3D Laser Scan Classification Using Web Data and Domain Adaptation

Kevin Lai **Dieter Fox**
University of Washington, Department of Computer Science & Engineering, Seattle, WA

***Abstract*— Over the last years, object recognition has become a more and more active field of research in robotics. An important problem in object recognition is the need for sufficient labeled training data to learn good classifiers. In this paper we show how to significantly reduce the need for manually labeled training data by leveraging data sets available on the World Wide Web. Specifically, we show how to use objects from Google's 3D Warehouse to train classifiers for 3D laser scans collected by a robot navigating through urban environments. In order to deal with the different characteristics of the web data and the real robot data, we additionally use a small set of labeled 3D laser scans and perform *domain adaptation*. Our experiments demonstrate that additional data taken from the 3D Warehouse along with our domain adaptation greatly improves the classification accuracy on real laser scans.**

I. Introduction

In order to navigate safely and efficiently through populated urban environments, autonomous robots must be able to distinguish between objects such as cars, people, buildings, trees, and traffic lights. The ability to identify and reason about objects in their environment is extremely useful for autonomous cars driving on urban streets as well as robots navigating through pedestrian areas or operating in indoor environments. Over the last years, several robotics research groups have developed techniques for classification tasks based on visual and laser range information [22, 1, 7, 21, 15, 17]. A key problem in this context is the availability of sufficient labeled training data to learn classifiers. Typically, this is done by manually labeling data collected by the robot, eventually followed by a procedure to increase the diversity of that data set [17]. However, data labeling is error prone and extremely tedious. We thus conjecture that relying solely on manually labeled data does not scale to the complex environments robots will be deployed in.

The goal of this research is to develop learning techniques that significantly reduce the need for labeled training data for classification tasks in robotics by leveraging data available on the World Wide Web. The computer vision community has recently demonstrated how web-based data sets can be used for various computer vision tasks such as object and scene recognition [16, 14, 20] and scene completion [10]. These techniques take a radically different approach to the computer vision problem; they tackle the complexity of the visual world by using millions of weakly labeled images along with non-parametric techniques instead of parametric, model-based approaches. In robotics, Saxena and colleagues [18] recently used synthetically generated images of objects to learn grasp points for manipulation. Their system learned good grasp points solely based on synthetic training data.

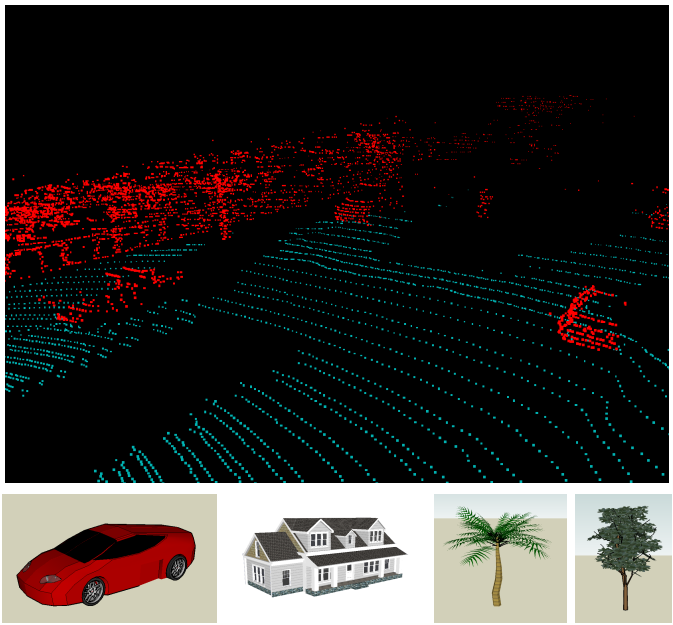

Fig. 1. (Upper row) Part of a 3D laser scan taken in an urban environment (ground plane points shown in cyan). The scan contains multiple cars, a person, and trees and buildings in the background. (lower row) Example models from Google's 3D Warehouse.

Based on these successes, it seems promising to investigate how external data sets can be leveraged to help sensor-based classification tasks in robotics. Unfortunately, this is not as straightforward as it seems. A key problem is the fact that the data available on the World Wide Web is often very different from that collected by a mobile robot. For instance, a robot navigating through an urban environment will often observe cars and people from very close range and angles different from those typically available in data sets such as LabelMe [16]. Furthermore, weather and lighting conditions might differ significantly from web-based images.

The difference between web-based data and real data collected by a robot is even more obvious in the context of classifying 3D laser scan data. Here, we want to use objects from Google's 3D Warehouse to help classification of 3D laser scans collected by a mobile robot navigating through urban terrain (see Fig. 1). The 3D Warehouse dataset [9] contains thousands of 3D models of user-contributed objects such as furniture, cars, buildings, people, vegetation, and street signs. On the one hand, we would like to leverage such an extremely

rich source of freely available and labeled training data. On the other hand, virtually all objects in this dataset are generated manually and thus do not accurately reflect the data observed by a 3D laser scanner.

The problem of leveraging large data sets that have different characteristics than the target application is prominent in natural language processing (NLP). Here, text sources from very different topic domains are often combined to help classification. Several relevant techniques have been developed for transfer learning [4] and, more recently, domain adaptation [11, 6, 5]. These techniques use large sets of labeled text from one domain along with a smaller set of labeled text from the target domain to learn a classifier that works well on the target domain.

In this paper we show how domain adaptation can be applied to the problem of 3D laser scan classification. Specifically, the task is to recognize objects in data collected with a 3D Velodyne laser range scanner mounted on a car navigating through an urban environment. The key idea of our approach is to learn a classifier based on objects from Google's 3D Warehouse along with a small set of labeled laser scans. Our classification technique builds on an exemplar-based approach developed for visual object recognition [14]. Instead of labeling individual laser points, our system labels a soup of segments [13] extracted from a laser scan. Each segment is classified based on the labels of exemplars that are "close" to it. Closeness is measured via a learned distance function for spin-image signatures [12, 2] and other shape features. We show how the learning technique can be extended to enable domain adaptation. In the experiments we demonstrate that additional data taken from the 3D Warehouse along with our domain adaptation greatly improves the classification accuracy on real laser scans.

This paper is organized as follows. In the next section, we provide background on exemplar-based learning and on the laser scan segmentation used in our system. Then, in Section III, we show how the exemplar-based technique can be extended to the domain adaptation setting. Section IV introduces a method for probabilistic classification. Experimental results are presented in Section V, followed by a discussion.

II. Learning Exemplar-based Distance Functions for 3D Laser Scans

In this section we review the exemplar-based recognition technique introduced by Malisiewicz and Efros [14]. While the approach was developed for vision-based recognition tasks, we will see that there is a rather natural connection to object recognition in laser scans. In a nutshell, the approach takes a set of labeled segments and learns a distance function for each segment, where the distance function is a linear combination of feature differences. The weights of this function are learned such that the decision boundary maximizes the margin between the associated subset of segments belonging to the same class and segments belonging in other classes. We describe the details of the approach in the context of our 3D laser classification task.

A. Laser Scan Segmentation and Feature Extraction

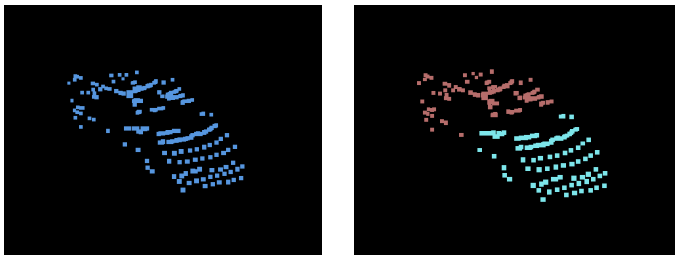

Fig. 2. (left) Laser points of a car extracted from a 3D scan. (right) Segmentation via mean-shift. The soup of segments additionally contains a merged version of these segments.

Given a 3D laser scan point cloud of a scene, we first segment out points belonging to the ground from points belonging to potential objects of interest. This is done by fitting a ground plane to the scene. To do this, we first bin the points into grid cells of size $25 \times 25 \times 25cm^3$, and run RANSAC plane fitting on each cell to find the surface orientations of each grid cell. We take only the points belonging to grid cells whose orientations are less than 30 degrees with the horizontal and run RANSAC plane fitting again on all of these points to obtain the final ground plane estimation. The assumption here is that the ground has a slope of less than 30 degrees, which is usually the case and certainly for our urban data set. Laser points close to the ground plane are labeled as ground and not considered in the remainder of our approach. Fig. 1 displays a scan with the automatically extracted ground plane points shown in cyan.

Since the extent of each object is unknown, we perform segmentation to obtain individual object hypotheses. We experimented with the Mean-Shift [3] and Normalized Cuts [19] algorithms at various parameter settings and found that the former provided better segmentation. In the context of vision-based recognition, Malisiewicz and Efros recently showed that it is beneficial to generate multiple possible segmentations of a scene, rather than relying on a single, possibly faulty segmentation [13]. Similar to their technique, we generate a "soup of segments" using mean-shift clustering and considering merges between clusters of up to 3 neighboring segments. An example segmentation of a car automatically extracted from a complete scan is shown in Fig. 2. The soup also contains a segment resulting from merging the two segments.

We next extract a set of features capturing the shape of a segment. For each laser point, we compute spin image features [12], which are 16×16 matrices describing the local shape around that point. Following the technique introduced by Assfalg and colleagues [2] in the context of object retrieval, we compute for each laser point a spin image signature, which compresses information from its spin image down to an 18-dimensional vector. Representing a segment using the spin image signatures of all its points would be impractical, so the final representation of a segment is composed of a smaller set of spin image signatures. In [2], this final set of signatures is computed by clustering all spin image signatures describing an object. The resulting representation is rotation-invariant,

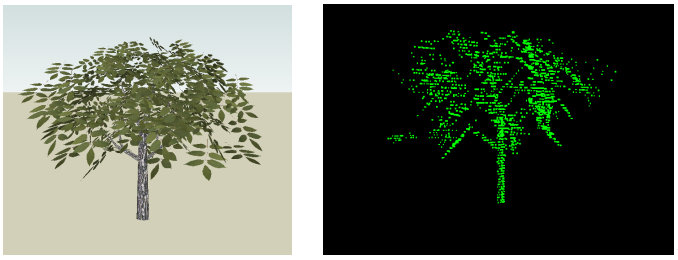

Fig. 3. (left) Tree model from the 3D Warehouse and (right) point cloud extracted via ray tracing.

which is beneficial for object retrieval. However, in our case the objects of concern usually appear in a constrained range of orientations. Cars and trees are unlikely to appear upside down, for example. The orientation of a segment is actually an important distinguishing feature and so unlike in [2], we partition the laser points into a $3 \times 3 \times 3$ grid and perform k-means clustering on the spin image signatures within each grid cell, with a fixed $k = 3$. Thus, we obtain for each segment $3 \cdot 3 \cdot 3 = 27$ shape descriptors of length $3 \cdot 18 = 54$ each. We also include as features the width, depth and height of the segment's bounding box, as well as the segment's minimum height above the ground. This gives us a total of 31 descriptors.

In order to make segments extracted from a 3D laser scan comparable to objects in the 3D-Warehouse, we perform segmentation on a point cloud generated via ray tracing on the object (see Fig. 3).

B. Learning the Distance Function

Assume we have a set of n labeled laser segments, $\mathcal{E} = \{e_1, e_2, \ldots, e_n\}$. We refer to these segments as *exemplars*, e, since they serve as examples for the appearance of segments belonging to a certain class. Let $\mathbf{f}_e$ denote the features describing an exemplar e, and let $\mathbf{f}_z$ denote the features of an arbitrary segment z, which could also be an exemplar. $\mathbf{d}_{ez}$ is the vector containing component-wise, L_2 distances between individual features describing e and z: $\mathbf{d}_{ez}[i] = ||\mathbf{f}_e[i] - \mathbf{f}_z[i]||$. In our case, features $\mathbf{f}_e$ and $\mathbf{f}_z$ are the 31 descriptors describing segment e and segment z, respectively. $\mathbf{d}_{ez}$ is a $31 + 1$ dimensional distance vector where each component, i, is the L_2 distance between feature i of segments e and z, with an additional bias term as described in [14]. Distance functions between two segments are linear functions of their distance vector. Each exemplar has its own distance function, D_e, specified by the weight vector $\mathbf{w}_e$:

$$D_e(z) = \mathbf{w}_e \cdot \mathbf{d}_{ez} \tag{1}$$

To learn the weights of this distance function, it is useful to define a binary vector $\boldsymbol{\alpha}_e$, the length of which is given by the number of exemplars with the same label as e. During learning, $\boldsymbol{\alpha}_e$ is non-zero for those exemplars that are in e's class and that should be similar to e, and zero for those that are considered irrelevant for exemplar e. The key idea behind these vectors is that even within a class, different segments can have very different feature appearance. This could depend, for example, on the angle from which an object is observed.

The values of $\boldsymbol{\alpha}_e$ and $\mathbf{w}_e$ are determined for each exemplar separately by the following optimization:

$$\{\mathbf{w}_e^*, \boldsymbol{\alpha}_e^*\} = \underset{\mathbf{w}_e, \boldsymbol{\alpha}_e}{\operatorname{argmin}} \sum_{i \in \mathcal{C}_e} \boldsymbol{\alpha}_{ei} L(-\mathbf{w}_e \cdot \mathbf{d}_{ei}) + \sum_{i \notin \mathcal{C}_e} L(\mathbf{w}_e \cdot \mathbf{d}_{ei})$$
$$\text{subject to } \mathbf{w}_e \geq 0;\ \boldsymbol{\alpha}_{ei} \in \{0, 1\};\ \sum_i \boldsymbol{\alpha}_{ei} = K \tag{2}$$

Here, $\mathcal{C}_e$ is the set of examplars that belong to the same class as e, $\boldsymbol{\alpha}_{ei}$ is the i-th component of $\boldsymbol{\alpha}_e$, and L is an arbitrary positive loss function. The constraints ensure that K values of $\boldsymbol{\alpha}_e$ are non-zero. Intuitively, this ensures that the optimization aims at maximizing the margin of a decision boundary that has K segments from e's class on one side, while keeping exemplars from other classes on the other side. The optimization procedure alternates between two steps. The $\boldsymbol{\alpha}_e$ vector in the k-th iteration is chosen such that it minimizes the first sum in (2):

$$\boldsymbol{\alpha}_e^k = \underset{\boldsymbol{\alpha}_e}{\operatorname{argmin}} \sum_{i \in \mathcal{C}_e} \boldsymbol{\alpha}_{ei} L(-\mathbf{w}_e^k \cdot \mathbf{d}_{ei}) \tag{3}$$

This is done by simply setting $\boldsymbol{\alpha}_e^k$ to 1 for the K smallest values of $L(-\mathbf{w}_e \cdot \mathbf{d}_{ei})$, and setting it to zero otherwise. The next step fixes $\boldsymbol{\alpha}_e$ to $\boldsymbol{\alpha}_e^k$ and optimizes (2) to yield the new $\mathbf{w}_e^{k+1}$:

$$\mathbf{w}_e^{k+1} = \underset{\mathbf{w}_e}{\operatorname{argmin}} \sum_{i:\in \mathcal{C}_e} \boldsymbol{\alpha}_{ei}^k L(-\mathbf{w}_e \cdot \mathbf{d}_{ei}) + \sum_{i \notin \mathcal{C}_e} L(\mathbf{w}_e \cdot \mathbf{d}_{ei}) \tag{4}$$

When choosing the loss function L to be the square hinge-loss function, this optimization yields standard Support Vector Machine learning. The iterative procedure converges when $\boldsymbol{\alpha}_e^k = \boldsymbol{\alpha}_e^{k+1}$.

Malisiewicz and Efros showed that the learned distance functions provide excellent recognition results for image segments [14].

III. Domain Adaptation

So far, the approach assumes that the exemplars in the training set $\mathcal{E}$ are drawn from the same distribution as the segments on which the approach will be applied. While this worked well for Malisiewicz and Efros, it does not perform well when training and test domain are significantly different. In our scenario, for example, the classification is applied to segments extracted from 3D laser scans, while most of the training data is extracted from the 3D-Warehouse data set. As we will show in the experimental results, combining training data from both domains can improve classification over just using data from either domain, but this performance gain cannot be achieved by simply combining data from the two domains into a single training set.

In general, we distinguish between two domains. The first one, the *target domain*, is the domain on which the classifier will be applied after training. The second domain, the *source domain*, differs from the target domain but provides additional data that can help to learn a good classifier for the target domain. In our context, the training data now consists of exemplars chosen from these two domains: $\mathcal{E} = \mathcal{E}^t \cup \mathcal{E}^s$.

Here, $\mathcal{E}^t$ contains exemplars from the target domain, that is, labeled segments extracted from the real laser data. $\mathcal{E}^s$ contains segments extracted from the 3D-Warehouse. As typical in domain adaptation, we assume that we have substantially more labeled data from the source domain than from the target domain: $|\mathcal{E}^s| \gg |\mathcal{E}^t|$. We now describe two methods of domain adaptation in the context of the exemplar-based learning technique.

A. Domain Adaptation via Feature Augmentation

Daume introduced feature augmentation as a general approach to domain adaptation [5]. It is extremely easy to implement and has been shown to outperform various other domain adaptation techniques and to perform as well as the thus far most successful approach to domain adaptation [6]. The approach performs adaptation by generating a stacked feature vector from the original features used by the underlying learning technique. Specifically, let $\mathbf{f}_e$ be the feature vector describing exemplar e. Daume's approach generates a stacked vector $\mathbf{f}_e^*$ as follows:

$$\mathbf{f}_e^* = \begin{pmatrix} \mathbf{f}_e \\ \mathbf{f}_e^s \\ \mathbf{f}_e^t \end{pmatrix} \tag{5}$$

Here, $\mathbf{f}_e^s = \mathbf{f}_e$ if e belongs to the source domain, and $\mathbf{f}_e^s = \mathbf{0}$ if it belongs to the target domain. Similarly, $\mathbf{f}_e^t = \mathbf{f}_e$ if e belongs to the target domain, and $\mathbf{f}_e^t = \mathbf{0}$ otherwise. Using the stacked feature vector, it becomes clear that exemplars from the same domain are automatically closer to each other in feature space than exemplars from different domains. Daume argued that this approach works well since data points from the target domain have more influence than source domain points when making predictions about test data.

B. Domain Adaption for Exemplar-based Learning

We now present a method for domain adaptation specifically designed for the exemplar-based learning approach. The key difference between our domain adaptation technique and the single domain approach described in Section II lies in the specification of the binary vector $\boldsymbol{\alpha}_e$. Instead of treating all exemplars in the class of e the same way, we distinguish between exemplars in the source and the target domain. Specifically, we use the binary vectors $\boldsymbol{\alpha}_e^s$ and $\boldsymbol{\alpha}_e^t$ for the exemplars in these two domains. The domain adaptation objective becomes

$$\begin{aligned}\{\mathbf{w}_e^*, \boldsymbol{\alpha}_e^{s*}, \boldsymbol{\alpha}_e^{t*}\} = \underset{\mathbf{w}_e, \boldsymbol{\alpha}_e^s, \boldsymbol{\alpha}_e^t}{\operatorname{argmin}} \\ \sum_{i \in \mathcal{C}_e^s} \boldsymbol{\alpha}_{ei}^s L(-\mathbf{w}_e \cdot \mathbf{d}_{ei}) + \sum_{i \in \mathcal{C}_e^t} \boldsymbol{\alpha}_{ei}^t L(-\mathbf{w}_e \cdot \mathbf{d}_{ei}) + \\ \sum_{i \notin \mathcal{C}_e} L(\mathbf{w}_e \cdot \mathbf{d}_{ei}) \ ,\end{aligned} \tag{6}$$

where $\mathcal{C}_e^s$ and $\mathcal{C}_e^t$ are the source and target domain exemplars with the same label as e. The constraints are virtually identical to those for the single domain objective (2), with the constraints on the vectors becoming $\sum_i \boldsymbol{\alpha}_{ei}^s = K^s$ and $\sum_i \boldsymbol{\alpha}_{ei}^t = K^t$. The values for K^s and K^t give the number of source and target exemplars that must be considered during the optimization.

The subtle difference between (6) and (2) has a substantial effect on the learned distance function. To see this, imagine the case where we train the distance function of an exemplar from the source domain. Naturally, this exemplar will be closer to source domain exemplars from the same class than to target domain exemplars from that class. In the extreme case, the vectors determined via (3) will contain 1s only for source domain exemplars, while they are zero for all target domain exemplars. The single domain training algorithm will thus not take target domain exemplars into account and learn distance functions for source domain exemplars that are good in classifying source domain data. There is no incentive to make them classify target domain exemplars well. By keeping two different α-vectors, we can force the algorithm to optimize for classification on the target domain as well. The values for K^s and K^t allow us to trade off the impact of target and source domain data. They are determined via grid search using cross-validation, where the values that maximize the area under the precision-recall curve are chosen.

The learning algorithm is extremely similar to the single domain algorithm. In the k-th iteration, optimization of the α-vectors is done by setting $\boldsymbol{\alpha}_e^{s\,k}$ and $\boldsymbol{\alpha}_e^{t\,k}$ to 1 for the exemplars yielding the K^s and K^t smallest loss values, respectively. Then, the weights $\mathbf{w}_e^{k+1}$ are determined via convex SVM optimization using the most recent α-vectors within (6).

IV. Probabilistic Classification

To determine the class of a new segment, z, Malisiewicz and Efros determine all exemplars e for which $\mathbf{d}_{ez} \leq 1$ and then choose the majority among the classes of these exemplars. However, this approach does not model the reliability of individual exemplars and does not lend itself naturally to a probabilistic interpretation. Furthermore, it does not take into account that the target domain is different from the source domain.

To overcome these limitations, we choose the following naïve Bayes model over exemplars. For each exemplar e and each segment class c we compute the probability p_{ec} that the distance $\mathbf{d}_{ez}$ between the exemplar and a segment from that class is less than 1:

$$p_{ec} \quad := \quad p(\mathbf{d}_{ez} \leq 1 \mid C(z) = c) \tag{7}$$

Here, $C(z)$ is the class of segment z. Since the ultimate goal is to label segments from the target domain only, we estimate this probability solely based on the labeled segments from the target domain. Specifically, p_{ec} is determined by counting all segments z in $\mathcal{E}^t$ that belong to class c *and* that are close to e, that is, for which $\mathbf{d}_{ez} \leq 1$. Normalization with the total number of target domain segments in class c gives the desired probability.

Assuming independence among the distances to all exemplars given the class of a segment z, the probability distribution

over z's class can now be computed as

$$p(C(z) = c \mid \mathcal{E}) \;\propto\; p(C(z) = c) \prod_{e \in \mathcal{E}, \mathbf{d}_{ez} \leq 1} p_{ec} \prod_{e \in \mathcal{E}, \mathbf{d}_{ez} > 1} (1 - p_{ec}) \tag{8}$$

where $p(C(z) = c)$ is estimated via class frequencies in the target domain data. We found experimentally that using eq. 8 as described, where it includes influence from both associated ($d_{ez} \leq 1$) and unassociated ($d_{ez} > 1$) exemplars, led to worse results than including just the associated exemplars. This is because there are many more unassociated exemplars than associated ones, and so they have undue influence over the probability. We instead compute the *positive support* using just the associated exemplars.

We can apply the results of segment classification to individual laser points. As described in Section II-A, we extract a soup of segments from a 3D laser scan. Thus, each laser point might be associated to multiple segments. Using the probability distributions over the classes of these segments (with *positive support* only), the distribution over the class of a single laser point l is given by

$$p(C(l) = c \mid \mathcal{E}) \;\propto\; p(C(z) = c) \prod_{z \in Z_l} \prod_{e \in \mathcal{E}, \mathbf{d}_{ez} \leq 1} p_{ec} \tag{9}$$

where Z_l is the set of segments associated with point l. In our setup, a test segment is assigned to the class with the highest probability.

V. Experimental Results

We evaluate different approaches to 3D laser scan classification based on real laser scans and objects collected from the Google 3D Warehouse. The task is to classify laser points into the following seven classes: cars, people, trees, street signs, fences, buildings, and background. Our experiments demonstrate that both domain adaptation methods lead to improvements over approaches without domain adaptation and alternatives including LogitBoost. In particular, our exemplar-based domain adaptation approach obtains the best performance.

A. Data Set

We evaluated our approach using models from Google 3D Warehouse as our source domain set, $\mathcal{E}^s$, and ten labeled scans of real street scenes as our target domain set, $\mathcal{E}^t$. The ten real scans, collected by a vehicle navigating through Boston, were chosen such that they did not overlap spatially. Labeling of these scans was done by inspecting camera data collected along with the laser data. We automatically downloaded the first 100 models of each of cars, people, trees, street signs, fences and buildings from Google 3D Warehouse and manually pruned out low quality models, leaving around 50 models for each class. We also included a number of models to serve as the background class, consisting of various other objects that commonly appear in street scenes, such as garbage cans, traffic barrels and fire hydrants. We generated 10 simulated laser scans from different viewpoints around each of the downloaded models, giving us a total of around 3200 exemplars in the source domain set. The ten labeled scans totaled to around 400 exemplars in the six actual object classes. We generate a "soup of segments" from these exemplars, using the data points in real scans not belonging to the six actual classes as candidates for additional background class exemplars. After this process, we obtain a total of 4,900 source domain segments and 2,400 target domain segments.

B. Comparison with Alternative Approaches

We compare the classification performance of our exemplar-based domain adaptation approach to several approaches, including training the single domain exemplar-based technique only on Warehouse exemplars, training it only on the real scans, and training it on a mix of Warehouse objects and labeled scans. The last combination can be viewed as a naïve form of domain adaptation. We also tested Daume's feature augmentation approach to domain adaptation. Our software is based on the implementation provided by Malisiewicz.

The optimal K values (length of the $\boldsymbol{\alpha}$ vectors) for each approach were determined separately using grid search and cross validation. Where training involves using real scans, we repeated each experiment 10 times using random train/test splits of the 10 total available scans. Each labeled scan contains around 240 segments on average.

The results are summarized in Fig. 4. Here the probabilistic classification described in Section IV was used and the precision-recall curves are generated by varying the probabilistic classification threshold between $[0.5, 1]$. The precision and recall values are calculated on a per-laser-point basis. Each curve corresponds to a different experimental setup. The left plot shows the approaches trained on five real laser scans, while the right plot shows the approaches trained on three real laser scans. All approaches are tested on real laser scans only. 3DW stands for exemplars from the 3D-Warehouse, and Real stands for exemplars extracted from real laser scans. Note that since the first setup (3DW) does not use real laser scans, the curves for this approach on the two plots are identical. Where exemplars from both the 3D-Warehouse and real scans are used, we also specify the domain adaptation technique used. By Simple we denote the naïve adaptation of only mixing real and Warehouse data. Stacked refers to Daume's stacked feature approach, applied to the single domain exemplar technique. Finally, Alpha is our technique.

It comes as no surprise that training on Warehouse exemplars only performs worst. This result confirms the fact that the two domains actually have rather different characteristics. For instance, the windshields of cars are invisible to the real laser scanner, thereby causing a large hole in the object segment. In Warehouse cars, however, the windshields are considered solid, causing a locally very different point cloud. Also, Warehouse models, created largely by casual hobbyists, tend to be composed of simple geometric primitives, while the shape of objects from real laser scans can be both more complex and more noisy.

Somewhat surprisingly, the naïve approach of training on a mix of both Warehouse and real scans leads to worse

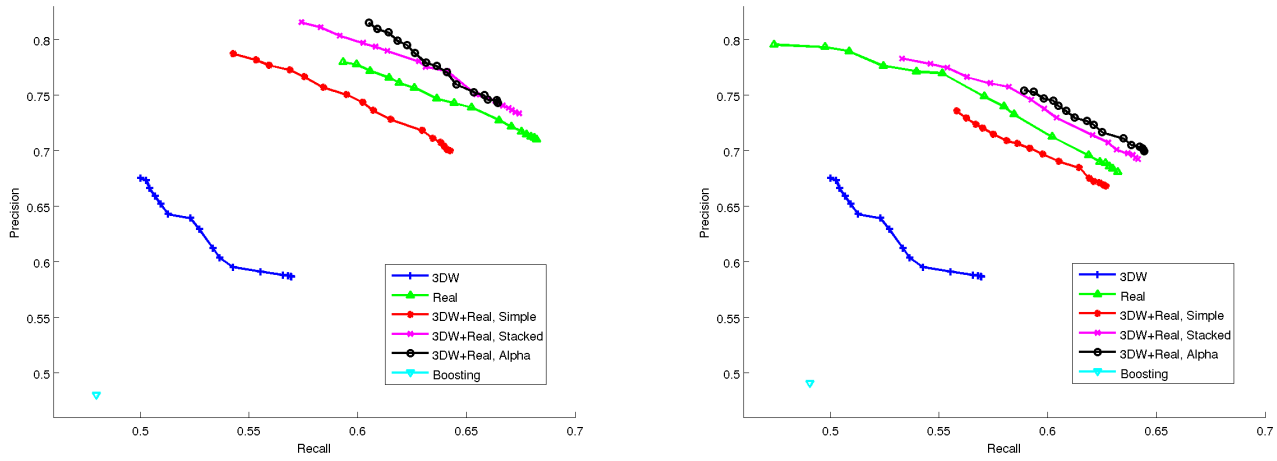

Fig. 4. Precision-recall curves comparing. (left) Performance of the various approaches, trained five real scans where applicable. (right) Performance of the various approaches, trained on three real scans where applicable.

performance than just training on real scans alone. This shows that domain adaptation is indeed necessary when incorporating training data from multiple domains. Both domain adaptation approaches outperform the approaches without domain adaptation. Our exemplar-based approach outperforms Daume's feature augmentation approach when target domain training data is very scarce (when trained with only 3 real scans).

To gauge the overall difficulty of the classification task, we also trained a LogitBoost [8] classifier on the mix of Warehouse and real scans. LogitBoost achieved a maximum F-score of 0.48 when trained on five scans, and a maximum F-score of 0.49 when trained on three scans (see Fig. 4). The F-score is the harmonic mean between precision and recall: $F = 2 \cdot Precision \cdot Recall/(Precision + Recall)$. As a comparison, our approach achieves an F-score of 0.70 when trained on five scans and 0.67 when trained on three scans. The inferior results achieved by LogitBoost demonstrate that this is not a trivial classification problem and that the exemplar-based approach is an extremely promising technique for 3D laser scan classification. Our approach has an overall accuracy of 0.57 for cars, 0.31 for people 0.55 for trees, 0.35 for street signs, 0.32 for fences and 0.73 for buildings.

C. Feature Selection and Thresholding Comparisons

To verify that all of the selected features contribute to the success of our approach, we also compared the performance of our approach using three different sets of features. We looked at using just bounding box dimensions and the minimum height off the ground (dimensions only), adding in the original, rotation-invariant Spin Image Signatures as described in [2] (Original Spin Signatures + dimensions), and adding in our $3 \times 3 \times 3$ grid of Spin Image Signatures (Grid Spin Signatures + dimensions). When trained on 3 scans using dimensions features only, our approach achieves a maximum F-score of 0.63. Using Original Spin Signatures + dimensions, we achieved an F-score of 0.64. Finally, using Grid Spin Signatures and dimensions achieved an F-score of 0.67. Due to noise and occlusions in the scans, as well as imperfect segmentation, the classes are not easily separable just based on dimensions. Also, our Grid Spin Image Signature features perform better than the original, rotation-invariant, Spin Image Signatures, justifying our modification to remove their rotation-invariance.

We also compared our probabilistic classification approach to the recognition confidence scoring method described by Malisiewicz in [14] and found that the precision-recall curves generated by probabilistic classification attain recalls between $30-50$ percentage points above recognition confidence scoring for corresponding precision values.

D. Examples

Fig. 5 provides examples of exemplars matched to the three laser segments shown in the panels in the left column. The top row gives ordered matches for the car segment on the left, the middle and bottom row show matches for a person and tree segment, respectively. As can be seen, the segments extracted from the real scans are successfully matched against segments from both domains, real and Warehouse. The person is mis-matched with one object from the background class "other" (second row, third column). Part of a laser scan and its ground truth labeling is shown in Fig. 6, along with the labeling achieved by our approach.

VI. Conclusion

The computer vision community has recently shown that using large sets of weakly labeled image data can help tremendously to deal with the complexity of the visual world. When trying to leverage large data sets to help classification tasks in robotics, one main obstacle is that data collected by a mobile robot typically has very different characteristics from data available on the World Wide Web, for example. For instance, our experiments show that simply adding Google 3D Warehouse objects when training 3D laser scan classifiers can *decrease* the accuracy of the resulting classifier.

In this paper we presented a domain adaptation approach that overcomes this problem. Our technique is based on an exemplar learning approach developed in the context of image-based classification [14]. We showed how this approach can be

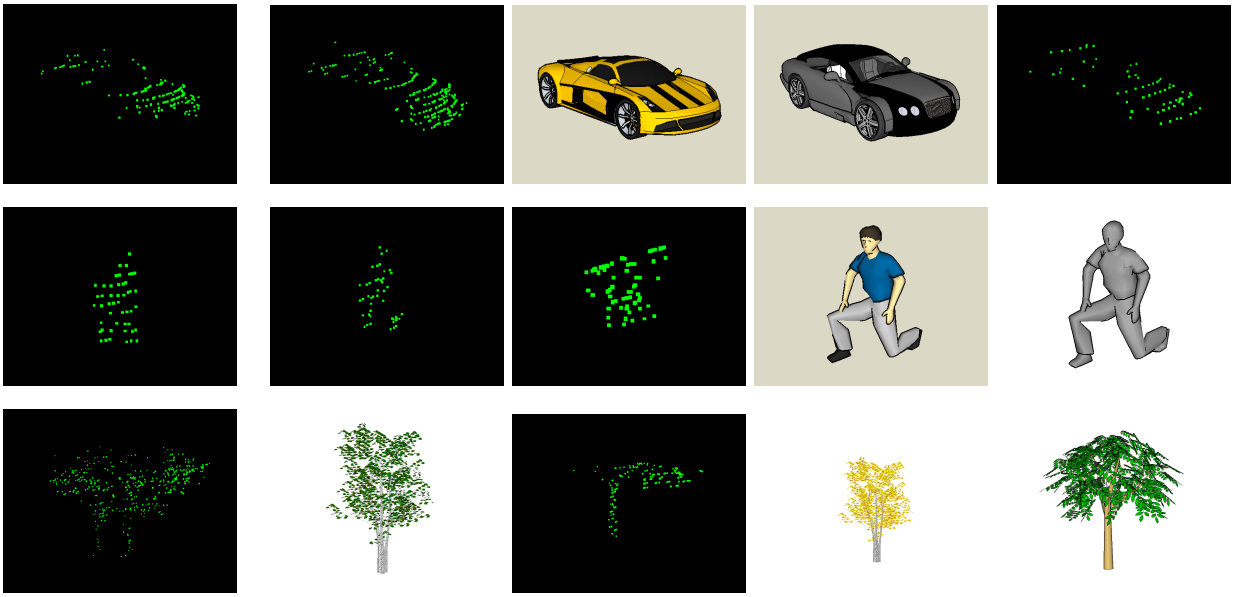

Fig. 5. Exemplar matches. The leftmost column shows example segments extracted from 3D laser scans: car, person, tree (top to bottom). Second to last columns show exemplars with distance below threshold, closer exemplars are further to the left.

applied to 3D laser scan data and be extended to the domain adaptation setting. For each laser scan, we generate a "soup of segments" in order to generate multiple possible segmentations of the scan. The experimental results show that our domain adaptation improves the classification accuracy of the original exemplar-based approach. Furthermore, our approach clearly outperformed a boosting technique trained on the same data.

There are several areas that warrant further research. First, we classified laser data solely based on shape. While adding other sensor modalities is conceptually straightforward, we believe that the accuracy of our approach can be greatly improved by adding visual information. Here, we might also be able to leverage additional data bases on the Web. We only distinguish between six main object classes and treat all other segments as belonging to a background class. Obviously, a realistic application requires us to add more classes, for example distinguishing different kinds of street signs. So far, we only used small sets of objects extracted from the 3D Warehouse. A key question will be how to incorporate many thousands of objects for both outdoor and indoor object classification. Finally, our current implementation is far from being real time. In particular, the scan segmentation and spin image feature generation take up large amounts of time. An efficient implementation and the choice of more efficient features will be a key part of future research. Despite all these shortcomings, however, we believe that this work is a promising first step toward robust many-class object recognition for mobile robots.

Acknowledgments

We would like to thank Michael Beetz for the initial idea to use 3D-Warehouse data for object classification, and Albert Huang for providing us with the urban driving data set. This work was supported in part by a ONR MURI grant number N00014-07-1-0749, by the National Science Foundation under Grant No. 0812671, and by a postgraduate scholarship from the Natural Sciences and Engineering Research Council of Canada. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] 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 *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[2] J. Assfalg, M. Bertini, A. Del Bimbo, and P. Pala. Content-based retrieval of 3-D objects using spin image signatures. *IEEE Transactions on Multimedia*, 9(3), 2007.

[3] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(5), 2002.

[4] W. Dai, Q. Yang, G. Xue, and Y. Yu. Boosting for transfer learning. In *Proc. of the International Conference on Machine Learning (ICML)*, 2007.

[5] H. Daumé. Frustratingly easy domain adaptation. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2007.

[6] H. Daumé and D. Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research (JAIR)*, 26, 2006.

[7] B. Douillard, D. Fox, and F. Ramos. Laser and vision based outdoor object mapping. In *Proc. of Robotics: Science and Systems (RSS)*, 2008.

[8] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 28(2), 2000.

[9] Google. 3d warehouse. http://sketchup.google.com/3dwarehouse/.

[10] J. Hays and A. Efros. Scene completion using millions of photographs. *ACM Transactions on Graphics (Proc. of SIGGRAPH)*, 26(3), 2007.

[11] J. Jiang and C. Zhai. Instance weighting for domain adaptation in NLP. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2007.

[12] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 21(5), 1999.

[13] T. Malisiewicz and A. Efros. Improving spatial support for objects via multiple segmentations. In *Proc. of the British Machine Vision Conference*, 2007.

[14] T. Malisiewicz and A. Efros. Recognition by association via learning per-examplar distances. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

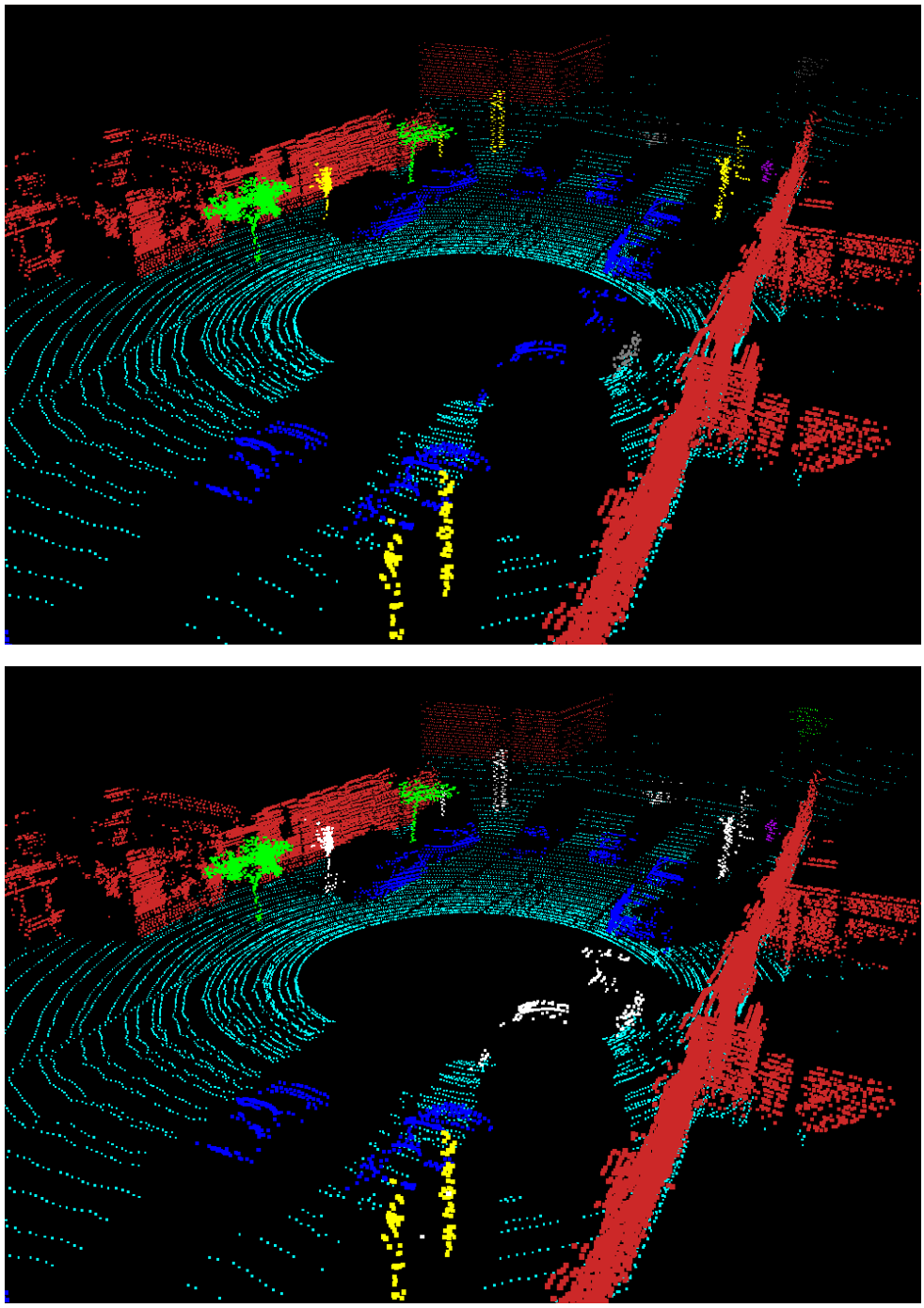

Fig. 6. (top) Ground truth classification for part of a 3D laser scan. Colors indicate ground plane (cyan) and object types (green: tree, blue: car, yellow: street sign, purple: person, red: building, grey: other, white: not classified). (bottom) Classification achieved by our approach. As can be seen, most of the objects are classified correctly. The street signs in the back and the car near the center are not labeled since they are not close enough to any exemplar.

[15] I. Posner, M. Cummins, and P. Newman. Fast probabilistic labeling of city maps. In *Proc. of Robotics: Science and Systems (RSS)*, 2008.

[16] B. Russell, K. Torralba, A. Murphy, and W. Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3), 2008.

[17] B. Sapp, A. Saxena, and A. Ng. A fast data collection and augmentation procedure for object recognition. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2008.

[18] A. Saxena, J. Driemeyer, and A. Ng. Robotic grasping of novel objects using vision. *International Journal of Robotics Research*, 27(2), 2008.

[19] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(8), 2000.

[20] A. Torralba, R. Fergus, and W. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(11), 2008.

[21] R. Triebel, R. Schmidt, O. Martinez Mozos, and W. Burgard. Instance-based amn classification for improved object recognition in 2d and 3d laser range data. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

[22] C. Wellington, A. Courville, and T. Stentz. Interacting Markov random fields for simultaneous terrain modeling and obstacle detection. In *Proc. of Robotics: Science and Systems (RSS)*, 2005.