Object Recognition and Segmentation in Indoor Scenes from RGB-D Images

Md. Alimoor Reza
Department of Computer Science
George Mason University
mreza@masonlive.gmu.edu

Jana Kosecka
Department of Computer Science
George Mason University
kosecka@cs.gmu.edu

***Abstract*—We study the problem of automatic recognition and segmentation of objects in indoor RGB-D scenes. We propose to formulate the object recognition and segmentation in RGB-D data as a binary object-background segmentation, using an informative set of features and grouping cues for small regular superpixels. The main novelty of the proposed approach is the exploitation of the informative depth channel features which indicate presence of depth boundaries, the use of efficient supervised object specific binary segmentation and effective hard negative mining exploiting the object co-occurrence statistics. The binary segmentation is meaningful in the context of robotics applications, where often only an object of interest needs to be sought. This yields an efficient and flexible method, which can be easily extended to additional object categories. We report the performance of the approach on NYU-V2 indoor dataset and demonstrate improvement in the global and average accuracy compared to the state of the art methods.**

I. Introduction

In the presented work, we study the problem of recognition and segmentation of objects in indoor scenes from RGB-D data. With the advent of RGB-D sensors, 3D scene understanding has gained considerable attention in the past few years in both computer vision and robotics communities. The existing methods vary in the choice of features, methods to generate object hypotheses and final classification and inference algorithms. The approaches which consider understanding of open indoor scenes and are typically evaluated on scene datasets such as NYU-V2 dataset by [15], while other methods focus on object detection and categorization in table-top settings [9] as in RGB-D Object Dataset from the University of Washington. These two scenarios have very different background clutter and different intra and interclass variations between the considered objects. We are interested in object recognition and segmentation of open indoor scenes. This problem is closely related to the general problem of semantic segmentation, where the goal is to simultaneously segment an image into meaningful regions and associate semantic labels with them.

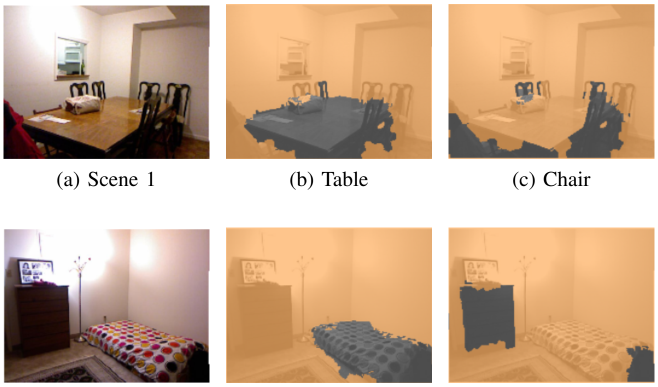

(a) Scene 1
(b) Table
(c) Chair
(d) Scene 2
(e) Bed
(f) Dresser

Fig. 1: Our recognition and segmentation output in the NYU V2 dataset. 6a is an example scene. The final segmentation for *Table* (an object of interest) is shown in 6b in gray color. If we seek for the *Chair* object on the same image, we get the segmentation in 1c. In 1e ,1f we demonstrated our recognition and segmentation results for *Bed*, *Dresser* respectively for another scene.

Proposed Approach: In the presented work, we subscribe to the segmentation strategy for object detection and categorization using elementary regions determined by efficient low-level superpixel segmentation. Instead of using multiple segmentations for generating a large number of object proposals or using computationally expensive grouping strategies for generating high quality bottom-up segmentations, we learn per object category grouping rules in the Conditional Random Field (CRF) framework. We endow SLIC superpixels which partition the image into regular sized regions with rich appearance and geometric features.We then proceed with the CRF learning of the two class figure-background segmentation for each object category, exploiting hard negative mining strategies guided by object class co-occurrence. We evaluate the approach on NYU-V2 dataset [15] demonstrating superior global and average accuracy and improving per class accuracies for several object classes. The accuracy of object recognition for each category is measured using Jaccard Index and evaluated on images where the object is present. The proposed approach exploits an alternative avenue for incorporating the top-down object class information and demonstrates simple yet efficient method for object recognition and segmentation exploiting the object presence information.

Related Work: Several semantic segmentation methods have been developed exploiting RGB-D data in indoor environments. In [8], authors highlighted the need for efficiency of the final inference and used up to 17 object classes and moderate variations in the scenes to evaluate their results. They were able to exploit stronger appearance and contextual cues due to the scale and the nature of the environment. More recently, more comprehensive experiments have been conducted on larger NYU-V2 RGB-D dataset introduced in [14]. Authors in [13] focus on local patch based kernel features achieved very good average performance while considering 13 structural and furniture classes and grouping all the smaller objects in 'other' category. The proposed features are computed over high quality superpixels obtained with a computationally expensive boundary detector [10]. In addition to inference of semantic labels in the work of [15], the authors simultaneously considered the problem of inference of the support relations between different semantic categories. The approach relied on elaborate pre-processing stage involving hierarchical segmentation stage, reasoning about occlusion boundaries and piecewise planar segmentation. All these stages required a solution to a separate inference problem, using additional features and stage specific energy functions. A different strategy is followed by [5] where the authors used the available depth information to improve the initial segmentation, followed by classification of obtained segments. In order to associate disconnected segments belonging to the same object category, they also propose a long-range amodal completion to improve the segmentation consistency. The above mentioned approaches relied on improvements of bottom up segmentation using the depth data as additional cue, following by classification of obtained regions. In the work of [4], authors bypass the complex feature computation and segmentation stage and use convolutional networks for semantic segmentation. The final hypotheses are then evaluated for the indoor scenes labeling task using the superpixel over-segmentation.

The problem of object recognition and detection has been also tackled using the commonly used sliding windows processing pipelines, where the window pruning strategies and the features have been adopted to RGB-D data. The commonly used approaches include [9][11][18]. The bounding boxes generated by the sliding windows approaches, while efficient, provide only poor segmentation and are often suitable for objects whose shape can be well approximated by a bounding box. Alternative strategies for object detection and semantic segmentation, proceed with mechanisms for generating object proposals from elementary segments using low-level grouping cues of color and texture. This approach has been pursued in the context of generic object detection followed by recognition in the absence of depth data by [17]. Strategy for generating and ranking multiple object proposals has been also explored by [3] and evaluated on MRSC dataset.

II. Approach

In the following sections we describe the ingredients of our approach starting with superpixel segmentation, feature computation and CRF learning and inference.

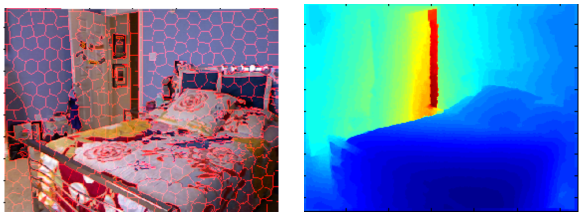


Fig. 2: SLIC superpixel and the depth image from NYUD V2.

We over-segment the RGB image using Simple Linear Iterative Clustering (SLIC) [1]. SLIC works by clustering pixels in the five dimensional color and image space by generating nearly compact and uniform superpixels. This transformation from the space of pixels to superpixels allows us to work with around few hundreds of superpixels per image as opposed to 480x640 pixels in the original image. Figure 2 shows an example image over-segmentation. The individual superpixels are marked by the red boundaries.

A. *Features*

Given the superpixels obtained in the previous stage, we endow them with representative features that capture local statistics of both geometry and appearance as well as level of agreement/alignment with the global geometry of the scene. Furthermore, we want the features to be sufficiently rich and discriminative to enable good disambiguation between different classes. We adopt a subset of geometric features proposed in [5], but in our setting the features are computed over small SLIC superpixels. The appearance is in our case captured by commonly used color and texture features as used in [16]. We also adopt efficient features indicative of the presence of occlusion boundaries introduced in [2].

1) Color feature: We represent the color distribution of a superpixel by a fixed histogram of color in HSV color space as done in [16]. For each channel, we extract 25 dimensional histogram and concatenate those to create a 75 bin histogram.

2) Texture feature: The texture is represented using oriented filters. Gaussian derivatives are taken in each of the eight orientations for each color channel in HSV and a 10 bin histogram is extracted for each orientation in a particular channel. The concatenation of these histograms over the three separate channels results in a histogram of 240 bins that represents our texture feature.

3) Geometric Features: From the work of Cadena et al. [2], we utilize the following geometric features computed from the 3D point cloud:

- Point cloud centroid (3): The mean 3D position of all the 3D points.
- Normal (3): The normal of the super pixel's 3D points estimated using singular value decomposition (SVD) based approach.
- Relative depth with the superpixels neighbors (2): The mean and standard deviation of the absolute value of the depth difference is computed. This value is captured only

when the 3D point cloud of the current superpixel is in front of the neighboring superpixels.

- Neighborhood planarity (1): The dot product of the superpixel normal with all the neighbors' normal are computed. The mean of these dot products captures how well neighboring normals are oriented relative to the superpixel under consideration. If all are oriented to the same direction (for example wall superpixels and its neighboring wall superpixels), then the mean would be higher (close to 1). The complement of the mean of the dot product is used as the neighborhood planarity feature.
- Superpixel planarity (1): The normal of the 3D points inside the superpixel and the distance offset. The distance offset is used as superpixel planarity feature.
- Vanishing direction entropy (1): This feature captures the entropy of the probability distribution of the superpixel boundaries oriented towards the dominant vanishing points. It is computed from the observation that the superpixel boundaries are often aligned with the dominant vanishing direction in indoor setting. We refer to the work of Cadena et al. for additional details [2].

4) Generic Features: We also adopt several generic and category specific features introduced in Gupta et al. [5], who achieved state-of-the-art results for semantic segmentation using hierarchical segmentation framework. Their generic features are computed from both the regions and their amodal completion, but used only in the context of superpixels only. We have adopted the following features:

- Orientation (6): The angular orientation of the superpixel plane normal with respect to the gravity direction, along with other orientation statistics such as the mean and median of normals computed for a superpixel's constituent pixels etc.
- Planarity (12): These features capture the planarity statistics such as mean and variance of the distance to the plane from the point cloud as well fractions of the points on the left/right of the estimated mean.
- Size/Area Features (9+10): These statistics capture the spatial extent, total area of the superpixel as well vertical and horizontal area of pixels.
- Clipping (5): Statistics capturing if the superpixel is clipped, fraction of the convex hull which is occluded etc.
- Orientation context (18): The mean, median (9+9) orientations of bounding boxes around a superpixel.

The final dimensions of our features can be found in Table I. The concatenation of these features represents a particular superpixel in a 386 dimensional feature space. In order to predict the class labels of superpixels, we train an AdaBoost classifier from features extracted on the training images. The probabilistic output from the AdaBoost is used as the local prior for the feature function of the data term in our CRF model.

feature type	description
C1	75 dim. histogram HSV values
T1	240 dim. histogram
G1	11 dim. feature of geometric features
G2	60 dim. feature of generic features

TABLE I: Summary of the features. C1, T1, G1, G2 denote color, texture, geometric, and generic features respectively.

B. Classifier

AdaBoost is a powerful technique of combining weak learners to provide a more complex hypothesis. In AdaBoost settings, we can learn a collection of n hypotheses, each has an associated weight α, which is learned during the training phase. Incorporating the Decision tree as weak learner has the inherent advantage of selecting the more relevent feature out of the high dimensional feature vector. In general, each decision tree separates the training data into desired partitions. We used the logistic regression version of AdaBoost proposed by the Hoiem et al. [7]. It gives the predicted output as confidence measure. Applying sigmoid function over this confidence gives a probabilistic output, which is very useful in classification task. We learn a collection of $\mathbf{T}$ decision trees, where weights of the nodes for each tree are learned according to the algorithm described in Hoeim et al. [7]. We learn the AdaBoost classifier for each of the object category e.g., *Bed*,*Table*,*Sofa* etc in a one-vs-all fashion.

If the object of interest is *Table*, we select all the images from the training set that contains the instances of table. The computed features produce an example in the high dimensional space. We label an example as an instance of the positive class (e.g., *Table* class) if more than 80% of the constituent pixels' labels belong to the positive class. The remaining data points are instances of the negative class (i.e., *Non-table* class).

In our experiments, we maintain approximately equal proportions of positive and negative data points during training an Adaboost classifier. We have more negative data points than positives for training and the trained model performance not only depends on the size of the data points but also on the type of selected samples. We applied a negative mining approach to sample from the large set of negative examples. We generate a joint distribution of object co-occurrence presence in the training images for each object of interests. Figure 3 shows the co-occurrence distribution for the object *Bookshelf*. Notice here that the object *Books* is one of the most frequent objects that co-occur with object *Bookshelf*. While training an Adaboost classifier, we pick negative samples from a particular object class in proportion to the object's co-occurrence value in the distribution. Similar context knowledge has been exploited previously in the context for designing the co-occurrence terms [12].

We evaluate the performance of the classifier on the images from a separate test set. Like the training stage, we select those images from the test set that contain the object of interest.

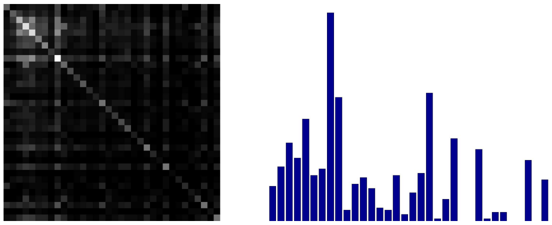


Fig. 3: a) 34x34 bject class co-ocurrence matrix (white entries denoting higher co-occurrence values); b) corresponding bookshelf row where two highest peaks belong to book and bookshelf categories.

Then we generate the positive and negative data points from those images. The metric we choose for evaluation is Jaccard Index (JI), which we discuss in the subsequent section. We use all the generated data points for evaluation unlike the training step, where we sample subset of negative data points. We learn 20 different Adaboost classifiers for 20 different randomly sampled negative subsets, and pick the one that gives us the best performance in JI for the positive class.

C. Conditional Random Fields

A Conditional Random Field (CRF) directly models the conditional distribution $P(\mathbf{y}|\mathbf{z})$ over the hidden variables $\mathbf{y}$ given the observations $\mathbf{z}$. In our settings, the hidden random variables $\mathbf{y} = \{y_1, y_2, ..., y_n\}$ correspond to the nodes in the CRF. Each hidden random variable can take two discrete values: $y_i = \{Object, Nonobject\}$, , where $Object \in \{Bed, Sofa, ..., Whiteboard\}$. We can factorize the conditional probability in CRF into a product of potentials. These potentials are functions that map a variable configuration to a non-negative number. The larger the value of a potential, the more the variable configuration conforms to the given observation. The CRF graph is denoted by $G = \{V, E\}$, where V, E are the graph nodes and edges respectively. The conditional distribution can be reformulated to represent the potentials in a log-linear combination of feature functions as denoted by Equation 1. We learn the weights w_1, w_2, w_3 from the labeled training images by maximizing the conditional likelihood. For the inference problem, we find the *maximum a-posteriori (MAP)* solution $\mathbf{y}$ for which the conditional probability $p(\mathbf{y}|\mathbf{z})$ is maximum.

$$p(\mathbf{y}|\mathbf{z}) = \frac{1}{Z(\mathbf{z})}\exp(w_1 \sum_{i}^{V} \theta_d(y_i, \mathbf{z}) + w_2 \sum_{(i,j)}^{E} \theta_{pc}(y_i, y_j, \mathbf{z}) + w_3 \sum_{(i,j)}^{E} \theta_{px}(y_i, y_j, \mathbf{z})) \tag{1}$$

$Z(\mathbf{z})$ is the partition function for normalization.

We use a unary potential $\phi_d(y_i, \mathbf{z})$ and two pairwise potentials: $\psi_{pc}(y_i, y_j, \mathbf{z})$ defined over the color space and $\psi_{pd}(y_i, y_j, \mathbf{z})$ defined over the Euclidean 3D space. Each of these potentials is defined in terms of its feature functions $\theta_d(y_i, \mathbf{z})$, $\theta_{px}(y_i, y_j, \mathbf{z})$, and $\theta_{pc}(y_i, y_j, \mathbf{z})$ respectively. We now discuss the feature functions in the following.

1) Unary Feature Function: The feature function for our unary potential is defined as the following function:

$$\theta_d(y_i, \mathbf{z}) = -log(P_i(y_i|\mathbf{z})) \tag{2}$$

Here $P_i(y_i|\mathbf{z})$ is the output of the Adaboost classifier from a set of observation $\mathbf{z}$. The observations are computed for each superpixel i using appearance, geometric, and generic cues obtained from the RGB and the depth image.

2) Pairwise Feature Function: We compute two pairwise functions for every edge in our CRF graph structure as defined in Equation 1. These pairwise functions ensure the smoothness while assigning labels over the superpixels. We penalize the labelings of two adjacent superpixels according to their color and spatial differences separately. The $\theta_{pc}(y_i, y_j, \mathbf{z})$ is defined over the LAB-color space, and computed using the following equation:

$$\theta_{pc}(y_i, y_j, \mathbf{z}) = \begin{cases} 1 - exp(-\|c_i - c_j\|) & , \quad l_i = l_j \\ exp(-\|c_i - c_j\|) & , \quad l_i \neq l_j \end{cases} \tag{3}$$

$\|c_i - c_j\|$ is the L2-norm of the difference between the mean colors c_i, c_j of the superpixels i, j respectively. l_i, l_j are the assigned labels for the superpixels.

The other pairwise function is computed on the edges in the graph from the difference of the superpixels' 3D positions:

$$\theta_{px}(y_i, y_j, \mathbf{z}) = \begin{cases} 1 - exp(-\|x_i - x_j\|) & , \quad l_i = l_j \\ exp(-\|x_i - x_j\|) & , \quad l_i \neq l_j \end{cases} \tag{4}$$

x_i, x_j are 3D positions of the superpixels i, j respectively and are computed from the depth data.

III. Results

In our experiments we used the NYU Depth V2 [15] dataset. It has 1449 RGB-D images consisting of 464 different indoor scene across 26 scene classes. The scenes contain multiple instances of objects. In total, there are 35064 distinct objects spanning across 894 different classes. For our experiments, we selected the 34 most frequent objects spanning across the Structure, Furniture, and Props super-categories as introduced by Silberman et al. [15]. These 34 object categories are also analyzed in the context of semantic segmentation task by Gupta et al. [5]. Gupta et al. considers 6 more objects of interest. They are *Wall, Ceiling, Floor, Rest-of-the-structure, Rest-of-furniture*, and *Rest-of-the-props* objects. The *Wall, Ceiling, Floor* are the categories that form the background in the indoor scenes, and we excluded those objects from our study. The range of objects in our analysis includes *Sofa, Chair, Blinds, Whiteboard, Lamp, TV, Dresser*.

We used the standard split of the NYUD V2 with 795 images for training and 654 images for evaluation as used by [5][15][2]. The training images are used to learn the CRF parameters w_1, w_2, w_3 and decision tree weights of the Adaboost. We evaluated the performance of our method against the methods of Gupta et al. [5], Ren et al. [13], and Silberman et al. [15] for the semantic segmentation task.

The qualitative results are shown in Figure 4, 5. Figure 4 shows the objects for which we have the good recognition and segmentation results. The last column shows our CRF segmentation in gray color. Figure 5 shows example images where the semantic segmentation did not work well. These cases include some small objects such as *Lamp* and *Box* that lie at a far away position from the camera. The *Clothes* object segmentation gets confused with the *Bed* surface since most of the time *Clothes* appear on the Bed in NYU-D V2, and *Clothes* has very similar appearance to the *Bed* surface. Other source of confusion comes from the inconsistency in labeling the object ground-truth, for example, the *Sink* object shown in Figure 5. Only the circular part in this image is labeled as *Sink* in the ground truth annotation; however there are also images in NYU-D V2 where the surrounding rectangular region including the circular part of the *Sink* object is labeled as *Sink* in ground truth annotation. The large intra-class variance for the *Bag* object makes it hard for recognition and segmentation.

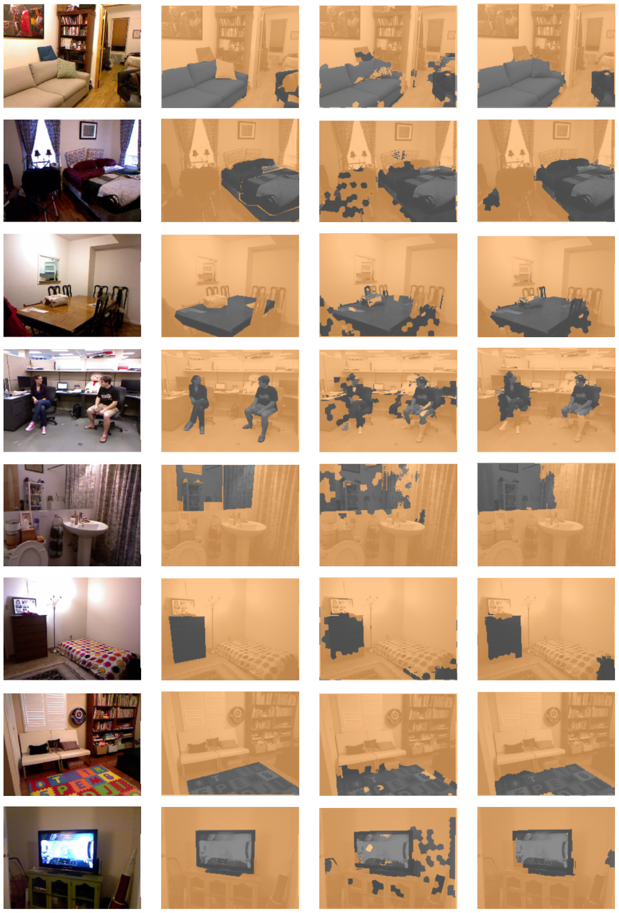


Fig. 4: The columns represent the original RGB image, ground truth, unary, and CRF output respectively from our approach in the NYU V2 dataset. Each row represents a case study for a particular object interest. From top to bottom, the objects of interest are *Sofa, Bed, Table, Person, Mirror, Dresser*, *Floor-mat*, and *TV* respectively.

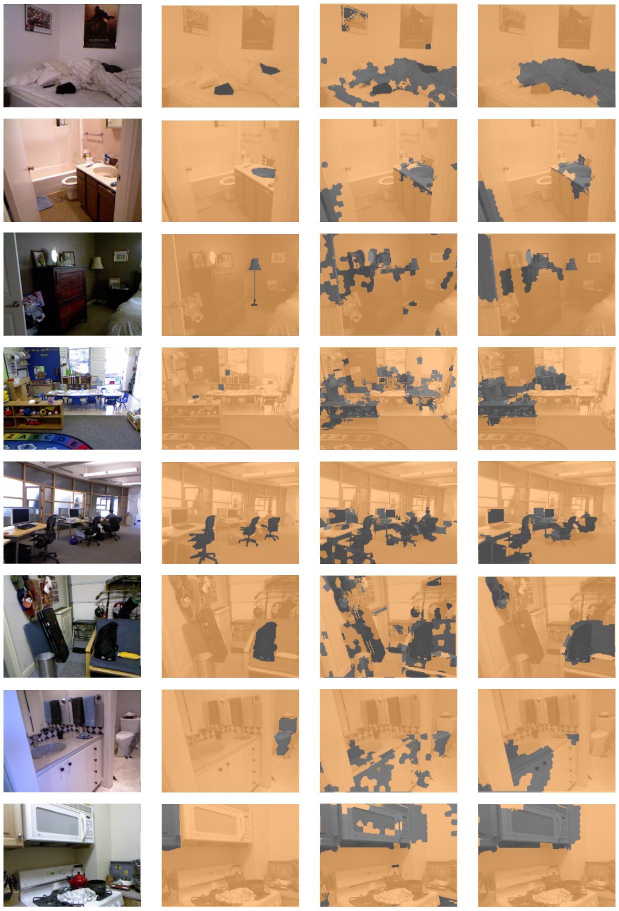


Fig. 5: The columns represent the original RGB image, ground truth, unary, and CRF output respectively from our approach in the NYU V2 dataset. Each row represents a case study for a particular object interest. From top to bottom, the objects of interest are *Clothes, Sink, Lamp, Box, Chair, Bag, Toilet*, and *Cabinet* respectively.

A. Evaluation

We used the Jaccard Index (JI) to measure our semantic segmentation performance. JI is a measure of true prediction divided by the union of true prediction and true labels as shown in Equation 5. Here P denotes the predicted label and G denotes the ground truth label. False alarm and missed values are both taken into account in this metric. We compute the JI for each category of objects from the final output of our CRF framework.

$$JI = \frac{|P \bigcap G|}{|P \bigcup G|} \tag{5}$$

The accuracy of object recognition for each category is evaluated on test images, where the object is present. We report the average Jaccard Index on all the test images, where the object is present. We compare the performance with the method of Gupta et al. [5], Ren et al. [13], and Silberman et al. [15]. The comparisons are shown in Table II.

Comparison with state-of-the-arts in JI metric: We have

	Bed	Sofa	Chair	Table	Window	Bookshelf	TV	Bag	Bathtub	Blinds	Books	Box	Cabinet	Clothes	Counter	Curtain	Desks
Silberman[15]	40.00	25.00	32.00	21.00	30.00	23.00	5.70	0.00	0.00	40.00	5.50	0.13	33.00	6.50	33.00	27.00	4.60
Ren[13]	42.00	28.00	33.00	17.00	28.00	17.00	19.00	1.20	7.80	27.00	**15.00**	3.30	37.00	9.50	39.00	28.00	10.00
Gupta[5]	55.00	**44.00**	**40.00**	**30.00**	**33.00**	20.00	9.30	0.65	33.00	**44.00**	4.40	**4.80**	**48.00**	6.90	**47.00**	**34.00**	10.00
Ours(unary)	50.64	37.44	25.00	19.19	25.93	23.88	26.40	**3.28**	32.12	29.77	9.17	2.89	27.42	9.79	34.68	25.59	21.04
Ours(CRF)	**56.85**	42.29	31.44	20.78	30.16	**30.29**	**34.97**	3.00	32.95	33.09	10.06	3.99	29.34	10.04	33.82	30.11	**23.35**

	Door	Dresser	Floor-mat	Lamp	Mirror	Night-stand	Paper	Person	Picture	Pillow	Refridgerator	Shelves	Shower-curtain	Sink	Toilet	Towel	Whiteboard
Silberman[15]	5.90	13.00	7.20	**16.00**	4.40	6.30	**13.00**	6.60	36.00	19.00	1.40	3.30	3.60	25.00	27.00	0.11	0.00
Ren[13]	13.00	7.00	20.00	14.00	18.00	9.20	12.00	14.00	32.00	20.00	1.90	6.10	5.40	**29.00**	35.00	13.00	0.15
Gupta[5]	8.30	22.00	22.00	6.80	19.00	20.00	1.90	16.00	**40.00**	**28.00**	15.00	5.10	18.00	26.00	**50.00**	**14.00**	37.00
Ours(unary)	14.72	32.35	32.81	6.68	23.09	16.22	7.64	19.54	17.93	16.16	16.86	10.67	25.54	10.98	26.06	7.62	36.25
Ours(CRF)	**17.16**	**35.73**	**34.19**	12.14	**27.41**	**21.54**	10.07	**30.31**	22.21	22.98	**20.59**	**13.46**	**26.84**	11.04	38.65	8.61	**37.69**

TABLE II: Performance on the NYUD-V2 dataset in Jaccard Index.

competitive or better JI scores for the individual objects compared to Gupta et al. [5]. Our method performed better for 19 objects out of the 34. We consistently outperform Gupta et al. [5] on small to medium sized objects e.g., *Lamp, Television, Dresser*. If we measure the average JI across the 34 objects, we improve the performance by more than 1.0% as shown in Table III.

	Mean JI
Silberman[15]	15.12
Ren[13]	17.99
Gupta[5]	23.92
Ours	**24.92**

TABLE III: Summary of comparison on NYU-D V2 in JI metric.

	Bed	Sofa	Chair	Table	Window	Books	TV
Couprie[4]	38.4	24.6	34.1	10.2	15.9	13.7	6.0
Hermans[6]	68.4	28.5	41.9	27.1	46.1	45.4	38.4
Ours	**87.8**	**86.1**	**82.7**	**78.0**	**78.1**	**71.7**	**81.8**

TABLE IV: Performance on the NYUD V2 dataset in per-class accuracy metric.

Comparison with other methods in per-class accuracy metric: We also compared our results with other methods that reported the performances of the semantic segmentation task using per-class accuracy metric. Couprie et al. [4] group objects into 13 semantic classes on the NYUD V2 dataset. They reported the per-class accuracy of an object that measures the proportions of correctly labelled pixels for each class. Hermans et al. [6] also reported the performance in per-class accuracy metric and showed that they can improve on 9 out of 13 classes. We also evaluated our method using per-class accuracy and compared the results on the 7 common objects[1] found in our experiments that are present in the group of 13 classes introduced by Couprie et al. [4]. We get superior performance on all the common 7 objects as shown in Table IV.

Ablation study: In order to gain a better understanding how much each type of features contributes to the final object recognition and segmentation task. We conducted the experiments by retaining only one set of feature at a time and computed the average JI of all the objects. We report the summary in the first two columns of Table V. Here we compared the performances of our final output (with or without CRF) in JI metric. Our pairwise terms used in CRF boosts the overall accuracy by 3% approximately in all of our studies as shown in Table V except the appearance features. Our appearance features (C1 and T1) individually do not perform as good as others. We also conducted an ablation study, where we incrementally added a set of features and analyzed how much they contribute to the final performance. The results are reported in columns three and four of Table V. Here we report only the performance for the unary term. The details performances of our system in Table VI. The generic features contributed to the performance significantly than others as shown in Table VI. We also analyzed the effect of the weights w_1, w_2, w_3 learned in our CRF settings. The learned weights are visualized in triplets as shown in Figure 6. For certain object categories e.g., *Night-stand, Paper, Person* all three weights have similar importance as shown in Figure 6 b. For some other object categories e.g., *Bed, Sofa, Window* the influence of the data term is significantly larger than the pairwise terms as depicted in Figure 6 a.

[1] Couprie et al. [4] introduced 13 categories which are formed by clustering. The remaining 6 categories are: Ceiling, Floor, Wall, Furniture, Deco., Objects. The first 3 of these are background categories (Ceiling, Floor, Wall). The remaining 3 categories are composed of multiple categories that formed into 3 clusters. In our setting, we are considering one object of interest, hence these are discarded.

	Bed	Sofa	Chair	Table	Window	Bookshelf	TV	Bag	Bathtub	Blinds	Books	Box	Cabinet	Clothes	Counter	Curtain	Desks
C1	25.19	20.06	17.68	6.86	17.92	21.66	27.67	1.72	7.60	19.01	9.22	2.13	14.46	9.08	4.87	10.29	14.43
T1	24.82	8.53	10.93	2.35	19.09	19.17	14.26	0.92	7.73	13.56	6.23	1.19	7.12	5.65	8.41	15.98	6.68
G1	49.15	33.27	23.38	18.16	17.30	22.61	8.55	2.84	19.38	25.53	7.05	2.81	20.78	10.92	26.12	23.27	18.32
G2	**58.28**	**43.72**	31.26	20.77	21.85	25.44	18.62	2.86	31.89	29.76	**10.44**	3.26	28.86	**10.26**	32.48	26.04	**24.15**
C1+T1+G1+G2	56.85	42.29	**31.44**	**20.78**	**30.16**	**30.29**	**34.97**	**3.00**	**32.95**	**33.09**	10.06	**3.99**	**29.34**	10.04	**33.82**	**30.11**	23.35

	Door	Dresser	Floor-mat	Lamp	Mirror	Night-stand	Paper	Person	Picture	Pillow	Refridgerator	Shelves	Shower-curtain	Sink	Toilet	Towel	Whiteboard
C1	7.09	30.72	9.19	3.69	8.95	11.24	4.66	13.22	4.74	5.84	8.07	8.23	8.46	2.60	10.96	4.07	29.52
T1	8.44	14.81	8.97	2.50	7.89	4.79	4.54	14.41	3.82	3.83	9.41	5.78	18.00	3.47	6.48	3.02	13.51
G1	15.63	28.77	31.10	7.86	23.38	17.97	7.00	25.58	16.97	13.21	14.69	10.29	27.08	8.64	26.29	7.26	28.45
G2	16.92	**41.47**	31.96	8.61	**28.23**	20.21	8.69	26.06	17.82	21.35	18.66	10.33	**32.97**	**12.00**	33.75	**9.61**	29.69
C1+T1+G1+G2	**17.16**	35.73	**34.19**	**12.14**	27.41	**21.54**	**10.07**	**30.31**	**22.21**	**22.98**	**20.59**	**13.46**	26.84	11.04	**38.65**	8.61	**37.69**

TABLE VI: Details of the our ablation studies on the NYUD-V2 dataset. Performances are reported in JI. C1, T1, G1, G2 denote color, texture, geometric, and generic features respectively. The reported scores are derived from the CRF inference output.

features	Unary	CRF	features	Unary
C1 only	10.72	**11.80**	C1+G2	20.94
T1 only	8.39	**9.01**	C1+T1+G2	20.95
G1 only	15.80	**18.81**	C1+T1+G1+G2	21.33
G2 only	19.51	**23.18**		
C1+T1+G1+G2	21.33	**24.92**		

TABLE V: Summary of the ablation study on NYU-D V2. Performances are reported in Jaccard Index (JI).

IV. Conclusion

We have described the basic components of the proposed approach for object recognition and segmentation in RGB-D data. While the individual technique of Conditional Random Fields on superpixels has been explored before, we have demonstrated that the choice of the inference problem (two class foreground-background segmentation for each object category), along with strong informative features capturing both appearance and geometric properties of the scene is effective and efficient, yielding superior or comparable global accuracy compared to the state of the art results. The choices were guided by the observations that in many practical robotic settings it is not necessary to generate complete semantic parsing of the scene. Obtaining categorization and segmentation of the objects of interest yields simple and more efficient inference problem. We have also demonstrated that one can explore the contextual relationships effectively, in our case in the hard negative mining stage for training the data term. From the ablation study, we also observed that our geometric feature has significant contribution towards the semantic segmentation task in our experiments.

The feature computation is currently the largest computational bottleneck of the approach for the on-line robotic operation. At present, we are exploring alternative representations of statistics of elementary regions and dependencies between geometric and appearance features. More extensive experiments with different datasets are underway. The assumption of our object's presence in the test images during evaluation could be replaced in the future by scene classification and object prior modeling.

Acknowledgments

This research has been supported by the US Army Research Office Grant W911NF-1110476.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and Sabine. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2012.

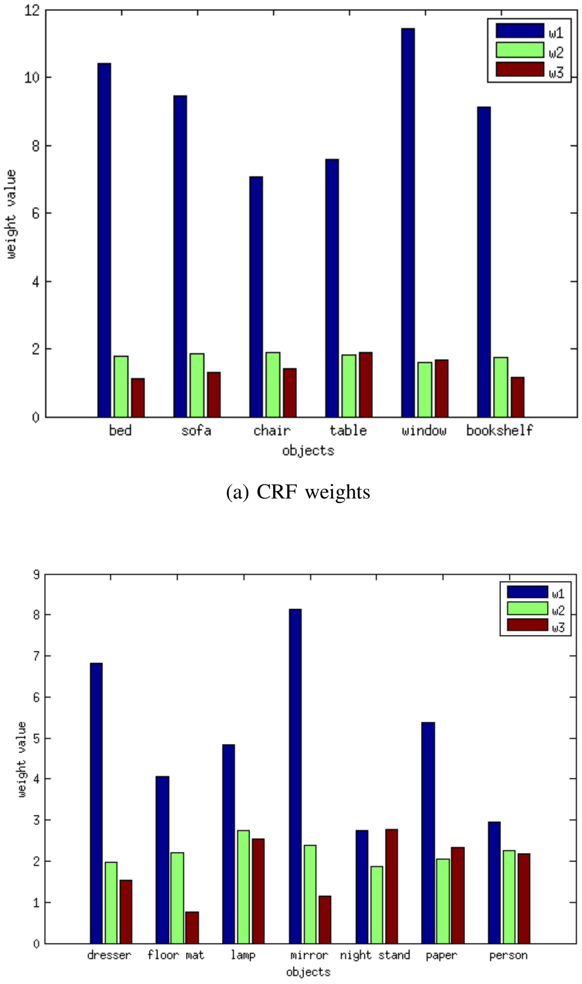
[2] C. Cadena and J. Košecka. Semantic parsing for priming object detection in RGB-D scenes. *International Conference on Robotics Automation (ICRA) - 3rd Workshop on Semantic Perception, Mapping and Exploration (SPME)*, 2013.

[3] J. Carreira and C. Sminchisescu. CPMC: Automatic object segmentation using constrained parametric mincuts. *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, 2012.

[4] C. Couprie, C. Farabet, L. Najman, and Y. LeCun. Indoor semantic segmentation using depth information. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2013.

[5] S. Gupta, P. Arbelaez, and J. Malik. Perceptual organization and recognition of indoor scenes from RGB-D images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[6] A. Hermans, G. Floros, and B. Leibe. Dense 3d semantic mapping of indoor scenes from rgb-d images. In *International Conference on Robotics Automation (ICRA)*, 2014.



(a) CRF weights

(b) CRF weights

Fig. 6: Importance of the learned weights for unary and pairwise terms in our CRF framework for all the objects.

[7] D. Hoiem, A. Efros, and M. Hebert. Recovering surface layout from an image. *International Journal of Computer Vision (IJCV)*, 2007.

[8] H. Koppula, Anand A., Joachims T., and A. Saxena. Semantic labeling of 3d point clouds for indoor scenes. In *NIPS*, 2011.

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

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

[11] B. Pepik, P. Gehler, M. Stark, and B. Schiele. 3D2PM - 3D deformable part models. In *In European Conference on Computer Vision (ECCV)*, 2012.

[12] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *ICCV*, 2007.

[13] X. Ren, L. Bo, and D. Fox. RGB-(D) scene labeling: Features and algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[14] N. Silberman and R. Fergus. Indoor scene segmentation using a structured light sensor. In *International Conference on Computer Vision (ICCV) - Workshop on 3D Representation and Recognition*, 2011.

[15] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision (ECCV)*, 2012.

[16] J. Uijlings, K. Van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *International Journal of Computer Vision (IJCV)*, 2013.

[17] K. van de Sande, J. Uijlings, T. Gevers, and A. Smeulders. Segmentation as selective search for object recognition. In *IEEE International Conference on Computer Vision*, 2011.

[18] E.S. Ye. Object detection in RGB-D indoor scenes. Master's thesis, EECS Department, University of California, Berkeley, 2013.