The Berkeley 3D Object Dataset

Allison Janoch
Trevor Darrell, Ed.
Pieter Abbeel, Ed.
Jitendra Malik, Ed.

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2012-85
http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-85.html

May 10, 2012

The Berkeley 3D Object Dataset

By

Allison Elizabeth Janoch

A thesis submit in partial satisfaction of the

Requirements for the degree of

Master of Science

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Trevor Darrell
Professor Jitendra Malik
Professor Pieter Abbeel

Spring 2012

Abstract

The Berkeley 3D Object Dataset

by

Allison Elizabeth Janoch

Masters of Science in Computer Science

University of California, Berkeley

Professor Trevor Darrell, Chair

The recent proliferation of the Microsoft Kinect [4], a cheap but quality depth sensor, has brought the need for a challenging category-level 3D object detection dataset to the forefront. Such a dataset can be used for object recognition in a spirit usually reserved for the large collections of intensity images typically collected from the Internet. The existence of such a dataset introduces new challenges in recognition, including the challenge of identifying valuable features to extract from range images.

This thesis will review current 3D datasets and find them lacking in variation of scenes, categories, instances, and viewpoints. The Berkeley 3D Object Dataset (B3DO), which contains color and depth image pairs gathered in real domestic and office environments will be presented. B3DO includes over 50 classes across 850 images. Baseline object recognition performance in a PASCAL VOC-style detection task is established, and two ways that inferred world size of the object can be used to improve detection are suggested.

In an effort to make more significant performance progress, the problem of extracting useful features from range images is addressed. There has been much success in using the histogram of oriented gradients (HOG) as a global descriptor for object detection in intensity images. There are also many proposed descriptors designed specifically for depth data (spin images, shape context, etc), but these are often focused on the local, not global descriptor paradigm. This work explores the failures of gradient based descriptors when applied to depth, and proposes that the proper global descriptor in the realm of 3D should be based on curvature, not gradients. This descriptor, the histogram of orientated curvature, exhibits superior performance for some classes of objects in the B3DO.

1 Introduction

The task of object recognition has made significant advances in the past decade and crucial to this success has been the creation of large datasets. Unfortunately, these successes have been limited to the use of intensity images and have chosen to ignore the very important cue of depth. Depth has long been thought to be an essential part of successful object recognition, but the reliance on large datasets has minimized the importance of depth. Collection of large datasets of intensity images is no longer difficult with the wide spread availability of images on the web and the relative ease of annotating datasets using Amazon Mechanical Turk. Recently, there has been a resurgence of interest in available 3-D sensing techniques due to advances in active depth sensing, including techniques based on LIDAR, time-of-flight (Canesta), and projected texture stereo (PR2). The Primesense sensor used on the Microsoft Kinect [4] gaming interface offers a particularly attractive set of capabilities, and is quite likely the most common depth sensor available worldwide due to its rapid market acceptance (8 million Kinects were sold in just the first two months).

There is a large body of literature on instance recognition using 3-D scans from the computer vision and robotics communities. However, there are surprisingly few existing datasets for category-level 3-D recognition, or for recognition in cluttered indoor scenes, despite the obvious

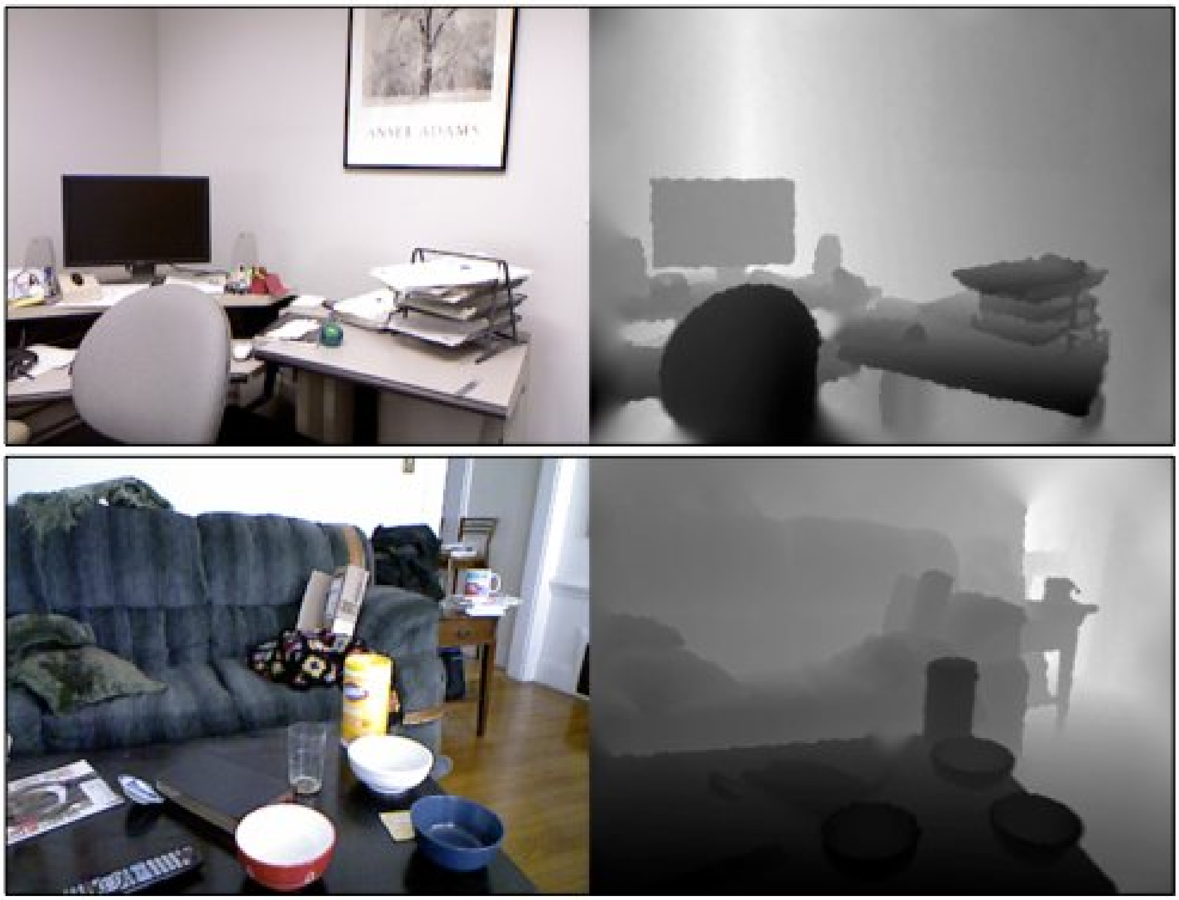

Figure 1: Two scenes typical of our dataset.

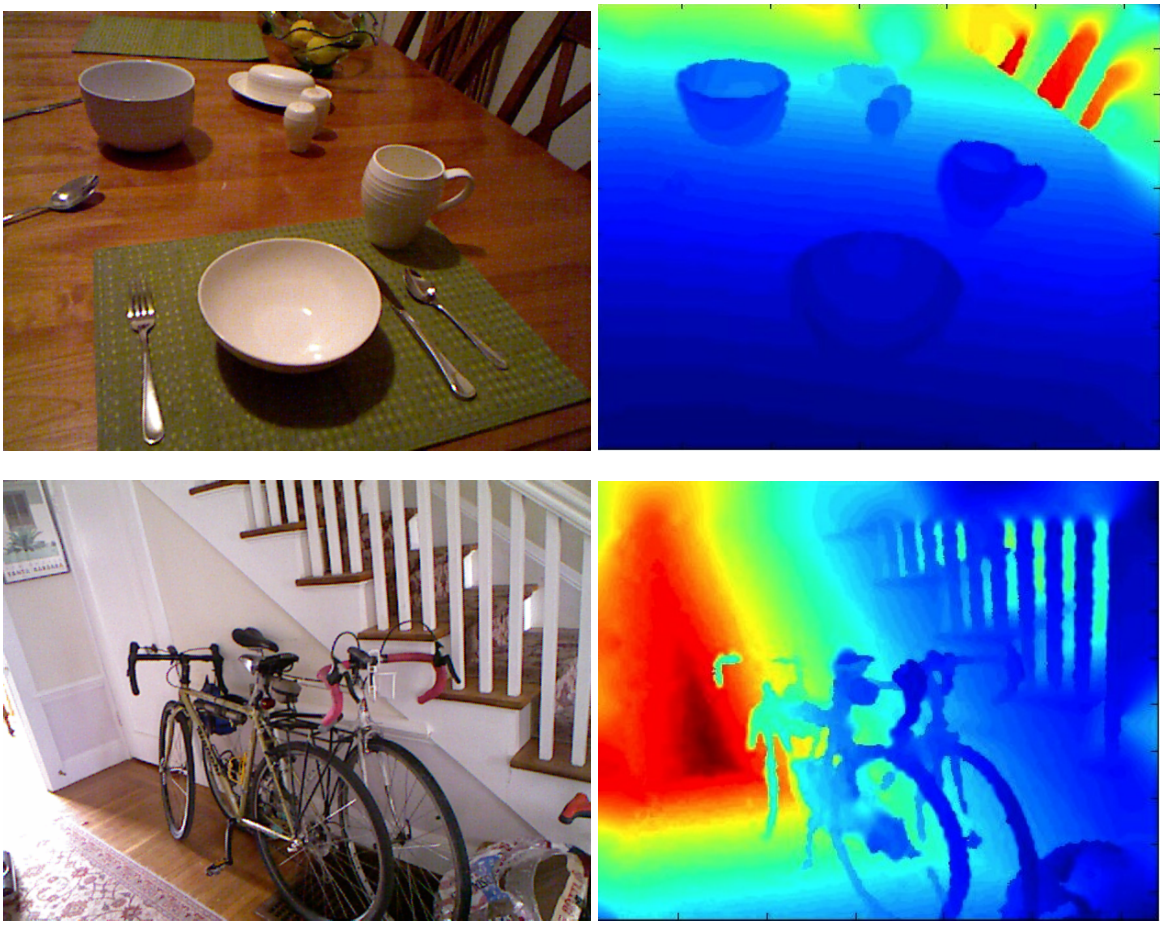

Figure 2: Typical scenes found in the B3DO. The intensity image is shown on the left, the depth image on the right.

importance of this application to both communities. As reviewed below, published 3-D datasets have been limited to instance tasks, or to a very small numbers of categories. Described here is the Berkeley 3-D Object dataset (B3DO) [21], a dataset for category level recognition, collected using the Kinect sensor in domestic and office environments. Figures 1 and 2 shows images representative of B3DO. The dataset has an order of magnitude more variation than previously published datasets. Since B3DO was collected using Kinect hardware, which uses active stereo sensing, the quality of the depth scans is much higher than in datasets based on passive stereo or sparsely sampled LIDAR. The dataset is available at http://www.kinectdata.com.

As with existing 2-D challenge datasets such as the Pascal VOC [12], B3DO has considerable variation in pose and object size, with objects covering a range of sizes from nearly 5% to almost 75% of image width. An important observation the dataset enables is that the actual world size distribution of objects has less variance than the image-projected, apparent size distribution. The statistics of these and other quantities for categories in the dataset are reported in Section 3.4.

A key question is what value does depth data offer for category level recognition? Conventional wisdom is that ideal 3-D observations provide strong shape cues for recognition, but in practice even the cleanest 3-D scans may reveal less about an object than available 2-D intensity data. Numerous schemes for defining 3-D features analogous to popular 2-D features for category-level recognition have been proposed and can perform in uncluttered domains. Section 4 evaluates the application of histogram of gradients (HOG) descriptors on 3D data and evaluates the benefit of such a scheme on our dataset. Observations about world size distribution can also be used to place a size prior on detections, which can improve detection performance as evaluated by average precision, as well as provide a potential benefit for detection efficiency.

For more significant performance improvements, features besides HOG must be explored. Much of the recent success of object recognition based solely on intensity images begins with the use of features derived from histograms of gradients. Detectors such as the deformable parts model proposed by Felzenszwalb et al. [14] begin with feature inspired by the HOG features described by Dalal and Triggs [10]. Such features have been demonstrated to have some success when used on range images [23] as shown in Section 4, but the feature was not originally designed to be used as a depth descriptor. In fact, a gradient based descriptor tends to identify discontinuities in depth, which in many cases is very similar to the representation that is learned by computing HOG features on intensity images. Despite this, there will be some differences in the features computed using gradients on intensity and range images and both will be useful at times. For example, in Figure 3 the back of the office chair would be easier to identify using HOG on the depth image.

Merely identifying discontinuities in depth does not capture much of the signal provided by depth. For example, an important characteristic of a bowl, like the one in Figure 3, is that it is concave on the inside, something that will not be captured by HOG on range images. There have been a number of features proposed for depth as described in Section 2.3, including both local features such as spin images [22], 3D shape context [16], the VFH model [27] and the features used for pose estimation in the Microsoft Kinect [28].

This work proposes that the proper feature to use in coordination with HOG should be similar, but instead of being based on first order statistics and gradients, should be based on second order statistics or curvature. Curvature is an appealing concept because the same surface in a range image will have the same Gaussian and mean curvature from any viewpoint under orthographic projection. This is because both Gaussian and mean curvature encode the first and second principal curvature in a way that is invariant to rotation, translations and changes in parameterization [6]. The curvature based feature, which we call a histogram of curvature or HOC, would be able to capture the fact that a bowl is concave on the inside, while maintaining the spatial binning that is appealing in HOG.

2 Related Work

There have been numerous previous efforts in collecting datasets with aligned 2D and 3D observations for object recognition and localization. Below is a review of the most pertinent ones, and a brief highlight of how B3DO is different. Also included in this section is a summary of related past work in 2D object recognition as well as an overview of previous work targeting the integration of 2D appearance and depth modalities.

2.1 3D Datasets for Detection

We present an overview of previously published datasets that combine 2D and 3D observation and contrast our dataset from those previous efforts:

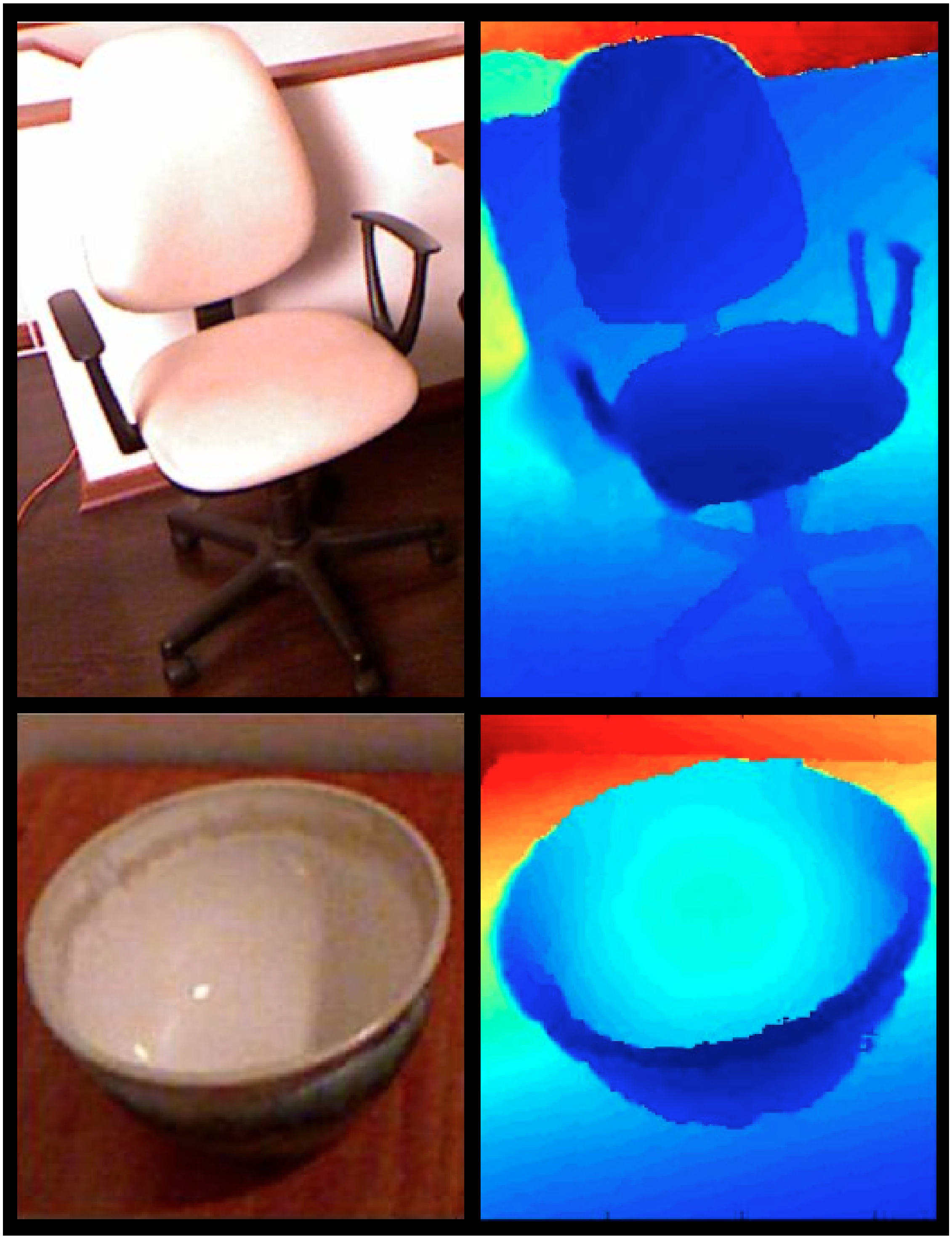

Figure 3: The office chair on top illustrates an example where the depth discontinuities identified by HOG on a depth image would offer additional information not as easily identified from the intensity image. The bowl on the bottom shows an example where gradients on the depth image would not be expected to yield much that could not be understood from the intensity image.

RGBD-dataset of [23]: This dataset from Intel Research and University of Washington features 300 objects in 51 categories. The category count refers to nodes in a hierarchy, with, for example, *coffee mug* having *mug* as parent. Each category is represented by 4 to 6 instances, which are densely photographed on a turntable. For object detection, only 8 short video clips are available, which lend themselves to evaluation of just 4 categories (bowl, cap, coffee mug, and soda can) and 20 instances. There does not appear to be significant viewpoint variation in the detection test set.

UBC Visual Robot Survey [3, 20]: This dataset from University of British Columbia provides training data for 4 categories (mug, bottle, bowl, and shoe) and 30 cluttered scenes for testing. Each scene is photographed in a controlled setting from multiple viewpoints.

3D table top object dataset [30]: This dataset from University of Michigan covers 3 categories (mouse, mug and stapler) and provides 200 test images with cluttered backgrounds. There is no significant viewpoint variation in the test set.

Solutions in Perception Challenge [2]: This dataset from Willow Garage forms the challenge which took place in conjunction with International Conference on Robotics and Automation 2011, and is instance-only. It consists of 35 distinct objects such as branded boxes and household cleaner bottles that are presented in isolation for training and in 27 scenes for test.

Max Plank Institute Kinect dataset [8]: This dataset was designed for category level recognition and contains 82 objects for training and 72 objects for testing across 14 different categories. Objects were photographed densely in isolation for both training and testing. The same object (but at a different viewing angle) was included in both the training and test sets.

Indoor Scene Segmentation dataset [29]: This dataset from NYU includes videos of 64 different scenes in 7 different types of rooms. Approximately 2300 of the 100,000 frames are segmented into regions.

Other datasets: Beyond these, other datasets have been made available which do include simultaneous capture of image and depth but serve more specialized purposes like autonomous driving [1], pedestrian detection [11] and driver assistance [32]. Their specialized nature means that they cannot be leveraged for the multi-object category localization task that is our goal.

In contrast to all of these datasets, B3DO contains both a large number of categories and many different instances per category. In addition, it is photographed "in the wild" instead of in a controlled turntable setting, and has significant variation in lighting and viewpoint throughout the set. For an illustration, consider Figure 4, which presents examples of the "chair" category in B3DO. These qualities make B3DO more representative of the kind of data that can actually be seen in people's homes; data that a domestic service robot would be required to deal with and use for online training.

2.2 2D Object Recognition

Robust multi-class object detection is a fundamental challenge in computer vision, and the literature on it is extensive. A common approach to detection employs a sliding window over the image, with each window considered for the presence of an object of a given object class. Efficiency of detection may be improved by employing cascades of detectors [31], or by window location and scale pruning [24].

Within a window, the image is featurized for input into the classifier. Empirical success has been found in features that encode spatial histograms of gradient orientations [25, 10]. Such fea-

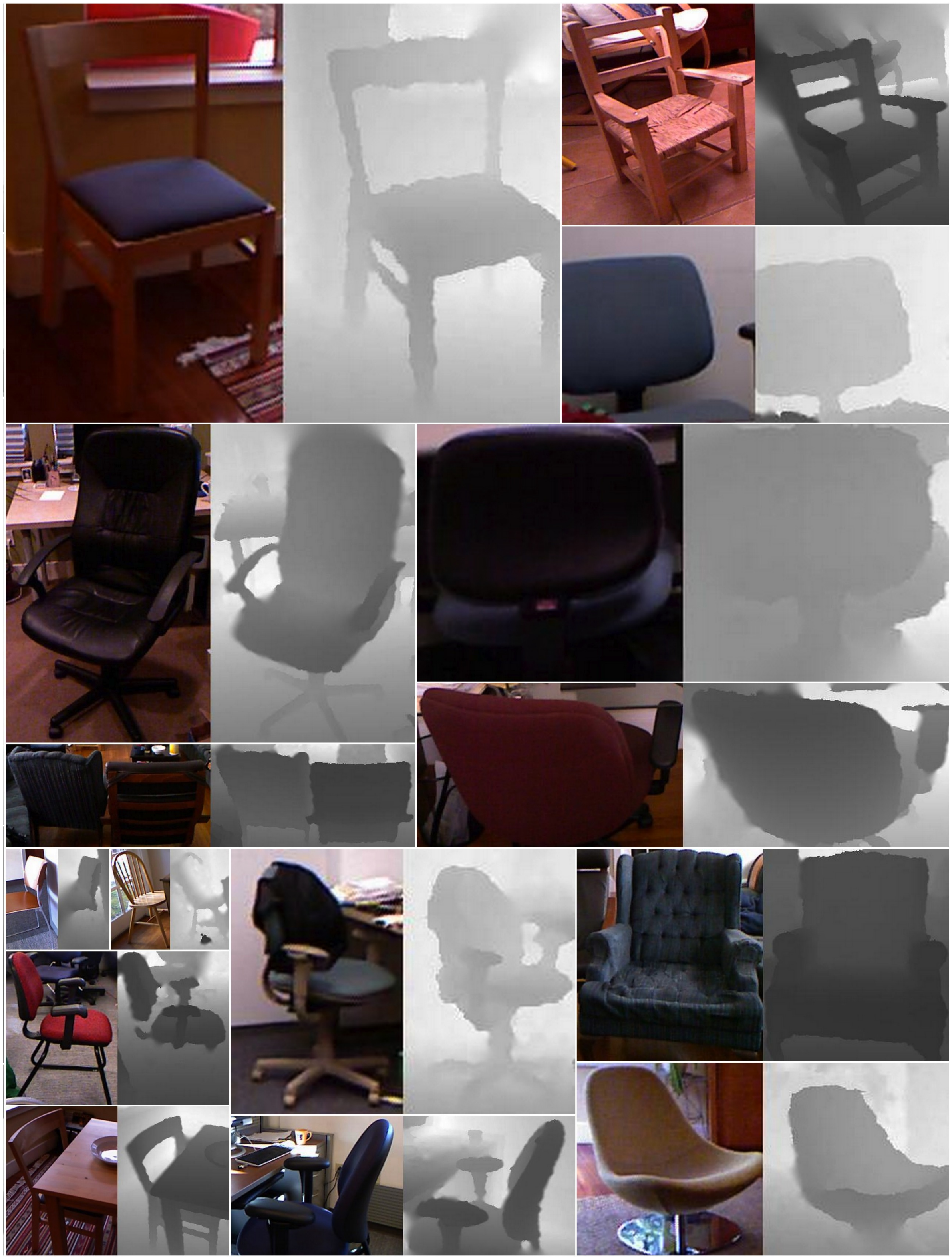

Figure 4: Instances of the “chair” class in our dataset, demonstrating the diversity of object types, viewpoint, and illumination.

tures achieve some invariance to slight viewpoint changes by spatially aggregating gradients, and to illumination differences by engineered normalization schemes. A classic combination of sliding window detection and gradient statistics-based local features is the Dalal-Triggs detector [10], which learns object categories as single filters over HOG features, and then applies the filter at all positions and scales in the image. The power of the PASCAL 2006-winning detector seems to come from the normalization scheme of its features.

The restriction to a single template per category was lifted most notably by Deformable Part Models [14], which keep HOG features, but enrich the Dalal-Triggs model to star-structured part-based models consisting of a root filter, part filters, and associated deformations. In addition, multiple such models may be learned per category, enabling increased discriminative power for different views of objects. These models may be learned from only object-class bounding boxes using a semi-convex optimization over part deformations. Due to the public availability of code and good detection performance, the detection experiments in Section 4 were based on the Deformable Part Models approach.

2.3 3D and 2D/3D Recognition

There have been a number of 3D features proposed for object recognition as well as a number of systems that combine intensity images with depth for object recognition. Although this is by no means an inclusive list, some local 3D features that have been proposed include spin images [22], 3D shape context [16], and the VFH model [27]. Both spin images and 3D shape context define a support region around interest points and then compute a histogram centered at that point. The support region is oriented with the surface normal in both cases, but for spin images the support region is a cylinder and for 3D shape context it is a sphere. For spin images the cylinder is broken up into bins radially and with the cylinders height. In contrast, 3D shape context breaks up the sphere into bins in the azimuth, elevation and radial dimensions, thus unlike spin images, 3D shape context is not rotationally invariant. Recently, Shotten et al [28] proposed a pose detector based on a random forest of decision trees. The features used in the trees examine a specific point and compare its depth to two other points to traverse the tree.

A number of 2D/3D hybrid approaches have been recently proposed, and B3DO should be a relevant testbed for these methods. A multi-modal object detector in which 2D and 3D are traded off in a logistic classifier is proposed by [17]. Their method leverages additional handcrafted feature derived from the 3D observation such as “height above ground” and “surface normal”, which provide contextual information. [30] shows how to benefit from 3D training data in a voting based method. Fritz et al. [15] extends branch and bound’s efficient detection to 3D and adds size and support surface constraints derived from the 3D observation.

Most prominently, a set of methods have been proposed for fusing 2D and 3D information for the task of pedestrian detection. The popular HOG detector [10] to disparity-based features is extended by [19]. A late integration approach is proposed by [26] for combining detectors on the appearance as well as depth image for pedestrian detection. Instead of directly learning on the depth map, [32] uses a depth statistic that learns to enforce height constraints of pedestrians. [11] explores pedestrian detection by using stereo and temporal information in a hough voting framework also using scene constraints. Recently, Lai et al. [23] evaluated object detection of a challenging dataset collected with the Kinect. They combined three features: HOG on intensity images, HOG on depth images and a histogram calculated based on the estimated scale of an

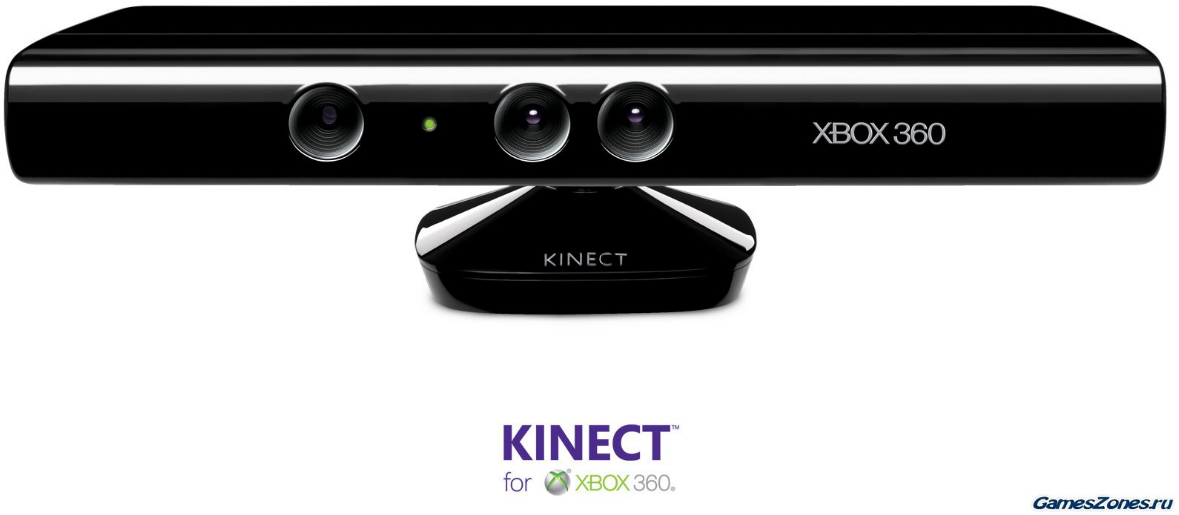

Figure 5: The Microsoft Kinect sensor [4].

object. They found the combination of the three features yields significantly improved results over a detector based solely on intensity images.

3 The Dataset

The Berkeley 3D Object Dataset is a large-scale dataset of images taken in domestic and office settings with the commonly available Kinect sensor. The sensor provides a color and depth image pair, and is processed for alignment and inpainting (see Section 3.3). The data was collected by many members of the research community, as well as an Amazon Mechanical Turk (AMT) worker, providing an impressive variety in scene and object appearance. As such, the dataset is intended for evaluating approaches to category-level object recognition and localization.

The dataset was collected with ten different Kinects that were taken to the homes and offices of 19 different volunteers who collected 849 images from 75 different scenes or rooms. Volunteers were given relatively simple instruction on how specifically to collect images. They were told a list of objects that would be labeled and were told to take images that did not looked staged containing one or more of these objects. Obviously, the more restrictive the instructions for collection are, the more difficult it is to gather data. The hope was that simple instructions would enable the dataset to grow more by using AMT workers for collection. This turned out to be more difficult than anticipated, which is discussed in Section 4.

Over 50 different object classes are represented in the dataset by crowd-sourced labels. The annotation was done by AMT workers in the form of bounding boxes on the color image, which are automatically transferred to the depth image.

3.1 Data Annotation

Crowd sourcing on AMT was used to label the data collected. AMT is a well-known service for “Human Intelligence Tasks” (HIT), which are typically small tasks that are too difficult for current machine intelligence. Our labeling HIT gives workers a list of eight objects to draw bounding boxes around in a color image. Each image is labeled by five workers for each set of labels in order to provide sufficient evidence to determine the validity of a bounding box. A proposed annotation or bounding box is only deemed valid if at least one similarly overlapping bounding box is drawn by another worker. The criteria for similarity of bounding boxes is based on the PASCAL VOC [12] overlap criterion (described in more detail in Section 4.1), with the acceptance threshold set to 0.3. If only two bounding boxes are found to be similar, the larger one is chosen. If more than two are deemed similar, the bounding box which overlaps the most with the other bounding boxes is kept, and rest are discarded.

3.2 The Kinect Sensor

The Microsoft Xbox Kinect [4] was originally designed as a video game peripheral designed for controller-free gaming through human pose estimation and gesture recognition. The sensor (Figure 5) consists of a horizontal bar with cameras, a structured light projector, an accelerometer and an array of microphones mounted on a motorized pivoting foot. Since its release in November 2010, much open source software has been released allowing the use of the Kinect as a depth sensor [9]. Across the horizontal bar are two sensors, an infrared camera and a RGB camera (640 x 480 pixels). Depth is measured using a laser projector that projects a structured light pattern on the surface to be sensed by the infrared camera. The depth range is approximately 0.6 to 6.0 meters. [4]. Depth reconstruction uses proprietary technology from Primesense, consisting of continuous infrared structured light projection onto the scene.

The Kinect color and infrared cameras are a few centimeters apart horizontally, and have different intrinsic and extrinsic camera parameters, necessitating their calibration for proper registration of the depth and color images. Calibration parameters differ significantly from unit to unit, which poses a problem to totally indiscriminate data collection. Fortunately, the calibration procedure is made easy and automatic due to efforts of the open source community [9, 7].

3.3 Smoothing Depth Images

The structured-light method used for recovering ground-truth depth-maps necessarily creates areas of the image that lack an estimate of depth. In particular, glass surfaces and infrared-absorbing surfaces can be missing in depth data. Tasks such as getting the average depth of a bounding box, or applying a global descriptor to a part of the depth image therefore benefit from some method for “inpainting” this missing data.

This work assumes that proper inpainting of the depth image requires some assumption of the behavior of natural shapes and that objects have second order smoothness (that curvature is

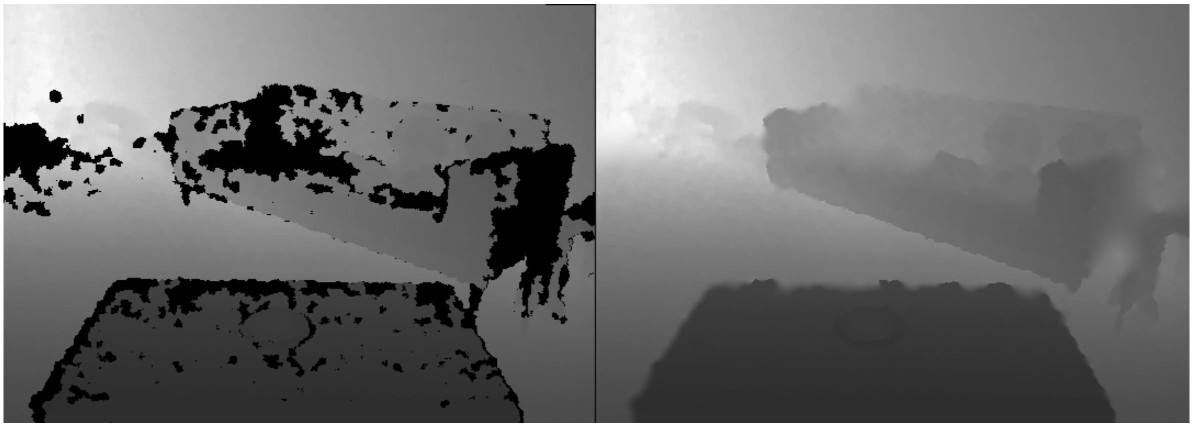

Figure 6: Illustration of our depth smoothing method.

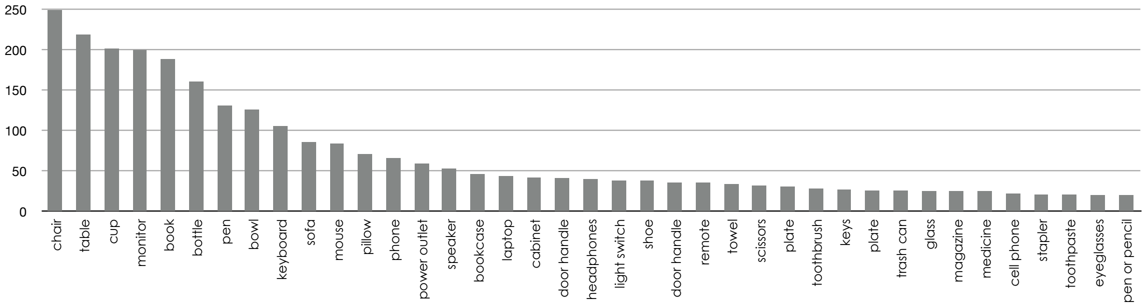

Figure 7: Object frequency for 39 classes with 20 or more examples.

minimized)—a classic prior on natural shapes [18, 34]. In short, the inpainting algorithm minimizes

$$\|h * Z\|_F^2 + \|h^{\mathrm{T}} * Z\|_F^2 \tag{1}$$

with the constraints $Z_{x,y} = \hat{Z}_{x,y}$ for all $(x, y) \in \hat{Z}$, where $h = [-1, +2, -1]$, is an oriented 1D discrete Laplacian filter, $*$ is a convolution operation, and $\|\cdot\|_F^2$ is the squared Frobenius norm. The solution to this optimization problem is a depth-map Z in which all observed pixels in $\hat{Z}$ are preserved, and all missing pixels have been filled in with values that minimize curvature in a least-squares sense. This problem is occasionally ill-conditioned near the boundaries of the image, so a small additional regularization term is introduced for first-order smoothness. For speed considerations, the hard constraints in the problem above are relaxed to heavily penalized soft constraints, and solve the induced least-square problem.

Figure 6 illustrates this algorithm operating on a typical input image with missing depth in B3DO to produce the smoothed output.

3.4 Data Statistics

The distribution of objects in household and office scenes as represented in B3DO is shown in Figure 7. The typical long tail of unconstrained datasets is present, and suggests directions for targeted data collection. There are 12 classes with more than 70 examples, 27 classes with more than 30 examples, and over 39 classes with 20 or more examples.

Unlike other 3D datasets for object recognition, B3DO features large variability in the appearance of object class instances. This can be seen in Figure 4, presenting random examples of the chair class in the dataset; the variation in viewpoint, distance to object, frequent presence of partial occlusion, and diversity of appearance in this sample poses a challenging detection problem.

The apparent size of the objects in the image, as measured by the bounding box containing them, can vary significantly across the dataset. The real-world size of the objects in the same class varies far less, as can be seen in Figure 8. As a proxy for the real-world object size, the product of the diagonal of the bounding box l and the distance to the object from the camera D is used, which is roughly proportional to the world object size by similar triangles (of course, viewpoint variation slightly scatters this distribution–but less so than for the bounding box size).

This work found that mean smoothed depth is roughly equivalent to the median depth of the depth image ignoring missing data, and so this is used to measure distance. The Gaussian was found to be a close fit to these size distributions, allowing estimation of the size likelihood of a bounding box as $\mathcal{N}(x|\mu, \sigma))$, where μ and σ are learned on the training data. This result will be used further in Section 4.3.

4 Detection Baselines

The cluttered scenes of B3DO provide for a challenging object detection task, where the task is to localize all objects of interest in an image. Here, the task is constrained to finding eight different object classes: chairs, monitors, cups, bottles, bowls, keyboards, computer mice, and phones. These object classes were among the most well-represented in our dataset.[1]

4.1 Sliding window detector

The baseline system is based on a standard detection approach of sliding window classifiers operating on a gradient representation of the image [10, 14, 33]. Such detectors are currently the state of the art on cluttered scene datasets of varied viewpoints and instance types, such as the PASCAL-VOC challenge [12]. The detector considers windows of a fixed aspect ratio across locations and scales of an image pyramid and evaluates them with a score function, outputting detections that score above some threshold.

Specifically, the implementation of the Deformable Part Model detector [14] is followed. This uses the LatentSVM formulation

$$f_\beta(x) = \max_z \beta \cdot \Phi(x, z) \tag{2}$$

for scoring candidate windows, where β is a vector of model parameters and z are latent values (allowing for part deformations). Optimizing the LatentSVM objective function is a semi-convex problem, and so the detector can be trained even though the latent information is absent for negative examples.

[1] We chose not to include a couple of other well-represented classes into this test set because of extreme variation in interpretation of instances of object by the annotators, such as the classes of "table" and "book."

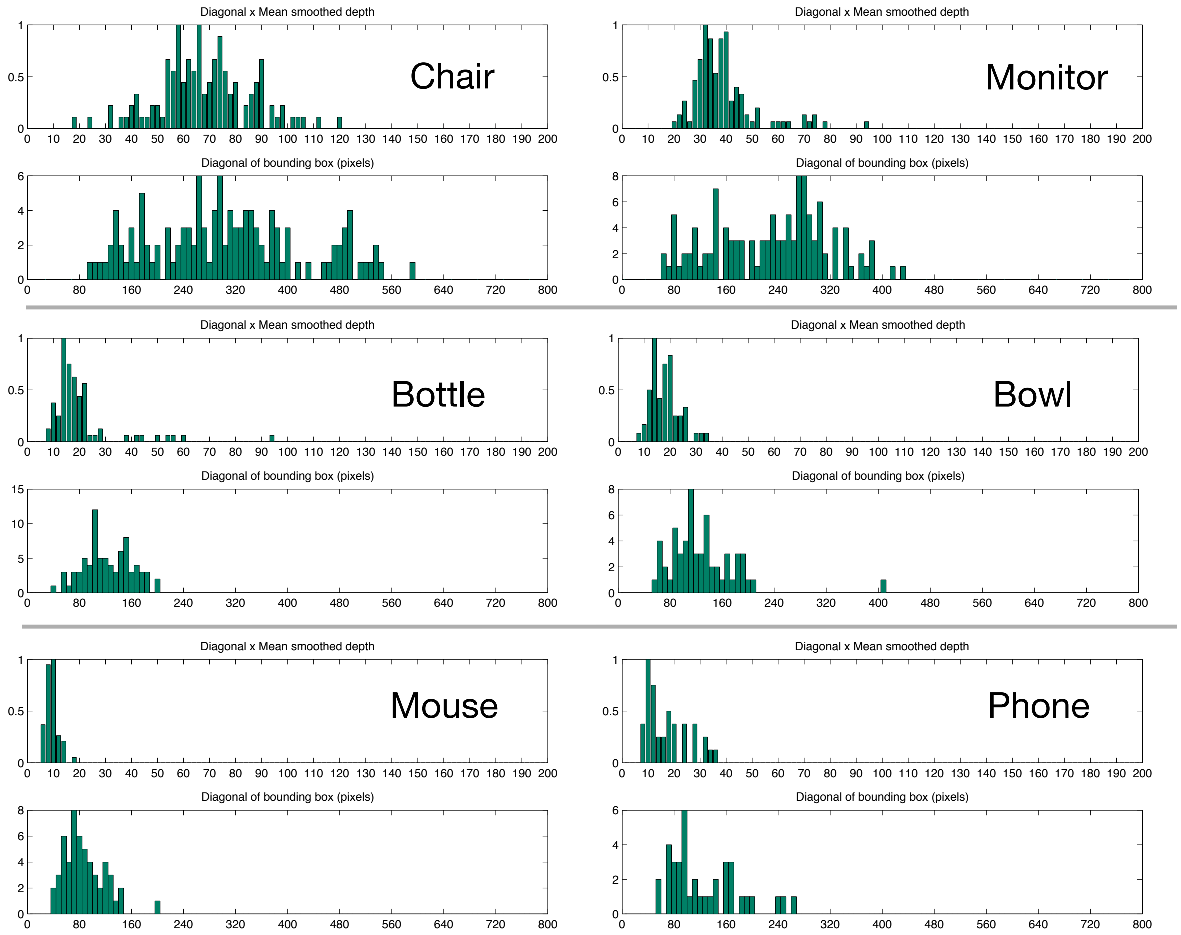

Figure 8: Statistics of object size. For each object class, the top histogram is inferred world object size, obtained as the product of the bounding box diagonal and the average depth of points in the bounding box. The bottom histogram is the distribution of just the diagonal of the bounding box size. (Note the difference in scale on the x-axis for these histograms)

Since finding good negative examples to train on is of paramount importance in a large dataset, the system performs rounds of data mining for small samples of hard negatives, providing a provably exact solution to training on the entire dataset.

To featurize the image, HOG with both contrast-sensitive and contrast-insensitive orientation bins, four different normalization factors, and 8-pixel wide cells is used. The descriptor is analytically projected to just 31 dimensions, motivated by the analysis in Felzenszwalb et al. [14].

Two feature channels for the detector are explored. One consists of featurizing the color image, as is standard. For the other, this work applies HOG to the depth image (Depth HOG), where the intensity value of a pixel corresponds to the depth to that point in space, measured in meters. This application of a gradient feature to depth images has little theoretical justification, since first-order statistics do not matter as much for depth data (this is why we use second-order smoothing in Section 3.3). Yet this is an expected first baseline that also forms the detection approach on some other 3D object detection tasks, such as in [23]. Section 5 will explore features based on second-order statistics.

Detections are pruned by non-maximum suppression, which greedily takes the highest-scoring bounding boxes and rejects boxes that sufficiently overlap with an already selected detection. This procedure results in a reduction of detections on the order of ten, and is important for the evaluation metric, which penalizes repeat detections.

4.2 Evaluation

Evaluation of detection is done in the widely adopted style of the PASCAL detection challenge, where a detection is considered correct if

$$\frac{area(B \cap G)}{area(B \cup G)} > 0.5 \tag{3}$$

where B is the bounding box of the detection and G is the ground truth bounding box of the same class. Only one detection can be considered correct for a given ground truth box, with the rest considered false positives. Detection performance is represented by precision-recall (PR) curves, and summarized by the area under the curve–the average precision (AP). Evaluation is done on six different splits of the dataset, averaging the AP numbers across splits.

The goal of this work is category, not instance-level recognition. As such, it is important to keep instances of a category confined to either training or test set. This makes the recognition task much harder than if training on the same instances of a category as exists in the test set was allowed (but not necessarily the same views of them). To enforce this constraint, images from the same scene or room are never in both the training and test sets. This is a harder constraint than needed, and is not necessarily perfect (for example many different offices might contain the same model laptop). As there is no scalable way to provide per-instance labeling of a large, crowd-sourced dataset of cluttered scenes, this method is settled upon, and keep the problem open for further research.

Figure 9 shows the detector performance on 8 different classes. Note, depth HOG is never better than HOG on the 2D image. This can be attributed to the inappropriateness of a gradient feature on depth data, as mentioned earlier, and to the fact that due to the limitations of the infrared structured light depth reconstruction, particular objects (such as monitors) tend to have significant missing depth data. Figure 10 provides an illustration of cases in which objects are missing depth data, along with objects from the same class which are missing much less depth data.

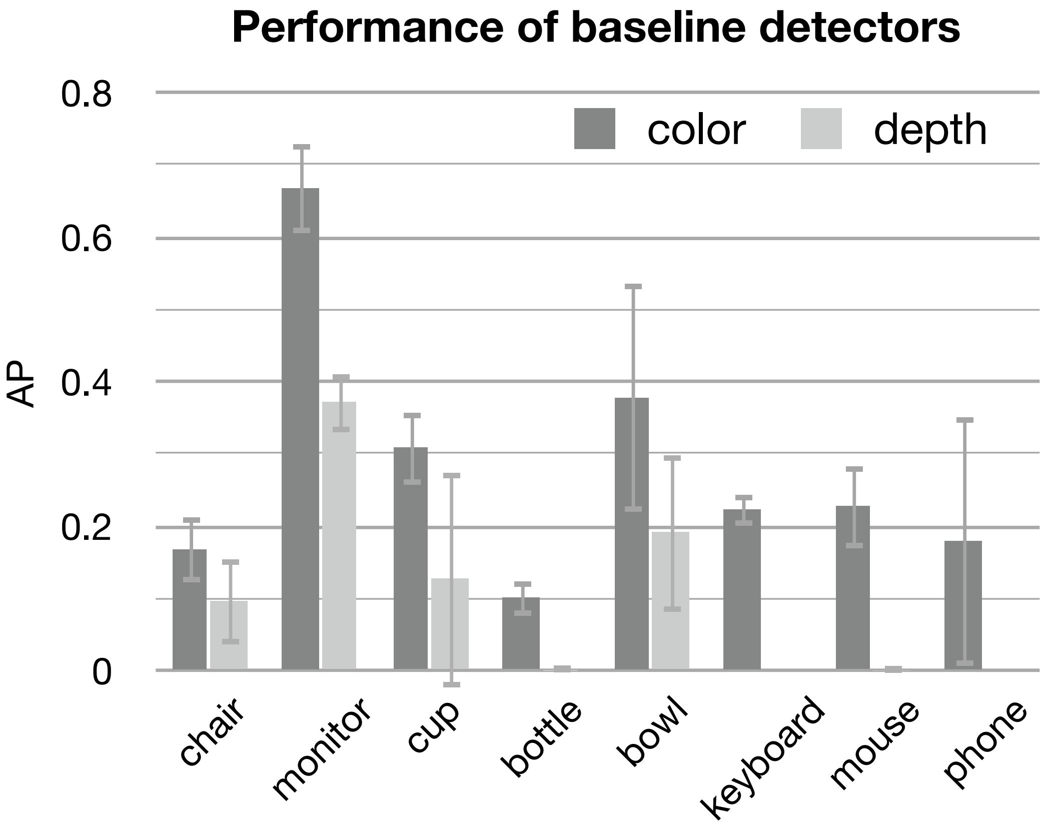

Figure 9: Performance of the baseline detector on our dataset, as measured by the average precision. Depth HOG fails completely on some categories, for reasons explained in the text.

4.3 Pruning and rescoring by size

In Section 3.4, the distributions of object size demonstrated that true object size, even as approximated by the product of object projection in the image and median depth of its bounding box, varies less than bounding box size. In the following, two ways of using approximated object size as an additional source of discriminative signal to the detector are investigated.

The first way of using size information consists of pruning candidate detections that are sufficiently unlikely given the size distribution of that object class. The object size distribution is modeled with a Gaussian, which is a close fit to the underlying distribution; the Gaussian parameters are estimated on the training data only. Boxes that are more than $\sigma = 3$ standard deviations away from the mean of the distribution are pruned.

Figure 11 shows that the pruning results provide a boost in detection performance, while rejecting from 12% to 68% of the suggested detection boxes (on average across the classes, 32% of candidate detections are rejected). This observation can be leveraged as part of an “objectness” filter or as a thresholding step in a cascaded implementation of this detector for detection speed gain [5, 13]. The classes chair and mouse are the two classes most helped by size pruning, while monitors and bottle are the least helped.

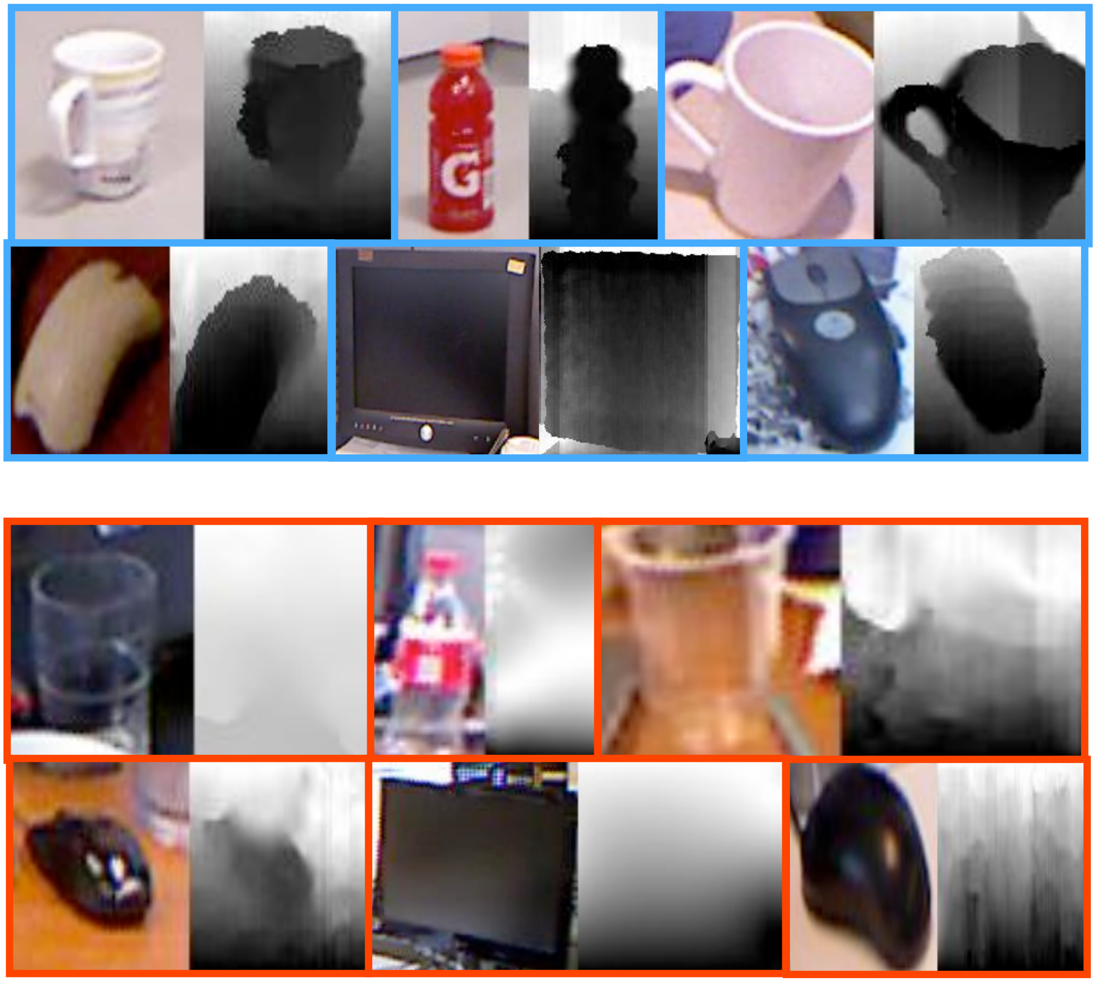

Figure 10: Top set of examples show good depth data for the objects. Bottom set of examples shows examples of missing depth data for objects of the same classes.

Using bounding box size of the detection (as measured by its diagonal) instead of inferred world size results in no improvement to AP performance on average. Two classes that are most hurt are bowl and plate; two that are least hurt by the bounding box size pruning are bottle and mouse.

The second way we use size information consists of learning a rescoring function for detections, given their SVM score and size likelihood. A simple combination of the two values is learned:

$$s(x) = \exp(\alpha \log(w(x)) + (1-\alpha)\log(\mathcal{N}(x|\mu,\sigma))) \tag{4}$$

where $w(x) = 1/(1+\exp(-2f_\beta(x)))$ is the normalized SVM score, $\mathcal{N}(x|\mu,\sigma))$ is the likelihood of the inferred world size of the detection under the size distribution of the object class, and α is a parameter learned on the training set. This corresponds to unnormalized Naive Bayes combination of the SVM model likelihood and object size likelihood. Since what matters for the precision-recall evaluation is the ordering of confidences and whether they are normalized is irrelevant, $s(x)$ can be evaluated.

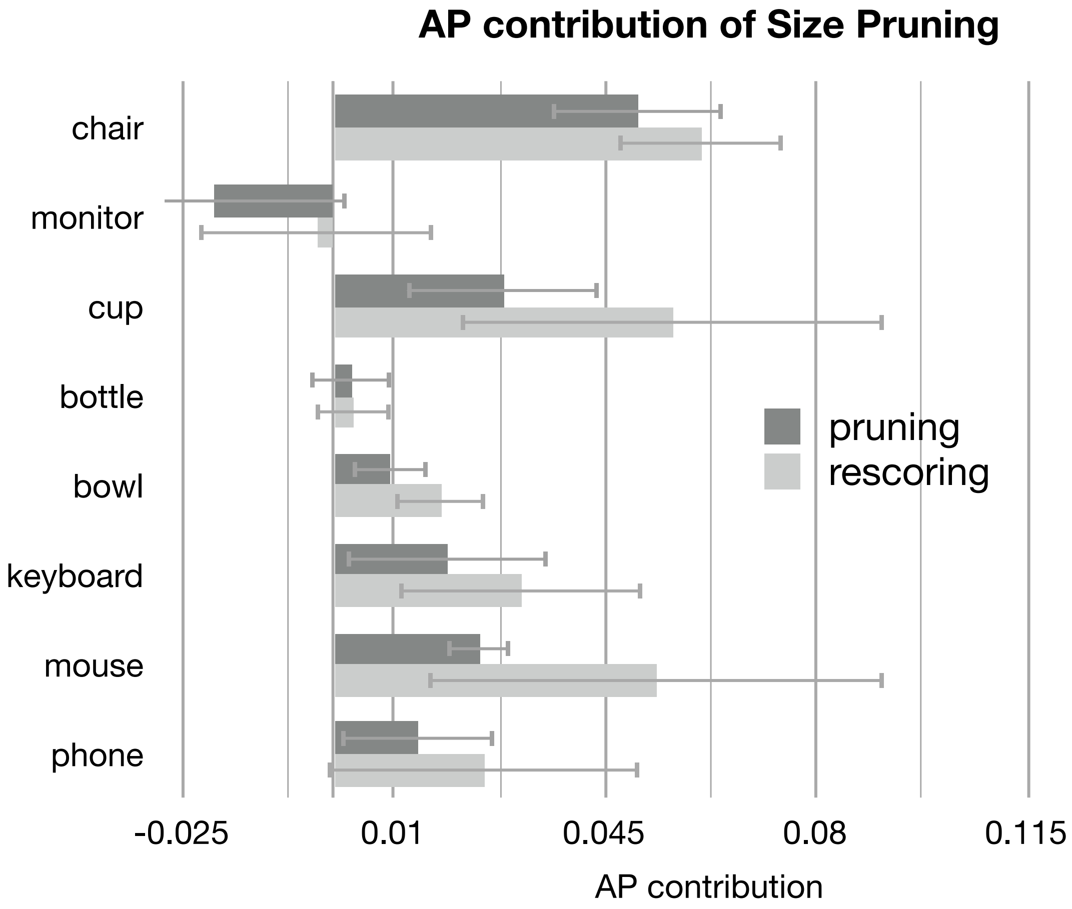

Figure 11: Effect on the performance of our detector shown by the two uses of object size we consider.

As Figure 12 demonstrates, the rescoring method works better than pruning. This method is able to boost recall as well as precision by assigning a higher score to likely detections in addition to lowering the score (which is, in effect, pruning) of unlikely detections.

5 A Histogram of Curvature (HOC)

The previous section demonstrated how HOG could be used to featurize range images. As mentioned earlier, this is not the ideal use of HOG since it is designed to be used on intensity images. This work seeks to define a feature representation analogous to HOG that is more appropriate for range images. Curvature is an appealing feature to work with when range data is available because it is potentially less sensitive to changes in viewpoint than gradient based descriptors (such as HOG). As mentioned in the introduction, a surface in a range image will have the same Gaussian and mean curvature from any viewpoint under orthographic projection.

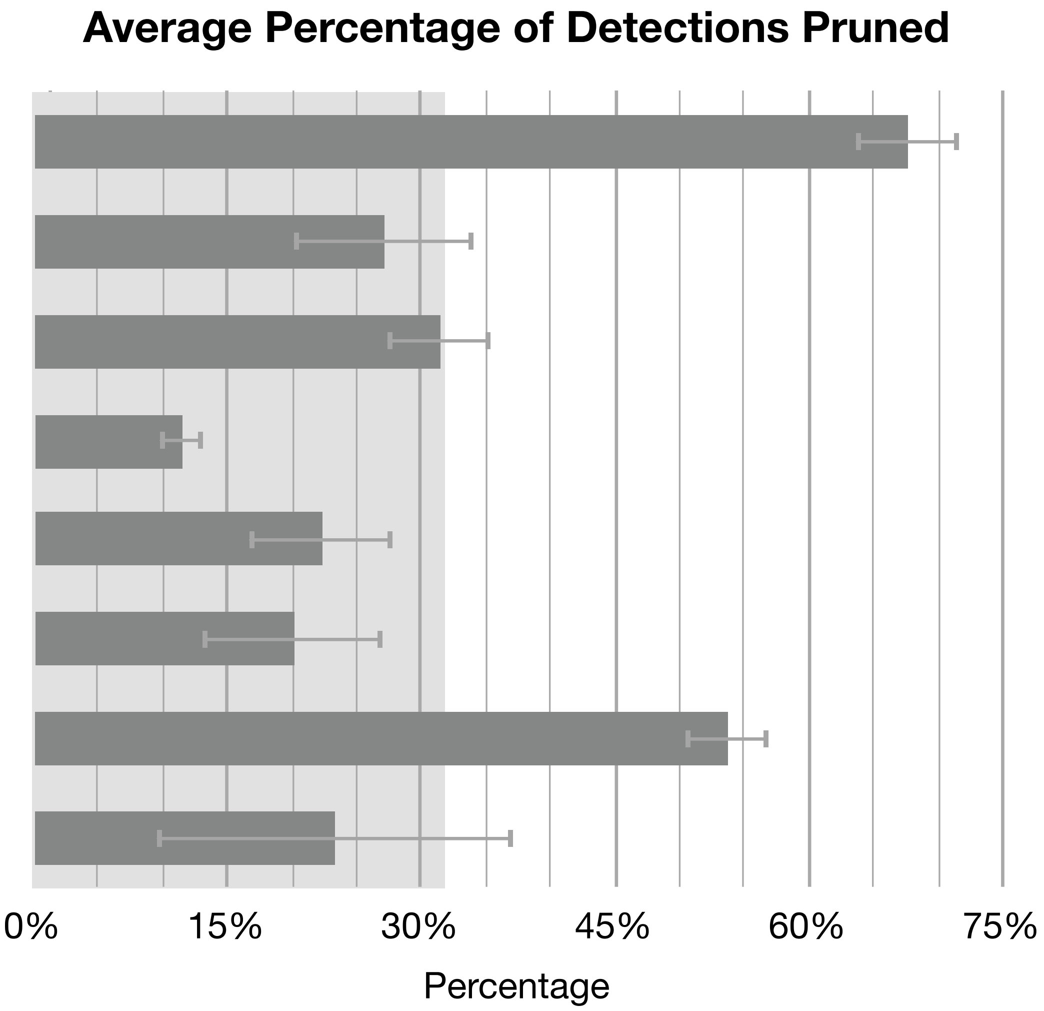

Figure 12: Average percentage of past-threshold detections pruned by considering the size of the object. The light gray rectangle reaching to 32% is the average across classes. In both cases, error bars show standard deviation across six different splits of the data.

5.1 Curvature

Curvature is a measurement of the rate of change of the orientation of the tangent vector to a curve. In the case of a surface, the curvature can be measured at any point P by computing the curvature for all curves along the surface passing through P. The principal curvatures for this point P is the maximum (K_1) and minimum (K_2) curvature for all curves passing through P. To further reduce curvature to a single measurement one can either calculate Gaussian curvature,

$$K_{gauss} = K_1 K_2 \tag{5}$$

or mean curvature,

$$K_{mean} = (K_1 + K_2)/2 \tag{6}$$

The sign of the Gaussian and mean curvature are enough to characterize the surface at a point P into one of eight fundamental surface types: peak, pit, ridge, valley, saddle ridge, saddle valley, flat or minimal [6]. Table 5.1 shows which type of surface is present for different values of Gaussian and mean curvature.

	$K_{gauss} > 0$	$K_{gauss} = 0$	$K_{gauss} < 0$
$K_{mean} < 0$	Peak	Ridge	Saddle Ridge
$K_{mean} = 0$		Flat	Minimal
$K_{mean} > 0$	Pit	Valley	Saddle Valley

Table 1: Different types of surfaces that are possible depending on the value of the surfaces mean and Gaussian curvature at a particular point.

5.2 HOC

The first step in compute a histogram of curvature is to compute curvature at every pixel. A simple computation of curvature using second derivatives is very sensitive to noise and the Kinect sensor is by no means a noiseless sensor. As a first attempt to remove noise, range images are smoothed using a simple convolution with an averaging filter. This type of preprocessing is commonly used by the computer vision community before processing intensity images. In order to further overcome the obstacle of noise, Besl describes how Gaussian and mean curvature can be computed robustly for points on a surface [6]. The equations below are from the method described in [6], with the only modification being the following 3 x 3 filter windows are used instead of 7 x 7 windows.

$$F_u = 1/8 \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}$$

$$F_v = 1/8 \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

$$F_{uu} = 1/4 \begin{pmatrix} 1 & -2 & 1 \\ 2 & -4 & 2 \\ 1 & -2 & 1 \end{pmatrix}$$

$$F_{vv} = 1/4 \begin{pmatrix} 1 & 2 & 1 \\ -2 & -4 & -2 \\ 1 & 2 & 1 \end{pmatrix}$$

$$F_{uv} = 1/4 \begin{pmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{pmatrix}$$

These filters are then convolved (denoted by $*$) with the depth Z to produce intermediate values that can be used to compute mean and gaussian curvatures in equations 8 and 9:

$$\begin{aligned} g_u(i,j) =& F_u * Z(i,j) \quad g_v(i,j) = F_v * Z(i,j) \\ g_{uu}(i,j) =& F_{uu} * Z(i,j) \quad g_{vv}(i,j) = F_{vv} * Z(i,j) \\ g_{uv}(i,j) =& F_{uv} * Z(i,j) \end{aligned} \tag{7}$$

$$K_{mean}(i,j) = \frac{(1 + g_v^2(i,j))g_{uu}(i,j) + (1 + g_u^2(i,j))g_{vv}(i,j) - 2g_u(i,j)g_v(i,j)g_{uv}(i,j)}{2(\sqrt{1 + g_u^2(i,j) + g_v^2(i,j)})^3} \tag{8}$$

$$K_{gauss}(i,j) = \frac{g_{uu}(i,j)g_{vv}(i,j) - g_{uv}^2(i,j)}{(1 + g_u^2(i,j) + g_v^2(i,j))^2} \tag{9}$$

After computing both Gaussian and mean curvature at every point in the range image, the goal is to compute some sort of histogram over a window of the image based on curvature. This work experiments with four different types of features with varying number of bins.

The feature vector for each window is actually computed for a pyramid of different resolution windows similarly to [14]. For each of the different variations of HOC, the first step in computing a feature vector for a particular level of the pyramid is to divide the window into spatial bins, or cells. More specifically the number of cells in the horizontal direction is equal to w/k, where w is the width of the window and k is some constant, in this case k = 8. The number of cells in the vertical direction is equal to h/k, where h is the height of the window. A histogram is then computed for each cell and the resulting histograms for each cell and each level of the pyramid are concatenated to create a feature vector for the entire window.

The first HOC methods are inspired by the fact that mean curvature might be a sufficient feature because if the boundary of a curve is specified, mean curvature uniquely determines the shape of the surface [6]. Since noise is such a concern when computing curvature the first two HOC features are not actually histograms, but simply averages over a spatial area. For each spatial cell(i,j), the average mean curvature is computed as

$$a_{curv}(i,j) = \frac{\sum_{\text{pixel}(x,y)\in\text{cell}(i,j)} K_{mean}(x,y)}{\sum_{\text{pixel}(x,y)\in\text{cell}(i,j)} 1} \tag{10}$$

A single number is assigned for that cell based on the average:

$$\text{HOC}_1(i,j) = \begin{cases} -1 & if\ a_{curv}(i,j) < -t \\ 0 & if\ -t < a_{curv}(i,j) < t \\ 1 & if\ a_{curv}(i,j) > t \end{cases} \tag{11}$$

Experiments were also conducted using two thresholds instead of just one. Using one threshold assigns negative, zero and positive curvature to different values (or in the case of the histograms below, different bins). Using two thresholds assigns strongly negative, weak negative, zero, weak positive and strongly positive curvature to different values. This is a intuitively desirable effect because it might bin depth discontinuities (strong curvature) into different bins than small changes in curvature that can be seen within the edges of an object. This intuition leads to the hypothesis that, without two thresholds the features would be dominated by the strong curvature at depth discontinuities, thus making HOC similar to HOG on a range image. Obviously, this should be avoided so the second HOC feature is assigned using two thresholds:

$$\text{HOC}_2(i,j) = \begin{cases} -2 & if\ a_{curv}(i,j) < -t_2 \\ -1 & if\ -t_2 < a_{curv}(i,j) < -t_1 \\ 0 & if\ -t_1 < a_{curv}(i,j) < t_1 \\ 1 & if\ t_1 < a_{curv}(i,j) < t_2 \\ 2 & if\ a_{curv}(i,j) > t_2 \end{cases} \tag{12}$$

Since the features described in equations 11 and 12 are not actually histograms, the following similar features to be experimented with are actually histograms of the average curvature in a spatial bin:

$$\text{HOC}_3(i,j,1) = \begin{cases} 1 & if\ a_{curv}(i,j) < -t \\ 0 & otherwise \end{cases}$$

$$\text{HOC}_3(i,j,2) = \begin{cases} 1 & if\ -t < a_{curv}(i,j) < t \\ 0 & otherwise \end{cases}$$

$$\text{HOC}_3(i,j,3) = \begin{cases} 1 & if\ a_{curv}(i,j) > t \\ 0 & otherwise \end{cases} \tag{13}$$

As before a fourth feature that uses two thresholds instead of one can be defined:

$$\text{HOC}_4(i,j,1) = \begin{cases} 1 & if\ a_{curv}(i,j) < -t_2 \\ 0 & otherwise \end{cases}$$

$$\text{HOC}_4(i,j,2) = \begin{cases} 1 & if\ -t_2 < a_{curv}(i,j) < -t_1 \\ 0 & otherwise \end{cases}$$

$$\text{HOC}_4(i,j,3) = \begin{cases} 1 & if\ -t_1 < a_{curv}(i,j) < t_1 \\ 0 & otherwise \end{cases}$$

$$\text{HOC}_4(i,j,4) = \begin{cases} 1 & if\ t_1 < a_{curv}(i,j) < t_2 \\ 0 & otherwise \end{cases}$$

$$\text{HOC}_4(i,j,5) = \begin{cases} 1 & if\ a_{curv}(i,j) > t_2 \\ 0 & otherwise \end{cases} \tag{14}$$

Of course, averaging might not be the right solution, a lot of signal might be lost in attempts to denoise. As mentioned before, Gaussian curvature may or may not be useful, so the following HOC features continue to use just mean curvature (K_{mean}). (Gaussian curvature will be used later.)

In the following feature descriptor, instead of averaging, a true histogram is computed by counting the number of pixels in each cell that fall into each of the three bins of the histogram:

$$\begin{aligned}\text{HOC}_5(i,j,1) &= \sum_{\text{pixel}(x,y)\in\text{cell}(i,j)} (K_{mean}(x,y) < -t)\\ \text{HOC}_5(i,j,2) &= \sum_{\text{pixel}(x,y)\in\text{cell}(i,j)} (-t < K_{mean}(x,y) < t)\\ \text{HOC}_5(i,j,3) &= \sum_{\text{pixel}(x,y)\in\text{cell}(i,j)} (K_{mean}(x,y) > t)\end{aligned} \tag{15}$$

As before, a five bin version of the feature vector can also be formulated:

$$\begin{aligned}\text{HOC}_6(i,j,1) &= \sum_{\text{pixel}(x,y)\in\text{cell}(i,j)} (K_{mean}(x,y) < -t_2)\\ \text{HOC}_6(i,j,2) &= \sum_{\text{pixel}(x,y)\in\text{cell}(i,j)} (-t_2 < K_{mean}(x,y) < -t_1)\\ \text{HOC}_6(i,j,3) &= \sum_{\text{pixel}(x,y)\in\text{cell}(i,j)} (-t_1 < K_{mean}(x,y) < t_1)\\ \text{HOC}_6(i,j,4) &= \sum_{\text{pixel}(x,y)\in\text{cell}(i,j)} (t_1 < K_{mean}(x,y) < t_2)\\ \text{HOC}_6(i,j,5) &= \sum_{\text{pixel}(x,y)\in\text{cell}(i,j)} (K_{mean}(x,y) > t_2)\end{aligned} \tag{16}$$

After experimenting with different thresholds, we found empirically that $t = t_1 = 0.005$ and $t_2 = 0.05$ worked best.

Finally, it is necessary to evaluate feature descriptors that use Gaussian curvature as well as mean curvature. To do this additional bins must be added to either HOC_5 or HOC_6. A six bin histogram of mean and gaussian curvature (K_{gauss}) is computed as follows:

$$\begin{aligned}\text{HOC}_7(i,j,k) &= \text{HOC}_5(i,j,k) \text{ for } k = 1,2,3\\ \text{HOC}_7(i,j,4) &= \sum_{\text{pixel}(x,y)\in\text{cell}(i,j)} (K_{gauss}(x,y) < -t_g)\\ \text{HOC}_7(i,j,5) &= \sum_{\text{pixel}(x,y)\in\text{cell}(i,j)} (-t_g < K_{gauss}(x,y) < t_g)\\ \text{HOC}_7(i,j,6) &= \sum_{\text{pixel}(x,y)\in\text{cell}(i,j)} (K_{gauss}(x,y) > t_g)\end{aligned} \tag{17}$$

A similar feature descriptor (HOC_8) can be computed for a 8 bin histogram using two thresholds for mean curvature:

$$\text{HOC}_8(i,j,k) = \text{HOC}_6(i,j,k) \text{ for } k = 1,2,3,4,5 \tag{18}$$

$$\text{HOC}_8(i,j,k) = \text{HOC}_7(i,j,k-2) \text{ for } k = 6,7,8 \tag{19}$$

We found empirically that $t_g = 0.00005$ worked well.

6 Histogram of Curvature Experiments

6.1 Experimental Setup and Baselines

All the experiments in this section are based on a sliding window linear SVM classifier trained in two phases, one using random negative examples and one using "hard" negatives generated using the code from Felzenszwalb et al. [14]. Two mirrored models are trained for each class and windows are constrained to a fixed aspect ratio but varying position and scale. All features are evaluated as a pyramid of scales. In contrast to the experiments in Section 4, the models computed in this section were not based on the deformable parts model. As in Section 4, nonmaximal suppression is used at test time and the same evaluation paradigm (equation 3) is used.

Two baselines were performed, both based on the use of a HOG feature descriptor that uses both contrast-sensitive and contrast-insensitive bins, and four different normalization schemes [14]. The first baseline simply ignores depth and just computes HOG features for the color image. The second baseline concatenates HOG features for both color and depth images.

Experimental results were computed for 16 different feature vectors. The first 8 consist of a HOG feature descriptor for intensity image concatenated with one of the 8 different HOC features. The second 8 features consist of the concatenation of HOG on the intensity image, HOG on the range image and one of the eight HOC features.

6.2 Results

Figure 13 shows average precision (the area under a precision recall curve), for 8 different classes of objects and all 16 feature vectors in addition to the two baselines (Intensity HOG and Intensity HOG + Depth HOG). For most categories, using HOG on intensity images and depth images in conjunction with HOC performed better than leaving out HOG on the depth images. The biggest exception to this is for computer monitors. Most of the monitors in B3DO are turned off and are thus completely black. The structured light sensor used by the Kinect does not always work well for black objects, and monitors are an example of surface that often has significant missing data. Thus, increased performance by adding a depth channel should not be expected.

In order to visualize results more clearly, Figure 14 shows results for only the features that combine HOG on intensity and depth images with HOC, as well as the baselines. The most noticeable result is that the best performance for bottle, chair, keyboard, monitor, computer mouse and phone occurs when depth is ignored. There are positive results for the categories of cup and bowl. For bowls, both HOC_4 and HOC_7 outperform the baseline that ignores depth by approximately 5% and the baseline that uses HOG on depth and no curvature by approximately 10%. Similar results can be observed for cups, but for cups the best performing features are HOC_6 and HOC_7. This result is somewhat intuitive, the shape of cups and bowls is very simple, and likely easier to learn than the shape of more complicated objects like chairs and telephones.

7 Discussion

The Berkeley 3D Object Dataset provides a challenging dataset on which to test the ability of object detectors to take advantage of 3D signal. This dataset provides a unique opportunity for researchers

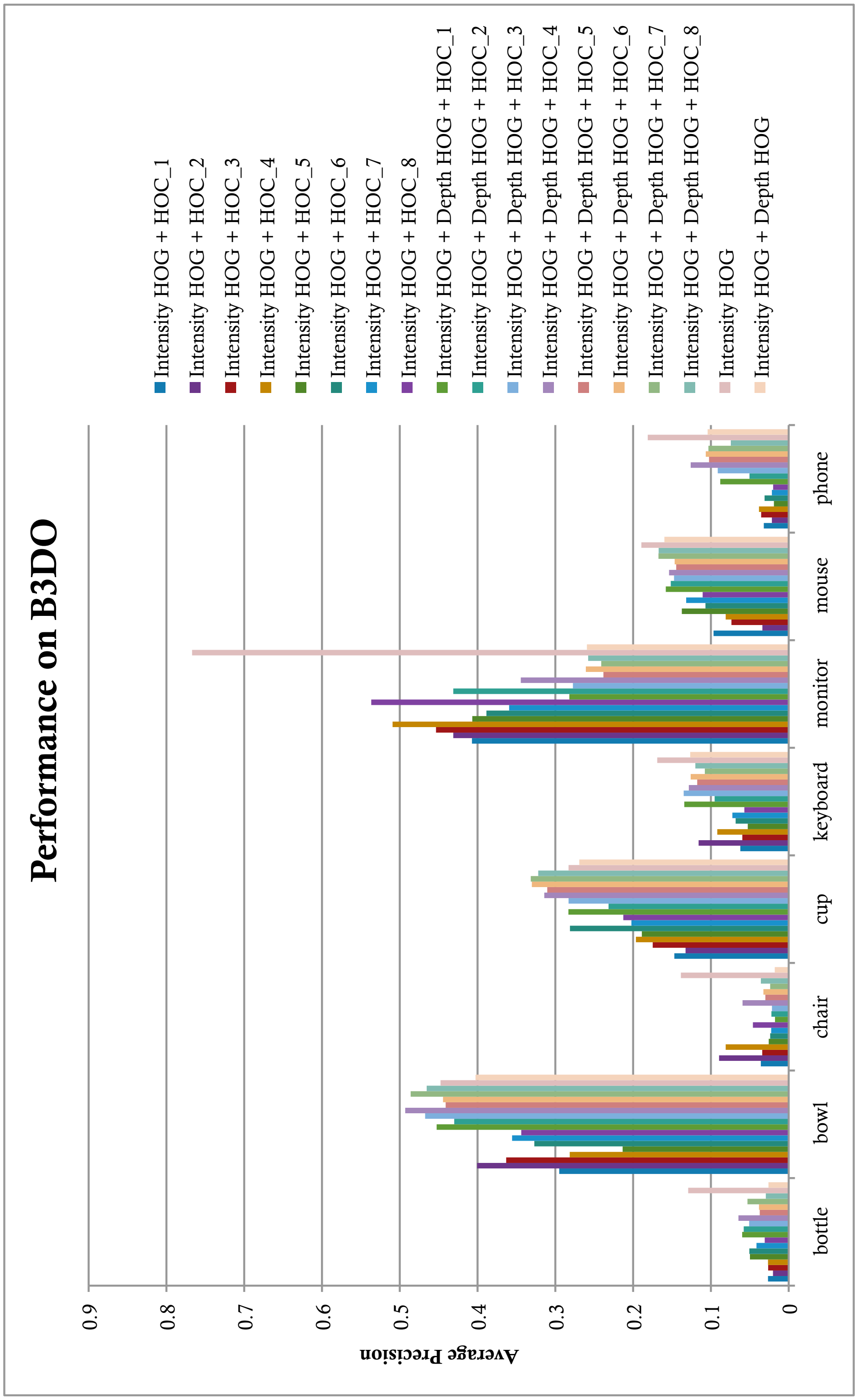

Figure 13: Average Precision for all sixteen different feature vectors as well as the two baselines. Performance is averaged over 6 different splits of the data.

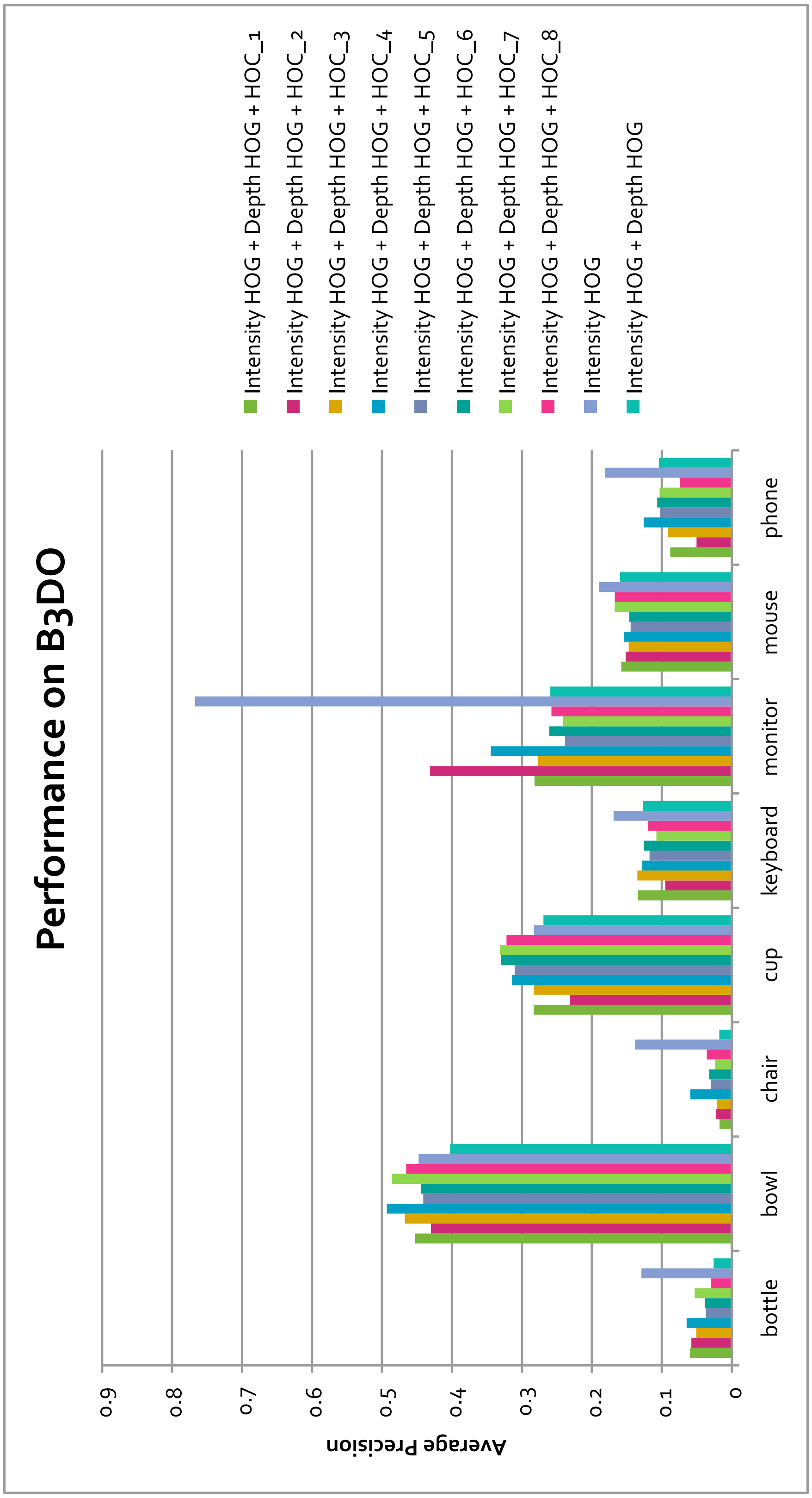

Figure 14: Similar to Figure 13, the chart shows performance just for the features that combine HOG on the intensity image and depth image with a HOC feature.

to test their methods in the face of large variation in pose and viewpoint. In addition, the lack of dense training data (for example on a turntable) and the simple collection process enables this dataset to continue to grow with contributions from the world outside the research community.

One difficult of continuing to grow the dataset is that obtaining volunteers willing and able to collect is not easy. The concept of paying Kinect owners to collect data seems promising, but using AMT as an avenue to find workers did not work as well as hoped. The first difficultly in paying novices to collect data is that an easy to use software tool must be designed for collection. This tool was produced by modifying software available online [9], but distributing the software turned out to be difficult. AMT has rules against requiring workers to download software and our hit was removed when we inadvertently broke this rule. In addition to this difficulty, is the fact that most AMT workers don't own a Kinect and there is not way to target those that do. Finally, those that do own a Kinect might not be looking for the kind of work that takes more set up time than the typical AMT hit, even if the potential to earn more money is significant. It is quite possible that paying for images is the right tack, but AMT is not the best source of workers. Advertising on websites for Xbox or Kinect enthusiasts might be more successful. A video demonstrating the Kinect being using for object recognition could be used to drum up excitement within the gaming, robotics or hacking communities. We also discussed reaching out to undergraduates at our University who could presumably be trusted to borrow a Kinect from the lab. Undergrads owning an Xbox but no Kinect might be willing to collect data if they were paid and were also able to play with the Kinect for a week or two. Future work could focus on exploring these opportunities in order to identify the ideal way to find paid workers.

Section 4 demonstrated that techniques based on estimating the size of objects can be used to slightly improve performance. Simple solutions such as computing a histogram of gradient for range images can extract some of the information present in the range image but not all. In order to extract all the available information from depth signal, features that can learn the shape of the objects that one wishes to recognize must be used. To this end, this work proposes the histogram of curvature, or HOC. Unfortunately, experiments with HOC in Section 6 have not been overly successful and the results bring up some important questions:

- Why does adding features hurt performance? (This should not happen.)
- Why do only cups and bowls perform well using HOC?
- Which HOC feature is best?

The first question is perhaps the most important one. Adding features should not hurt classification performance because at worst the classifier should learn to ignore the new feature if it isn't helpful. Before even adding HOC, we can see that adding HOG on depth images decreases performance across the board. This is contradictory to the results in [23], where it was shown that HOG on depth images will increase performance significantly. (They use a different dataset in which the training data consists of objects on a turntable. This setup provides much more training data which is uncluttered and might account for their different results.) If adding new features is decreasing performance it can be concluded that some over fitting to the training set is occurring, but what can be done about it? One answer is that there needs to be a different regularizer, but preliminary experiments with this idea yielded no results. Another possibility is that linear classifiers are not powerful enough. HOG has been hand tuned with various normalization factors in order to work well with linear classifiers, but as HOC is missing this, it may require nonlinear kernels. In addition, by simply concatenating feature vectors, the fact that the three feature vectors were obtained by different processes is lost. A multiple kernel learning framework may be better able to handle the fact that there are in fact three feature vectors without simply concatenating them.

The second question is perhaps simpler. Why do cups and bowls actually perform well with HOC? The answer is probably that they have a very simple shape and there is not a lot of variation in pose or viewpoint since they are symmetrical in multiple directions. In fact, baseline performance for bowls and cups is higher than all the other categories except for monitors. Our initial inspiration to use curvature as a feature vector was motivated by simple shapes like bowls and cups. The success of bowls and cups might also be related to the size of the dataset. More complicated shapes will obviously require more data, and perhaps the dataset does not contain enough examples of more complicated objects like chairs to learn their representation. Finally, the third question is unanswerable at this point. There are not enough categories that performed well with HOC to conclude which HOC feature is best.

References

[1] Ford campus vision and lidar dataset. http://robots.engin.umich.edu/Downloads.

[2] Solution in perception challenge. http://opencv.willow-garage.com/wiki/SolutionsInPerceptionChallenge.

[3] UBC Robot Vision Survey. http://www.cs.ubc.ca/labs/lci/vrs/index.html.

[4] Introducing Kinect for Xbox 360. http://www.xbox.com/en-US/Kinect/, 2011.

[5] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[6] P. J. Besl and R. C. Jain. Segmentation through variable-order surface fitting. *IEEE Pattern Analysis and Machine Intelligence (PAMI)*, 10, 1988.

[7] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[8] B. Browatzki, J. Fischer, G. Birgit, H. Bulthoff, and C. Wallraven. Going into depth: Evaluating 2d and 3d cues for object classification on a new, large-scale object dataset. In *ICCV Workshop on Consumer Depth Cameras for Computer Vision (CDC4CV)*, 2011.

[9] N. Burrus. Kinect RGB Demo v0.4.0. http://nicolas.burrus.name/index.php/Research/ KinectRgbDemoV4?from=Research.KinectRgbDemoV2, Feb. 2011.

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

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

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

[13] P. F. Felzenszwalb, R. B. Girschick, and D. McAllester. Cascade object detection with deformable part models. *Conference on Computer Vision and Pattern Recogntition (CVPR)*, 2010.

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

[15] M. Fritz, K. Saenko, and T. Darrell. Size matters: Metric visual search constraints from monocular metadata. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[16] A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik. Recognizing objects in range data using regional point descriptors. In *European Conference on Computer Vision (ECCV)*, 2004.

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

[18] W. Grimson. *From images to surfaces: A computational study of the human early visual system*. MIT Press, 1981.

[19] H. Hattori, A. Seki, M. Nishiyama, and T. Watanabe. Stereo-based pedestrian detection using multiple patterns. In *British Machine Vision Conference (BMVC)*, 2009.

[20] S. Helmer, D. Meger, M. Muja, J. J. Little, and D. G. Lowe. Multiple viewpoint recognition and localization. In *Asian Conference on Computer Vision (ACCV)*, 2010.

[21] A. Janoch, S. Karayev, Y. Jia, J. T. Barron, M. Fritz, K. Saenko, and T. Darrell. A category-level 3-D object dataset: Putting the kinect to work. In *ICCV Workshop on Consumer Depth Cameras for Computer Vision (CDC4CV)*, 2011.

[22] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Pattern Analysis and Machine Intelligence (PAMI)*, 21(5):433 –449, May 1999.

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

[24] C. H. Lampert, M. B. Blaschko, and T. Hoffman. Beyond sliding windows: Object localization by efficient subwindow search. *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, Mar 2008.

[25] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, Jan 2004.

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

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

[28] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[29] N. Silberman and R. Fergus. Indoor scene segmentation using a structured light sensor. In *ICCV Workshop on 3D Representation and Recognition (3DRR)*, 2011.

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

[31] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision*, pages 1–25, Nov 2001.

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

[33] X. Wang, T. X. Han, and S. Yan. An HOG-LBP human detector with partial occlusion handling. *International Conference on Computer Vision (ICCV)*, 2009.

[34] O. Woodford, P. Torr, I. Reid, and A. Fitzgibbon. Global stereo reconstruction under second-order smoothness priors. *IEEE Pattern Analysis and Machine Intelligence (PAMI)*, 2009.