Practical 3-D Object Detection Using Category and Instance-level Appearance Models

Kate Saenko, Sergey Karayev, Yangqing Jia, Alex Shyr, Allison Janoch, Jonathan Long, Mario Fritz, Trevor Darrell

***Abstract*— Effective robotic interaction with household objects requires the ability to recognize both object instances and object categories. The former are often characterized by locally discriminative texture cues (e.g., instances with prominent brand names and logos), and the latter by salient global shape properties (plates, bowls, pots). We describe experiments with both types of cues, combining a template-and-deformable-parts detector to capture overall shape properties with a local feature Naive-Bayes nearest neighbor model to capture local texture properties. We base our implementation on the recently introduced Kinect sensor, which provides reliable depth estimates of indoor scenes. Depth cues provide segmentation and size constraints to our method. Depth affinity is used to modify the appearance term in a segmentation-based proposal step, and size priors are imposed on object classes to prune false positives. We address the complexity of scanning window HOG search using multi-class pruning schemes, first applying a generic object detection scheme to prune unlikely windows, and then focusing only on the most likely class per remaining window. Our method is able to handle relatively cluttered scenes involving multiple objects with varying levels of surface texture, and can efficiently employ multi-class scanning window search.**

I. INTRODUCTION

The success of mobile robotics platforms hinges on their ability to perform effective and efficient autonomous perception and interaction with objects in unstructured environments. The availability of low cost, high quality sensors and computing systems has equipped us with the necessary data and computational resources for multi-category recognition tasks. Yet few existing systems perform recognition of both highly-textured specific instances and textureless generic objects in cluttered scenes. Such a capability would lay the foundations for robust and reliable mobile manipulation and interaction in domestic settings, where human users often require robotic agents to retrieve and manipulate specific instances or generic categories.

In this paper we explore integrated global template and local feature based recognition, employing efficient multi-class search. Our method (see Figure 1) is effective both on the textured objects for which local feature models have traditionally succeeded, and on objects with little texture for which object-level templates have proven successful. We use the recently introduced Kinect sensor as the basis for our method: its depth estimates provide segmentation cues for a region proposal process and 3-D size constraints for object instance models [1].

The authors are with the University of California at Berkeley and Max-Planck-Institute for Informatics. E-mail: {saenko, sergeyk, jiayq, xshyr, alliejanoch, jonlong, trevor} @eecs.berkeley.edu, mfritz@mpi-inf.mpg.de.

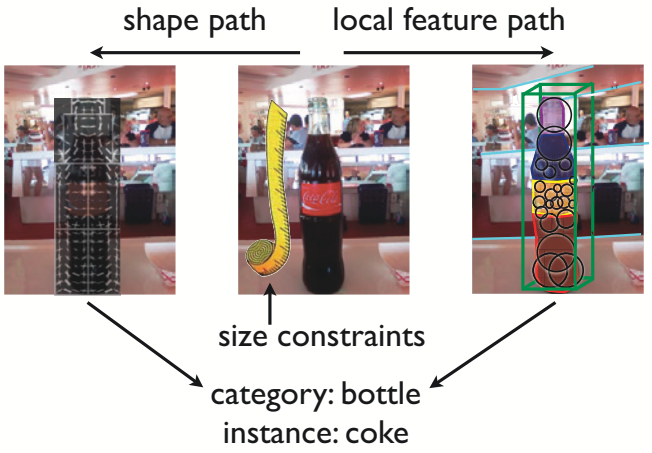

Fig. 1: We propose a practical object detection method, which uses the Kinect sensor for segmentation and size priors, and recognizes both specific object instances defined by distinct local features (coke) and generic categories defined by overall object shape (bottle).

We employ a local feature path following the approach that is commonly used in instance detection schemes, leveraging contemporary local feature models (SIFT, SURF, etc.) and voting or "bag-of-word" matching schemes. We base our approach on the NBNN method, which approximates the likelihood of a new image using the product of individual distance-to-class measures for each local feature in the test image [2]. For the global path, we rely on a contemporary histogram of gradients descriptor that includes latent part components [3]; this method extends the histogram-of-gradients (HOG) object template model with a deformable high-resolution part structure.

Brute force search of local and global models across all possible image windows, part configurations, and all possible category labels is very computationally expensive, and possibly infeasible for practical robotics applications. A single-class speedup incorporated in [4] prunes computation via a cascade of sequential tests. Even with such optimization, existing scanning-window deformable part models are very computationally expensive in a multi-label setting, on the order of one to two seconds per label for complex scenes.

In this paper we propose to control search complexity in the multi-class setting using depth-enabled region proposal schemes based on segmentation and generic object detection

processes. Our segmentation method exploits both depth and RGB cues, and detects and removes support surfaces from input scenes. Coherent regions from the segmentation process are used to define regions for local feature computation. Generic object detection is performed using a domain-adapted variant of a recent method based on 2-D visual attention primitives [5]. This method removes regions unlikely to have any detection. We further propose a novel multi-class extension to the cascade pruning scheme which interleaves search over classes with the standard single-class cascade computation: we add the constraint that only the most likely classes are considered per location.

We experimented with our method on a set of home and office object categories captured using the Kinect sensor. Our dataset contains both category-level and instance-level labels, as well as both objects defined by their local texture and textureless objects defined by their overall shape. Somewhat surprisingly, we found that the global model outperformed the local-feature model across the range of tasks we tested, including both category- and instance-level tasks. We found marginally improved performance with a fused local and global scheme; only in very few categories did the global model perform poorly, and the local feature model provided some benefit. We report on the computational advantage of our pruning schemes, which significantly reduce the amount of computation required in a multi-class global template scheme.

II. RELATED WORK

A comprehensive review of the vast literature on object detection and recognition is certainly beyond the scope of this paper. Here we specifically review the topics relevant to our method: bottom-up segmentation, local feature processing, scanning window models, depth constraints, and generic object detection.

A. Segmentation

Segmentation has been dominated by graph-based models, posed either as an optimization graph-cut framework [6], [7] or with a probabilistic graphical model such as a Markov random field (MRF) [8], [9], [10]. In these approaches, the image is modeled as an undirected graph at the level of pixels. In the case of MRFs, node potentials describe local evidence and edge potentials usually encourage smoothness for neighboring pixels with the same label or similar intensity. Several approaches to inference have been proposed, such as graph cuts [11], belief propagation [12] and iterative merging [13].

To incorporate bottom-up information and speed up computation, the image is often over-segmented via superpixelation [14], [15], [16], [17]. These approaches make use of low-level cues and generate a large number of segments or *superpixels* to ensure that segment boundaries are not lost; this serves as a preprocessing step and most segmentation models operate at the superpixel level for computational efficiency.

Current state-of-the-art segmentation techniques on complex datasets, such as the PASCAL segmentation challenge [18], are framed as two step processes [7], [19]. First, candidate segments, which can be part of an object, are identified. The segmentation problem is then formulated as a ranking problem over the candidate segments, which involves computing segment-level features. Generative models that incorporate naturally occuring power-law prior distributions also exist [20].

B. Local feature models

A particularly effective image representation was developed in recent years. It is formed by computing statistics from distinctive local image points. An effective example is the SIFT feature [21] that has been shown to have extraordinary descriptiveness on precise instance recognition tasks, and has been designed with invariances to many common nuisance parameters such as illumination and slight viewpoint variation. Significant motivation for these architectures arises from biology. Models of the early visual system are frequently taken to integrate statistics over columns of orientation selective units [22]. The bag of local visual words (BoW) model [23] has been shown effective in representing both images and regions. Recently, the naive Bayes nearest neighbor (NBNN) algorithm [2] has been shown to have surprisingly strong performance both in theory and in practice on pure classification tasks (e.g., Caltech-101). The lack of any model of geometry limits the utility of this approach to detecting objects with only generic texture, or where the spatial relationship of local features is discriminative.

C. Scanning window search models

For datasets featuring large pose variation and multiple instance types in cluttered scenes, such as the PASCAL VOC [18], state-of-the-art detection approaches have for several years consisted of global feature sliding window detectors [24], [3], [25]. These detectors work on a representation of the image based on gradient statistics, like SIFT, but consider such statistics over an object-scale window, not in local patches. A classifier evaluates windows of a fixed aspect ratio across locations and scales of an image pyramid. One of the leading detectors augments the object models with parts, which can be placed at some offset relative to the root template, with fitting cost increasing with the deformation [3]. Windows scored higher than a threshold are considered object detections. These are further agglomerated and pruned with non-maximum suppression, and sometimes by considering additional cues such as inferred geometry of the scene or the context provided by other detections [26]. For good localization accuracy, the coverage of the image pyramid with considered windows must be dense, which is computationally expensive. An important speedup for deformable part models consists in "cascading" the detector by only trying to fit a part at a location if the root filter score and all part scores earlier in the cascade are past some threshold [4].

D. Depth constraints

There have been many 3-D features proposed for recognition. Briefly, prominent techniques include spin images [27], 3-D shape context [28], and the recent VFH model [29]. While we have not directly employed 3-D local features in the work reported here, this is anticipated future work.

A number of 2D/3D hybrid approaches have been recently proposed. A multi-modal object detector in which 2D and 3D are traded off in a logistic classifier is proposed by [30]. Their method leverages additional hand-crafted features derived from the 3D observation such as "height above ground" and "surface normal," which provide contextual information. Work such as [31] shows how to benefit from 3D training data in a voting based method. Fritz et al. [1] extend branch-and-bound to 3D by adding size and support surface constraints derived from the 3D observation.

Most prominently, a set of methods have been proposed for fusing 2D and 3D information for the task of pedestrian detection. The popular HOG detector [32] to disparity-based features is extended by [33]. A late integration approach is proposed by [34] for combining detectors on the appearance as well as depth image for pedestrian detection. Instead of directly learning on the depth map, [35] uses a depth statistic that learns to enforce height constraints of pedestrians. Finally, [36] explores pedestrian detection by using stereo and temporal information in a Hough-voting framework, also using scene constraints.

The use of both visual and depth information for object detection and classification in indoor scenes has recently been proposed in [37], [38]. Specifically, spin images are used to model the shape information and bag of local image features are used to model the visual information, and a classifier (such as an SVM) is trained to perform detection/classification.

E. Generic object detection

Several recent works tackle the problem of generic object detection, where the goal is to find objects without regard for what they are. Some attempt to produce fully segmented regions using complex boundary, color, texture, and context cues [39], [40], while others use simpler cues that can be computed quickly to produce bounding boxes that are likely to contain objects [5]. The latter case may be immediately applied to prune the windows considered by a sliding window detector, and has already been shown by the authors of [5] to be effective in reducing the number of windows considered by several common sliding window class detectors, without an appreciable loss in the number of objects found. The cues considered include a χ^2 distance between color histograms inside and surrounding the bounding box, a spectral saliency measure [41] at multiple scales, and a measure of the amount of superpixel straddling in the bounding box. The superpixels use the same segmentation algorithm from [13], described elsewhere in this work.

III. Approach

We experiment with a detector pipeline that includes both local and global feature models. For local feature processing, we first compute candidate segments using bottom-up segmentation cues, using a fast segmentation method that leverages depth images provided by the Kinect sensor in conjunction with a support surface elimination method, and then apply an NBNN classification scheme. For global processing, we employ a domain-adapted visual attention scheme that can identify salient regions in a scene, and use this as a pre-filter for a HOG-based deformable template scanning window detector. We then combine the detections from the two pipelines to produce a list of bounding boxes including both category and instance labels.

A. Segmentation

Segmentation as a preprocessing step can substantially reduce the search time over bounding boxes, and can potentially increase the robustness of our algorithm as the resulting segments are often tighter than bounding boxes produced by a branch-and-bound search. We use a depth-enhanced segmentation process as a source of proposed candidate regions for local feature descriptor extraction and NBNN classification.

For computational efficiency we choose to base our segmentation method on a graph-based segmentation scheme, as described in [13]. In their framework, a graph is constructed where the nodes correspond to pixels and the edges are defined between select neighboring pixels. Edge weights are the L_2-norms between the intensity of pixels. The graph then serves as the input into the segmentation algorithm, which iteratively merges adjacent components whose connecting edge weights are small when compared to the internal differences of the two components.

The method, however, has limitations; in cases where RGB information gives the wrong cue (for example, in scenarios with shadows or transparent objects), the algorithm will generate incorrect segments. The availability of depth data can be helpful in these situations, and the edge weights described above can be modified to include depth information.

Rao et al. [42] propose an extension to the graph-based segmentation framework by augmenting the intensity of a pixel with a fourth dimension from the depth value, and setting edge weights to a weighted L_2-norm. The intuition behind the extension is that pixels with the same depth should be in the same component. However, in the extreme case where the weighting of the depth channel vastly dominates the weighting of the color channels, the components become level sets of depth. This is undesirable as a tabletop or a piece of wall will get segmented into different slices (see Fig. 2.

We propose to add another distance metric based on the depth gradient. The intuition here is that pixels with the same depth gradient (ie, lying on the same plane) should be in the same component. The final edge weight between pixels p_i and p_j is

$$||\mathbb{I}(p_i) - \mathbb{I}(p_j)||_2 + c_1|\mathbb{D}(p_i) - \mathbb{D}(p_j)| + c_2|\partial\mathbb{D}(p_i) - \partial\mathbb{D}(p_j)|, \quad (1)$$

where $\mathbb{I}$ is the intensity, $\mathbb{D}$ is the depth map, $\partial\mathbb{D}$ is the approximated depth gradient evaluated along the direction from p_i to p_j, and c_1, c_2 are coefficients for weighting the two metric extensions. We found that setting c_1 and c_2 to 20 and 0.5 gives the best results.

Given that scenes in our dataset consist of a table with objects on top, we post-process the segments to identify the table surface. We first assume the camera angle is upright and the table's normal vector points upward. We then fit planes to each segment, and select one to be the table that has a desired normal and is flat. The segments immediately above the table are then aggregated, and disjoint components are added to the list of candidate regions. This last step aims to combine different parts of an object (for example, the body of a bottle is likely to be separate from the lid due to dissimilar color and depth).

B. Local Pipeline

On detected regions from the segmentation path, we apply a local feature NBNN model. Our local pipeline takes the segmentation result, represented as image regions, as possible object proposals, and predicts object labels (including background) for each region. In the training phase, local image descriptors are extracted from image regions that correspond to objects of each class. For each object proposal in the testing phase, we extract descriptors $d_1, d_2, \cdots, d_n$ from the corresponding image region. The score of each class $c \in C$ is then computed as

$$\text{score}(c) = \sum_{i=1}^{n} \|d_i - NN_c(d_i)\|^2 \quad (2)$$

where $NN_c(d_i)$ is the nearest neighbor of d_i in the training local descriptors that belong to c. A smaller score implies a higher probability that the proposal belongs to the class. To enforce depth constraints within an NBNN paradigm, we follow the method outlined in [1], with category and instance size estimated directly from the available 3-D training data.

Notice that the original NBNN method [2] is designed for classification instead of detection. It is straightforward to add a "background" class to the object labels, and treat detection as a classification problem. However, in practice this leads to very poor performance that results from the incompleteness of labeling: there are cases when an object is not labeled in the training data although it is present in an image. This makes the background class contain local features from objects too. During classification, there is a high probability that these objects are then classified as background, leading to a poor recall.

While a more exhaustive and accurate labeling of the data would solve the problem (but would be very hard to carry out, as we use crowd-sourcing for labeling), we follow a simple approach that is similar to the idea of local feature matching adopted in [21]. Specifically, we compute the ratio of the smallest score to the second smallest score, and reject all proposals whose score ratio is greater than a certain threshold. This worked well in practice: if an image region belongs to a certain class, it should be distinctively different from other classes while being similar to the correct class, leading to a small ratio. In practice, we set the ratio threshold to 0.85.

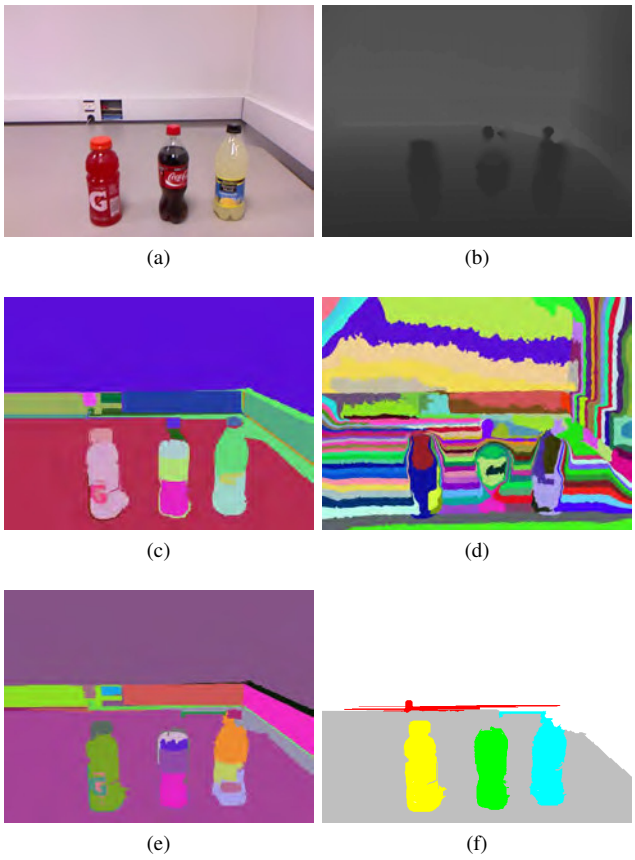

Fig. 2: Example of segmentation. (a), (b): RGB and depth of input image. (c): Output segmentations using only RGB information; note the incorrect segmentation caused by the leftmost bottle's shadow. (d): Output segmentation using only absolute depth data. (e): Output segmentation with both absolute depth and depth gradient. (f): Additional masks with the assumption of a table top. The gray plane represents the table; note how the different components of a bottle (lid, body, label) are combined to form a better segmentation.

C. Shape template processing

We evaluate a sliding window HOG model for each class, pruning locations that were not proposed by the generic object detector stage. We follow the implementation of the deformable part model detector [3] that scores candidate windows with the LatentSVM formulation

$$f_\beta(x) = \max_z \beta \cdot \Phi(x, z) \quad (3)$$

where β is a vector of model parameters and $\Phi(x, z)$ is a feature function of the image window x and latent values

z, allowing for flexible part locations in the model. This objective function is a semi-convex optimization problem; the detector can be trained even though the latent information is absent for negative examples.

In a large dataset, training a detector necessarily considers only a small fraction of possible examples. For this reason, the system performs rounds of data mining for samples of hard negatives, providing the exact solution to training on the entire dataset.

To featurize the image, we use a histogram of oriented gradients (HOG) with both contrast-sensitive and contrast-insensitive orientation bins, four different normalization factors, and 8-pixel wide cells. The descriptor is analytically projected to just 31 dimensions, motivated by the analysis in [3].

Raw detections go through non-maximum suppression—a greedy selection of highest-scoring bounding boxes and corresponding rejection of overlapping detections. This procedure results in a significant reduction of detections. As our evaluation metric penalizes repeat detections, NMS is an important step in the process.

D. Multi-class pruning

Additionally, we investigate a novel multi-class pruning scheme for detection. Observe that a window of a certain scale centered on a certain point should only be expected to contain one object, which can be assigned a category and an instance label. With this expectation, there is little reason to consider every location with a detector of every class. For example, if a location is highly likely to be detected as a bottle, clock and bowl detectors can safely skip it.

Our multi-class pruning scheme builds on the scheme of the cascaded LatentSVM detector, which only performs expensive part fitting at a given location if the score of the model root filter is past some threshold [26]. Additionally, a low-dimensional PCA projection of the HOG feature and model weights is used for additional speedup. We augment this scheme to deal with multiple classes by first densely scoring the root filter projections for all classes. This gives the maximum score for all locations, and thus the detector(s) most likely to have the highest score after full part fitting. This expensive part fitting then only occurs for those detectors that are within some threshold of the maximum score. The threshold thus determines what percentage of the image feature pyramid is pruned away.

E. Domain-adapted generic object detection

We construct an efficient mechanism for proposing windows to our LSVM class detectors that are likely to contain objects using a domain dependent improvement to [5]. In that work, a number of local cues described in section II-E are combined with naive Bayes in order to score bounding boxes. We add to these a domain dependent size cue.

Our improvement is based on the observation that absolute bounding box size has a particular distribution which depends on the domain of application, while the local cues employed by generic object detection implicitly define a size distribution which is quite different. For example, our tabletop images are typically cluttered with many small, upright objects. Without properly incorporating this information, generic object detection tends to produce bounding boxes which enclose several adjacent objects.

Our size cue is the probability of a bounding box of the observed size under the distribution defined by Gaussian kernel density estimation on the sizes of ground truth bounding boxes in the training set. The distribution is smoothed with $\sigma = 50$.

As in [5], these scores may be used to form a distribution over windows. Samples from this distribution are passed on to a class specific detector. Thus higher scoring boxes are more likely to be considered, while some lower scoring boxes will also be considered. However, this procedure can create a tendency to fixate on a few high scoring areas, even after all meaningful windows in those areas have been examined.

To improve the rate at which objects are found, we take several additional steps which integrate and improve the post processing described in [5]. We replace the sampling procedure entirely with a simple sorting one — boxes are considered highest score first, and any boxes that have high overlap with previous ones (defined by thresholding intersection/union) are skipped. This inexpensive procedure ensures that the best windows are considered first and that additional windows explore the space of bounding boxes rather than sticking to a few highly salient objects.

Since boxes suggested by generic object detection do not always line up with the boxes that a HOG detector scores highest, even when they indicate the same objects, we perform a local search around the generic boxes to improve the chance of a detection. After converting a bounding box to the best matching location in a feature pyramid, we iteratively move to whichever of the four nearest neighbors at the same scale in the pyramid scores highest in a part cascade. We typically find a local maximum, and a detection, if one exists, in less than five moves.

F. Category-sensitive fusion

Our final result is computed by taking the union of detections from the local feature and global feature paths. When computing non-maximum suppression of detections, we provide exceptions for the case where an instance label detection overlaps a category label detection and the instance is a member of the category, so that our final output labels a coke bottle both with the category label “bottle” and the instance label “coca-cola”.

IV. EXPERIMENTS

A. Data Collection

We collected images of objects typically found in both households and offices using the Microsoft Kinect sensor. The dataset consists of 269 images used for training and 85 images used for testing and contains both textured objects appropriate for instance recognition tasks, and untextured objects appropriate for category-level recognition. We include textured objects such as Coca-Cola bottles, Listerine,

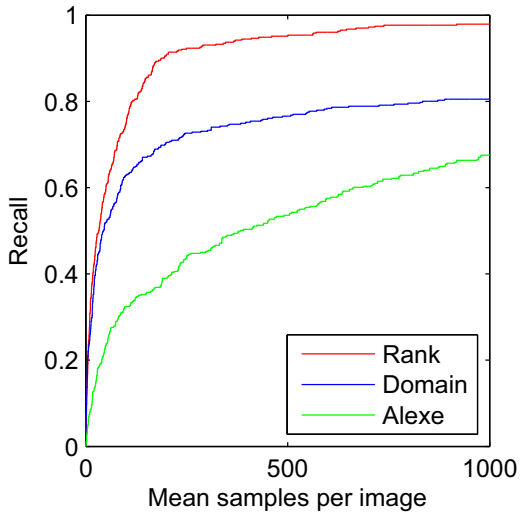

Fig. 3: Pruning windows with a generic object detector. The y-axis gives recall on the detections found by applying our HOG LSVM class detectors to a dense grid of windows. The x-axis gives the average number of windows proposed by a generic object detector per image, at a particular threshold of confidence. From bottom to top, the curves show: (a) performance using the default system of [5] (green) (b) performance using the using the domain-adapted method (blue) (c) performance using the domain-adapted method and suppressing overlapping boxes rather than sampling as described in the text (red).

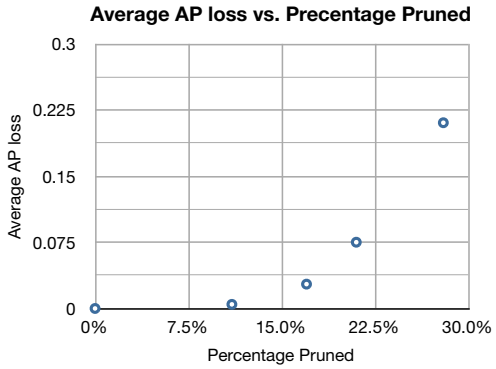

Fig. 4: Effect of increasing amount of pruning in the multiclass LSVM cascade. The y-axis shows the change in the average precision, averaged across all classes, while the x-axis shows the ratio by which computation is decreased. As can be seen from the plot, 15-20% of the computation can be eliminated with minimal loss in the average precision.

and Gatorade, and untextured object classes such as pot, clock, clipboard and plate. The images were collected in a controlled environment: they were all captured on office table tops with a relatively uncluttered background. Although the environment is controlled there is much variability in the pose and location of the objects. In addition, objects were often photographed in groups arranged in such a way that some of the objects were occluded.

We used crowd sourcing on Amazon Mechanical Turk (AMT) in order to label the training data we collected with bounding boxes. Amazon Mechanical Turk is a well-known service for "Human Intelligence Tasks" (HIT). A labeler verification step was employed to ensure the majority of labelers agree on a box before it is accepted as a true label.

B. Evaluation

First, we evaluate the window proposal scheme described in Section III-E. Figure 3 shows the effectiveness of domain-adapted generic object detection on our data. By considering only a few hundred windows per image, we can find (in the sense of intersection/union > 0.5) nearly all objects that we would find by searching a dense gridding of windows.

Finally, we report detection results. Figure 5 shows detection results, in terms of the average precision (AP) for each class; Figure 6 shows example detections overlaid on test images. We find that the local-feature NBNN detector alone has inferior performance, except in the cases of 'clock' and 'scissors', where the global HOG detector fails to detect anything. However, if the number of training examples is insufficient for the LSVM, the local feature path could provide an advantage, and so is worth considering. We also find that the combined system is only slightly better than the HOG detector alone. We compute the AP results when limited to the detections that are found from a strict hill climbing search starting from the top 1000 proposed windows from the generic object detection method. The effect on the average precision is minimal. However, the savings in computation are significant: the detector only needs to be run for the 1000 boxes and a few neigbors. On average, 14.4% of the image feature pyramid is evaluated in this fashion, yielding a speed-up factor of two with a memoizing implementation. We also performed a post-hoc evaluation of the multiclass pruning scheme described in Section III-D. We can vary the threshold to obtain increasing amounts of pruning in the cascade. Figure 4 shows the average loss in AP at varying thresholds. As can be seen from the plot, a further 15-20% of the computation can be eliminated with minimal loss in the average precision.

V. CONCLUSIONS

We have compared local and global feature models for robotic recognition of household objects at both a category and instance level. Our method includes both a local feature pipeline, based on a naive Bayes classification model, and a more holistic, template-and-deformable-parts model typically employed in a scanning window fashion. We base our implementation on the recently available Kinect sensor, which provides reliable depth estimates of indoor scenes; depth cues provide segmentation and size constraints to our method. Depth affinity is used to modify the appearance term in a segmentation-based proposal step, and size priors are imposed on object classes to prune false positives.

For the categories and instances we investigated, we found few examples where the local feature path outperformed

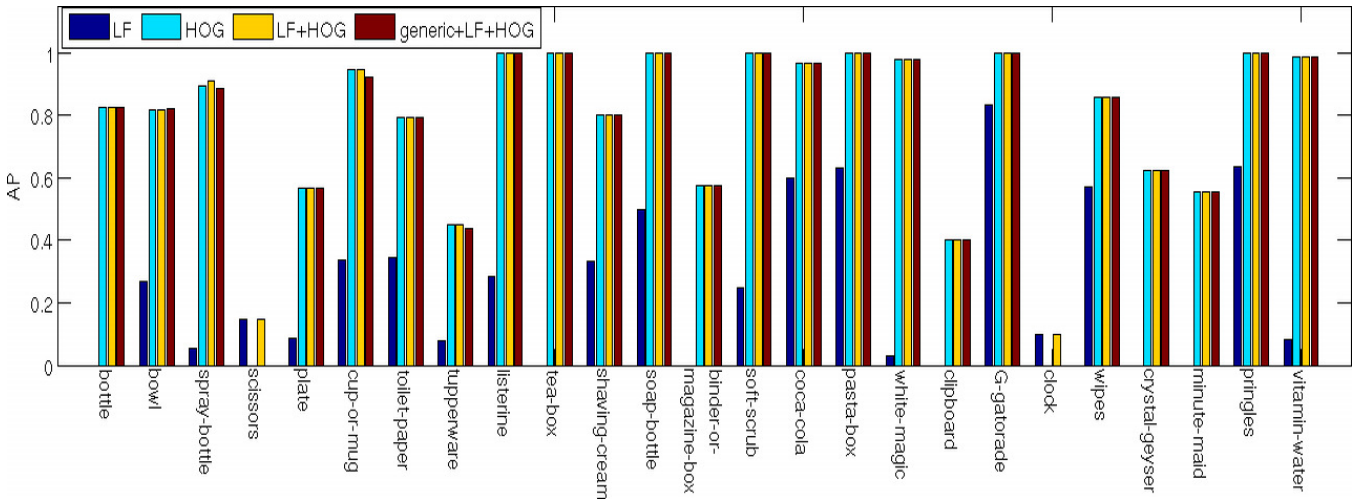

Fig. 5: Comparison of average precision (AP) for several detectors: LF: only local feature NBNN, HOG: only scanning-window HOG LSVM, $LF+HOG$: proposed system combining both pipelines, and $generic+LF+HOG$: combined system with bounding boxes pre-filtered by the generic object detector. The average AP over all classes is $LF = .25$, $HOG = .76$, $LF + HOG = .77$, $generic + LF + HOG = .76$.

Fig. 6: Each row shows sample detections on a test scene. From left to right: (1) human annotations, (2) local feature NBNN (LF), (3) our method.

the holistic path. This may be due in part to our training regime, where a sufficient amount of training examples were available for SVM learning; were we to re-evaluate in a regime with only one or two training images per label, the NBNN model might dominate the HOG LatentSVM path. (Such experiments are future work.)

To enable efficient multi-class use of scanning-window deformable-parts models, we proposed two multi-class pruning schemes. We showed how a domain-adapted generic object detection scheme could prune unlikely windows, reducing the number of windows significantly with only a modest loss in AP. We also developed a multi-class extension of the LSVM cacade, where only the most likely class is considered at a given window, further reducing computation by a factor of 10-20%.

Our method is able to handle relatively cluttered scenes involving multiple objects with varying levels of surface texture, and can efficiently employ multi-class scanning window search; our system should serve as a starting point for tasks involving object interaction in everyday environments.

REFERENCES

[1] M. Fritz, K. Saenko, and T. Darrell, "Size matters: Metric visual search constraints from monocular metadata," in *Advances in Neural Information Processing Systems 23*, 2010.

[2] O. Boiman and E. Shechtman, "In defense of nearest-neighbor based image classification," *CVPR*, 2008.

[3] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE Pattern Analysis and Machine Intelligence (PAMI)*, pp. 1–20, Jul 2009.

[4] P. F. Felzenszwalb, R. B. Girschick, and D. McAllester, "Cascade object detection with deformable part models," *CVPR*, pp. 1–8, Mar 2010.

[5] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?" in *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2010.

[6] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Pattern Analysis and Machine Intelligence*, vol. 22, 2000.

[7] J. Carreira and C. Sminchisescu, "Constrained parametric min-cuts for automatic object segmentation," in *Computer Vision and Pattern Recognition*, 2010.

[8] P. Kohli, M. Pawan, K. Philip, and H. S. Torr, "p^3 & beyond: Solving energies with higher order cliques," in *Computer Vision and Pattern Recognition*, 2007.

[9] M. P. Kumar, P. H. S. Torr, and A. Zisserman, "Obj cut," in *Computer Vision and Pattern Recognition*, 2005.

[10] L. Ladicky, C. Russell, P. Kohli, and P. Torr, "Graph cut based inference with co-occurrence statistics," in *European Conference on Computer Vision*, 2010.

[11] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *Pattern Analysis and Machine Intelligence*, vol. 26, 2004.

[12] P. Felzenszwalb and D. Huttenlocher, "Efficient belief propagation for early vision," *International Journal of Computer Vision*, vol. 41, 2006.

[13] ——, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, 2004.

[14] X. Ren and J. Malik, "Learning a classification model for segmentation," in *International Conference on Computer Vision*, 2003.

[15] A. Moore, S. Prince, J. Warrell, U. Mohammed, and G. Jones, "Superpixel lattices," in *Computer Vision and Pattern Recognition*, 2008.

[16] A. Levinstein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi, "Turbopixels: Fast superpixels using geometric flows," *Pattern Analysis and Machine Intelligence*, vol. 31, 2009.

[17] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "From contours to regions: An empirical evaluation," in *Computer Vision and Pattern Recognition*, 2009.

[18] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results," http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.

[19] I. Endres and D. Hoiem, "Category independent object proposals," in *European Conference on Computer Vision*, 2010.

[20] E. Sudderth and M. Jordan, "Shared segmentation of natural scenes using dependent pitman-yor processes," in *Advances in Neural Information Processing Systems*, 2008.

[21] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, 2004.

[22] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *nature neuroscience*, vol. 2, pp. 1019–1025, 1999.

[23] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," *CVPR*, pp. 524–531, 2005.

[24] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *CVPR*, pp. 1–8, 2005.

[25] X. Wang, T. X. Han, and S. Yan, "An hog-lbp human detector with partial occlusion handling," *ICCV*, pp. 1–8, Jul 2009.

[26] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert, "An empirical study of context in object detection," *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pp. 1271–1278, jun 2009.

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

[28] A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *ECCV04*, 2004, pp. Vol III: 224–237.

[29] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3d recognition and pose using the viewpoint feature histogram," in *Proceedings of the 23rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 10/2010 2010.

[30] S. Gould, P. Baumstarck, M. Quigley, A. Y. Ng, and D. Koller, "Integrating visual and range data for robotic object detection," in *ECCV Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications (M2SFA2)*, 2008.

[31] M. Sun, G. Bradski, B.-X. Xu, and S. Savarese, "Depth-encoded hough voting for joint object detection and shape recovery," in *Proceedings of European Conference on Computer Vision*, 2010.

[32] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2005.

[33] H. Hattori, A. Seki, M. Nishiyama, and T. Watanabe, "Stereo-based pedestrian detection using multiple patterns," in *Proceedings of British Machine Vision Conference*, 2009.

[34] M. Rohrbach, M. Enzweiler, and D. M. Gavrila, "High-level fusion of depth and intensity for pedestrian classification," in *Annual Symposium of German Association for Pattern Recognition (DAGM)*, 2009.

[35] S. Walk, K. Schindler, and B. Schiele, "Disparity statistics for pedestrian detection: Combining appearance, motion and stereo," in *Proceedings of European Conference on Computer Vision (ECCV)*, 2010.

[36] A. Ess, K. Schindler, B. Leibe, and L. V. Gool, "Object detection and tracking for autonomous navigation in dynamic environments," *International Journal on Robotics Research*, 2010.

[37] K. Lai, L. Bo, X. Ren, and D. Fox, "Sparse Distance Learning for Object Recognition Combining RGB and Depth Information," in *ICRA*, 2011.

[38] ——, "A Large-Scale Hierarchical Multi-View RGB-D Object Dataset," in *ICRA*, 2011.

[39] I. Endres and D. Hoiem, "Category independent object proposals," in *Proceedings of European conference on Computer Vision (ECCV)*, 2010, pp. 575–588.

[40] J. Carreira and C. Sminchisescu, "Constrained parametric min-cuts for automatic object segmentation," in *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2010.

[41] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2007.

[42] D. Rao, Q. Le, T. Phoka, M. Quigley, A. Sudsang, and A. Ng, "Grasping novel objects with depth segmentation," in *Proceedings of International Conference on Intelligent Robots and Systems*, 2010.