Classification of Vehicle Parts in Unstructured 3D Point Clouds

Allan Zelener
Department of Computer Science
The Graduate Center of CUNY
New York City, USA
azelener@gc.cuny.edu

Philippos Mordohai
Department of Computer Science
Stevens Institute of Technology
Hoboken, USA
mordohai@cs.stevens.edu

Ioannis Stamos
Department of Computer Science
Hunter College & Graduate Center of CUNY
New York City, USA
istamos@hunter.cuny.edu

International Conference on 3D Vision (3DV), December 2014, Tokyo, Japan

***Abstract*—Unprecedented amounts of 3D data can be acquired in urban environments, but their use for scene understanding is challenging due to varying data resolution and variability of objects in the same class. An additional challenge is due to the nature of the point clouds themselves, since they lack detailed geometric or semantic information that would aid scene understanding. In this paper we present a general algorithm for segmenting and jointly classifying object parts and the object itself. Our pipeline consists of local feature extraction, robust RANSAC part segmentation, part-level feature extraction, a structured model for parts in objects, and classification using state-of-the-art classifiers. We have tested this pipeline in a very challenging dataset that consists of real world scans of vehicles. Our contributions include the development of a segmentation and classification pipeline for objects and their parts; and a method for segmentation that is robust to the complexity of unstructured 3D points clouds, as well as a part ordering strategy for the sequential structured model and a joint feature representation between object parts.**

***Keywords*-Parts-based classification; Structured prediction; 3D point clouds; Urban range scans;**

I. Introduction

The photorealistic modeling of large-scale scenes, such as urban structures, has received significant attention in recent years (see for example [1]). State-of-the-art range sensors can produce millions of 3D points of large-scale urban areas. The complexity of urban environments is due to the variability of objects (such as buildings, people, vehicles, street level structures, roadways, curbs, etc.), partial visibility and occlusions, and varying object resolution. In addition, new low-cost range sensors provide us the ability to acquire voluminous amounts of 3D range data indoors. Applications can be seen in 3D map generation, urban planning, asset generation for game & film industries, robotic navigation, etc.

Figure 1. A scene tile from the Wright State Ottawa dataset.

Unfortunately, 3D point clouds are a very basic data structure and lack the semantic and more complete geometric knowledge about the underlying surfaces and volumes of the scene. A number of research groups address the problem of classification of various types of objects in point clouds [2], [3], [4], [5]. Classification into distinct classes (such as cars, vegetation, façades, wires, etc.) can be achieved with a high level of success. A harder problem, though, is the classification of similar class categories. For example, one would like to distinguish between sedans vs. SUVs within the vehicle class. This problem is significantly harder due to the similarity of the distributions of local features between these two classes. Our experiments show that traditional bag-of-word approaches over the entire object are not effective in that setting. However we observed, as shown in Figure 3, that the variability between different areas of the scanned object was significant. This motivated us to develop a method for the segmentation and classification of object parts that can be used to accomplish higher level tasks such as vehicle classification and semantic understanding of vehicles. We loosely define object parts in 3D point clouds as mostly planar man-made surfaces that form approximately right angles with each other. Although in this work we present vehicle parts, our approach can be generalized to many man-made objects with surfaces that are well approximated by planar segments.

In this paper we present a general algorithm for segmenting objects into semantic parts for jointly classifying an object and its parts. Our pipeline consists of local feature extraction (spin images), part segmentation (RANSAC), part-level feature extraction (bag-of-words), structured modeling of part classes and object class, and finally classification using various classifiers.

We have tested this pipeline in a very challenging dataset that consists of vehicles. The data has been acquired via

multiple sensors and includes only partial object views from varying viewpoints and at varying resolutions. The sensors were rigidly mounted on moving trucks and planes and these scans were registered together as seen in Figure 1. As such, our point clouds are unstructured, meaning we have no raster ordering and no sensor viewpoint information. Moreover, the cars in our dataset are complex and noisy because of the existence of large regions of missing surface data caused by transparent windows and reflective metallic surfaces. Our experiments in this dataset of 222 objects demonstrate high accuracy for part classification and an improvement in object classification over standard methods.

II. Related Work

In this section, we briefly review research related to ours, focusing on part-based approaches for the analysis of 3D data. Covering the 2D literature is a daunting task, and is beyond the scope of this paper. Since we work with partial and noisy scans, we pay more attention to methods that are applicable to such data. Thus, we exclude research that requires watertight, manifold meshes as input.

Several approaches for object recognition in noisy 3D point clouds utilize local descriptors, such as spin images [6], [7], [3], shape contexts [8], salient points [9], fast point feature histograms [10], unique shape contexts (USC) [11], 3D SURF [12], Signatures of Histograms of Orientations [13], visibility contexts [14] or compact covariance descriptors [15]. Many of them have borrowed the bag of words concept from the natural language analysis literature via the object recognition in images literature. We, however, are interested in recognizing specific surfaces encountered in vehicles. These would appear indistinguishable from the perspective of these algorithms, considering their invariances. The parts we are interested in are approximately flat or slightly curved, man-made surfaces forming right angles with other similar surfaces. The majority of the local descriptors would be uninformative in such a scenario.

Semi-local descriptors, or collections of local descriptors, have been effective for this type of classification due to their robustness to missing data. By slight abuse of the term, these semi-local, not necessarily semantic, features have been called "parts". We adopt the same term in this section. Huber et al. [16] addressed a problem similar to ours relying on a parts-based approach, in which parts were represented by collections of local descriptors. Part classes were discovered via agglomerative clustering and once the prototype parts have been identified, the probability of a specific vehicle class conditioned on the presence of particular parts can be learned and used for classification. Shan et al. [17] proposed the *shapeme* histogram projection algorithm in which objects are represented as collections of shapeme histograms. A shapeme is a cluster of shape descriptors that correspond to an object segment. To enable partial matches, the shapeme histogram of the query is projected onto the subspace of the model database and matching is performed on segments shared by the model and query.

A domain in which relationships between approximately planar surfaces are explicitly taken into account is that of building analysis. As laser scanners become more prevalent and mobile, approaches for detecting buildings and recognizing their constituent parts are also on the rise. In most cases [18], [19], [20], [21], [22], [23], [24], [25], [26], the steps are similar to the first parts of our algorithm: segmentation of the unorganized point cloud and reasoning on the detected surfaces to infer the types of these surfaces as well as their role in potential building configurations. Building analysis typically ends at this point and does not proceed to classification.

Our contributions with respect to current work include the development of a pipeline for segmentation and classification of object parts and a method for segmentation of surface parts that is robust to variations in scan density, occlusions, and interior points. Scanned interior points due to transparent surface elements have particularly been neglected in works using synthetic models. Whereas Huber et al. [16] segment vehicles into three parts: front, middle, and back, using prior orientation information for vehicle models, our approach utilizes only the gravity direction for orientation of automatically segmented parts. We also present a sequential model for structured prediction over the segmented parts. This includes a method for ordering the parts to reduce the effects of density variations due to sensor orientation with respect to visible surfaces and increase the density of specific joint part occurrences for learning binary potentials in the sequential model. For this model we have generated a representation for joint features computed between parts that is used for discriminatively training binary potentials.

III. Part Segmentation & Classification Pipeline

Given a set of objects extracted from range scans, we propose a pipeline for the segmentation and joint classification of the object and its parts. The pipeline components include local feature extraction, segmentation of an object's surface into planar segments, and construction of part-level features for each segment as well as joint features between parts for the structured classification model.

In Section III-A, we describe preprocessing and local feature extraction across the given object point clouds. This includes the selection and parametrization of local 3D feature descriptors and the sampling strategy for where local features will be computed.

Section III-B discusses the segmentation of object point clouds into planar segments using a RANSAC approach. We present a model hypothesis selection strategy using the object's convex hull that favors planar segments on the exterior surface of each object. This produces a more consistent segmentation across all objects and avoids planes fit through the object's interior.

We combine the local features and the object segments in Section III-C in order to build part-level features for part classification using a bag-of-words model over our chosen 3D feature descriptor. In Section III-D, we describe a structured model and features over neighboring parts for structured classification. We compare classification results between structured and unstructured models in Section IV.

A. Local Feature Extraction

We define local features as statistics computed with respect to a reference point and neighboring points within a fixed radius of support. For 3D feature descriptors these statistics typically include quantizations of point positions or normal orientations parameterized within the support space. In preliminary tests using the spin image [6], FPFH [10], and USC [11] feature representations we found that the spin image was marginally superior in our pipeline and we used it for our final experiments.

In order to ensure that we only consider points with well-populated supports, we first pass each object through a statistical outlier filter [27]. Assuming a Gaussian distribution over the means of the distances between points and their k-nearest neighbors, here we take $k = 8$, the filter removes points whose average distance to their neighbors is more than a standard deviation away from the mean of average distances for all points in the given object.

Surface normals are estimated locally at each point using PCA with a 0.3m radius of support. Because objects in our dataset come from registered range scans of heterogeneous sensors we assume that only the geo-registered up-direction can be relied upon. To give a consistent orientation to all the normals we find the centroid of the object's footprint on the ground and orient the normals away from this reference point.

Each object is subsampled using a 0.2m^3 voxel grid, taking the object point within each voxel that is closest to the voxel centroid. We compute spin images that are densely sampled on the point cloud instead of using keypoint detection because we are interested in the distribution of local features in an object part rather than simply matching distinct features between point clouds. Spin image features of size 9 by 17 are computed at each sampled point using a 1.8m cylindrical radius and 3.6m height. Bilinear interpolation is used to smooth the spin image quantization. In order to account for variable density scans, the contribution of each point to a spin image is weighted by its inverse density defined as the inverse of the number of neighbors in a 0.3m radius. Finally the spin image is normalized using the L_1 norm.

By using a large support radius with a small grid quantization we can capture localized features that broadly indicate feature position with respect to the global object shape. This parameterization of the features makes them amenable to quantization for a visual bag-of-words model and well-suited to the part classification task.

B. Part Segmentation

Our segmentation technique assumes that the objects have roughly piecewise planar exterior surfaces which is a reasonable assumption for man-made objects such as vehicles at the level of detail of these range scans. This unsupervised segmentation into planar segments will be the basis for our part-level feature representation and we expect the distribution of local features will vary significantly between planar segments as can be seen in Figure 3.

Planar segments are iteratively extracted by fitting planar models using an adaptive RANSAC approach as described by Hartley and Zisserman [28]. Candidate planar model parameters are randomly generated and evaluated by counting the number of points within a point-to-plane projection distance threshold. After a sufficient number of random trials, depending on the fraction of model inliers, the model with the most inliers is selected as the most likely planar segment. We set an upper limit of 500 on the number of trials in case no good model can be fit.

Typically a planar model hypothesis can be generated from a random sampling of three points that are not colinear. However because vehicles are often occluded in urban settings and contain transparent windows, it is possible to fit poor parts that intersect through the interior due to a relative oversampling of interior points. We avoid such undesirable segmentation by only admitting the facets of an object's convex hull, generated using the QHull [29] algorithm, as possible model hypothesis. So for each RANSAC trial we randomly select a convex hull facet and fit inliers to the facet's planar parameters. There are typically several hundered initial facets and these are pruned from the available pool of candidate models when there are no remaining inliers for the facet's planar model.

To determine a distance threshold for planar model inliers we specify several assumptions about the distribution of inliers. We conservatively set the probability of any given point being an inlier for a given plane to $\alpha = 0.01$. We also assume that the position of each point is associated with a Gaussian noise with variance $\sigma^2 = 0.002$. Therefore the squared distance error for a point from a planar model takes a χ^2-distribution and we can use a t-test to determine the distance threshold d_t as

$$d_t = \sqrt{\frac{\sigma^2}{\chi^2_{cdf}(\alpha, 1)}} \tag{1}$$

where $\chi^2_{cdf}(\alpha, 1)$ is the cumulative distribution function of the χ^2-distribution at α with 1 degree of freedom.

Once a sufficent number of RANSAC trials have determined the most likely planar model, it is robustly re-estimated several times through expectation-maximization

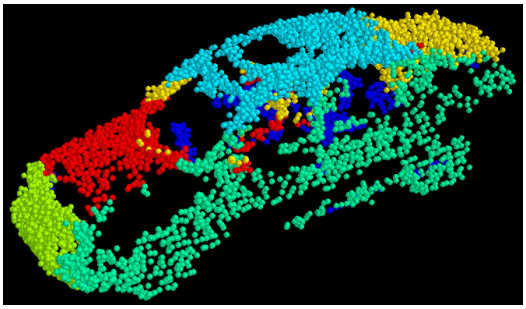

Figure 2. Planar segmentation of a sedan. Dark blue points correspond to unsegmented and unlabled points, typically interior points. Here the manual ground truth labels for each segment in the order the segments were automatically extracted are light blue roof, cyan lateral-side, lime green front-bumper, yellow trunk, and red hood. Our method is robust to some interior points being included in these segments.

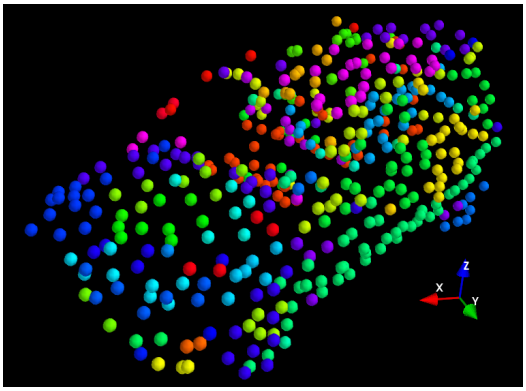

Figure 3. Each vehicle point corresponds to a sampled feature with the color corresponding to the closest codebook word in the codebook of size $k = 50$. Note that the distribution of corresponding codebook words changes significantly along different parts of the vehicle, this motivates our part-based classification.

using PCA on all of the inliers. This step is particularly helpful since using only the convex hull facets as the initial models biases the estimated planes towards noisy exterior points. The final part inliers are then removed from the object and the procedure is repeated on the remaining points until 5 planes have been extracted or less than 10% of the original points remain. Examples of the resulting segmentation can be seen in Figures 2 and 6.

C. Part-Level Feature Extraction

Using the densely sampled local descriptors, we construct feature vectors for each extracted planar segment using a bag-of-words model to generate a part-level representation. We create a codebook of spin image features shown in Figure 4 by using the k-means algorithm over all of the local features in those objects in the training set for classification. Here the size of our codebook is $k = 50$. In early experiments codebook size did not significantly impact classification results.

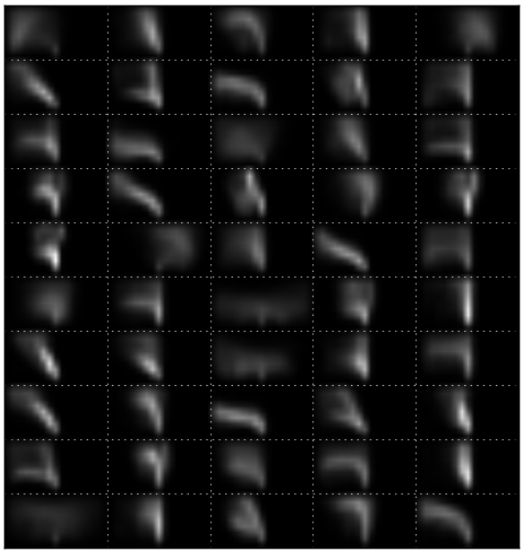

Figure 4. Spin image codebook. The bottom center of each spin image corresponds to the origin from which the spin image is computed. The y-axis corresponds to the radial direction and the x-axis to the cylinder height.

For unseen spin images the corresponding codebook word is found using a kd-tree. The feature vector for each part is then computed as the counts of each codebook word correspondence for the spin images within that part.

Additional part-level features that give a more global description of the parts are also computed. We include the average height of the points in each part as a feature because we assume the up direction is reliable in the registered scene coordinate frames. We also include a horizontal/vertical indicator that is determined by the part normal $\boldsymbol{n}$ as

$$I(\boldsymbol{n}) = \begin{cases} 0 & \text{if } \boldsymbol{n}^T \boldsymbol{z} > \cos(\frac{\pi}{4}) \\ 1 & \text{if } \boldsymbol{n}^T \boldsymbol{z} \leq \cos(\frac{\pi}{4}) \end{cases} \quad (2)$$

where $\boldsymbol{z}$ is the up-direction vector. The indicator is 0 for a horizontal part and 1 for a vertical part. In the vehicle part domain most of the parts are either clearly vertical or horizontal. Finally we include the mean, median, and max of the plane fit errors of the points in each part, the three eigenvalues from the plane estimation ($\lambda_1, \lambda_2, \lambda_3$ in descending order), and the differences between adjacent eigenvalues that have been referred to as linearity ($\lambda_1 - \lambda_2$) and planarity ($\lambda_2 - \lambda_3$) in previous work [30], [31].

D. Structured Part Modeling

Although our objects come from unstructured point clouds with variable density, making structured prediction models difficult to apply, it is possible to consider the structure over our limited number of high level parts. We consider the sequential Hidden Markov Model which can be trained

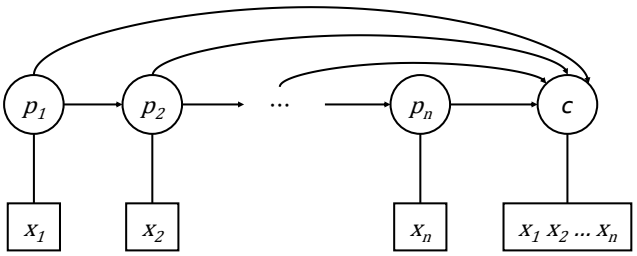

Figure 5. Generalized HMM for jointly classifying a sequence of object parts and object class. Part labels depend only upon part features and joint features with the previously predicted part. Class labels depend on the classification of all parts and their features.

online and discriminatively by averaged structured perceptron [32]. The observed variables in the HMM are the part-level features x_i and the hidden variables are the part classification labels p_i; unary and binary potentials are learned as linear combinations of observed features. Additionally we generalize the HMM to include a final hidden variable c corresponding to the object class that depends on all previously observed variables, a graph depicting this model can be found in Figure 5.

This approach requires a sequential ordering of the parts and while the initial sequential ordering from the RANSAC part extraction is superior to arbitrary permutations, it is too heavily influenced by variations in scanning directions and occlusions. Again we utilize the known up-direction to find a more reliable ordering. Parts for each object are sorted lexicographically using the tuple $(I(\boldsymbol{n}), \bar{h})$ where $I(\boldsymbol{n})$ is as defined in Eq (2) and $\bar{h}$ is the average height of the points. This leads to a more consistent ordering across all objects, where horizontal parts precede vertical parts and they are ordered from top to bottom by the height of their centroids.

We also exploit structure by computing additional joint features, $x_{i-1,i}$, between adjacent parts in the sequential ordering which will be used to discriminatively learn the HMM binary potentials. The features we use here include the cosine of the angle between part normals ($\boldsymbol{n}_1^T \boldsymbol{n}_2$), the absolute difference in average heights ($|\bar{h}_1 - \bar{h}_2|$) the distance between part centroids ($||\boldsymbol{c}_1 - \boldsymbol{c}_2||$), the closest distance between points from each part, and the coplanarity [33] as the mean, median, and max plane fit errors from the points of one part to the estimated plane of the other and vice versa.

Part labels are determined by finding the part label that maximizes the scoring function

$$s(p_i) = \max_{p_{i-1}} s(p_{i-1}) + p(x_i|p_i) + p(x_{i-1,i}|p_{i-1}, p_i). \quad (3)$$

Where $p(x|Y) = \boldsymbol{x}^T \boldsymbol{w}_Y$, the dot product of the observed features x and the learned weight vector $\boldsymbol{w}_Y$ for labels Y. Here x may either be the unary part features or the joint features between parts, and Y may either be a single part label or a pair of part labels respectively. This recursive function is evaluated by the Viterbi algorithm over the HMM.

The overall class label c is determined by

$$\max_c \sum_i p(x_i|p_i, c) + \sum_{i,j} p(x_{i-1,i}|p_{i-1}, p_i, c). \quad (4)$$

Note that here Y is either a pair of part and class labels or a triple containing two part labels and a class label. This means for example that $\boldsymbol{w}_{p_i}$ is distinct from $\boldsymbol{w}_{p_i,c}$. During training, the weight vectors for determining class are updated only if the classification for the corresponding part was correct, otherwise we may be penalizing the wrong weight vector. The convergence of perceptron training relies on updates only on incorrect examples.

IV. Experimental Results

We tested our proposed pipeline using vehicles from the Wright State Ottawa[1] data set. Our annotations of vehicle parts and classes for this data set will be made available. Each vehicle was manually segmented from registered scene tiles. A total of 331 vehicle point clouds were collected, however we restricted our experiments to the most common vehicle categories of sedan and SUV. We partitioned the 222 sedan and SUV models into training and test sets as well as a development set for tuning an online classifier. The training set for classification consists of parts extracted from 81 sedans and 35 SUVs. The development set and test set each contain 37 sedans and 16 SUVs. The vehicle models in this data vary greatly in occlusions, observed surfaces, and density. On average each vehicle consists of several thousand points however there are some sparse vehicles with only a hundred. The densest model in our data set contains 16k points.

After the planar segments were extracted from each vehicle, as described in Section III-B, they were manually labeled into one of the following positional categories: roof, roof-hood, roof-trunk, hood, trunk, hood-trunk-merged, lateral-side, front-bumper, rear-side, interior, and misc. Examples of these labels can be found in Figures 2 and 6. The roof-hood and roof-trunk labels correspond to under-segmentation generally caused by inclines in the hoods or trunks of vehicles or occlusions of these parts which limit the number of points that can be fit. The hood-trunk-merged label also corresponds to the more rare undersegmentation of a vehicle's hood and trunk. Although generally not planar, interior segments are often extracted for particularly occluded models. The rear-side label is used for both rear

[1] http://wsri.wright.edu/applied-research-corporation/ottawa-data-files.html

Table I
COUNT OF EACH PART LABEL IN EACH DATASET.

Label	Train	Development	Test	Total
front-bumper	40	23	23	86
hood	26	7	9	42
interior	62	25	28	115
lateral-side	117	52	53	222
rear-side	39	16	18	73
roof	40	19	18	77
trunk	33	20	14	67
roof-hood	63	33	37	133
roof-trunk	12	5	5	22
hood-trunk-merged	1	3	0	4
misc	7	2	2	11

bumpers and vertical trunk doors. In rare cases noisy points and remaining ground plane elements may also be extracted and are given the misc label. The number of each extracted part label for the data set can be found in Table I.

We tested a variety of classifiers in the scikit-learn [34] package and found that we achieved the best performance with the SVM and random forest classifiers. For the SVM and random forest we used the codeword count feature vector augmented with $I(\mathbf{n})$ and $\bar{h}$ described in Section III. Ideal settings for classifiers were found using a grid search with 3-fold cross validation on the training set. For the SVM the best results were using the LIBLINEAR [35] implementation with an L1 norm in the penalization and penalty parameter $C = 1.0$ and using the one-vs-rest multiclass approach. The random forest classifier was trained with 1000 trees each of which drawing up to the square root of the number of features from the feature vectors.

The structured model that we presented in Section III-D was trained discriminatively by an averaged structured perceptron [32]. The bag-of-word features from Section III-C are normalized by the L_2 norm and features between parts are generated as described in Section III-D. We trained the model with 30 passes over the training set and validated model weights against the development set after approximately every 10% of each training pass. The structured perceptron quickly achieved best performance on the development set within 5 epochs.

Overall part classification results are presented in Table II. Using simple joint features, the large-margin structured perceptron is able to outperform both the max-margin SVM and the random forest. This shows the advantage of using additional structural information even when compared to more powerful classification algorithms.

To evaluate the automatic segmentation a subset of the data containing 90 sedans and all 67 SUVs was manually segmented. We see in table II that while the SVM and random forest perform similarly with both segmentations, the structured perceptron is better able to utilize the manual segmentation to classify parts. Table III shows the results for the sedan vs SUV object classification, using only the unary features for SVM and random forest. While the automatic labeling does not achieve the same performance as the SVM we see that with perfect segmentation the structured perceptron would be superior.

Table II
OVERALL PART CLASSIFICATION RESULTS. PART ACC CORRESPONDS TO THE PERCENTAGE OF CORRECTLY CLASSIFIED PARTS. ALL ACC IS THE PERCENTAGE OF VEHICLES FOR WHICH ALL PARTS ARE CORRECTLY CLASSIFIED. MANUAL REFERS TO THE MANUALLY SEGMENTED DATA SET.

Classifier	Part Acc	All Acc
SVM	76.10	41.50
RF	82.44	54.72
SP	**88.29**	**56.60**
Manual SVM	82.18	40.00
Manual RF	86.14	50.00
Manual SP	**93.56**	**65.00**

Table III
CLASSIFICATION ACCURACY FOR SEDAN VS SUV. WITHOUT PARTS THE SVM ACHIEVES GOOD ACCURACY AND THE UNSTRUCTURED PERCEPTRON IS SIGNIFICANTLY LESS POWERFUL. USING PART STRUCTURE THE PERCEPTRON CAN COMPETE WITH AND EXCEED THE UNSTRUCTURED CLASSIFIERS

Classifier	Unstructured	Automatic	Manual
SVM	**83.02**	–	–
RF	79.25	–	–
Perceptron	62.26	77.36	**87.5**

Results for each part class can be found in Table IV. The averaged structured perceptron obtains the best results across most part classes. We note that the uncommon hood and roof-trunk classes are handled better by the structured perceptron, using context queues to better disambiguate those classes from the more common roof-hood class which has similar position and overlapping undersegmented components. There is an additional benefit here due to our sequential part ordering which places horizontal surfaces next to each other, allowing the perceptron to determine whether a roof and hood co-occur or whether there is a roof-hood undersegmentation.

V. CONCLUSION

In this paper we present a general algorithm for segmenting objects in unstructured 3D point clouds into semantic parts and a model for structured prediction over those parts. In particular our segmentation algorithm is robust to the complexities of point clouds and avoids non-surface segments as compared to naive RANSAC segmentation.

We evaluated our classification pipeline on a challenging dataset consisting of similar vehicle objects. We achieved

Table IV
RESULTS PER PART LABEL. THE AVERAGE F-MEASURE IS WEIGHTED BY THE COUNT OF EACH PART LABEL IN THE TEST SET. MISC AND HOOD-TRUNK-MERGED LABELS WHICH APPEAR INFREQUENTLY IN THE DATA SET ARE NEVER PREDICTED AND ARE NOT SHOWN.

	Precision			Recall			F-Measure			
Part Label	SVM	RF	SP	SVM	RF	SP	SVM	RF	SP	Count
front-bumper	0.87	0.80	**1.00**	0.57	0.52	**0.87**	0.68	0.63	**0.93**	23
hood	0.00	**1.00**	0.86	0.00	0.33	**0.67**	0.00	0.50	**0.75**	9
interior	0.66	0.65	**0.77**	0.68	**0.86**	**0.86**	0.67	0.74	**0.81**	28
lateral-side	0.83	0.89	**0.93**	0.91	**0.96**	0.94	0.86	**0.93**	**0.93**	53
rear-side	**0.94**	0.84	**0.94**	**0.89**	0.89	**0.89**	**0.91**	0.86	**0.91**	18
roof	0.64	**0.93**	0.85	0.78	0.78	**0.94**	0.70	0.85	**0.89**	18
roof-hood	0.78	0.88	**0.95**	0.95	**0.97**	**0.97**	0.85	0.92	**0.96**	37
roof-trunk	0.00	0.00	**0.40**	0.00	0.00	**0.40**	0.00	0.00	**0.40**	5
trunk	0.58	0.72	**0.77**	0.79	**0.93**	0.71	0.67	**0.81**	0.74	14
Avg	0.72	0.82	**0.89**	0.76	0.82	**0.88**	0.73	0.81	**0.88**	205

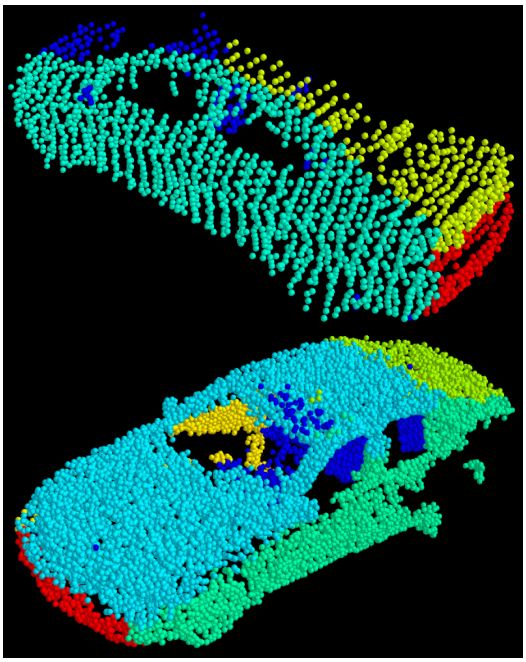

Figure 6. Automatic segmentation results with ground truth labels. Dark blue points are unlabeled. Top: yellow roof-trunk, cyan side, red rear-side. Bottom: light blue roof-hood, cyan side, red front-bumper, lime green trunk, yellow interior.

high accuracy on most of the well represented parts in our dataset. Our structured model exhibited an ability to predict under-represented classes and the structured perceptron achieved competitive performance with classifiers that have stronger theoretical properties. Future work in this direction would further refine the quality of automatic segmentation and the power of the structured prediction model. The method can also be adapted to generate semantic object models for understanding and planning interactions with man-made objects, for example localizing and manipulating the door on the side of a vehicle.

We believe that our proposed pipeline could be used as a framework for joint part and object classification in point clouds, particularly for discriminating structurally similar classes of man-made objects.

ACKNOWLEDGMENT

This work has been supported in part by NSF grants IIS-0915971 and CCF-0916452.

REFERENCES

[1] P. Musialski, P. Wonka, D. G. Aliaga, M. Wimmer, L. V. Gool, and W. Purgathofer, "A survey of urban reconstruction," in *EUROGRAPHICS State of the Art Reports*, 2012.

[2] D. Anguelov, B. Taskar, 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*, vol. 2, 2005, pp. 169–176.

[3] A. Patterson, P. Mordohai, and K. Daniilidis, "Object detection from large-scale 3D datasets using bottom-up and top-down descriptors," in *ECCV*, 2008, pp. 553–566.

[4] D. Munoz, J. Bagnell, and M. Hebert, "Co-inference for Multi-modal Scene Analysis," in *ECCV*, 2012.

[5] I. Stamos, O. Hadjiliadis, H. Zhang, and T. Flynn, "Online algorithms for classification of urban objects in 3D point clouds," in *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, 2012.

[6] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 433–449, 1999.

[7] B. Matei, Y. Shan, H. Sawhney, Y. Tan, R. Kumar, D. Huber, and M. Hebert, "Rapid object indexing using locality sensitive hashing and joint 3D-signature space estimation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 7, pp. 1111–1126, July 2006.

[8] A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *ECCV*, 2004, pp. Vol III: 224–237.

[9] U. Castellani, M. Cristani, S. Fantoni, and V. Murino, "Sparse points matching by combining 3D mesh saliency with statistical descriptors," *Computer Graphics Forum*, vol. 27, no. 2, pp. 643–652, 2008.

[10] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3D registration," in *ICRA*, 2009, pp. 3212–3217.

[11] F. Tombari, S. Salti, and L. Di Stefano, "Unique shape context for 3D data description," in *Proceedings of the ACM workshop on 3D object retrieval*. ACM, 2010, pp. 57–62.

[12] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. Van Gool, "Hough transform and 3D SURF for robust three dimensional classification," in *ECCV*, 2010, pp. VI: 589–602.

[13] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *ECCV*, 2010, pp. 356–369.

[14] E. Kim and G. Medioni, "3D object recognition in range images using visibility context," in *IROS*, 2011, pp. 3800–3807.

[15] D. Fehr, A. Cherian, R. Sivalingam, S. Nickolay, V. Morellas, and N. Papanikolopoulos, "Compact covariance descriptors in 3d point clouds for object recognition," in *ICRA*, 2012.

[16] D. F. Huber, A. Kapuria, R. Donamukkala, and M. Hebert, "Parts-based 3D object classification," in *CVPR*, 2004, pp. II: 82–89.

[17] Y. Shan, H. Sawhney, B. Matei, and R. Kumar, "Shapeme histogram projection and matching for partial object recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 568–577, April 2006.

[18] H. Maas and G. Vosselman, "Two algorithms for extracting building models from raw laser altimetry data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 54, no. 2-3, pp. 153–163, 1999.

[19] N. Haala, S. Becker, and M. Kada, "Cell decomposition for the generation of building models at multiple scales," in *Photogrammetric Computer Vision*, 2006.

[20] C. C. Chen and I. Stamos, "Range image segmentation for modeling and object detection in urban scenes," in *3DIM*, 2007, pp. 185–192.

[21] B. C. Matei, H. S. Sawhney, S. Samarasekera, J. Kim, and R. Kumar, "Building segmentation for densely built urban regions using aerial lidar data," in *CVPR*, 2008.

[22] C. Poullis and S. You, "Automatic reconstruction of cities from remote sensor data," in *CVPR*, 2009, pp. 2775–2782.

[23] A. Toshev, P. Mordohai, and B. Taskar, "Detecting and parsing architecture at city scale from range data," in *CVPR*, 2010.

[24] Q. Zhou and U. Neumann, "2.5D building modeling with topology control," in *CVPR*, 2011, pp. 2489–2496.

[25] F. Lafarge and C. Mallet, "Creating large-scale city models from 3D-point clouds: A robust approach with hybrid representation," *IJCV*, vol. 99, no. 1, pp. 69–85, 2012.

[26] H. Lin, J. Gao, Y. Zhou, G. Lu, M. Ye, C. Zhang, L. Liu, and R. Yang, "Semantic decomposition and reconstruction of residential scenes from lidar data," *ACM Transactions on Graphics*, vol. 32, no. 4, 2013.

[27] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *ICRA*, May 9-13 2011.

[28] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[29] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software (TOMS)*, vol. 22, no. 4, pp. 469–483, 1996.

[30] A. Anand, H. S. Koppula, T. Joachims, and A. Saxena, "Contextually guided semantic labeling and search for three-dimensional point clouds," *The International Journal of Robotics Research*, vol. 32, no. 1, pp. 19–34, 2013.

[31] O. Kahler and I. Reid, "Efficient 3d scene labeling using fields of trees," in *ICCV*, 2013, pp. 3064–3071.

[32] M. Collins, "Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 2002, pp. 1–8.

[33] N. Silberman and R. Fergus, "Indoor scene segmentation using a structured light sensor," in *ICCV Workshops*, 2011, pp. 601–608.

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[35] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.