Surface Based Object Detection in RGBD Images

Siddhartha Chandra, Grigoris Chrysos, Iasonas Kokkinos

Surface Based Object Detection in RGBD Images

Siddhartha Chandra*[1]
siddhartha.chandra@inria.fr
Grigorios G. Chrysos*[2]
grigoris.chrysos@gmail.com
Iasonas Kokkinos[1]
iasonas.kokkinos@inria.fr

[1] INRIA GALEN
Centrale Supélec Paris
France
[2] Imperial College
London

*These authors contributed equally to this work. G. Chrysos was at Centrale Supélec while conducting this research.

Abstract

Viewpoint variation is one of the main challenges for object-detection frameworks. In this work we describe strategies to improve object detection pipelines by introducing viewpoint based mixture components. We learn accurate mixtures of object detectors for RGB-Depth (RGBD) data using the latent SVM framework. Our contributions are three-fold. First, we use surface-based object representations (3D mesh models) from available 3D object model repositories to learn strongly supervised viewpoint classifiers. These are used to guide the first stages of model learning, and help avoid inaccurate local minima of latent SVM training. Second, we develop a geometric dataset augmentation scheme that uses scene geometry to 'take another look' at the training data, simulating the effect of camera viewpoint changes. Third, to better exploit depth information, we develop a novel depth-based dense feature extraction method that provides a robust statistical description of scene geometry. We evaluate our learned detectors on the NYU dataset, and demonstrate that each of our advances results in systematic performance improvements over the traditional HOG-based detection pipeline.

1 Introduction

Intra-class variation in object detection is due to both intrinsic (texture and shape), and extrinsic (viewpoint and illumination) factors. Information about all such factors is rarely available. The recent enthusiasm about commercial RGBD sensors can be understood as being due to the reduction in intra-class variability: depth maps are invariant to texture and illumination changes, leaving us mostly with shape and viewpoint variation. This simplification was at the root of the success of Kinect's entirely depth-based pose estimation system [25] and has led to a proliferation of works around holistic scene analysis through shape, appearance and even physics-based cues [17, 34, 35]. While many recent approaches to learning from depth data have either exploited only RGB information from 3D object models [21] or only partially used surface information [26], we assume that depth is available both at training, and test time.

 This work has been partially funded by the EU Project MOBOT FP7-ICT-2011-600796.

In this work we use a mixture of object detectors for RGBD data. The mixture components in our model correspond to viewpoints and are treated as latent variables. We make three contributions. First, in section 3, we develop efficiently computable *displacement features* that statistically encode the shape of a surface within a sliding window. These complement the Histogram-Of-Gradient (HOG)[5, 8] and the Histogram-of-Depth Gradient (HOD) [29] features.

Second, in section 4, we exploit publicly available 3D object datasets that contain hand-crafted surface models of objects. We use these to construct a synthetic dataset comprising (a) depth maps from the rendered objects and (b) ground truth information regarding the viewpoint of an object. These datasets allow us to train viewpoint classifiers in a strongly supervised manner. We employ these classifiers to initialize the viewpoint components of natural images in our mixture model, so as to guide the first stages of model learning. Effectively these classifiers make an 'informed guess' about the viewpoints of objects. Our experiments indicate that this strategy consistently helps improve the detection accuracy.

Third, in section 5, we exploit the scene geometry for dataset augmentation. For this we simulate the effects of camera viewpoint changes by a combination of re-rendering and inpainting. This allows us to train detectors that are more tolerant to viewpoint changes at test time, by accommodating their variability at training-time.

We start by formally introducing our mixture of components model for object detection. We will be separately presenting prior works relevant to the aforementioned three advances.

2 Viewpoint Mixture Model

Our object detection framework consists of a mixture model where each mixture component corresponds to a viewpoint. Formally, we denote the parameters of our mixture model by $\mathbf{w} = \{\mathbf{w}_1, \cdots, \mathbf{w}_C\}$, where C is the total number of mixture components ($C = 3$ in our experiments). Given a bounding box $\mathbf{x}$ of an RGBD image, as well as a component ID $c \in \{1, \cdots, C\}$, the model provides a score, which is equal to the dot product of its parameters $\mathbf{w}_c$ and the feature vector of the bounding box $\Psi(\mathbf{x})$. The feature vector $\Psi(\mathbf{x})$ consists of two types of features: (i) the gradient and/or color features for the bounding box $\mathbf{x}$, which we denote by $\phi_g(\mathbf{x})$; (ii) the depth features for the bounding box $\mathbf{x}$, which we denote by $\phi_d(\mathbf{x})$. In other words, the score $s_{\mathbf{w}}(\mathbf{x}, c)$ for a bounding box $\mathbf{x}$, and component ID c is given by

$$\begin{aligned} s_{\mathbf{w}}(\mathbf{x}, c) &= \mathbf{w}_c^{\top} \Psi(\mathbf{x}), \\ \Psi(\mathbf{x}) &= [\phi_g(\mathbf{x}); \phi_d(\mathbf{x})] \end{aligned} \tag{1}$$

We use HOG [8] to define $\phi_g(\mathbf{x})$and a combination of HOD [29] and our novel *displacement* features to define $\phi_d(\mathbf{x})$. We describe these features in the next section. We will describe our learning framework in detail after the model description.

3 Surface Description using Displacement Features

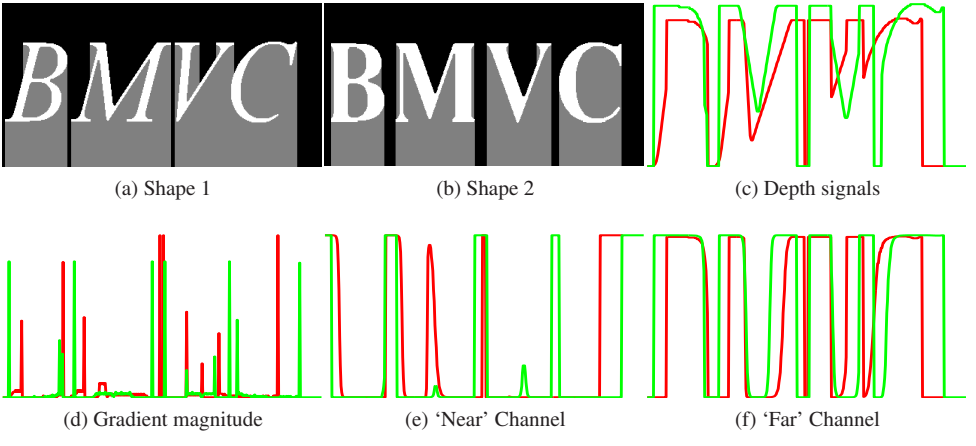
Most standard feature extraction schemes, for example [5, 18], employ a sequence of differentiation and L2-norm normalization; these two steps discard the effects of additive and multiplicative illumination changes respectively. This processing is combined with spatial pooling to render the descriptors robust with respect to small translations. Even though the aforementioned steps yield increased robustness, a substantial part of the signal information is lost during the processing steps. This information loss is considered to be partially responsible for the saturation of HOG-based detection performance [31].

More dedicated point cloud descriptors have been proposed [2, 3, 10, 15, 24, 28]. However, these point cloud descriptors are not efficient to compute densely and are primarily used either globally or at interest points.

Inspired by the success of simple depth comparision based features in [25], we propose efficient, densely computable depth features which complement gradient based features with surface-based information. Instead of relying on the few, and variable, positions where the depth signal changes, as in HOG/HOD, we describe our signal within a window in terms of the depth displacements with respect to its center.

Given a region/bounding box $\mathbf{x}$ in a depth image, and a cellsize s, we first subsample $\mathbf{x}$ using average pooling. In other words, we replace each non-overlapping $s \times s$ cell in $\mathbf{x}$ by the average depth value of the pixels in the cell. We denote the subsampled image region by $\widetilde{\mathbf{x}}$. Given $\widetilde{\mathbf{x}}$, we compute the displacement of each pixel p_i in $\widetilde{\mathbf{x}}$ from the center pixel p_0. This displacement is given by, $\delta_i = d^{p_0} - d^{p_i}$, where d^{p_i} is the depth value at pixel p_i.

We quantize δ_i into a set of N displacement bins, using a hard quantization function: $q(\delta) : R \rightarrow R^N$. Since δ_i can assume both positive and negative values, we have symmetric displacement bins corresponding to the positive and negative values. Thus δ_i is expressed as a sparse-indicator-feature vector of size N. We concatenate these sparse vectors to get the displacement feature of $\mathbf{x}$. In our experiments, hard quantization outperformed soft quantization schemes. Hard quantization also yields sparse features, enabling us to speed up convolutions. The number of displacement bins N was set with cross-validation. The cellsize s was set equal to the size of HOG cells.



(a) Shape 1 (b) Shape 2 (c) Depth signals

(d) Gradient magnitude (e) 'Near' Channel (f) 'Far' Channel

Figure 1: Gradient-versus-Displacement based features for two synthetic 'flatland', 2D shapes: as seen from above, their respective 'depth' signals would correspond to the functions shown on the top right. Gradient-based features like HOG (d) are sensitive to shape variation and alignment, while displacement-based features (e,f) encode relative depth, which is similar for both shapes on a larger extent of their domains.

The motivation for these features is illustrated intuitively in Figure 1. A feature extraction pipeline that relies on signal gradients (Figure 1 (d)) will either consider their non-overlapping gradient signals distinct, or resort to smoothing to make them comparable. Instead, our displacement-based features (Figure 1, (e),(f)) quantize the signal's domain into regions that are 'far' or 'near' from the center of the signal in depth, delivering features that exhibit a smaller amount of intra-class variation. In a certain sense both our displacement-

based and the gradient-based representations contain the same information, but in different 'formats': intuitively, our displacement features are more appropriate when the boundaries are variable, but the depth variability is consistent, while the HOD features could be more appropriate for well localized boundaries but potentially a larger breadth of depth differences.

4 Learning A Viewpoint Mixture Model

Mixture models are often used in object detection systems [8] to cope with viewpoint variations of objects in the data. Most training datasets for object detection do not contain any viewpoint information about the samples. In the absence of explicit ground truth annotations, viewpoints as commonly treated as latent variables.

In particular, given a training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i), i = 1, \cdots, N\}$, we wish to estimate the parameters of our mixture model such that it provides accurate detections for previously unseen test images. Here $\mathbf{x}_i$ is a training sample (specifically, a bounding box of an RGBD image) and $y_i \in \{+1, -1\}$ indicates whether the sample $\mathbf{x}_i$ belongs to the object class or the background class. For convenience, we denote the indices of all object samples as $\mathcal{O}$ and the indices of all background samples as $\mathcal{B}$.

In the absence of viewpoint information, we estimate the parameters of the model using weak supervision. We employ the following latent support vector machine (latent SVM) formulation [8, 32]:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2}||\mathbf{w}||^2 + C_O \sum_{i \in \mathcal{O}} \xi_i + C_B \sum_{j \in \mathcal{B}} \xi_j, \\ \text{s.t.} \quad & \max_c \mathbf{w}_c^\top \Psi(\mathbf{x}_i) \geq 1 - \xi_i, \xi_i \geq 0, \forall i \in \mathcal{O}, \\ & \max_c \mathbf{w}_c^\top \Psi(\mathbf{x}_j) \leq -1 + \xi_j, \xi_j \geq 0, \forall j \in \mathcal{B} \end{aligned} \tag{2}$$

where c denotes the viewpoint of the sample. Intuitively, the above problem considers the maximum score of a sample over all viewpoints. This score is encouraged to be larger than 1 for an object sample and smaller than -1 for a background sample. The hyperparameters C_O and C_B are the relative weights for the classification error of the object and background samples respectively. The above optimization problem is a difference-of-convex program whose local minimum or saddle point solution can be obtained using the CCCP algorithm [33].

The CCCP algorithm starts by initializing the viewpoints (or component IDs) and then alternatively estimates the model parameters w and viewpoints in an iterative manner. The main challenge in using this algorithm is that the initialization of the viewpoints greatly affects the performance in practice. In order to better initialize the viewpoints, we propose a novel approach in the next subsection, which exploits the availability of 3D models.

4.1 Component Initialization with 3D Models

The intuition behind using a mixture model for object detection is that each component can capture the statistics of an object category specific to one viewpoint. For example, in order to detect a 'car', we may use a mixture of three components, one each for the front, back and side view of a car, since each of these views has significantly different appearance. In order to capture this intuition, Felzenszwalb *et al.* [8] proposed to cluster the object samples into C clusters according to the aspect ratios of their corresponding bounding boxes. While the aspect ratio of the bounding box is an informative cue for the viewpoint of an object, it is not

always accurate. For example, it can help distinguish between the front and the side views of a car, but not the front and the back. In order to overcome this deficiency, we exploit the fact that our training samples contain depth information.

The idea of using 3D information to guide model learning has been recently explored in several works. For instance [21] used a 3D CAD model to guide the part placement for cars in RGB images, by rendering the CAD model from different viewpoints and predicting where its parts should be in the image. More recently [26] used the depth field to guide the training of DPMs [8] on RGBD images, but evaluated their models using RGB images. Recent literature also discusses reasoning in 3D from 2D images. Fidler *et al.* [9] extend DPMs to predict 3D bounding boxes around objects, while [1] uses a large number of colour images rendered from synthetic 3D models to learn exemplar SVMs and perform 3D-2D registration. Zia *et al.* [36] use 3D scene modeling to localize objects in RGB images using more detailed shapes than 2D bounding boxes. Unlike these approaches, we aim at a full-blown RGBD training algorithm that uses depth during both training and testing.

Specifically, we develop a two step strategy to initialize the component IDs for the object samples $\mathcal{O}$. First, we use 3D models of object categories from the Google Warehouse to generate synthetic samples with known viewpoints. The synthetic samples are used to learn a viewpoint classifier. Next the cues obtained by the viewpoint classifier are combined with the aspect ratio information in order to obtain the component IDs of the object samples.

4.2 Learning a Viewpoint Classifier

Figure 2: Depth images rendered from the synthetic 3D models showing the three candidate view-points used to initialize our mixture components. We first align each model to the fronto-parallel view, then rotate it to get the desired view. We rotate these samples by small angles ($< 10^{\circ}$) around random unit vectors to generate more samples for a view.

Using 3D models for the object category of interest, we generate a synthetic dataset $\mathcal{S} = \{(\mathbf{x}'_i, c'_i), i = 1, \cdots, M\}$, where $\mathbf{x}'_i$ is the projection of the 3D model on a 2D plane, and $c'_i \in \{1, \cdots, C\}$ is its component ID. The component IDs are chosen such that two samples $\mathbf{x}'_i$ and $\mathbf{x}'_j$ that correspond to similar viewpoints get assigned to the same component. Using the dataset $\mathcal{S}$, we learn a C-class SVM parametrized by $\mu = \{\mu_1, \cdots, \mu_C\}$ by solving the following convex optimization problem:

$$\begin{aligned} \min_{\mu} \quad & \frac{\lambda}{2}||\mu||^2 + \sum_{i=1}^{M} \xi_i, \\ \text{s.t.} \quad & \mu_{c'_i}^{\top} \phi_d(\mathbf{x}'_i) - \mu_c^{\top} \phi_d(\mathbf{x}'_i) \geq \Delta(c'_i, c) - \xi_i, \forall c \in \{1, \cdots, C\}, \forall i \in \{1, \cdots, M\}. \end{aligned} \quad (3)$$

The loss function $\Delta(\cdot,\cdot)$ is the standard 0-1 loss, that is,

$$\Delta(c_i', c) = \begin{cases} 0 & \text{if } c_i' = c, \\ 1 & \text{otherwise.} \end{cases} \tag{4}$$

Intuitively, the above problem learns the classifier μ that gives a high score to the correct component ID for each synthetic sample. The hyperparameter λ is the relative weight of the regularizer. Note that we only use the depth features $\phi_d(\cdot)$ to learn the viewpoint classifier. This is due to the fact that while the RGB information that is sometimes provided in the 3D CAD models is not realistic, the depth information is highly accurate.

4.3 Computing Component Initialization

Having learnt the viewpoint classifiers as described in the previous subsection, we now turn to combining them with the aspect ratio cues in order to obtain an accurate initialization of the component IDs. Intuitively, our method tries to assign component IDs to the samples based on the viewpoint classifier scores, while also trying to constrain samples having similar aspect ratios to belong to the same component. To this end, we specify a random field $\mathcal{G} = (\mathcal{X}_O \cup \mathcal{X}_C, \mathcal{E}_O \cup \mathcal{E}_C)$.

Variables of the Random Field. The variables of the random field are of two types. The variable $\mathbf{X}_i \in \mathcal{X}_O$ represents the i-th object sample $\mathbf{x}_i$ where $i \in \mathcal{O}$. The variable $\mathbf{X}_j \in \mathcal{X}_C$ represents the j-th cluster center of the clustering obtained using the aspect ratio of the object samples as proposed in [8], where $j \in \{1, \cdots, C\}$. Each variable can be assigned a label from the set $\mathcal{L} = \{l_1, \cdots, l_C\}$, where the label l_a represents component ID a. The unary potentials are defined as follows:

$$\begin{aligned} \theta_i(l_a) &= -\mu_a^\top \phi_d(\mathbf{x}_i), \forall \mathbf{X}_i \in \mathcal{X}_O, l_a \in \mathcal{L}, \\ \theta_j(l_a) &= 0, \forall \mathbf{X}_j \in \mathcal{X}_C, l_a \in \mathcal{L}. \end{aligned} \tag{5}$$

Thus, the unary potentials are the negative of the viewpoint classifier scores for the variables representing the samples, and 0 for the cluster centers.

Edges of the Random Field. We have two types of edges. Edges $\mathbf{E}_{ij} \in \mathcal{E}_O$ connect the sample $\mathbf{X}_i \in \mathcal{X}_O$ to the cluster center $\mathbf{X}_j \in \mathcal{X}_C$. The pairwise potentials for these edges are:

$$\theta_{ij}(l_a, l_b) = \begin{cases} 0 & \text{if } a = b, \\ -d(\mathcal{A}(\mathbf{X}_i), \mathcal{A}(\mathbf{X}_j)) & \text{otherwise,} \end{cases} \tag{6}$$

where $\mathcal{A}(\cdot)$ is the aspect ratio of the bounding box corresponding to the variable, and $d(\cdot,\cdot)$ is the difference between the aspect ratios. Thus, the pairwise potential is 0 if the object sample is assigned to the cluster center, and is proportional to the negative of the difference in their aspect ratios otherwise. The edges $\mathbf{E}_{jj'} \in \mathcal{E}_C$ connect the two cluster centers $\mathbf{X}_j, \mathbf{X}_{j'} \in \mathcal{X}_C$. The pairwise potentials corresponding to these edges are as follows:

$$\theta_{jj'}(l_a, l_b) = \begin{cases} \infty & \text{if } a = b, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

In other words, two cluster centers are prohibited from belonging to the same component.

Labeling. We represent a labeling of the above random field by a function $f : \mathcal{X}_O \cup \mathcal{X}_C \rightarrow \mathcal{L}$, that is, the variable $\mathbf{X} \in \mathcal{X}_O \cup \mathcal{X}_C$ takes the label $f(\mathbf{X})$. The energy of the labeling f is given by the sum of the corresponding unary and pairwise potentials, that is,

$$Q(f) = \sum_{\mathbf{X}_i \in \mathcal{X}_O} \theta_i(f(\mathbf{X}_i)) + \sum_{\mathbf{X}_j \in \mathcal{X}_C} \theta_i(f(\mathbf{X}_j)) + \lambda \left(\sum_{(\mathbf{X}_i, \mathbf{X}_j) \in \mathcal{E}_O} \theta_{ij}(f(\mathbf{X}_i), f(\mathbf{X}_j)) + \sum_{(\mathbf{X}_j, \mathbf{X}_{j'}) \in \mathcal{E}_C} \theta_{jj'}(f(\mathbf{X}_j), f(\mathbf{X}_{j'})) \right). \quad (8)$$

The hyperparameter λ is the relative weight of the aspect ratio cue with respect to the viewpoint classifier cue. When $\lambda = 0$, the inferred components correspond to the highest scoring viewpoint, and when $\lambda = \infty$, the inferred components are equal to the initialization of [8]. In our experiments, we determined the value of λ through cross-validation.

Energy Minimization. In order to obtain the initial viewpoint of each object sample $\mathbf{x}_i, i \in \mathcal{O}$, we wish to minimize the energy function (8). To this end, we observe that given the variables $\mathcal{X}_C$, the variables in $\mathcal{X}_O$ are conditionally independent of each other. Since C is typically very small, this implies that the cardinality of the variable set $\mathcal{X}_C$ and the label set $\mathcal{L}$ is small. This observation allows us to minimize the energy function over all possible labellings by exhaustively searching for all labellings of $\mathcal{X}_C$ and obtaining the best label for each $\mathbf{X}_i \in \mathcal{X}_O$ independently. The labeling with the minimum energy provides us with the initial component IDs of all the object samples. In our experiments $C = 3$ and the exhaustive search takes a few minutes for each experiment.

5 Dataset Augmentation with Geometric Jittering

Having exploited 3D models, in the last section, to learn more accurate object detectors, we now describe a strategy to make object detectors more robust to viewpoint variations. Camera viewpoint variation is a major challenge in object recognition; camera rotations affect object appearance radically and mixtures of viewpoint-tuned classifiers are imperative for multi-view detection. This challenge remains with RGBD sensors, as they only record the side of the object's surface facing the camera. Here we exploit depth information to learn robust detectors by accommodating variability due to moderate camera rotations.

We propose a dataset augmentation scheme that uses geometric information to take 'a different look' at objects in our training set. We simulate the effects of small camera rotations around the object, and obtain new samples that see the object from novel viewpoints. Our technique can be understood as a generalization of the 'jittering' technique that is known to drastically improve detection accuracy by using translated/scaled/rotated samples of an object during training. All these transformations assume that the azimuth and elevation of the camera stays fixed; here instead we let azimuth and elevation vary moderately ($\pm 10^o$) and use the resulting images to enhance the variability of our training set.

We use Depth-Image-Based Rendering (DIBR) [6, 7, 16, 19] which is common in applications [12] that require the synthesis of novel "virtual" views of an image, based on its intensity and depth values. Even though the geometric transformations involved in DIBR are straightforward, a host of image processing problems emerge [30], involving ghost contours, cracks, and most importantly, disocclusions, namely areas that cannot be viewed from the original viewpoint. More recently, [22] use structural information from 3D models to synthesize novel-views of cars from images, which are used to amplify training data for DPMs.

However, the method proposed in [22] requires alignment of real images with 3D models, which they achieve manually. In this work, we use the free viewpoint rendering method described in [6] to render both depth and RGB images of our objects. These rendered views suffer from the presence of holes. The existing literature describes several image inpainting techniques [4, 11, 13, 20] to remedy these holes. We post-process the RGB image renderings by performing an inverse warping of the missing pixels to retrieve the textures as described in [6], and the depth renderings using the bilateral filtering approach as proposed in [27].

(a) original (b) zoomed

Figure 3: We render novel views using the free viewpoint rendering method described in [6]: (a) original image, (b) zoomed in area of interest, and two views rendered by moving the camera to the left and top respectively. Despite some noticeable distortions in the rendered images, we empirically demonstrate that performing this geometric dataset augmentation during training is beneficial.

6 Experimental Results

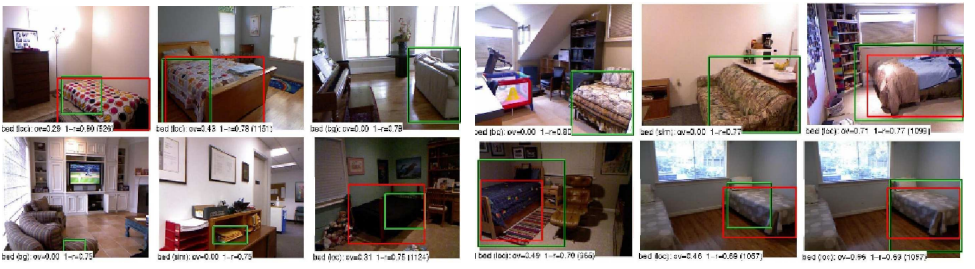
Method	Bed	Chair	M.+TV	Sofa	Table	Avg.
gdpm [26]	0.3339	0.1372	0.0928	0.1104	0.0405	0.1430
rgbd-hog + sparse [23]	0.4835	0.2435	0.2769	0.2869	0.1726	0.2927
rgb-hog	0.2337	0.1335	0.1339	0.1094	0.0376	0.1296
rgbd-hog	0.4660	0.2773	0.2480	0.2295	0.1430	0.2628
rgbd-hog + disp	0.5178	0.2771	0.2591	0.3440	0.1683	0.3133
rgbd-hog + disp + aug	0.5406	0.2919	0.2583	0.3470	0.1653	0.3206
rgbd-hog + disp + aug + vp	**0.5675**	**0.3023**	**0.3093**	**0.3719**	**0.1957**	**0.3493**
improvement over rgbd-hog	0.1015	0.0250	0.0613	0.1424	0.0527	0.0766

Table 1: Average Precision for the Detection Task on the NYU Dataset. rgb-hog uses only HOG features, rgbd-hog uses HOG+HOD, disp indicates displacement features, aug indicates the use of augmented data, vp indicates the viewpoint initialization using 3D models.

We now describe our experimental setup and empirical results. We used the NYU Depth v2 dataset [27] that contains 1449 RGB images with aligned depth maps. The dataset comes with train-test splits and pixel-wise object labels. From the pixel-wise labels we generate tight bounding box ground truth annotations for 5 object categories: bed, chair, monitor + television (M.+TV), sofa, and table as in [26]. We acquired 3-5 synthetic models for each class from the Google 3D warehouse, and rendered depth images centered around three landmark viewpoints: one frontal and two side views (Figure 2). We used these depth images

to initialize viewpoints for our real samples. We compute HOG features at a spatial bin size of 8. The displacement features were computed using a bin size of 15. Our view-point initialization method was trained using depth HOG and displacement feature descriptors.

For the mixture model, we learn 3 components as in [26]. Our augmented data was rendered by simulating the effect of the camera moving to the left, and top from the original orientation (Figure 3). This rendering is done offline, and takes approximately 2 seconds for each pair of images (RGB+Depth) on a single-core machine.As in [8, 26] we use lateral flipping of training images, so we do not render views where the camera moves to the right.



(a) rgbd-hog (b) rgbd-hog+disp+aug+vp

Figure 4: Top false positive detections for 'bed' for the baseline rgbd-hog detector, and our final rgbd-hog+disp+aug+vp detector. The green box shows the detection, the red shows the ground truth if present. Our overall approach makes fewer, more reasonable mistakes.

We evaluate the various methods outlined in this paper on the task of 2-D Object Detection with RGBD images on the NYU dataset. In Table 1, we report the Average Precision values for different methods. We also compare our results to the methods published in [23] and [26]. While [23] uses sparse coding features with DPM supervision, [26] uses depth information to weakly supervise the latent update step in the latent SVM formulation. Authors in [23] report using only RGB data; We used their framework with RGBD data. Figure 6 shows the precision-recall curves for the competing methods.

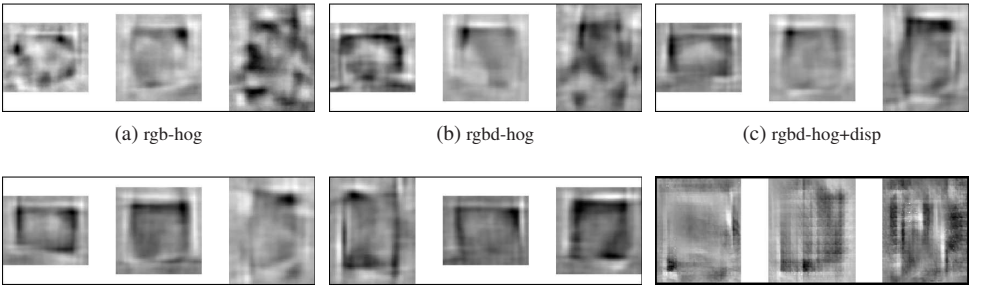
We get systematic improvements in detection performance as we introduce displacement features (+disp), augmented data (+aug), and viewpoint initialization (+vp) to the baseline detector. Our results also outperform those in [23, 26], even though [26] uses a deformable-parts model. Compared to the *rgbd-hog* baseline, we show an average improvement of 7.7%. While the displacement features allow us to better capture the depth variations of the surface, the augmentation and viewpoint initialization allow us to better model viewpoint effects, as reflected by the sharper templates in Figure 5. We visualized our detector mistakes using the techniques proposed in [14]. These mistakes are shown in Figure 4 as top false positive detections. It can be observed that our detector makes fewer mistakes, while just the rgbd-hog detector makes several obvious mistakes, such as firing on random objects, and parts of objects. Both the detectors often mistake a sofa for a bed, which is a reasonable mistake.

Acknowledgement: We would like to thank M. Pawan Kumar for providing his insights and expertise that greatly assisted this work.

7 Conclusions

In this paper, we have proposed strategies to improve object detection in RGBD images. Our contributions include our novel displacement features, a scheme for training data aug-

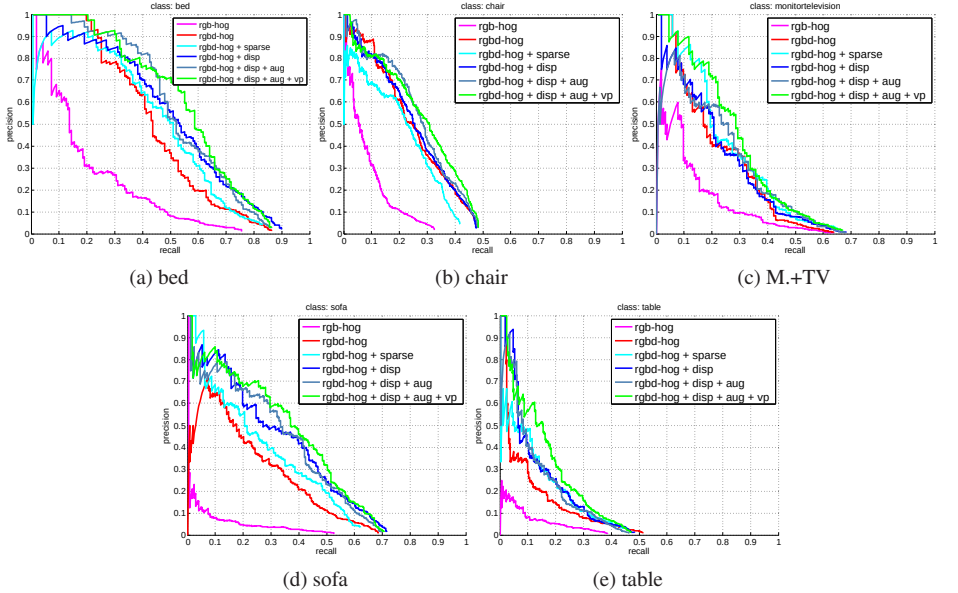
mentation which relies on rendering novel views from 2-D images, and a strategy for using existing 3D object models to initialize viewpoints in a latent SVM formulation. Our contributions systematically improve detection performance on a popular benchmark RGBD dataset. In this work, we exploited available 3D models to estimate the view-points of objects of interest in natural images. We rendered synthetic images with known viewpoints and used these to train viewpoint classifiers for natural images. The synthetic data we generated was clean of all clutter, while the natural images usually have objects surrounded by clutter. Learning viewpoint classifiers that are invariant to clutter is the next natural extension of this approach. The number of viewpoints dictates the number of components in our mixture model, and thus controls the model complexity. In the future we would like to develop models where we can express the viewpoint/orientation of an object in continuous space, while at the same time keeping a check on the model complexity. We would also like to extend our objective to solve the tasks of object detection and pose estimation simultaneously. Finally, we would like to use these strategies in the context of deep learning.



(a) rgb-hog (b) rgbd-hog (c) rgbd-hog+disp

(d) rgb-hog+aug (e) rgbd-hog+disp+aug+vp (f) vp classifier

Figure 5: Hoggles [31] visualization of model parameters learnt by our 3 component mixture of detectors for "M.+TV" for methods in Table 1. Filters in (c)-(e) resemble the object shape more than those in (a),(b). (f) shows the viewpoint classifiers learnt using 3D models.



(a) bed (b) chair (c) M.+TV

(d) sofa (e) table

Figure 6: Precision recall curves for the detection task for the 5 classes

References

[1] Aubry, Mathieu, Maturana, Daniel, Efros, Alexei, Russell, Bryan, and Sivic Josef. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *CVPR*, 2014.

[2] Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Depth kernel descriptors for object recognition. In *IROS*, 2011.

[3] Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Unsupervised feature learning for rgb-d based object recognition. In *ISER*, 2012.

[4] Antonio Criminisi, Patrick Perez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *Image Processing, IEEE Transactions on*, 13(9):1200–1212, 2004.

[5] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 2, pages 886–893, 2005.

[6] Luat Do, Sveta Zinger, et al. Quality improving techniques for free-viewpoint dibr. In *IS&T/SPIE Electronic Imaging*, pages 75240I–75240I. International Society for Optics and Photonics, 2010.

[7] Christoph Fehn. Depth-image-based rendering (dibr), compression, and transmission for a new approach on 3d-tv. In *Electronic Imaging 2004*, pages 93–104. International Society for Optics and Photonics, 2004.

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[9] Sanja Fidler, Sven J. Dickinson, and Raquel Urtasun. 3d object detection and viewpoint estimation with a deformable 3d cuboid model. In *NIPS*, pages 620–628, 2012.

[10] Andrea Frome, Daniel Huber, Ravi Kolluri, Thomas Bülow, and Jitendra Malik. Recognizing objects in range data using regional point descriptors. In *ECCV (3)*, 2004.

[11] Pascal Getreuer. Total variation inpainting using split Bregman. *Image Processing On Line*, 2012. doi: 10.5201/ipol.2012.g-tvi.

[12] Tal Hassner. Viewing real-world faces in 3d. ICCV, 2013.

[13] Daniel Herrera, Juho Kannala, Janne Heikkilä, et al. Depth map inpainting under a second-order smoothness prior. In *Image Analysis*, pages 555–566. Springer, 2013.

[14] Derek Hoiem, Yodsawalai Chodpathumwan, and Qieyun Dai. Diagnosing error in object detectors. In *ECCV (3)*, pages 340–353, 2012.

[15] Andrew Edie Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(5):433–449, 1999.

[16] Müller Karsten, Smolic Aljoscha, Dix Kristina, Merkle Philipp, Kauff Peter, Wiegand Thomas, et al. View synthesis for advanced 3d video systems. *EURASIP Journal on Image and Video Processing*, 2008, 2009.

[17] Dahua Lin, Sanja Fidler, and Raquel Urtasun. Holistic scene understanding for 3d object detection with rgbd cameras. In *ICCV*, 2013.

[18] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2): 91–110, 2004.

[19] Ha T Nguyen and Minh N Do. Image-based rendering with depth information using the propagation algorithm. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc*, volume 2, pages 589–592, 2005.

[20] George Papandreou, Petros Maragos, and Anil Kokaram. Image inpainting with a wavelet domain hidden markov tree model. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 773–776. IEEE, 2008.

[21] Bojan Pepik, Michael Stark, Peter Gehler, and Bernt Schiele. Teaching 3d geometry to deformable part models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012.

[22] Konstantinos Rematas, Tobias Ritschel, Mario Fritz, and Tinne Tuytelaars. Image-based synthesis and re-synthesis of viewpoints guided by 3d models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. URL http://homes.esat.kuleuven.be/~krematas/imgSynth/.

[23] Xiaofeng Ren and Deva Ramanan. Histograms of sparse codes for object detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3246–3253. IEEE, 2013.

[24] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *ICRA*, 2009.

[25] Jamie Shotton, Andrew W. Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*, pages 1297–1304, 2011.

[26] Abhinav Shrivastava and Abhinav Gupta. Building part-based object detectors via 3d geometry. In *ICCV*, 2013.

[27] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012*, pages 746–760. Springer, 2012.

[28] Richard Socher, Brody Huval, Bharath Putta Bath, Christopher D. Manning, and Andrew Y. Ng. Convolutional-recursive deep learning for 3d object classification. In *NIPS*, 2012.

[29] Luciano Spinello and Kai Oliver Arras. People detection in rgb-d data. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3838–3843. IEEE, 2011.

[30] Wenxiu Sun, Lingfeng Xu, Oscar C Au, Sung Him Chui, and Chun Wing Kwok. An overview of free view-point depth-image-based rendering (dibr). In *APSIPA Annual Summit and Conference*, 2010.

[31] Carl Vondrick, Aditya Khosla, Tomasz Malisiewicz, and Antonio Torralba. Hoggles: Visualizing object detection features. In *ICCV*, 2013.

[32] Chun-Nam John Yu and Thorsten Joachims. Learning structural svms with latent variables. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, 2009. ISBN 978-1-60558-516-1.

[33] A. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, 2003.

[34] Jian Zhang, Chen Kan, Alexander G. Schwing, and Raquel Urtasun. Estimating the 3d layout of indoor scenes and its clutter from depth sensors. In *ICCV*, 2013.

[35] B. Zheng, Y. Zhao, Joey C. Yu, K. Ikeuchi, and S.-C. Zhu. Beyond point clouds: Scene understanding by reasoning geometry and physics. In *CVPR*, 2013.

[36] M.Zeeshan Zia, Michael Stark, and Konrad Schindler. Are cars just 3d boxes? - jointly estimating the 3d shape of multiple objects. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, USA, 2014.