RGB-D Multi-View Object Detection with Object Proposals and Shape Context

Georgios Georgakis and Md. Alimoor Reza and Jana Kosecka

***Abstract*— We propose a novel approach for *multi-view* object detection in 3D scenes reconstructed from RGB-D sensor. We utilize shape based representation using local shape context descriptors along with the voting strategy which is supported by unsupervised object proposals generated from 3D point cloud data. Our algorithm starts with a *single-view* object detection where object proposals generated in 3D space and combined with object specific hypotheses generated by the voting strategy. To tackle the multi-view setting the data association between multiple views enabled view registration and 3D object proposals. The evidence from multiple views is combined in simple bayesian setting. The approach is evaluated on the WRGB-D Object datasets [1], [2] containing several classes of objects in a table top setting. We evaluated our approach against the other state-of-the-art methods and demonstrated superior performance to the state-of-the-art on the same dataset.**

I. INTRODUCTION

Object detection is a key ingredient of scene understanding which is leveraged by other high-level service robotics tasks, such as fetch and delivery of objects, object manipulation and object search. While this problem is widely studied with image only sensing modality, it has been demonstrated that the availability of the depth information can be effectively utilized to improve performance and the robustness of the existing systems [2], [3]. The majority of the existing approaches use the traditional object detection and recognition pipelines in the RGB-D setting treating the depth as an additional channel. These include sliding window based detectors [4], [5], [1] or methods based on local feature descriptors [6], [7], [8], [9], [10]. The local descriptors, which capture the local appearance or geometry statistics of the objects are effective for situations with large amounts of clutter and occlusion. In the proposed approach we utilize local shape based descriptors extracted from image contours and implicit shape models to generate the object hypotheses. In the testing, the shape descriptors are extracted along depth discontinuities which often coincide with the object boundaries. The voting stage is further supported by generation of unsupervised object proposals, which are generated by mean-shift clustering in 3D.

In the *single-view* object detection many of the misses or false positives are often caused by occlusion or view-dependent ambiguities which can be often resolved in a multi-view detection settings. Towards this end we propose to integrate the single view detections in the sequential setting. Exploiting the fact that the views are registered together (poses of the cameras are known), we use the projections of generated 3D object proposals and object category hypotheses to the next view and combine it with the single-view hypotheses to generate new predictions. Using this simple data association method between the frames, the class distribution is updated using a Bayesian update. In summary:

- We show that unsupervised 3D object proposals support the implicit shape models favorably, reducing the number of false positives in *single-view object detection*. Experimental results corroborates this observation.
- We further show that the performance of object detection can be significantly improved by integrating the evidence from multiple views

We validate our approach on the WRGB-D benchmark dataset [1], [2] and experimentally achieve superior performance.

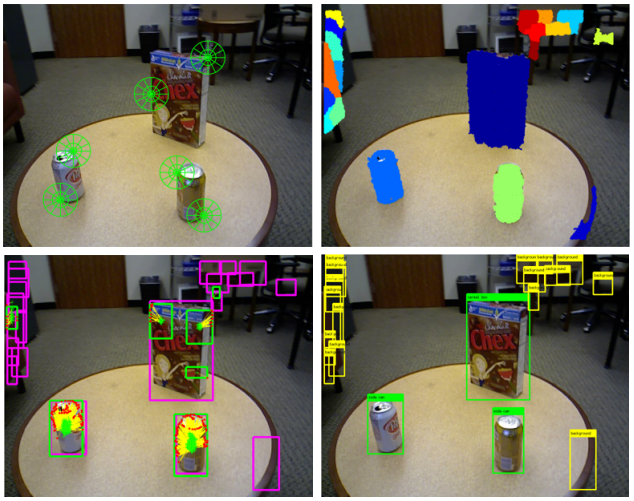


Fig. 1: Brief overview of our approach. Our first step is the extraction of the shape contexts (top left), followed by the generation of the object proposals (top right). The hypotheses created by our detectors (shown in green bounding boxes) along with the votes and the proposals (shown in magenta bounding boxes) are illustrated in the bottom left image. For clarity, only the soda can detector is shown. Finally, after the integration of all detectors and the multi-view information, we get our final output (bottom right). The green bounding boxes indicate correct detections and the yellow indicate background.

The authors are with the Department of Computer Science, George Mason University, Fairfax, VA, USA. {ggeorgak, mreza, kosecka}@gmu.edu

Preprint submitted to 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems. Received March 1, 2016.

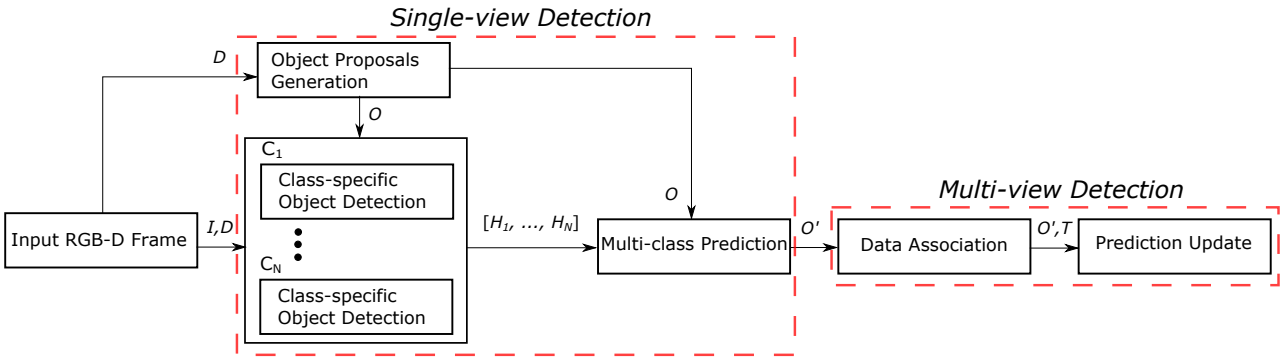


Fig. 2: Outline of our approach. Given the image frame I and the depth map D as input we first obtain the set of object proposals O and apply the class-specific object detectors for all object categories $[C_1, ..., C_N]$. For each category C we obtain a set of hypotheses H where $[H_1, ..., H_N]$ is the set of all hypotheses from all object detectors. The hypotheses scores are then normalized and associated with the object proposals to get the multi-class prediction O' on the frame. Each proposal in O' now holds a probability distribution over the object categories. We then associate the proposals with the object tracks T in the scene, and update the objects probability distribution using the history of observations in the tracks.

II. Related Work

The problem of object detection and categorization has been studied extensively both in RGB and RGB-D settings. Here we briefly review the closest works to ours with respect to object proposals, local shape descriptors, voting and multi-view detection techniques. In an attempt to reduce the search space of the traditional sliding window techniques such as [11], [12], some recent works have concentrated in generating category-independent object proposals. These methods usually start with bottom-up segmentation generating regions which are likely belong to objects. Pre-trained classifiers are then evaluated for these regions [13] or various matching and clustering techniques are used to extract object models in an unsupervised or weakly supervised manner [14]. In the table-top RGB-D settings, Mishra et al. [15] uses object boundaries to guide the detection of fixation points that denote the presence of objects, while Karpathy et al. [16] performs object discovery by ranking 3D mesh segments based on objectness scores. Firman et al. [17] initially removes the large planar regions in the image, groups pixels in the image and 3D space, and then uses correlation clustering to discover objects across different scenes. Triebel et al. [18] applies graph-based clustering in geometric space and models the interactions of the clusters through a CRF framework in order to assign object labels. In our work, we exploit the availability of 3D data and generate object proposals by mean-shift clustering of 3D points and project them to the respective views.

Object representations which use local features are one of the most popular approaches in object detection and recognition, due to their capability to deal with textured household objects. Works of Collet et al. [6] and Tang et al. [7] take advantage of SIFT's discriminative nature and use it to create 3D object representations during training. A disadvantage of these descriptors is that they usually perform poorly in the presence of non-textured objects. Since we are interested in detecting both textured and non-textured objects, we take advantage of objects' shape properties as encoded by the Shape Context [19] descriptor. A Shape Context is defined as the concatenation of histograms of quantized edge orientations around a reference point. Two examples of Shape Contexts usage are Wang et al. [20] for the detection of the pedestrians, cars, and bikes, and Teo et al. [21] for estimating the pose of objects in cluttered RGB-D settings. Given local features representations, one of the common strategies for detection and localization of objects in images is provided by Implicit Shape models [22] and variations of these [20], [23], [24]. In these models local features are augmented their coordinates relative to the center of the object. During testing, matched codebook entries cast probabilistic votes which lead to the formation of object hypotheses. An issue with these works is that they have to repeat the voting at several scales. We exploit the depth information to avoid the voting at several scales and use the object proposals to remove unwanted votes that generate false positives.

In robotic settings, where the sensor has a capability of moving, it is particularly attractive to consider multi-view detection. Thomas et al. [25] learns a separate Implicit shape model for each viewpoint which are connected with correspondences so as to transfer information between the views during detection. Similarly, Sun et al. [26] learns a probabilistic model that establishes correspondences of parts across a discretized set of viewpoints, for the goal of 3D object categorization. Herbst et al. [27] creates 3D reconstructions of scenes and discovers objects based on appearance and 3D alignment-based matching from different views. Pillai et al. [28] uses monocular SLAM on an RGB video to reconstruct the scene which is then segmented into smaller 3D point clouds. The projection of the 3D point clouds on the video frames creates multi-view object proposals. The single frame predictions are then aggregated across the frames to produce their results. A problem with this approach is that to get accurate object proposals the

Preprint submitted to 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems. Received March 1, 2016.

monocular SLAM has to run for a significant amount of time on the video. In contrast, in our work we use depth to generate object proposals in each frame and refine our predictions online using sequential Bayesian update. Multi-view object detection has also been used for other tasks such as 3D scene labelling by Lai et al. [29], [2]. The goal of labelling each 3D point in a scene is achieved by integrating the class probabilities found by view-based detectors for each corresponding pixel through an MRF framework.

III. Approach

We start first by excluding large planar surfaces from RGB-D data, followed by 3D point clustering to generate object proposals. We then sequentially apply class-specific object detectors in order to generate a set of hypotheses for each object category. The object models are trained by collecting a set of local shape descriptors from the training set images in uncluttered background. In the test stage the object proposals guide the sampling of shape descriptors and accepting the object's hypotheses generated by voting. The hypotheses scores are normalized across all object categories to get class probability distribution and are associated with the object proposals. To incorporate multi-view information, for the next RGB-D frame previous hypotheses are projected to the next frame and new observations are used to update the distribution. Figure 2 shows an overview of our approach. We discuss in more detail the object proposals generation in Section III-A, the single-view object detection in Section III-B, and the multi-view object detection in Section III-C.

A. Object Proposals Generation

In the table-top scenes, like the ones in WRGB-D Objects Dataset [1], [2], small objects lie on top of the planar surfaces. In order to identify these smaller objects, we use the methods described in [30] to fit large planar surfaces aligned with dominant orientations to 3D point clouds. The large horizontal surfaces with normals oriented towards the gravity direction are classified as the support surfaces and 3D points belonging to any large surfaces (including support surfaces) are not considered in the proposal generation stage. The remaining 3D points are clustered using mean-shift clustering [31], generating compact clusters of 3D points. Figure 3 shows examples of clusters in different colors in the image.

The clustering requires only a radius parameter that defines the maximum euclidean distance between points to be included in the neighbourhood. We change the radius parameter for different RGB-D frames based on the median depth value of the support surfaces. We then project the clustered 3D points on the image space and divide them to connected components to ensure continuity. We reject some connected components that are too small and also eliminate connected components that are further away from the support surface by utilizing the distance of the centroid of a connected component in 3D from the support surfaces. The remaining connected components form our valid object proposals $O = \{o_1, ..., o_K\}$ for a particular image. Each object proposal $o_k = \{pc_k, m_k\}$ is characterized by its 3D point cloud pc_k and its segmentation mask m_k in image space.

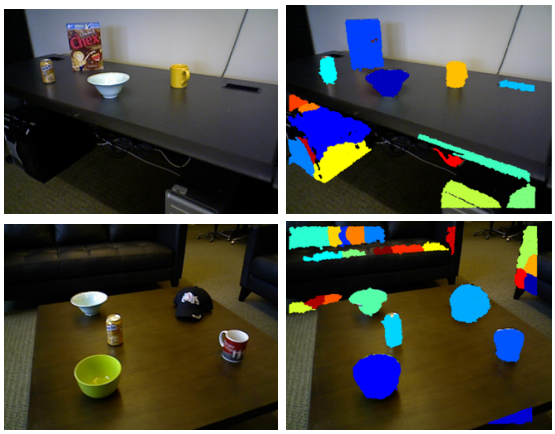


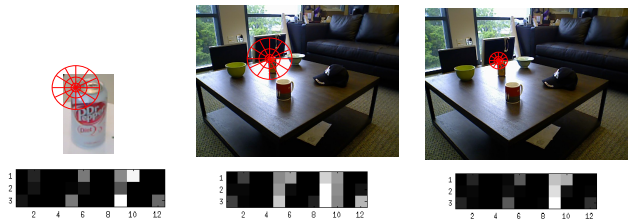
Fig. 3: Examples of our object proposals generation output. The right column shows the segmentation masks of the proposals for each frame shown on the left column.

B. Single-View Object Detection

In the training stage we collect a set of shape contexts [20] for each object category from class-specific training images (see Figure 5). The training images contain a single instance of our object in a clear background along with the segmentation mask. Descriptors are extracted on sampled 2D edge points. We utilize edges from both RGB as well as depth channel. Depth edges are found using Canny edge detector and are complemented using the boundaries found by *gPb* detector [32] in the RGB image. Although there are more recent edge detectors available [33], they are trained on more generic scenes and as a result they are more sensitive to texture.

We extract the *shape contexts* on our combined edge map found from the RGB-D image. Training images are filtered with the ground truth segmentation mask to ensure that only edge points that belong to the object are used. The shape context descriptor is defined by the following parameters: n_θ, n_ϕ, and n_r. Here, n_θ is the number of orientations that the edge map is divided, n_ϕ is the number of angular bins, and n_r is the number of radial bins. For each edge orientation, a histogram of $n_\phi \times n_r$ dimensions is computed, producing a total dimensionality of $n_\theta \times n_\phi \times n_r$ for each shape context. In our case we used $n_\theta = 4$, $n_\phi = 12$, and $n_r = 3$, resulting in a 144 dimensional descriptor. In addition, $\hat{r}$ is a vector defining the extend of the radial bins in image space. We refer to this parameter as the radius of shape context. An example of shape context applied on a training image can be seen in Figure 4a. Objects sizes vary in training images and the *shape context* descriptor's radius parameter needs to be conformed to this size variance. We tackle this size variance by determining the appropriate radius of the descriptor for each object on an independent validation set.

Preprint submitted to 2016 IEEE/RSJ International Conference
on Intelligent Robots and Systems. Received March 1, 2016.



(a) (b) (c)

Fig. 4: Shape Context radius scaling. In the first row (a) shows a shape context applied on a training image, (b) shows a shape context with the same radius as (a) applied on a test image, and (c) shows a shape context with a scaled radius applied on the same location on the test image in (b). The second row illustrates the corresponding histograms for a single edge orientation. Note that the histogram in (c) is more similar to the histogram in (a) than the one in (b), which increases the likelihood of matching during detection. Figure is best viewed in color.

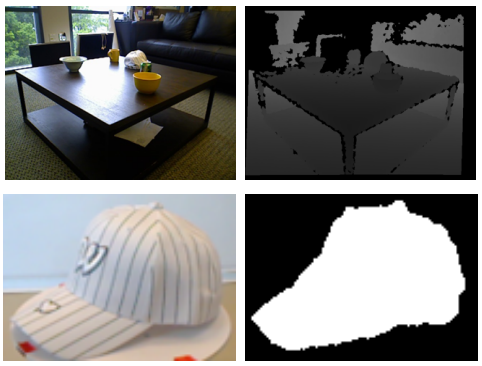


Fig. 5: The first row presents an example of our RGB-D test input, where the second row presents an example of an object training image with its provided segmentation mask.

Each feature $f_i = (\hat{d}_i, \hat{q}_i)$ of a 2D point i in the image space is defined by the shape context vector $\hat{d}_i$ and the 2D vector $\hat{q}_i$ that denotes the relative position of point i towards the center of the object. The set of features $F = \{f_1, f_2, ..., f_n\}$ extracted from training images of a particular object category represents the implicit shape model for the current object class.

Sampling for local descriptors during testing: Test images contain multiple-object instances besides background (see Figure 5) unlike training images that each consists of a single object of interest. Instead of dense sampling of edge points from the entire image (Figure 6a), we utilize the set $O = \{o_1, ..., o_K\}$ of generated object proposals described in Section III-A of the test image to guide the descriptor sampling. Each object proposal comes with a segmentation mask in the image. We define the set of segmentation masks for the object proposals to be the set $\{m_1, ..., m_K\}$. We synthesize the sampling area M simply by taking the union of all segmentation masks of object proposals:

$$M = \bigcup_{k=1}^{K} m_k \tag{1}$$

Area M in Figure 6b is the union of all coloured segmentation masks and this area is then used to filter the edge map from which we will extract our descriptors (Figure 6c). This does not only reduce the amount of descriptors but also removes edge points corresponding to background, which would otherwise serve as noise in the calculation of the shape contexts. Finally, we uniformly sample from the remaining edge points. We follow the same edge-map generation strategy as in our training step.

Objects in test images can have large scale variation compared to training images, which could lead to very different shape context representations. We avoid this by scaling the radius of the shape context using Equation 2, where $\hat{r_t}$ is the radius used in training, z_t is the median depth of the object in the training images, z is the sampled depth from the test image, and $\hat{r}$ is the scaled radius:

$$\hat{r} = \frac{\hat{r}_t z_t}{z} \tag{2}$$

Figure 4 illustrates an example where the scaling of the shape context radius helps the detection of a soda can.

Hypotheses Generation: Each descriptor extracted from the test image at location l is compared to all shape context vectors $\{\hat{d}_i\}$ from the training set F using the χ^2 distance and the K best matches are chosen. For each match, the relative position towards the center of the object in the training image $\hat{q}_i$ is used to cast a vote on the test image making a prediction about the object's center (see Figure 6d). Each vote is weighted using the matching score, accumulated at each location and are smoothed with a Gaussian kernel to create the heat map shown in Figure 6e. To adjust the scale difference that might occur, we scale $\hat{q}_i$ using Equation 3:

$$\hat{q} = \frac{\hat{q}_i z}{z_l} \tag{3}$$

where z is the median depth of the training image where f_i was extracted from, z_l is the sampled depth from location l, and $\hat{q}$ is the scaled relative position vector. We show the new vectors in Figure 6f. We prune the votes for which their location l_i and $l_i + \hat{q}_i$ lie on different object proposals:

$$vote_i = \begin{cases} 1 & l_i \in m_{k1},\ l_i + \hat{q}_i \in m_{k2},\ k1 = k2 \\ 0 & otherwise \end{cases} \tag{4}$$

where $vote_i$ is a variable indicating the validity of each vote. We consider a vote's position to be valid when it is towards the object center. Figure 6g shows the overlap of the votes with the proposal masks, Figure 6h presents the votes that survive the pruning, and Figure 6i illustrates the updated heat map. Local maxima in the updated heat map give the locations of the hypotheses with a certain score. We finally form the set of hypotheses $H = \{h_1, h_2, ..., h_J\}$. Each hypothesis consists of $h_j = (x_j, s_j, v_j)$. Here x_j is the

Preprint submitted to 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems. Received March 1, 2016.

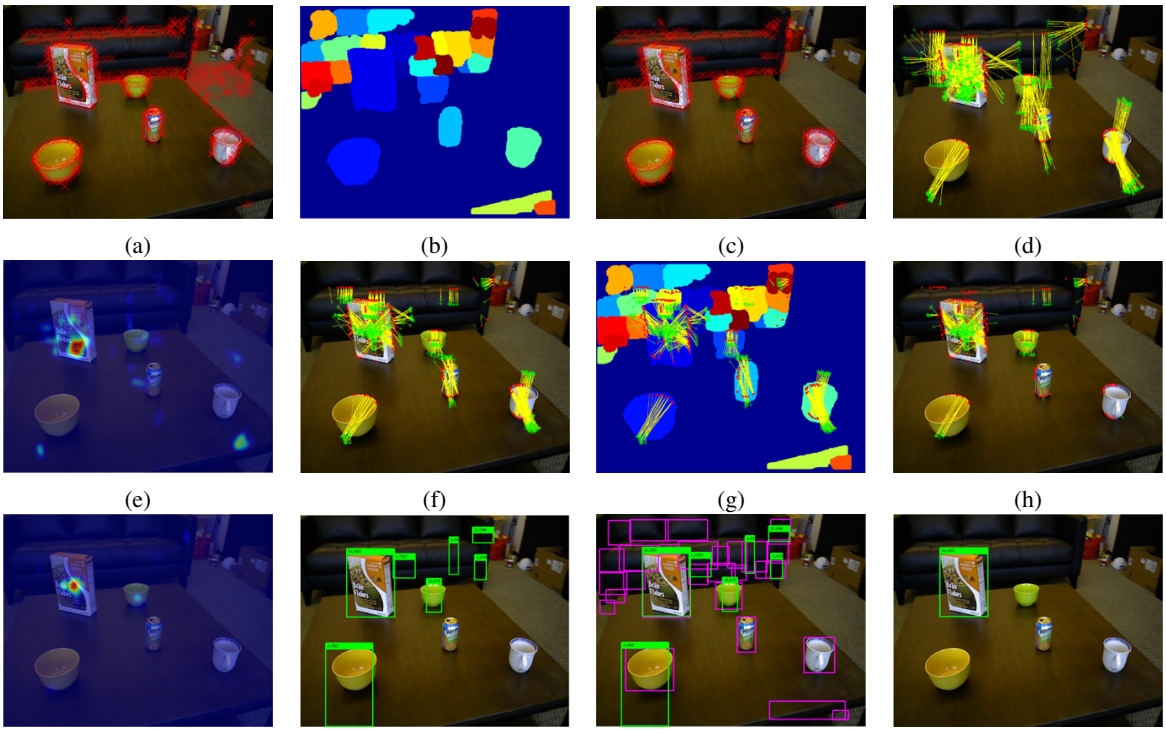


Fig. 6: Class specific object detection example where the object of interest is the cereal box. (a) depicts the initial sampling points from the test image which is then filtered (c) using the segmentation masks of the object proposals (b). We show the RGB image instead of the edge map for clarity. Votes cast after matching are shown in (d) and the resulting heat map is illustrated in (e). In (f) we scale the relative position of each vote $\mathbf{q_i}$ in order to adjust for the depth difference from the training images. (g) shows the overlay of the votes with the object proposals which we use to prune votes following equation 4. The remaining votes and their resulting heat map are depicted in (h) and (i) respectively. Notice the difference between the heat maps of (e) and (i). In (i) the object of interest is better localized with fewer wrong local minimums overall in the map. Finally, (j), (k), and (l) present the generated hypotheses, the verification of the hypotheses using the object proposals, and the remaining hypothesis respectively. The hypotheses are shown in green bounding boxes, and the object proposals in magenta. Figure is best viewed in color.

location of the hypothesis, s_j is the score, and v_j is the list of votes that contributed to the hypothesis. Figure 6j presents the formed hypotheses.

Verification using object proposals.: Given the set of hypotheses we want to keep those that most likely enclose an object. For each hypothesis we find the closest object proposal based on the overlapping area between the bounding box of the hypothesis BB_h and the bounding box of the object proposal BB_o. The overlap is calculated as the intersection over union: $IOU = \frac{area(BB_h \cap BB_o)}{area(BB_h \cup BB_o)}$. Hypotheses with $IOU < 0.5$ with their closest object proposal and with very small score are pruned. Figure 6k shows the hypotheses overlaid on the object proposals, while Figure 6l depicts the surviving hypothesis after verification.

Score normalization and multi-class prediction: Given the hypotheses generated from each class-specific object detector, we combine their responses to get a multi-class prediction over the objects in each frame. The score of each hypothesis depends on the number and the quality of its votes and depends on the size of the object. Each object model produces a different range of scores since each object category varies in size. In order to normalize the scores across all object categories, we get a distribution of the number of edge points for each category from the training images and normalize the scores as follows:

$$sn_j^c = \frac{s_j^c - \mu^c}{\sigma^c} \tag{5}$$

where s_j^c is the score of the j hypothesis for category c, and μ^c and σ^c are the mean and standard deviation respectively of the aforementioned distribution for category c. We then associate all hypotheses from all detectors with their closest object proposal in order to get a score for each proposal for each category. The proposal's label is determined by the category who has the highest normalized score. Proposals that are not associated with any hypotheses are labelled as background. For a single frame, each object proposal's $o_k = \{pc_k, m_k, w_k, y_k\}$ definition is now augmented with w_k which is the normalized score distribution over the object categories, and y_k which is the predicted category label. The evaluation of this single view strategy

is described in the experiments section, where we compare against sliding window techniques [7] and show superior performance.

C. Multi-view Object Detection

In the robotic setting we often have multiple viewpoints available as the robots move around. Our multi-view object detection approach consists of a data association step, where we link object proposals along the frame sequence into tracks, and a prediction refinement step where we update the probability distribution over the classes for each object proposal using Bayesian update rule. Given a sequence of N-frames in a video sequence, there are $|T|$ object-tracks each representing an object in the entire sequence.

We follow a greedy approach to create tracks of object proposals across the video sequence. We assume the first frame of the sequence to be the reference frame, and initialize a new track for each of its object proposals. When a new frame is observed, the 3D centroid of each object proposal is transformed to the new frame and associated with the closest 3D centroid in the reference frame. We do not accept associations when the distance between the 3D centroids is more than a small threshold. If an object proposal cannot be associated with any proposal in the reference frame, then a new track is initialized. Each track t will consist of a set of object proposals from different frames over the whole sequence.

Class distribution update: For each proposal o in a single-view frame, we normalize its score distribution to get the probability distribution $p(y)$. y is a random variable corresponding to proposal o defined over the category labels. We keep a probability distribution $p(C_t)$ for each track t, where C_t is a random variable associated with track t and it is defined over the object category labels. We begin with a uniform distribution for $p(C_t)$, and when a new proposal is associated to the track, we update $p(C_t)$ given the history of the predictions of the proposals $\{y^1, y^2, ..., y^n\}$ using Bayes rule:

$$\begin{aligned} p(C_t|y^{1:n}) &= \frac{p(y^n|C_t, y^{1:n-1})p(C_t|y^{1:n-1})}{p(y^n|y^{1:n-1})} \\ &= \frac{p(y^n|C_t)p(C_t|y^{1:n-1})}{p(y^n|y^{1:n-1})} \end{aligned} \tag{6}$$

where $y^{1:n}$ is a short-hand notation for $\{y^1, y^2, ..., y^n\}$. We use the first order Markov assumption and assume that $p(y^n|C_t, y^{1:n-1}) = p(y^n|C_t)$. In other words, we are computing the label C_t of the track t given the history of the detections up to the current frame. For the term $p(y^n|C_t)$ we use the probability distribution $p(y)$ of the associated proposal computed in the single-view frame n, and $p(y^n|y^{1:n-1}) = \sum_{C_t} p(y^n|C_t)p(C_t|y^{1:n-1})$ is the normalizing factor. We decide on the track category label that maximizes the posterior probability distribution $p(C_t)$, and assign this label to all relevant views of the track.

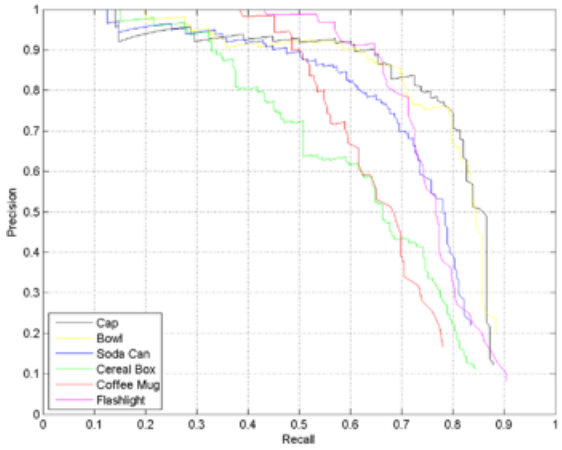


Fig. 7: Precision-recall curves for class-specific object detection obtained on the WRGB-D v1 scenes Dataset [1].

IV. Experiments

We evaluate our approach on the WRGB-D Object and Scenes Datasets: v1 [1] and v2 [2]. The WRGB-D v1 is divided in two parts. The first contains cropped images of 300 instances of objects of 51 categories, and the second is comprised of eight video scene sequences that contain some of these object instances seen from various viewpoints and with a considerable amount of occlusion. Recently, the second version of this dataset (v2) [2] was released containing fourteen larger video scenes, which offer more variability on the amount of viewpoints. We use the cropped images for our training, and evaluate on the video scenes.

We perform three experiments. Class-specific object detection is evaluated on scenes from v1, while single-view multi-class, and multi-view object detection are evaluated on v2 scenes. In all experiments we subsample the videos and run the detection on every 5th frame. In the experiment on the v1 scenes the objects of interest are *bowl*, *cap*, *cereal box*, *coffee mug*, *soda can*, and *flashlight*. In v2 scenes we use the same set of objects except *flashlight* since it is not present. The evaluation is being done at the bounding box level and more specifically we count as a true positive a bounding box with an Intersection over Union (IOU) larger than 0.5 with the ground truth.

Class-specific object detection: We evaluate each object detector individually on the v1 scenes and calculate their Average Precision (AP) as described in [35] at the level of object categories. We compare to two other methods which consist of sliding window detectors, and present superior performance in almost all categories. Figure 7 presents the precision-recall curves for each object category, while Table I shows the average precision comparison with the other methods. On average we achieve 9.5% increase in performance when compared to the Tang et al. [7] methods.

Single-view multi-class detection: Here we evaluate our multi-class predictions on individual frames. Using the hypotheses generated from the class-specific detectors we decide on a class label for each of our object proposals. We

	Bowl	Cap	Cereal Box	Coffee Mug	Soda Can	Flashlight	Average
Tang et al. [34](HOG)	51.6	33.3	21.4	54.1	**71.0**	32.1	43.9
Tang et al. [34](HH)	71.6	71.4	50.0	61.8	60.6	44.4	60.0
Ours	**75.1**	**74.5**	**61.2**	**62.8**	69.5	**73.6**	**69.5**

TABLE I: Illustration of class-specific object detection results obtained on the WRGB-D v1 scenes Dataset [1]. Comparison of Average Precision (%) for each object category and the average over all classes. HH is the combination of HOG and HONV, the feature that is introduced in [34].

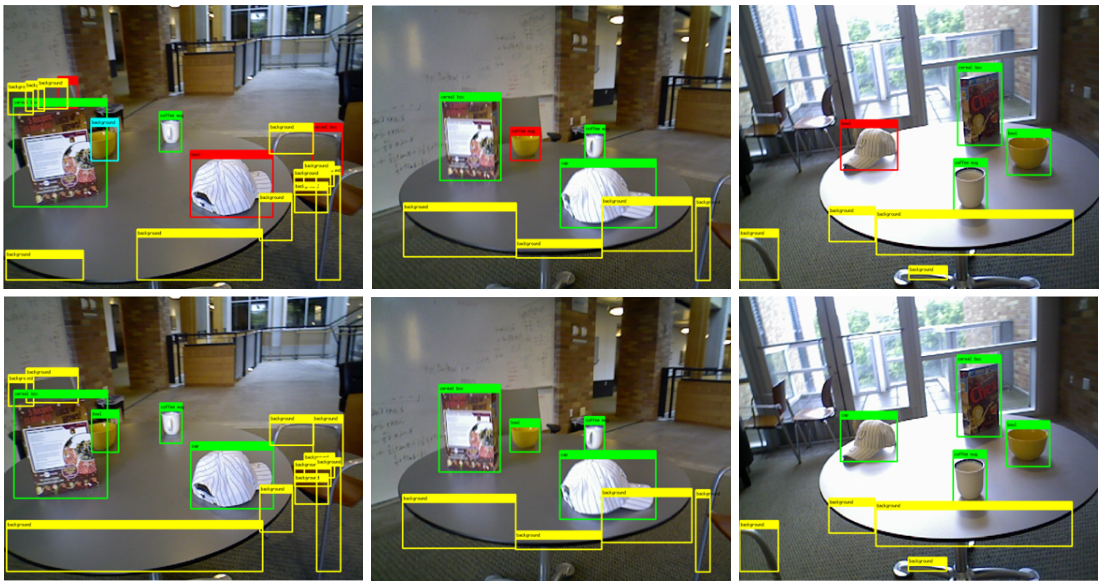


Fig. 8: A qualitative comparison between single-view and multi-view detections. The top row shows single-view detections while the bottom row shows multi-view detections for the same frames. The green bounding boxes signify correct detections, the yellow signify correctly identified background, the cyan signify missed detections, and the red signify false detections. Notice that in the single-view examples we have missing and false detections which are recovered and fixed respectively in the multi-view approach. For example, in the single-view detection of the first column, the bowl is missed, the cap is mislabelled, and two background hypotheses are labelled as objects. These incorrect detections are resolved in the multi-view detection. Figure is best viewed in color.

	Bowl	Cap	Cereal Box	Coffee Mug	Soda Can	Background	Average
Single-View							
Pillai et al. [28]	**88.6/71.6**	85.2/**62.0**	83.8/75.4	70.8/**50.8**	78.3/42.0	**95.0**/90.0	81.5/59.4
Ours	70.7/56.8	**87.2**/49.0	**84.6/83.3**	**83.7**/34.3	**85.6/55.6**	89.0/**98.1**	**83.5/62.8**
Multi-View							
Pillai et al. [28]	88.7/70.2	89.4/72.0	**95.6**/84.3	80.1/64.1	**89.1**/75.6	96.6/96.8	89.8/72.0
Ours	**92.7/89.8**	**96.9/81.0**	87.4/**97.8**	**88.4/87.0**	86.7/**84.2**	**97.3/98.0**	**91.6/89.6**

TABLE II: Precision/recall (%) results for single-view multi-class and multi-view detections run on the WRGB-D v2 [2] scene dataset. We compare against [28] who does not exploit the depth channel and uses object proposals generated from a reconstructed scene rather than a single frame. We show superior average results in both single-view and multi-view detections.

used an empirical threshold to discard low scoring predictions and report precision/recall for each object category in rows $2, 3$, and 4 of Table II. We compare our results to the state-of-the-art Pillai et al. [28] and achieve higher average performance, with an increase of 2% in precision and 3.4% in recall. Examples of single-view multi-class detections are illustrated in Figure 8.

Multi-view Object Detection: Finally, we investigate the performance of the multi-view object detection approach on the WRGB-D v2 scenes dataset [2]. In order to perform the data association step, we use the ground truth poses provided by the dataset. In an on-line settings this can be done by estimating the relative poses between the views using one of the state-of-the-art methods. The consistency of our object proposals from frame-to-frame allows for the creation of long tracks throughout the scene. We consider a track to be valid if it is present at least in 60% of the total number of frames in each scene.

Using the probability distribution over the categories for each proposal obtained from our single-view multi-

class detection, we refine our predictions through sequential Bayesian estimation as discussed in section III-C. We compare our results to the state-of-the-art Pillai et al. [28] which in contrast to our work, simply aggregates the detector's responses over multiple frames to make a prediction. We present superior results in rows $5, 6$, and 7 of Table II in almost all object categories and overall, with an increase of 1.8% in precision, and 17.6% in recall. We also notice a high increase in performance compared to the single-view detections. Missing detections in the single-view approach can be recovered from the history of previous observations, and similarly false detections get discarder. Figure 8 qualitatively compares single-view with multi-view detections and illustrates the strength of the latter over the former.

V. CONCLUSIONS

We have demonstrated a novel approach for multi-view object detection in RGB-D table-top settings. Our approach adopts a shape-based representation with a voting strategy to generate object hypotheses. We showed that in a single-view object detection many of the false-positive hypotheses by can be reduced by 3D object proposals generated from 3D point clouds using mean-shift clustering. We also demonstrated that the evidence from multiple views can further improve the object detection's performance. Our method achieves state-of-the-art performance in multi-view object detection on the WRGB-D scenes datasets. In the future we plan to take advantage of our multi-view detection strategy and 3D object proposals to facilitate the task of 3D point labelling of reconstructed scenes. In addition, more extensive experiments with a larger number of object categories in complex scenes will be conducted.

REFERENCES

[1] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in *in IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[2] K. Lai, L. Bo, and D. Fox, "Unsupervised feature learning for 3d scene labeling," in *in IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[3] L. Bo, X. Ren, and D. Fox, "Learning hierarchical sparse features for RGB-D object recognition," in *in International Journal of Robotics Research (IJRR)*, 2014.

[4] L. Spinello and K. O. Arras, "People detection in RGB-D data," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[5] B. S. Kim, S. Xu, and S. Savarese, "Accurate localization of 3d objects from rgb-d data using segmentation hypotheses," in *in IEEE Conference Computer Vision and Pattern Recognition (CVPR)*, 2013.

[6] A. Collet, M. Martinez, and S. S. Srinivasa, "The moped framework: Object recognition and pose estimation for manipulation," in *in International Journal of Robotics Research (IJRR)*, 2011.

[7] J. Tang, S. Miller, A. Singh, and P. Abbeel, "A textured object recognition pipeline for color and depth image data," in *in IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[8] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes," in *in IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 1999.

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

[10] M. B. R. B. Rusu, A. Holzbach and G. Bradski, "Detecting and segmenting objects for mobile manipulation," in *in IEEE International Conference on Computer Vision (ICCV), S3DV Workshop*, 2009.

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

[12] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," in *in IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2010.

[13] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," in *in International Journal of Computer Vision (IJCV)*, 2013.

[14] M. M. Cheng, Z. Zhang, W. Y. Lin, and P. Torr, "Bing: Binarized normed gradients for objectness estimation at 300fps," in *in IEEE Conference Computer Vision and Pattern Recognition (CVPR)*, 2014.

[15] A. Mishra, A. Shrivastava, and Y. Aloimonos, "Segmenting "simple" objects using rgb-d," in *in IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[16] A. Karpathy, S. Miller, , and L. Fei-Fei, "Object discovery in 3d scenes via shape analysis," in *in IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[17] M. Firman, D. Thomas, S. Julier, and A. Sugimoto, "Learning to discover objects in rgb-d images using correlation clustering," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[18] R. Triebel, J. Shin, and R. Siegwart, "Segmentation and unsupervised part-based discovery of repetitive objects."

[19] G. Mori, S. Belongie, and J. Malik, "Efficient shape matching using shape contexts," in *in IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2005.

[20] L. Wang, J. Shi, G. Song, and I. Shen, "Object detection combining recognition and segmentation," in *in Asian Conference on Computer Vision (ACCV)*, 2007.

[21] C. Teo, A. Myers, C. Fermuller, and Y. Aloimonos, "Embedding high-level information into low level vision: Efficient object search in clutter," in *in IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[22] B. Liebe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," in *in International Journal of Computer Vision (IJCV)*, 2008.

[23] J. Gall and V. Lempitsky, "Class-specific hough forests for object detection," in *in IEEE Conference Computer Vision and Pattern Recognition (CVPR)*, 2009.

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

[25] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and V. Gool, "Towards multi-view object class detection,," in *in IEEE Conference Computer Vision and Pattern Recognition (CVPR)*, 2006.

[26] M. Sun, H. Su, S. Savarese, and L. Fei-Fei, "A multi-view probabilistic model for 3d object classes," in *in IEEE Conference Computer Vision and Pattern Recognition (CVPR)*, 2009.

[27] E. Herbst, X. Ren, and D. Fox, "RGB-D object discovery via multi-scene analysis," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[28] S. Pillai and J. Leonard, "Monocular slam supported object recognition," in *Proceeding of Robotics: Science and Systems (RSS)*, 2015.

[29] K. Lai, L. Bo, X. Ren, and D. Fox, "Detection-based object labeling in 3d scenes," in *in IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[30] C. J. Taylor and A. Cowley, "Parsing indoor scenes using rgb-d imagery," in *Proceeding of Robotics: Science and Systems (RSS)*, 2012.

[31] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," in *in IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2002.

[32] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," in *in IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2004.

[33] P. Dollr and C. Zitnick, "Structured forests for fast edge detection," in *in IEEE International Conference on Computer Vision (ICCV)*, 2013.

[34] S. Tang, X. Wang, X. Lv, T. X. Han, and J. Keller, "Histogram of oriented normal vectors for object recognition with a depth sensor," in *in Asian Conference on Computer Vision (ACCV)*, 2012.

[35] M. Everingham, L. V. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," in *in International Journal of Computer Vision (IJCV)*, 2010.