Sparse Distance Learning for Object Recognition Combining RGB and Depth Information

Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox

***Abstract*— In this work we address joint object category and instance recognition in the context of RGB-D (depth) cameras. Motivated by local distance learning, where a novel view of an object is compared to individual views of previously seen objects, we define a view-to-object distance where a novel view is compared simultaneously to *all* views of a previous object. This novel distance is based on a weighted combination of feature differences between views. We show, through jointly learning per-view weights, that this measure leads to superior classification performance on object category and instance recognition. More importantly, the proposed distance allows us to find a sparse solution via Group-Lasso regularization, where a small subset of representative views of an object is identified and used, with the rest discarded. This significantly reduces computational cost without compromising recognition accuracy. We evaluate the proposed technique, Instance Distance Learning (IDL), on the RGB-D Object Dataset, which consists of 300 object instances in 51 everyday categories and about 250,000 views of objects with both RGB color and depth. We empirically compare IDL to several alternative state-of-the-art approaches and also validate the use of visual and shape cues and their combination.**

I. Introduction

Visual recognition of objects is a fundamental and challenging problem and a major focus of research for computer vision, machine learning, and robotics. In the past decade, a variety of features and algorithms have been proposed and applied to this problem, resulting in significant progress in object recognition capabilities, as can be seen from the steady improvements on standard benchmarks such as Caltech101 [7].

The goal of our work is to study the recognition problem at both the category and the instance level, on objects that we commonly use in everyday tasks. Category level recognition involves classifying objects as belonging to some category, such as coffee mug or soda can. Instance level recognition is identifying whether an object is physically the same object as one that has previously been seen. Most recognition benchmarks are constructed using Internet photos at the category level only, but the ability to recognize objects at both levels is crucially important if we want to use such recognition systems in the context of specific tasks, such as human activity recognition or service robotics. For example, identifying an object as a generic "coffee mug" (category) or as "Amelia's coffee mug" (instance) can lead to substantially different implications depending on the context of a task. In this paper we use the term *instance* to refer to a single object.

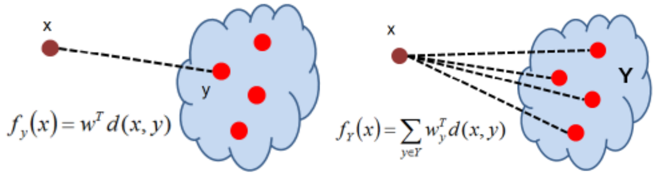

Fig. 1. Two distance learning approaches. (Left) Local distance learning uses a view-to-view distance, typically followed by a k-nearest neighbor rule. (Right) The proposed instance distance learning, where we use the weighted average distance from a view x to an object instance $\mathbf{Y}$ which consists of a set of views of the same object.

In addition to category and instance level recognition, we want to enrich the recognition data by taking advantage of recent advances in sensing hardware. In particular, the rapidly maturing technologies of RGB-D (Kinect-style) depth cameras [20], [13] provide high quality synchronized videos of both color and depth, presenting a great opportunity for combining color- and depth-based recognition. To take advantage of this rich new data in object recognition, the classifier needs to combine visual and shape information. However, not all features are always useful. Some features may be more discriminative for certain objects, while other features are more useful for other objects. The best features to use may also depend on the task at hand, for example whether we are trying to find "Amelia's coffee mug" or just any coffee mug (category versus instance recognition). The recognition system should learn which features are useful depending on the particular object and task at hand.

One successful line of work on combining heterogeneous features is distance learning (e.g. [27], [26]), in particular local distance learning [23]. Local distance learning has been extensively studied and demonstrated for object recognition, both for color images [9], [10], [18] and 3D shapes [15]. A key property of these approaches is that they can model complex decision boundaries by combining elementary distances. Local distance learning, however, is not without issues. For our problem setting, there are two main limitations to overcome: (1) existing formulations of local distance learning do not capture the relations between object categories and specific instances under them; (2) they provide no means for selecting representative views, or example images, of

This work was funded in part by an Intel grant, by ONR MURI grants N00014-07-1-0749 and N00014-09-1-1052, by the NSF under contract IIS-0812671, and through the Robotics Consortium sponsored by the U.S. Army Research Laboratory under Cooperative Agreement W911NF-10-2-0016.

Kevin Lai and Liefeng Bo are with the Department of Computer Science & Engineering, University of Washington, Seattle, WA 98195, USA. {kevinlai,lfb}@cs.washington.edu

Xiaofeng Ren is with Intel Labs Seattle, Seattle, WA 98105, USA. xiaofeng.ren@intel.com

Dieter Fox is with both the Department of Computer Science & Engineering, University of Washington, and Intel Labs Seattle. fox@cs.washington.edu

instances and thus become very inefficient if a large number of views are collected for each object.

The explosive growth of the web has led to the availability of large repositories of images like Flickr and 3D models like Google 3D Warehouse. The computer vision community has recently released ImageNet [5], a growing database of millions of images organized according to WordNet hypernym-hyponym relations. Although these large databases contain a wealth of information that can potentially be used to solve robot perception problems, it remains difficult to create algorithms that can take advantage of these large datasets while still retaining the efficiency required for robotics applications.

In this paper we propose an approach to sparse **I**nstance **D**istance **L**earning (IDL): instead of learning per-view distances, we define and optimize a *per-instance distance* that combines all views of an object instance (see Fig. 1). By learning a distance function jointly for all views of a particular object, our approach significantly outperforms view-based distance learning for RGB, Depth, and RGB+Depth recognition. This result can also be motivated as subclass classification [25], [6]. Even more importantly, joint instance distance learning naturally leads to a sparse solution using Group-Lasso regularization, where a sparse set of views of each instance is selected from a large pool of views. Thus, IDL provides a data-driven way to select informative training examples for each object and significantly sparsify the data set, discarding redundant views and speeding up classification. We show that IDL achieves sparse solutions without any decrease in performance.

II. Learning Instance Distances

In this section, we describe how to learn instance distance functions for classification tasks in the context of image classification. In image classification, we are given a set of objects Y. The goal is to learn a classifier to predict category and instance labels of images, or views, outside the training set. One of the simplest methods to do this is to find nearest neighbors of the test view and make a prediction based on the labels of these nearest neighbors. In this section, we show how to improve this approach by learning an instance distance function. We start by considering a simple classification rule, the *nearest instance classifier*, which labels incoming test images x using the label of the nearest instance (an extension to k-nearest instances is straightforward):

$$c_x = \operatorname*{argmin}_{i,j} \frac{1}{|Y_{ij}|} \sum_{y \in Y_{ij}} d(x,y) \tag{1}$$

Here, Y_{ij} denotes the set of views taken of the j-th instance of the i-th category. As can be seen, c_x is the object that appears most similar to the test image, averaged over its views. $d(x,y)$ can be any distance function between views x and y. In this paper, we use the l_2 distance $d(x,y) = \|x-y\|$. The nearest instance classifier given in (1) can be used for both category and instance recognition: The index i provides the category and the index j gives the corresponding instance. Unfortunately, the nearest instance classifier can often perform poorly in practice due to the difficulties of finding a good distance measure.

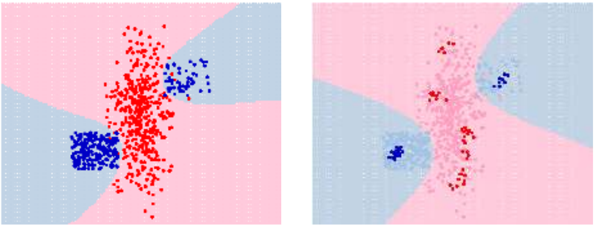

Fig. 2. Decision boundaries found by two instance distance classifiers on a two-dimensional dataset. (Left) instance distance learning with l_2 regularization. (Right) instance distance learning with data sparsification, which retains only 8% of data (stronger colors) and still has a similar decision boundary.

We now consider a significantly more powerful variant by learning an instance distance function for recognition. In many problems there are multiple features available and the best performance is obtained by using all available information. To do so, we replace the scalar distance $d(x,y)$ between two views x and y by a vector $\mathbf{d}(x,y)$ of separate l_2 feature distances. The corresponding instance distance function between example x and the j-th instance of i-th category Y_{ij} can then be written as

$$f_{ij}(x) = \frac{1}{|Y_{ij}|} \sum_{y \in Y_{ij}} \mathbf{w}_y^\top \mathbf{d}(x,y) + b_{ij} \tag{2}$$

where $\mathbf{W}$ is a set of weight vectors $\mathbf{w}_y$ for all $y \in Y_{ij}$. Unlike the *nearest instance classifier*, this significantly more expressive distance function allows the classifier to assign different weights to each feature and for each view, enabling it to adapt to the data. Note that we have added a bias term, b_{ij}, to the instance distance function to allow negative values. The weight vector $\mathbf{w}_y$ is D-dimensional, where D is the number of different features extracted for each view. Note also that each example view has a different weight vector. Due to this, the functions do not define a true distance metric, as they are asymmetric. This is advantageous since different examples may have different sets of features that are better for distinguishing them from other examples, or views.

When learning the weight vector for an instance, it is necessary to distinguish between category and instance classification. For *instance recognition*, the weight $\mathbf{W}_{ij}$ defining the distance function for the j-th instance in category i can be learned using the following l_2 regularized loss function:

$$\sum_{x \in Y_{ij}} L(-f_{ij}(x)) + \sum_{x \in Y \setminus Y_{ij}} L(f_{ij}(x)) + \lambda \mathbf{W}_{ij}^\top \mathbf{W}_{ij} \tag{3}$$

where we have chosen $L(z) = \max(0, 1-z)^2$, the squared hinge loss. The first term penalizes misclassification of views $x \in Y_{ij}$ that belong to the same instance. The second term similarly penalizes misclassification of negative examples, or views, by incurring a loss when their distance is small. Note that the negative examples also include views of different instances that belong to the same category i. The final term is a standard l_2 regularizer, biasing the system to learn smaller weight vectors. This objective function is convex and can be optimized using standard optimization algorithms. Given a test image x, we assign to it the category or instance label of the nearest object using $c_x = \mathrm{argmin}_{i,j} f_{ij}(x)$.

For *category recognition*, we learn the instance distance by minimizing the following l_2 regularized loss:

$$\sum_{x \in Y_i} L(-f_{ij}(x)) + \sum_{x \in Y \setminus Y_i} L(f_{ij}(x)) + \lambda \mathbf{W}_{ij}^{\top} \mathbf{W}_{ij} \qquad (4)$$

where $Y_i = \bigcup_{s=1}^{N_i} Y_{is}$ and N_i is the number of instances in the $i-th$ category. The key difference between the instance recognition and the category recognition loss is that in the former, only the views of the same instance are positive examples, whereas in the latter the views of *all* instances in the same category become positive examples.

Fig. 2 (left) shows the decision boundary obtained with instance distance learning on a two-dimensional dataset. The dataset contains two classes: red and blue. There are two separate instances in the blue class and they lie on opposite sides of the single red class instance. Instance distance learning is able to find a very good decision boundary separating the two classes.

III. Example Selection via Group-Lasso

An important property of the instance distance we defined in Section II is that it allows for data sparsification. This is achieved by replacing l_2 regularization in (3) with Group-Lasso [28], [19], resulting in the following objective function:

$$\sum_{x \in Y_{ij}} L(-f_{ij}(x)) + \sum_{x \in Y \setminus Y_{ij}} L(f_{ij}(x)) + \lambda \sum_{y \in Y_{ij}} \sqrt{\mathbf{w}_y^{\top} \mathbf{w}_y} \qquad (5)$$

Here, the first two terms optimize over individual components of the instance weight vector, and the third, Group-Lasso, term drives the weight vectors of individual views toward zero. Group-Lasso achieves this by grouping the weight components of individual views in the penalty term. In contrast to previous work that make use of the Group-Lasso for encouraging feature sparsity, here we use it to encourage data sparsity. In other words, optimizing this objective function yields a supervised method for choosing a subset of representative examples, or views. If the Group-Lasso drives an entire weight vector $\mathbf{w}_y$ to $\mathbf{0}$, the corresponding example no longer affects the decision boundary and has effectively been removed by the optimization. The degree of sparsity can be tuned by varying the λ parameter. Intuitively, data sparsity is often possible because many examples may lie well within the decision region or are densely packed together. Removing such examples would reduce the magnitude of the regularization term while having little or no effect on the loss terms. Each data point is only one of many that contribute to the instance distance and redundant examples would not significantly influence the decision boundary.

The advantage of data sparsification using the proposed objective function is twofold. As explained above, the proposed technique can remove redundant and uninformative examples. Secondly, removing examples from consideration at test time results in computational cost savings which counteracts the data-size-dependent time complexity of nearest neighbor techniques.

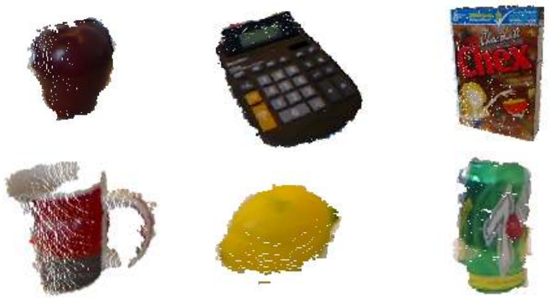

Fig. 3. Views of objects from the RGB-D Object Dataset shown as 3D point clouds colored with RGB pixel values. From left to right, top to bottom, they are apple, calculator, cereal box, coffee mug, lemon, and soda can.

For category level, the group lasso based instance distance learning uses the following objective function

$$\sum_{x \in Y_i} L(-f_{ij}(x)) + \sum_{x \in Y \setminus Y_i} L(f_{ij}(x)) + \lambda \sum_{y \in Y_{ij}} \sqrt{\mathbf{w}_y^{\top} \mathbf{w}_y} \qquad (6)$$

Fig. 2 (right) shows a data sparsification example using instance distance learning with Group-Lasso. In this two-dimensional dataset, the technique is able to throw away 92% of the data and still obtain decision boundaries that closely match the one learned without data sparsification.

IV. Experiments

We apply the proposed instance distance learning (IDL) to two related object recognition tasks: category recognition and instance recognition. In category recognition, the system is trained on several objects belonging to each category and the task is to classify a never-before-seen object into one of the categories. In the instance recognition task, the system is presented with multiple views of each object, and the task is to classify never-before-seen views of these same objects. The experimental results in this section demonstrate that our technique obtains good performance on both recognition tasks, particularly when taking full advantage of both shape and visual information available from the sensor. The technique is able to not only automatically sparsify training data, but it also exceeds the performance of several alternative approaches and baselines, even after sparsification. We also apply the instance distance learning technique to object detection and show that it is able to detect objects in a cluttered scene.

A. Experimental Setup

We evaluate our technique on the RGB-D Object Dataset [14], a novel dataset consisting of cropped and segmented images of distinct objects spun around on a turntable. The dataset consists of 300 object instances in 51 categories. There are between three to twelve instances in each category. The images are collected with an RGB-D camera that can simultaneously record both color image and depth data at 640×480 resolution. In other words, each 'pixel' in the

RGB-D frame contains four channels: red, green, blue and depth. The 3D location of each pixel in physical space can be computed using known sensor parameters. Each object was placed on the turntable and rotated. Data was recorded from three viewing heights, at approximately 30, 45 and 60 degrees above the horizon. We used around 50 views at each height, giving around 150 views per instance, or 45000 RGB + Depth images in total, each of which serves as a data point in training or testing. Fig. 3 shows some example views of objects from the data set. Each view shown here is a 3D point cloud where the points have been colored with their corresponding RGB pixel values. The segmentation procedure uses a combination of visual and depth cues and is described in detail in [14].

We extract features that capture both the visual appearance and shape of each view (image) of a particular object. The presence of synchronized visual and 3D data greatly enhances the amount of information available for performing object recognition and our technique naturally combines multiple features in a single framework. We first compute spin images [12] for a randomly subsampled set of 3D points. Each spin image is centered on a 3D point and captures the spatial distribution of points within its neighborhood. The distribution, captured in a two-dimensional 16×16 histogram, is invariant to rotation about the point normal. We use these spin images to compute efficient match kernel (EMK) features using random fourier sets as proposed in [2]. EMK features are similar to bag-of-words (BOW) features in that they both take a set of local features and generate a fixed length feature vector describing the bag. EMK features approximate the Gaussian kernel between local features and give a continuous measure of similarity. To incorporate spatial information, we divide an axis-aligned bounding cube around each view into a $3 \times 3 \times 3$ grid. We compute a 1000-dimensional EMK feature in each of the 27 cells separately. We perform principal component analysis (PCA) on the EMK features in each cell and take the first 100 components. Finally, we include as shape features the width, depth and height of a 3D bounding box around the view. This gives us a total of 30 shape descriptors.

To capture the visual appearance of a view, we extract SIFT [17] on a dense grid of 8×8 cells. To generate image-level features and capture spatial information we compute EMK features on two image scales. First we compute a 1000-dimensional EMK feature using SIFT descriptors from the entire image. Then we divide the image into a 2×2 grid and compute EMK features separately in each cell from only the SIFT features inside the cell. We perform PCA on each cell and take the first 300 components, giving a 1500-dimensional EMK SIFT feature vector. Additionally, we extract texton histograms [16] features, which capture texture information using oriented gaussian filter responses. The texton vocabulary is built from an independent set of images on LabelMe. We also include a color histogram and also use the mean and standard deviation of each color channel as visual features. There are a total of 13 visual descriptors.

B. Performance Comparisons

Given the above set of features, we evaluate the category and instance recognition performance of the proposed instance distance learning technique and compare it to a number of alternative state-of-the-art classifiers:

- IDL: Our proposed instance distance learning algorithm with l_2 regularization.
- EB LOCAL: An exemplar-based local distance function learning technique by Malisiewicz et al. [18].
- SVM: linear support vector machine
- RF: random forest classifier [3]

We follow the experimental setup in [14] to allow for direct comparisons. For category recognition, we randomly leave one object out from each category for testing and train the classifier on all views of the remaining objects. For instance recognition, we divide each video into 3 consecutive sequences of equal length and for each object instance. There are 3 heights (videos) for each object, so this gives 9 video sequences for each instance. We randomly select 7 of these for training and test on the remaining 2.

To verify that our technique is indeed able to take advantage of both shape and visual information available from the RGB-D camera, we evaluated the performance of all the techniques using only shape-based features, only visual-based feature, and using both shape and visual features. Fig. 4 shows the overall classification performance of the different algorithms on both category-level and instance-level recognition. As can be seen from the results, our technique substantially improves upon the performance of a competitive exemplar-based local distance method and other state-of-the-art classification techniques in most cases or otherwise gets comparable performance.

Overall, visual features are more useful than shape features for both category level and instance level recognition. However, shape features are relatively more useful in category recognition, while visual features are relatively more effective in instance recognition. This is exactly what we should expect, since a particular object instance has a fairly constant visual appearance across views, while objects in the same category can have different texture and color. On the other hand, shape tends to be stable across a category in many cases, thereby making instance recognition via shape more difficult. The fact that combining both shape and visual features enables our technique to perform better on both tasks demonstrates that our technique can take advantage of both shape and visual features.

C. Data Sparsification Results

Fig. 6 shows the classification accuracy of two data sparsification techniques at varying levels of data sparsity: 1) running instance distance learning technique on a uniform random downsampling of the training data and 2) our sparse instance distance learning (IDL SPARSE). The curve for IDL SPARSE is generated by varying the regularization tradeoff parameter, λ. The plot shows that IDL SPARSE is able to sparsify the data considerably (up to a factor of

Technique	Classification Accuracy					
	Category			Instance		
	Shape	Vision	All	Shape	Vision	All
EBLocal	58.9 ± 2.1	70.1 ± 3.4	78.4 ± 2.8	41.2 ± 0.6	81.2 ± 0.6	84.5 ± 0.5
LinSVM	53.1 ± 1.7	74.3 ± 3.3	81.9 ± 2.8	32.4 ± 0.5	**90.9 ± 0.5**	90.2 ± 0.6
RF	66.8 ± 2.5	74.7 ± 3.6	79.6 ± 4.0	52.7 ± 1.0	90.1 ± 0.8	90.5 ± 0.4
IDL	**70.2 ± 2.0**	**78.6 ± 3.1**	**85.4 ± 3.2**	**54.8 ± 0.6**	89.8 ± 0.2	**91.3 ± 0.3**

Fig. 4. Classification performance of various techniques on the RGB-D data set. EBLocal is exemplar-based local distance learning, LinSVM is linear SVM, RF is Random Forest, and IDL is the instance distance learning proposed in this paper.

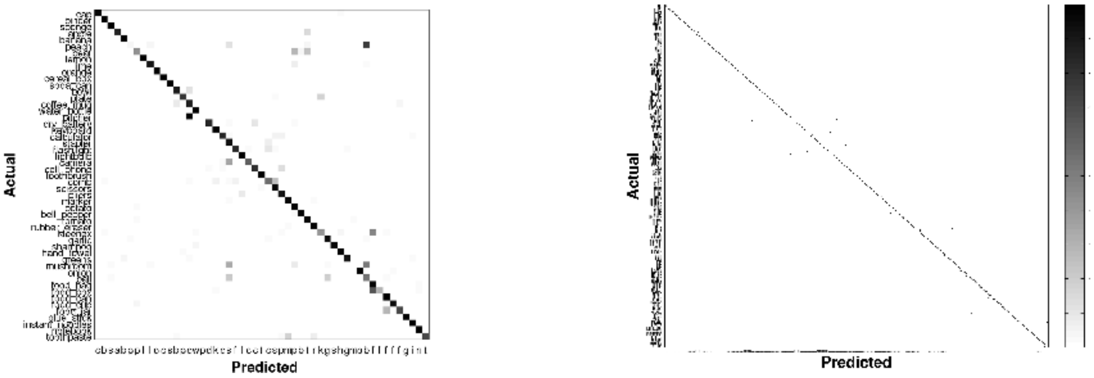

Fig. 5. Confusion matrices (row-normalized) for *sparse instance distance learning* on (left) category recognition and (right) instance recognition.

$\frac{1}{5}$) without causing any significant loss in accuracy. Note that IDL SPARSE does not necessarily converge to the same accuracy as IDL because the techniques use different regularization. The two techniques are only identical when the regularization tradeoff parameter is set to 0, but this would lead to overfitting.

Although uniform random downsampling is a naïve form of sparsification, it actually works very well on our dataset, since uniform sampling of video frames gives good coverage of object views. Nevertheless, the plot clearly shows that IDL SPARSE obtains higher classification accuracy than random downsampling across all levels of data sparsity. Fig. 7 shows some example views retained for several objects.

Fig. 5 shows the confusion matrices between the 51 categories for category recognition (left) and the 300 instances for instance recognition (right). In the category recognition run, the sparse instance distance learning obtained an overall accuracy of 83% and retained 15% of the training data. In the instance recognition run, the technique obtained an overall accuracy of 89.7% and retained 19% of the training data.

D. 3D Object Category Dataset

In addition to the novel RGB-D dataset that we collected, we also evaluated instance distance learning (IDL) on a publicly available image-only dataset: the 3D object category dataset presented by Savarese et al. [22]. There are 8 object categories in the dataset: bike, shoe, car, iron, mouse, cellphone, stapler, and toaster. For each category, the dataset contains images of 10 individual object instances under 8 viewing angles, 3 heights and 3 scales for a total number of 7000 images that are all roughly 400×300 pixels. We evaluated IDL on category level recognition on

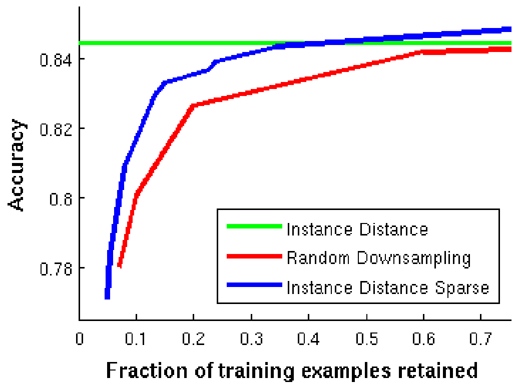

Fig. 6. Number of examples retained versus classification accuracy of two example selection techniques: 1) random downsampling and 2) sparse instance distance learning. Accuracy of Instance distance learning is shown for comparison.

this dataset using the same setup as [22]: we randomly select 7 instances per category for training and use the rest for testing. The furthest scale is not considered for testing. IDL obtains substantially higher accuracy (80.1%) than the results reported in [22] (75.7%).

E. Object Detection

Object recognition is often more than just classifying a cropped image of an object. For example, a robot may be tasked to search the environment for a specific set of objects, such as finding all coffee mugs and soda cans on a table. This problem is referred to as object detection. In object detection, the system is given a fixed set of objects to search for and trains the appropriate detectors beforehand. At test time, the system is presented with a set of images

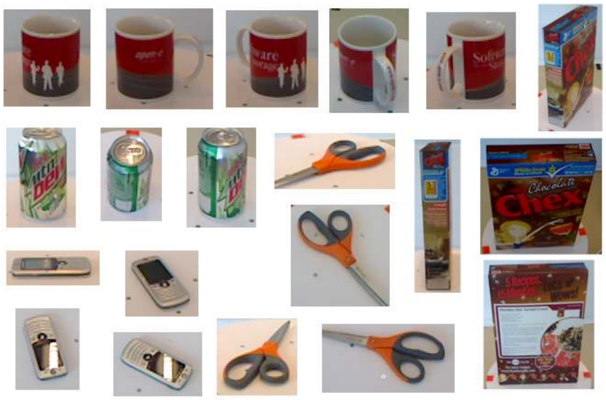

Fig. 7. Data selection with Group-Lasso: A small set of representative views that were chosen for several objects.

and must identify all objects of interest that are present in the image by specifying bounding boxes around them. We applied the instance distance learning technique to object detection. Given the task of identifying a particular set of objects, an instance distance classifier is trained for each instance by using views in the particular instance as positive examples. The set of negative examples is constructed from views of other objects as well as a separate set of background images that do not contain any objects the system is tasked to find. At test time, the system is presented with a video sequence taken from a particular scene, e.g. a kitchen area or an office table. The system runs a sliding window detector of a fixed size across each video frame, invoking the learned instance distance classifier at each window. The window sliding is done over an image pyramid to search across scales. The classifier returns a score, which we threshold to obtain bounding boxes. Since the distance from the camera to the object can vary, the size of an object in the image can also vary, so we run the sliding window detector on 20 image scales by rescaling the image. We perform non-maximum suppression to remove multiple overlapping detections.

The features we use for object detection differ from those used for recognition. This is because state-of-the-art object detection systems [4], [8] have shown histogram of oriented gradients (HOG) features to be effective and also because they can be efficiently computed on image windows using convolution. We divide each image into a grid of 8×8 cells and extract features in each cell. As visual features, we use a variation of HOG [8] computed on the RGB image. This version considers both contrast sensitive and insensitive features, where the gradient orientations in each cell (8×8 pixel grid) are encoded using two different quantization levels into 18 ($0° - 360°$) and 9 orientation bins ($0° - 180°$), respectively. This $4 \times (18 + 9) = 108$-dimensional feature vector is analytically projected into a 31-dimensional feature as described in [8].

As depth features, we compute HOG features over the depth image (i.e. treating the depth image as a regular image and computing histograms of oriented gradients on it). Additionally, we also compute a feature capturing the scale (physical size) of the object. The distance d of an object from the camera is inversely proportional to its scale, o. For an image at a particular scale s, we have $c = \frac{o}{s}d$, where c is constant. In the sliding window approach the detector window is fixed, meaning that o is fixed. Hence, $\frac{d}{s}$, which we call the normalized depth, is constant. We compute the average normalized depth in 8×8 grid and use this as a scale feature.

We evaluated the IDL classifier on the object detection task on a video sequence of an office environment. Objects were placed on a table and the system was tasked with finding the soda can, coffee mug, and cap in the video sequence. The cereal box acts as a distractor object and sometimes occludes the objects of interest. Following the PASCAL VOC evaluation metric, a candidate detection is considered correct if the intersection of the predicted bounding box and the ground truth bounding box is more than half of their union. Only one of multiple successful detections for the same ground truth is considered correct and the rest are counted as false positives. Fig. 8 (left) shows the precision-recall curves of the individual object detectors as well as the overall precision-recall curve for all the objects. Each precision-recall curve is generated by ranking the resulting detections using scores returned by the classifier and thresholding on them. Each threshold gives a point along the curve. We run only the detector for the particular object to generate the precision-recall curves for the individual objects. For the multiple-object curve, we run all three object detectors and pool all candidate detections across objects and generate a single precision-recall curve. The precision-recall curves show that IDL attains good performance on the object detection task. Even when searching for three different objects by running multiple detectors in the video sequence, there is only a slight drop in the precision and recall. Fig. 8 (right) shows an example multi-object detection. Here the system is able to correctly locate the three objects even though there are other objects and background clutter in the scene.

V. Conclusions

In this work we studied both object category and instance recognition using the RGB-D Object Dataset [14], a large RGB-D (color+depth) dataset of everyday objects. Our work is of interest both in terms of algorithm design and of the empirical validations on appearance and depth cues for recognition. Our key insight is that because a category consists of different objects, there is a natural division of a category into subclasses, and this motivates our use of the instance distance. We show that by jointly learning the weights in this distance function, we outperform alternative state-of-the-art approaches. The proposed instance distance learning provides a distance measure for evaluating the similarity of a view to a known set of objects. This information can be used as input to other robotics tasks, such as grasping. An interesting direction for future work is to treat the training data as an object database where grasping information is stored for each object. When the robot encounters an object in the world, it can use the instance distance classifier

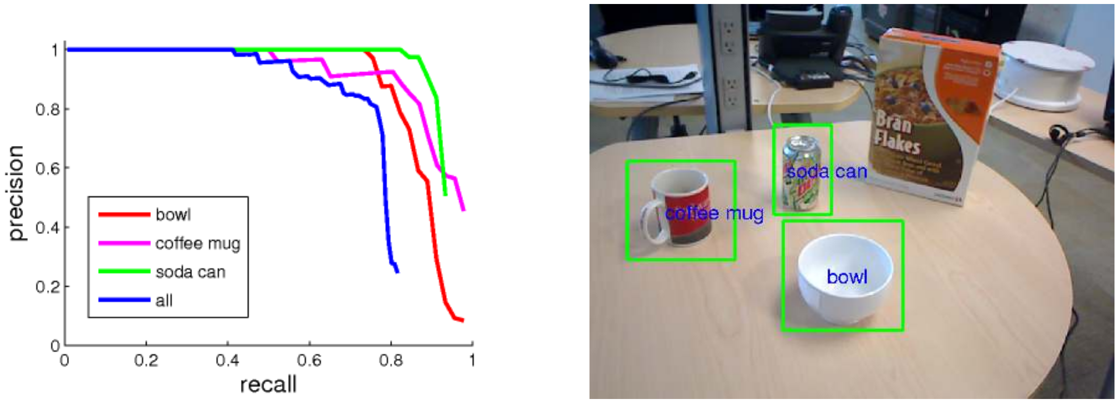

Fig. 8. Object detection results on a video sequence. (Left) Precision-recall curves of individual bowl, coffee mug, and soda can detectors and aggregated detections. (Right) Example video frame with detection results.

to match the object to objects in the database to retrieve potential grasps.

The use of Group-Lasso allows us to find a compact representation of each object instance as a small set of views without compromising accuracy. With the ever increasing size of data sets available on the World Wide Web, *sparsification* of such data will become more important. While the current technique assumes an offline setting, the development of online Group-Lasso style sparsification is an interesting and promising direction for future work.

Finally, we showed that using both shape and visual features achieves higher performance than either set of cues alone for both category and instance recognition. Considering the fast advances of RGB-D camera hardware, these results are extremely encouraging, supporting the belief that the combination of RGB and depth will find many uses in object recognition, detection, and other robotics perception tasks.

References

[1] F.R. Bach, G.R.G. Lanckriet, and M.I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the twenty-first international conference on Machine learning*, page 6. ACM, 2004.

[2] L. Bo and C. Sminchisescu. Efficient Match Kernel between Sets of Features for Visual Recognition. In *Advances in Neural Information Processing Systems*, December 2009.

[3] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[5] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-fei. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[6] Sergio Escalera, David M. J. Tax, Oriol Pujol, Petia Radeva, and Robert P. W. Duin. Subclass problem-dependent design for error-correcting output codes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(6):1041–1054, 2008.

[7] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(4):594–611, 2006.

[8] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[9] A. Frome, Y. Singer, and J. Malik. Image retrieval and classification using local distance functions. In *Advances in Neural Information Processing Systems (NIPS)*, 2006.

[10] A. Frome, Y. Singer, F. Sha, and J. Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classication. In *Proc. of the International Conference on Computer Vision (ICCV)*, 2007.

[11] A. Globerson and S. Roweis. Metric learning by collapsing classes. *Advances in Neural Information Processing Systems*, 18:451, 2006.

[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] Microsoft Kinect. `http://www.xbox.com/en-us/kinect`.

[14] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view RGB-D object dataset. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2011.

[15] K. Lai and D. Fox. Object Recognition in 3D Point Clouds Using Web Data and Domain Adaptation. *The International Journal of Robotics Research*, 29:1019–1037, 2010.

[16] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, June 2001.

[17] David G. Lowe. Object recognition from local scale-invariant features. *Computer Vision, IEEE International Conference on*, 2:1150, 1999.

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

[19] Lukas Meier, Sara Van De Geer, and Peter Bühlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society, Series B*, 70:53–71, 2008.

[20] PrimeSense. `http://www.primesense.com/`.

[21] D. Ramanan and S. Baker. Local Distance Functions: A Taxonomy, New Algorithms, and an Evaluation. In *Proc. of the International Conference on Computer Vision (ICCV)*, 2009.

[22] S. Savarese and L. Fei-Fei. 3d generic object categorization, localization and pose estimation. In *Proc. of the International Conference on Computer Vision (ICCV)*, pages 1 –8, oct. 2007.

[23] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *Advances in Neural Information Processing Systems (NIPS)*, page 41. The MIT Press, 2003.

[24] Noam Shental, Tomer Hertz, Daphna Weinshall, and Misha Pavel. Adjustment learning and relevant component analysis. In *Proceedings of the 7th European Conference on Computer Vision-Part IV*, pages 776–792, 2002.

[25] Cor J. Veenman and Marcel J. T. Reinders. The nearest subclass classifier: A compromise between the nearest mean and nearest neighbor classifier. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 27:1417–1429, 2005.

[26] K.Q. Weinberger and L.K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research (JMLR)*, 10:207–244, 2009.

[27] E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. *Advances in Neural Information Processing Systems (NIPS)*, pages 521–528, 2003.

[28] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68(1):49, 2006.