its label c_i; the pairwise term E_2 measure the consistency between adjacent point labels c_i and c_j. To avoid under or over segmentation which leads to false labeling, for each adjacent pair of points we define pairwise features y_{ij} to provide cues whether adjacent points should have the same label. Given C as a predefined set of labels, our goal is to label each point i with a label in C. This task is equivalent to minimize the following objective function:

$$E(c, \theta) = \sum_i E_1(c_i; x_i, \theta_1) + \sum_{i,j} E_2(c_i, c_j; y_{ij}, \theta_2) \tag{2}$$

where $\theta = (\theta_1, \theta_2)$ are the parameters of the model.

Our model is in contrast to MRF model, which defines a joint probability over the labels, from which the conditional may then be derived. For segmenting and labeling 3D point clouds, MRF model may have worse labeling performance, while CRF learning algorithms is optimized for labeling performance. Moreover, the pairwise term in CRF model expresses connection between adjacent objects, which is not true in MRF model. Our CRF model is similar to Kalogerakis [12] and Huang [14]. However, we use different 3D features for both the unary and pairwise potentials which are more suitable for labeling 3D point clouds.

3.1 Unary Energy Term

The unary classifier is the most important component of our system because it evaluates a classifier. The classifier takes the feature vector x as input, and returns a probability distribution of labels for that point: $P(c|x, \theta_1)$. Same as [14], we use JointBoost classifier [15] to learn this term. Then, the unary energy of a label c is equal to its negative log-probability:

$$E_1(c; x, \theta_1) = -\log P(c|x, \theta_1) \tag{3}$$

Features: We use geometric features as described in [11] [20] and FPFH descriptor to produce adequate 3D features. The estimation of a FPFH descriptor includes two steps. The first step is to compute the histogram of the three angles between a point i and its k-nearest neighbors to produce the Simplified Point Feature Histogram (SPFH). Then, for each point i, the values of the SPFH of its k neighbors are weight by their distance w to produce the FPFH:

$$FPFH(i) = SPFH(i) + \frac{1}{k}\sum_{i=1}^{k}\frac{SPFH(i)}{w_i} \tag{4}$$

Next, we describe the pairwise energy term in our model.

3.2 Pairwise Energy Term

The main role of the pairwise energy term is to prevent incompatible segments from being adjacent and take contextual information into account. In general, the pairwise term penalizes neighboring parts of point being assigned different labels:

$$E_2(c_i, c_j; y, \theta_2) = L(c_i, c_j)G(y_{i,j}) \tag{5}$$

where the term L enables the label compatibility, and term G describe a geometry dependent.

The label compatibility L measures the consistency between two adjacent labels. This term is represented as a matrix of penalties for each possible pair of labels, which allows different pairs of labels to incur different penalties. The geometry dependent term G measures the likelihood of there being a difference in labels. Similar to the unary energy term, this term also evaluates a JointBoost classifier as follow:

$$G(y_{i,j}) = -\lambda \log P(c_i, c_j | y_{i,j}, \theta_2) \tag{6}$$

where λ controls the contribution of the pairwise term. In implementation, we use Point Pair Feature (PPF) [2] as a pairwise feature. Given two points i_1 and i_2 and their normals n_1 and n_2, the PPF is given by:

$$PPF(i_1, i_2) = (|d|_2, \angle(n_1, d), \angle(n_2, d), \angle(n_1, n_2)) \tag{7}$$

where $\angle(a, b) \in [0, \pi]$ represents the angle between a and b and $d = i_2 - i_1$. A point pair (i_1, i_2) is aligned to a scene pair (s_1, s_2) that has the same feature vector.

3.3 JointBoost Classifier

JointBoost is a boosting algorithm that automatically performs feature selection process with a large numbers of input features for multiclass classification. This algorithm jointly trains multiple classifiers so that they share as many features as possible. The result is a classifier that runs faster and requires less data to train than original classifiers. In particular, the number of features required to reach a fixed level of performance grows sub-linearly with the number of classes, as opposed to the linear growth observed with original classifiers [15]. JointBoost has a fast sequential learning algorithm, and produces output probabilities suitable for combination with other terms in the CRF model.

The JointBoost classifier takes as input a feature vector z, and outputs a probability $P(c = l|z)$ for each possible class label $l \in C$, where C is the set of possible labels. In the unary energy term, a classifier computes the likelihood of a part label c given unary features x. In the pairwise energy term, a second JointBoost classifier is used to determine the likelihood whose adjacent points have different classes given pairwise features y. This model reduces generalization error for multiclass recognition when classes overlap in feature space. So, we believe that JointBoost is the best available classifier for this task.

Similar to Torralba [15], we use a decision stumps in our classifier. A decision stump is a simple classifier that scores each possible class label l. Given a feature vector z and its threshold z_f of f-th entry, a JointBoost decision stump can be written as:

$$h(z, l; \phi) = \begin{cases} \alpha & \text{if } z_f > \tau \text{ and } l \in C' \\ \beta & \text{if } z_f \leq \tau \text{ and } l \in C' \\ k_l & \text{if } l \notin C' \end{cases} \tag{8}$$

Each decision stump stores a set of classes C'. If $l \in C'$, then the stump compares z_f against a threshold τ. It returns a constant α if $z_f > \tau$, and β otherwise. If $l \notin C'$, then the comparison is ignored, and a constant k_l is returned instead. There is only one k_l for each $l \notin C'$. The parameters ϕ of a single decision stump are f, α, β, τ, the set C', and k_l for each $l \notin C'$.

The probability of a given class l is then computed by summing the decision stumps and then performing a softmax transformation:

$$H(z, l) = \sum_j h(z, l; \phi_j) \tag{9}$$

$$P(c = l|z, \xi) = \frac{\exp(H(z, l))}{\sum_{l \in C} \exp(H(z, l))} \tag{10}$$

The parameters ξ consist of the parameters ϕ_j of all the individual decision stumps.

4 Learning CRFs

A well known approach for learning CRFs is maximize the log-likelihood of $p(y|x)$ as described in Lim [16]. This nonlinear optimization problem is solved by applying the Broyden-Fletcher-Goldfarb-Shannon (BFGS) method. Unfortunately, computing the normalization $Z(x)$ is intractable and the computational cost is expensive. Instead of using BFGS, we use the work of Shotton et al. [17] by to apply these following steps. First, we randomly split the training data into an exemplar set and a validation set in a proportion of approximately 3:1. The JointBoost classifiers for the unary term and the pairwise term is learned from the exemplar set. Finally, the remaining CRF parameters are learned by iteratively optimizing segmentation performance on the validation set. The performance of our CRF model relies on the classification accuracy of the classifiers used to define both the unary and pairwise terms.

4.1 Learning JointBoost Classifiers

We learn the JointBoost classifiers as described in Torralba [15]. The input to the algorithm is a collection of N training pairs (z_i, c_i), where z_i is a feature vector and c_i is the corresponding class label. For the unary terms, the training pairs are the feature vectors and their labels (x_i, c_i) for all points in the exemplar set. For the pairwise terms, the training pairs are the pairwise feature vectors. Each training pair is assigned a per-class weight $w_{i,c}$.

JointBoost classifier minimizes the weighted multiclass exponential loss over the exemplar set by solving the following equation:

$$J = \sum_{i=1}^{M} \sum_{l=1}^{C} w_{i,c} \exp(-I(c_i, l) H(z_i, l)) \tag{11}$$

where $I(\delta, \delta')$ is an indicator function:

$$I(\delta, \delta') = \begin{cases} 1 & \text{if } \delta = \delta' \\ -1 & \text{otherwise} \end{cases} \tag{12}$$

We stores a set of weights $\tilde{w}_{ic}$ that are initialized to the weights $w_{i,c}$. Then, at each iteration, one decision stump is added to the classifier. The parameters ϕ_j of the stump at iteration j are computed to optimize the following weighted least squares problem at each iteration:

$$J_{wse}(\phi_j) = \sum_{l=1}^{C} \sum_{i=1}^{N} w_{i,l} (I(c_i, l) - h(z_i, l; \phi_j))^2 \tag{13}$$

Once the parameters ϕ_j are determined, the weights are updated as:

$$\tilde{w}_{i,c} \leftarrow \tilde{w}_{i,c} \exp(-I(c_i, l) h(z_i, l; \phi_j)) \tag{14}$$

and the algorithm continues with the next decision stump.

4.2 Learning the Remaining Parameters

Once the JointBoost classifiers have been learned, our model learn the remaining parameters of the pairwise term by cross validation. Specifically, for any particular setting of these parameters, we can apply the CRF to all of the validation points, and evaluate the classification results.

The parameter λ controls the contribution of the pairwise term. We learn this parameter using the validation data set by trying different λ and picking the λ with the smallest testing error. Other parameters are optimized in two steps. Firstly, the segmentation error is minimized over a coarse grid in parameter space by brute-force search. Then, starting from the minimal point in the grid, optimization continues by using preconditioned conjugate gradient with numerically estimated gradients.

5 Experimental Results

We evaluate our model on the VMR-Oakland [18] point cloud dataset. This dataset contains labeled point cloud data collected from a moving platform around Carnegie Mellon University campus. The points were collected using laser scanner and are saved in text format, three real valued coordinates of each point are written in each line on airborne and terrestrial laser scans. Seven classes are used: *wire, pole, ground, vegetation, trunk, building*, and *car*. Examples of classified scenes from this dataset are shown in Fig. 1.

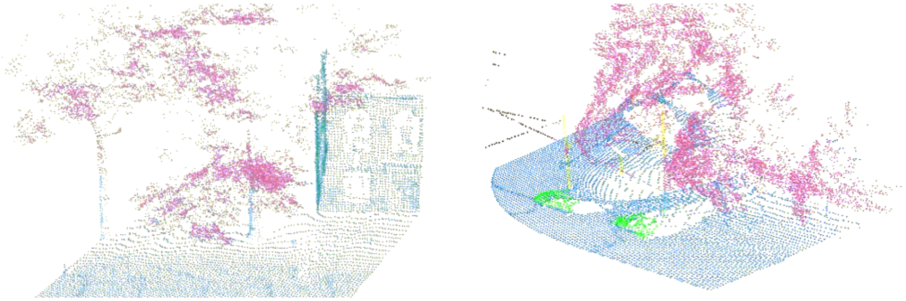
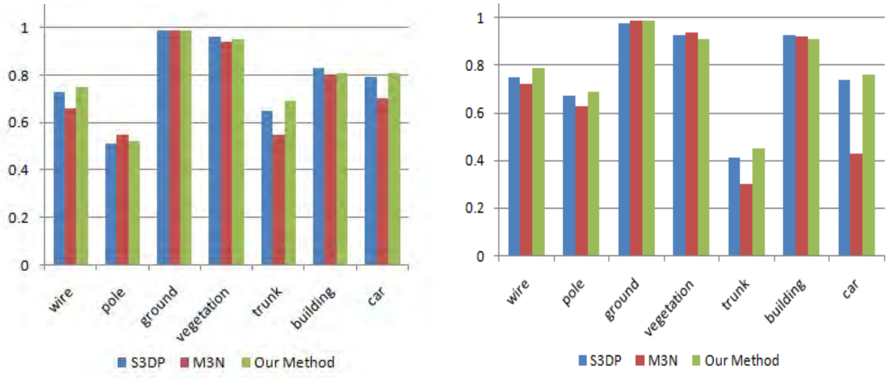


Fig. 1. Example classification results from VMR-Oakland dataset

In our experiments, the neighborhood of points is defined within a $1.0m$ radius. We use a fixed radius support region to compute following features: spectral and directional features to capture the local orientation, distribution of heights and related features [20], spin images [19] around z axis. We use spin images with 3×3 in the bottom level and 10×10 in the top level, with each cell being $0.3m \times 0.3m$. We run unexponentiated variant of the algorithm during 100 iterations and estimate the parameters on a validation set. All computations were performed on a Core 2 Duo @ 2 GHz CPU with 3 GB memory.

We compare our method with the Stacked 3D Parsing (S3DP) algorithm of Xiong at al. [18] and the linear, associative Max-Margin Markov Network (M3N) model of Munoz et al. [4]. Overall, methods that take contextual information into



(a) Precisions

(b) Recalls

Fig. 2. Precisions and Recalls for seven classes in VMR-Oakland dataset

account such as our method and S3DP outperform methods that use only local features. A complete comparison is shown in Fig.2. Some objects are easier to label: ground, vegetation, building are often isolated and have a lot of neighbor points so our algorithm performs well. Other objects, such as pole and trunk, are only have weak neighbor connection or often close to background clutter, so the precision and recall remain low.

6 Conclusion

We propose a new method for semantic labeling 3D point clouds. We consider CRF as a pairwise potentials, with the unary term measures consistency between features and the pairwise term between neighbor points. The JointBoost classifier is used to learn unary and pairwise terms. This approach reduces the size of feature vectors while still maintains the accuracy and encodes neighboring information. We include various features and contextual relations in our model. The experiments show that our approach achieves good performance when compared with state-of-the-art methods in complex urban scenes.

The main limitation of our approach is the need for labeled training data. As future work, we plan investigate this problem. Another way to improve our method is to add new features. Additional features such as geometric features or symmetry based features should significantly improve the results.

Acknowledgments. This work was supported by The National Foundation for Science and Technology Development, and Computer Science Department, University of Science, Vietnam. Author wishes to thank the authors who kindly provided the copies of their papers as references.

References

1. Rusu, R.B., Holzbach, B., Blodow, N., Beetz, M.: Fast Geometric Point Labeling using Conditional Random Fields. In: Proceedings of IROS, USA (2009)
2. Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model globally, match locally: Efficient and robust 3d object recognition. In: Proceedings of CVPR, San Francisco, CA, USA (2010)
3. Schoenberg, J., Nathan, A., Campbell, M.: Segmentation of dense range information in complex urban scenes. In: Proc. of IROS, Taipei (2010)
4. Munoz, D., Bagnell, J., Vandapel, N., Hebert, M.: Contextual classification with functional Max-Margin Markov Networks. In: Proc. of CVPR, USA (2009)
5. Anguelov, D., Taskar, B., Chatalbashev, V., Koller, D., Gupta, D., Heitz, G., Andrew, Y.: Discriminative Learning of Markov Random Fields for Segmentation of 3D Range Data. In: Proc. of CVPR, USA (2005)
6. Rusu, R.B., Blodow, N., Beetz, M.: Fast Point Feature Histograms (FPFH) for 3D Registration. In: Proceedings of ICRA (2009)
7. Shapovalov, R., Velizhev, A.: Cutting-Plane Training of Non-associative Markov Network for 3D Point Cloud Segmentation. In: International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (2011)
8. Golovinskiy, A., Funkhouser, T.: Min-cut based segmentation of point clouds. In: IEEE Workshop on Search in 3D and Video at ICCV (2009)
9. Nguyen, A., Le, B.: 3D Point Cloud Segmentation: A survey. In: Proceedings IEEE International Conference on Robotics, Automation and Mechatronics (RAM), Philippines (2013)
10. Rusu, R.B., Cousins, S.: 3D is here: Point Cloud Library (PCL). In: Proceedings of ICRA, China (2011)
11. Munoz, D., Vandapel, N., Hebert, M.: Onboard contextual classification of 3D point clouds with learned high-order Markov Random Fields. In: Proceedings of ICRA (2009)
12. Kalogerakis, E., Hertzmann, A., Singh, K.: Learning 3D Mesh Segmentation and Labeling. In: Proceedings of SIGGRAPH (2010)
13. Koppula, H.S., Anand, A., Joachims, T., Saxena, A.: Semantic Labeling of 3D Point Clouds for Indoor Scenes. In: Proceedings of NIPS (2011)
14. Huang, Q., Han, M., Wu, B., Ioffe, S.: A hierarchical conditional random field model for labeling and segmenting images of street scenes. In: Proceedings of CVPR (2011)
15. Torralba, A., Murphy, K.P., Freeman, W.T.: Sharing Visual Features for Multiclass and Multiview Object Detection. IEEE Trans. Pattern Anal. Mach. Intell. (2007)
16. Lim, E.H., Suter, D.: Conditional Random Field for 3D Point Clouds with Adaptive Data Reduction. In: International Conference on Cyberworlds (2007)
17. Shotton, J., Winn, J., Rother, C., Criminisi, A.: TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context. Int. J. Comput. Vision 81(1)
18. Xiong, X., Munoz, D., Bagnell, J.A., Hebert, M.: 3D scene analysis via sequenced predictions over points and regions. In: Proceedings of ICRA (2011)
19. Johnson, A.E., Hebert, M.: Surface matching for object recognition in complex three-dimensional scenes. Image Vision Comput. 16 (1998)
20. Munoz, D., Vandapel, N., Hebert, M.: Directional associative markov network for 3D point cloud classification. In: Proceedings of 3DPVT, Atlanta, GA (2008)