Accurate Localization of 3D Objects from RGB-D Data using Segmentation Hypotheses

Byung-soo Kim
University of Michigan
Ann Arbor, MI, U.S.A
bsookim@umich.edu

Shili Xu
University of Michigan
Ann Arbor, MI, U.S.A
stevenxu@umich.edu

Silvio Savarese
University of Michigan
Ann Arbor, MI, U.S.A
silvio@eecs.umich.edu

Abstract

In this paper we focus on the problem of detecting objects in 3D from RGB-D images. We propose a novel framework that explores the compatibility between segmentation hypotheses of the object in the image and the corresponding 3D map. Our framework allows to discover the optimal location of the object using a generalization of the structural latent SVM formulation in 3D as well as the definition of a new loss function defined over the 3D space in training. We evaluate our method using two existing RGB-D datasets. Extensive quantitative and qualitative experimental results show that our proposed approach outperforms state-of-the-art as methods well as a number of baseline approaches for both 3D and 2D object recognition tasks.

1. Introduction

The problem of detecting objects from images that are registered with depth maps (in short, RGB-D images) is receiving increasing interest in computer vision. This is coupled with recent widespread diffusion of depth sensors [1] which allows to accurately measure the distance between the camera and a point in 3D for each image pixel. Researchers have shown that the associated depth information can enhance detection performances [2, 3] and that, in general, the ability to reason in the 3D physical space provides critical contextual information that does facilitate object detection [4, 5, 6]. However, most of the existing approaches aim at localizing objects in the image and ignore the problem of estimating object location in the 3D space (we refer to this problem as to *3D object localization*) (Fig. 1). This capability is critical in applications related to robotics, object manipulation, safe driving and entertainment.

In this work we focus on the 3D object localization problem and propose a new method that is capable of jointly detecting objects in 2D images and the 3D physical space using RGB-D images. Instead of searching for objects in 3D as in [7], which is known to be computationally demanding and prone to false alarms, our approach leverages existing

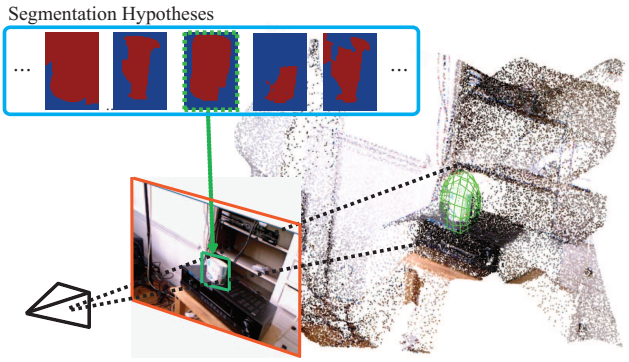
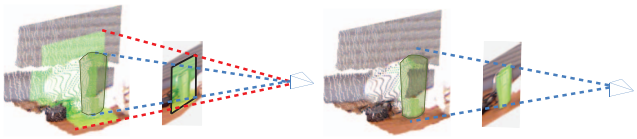


Figure 1: In this paper we propose a new framework to obtain accurate localizations of objects in 3D by exploring segmentation hypotheses of the object in 2D.

detection methods [8, 9, 10] which identify object proposals in the image by means of bounding boxes. Starting from these bounding box proposals, we introduce a novel framework that explores the compatibility between hypotheses of the object in the bounding box and the corresponding 3D map associated to the pixels within the bounding box. These object hypotheses are generated from foreground-vs-background object segmentation hypotheses within the bounding box. We call these Hypotheses object Foreground Masks (HFMs). The intuition is that the ability to combine appearance and corresponding depth values within the HFMs allows constructing more discriminative features for 2D and 3D localization than if such features are extracted from bounding boxes only (Fig. 2). Object models are learnt using a latent max-margin formulation whereby the latent variables are the object part locations in 3D. Features are extracted from appearance cues within the HFM and 3D descriptors computed on the associated 3D point cloud.

The deformation costs, or penalty costs, for the relative distance between object parts and the object root position, are calculated in 3D space, where a novel efficient 3D matching strategy is proposed. The proposed framework is illustrated in Fig. 3. This method is computationally in-

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.409



(a) 3D localization with a bounding box (b) 3D localization with the optimal HFM

Figure 2: (a) Accurate 3D localization (using RGB-D data) from a detected bounding box in the 2D image is challenging: the bounding box may include areas of the image that are not related to the foreground object and that correspond to different portions of 3D points in the RGB-D map that are located at completely different distances from the camera. This makes it hard to accurately localize the object position and pose in 3D. (b) In this paper we argue that by using segmentation hypotheses for the foreground object (the HFMs), we have the opportunity to identify points in 3D that are only relevant to the foreground object and therefore enable much more accurate 3D localization capabilities.

expensive compared to object detection schemes based on sliding bounding cubes in 3D space.

Related work and Contributions. Our overall approach of incorporating depth map to improve image recognition is related to several previous works [11, 3, 12, 2, 7, 13]. For example, [11, 12] built a CRF model using depth map, and showed that RGB-D is useful for indoor scene understanding. [3] used 3D features and obtained improvement in detection performance, and [2, 7] used 3D feature to achieve accurate 2D detection performance. [14] proposed depth map based kernel features for image classification. [13] proposed the method to detect object and localize objects in 3D from a RGB or RGB-D image. However, using RGB-D for modelling contextual segmentation or object recognition is still considered as a challenging problem.

The idea of associating detection and segmentation problems in 2D image is related to works such as [15, 16, 17, 18], where these problems are solved in a joint fashion. In these work, the benefit of a coherent reasoning about segmentation and detection is partially mitigated by high computational costs. In this paper, we use foreground segments as initial hypotheses as efficiently as in [19] and find out the optimal hypothesis using our novel formulation.

Authors in [7] tried to localize objects directly in 3D space using a simple bag-of-words model with linear weights within a branch-and-bound framework. However, the method is computational expensive since the search space is still large despite the efficiency gain achieved using branch-and-bound.

Our attempt to use a latent structural SVM formulation in 3D is clearly related to [8] as well as to recent work [10, 20] which propose to model an object as collection of 3D parts. The works [8, 10], however, focused on detecting objects in 2D images as opposed to RGB-D images as we seek to detect.

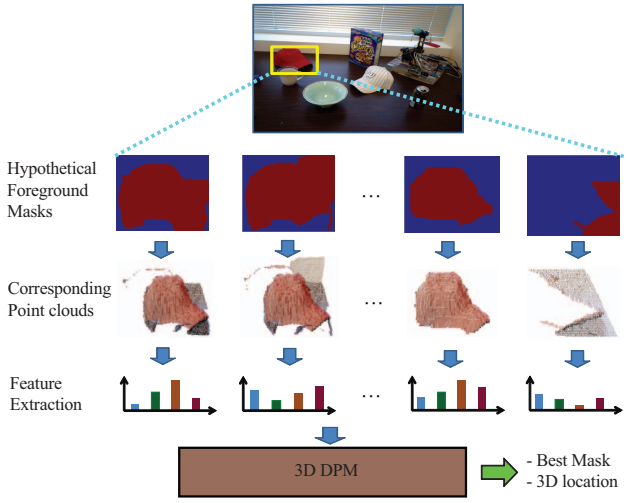


Figure 3: This figure shows the process of generating HFM and features from corresponding 3D point clouds. From each bounding box, multiple hypothetical object foreground masks are generated. For each mask, corresponding point clouds as well as features encoding 3D properties of point clouds are generated. From these features, the object's best foreground mask as well as its 3D location are estimated using our structural SVM formulation.

Contribution. Our main contributions are four-fold: *i)* we introduce HFM to help extract more descriptive 3D features, leading to a more robust 3D localization (Sec. 2.1); *ii)* we propose a novel matching process in 3D, integrating responses from deformable parts in 3D (Sec. 2.2.1); *iii)* we use our structural SVM scheme for joint 3D object localization and selection of the best segmentation hypothesis; finally, *iv)* we provide annotations for 3D object locations on top of existing RGB-D datasets (Sec. 3.1).

2. Accurate 3D Object Localization with Hypothetical Foreground Masks

In this section, we introduce our framework for accurate object detection and localization in 3D with RGB-D data from a single view. Our main idea is to use HFMs for achieving both efficiency and accuracy in 3D.

2.1. Hypothetical Foreground Masks

Bounding boxes. Bounding boxes have been widely used to generate hypotheses of object location in 2D from which features such as HOG can be extracted [8, 21]. The fact that a bounding box contains not only the foreground object but also the portions of the background scene is not necessarily an issue when it comes to object detection in 2D. The reason being that the appearance of the background is often correlated to the foreground object (think about a cow sitting on grass) and therefore the combination of the two can enhance object detection. This is much less of a case when RGB-D images are considered and features are extracted from both 2D and 3D point clouds. In such a case,

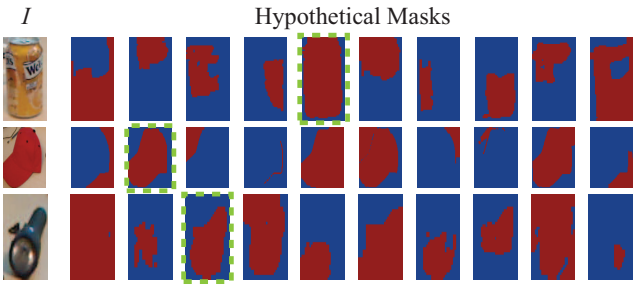


Figure 4: The first column is the RGB image inside bounding box. Remaining columns show top K foreground segmentation hypothesis (or, masks) when $K = 10$. The hypotheses highlighted with green lines indicate the segmentation hypothesis that is closest to the ground truth.

the 3D content associated to portions of a bounding box outside the foreground object can be fairly uncorrelated with the object and scattered in 3D space depending on the geometry of the background region (See Fig. 2(a)).

Hypothetical Foreground Mask. In this paper we propose to associate each bounding box hypothesis (a HBB) to a set of hypotheses for the foreground object segment (or mask) - the HFM. Specifically, each 2D HBB $y^{b,2D}$ with a height H and a width W is associated with an HFM $y^m \in \{0,1\}^{H \cdot W}$, which is a set of binary variables for all pixels where 1 indicates foreground pixels and 0 is for background. If the mask y^m tightly covers an objects itself, we can map the mask into 3D space as shown in Fig. 2(b).

Jointly estimating an accurate $y^{b,2D}$ and y^m is computationally more challenging than estimating $y^{b,2D}$ only. To resolve the problem, we narrow down the searching space for y^m using the top-K segmentation hypotheses (masks) provided by a state-of-the-art segmentation approach such as [19]. The typical results of top-K masks are illustrated in Fig. 4. To this end, we introduce an auxiliary indexing variable i_m where $y^m_{i_m}$ indicates i_mth mask among K masks.

Feature Extraction. From a HFM and the associated HBB, we extract two types of features. First, we extract 3D features from the projected 3D point clouds within the HFM. Designing a 3D feature is out of scope for this paper; for our work, we used the modified version of features introduced in [14], which capture 3D properties such as size, norm, etc. Details of our implementation can be found in Sec. 3.2. Refer to [22, 23] for examples of possible features that can be used along with our framework. On top of that, HOG features are extracted from a HBB and concatenated with 3D features.

3D Localization. We localize object in 3D space by projecting pixels within the HFM into 3D points to produce accurate localization results. Fig. 2 (a) and (b) show localization results from an estimated HBB and HFM, respectively. As the figure shows, when the correct HFM is used, the corresponding 3D point cloud enables much more accurate localization results than if an HBB is used in isolation. In Sec. 3.5, we quantitatively and qualitatively show that the proposed scheme significantly improves the 3D localization performance.

2.2. Part Based Model in 3D

Inspired by the deformable part based model (DPM) presented in [8] which estimates object bounding boxes and their latent part locations in the 2D image, our framework determines the optimal $3D$ location of the object $y^* = (y^{b,2D*}, y^{m*}_{i_m})$ as well as its parts location h^* in 3D as $(y^*, h^*) = argmax_{(y,h)} \langle \beta, \Psi(I, y^{b,2D}, y^m_{i_m}, h) \rangle$. The feature vector $\Psi(I, y^{b,2D}, y^m_{i_m}, h)$ concatenates features for M components of the mixture model, which encode 2D and 3D appearance cues, 3D distances between root and part filters and a offset value. The linear classifier β is learned using the Structural LSVM framework (Sec. 2.2.2).

2.2.1 3D Matching

The procedure that is used to estimate the root and part location in 2D is referred to as *matching* [8], which takes into account the 2D Euclidean distance between filter locations [24]. In contrast, our framework searches for the best 3D root and part locations, and this process is referred to as *3D matching*. By looking at 3D distance between root and part filters, this process suppresses false alarms in object part localization if the 3D distance between root and part is large, even they are close in the 2D image. As a result, possible false alarms in 2D matching results (Fig. 5(a)) are removed by our 3D matching strategy (Fig. 5(b)).

In details, our 3D matching mechanism involves the following steps. First, we project response maps of the filters into the 3D point cloud by associating a confidence value of a pixel to its corresponding point in 3D. Then, we define a score function which is obtained as the summation of the root and parts responses, with respect to their deformation costs in 3D. This score function gives a highest score at its optimal location and is expressed as follows:

$$\begin{aligned} \text{score}(x_0, y_0, z_0, l_0) &= R_{0,l_0}(x_0, y_0, z_0) \\ &+ \sum_i D_{i,l_0-\lambda}(2(x_0, y_0, z_0) + v_i) \end{aligned} \quad (1)$$

$R_{i,l}(\cdot)$ is filter responses projected into 3D space for a part i at scale l. The variable i indicates the part i if $i > 0$, or it indicates root if $i = 0$. v_i is the relative anchor position for the part i. λ is the scale difference between root and part filters. The transformation $D_{i,l}(\cdot)$ allows modelling the spatial uncertainty in parts location in 3D by balancing the part resposes $R_{i,l}(\cdot)$ and displacement cost $d(\cdot)$ as follows:

$$D_{i,l}(x, y, z) = \max_{dx,dy,dz} (R_{i,l}(x + dx, y + dy, z + dz) - d(x, y, z)) \quad (2)$$

where $d(x, y, z) = d_i \cdot \phi(dx, dy, dz)$ is the weighted Euclidean distance.

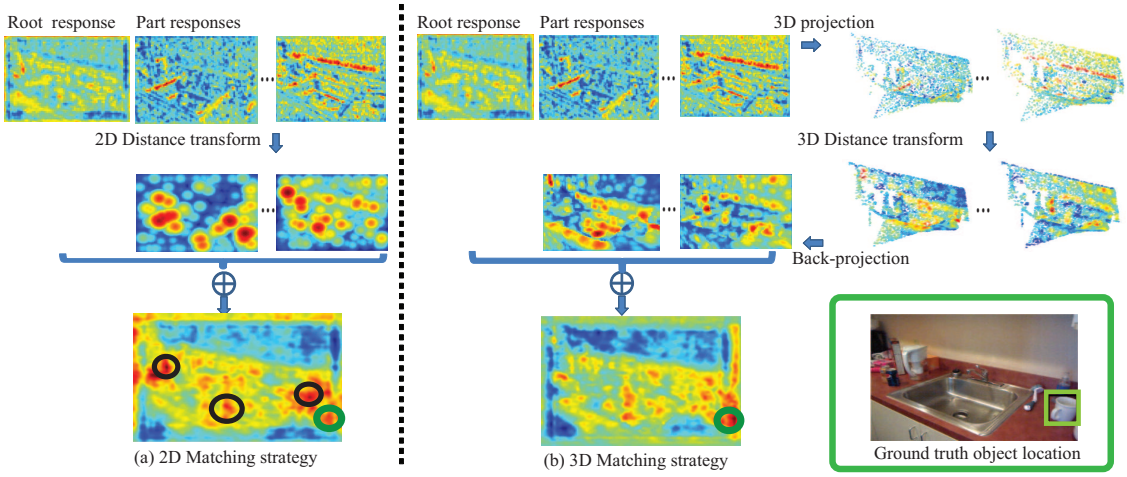


Figure 5: This figure shows a comparison between our 3D matching strategy (Fig.(b)) and the traditional 2D matching [8] (Fig.(a)). By using our 3D matching strategy, possible false alarms (black circles in Fig.(a)) can be suppressed if 3D distances between root and part filters are large. For 3D matching, part responses are firstly mapped into 3D space, and 3D distance transform is applied to efficiently calculate deformation costs between root and part filters. Details for the 3D matching can be found in the text.

Calculating $D_{i,l}(\cdot)$ over 3D space is computationally expensive and takes $O(N^3)$, where N is the size of the searching space for 1D. Note that [24] showed that this transformation can be efficiently calculated in the 1D case for a quadratic cost function. For 3D matching, our cost function is the 3D Euclidean distance, which is a quadratic function over (x, y, z). Thus, we can efficiently obtain the transformation in 3D by iteratively applying the 1D distance transform as follows:

$$\begin{aligned} D_{i,l}(x,y,z) &= \max_{dx',dy'} (R_{i,l|dz'}(x+dx, y+dy) - d_{|dz'}(x,y)) \\ &= \max_{dx'} (R_{i,l|dy',dz'}(x+dx) - d_{|dy',dz'}(x)) \end{aligned} \quad (3)$$

which makes computational time into $O(N)$.

Once the root location is found in 3D, parts locations also can be found by looking up the optimal displacements, similar to the 2D case [8].

2.2.2 Structural LSVM in 3D

To train model weights β, we propose to use Structural Latent SVM (StLSVM) framework [25] by considering 3D objects locations to construct the labeling space. This can improve the precision of decision boundaries of trained classifier since it penalizes inaccurate 3D localization predictions during the training process. In the following, we describe how the labeling space in 3D is formulated, and also introduce a loss function that penalizes inaccurate 3D localization predictions.

Labeling Space with Foreground Mask and Associated 3D Ellipsoid. Our training data is equipped with object class label y^l and the object foreground mask y^m, *i.e.,* $y = (y^l, y^m)$. To help associate the mask with 3D locations, we use y^s which is equivalent to y^m with different parametrization; $y^s = \{(u_1, v_1), ..., (u_S, v_S)\}$ is indicating pixels of the object foreground mask where $y^m(u, v) = 1$. S is the number of pixels belonging to the foreground region. $y^l \in \{-1, 1, ..., C\}$, where $1, ..., C$ indicates the class of the depicted object or -1 indicates background. The location of 2D bounding box ($y^{b,2D}$) is determined from y^s by retaining the minimum and maximum indices over the image axes u and v. On top of that, we obtain 3D object location by projecting y^s to point clouds $y^{s,3D}$ as follows:

$$\begin{aligned} y^{s,3D} &= g(y^s, \text{Depth}, \text{Camera}) \\ &= \{(u'_1, v'_1, z'_1), ..., (u'_S, v'_S, z'_S)\} \end{aligned} \quad (4)$$

where $g(\cdot)$ is the projection function given the depth map and camera parameters. (u'_i, v'_i, z'_i) is the 3D location of a point cloud. We use 3D ellipsoids in order to identify the point cloud $y^{s,3D}$ which identify an object in 3D space. As we will discuss in Sec. 3.1, ellipsoids are more convenient (than bounding cubes) for annotating objects in 3D. 3D ellipsoids are characterized with 9 parameters as follows:

$$\begin{aligned} y^{b,3D} &= \text{Ellipsoid}(y^{s,3D}) \\ &= [c_x, c_y, c_z, v_1, v_2, v_3, d_1, d_2, d_3] \end{aligned} \quad (5)$$

where $\{c_x, c_y, c_z\}$ is the center, $\{v_1, v_2, v_3\}$ are the 3 major axes, and $\{d_1, d_2, d_3\}$ are radii of the ellipsoid.

Training. The training data is $\{(I_i, y_i)\}_{1,...,N}$, where $\{I\}$ is the set of images, and $\{y_i = (y_i^l, y_i^s)\}$ are labels. The model learns the parameters β by solving the following latent max-margin optimization problem,

$$\begin{aligned} \min_{\beta,\xi} \quad & \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N}\xi_i \\ s.t. \quad & \forall i, I_i, \bar{y} \neq y_i : \max_{h_i} \langle \beta, \Psi(I_i, y_i, h_i)\rangle \\ & - \max_{h} \langle \beta, \Psi(I_i, \bar{y}, h)\rangle \geq \Delta(y_i, \bar{y}) - \xi_i \end{aligned} \quad (6)$$

where $\bar{y}$ is the most violating prediction, Ψ is the concatenated feature from a 2D BB and a HFM, and $\sum_{i=1}^{N} \xi_i$ is the sum of margins for the violated terms. Note that, since y contains information about the 3D ellipsoid location, it is able to take the 3D localization accuracy into account for designing the loss function $\Delta(y_i, \bar{y})$ for the training process.

Finding the Most Violating Sample. Obtaining the most violating sample $\bar{y}$ is computationally inefficient if we infer $\bar{y}^m$ with $2^{H \cdot W}$ binary variables, where H and W are the bounding box height and width, respectively. Instead, we resolve this problem. i_m which corresponds to the most violating $\bar{y}^m_{i_m}$ among $\forall i_m \in 1 \sim K$. This requires that a pre-processed HFMs $\{y^m_{i_m}\}$ are available, which is true in our case.

Loss Function in 3D. We design the loss function $\Delta(y_i, \bar{y})$ depending on both 2D and 3D localization accuracies. Similar to [10],

$$\Delta(y_i, \bar{y}) = \begin{cases} 0, & \text{if } y_i^l = \bar{y}^l = -1 \\ 1 - [y_i^l = \bar{y}^l] \frac{A(y_i \cap \bar{y})}{A(y_i \cup \bar{y})}, & \text{otherwise} \end{cases} \quad (7)$$

$A(y_1 \cap y_2)$ and $A(y_1 \cup y_2)$ are the intersection and union of two object locations, respectively. To take into account both 2D and 3D localization accuracy, we propose to use the following intersection and union re-weighted over 2D and 3D.

$$A(y_1 \cap y_2) = w_1 A(y_1^{b,2D} \cap y_2^{b,2D}) + w_2 A(y_1^{b,3D} \cap y_2^{b,3D}) \quad (8)$$

$$A(y_1 \cup y_2) = w_3 A(y_3^{b,2D} \cup y_2^{b,2D}) + w_4 A(y_1^{b,3D} \cup y_2^{b,3D}) \quad (9)$$

where $A(y_1^{b,2D} \cap y_2^{b,2D})$ is the intersecting area between two 2D bounding boxes, and $A(y_1^{b,3D} \cap y_2^{b,3D})$ is the intersecting volume between two 3D ellipsoids[1]. The union $A(y_1 \cup y_2)$ is calculated in a similar fashion. During the experiments, we set $w_{1,2,3,4} = 0.5$.

3. Experiments

In the following, we evaluate our framework on the Washington RGBD (WRGBD) dataset and the Berkeley 3D Object (B3DO) datasets. To provide an accurate ground truth 3D locations of objects for both training and testing, we propose an annotation procedure that allows to efficiently annotate an object foreground mask and the associated 3D ellipsoid (Sec. 3.1).

3.1. Annotation

Among the existing 3D datasets [2, 3, 11, 27], none of them provide accurate location of objects in 3D space (with the expection of [27]). [2, 3] annotated locations of objects in 2D space. [11] contains small object instance annotations, but the emphasis is more on providing annotations for the room layout. [27] include range data along with accurate location with 3D cubes for outdoor scenes.

In our work, we parameterize object locations using 3D ellipsoids. 3D ellipsoids are good to capture the size of the object using the 3 major axes of the object, and also describe objects' location in 3D space accurately. Also, as described next, they can be used for providing a ground truth 3D object location's and pose's annotations more accurately and efficiently than bounding boxes do. At that end, we have created an easy-to-use and efficient labeling tool. Using this tool, the annotator can simply draw a polygon capturing the object foreground, the 3D points corresponding to the pixels enclosed by the polygon are used to calculate the centroid and the principal axes of the ellipsoid tightly enclosing such 3D points. Principal axes are calculated using PCA on the point cloud. Statistics related to our annotated ellipsoids and its comparison with other statistics can be found in the supplementary material [26]. Note that, to ensure the quality of the annotation, annotators are asked to exclude any pixel from backrgound region. Also, the annotation tool allows user to immediately see the annotation results using a 3D visualization tool, so that they can annotate again if there is an error. Typical examples of the annotation results can be found in the second column of the Fig. 10. In our framework, the overlap ratio between ground truth and estimated ellipsoids are used to calculate the loss function for training the StLSVM model, as well as for evaluating 3D localization performance.

3.2. Implementation Details

As for the experiments with the B3DO dataset, we concatenated HOG features calculated from deformable parts [8] with 3D features proposed in [14]. As for the experiments with the WRGBD dataset, we further concatenated HOG features extracted from depth map as proposed by [3].

3.3. Foreground Mask Accuracy

There is a trade-off between the computational complexity and the number of hypothetical masks. By using a larger number of hypotheses, there is a higher chance to pick up the correct one. This is at the expense of the added computational time that is required to calculate features and apply the object model.

We measured a F-measure[2] for different number of HFMs (See the supplementary material [26] for the result.). When the number of HFMs is greater than 10, the performance gain becomes negligible. Thus, we set the number of hypothetical masks K to 10 for the experiments.

[1] See the supplementary material [26] for the method to calculate the intersecting volume between two ellipsoids.

[2] $F = \frac{2RP}{P+R}$, where P and R are the precision and recall of pixels in a segment relative to the ground truth [28].

3.4. Berkeley 3D Object Dataset

We first evaluated our method with the Berkeley 3D Object Dataset (B3DO) [2]. Among all the available object classes in the dataset, we tested 8 classes for which [2] evaluated the performance of 2D localization. 3D localization was not tested in [2], so we propose several baseline methods in Sec. 3.4.1. Our framework is compared against these methods.

3.4.1 3D Detection Performance

Similar to the Pascal Challenge criteria in 2D [29], the 3D localization is counted as correct if the overlapping volume between estimated ellipsoid and the ground truth ellipsoid is more than a threshold. Otherwise, it is counted as wrong. In our experiments, we set the threshold to be 25%.[3] We compare our method with the following baselines methods.

DPM+FillMask. For a detected 2D bounding box, we project all the pixels inside that bounding box. The ellipsoids are generated so as to enclose all the corresponding 3D points.

DPM+1stMask. Among K hypothetical masks, we choose the top-ranked mask from [19] as a foreground mask. The score corresponding to that mask is used to evaluate the detection.

DPM+SizePrior. 3D location and the size of the object is estimated based on statistics for each object category. In specific, a center of the 3D location for the object is set to the mean depth value inside the bounding box. The size of the object in 3D (width, height, thickness) is set to the average size of objects for the object category collected from the training set.

Results. Fig. 6 and Table. 1 shows the average precisions of 3D localization results of proposed method. Fig. 6 compares the performance of our method against baselines methods. Our method achieved the best performance for 7 out of 8 categories, and on average, it attains at least 6.2% higher average precision than baseline methods. For the class *cup*, DPM+SizePrior and our method achieve similar performance. The reason may stem from the fact that since there is small variance in the size of objects in the *cup* class, DPM+SizePrior can successfully capture its 3D location well. On the other hand, for classes having large variances in their depth due to different poses (for example, *monitor* or *keyboard*), our method works better than all baselines. Typical 3D localization results can be found in Fig. 10.

Table. 1 describes the effects of different components of the framework. While there are remarkable improvements by using features from HFMs and 3D loss function, the boost obtained by using the 3D matching strategy is relatively small. This may be due to the small intra-class variance in the B3DO dataset.

[3]While 2D detection often use 50% as a threshold, 3D localization is more challenging and 25% is a reasonable threshold for evaluation. For more details, see the supplementary material [26].

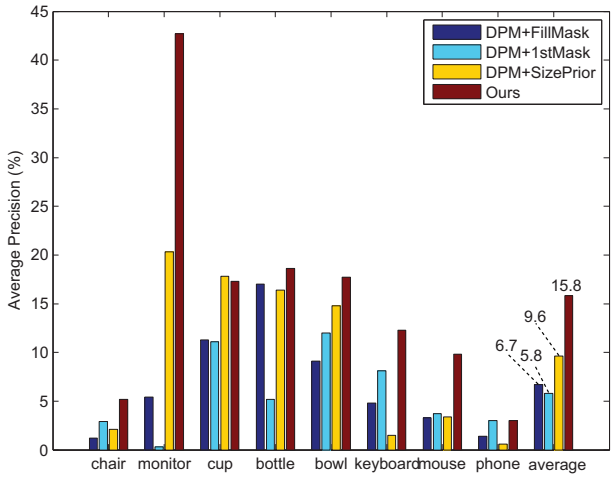


Figure 6: Average precisions of 3D object localization for 8 classes in B3DO dataset. Our method achieves best results compared to a number of baselines (See text for details).

	FM/3dLoss	FM/3dMatching	FM/3dMaskFeat	FM
B3DO	12.7%	15.5%	8.8%	15.8%
WRGBD	35.1%	34.2%	19.0%	35.6%

Table 1: This table compares the effect of different components of the model. *FM* refers to our full model, and first three columns are the accuracy without using 3D loss function, 3D matching process, and 3D Feature from HFM, respectively. See more discussion in Sec. 3.4.1

3.4.2 2D Detection Performance

We further show that our method improves 2D detection accuracy. Fig. 7 shows the average precisions of various detection results in 2D using the B3DO dataset. We compare our performances with DPM [8] and two methods proposed in [2]. The first method is called *pruning*, where detected results are pruned out if the approximated object size (bounding box diagonal times mean depth) is different from the statistics of the dataset. The second method is called *rescoring*, in which linear SVM is trained with additional features of approximated object size [2].

Note that we achieve better results for 6 out of 8 categories. This confirms that using HFM and associated 3D features is beneficial even for a 2D detection task. Notice that there is no improvement for the *chair* category. This may be due to severe occlusions that occur for the *chair* category in the dataset and that are not well characterized using our model.

3.5. Washington RGB-D Object Dataset

We also evaluated the proposed method using the Washington RGB-D Object Dataset (WRGBD) [3]. Note that in

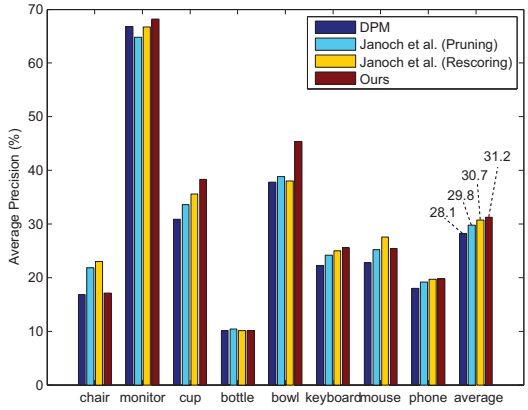


Figure 7: Average precisions of 2D object localization obtained by using DPM, the methods proposed in [2] and our method on B3DO dataset. Our proposed method consistently achieves better average precision over [2].

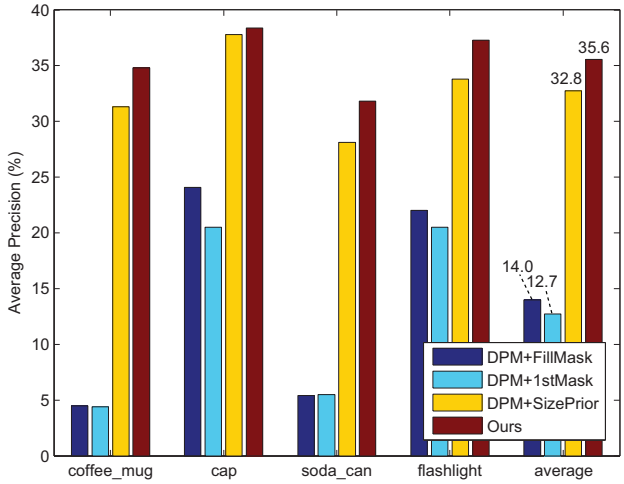


Figure 8: Average precisions of 3D object localization in the WRGBD dataset. Our method achieves best results compared with all baselines.

order to make the comparison with [3] fair, the features are extracted from RGB, depth map as well as estimated object size as in [3].

3D Detection Performance Fig. 8 shows the average precision for 4 classes, *coffee mug*, *cap*, *soda can*, and *flashlight*, in the WRGBD dataset. Again, we achieve the best accuracy compared to the baseline methods discussed in Sec. 3.4.1. We notice that the objects in this dataset have small variance in their size and pose, so that the baseline DPM+SizePrior already achieves a 3D localization accuracy of 32.8%. Note that our framework further improves the accuracy by roughly 3%. Table. 1 shows the effects of different components of the framework on WRGBD. We observe that the 3D features from HFMs are the most important component. Training with 3D loss function and using 3D matching further improve the performance.

2D Detection Performance Fig. 9 shows average precisions of our method, and the method proposed in [3]. Al-

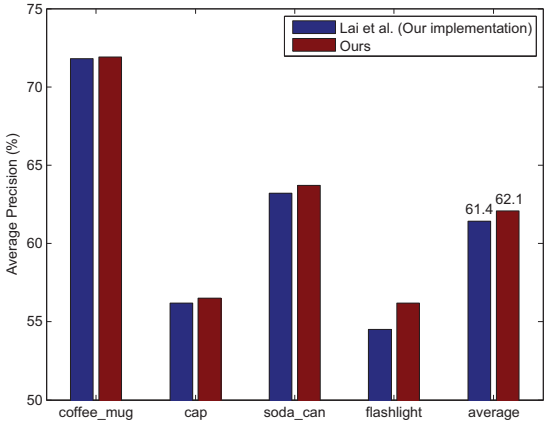


Figure 9: Average precisions of 2D object localization from DPM (with features proposed in [3]) and our method on the WRGBD dataset. Our method consistently achieves better average precision over [3].

though, as discussed earlier, the features used for baseline methods already contains information extracted from both RGB and depth map, our framework achieves the best performance compared to them.

4. Conclusions and Future work

In this work we proposed a new approach for localizing objects in 3D using RGB-D images. We explored the idea of using segmentation hypotheses for the foreground object to guide the process of accurately localizing the object in 3D. Extensive experimental analysis has demonstrated our theoretical claims. Directions for future work include the ability to integrate segmentation hypotheses in both 2D and 3D.

Acknowledgements

We acknowledge the support of ARO grant W911NF-09-1-0310, NSF CPS grant ♯0931474 and a KLA-Tencor Fellowship.

References

[1] Microsoft, "Microsoft kinect, http://www.xbox.com/en-us/kinect."

[2] A. Janoch, S. Karayev, Y. Jia, J. Barron, M. Fritz, K. Saenko, and T. Darrell, "A category-level 3-d object dataset: Putting the kinect to work," in *ICCV Workshops*, pp. 1168–1174, 2011.

[3] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in *ICRA*, pp. 1817–1824, 2011.

[4] D. Hoiem, A. Efros, and M. Hebert, "Putting objects in perspective," in *CVPR*, 2006.

[5] A. Gupta, A. Efros, and M. Hebert, "Blocks world revisited: Image understanding using qualitative geometry and mechanics," *ECCV*, 2010.

[6] S. Bao, M. Sun, and S. Savarese, "Toward coherent object detection and scene layout understanding," *Image and Vision Computing*, 2012.

[7] M. Fritz, K. Saenko, and T. Darrell, "Size matters: Metric visual search constraints from monocular metadata," *NIPS*, 2010.

Figure 10: This figure shows typical examples of object localization in 3D obtained using the proposed model and baseline methods. Each column represents ground truth bounding boxes in 2D, ground truth bounding boxes in 3D, 3D localization results using 3 baseline methods, and 3D localization results using our method, respectively. The localization results are drawn with black ellipsoids and green is used for ground truths. First four rows shows good examples. Notice the ellipsoids estimated by our framework is very close to ground truth ellipsoids, whereas baseline methods give less well localized ellipsoids. The last row shows failure cases. More typical examples can be found in the supplementary material [26].

[8] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *TPAMI*, vol. 32, no. 9, pp. 1627–1645, 2010.

[9] Y. Xiang and S. Savarese, "Estimating the aspect layout of object categories," in *CVPR*, 2012.

[10] B. Pepik, M. Stark, P. Gehler, and B. Schiele, "Teaching 3d geometry to deformable part models," in *CVPR*, 2012.

[11] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.

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

[13] M. Sun, G. Bradski, B.-X. Xu, and S. Savarese, "Depth-encoded hough voting for joint object detection and shape recovery," in *Computer Vision–ECCV 2010*, pp. 658–671, Springer, 2010.

[14] L. Bo, K. Lai, X. Ren, and D. Fox, "Object recognition with hierarchical kernel descriptors," in *ICCV*, 2011.

[15] M. Maire, S. Yu, and P. Perona, "Object detection and segmentation from joint embedding of parts and pixels," in *ICCV*, 2011.

[16] B. Kim, M. Sun, P. Kohli, and S. Savarese, "Relating things and stuff by high-order potential modeling," in *ECCV Workshop*, 2012.

[17] L. Ladickỳ, P. Sturgess, K. Alahari, C. Russell, and P. Torr, "What, where and how many? combining object detectors and crfs," *ECCV*, 2010.

[18] L. Cao and L. Fei-Fei, "Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes," in *ICCV*, 2007.

[19] J. Carreira and C. Sminchisescu, "Constrained parametric min-cuts for automatic object segmentation," in *CVPR*, pp. 3241–3248, 2010.

[20] B. Pepik, P. Gehler, M. Stark, and B. Schiele, "3d2pm–3d deformable part models," in *ECCV*, 2012.

[21] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman, "Multiple kernels for object detection," in *ICCV*, 2009.

[22] B. Steder, R. Rusu, K. Konolige, and W. Burgard, "Point feature extraction on 3d range scans taking into account object boundaries," in *ICRA*, 2011.

[23] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. Van Gool, "Hough transform and 3d surf for robust three dimensional classification," in *ECCV*, 2010.

[24] P. Felzenszwalb and D. Huttenlocher, "Distance transforms of sampled functions," 2004.

[25] T. Joachims, T. Finley, and C. Yu, "Cutting-plane training of structural svms," *Machine Learning*, vol. 77, no. 1, pp. 27–59, 2009.

[26] "Accurate localization in 3d project: http://www.eecs.umich.edu/vision/projects/al3d/al3dproj.html."

[27] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*, 2012.

[28] S. Alpert, M. Galun, R. Basri, and A. Brandt, "Image segmentation by probabilistic bottom-up aggregation and cue integration," in *CVPR*, 2007.

[29] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *IJCV*, 2010.