

CLASSIFYING URBAN LANDSCAPE IN AERIAL LIDAR USING 3D SHAPE ANALYSIS

Matthew Carlberg, Peiran Gao, George Chen, and Avideh Zakhor

Video and Image Processing Lab, EECS Department, University of California, Berkeley

ABSTRACT

The classification of urban landscape in aerial LiDAR point clouds is useful in 3D modeling and object recognition applications in urban environments. In this paper, we introduce a multi-category classification system for identifying water, ground, roof, and trees in airborne LiDAR. The system is organized as a cascade of binary classifiers, each of which performs unsupervised region growing followed by supervised, segment-wise classification. Categories with the most discriminating features, such as water and ground, are identified first and are used as context for identifying more complex categories, such as trees. We use 3D shape analysis and region growing to identify “planar” and “scatter” regions that likely correspond to ground/roof and trees respectively. We demonstrate results on two urban datasets, the larger of which contains 200 million LiDAR returns over $7km^2$. We show that our ground, roof, and tree classifiers, when trained on one dataset, perform well on the other dataset.

Index Terms— Airborne LiDAR, cascaded classifiers, region growing, 3D shape analysis

1. INTRODUCTION

The classification of aerial LiDAR point clouds plays an important role in providing high-level contextual information for 3D urban modeling and object recognition applications. In multi-category classification, aerial LiDAR point clouds are typically labeled using categories such as ground, building, and vegetation. For example, Charaniya *et al.* perform pixel-wise, four-category classification using expectation maximization with features such as height variation and return intensity, computed over a 2.5D height map [1]. Forlani *et al.* use two region growing segmentations followed by rule-based, segment-wise classification to identify three classes of LiDAR returns [2].

Whereas the airborne methods described above process data in the 2.5D domain, methods that use ground-based LiDAR often take advantage of full 3D analysis [3]. For example, Lalonde *et al.* use 3D shape analysis and a Bayesian classifier to identify planar, scatter, and linear LiDAR returns in two relatively complex, yet small, terrestrial datasets [3]. As a post-processing step, they perform region growing in 3D to connect proximal LiDAR returns of the same category.

In this paper, we present a modular system for labeling aerial LiDAR returns as *water*, *ground*, *roof*, *tree*, or *other*. The input to our system is an aerial, colored LiDAR point cloud with each return

specified by an (x, y, z) position in a global coordinate system and an associated (r, g, b) color value. Our system is composed of cascaded binary classifiers, as shown in Fig. 1. Each binary classifier “peels off” positively identified LiDAR returns from the processing pipeline. Categories of LiDAR returns that exhibit the most discriminating features, such as water and ground, are identified first and used as context for identifying more complex classes of urban landscape. The dotted lines in Fig. 1 indicate the passing of contextual information. The modularity of our system enables each binary classifier to use its own unique set of discriminating features. Unlike most work on aerial LiDAR classification, we use 3D shape analysis to grow “planar” (“scatter”) regions, which are then used by our classifiers to detect ground/roof(trees).

Our work most resembles that of [3] and [4]. Similar to [4], our binary classifiers perform region growing segmentation as a means of enforcing spatial coherency, followed by segment-wise classification. However, our system is a multi-category classification system that works in the 3D domain rather than in 2.5D. The 3D shape analysis aspect of our approach is similar to Lalonde *et al.* [3]. Whereas Lalonde *et al.* generate 3D segments corresponding to planar, scatter, and linear regions for the purpose of identifying navigable terrain, we perform an extra segment-wise classification on planar and scatter segments to arrive at urban landscape categories. In addition, our work deals with airborne data, which is inherently different from terrestrial data due to spatio-temporal resolution, acquisition geometry/distance, and noise levels.

We test our classification system on two different datasets D1 and D2. D1 captures range information over a hilly, lightly urban city in North America with modern architecture including skyscrapers. It contains approximately 125 million LiDAR returns over $3km^2$ with an average spatial density of $65 \text{ returns}/m^2$. D2 captures range information over a flat, densely urban city in Europe with mostly 19th century architecture. It contains approximately 200 million LiDAR returns over $7km^2$ with an average spatial density of $25 \text{ returns}/m^2$.

Section 2 describes our water classifier, and Section 3 outlines 3D segmentation and classification for ground, roofs, and trees. We present our results in Section 4.

2. WATER CLASSIFIER

The water classifier is first in our system, because water tends to exhibit the most discriminating features, the most notable of which is low return density as compared to highly scattering non-water areas. It is the only component of our system that processes a 2.5D depth image, which is justified considering that the water surfaces of interest typically lack 3D structure. Our proposed water classifier works in two steps. In the first step, we perform region growing segmentation on the 2.5D image based on the density of empty pixels with no returns in a neighborhood. In the second step, a trained random forest classifier [5] identifies each segment as either water or non-water. Empirically, we have found that large segments that are low in height and contain few returns tend to correspond to water. Therefore, for segment-wise classification, we use the fol-

This work is supported with funding from the Defense Advanced Research Projects Agency (DARPA) under the Urban Reasoning and Geospatial Exploitation Technology (URGENT) Program. This work is being performed under National Geospatial-Intelligence Agency (NGA) Contract Number HM1582-07-C-0018, which is entitled, Object Recognition via Brain-Inspired Technology (ORBIT). The ideas expressed herein are those of the authors, and are not necessarily endorsed by either DARPA or NGA. This material is approved for public release; distribution is unlimited. Distribution Statement “A” (Approved for Public Release, Distribution Unlimited)

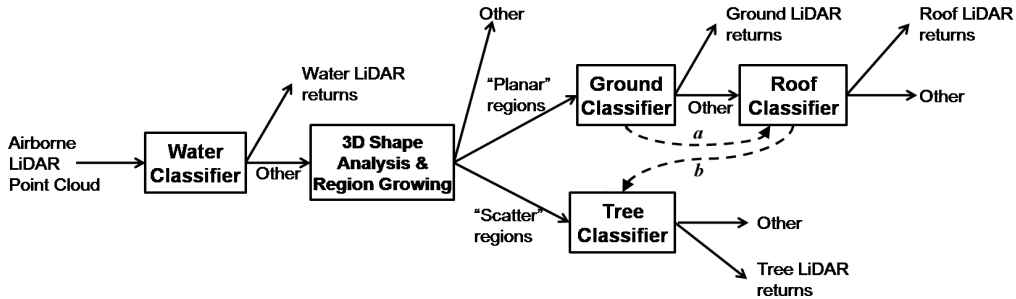


Fig. 1. Classification system composed of four binary classifiers. Dotted lines indicate the passing of contextual information.

lowing features: the segment-wise empty-pixel density, the median height of the segment, and the size of the segment in pixels.

3. THE 3D CLASSIFIERS

In contrast to our water classifier, our proposed ground, roof, and tree classifiers treat input LiDAR point clouds as fully three dimensional data. Since modern airborne LiDAR data acquisition systems often collect data in high-density, wide-angled swaths with multiple fly overs, treating the data as three dimensional is justified, if not ideal.

3.1. 3D Shape Analysis and Region Growing

Our 3D region growing algorithm is inspired by the saliency features of Medioni *et al.* [6]. Specifically, we use principle component analysis (PCA) to analyze the spatial distributions of neighborhoods of points. For each non-water point of the 3D dataset, we first collect all neighbors within a specified radius r_{pca} , which defines the scale of the neighborhood over which spatial analysis is performed. PCA results in a set of eigenvalues $(\lambda_{max}, \lambda_{mid}, \lambda_{min})$, which encode the size and shape of an ellipsoid fit over the neighborhood, and a set of eigenvectors $(\mathbf{e}_{max}, \mathbf{e}_{mid}, \mathbf{e}_{min})$, which describe the principal directions of the ellipsoid.

After PCA, we divide the LiDAR returns into three disjoint subsets—“planar” points, “scatter” points, and all remaining points. We assume that points corresponding to ground and roofs are a subset of the planar points, and that points corresponding to trees are a subset of the scatter points. Based on the interpretation of PCA as described above, we define a planar point as a point with $\lambda_{mid}/\lambda_{min} > t_{planar}$, and a scatter point as a point with $\lambda_{max}/\lambda_{mid} < t_{scatter1}$ and $\lambda_{mid}/\lambda_{min} < t_{scatter2}$, where t_{planar} , $t_{scatter1}$, and $t_{scatter2}$ are user-specified thresholds. While in an ideal case, these user specified thresholds would be learned from data, we have empirically verified our results to be insensitive to these parameters, and argue that learning these parameters would require point-wise ground truth that is tedious to produce and often inaccurate for such large, noisy datasets.

Our 3D region growing is similar to [3]. For the ground and roof classifiers, we group proximal, planar LiDAR returns into a set of segments. For the tree classifier, scatter points are segmented in a similar manner. For 3D segmentation, we grow a region through any pair of points in the same subset that are within a distance r_{seg} of each other, where r_{seg} is a user-specified parameter.

3.2. Classification

3.2.1. Ground Classifier

The input to our ground classifier is the set of non-water, locally planar segments in 3D, a subset of which is classified as ground.

We have empirically found that ground points are usually in a single large segment that is low in height, allowing for it to be identified without machine learning. For each segment, we consider a feature vector made of the median height and the number of returns in the segment. We normalize these two features to values between 0 and 1, and automatically tag as ground the segment that is closest in Euclidean distance to $(medianHeight, numReturns) = (0, 1)$. We also wish to account for the possibility of multiple ground segments; we have empirically found that most ground segments are within a factor of 10 in size of the main ground segment, and have few or no LiDAR returns that fall directly underneath them.

3.2.2. Roof Classifier

The input to the roof classifier is the set of non-water, non-ground, locally planar segments in 3D, a subset of which is classified as roof. Three segment-wise features are used for roof classification: (1) the difference between the median height of the segment and the median height of all ground returns identified in the $100m \times 100m$ tile in which the segment resides, (2) the count of LiDAR returns that fall in a window underneath a segment, normalized by the number of points in the segment, and (3) the z component of a segment’s normal vector, as calculated by a linear least squares plane fit over the segment. Intuitively, these features are used because we expect that roofs (1) are some distance above ground, (2) have a large number of LiDAR returns that fall below rooftop edges, corresponding to ground and building facades, and (3) have normal vectors that tend to point upward. To calculate a segment’s feature vector, our classifier uses the median height of ground LiDAR returns in the input point cloud. This is an example of the contextual information shared between our classifiers, as indicated by the arc labeled *a* in Fig. 1.

To create a training set for our roof classifier, we have developed a visualization tool to generate 3D segment-wise ground truth. Namely, for a training set, a human operator tags each 3D segment as ground, roof, tree, other, and none of the above. None of the above includes objects that are unidentifiable, or on which segmentation failed, causing two different classes to be grouped into a single segment. Segments that are labeled as none of the above are not included in training or cross-validation. For the roof classifier, locally planar segments labeled as “roof” (“other”) are used as the positive (negative) examples for training a random forest classifier.

3.2.3. Tree Classifier

The input to the tree classifier is the set of non-water, locally scatter segments, a subset of which is classified as tree. From our experiments, we observe that these input segments typically correspond to three types of objects: actual trees, rooftop clutter *e.g.* antennas

Classifier	Measure	D1	D1*	D2	D2*
2.5D Water	Correct Pos	597	396	43	0
	Correct Neg	4787	4965	116	123
	False Pos	25	1301	7	0
	False Neg	73	731	4	47
	Precision	96.0%	23.3%	86.0%	0%
	Recall	89.3%	35.1%	91.9%	0%
	Total Error	1.8%	27.5%	6.4%	27.6%
3D Roof	Correct Pos	446	440	287	290
	Correct Neg	38	38	27	22
	False Pos	12	12	15	20
	False Neg	5	11	8	5
	Precision	97.4%	97.3%	95.0%	93.5%
	Recall	98.9%	97.6%	97.3%	98.3%
	Total Error	3.4%	4.6%	6.8%	7.4%
3D Tree	Correct Pos	538	517	950	934
	Correct Neg	653	659	925	924
	False Pos	62	56	64	65
	False Neg	50	71	45	61
	Precision	89.6%	90.2%	93.7%	93.5%
	Recall	91.4%	87.9%	95.4%	93.9%
	Total Error	8.6%	9.7%	5.5%	6.4%
2.5D Tree	Correct Pos	12342	11024	12413	11821
	Correct Neg	7734	6883	9607	6776
	False Pos	652	1503	329	3161
	False Neg	445	1763	355	946
	Precision	95.0%	88.0%	97.4%	78.9%
	Recall	96.5%	86.2%	97.2%	92.6%
	Total Error	5.2%	15.4%	3.0%	18.1%

Table 1. Segment-wise confusion matrices and average precision, recall, and total error rates for our binary classifiers. The D1*(D2*) columns correspond to using the D2(D1) trained model to test on the D1(D2) training set.

and building edges, and objects low to the ground *e.g.* cars. Our first segment feature is the count of points that fall in a window below a segment, normalized by the number of points in the segment. We expect this feature to aid in discriminating between cars that are on the ground and tree foliage through which LiDAR returns tend to penetrate. The second segment feature is the percentage of points in a segment that are a significant distance from a LiDAR return identified as roof. This assists in discriminating between trees and rooftop clutter and is another example of the context used in our classifiers, as indicated by the arc labeled b in Fig. 1. Our third and fourth features are $\bar{\lambda}_{max}$ and $\bar{\lambda}_{mid}/\bar{\lambda}_{min}$, where the values ($\bar{\lambda}_{max}, \bar{\lambda}_{mid}, \bar{\lambda}_{min}$) are the eigenvalues obtained from PCA over the entire segment. In the same way that LiDAR returns corresponding to trees should be scattered on a local level, they should also be scattered on a larger scale. We do not use color as a feature for identifying trees, because shadows are particularly prevalent in urban canyon and foliage color is season and species dependent.

Segment-wise ground truth is obtained as described in Section 3.2.2; we use locally scattered segments identified as “trees” (“other”) as the positive (negative) examples for training a random forest classifier.

4. RESULTS

We test our classification system on the Centos Linux x64 5.2 platform with an 8-core Intel Xeon CPU and 4GB of RAM. For the D1 and D2 datasets, our classification system processes approximately 800,000 and 645,000 LiDAR returns per minute respectively, with a majority of processing time devoted to 3D region growing. This difference between the two datasets can be attributed to the increased complexity of neighborhood searches for PCA and

region growing in D2 due to the larger r_{pca} and r_{seg} parameters. In particular, we scale the parameters r_{pca} and r_{seg} inversely proportional with the square root of the average return density. By adjusting r_{pca} we are ensuring a similar number of points in the neighborhood used by PCA; adjusting r_{seg} results in segments of comparable physical size and shape for both datasets.

Figs. 2(a) and 2(b) demonstrate overall results for D1 and D2 respectively. For each of our classifiers, we build a random forest classifier with 300 decision trees and perform 10-fold cross validation. Segment-wise confusion matrices for each dataset, along with average precision, recall, and total error rates, are reported in the D1 and D2 columns of Table 1. Because the confusion matrices are in terms of segments, they fully describe the number of positive and negative training segments for each dataset. The training data for both D1 and D2 is over a $0.5km^2$ area for our water classifier and over a $0.3km^2$ area for our 3D classifiers. We also apply the random forest classifier as trained on the D2(D1) dataset to test on the D1(D2) training set, as shown in the D1*(D2*) column of Table 1. These two experiments provide a quantitative measure of the robustness of our classifiers and their generalization capability. Specifically, they indicate that our 3D classifiers suffer minimal performance degradation when testing on the dataset for which they have not been trained. Results for the entirety of the D1 dataset are available online [7].

Even though the water classifier performs well when tested on the same dataset for which it was trained, its performance degrades when it is trained on D2(D1) and tested on D1(D2), as shown in the D1*(D2*) columns of Table 1. We attribute this to the use of empty pixel density as a segment feature. Whereas a quantity such as the height of a segment is determined by the physical world, the empty pixel density is a sensor-specific feature, and highly dependent on the data acquisition process. Recall that the data acquisition sensors for D1 and D2 are different from each other. The discrepancy in the total number of segments for the two training sets has to do with the fact that the D2 dataset has more uniform point spacing, such that over the same area, it has fewer segments. This can be partially attributed to the physical characteristics of the two different LiDAR systems used in collecting these datasets. For the water classifier, columns D1*(D2*) are obtained using not only the classifier as trained on D2(D1) but also the segmentation parameters from D2(D1), resulting in different segment counts between the D1 and D1* experiments. The accuracy rates are still comparable because classified segments in all cases are evaluated based on a majority vote over pixel-wise ground truth.

For roof classification, the precision and recall rates are high, *i.e.* above 97%, because (1) we choose relevant features and (2) most segments processed by the roof classifier do indeed correspond to roofs. These confusion matrices do not take into account errors in segmentation, typically corresponding to segments that contain a mix of classes; however, less than 1% of all locally planar segments in the ground truth are mixed segments. This low figure is an encouraging indicator of the performance of our segmentation algorithm for detecting roofs.

The performance of N-category pointwise classification systems is usually characterized by an $N \times N$ confusion matrix [1]. Carrying out such a performance characterization for our system is not feasible for two reasons. First, we carry out segment-wise rather than point-wise classification and would therefore have to account for both segmentation and classification errors. Second, our system is made of a cascade of binary classifiers rather than a single multi-category classifier. To overcome this, we have opted to carry out extensive subjective evaluation to characterize the performance of our system. Specifically, a human operator has used a 3D viewer to subjectively analyze the results of our roof classifier

over the entirety of both datasets, as shown in Table 2. The operator is instructed to count the total number of buildings in each dataset, the number of buildings correctly identified by our classifier, and the number of false positive segments. A building is considered correctly identified if our roof classifier positively identifies all planar segments associated with its rooftop. Over the entirety of both datasets, our classifiers correctly classify 99% of buildings as identified by a human operator, with a false alarm rate of approximately 5%. In essence, this subjective evaluation confirms the performance of both our segmentation and classification algorithm. We have observed that false positives are usually attributed either to trees that are particularly flat or to railroad cars, which exhibit very similar physical characteristics as buildings. Segments that contain a mix of classes occur infrequently, but have been identified to happen on buildings such as parking garages that have planar ramps that connect the ground to the top of the building.

Dataset	Buildings Observed	Buildings Correct ID	Detection Rate	False Positives	False Alarm Rate
D1	975	969	99.4%	47	4.6%
D2	1812	1803	99.5%	102	5.4%

Table 2. Subjective results by human operator of roof classifier over entirety of both datasets

For the tree classifier, less than 5% of input scatter segments in the ground truth for both datasets correspond to mixed segments. Since trees exhibit significantly more 3D characteristics than water, ground, or roofs, we also compare our 3D tree classifier with a more “traditional” 2.5D tree classifier. The 2.5D tree classifier is described in detail in [8], and its cross-validation results are also shown in Table 1.

A human operator has utilized a 2D viewer to subjectively characterize the performance of both the 2.5D and 3D tree classifiers on the entirety of both datasets. Since it is nearly impossible for a human operator to count each individual tree, we instead quantize the approximate misclassification rate for a $100m \times 100m$ tile of LiDAR data into three broad categories: less than 10%, between 10% and 30%, and larger than 30%. The three categories are referred to as minimal, moderate, and significant error respectively in Table 3, which shows the subjective results for the 266 and 711 tiles of D1 and D2 respectively. Significant errors typically correspond to a large collection of trees misclassified as non-tree, or many building edges and significant rooftop clutter identified as trees. Moderate errors, on the other hand, are most often caused by mixed segments, containing a tree and some other object such as a car or building edge, being identified as a tree. Less frequent in this category are false positives corresponding to a single car or false negatives corresponding to one or two trees.

Dataset	Classifier	Total Tiles	Minimal Error	Moderate Error	Significant Error
D1	2.5D	266	49.6%	43.2%	7.1%
D1	3D	266	74.8%	23.3%	1.9%
D2	2.5D	711	71.6%	26.7%	1.7%
D2	3D	711	75.1%	20.8%	4.1%

Table 3. Subjective results by human operator for 2.5D and 3D tree classifier over entirety of both datasets.

In comparing our 2.5D and 3D tree classifiers in Tables 1 and 3, even though the 2.5D classifier is more accurate in 10-fold cross-validation, the 3D classifier appears to generalize better to the dataset on which it has not been trained, as demonstrated by the D1* and D2* columns of Table 1. In addition, Table 3 shows that from a subjective point of view our 3D classifier outperforms

the 2.5D classifier, particularly for the D1 dataset. By projecting our 3D classification results to 2D, we observe that the two classifiers disagree for 4.6%(5.6%) of pixels in the entire D1(D2) dataset. More importantly, the 3D classifier assigns a label to every 3D point by not projecting onto 2D, thus allowing for the correct identification of structures such as rooftops that lie underneath trees.

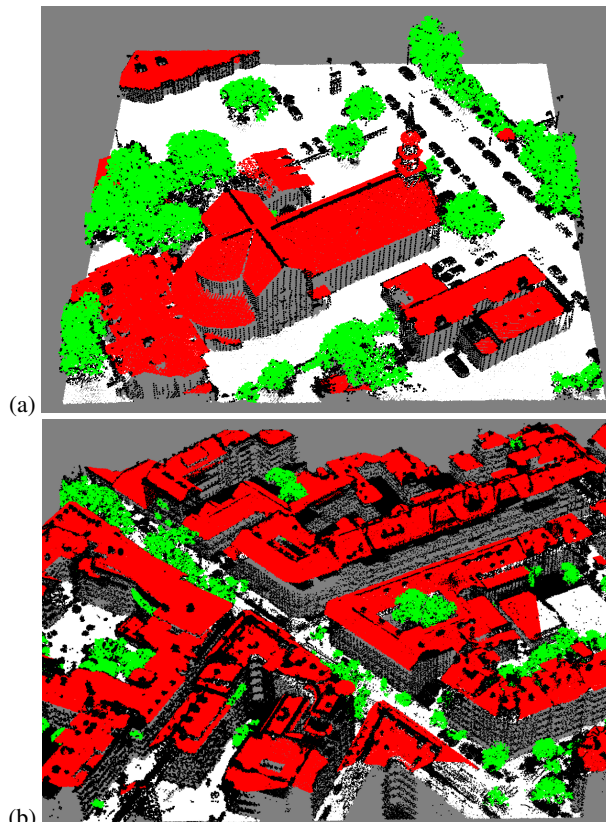


Fig. 2. Classified point cloud from (a) D1 and (b) D2.

References

- [1] Amin P. Charaniya, Roberto M, and Suresh K. Lodha, “Supervised parametric classification of aerial lidar data,” *Computer Vision and Pattern Recognition Workshop*, pp. 25–32, 2004.
- [2] G. Forlani, C. Nardinocchi, M. Scaioni, and P. Zingaretti, “Complete classification of raw lidar data and 3d reconstruction of buildings,” *Pattern Analysis and Applications*, vol. 9, no. 4, pp. 357–374, 2006.
- [3] Jean-Francois Lalonde, Nicolas Vandapel, Daniel Huber, and Martial Hebert, “Natural terrain classification using three-dimensional lidar data for ground robot mobility,” *Journal of Field Robotics*, vol. 23, no. 10, pp. 839 – 861, 2006.
- [4] J. Secord and A. Zakhor, “Tree detection in urban regions using aerial lidar and image data,” *IEEE Geo. and Remote Sensing Letters*, vol. 4, no. 2, pp. 196–200, 2007.
- [5] Leo Breiman, “Random forests,” *Machine Learning*, vol. 45, pp. 5–32, 2001.
- [6] Gerard Medioni, Mi-Suen Lee, and Chi-Keung Tang, *A Computational Framework for Segmentation and Grouping*, Elsevier Science B.V., 2000.
- [7] http://www.eecs.berkeley.edu/~carlberg/media/d1_results.zip.
- [8] George Chen and Avideh Zakhor, “2d tree detection in large urban landscapes using aerial lidar data,” To appear in *IEEE Int’l Conf. on Image Processing 2009*.