SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite

Shuran Song Samuel P. Lichtenberg Jianxiong Xiao
Princeton University
http://rgbd.cs.princeton.edu

Abstract

Although RGB-D sensors have enabled major breakthroughs for several vision tasks, such as 3D reconstruction, we have not attained the same level of success in high-level scene understanding. Perhaps one of the main reasons is the lack of a large-scale benchmark with 3D annotations and 3D evaluation metrics. In this paper, we introduce an RGB-D benchmark suite for the goal of advancing the state-of-the-arts in all major scene understanding tasks. Our dataset is captured by four different sensors and contains 10,335 RGB-D images, at a similar scale as PASCAL VOC. The whole dataset is densely annotated and includes 146,617 2D polygons and 64,595 3D bounding boxes with accurate object orientations, as well as a 3D room layout and scene category for each image. This dataset enables us to train data-hungry algorithms for scene-understanding tasks, evaluate them using meaningful 3D metrics, avoid overfitting to a small testing set, and study cross-sensor bias.

1. Introduction

Scene understanding is one of the most fundamental problems in computer vision. Although remarkable progress has been achieved in the past decades, general-purpose scene understanding is still considered to be very challenging. Meanwhile, the recent arrival of affordable depth sensors in consumer markets enables us to acquire reliable depth maps at a very low cost, stimulating breakthroughs in several vision tasks, such as body pose recognition [56, 58], intrinsic image estimation [4], 3D modeling [27] and SfM reconstruction [72].

RGB-D sensors have also enabled rapid progress for scene understanding (e.g. [20, 19, 53, 38, 30, 17, 32, 49]). However, while we can crawl color images from the Internet easily, it is not possible to obtain large-scale RGB-D data online. Consequently, the existing RGB-D recognition benchmarks, such as NYU Depth v2 [49], are an order-of-magnitude smaller than modern recognition datasets for color images (e.g. PASCAL VOC [9]). Although these small datasets successfully bootstrapped initial progress in RGB-D scene understanding in the past few years, the size limit is now becoming the critical common bottleneck in advancing research to the next level. Besides causing overfitting of the algorithm during evaluation, they cannot support training data-hungry algorithms that are currently the state-of-the-arts in color-based recognition (e.g. [15, 36]). If a large-scale RGB-D dataset were available, we could borrow the same success to the RGB-D domain as well. (Table 1 shows the performance improvement for a RGB-D deep learning algorithm [20] when a bigger training set is used.) Furthermore, although the RGB-D images in these datasets contain depth maps, the annotation and evaluation metrics are mostly in 2D image domain, but not directly in 3D (Figure 1). Scene understanding is much more useful in the real 3D space for most applications. We desire to reason about scenes and evaluate algorithms in 3D.

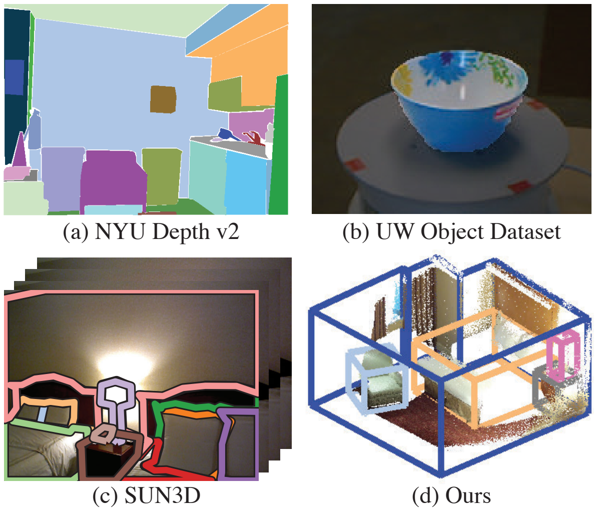

(a) NYU Depth v2
(b) UW Object Dataset
(c) SUN3D
(d) Ours

Figure 1. **Comparison of RGB-D recognition benchmarks.** Apart from 2D annotation, our benchmark provided high quality 3D annotation for both objects and room layout.

To this end, we introduce SUN RGB-D, a dataset containing 10,335 RGB-D images with dense annotations in both 2D and 3D, for both objects and rooms. Based on this dataset, we focus on six important recognition tasks

Training Set Testing Set	NYU (795 images)	SUN RGB-D (5,285 images)
NYU	32.50	**34.33**
SUN RGB-D	15.78	**33.20**

Table 1. **Performance improves as the size of training data increases.** We trained the Depth-RCNN [20] for 2D object detection using RGB-D images, and evaluated the mean average precision. Bigger training set produces better result. Especially for the first row using NYU as the testing set, the performance is still better using the bigger SUN RGB-D that is a superset of NYU, despite the domain gap due to dataset bias.

	RealSense	Xtion	Kinect v1	Kinect v2
weight (pound)	0.077	0.5	4	4.5
size (inch)	5.2×0.25×0.75	7.1×1.4×2	11×2.3×2.7	9.8×2.7×2.7
power	2.5W USB	2.5W USB	12.96W	115W
depth resolution	628×468	640×480	640×480	512×424
color resolution	1920×1080	640×480	640×480	1920×1080

Table 2. **Specification of sensors.** RealSense is very light, while Kinect v2 is heavier and has much higher power consumption.

towards total scene understanding, which recognizes objects, room layouts and scene categories. For each task, we propose metrics in 3D and evaluate baseline algorithms derived from the state-of-the-arts. Since there are several popular RGB-D sensors available, each with different size and power consumption, we construct our dataset using four different kinds of sensors to study how well the algorithms generalize across sensors. By constructing a PASCAL-scale dataset and defining a benchmark with 3D evaluation metrics, we hope to lay the foundation for advancing RGB-D scene understanding in the coming years.

1.1. Related work

There are many interesting works on RGB-D scene understanding, including semantic segmentation [53, 49, 19] object classification [69], object detection [59, 20, 62], context reasoning [38], mid-level recognition [32, 31], and surface orientation and room layout estimation [13, 14, 74]. Having a solid benchmark suite to evaluate these tasks will be very helpful in further advancing the field.

There are many existing RGB-D datasets available [54, 47, 1, 25, 44, 60, 49, 45, 29, 66, 57, 52, 46, 16, 21, 73, 35, 67, 3, 41, 10, 63, 42, 64, 65, 48, 12, 33, 8, 50, 26, 6]. Figure 1 shows some of them. Here we will briefly describe several most relevant ones[1]. There are datasets [61, 37] that capture objects on a turntable instead of real-world scenes. For natural indoor scene datasets, NYU Depth v2 [49] is probably the most popular one. They labeled 1,449 selected frames from short RGB-D videos using 2D semantic segmentation on the image domain. [18] annotates each object by aligning a CAD model with the 3D point cloud. However, the 3D annotation is quite noisy, and in our benchmark we reuse the 2D segmentation but recreate the 3D annotation by ourselves. Although this dataset is very good, the size is still small compared to other modern recognition datasets, such as PASCAL VOC [9] or ImageNet [7]. B3DO [28] is another dataset with 2D bounding box annotations on the RGB-D images. But its size is smaller than NYU and it has many images with an unrealistic scene layouts (e.g. snapshot of a computer mouse on the floor). The Cornell RGBD dataset [2, 34] contains 52 indoors scenes with per-point annotations on the stitched point clouds. SUN3D [72] contains 415 RGB-D video sequence with 2D polygon annotation on some key frames. Although they stitched the point cloud in 3D, the annotation is still purely in the 2D image domain, and there are only 8 annotated sequences.

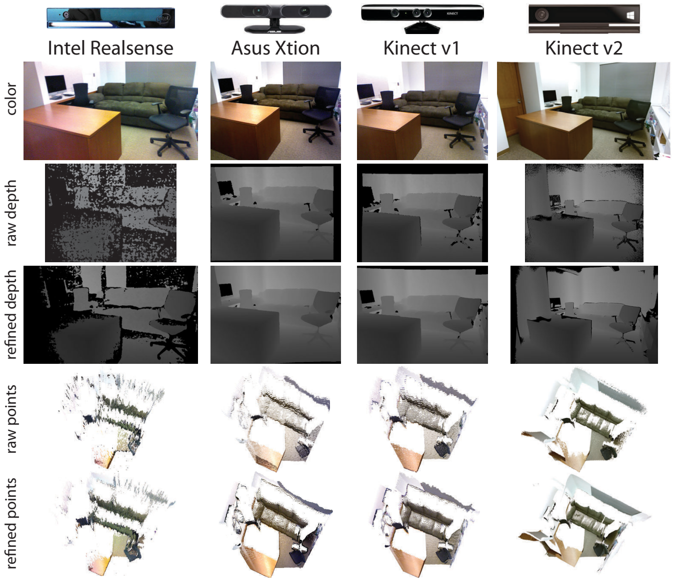


Figure 2. **Comparison of the four RGB-D sensors.** The raw depth map from Intel RealSense is noisier and has more missing values. Asus Xtion and Kinect v1's depth map have observable quantization effect. Kinect v2 is more accurate to measure the details in depth, but it is more sensitive to reflection and dark color. Across different sensors our depth improvement algorithm manages to robustly improve the depth map quality.

2. Dataset construction

The goal of our dataset construction is to obtain an image dataset captured by various RGB-D sensors at a similar scale as the PASCAL VOC object detection benchmark. To improve the depth map quality, we take short videos and use multiple frames to obtain a refined depth map. For each image, we annotate the objects with both 2D polygons and 3D bounding boxes and the room layout with 3D polygons.

2.1. Sensors

Since there are several popular sensors available, with different size and power consumption, we construct our dataset using four kinds – Intel RealSense 3D Camera for

[1] A full list with brief descriptions is available at http://www0.cs.ucl.ac.uk/staff/M.Firman/RGBDdatasets/.

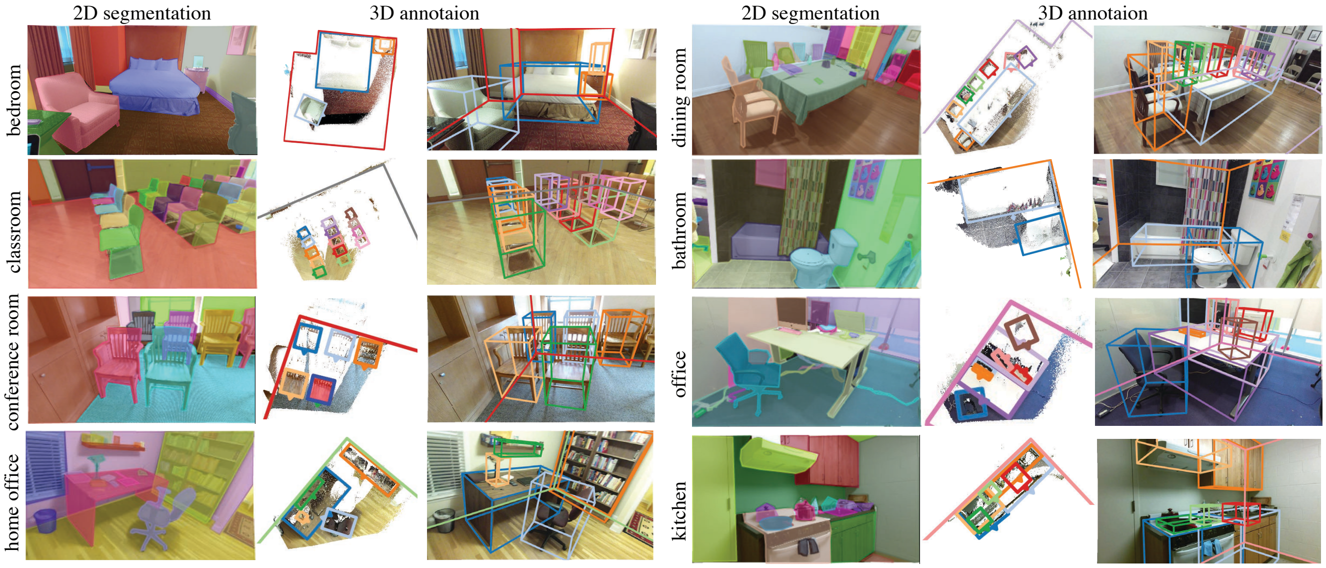


Figure 3. **Example images with annotation from our dataset.**

tablets, Asus Xtion LIVE PRO for laptops, and Microsoft Kinect versions 1 and 2 for desktop. Table 2 shows each sensor's specification. Figure 2 shows the example color and depth images captured.

Intel RealSense is a lightweight, low power consuming depth sensor designed for tablets. It will soon reach consumers; we obtained two pre-release samples from Intel. It projects an IR pattern to the environment and uses stereo matching to obtain the depth map. For outdoor environments, it can switch automatically to stereo matching without IR pattern; however, we visually inspect the 3D point cloud and believe the depth map quality is too low for use in accurate object recognition for outdoors. We thus only use this sensor to capture indoor scenes. Figure 2 shows its raw depth is worse than that of other RGB-D sensors, and the effective range for reliable depth is shorter (depth gets very noisy around 3.5 meters). But this type of lightweight sensor can be embedded in portable devices and be deployed at a massive scale in consumer markets, so it is important to study algorithm performance with it.

Asus Xtion and Kinect v1 use a near-IR light pattern. Asus Xtion is much lighter and powered by USB only, with worse color image quality than Kinect v1's. However, Kinect v1 requires an extra power source. The raw depth maps from both sensors have an observable quantization effect.

Kinect v2 is based on time-of-flight and also consumes significant power. The raw depth map captured is more accurate, with high fidelity to measure the detailed depth difference, but fails more frequently for black objects and slightly reflective surfaces. The hardware supports long distance depth range, but the official Kinect for Windows SDK cuts the depth off at 4.5 meters and applies some filtering that tends to lose object details. Therefore, we wrote our own driver and decoded the raw depth in GPU (Kinect v2 requires software depth decoding) to capture real-time video without depth cutoffs or additional filtering.

2.2. Sensor calibration

For RGB-D sensors, we must calibrate the camera intrinsic parameters and the transformation between the depth and color cameras. For Intel RealSense, we use the default factory parameters. For Asus Xtion, we rely on the default parameters returned by OpenNI library without modeling radial distortion. For Kinect v2, the radial distortion is very strong. So we calibrate all cameras with standard calibration toolbox [5]. We calibrate the depth cameras by computing the parameters with the IR image which is the same with the depth camera. To see the checkerboard without overexposure on IR, we cover the emitter with a piece of paper. We use the stereo calibration function to calibrate the transformation between the depth (IR) and the color cameras.

2.3. Depth map improvement

The depth maps from these cameras are not perfect, due to measurement noise, view angle to the regularly reflective surface, and occlusion boundary. Because all the RGB-D sensors operate as a video camera, we can use nearby frames to improve the depth map, providing redundant data to denoise and fill in missing depth.

We propose a robust algorithm for depth map integration from multiple RGB-D frames. For each nearby frame in a time window, we project the points to 3D, get the triangulated mesh from nearby points, and estimate the 3D rotation and translation between this frame and the target frame for depth improvement. Using this estimated transformation, we render the depth map of the mesh from the target frame camera. After we obtain aligned and warped depth maps,

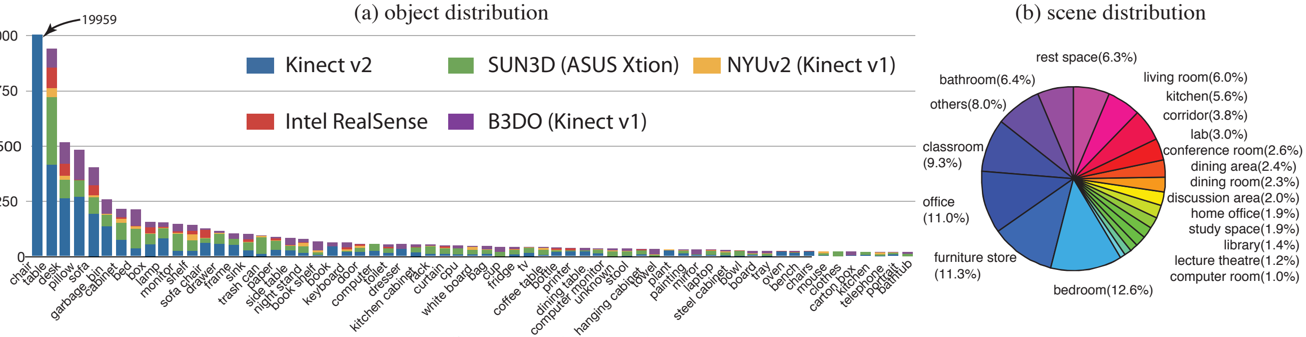
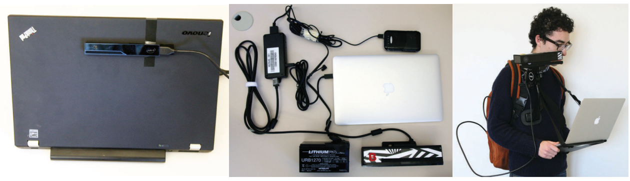


Figure 4. **Statistics of semantic annotation in our dataset.**



(a) (b) (c)

Figure 5. **Data Capturing Process.** (a) RealSense attached to laptop, (b) Kinect v2 with battery, (c) Capturing setup for Kinect v2.

we integrate them to get a robust estimation. For each pixel location, we compute the median depth and 25% and 75% percentiles. If the raw target depth is missing or outside the 25% − 75% range and the median is computed from at least 10 warped depth maps, we use the median depth value. Otherwise, we keep the original value to avoid oversmoothing. Examples are shown in Figure 2. Our depth map improvement algorithm, compared to [72] which uses a 3D voxel-based TSDF representation, requires much less memory and runs faster at equal resolution, enabling much high-resolution integration.

Robust estimation of an accurate 3D transformation between a nearby frame and target frame is critical for this algorithm. To do this, we first use SIFT to obtain point-to-point correspondences between the two color images, obtain the 3D coordinates for the SIFT keypoints from the raw depth map, and then estimate the rigid 3D rotation and translation between these two sparse 3D SIFT clouds using RANSAC with three points. To obtain a more accurate estimation, we would like to use the full depth map to do dense alignment with ICP, but depending on the 3D structure, ICP can have severe drifting. Therefore, we first use the estimation from SIFT+RANSAC to initialize the transformation for ICP, and calculate the percentage of points for ICP matching. Using the initialization and percentage threshold, we run point-plane ICP until convergence, then check the 3D distances with the original SIFT keypoint inliers from RANSAC. If the distances significantly increase, it means ICP makes the result drift away from the truth; we will use the original RANSAC estimation without ICP. Otherwise, we use the ICP result.

2.4. Data acquisition

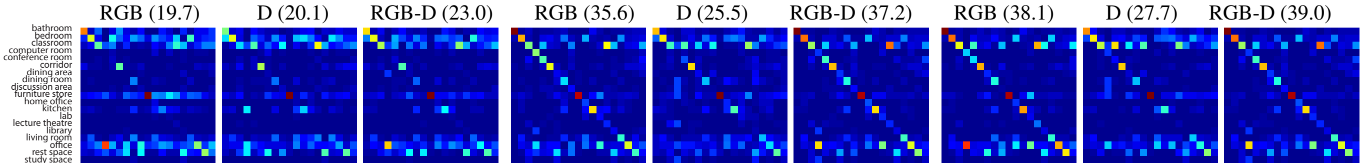
To construct a dataset at the PASCAL VOC scale, we capture a significant amount of new data by ourselves and combine some existing RGB-D datasets. We capture 3,784 images using Kinect v2 and 1,159 images using Intel RealSense. We included the 1,449 images from the NYU Depth V2 [49], and also manually selected 554 realistic scene images from the Berkeley B3DO Dataset [28], both captured by Kinect v1. We manually selected 3,389 distinguished frames without significant motion blur from the SUN3D videos [72] captured by Asus Xtion. In total, we obtain 10,335 RGB-D images.

As shown in Figure 5, we attach an Intel RealSense to a laptop and carry it around to capture data. For Kinect v2 we use a mobile laptop harness and camera stabilizer. Because Kinect v2 consumes a significant amount of power, we use a 12V car battery and a 5V smartphone battery to power the sensor and the adaptor circuit. The RGB-D sensors only work well for indoors. And we focus on universities, houses, and furniture stores in North America and Asia. Some example images are shown in Figure 3.

2.5. Ground truth annotation

For each RGB-D image, we obtain LabelMe-style 2D polygon annotations, 3D bounding box annotations for objects, and 3D polygon annotations for room layouts. To ensure annotation quality and consistency, we obtain our own ground truth labels for images from other datasets; the only exception is NYU, whose 2D segmentation we use.

For 2D polygon annotation, we developed a LabelMe-style [55] tool for Amazon Mechanical Turk. To ensure high label quality, we add automatic evaluation in the tool. To finish the HIT, each image must have at least 6 objects labeled; the union of all object polygons must cover at least 80% of the total image. To prevent workers from cheating by covering everything with big polygons, the union of the small polygons (area $<$ 30% of the image) must cover at least 30% of the total image area. Finally, the authors visually inspect the labeling result and manually correct the layer ordering when necessary. Low quality labelings are sent back for relabeling. We paid $0.10 per image;



(a) GIST[51] + RBF kernel SVM (b) Places-CNN[75] + Linear SVM (c) Places-CNN[75] + RBF kernel SVM

Figure 6. **Confusion matrices for various scene recognition algorithms.** Each combination of features and classifiers is run on RGB, D and RGB-D. The numbers inside the parentheses are the average accuracy for classification.

some images required multiple labeling iterations to meet our quality standards.

For 3D annotation, the point clouds are first rotated to align with the gravity direction using an automatic algorithm. We estimate the normal direction for each 3D point with the 25 closest 3D points. Then we accumulate a histogram on a 3D half-sphere and pick the maximal count from it to obtain the first axis. For the second axis, we pick the maximal count from the directions orthogonal to the first axis. In this way, we obtain the rotation matrix to rotate the point cloud to align with the gravity direction. We manually adjust the rotation when the algorithm fails.

We design a web-based annotation tool and hire oDesk workers to annotate objects and room layouts in 3D. For objects, the tool requires drawing a rectangle on the top view with an orientation arrow, and adjusting the top and bottom to inflate it to 3D. For room layouts, the tool allows arbitrary polygon on the top view to describe the complex structure of the room (Figure 3). Our tool also shows the projection of the 3D boxes to the image in real time, to provide intuitive feedback during annotation. We hired 18 oDesk workers and trained them over Skype. The average hourly rate is $3.90, and they spent 2,051 hours in total. Finally, all labeling results are thoroughly checked and corrected by the authors. For scene categories, we manually classify the images into basic-level scene categories.

2.6. Label statistics

For the 10,335 RGB-D images, we have 146,617 2D polygons and 64,595 3D bounding boxes (with accurate orientations for objects) annotated. Therefore, there are 14.2 objects in each image on average. In total, there are 47 scene categories and about 800 object categories. Figure 4 shows the statistics for the semantic annotation of the major object and scene categories.

3. Benchmark design

To evaluate the whole scene understanding pipeline, we select six tasks, including both popular existing tasks and new but important tasks, both single-object based tasks and scene tasks, as well as a final total scene understanding task that integrates everything.

Scene Categorization Scene categorization is a very popular task for scene understanding [70]. In this task, we are given an RGB-D image, classify the image into one of the predefined scene categories, and use the standard average categorization accuracy for evaluation.

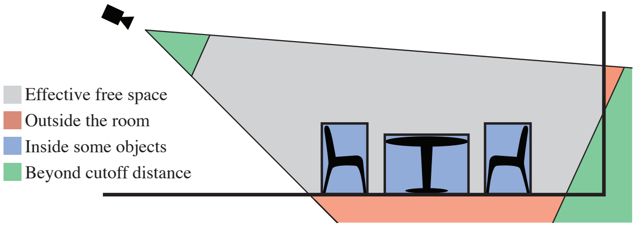


Figure 7. **Free space evaluation.** The free space is the gray area inside the room, outside any object bounding boxes, and within the effective minimal and maximal range [0.5m-5.5m]. For evaluation, we use IoU between the gray areas of the ground truth and the prediction as the criteria.

Semantic Segmentation Semantic segmentation in the 2D image domain is currently the most popular task for RGB-D scene understanding. In this task, the algorithm outputs a semantic label for each pixel in the RGB-D image. We use the standard average accuracy across object categories for evaluation.

Object Detection Object detection is another important step for scene understanding. We evaluate both 2D and 3D approaches by extending the standard evaluation criteria for 2D object detection to 3D. Assuming the box aligns with the gravity direction, we use the 3D intersection over union of the predicted and ground truth boxes for 3D evaluation.

Object Orientation Besides predicting the object location and category, another important vision task is to estimate its pose. For example, knowing the orientation of a chair is critical to sit on it properly. Because we assume that an object bounding box is aligned with gravity, there is only one degree of freedom in estimating the yaw angle for orientation. We evaluate the prediction by the angle difference between the prediction and the ground truth.

Room Layout Estimation The spatial layout of the entire space of the scene allows more precise reasoning about free space (e.g., where can I walk?) and improved object reasoning. It is a popular but challenging task for color-based scene understanding (e.g. [22, 23, 24]). With the extra depth information in the RGB-D image, this task is considered to be much more feasible [74]. We evaluate the room layout estimation in 3D by calculating the Intersection over Union

	▦	⌂	🪑	⊤	🛏	🗄	📖	🧍	mean
RGB NN	45.03	27.89	16.89	18.51	21.77	1.06	4.07	0	8.32
Depth NN	42.6	9.65	21.51	12.47	6.44	2.55	0.6	0.3	5.32
RGB-D NN	45.78	35.75	19.86	19.29	23.3	1.66	6.09	0.7	8.97
RGB [40]	47.22	39.14	17.21	20.43	21.53	1.49	5.94	0	9.33
Depth [40]	43.83	13.9	22.31	12.88	6.3	1.49	0.45	0.25	5.98
RGB-D [40]	48.25	49.18	20.8	20.92	23.61	1.83	8.73	0.77	10.05
RGB-D [53]	78.64	84.51	33.15	34.25	42.52	25.01	35.74	35.71	36.33

Table 3. **Semantic segmentation.** We evaluate performance for 40 object categories. Here shows 8 selected ones: floor, ceiling, chair, table, bed, nightstand, books, and person. The mean accuracy is for all the 40 categories. A full table is in the supp. material.

	🪑	⊤	🛏	🚽	🛋	mAP
Sliding Shapes [62]	33.42	25.78	42.09	61.86	23.28	37.29

Table 4. **3D object detection.**

(IoU) between the free space from the ground truth and the free space predicted by the algorithm output.

As shown in Figure 7, the free space is defined as the space that satisfies four conditions: 1) within camera field of view, 2) within effective range, 3) within the room, and 4) outside any object bounding box (for room layout estimation, we assume empty rooms without objects). In terms of implementation, we define a voxel grid of $0.1 \times 0.1 \times 0.1$ meter^3 over the space and choose the voxels that are inside the field of view of the camera and fall between 0.5 and 5.5 meters from the camera, which is an effective range for most RGB-D sensors. For each of these effective voxels, given a room layout 3D polygon, we check whether the voxel is inside. In this way, we can compute the intersection and the union by counting 3D voxels.

This evaluation metric directly measures the free space prediction accuracy. However, we care only about the space within a 5.5 meter range; if a room is too big, all effective voxels will be in the ground truth room. If an algorithm predicts a huge room beyond 5.5 meters, then the IoU will be equal to one, which introduces bias: algorithms will favor a huge room. To address this issue, we only evaluate algorithms on the rooms with reasonable size (not too big), since none of the RGB-D sensors can see very far either. If the percentage of effective 3D voxels in the ground truth room is bigger than 95%, we discard the room in our evaluation.

Total Scene Understanding The final task for our scene understanding benchmark is to estimate the whole scene including objects and room layout in 3D [38]. This task is also referred to "Basic Level Scene Understanding" in [71]. We propose this benchmark task as the final goal to integrate both object detection and room layout estimation to obtain a total scene understanding, recognizing and localizing all the objects and the room structure.

We evaluate the result by comparing the ground truth objects and the predicted objects. To match the prediction with ground truth, we compute the IoU between all pairs of predicted boxes and ground truth boxes, and we sort the IoU scores in a descending order. We choose each available pair with the largest IoU and mark the two boxes as unavailable. We repeat this process until the IoU is lower than a threshold τ ($\tau = 0.25$ in this case). For each matched pair between ground truth and prediction, we compare their object label in order to know whether it is a correct prediction or not. Let $|\mathcal{G}|$ be the number of ground truth boxes, $|\mathcal{P}|$ be the number of prediction boxes, $|\mathcal{M}|$ be the number of matched pairs with IoU$> \tau$, and $|\mathcal{C}|$ be the number of matched pairs with a correct label. We evaluate the algorithms by computing three numbers: $R_r = |\mathcal{C}| \,/\, |\mathcal{G}|$ to measure the recall of recognition for both semantics and geometry, $R_g = |\mathcal{M}|/|\mathcal{G}|$ to measure the geometric prediction recall, and $P_g = |\mathcal{M}|/|\mathcal{P}|$ to measure the geometric prediction precision. We also evaluate the free space by using a similar scheme as for room layout: counting the visible 3D voxels for the free space, i.e. inside the room polygon but outside any object bounding box. Again, we compute the IoU between the free space of ground truth and prediction.

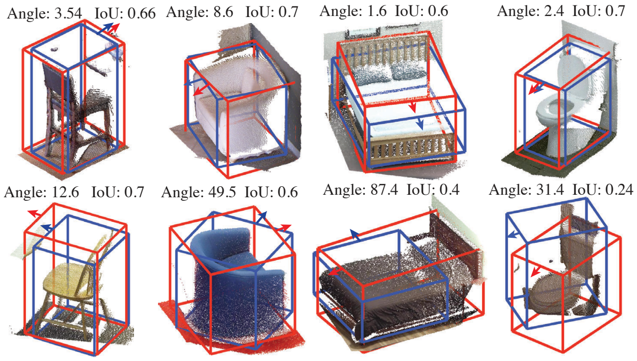


Figure 8. **Example results for 3D object detection and orientation prediction.** We show the angle difference and IoU between predicted boxes (blue) and ground truth (red).

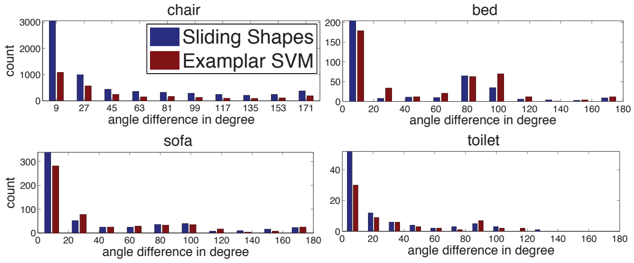


Figure 9. **Object orientation estimation.** Here we show the distribution of the orientation errors for all true positive detections.

4. Experimental evaluation

We choose some state-of-the-art algorithms to evaluate each task. For the tasks without existing algorithm or implementation, we adapt popular algorithms from other tasks. For each task, whenever possible, we try to evaluate algorithms using color, depth, as well as RGB-D images to study the relative importance of color and depth, and gauge to what extent the information from both is complementary.

																				mAP
RGB-D ESVM	7.38	12.95	7.44	0.09	12.47	0.02	0.86	0.57	1.87	6.01	6.12	0.41	6.00	1.61	6.19	14.02	11.89	0.75	14.79	5.86
RGB-D DPM	34.23	54.74	14.40	0.45	29.30	0.87	4.75	0.43	1.82	13.25	23.38	11.99	23.39	9.36	15.59	21.62	24.04	8.73	23.79	16.64
RGB-D RCNN[20]	**49.56**	**75.97**	**34.99**	**5.78**	**41.22**	**8.08**	**16.55**	**4.17**	**31.38**	**46.83**	**21.98**	**10.77**	**37.17**	**16.5**	**41.92**	**42.2**	**43.02**	**32.92**	**69.84**	**35.20**

Table 5. **Evaluation of 2D object detection.** We evaluate on 19 popular object categories using Average Precision (AP): bathtub, bed, bookshelf, box, chair, counter, desk, door, dresser, garbage bin, lamp, monitor, night stand, pillow, sink, sofa, table, tv and toilet.

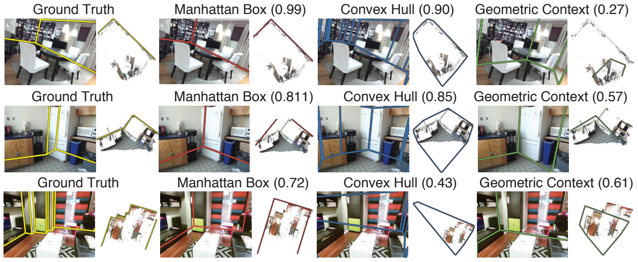


Figure 10. **Example visualization** to compare the three 3D room layout estimation algorithms.

	RGB-D RCNN				Sliding Shapes			
	(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)
P_g	21.5	21.7	21..4	22.3	33.2	37.7	33.2	**37.8**
R_g	38.2	39.4	**40.8**	39.0	32.5	32.4	32.5	32.3
R_r	21.5	**32.6**	20.4	21.4	23.7	23.7	23.7	23.7
IoU	59.5	60.5	59.5	59.8	65.1	65.8	65.2	**66.0**

Table 6. **Evaluation of total scene understanding.** With the objects detection result from Sliding Shape and RCNN and Manhattan Box for room layout estimation, we evaluate four ways to integrate object detection and room layout: (1) directly combine (2) constrain the object using room. (3) adjust room base on the objects (4) adjust the room and objects together.

Various evaluation results show that we can apply standard techniques designed for color (e.g. hand craft features, deep learning features, detector, sift flow label transfer) to depth domain and it can achieve comparable performance for various tasks. In most of cases, when we combining these two source of information, the performance get improved.

For evaluation, we carefully split the data into training and testing set, ensuring each sensor has around half for training and half for testing, Since some images are captured from the same building or house with similar furniture styles, to ensure fairness, we carefully split the training and testing sets by making sure that those images from the same building either all go into the training set or the testing set and do not spread across both sets. For data from NYU Depth v2 [49], we use the original split.

Scene Categorization For this task, we use the 19 scene categories with more than 80 images. We choose GIST [51] with a RBF kernel one-vs-all SVM as the baseline. We also choose the state-of-the-art Places-CNN [75] scene feature, which achieves the best performance in color-based scene classification on the SUN database [70]. This feature is learned using a Deep Convolutional Neural Net (AlexNet [36]) with 2.5 million scene images [75]. We use both linear SVM and RBF kernel SVM with this CNN feature. Also, empirical experiments [20] suggest that both traditional image features and deep learning features for color image can be used to extract powerful features for depth maps as well. Therefore, we also compute the GIST and Places-CNN on the depth images. We also evaluate the concatenation of depth and color features. The depth image is encoded as HHA image as in [20] before extract the feature. Figure 6 reports the accuracy for these experiments. We can see that the deep learning features indeed perform much better, and the combination of color and depth features also helps.

Semantic Segmentation We run the state-of-the-art algorithm for semantic segmentation [53] on our benchmark and report the result on Table 3. Since our dataset is quite large, we expect non-parametric label transfer to work well. We first use Places-CNN features [75] to find the nearest neighbor and directly copy its segmentation as the result. We surprisingly found that this simple method performs quite well, especially for big objects (e.g. floor, bed). We then adapt the SIFT-flow algorithm [40, 39], on both color and depth to estimation flow. But it only slightly improves performance.

Object Detection We evaluate four state-of-the-art algorithms for object detection: DPM [11], Exemplar SVM [43], RGB-D RCNN [20], and Sliding Shapes [62]. For DPM and Exemplar SVM, we use the depth as another image channel and concatenate HOG computed from that and from color images. To evaluate the first three 2D algorithms, we use 2D IoU with a threshold of 0.5 and the results are reported in Table 5. The 2D ground truth box is obtained by projecting the points inside the 3D ground truth box back to 2D and finding a tight box that encompasses these 2D points. For 3D detection, we evaluate the state-of-the-art Sliding Shapes algorithm, using the CAD models originally used in [62], and evaluate the algorithm for their five categories. We use 3D boxes for evaluation with 0.25 for the IoU as in [62], results are reported in Table 4.

Object Orientation We evaluate two exemplar-based approaches: Exemplar SVM [43] and Sliding Shapes [62]. We transfer the orientations from the training exemplars to the predicted bounding boxes. Some categories (e.g. round table) do not have well-defined orientations and are not included for evaluation. Figure 8 shows example results, and Figure 9 shows the distribution of prediction error.

Room Layout Estimation Although there exists an algorithm for this task [74], we could not find an open source implementation. Therefore, we design three baselines: the simplest baseline (named Convex Hull) computes the floor and ceiling heights by taking the 0.1 and 99.9 percentiles

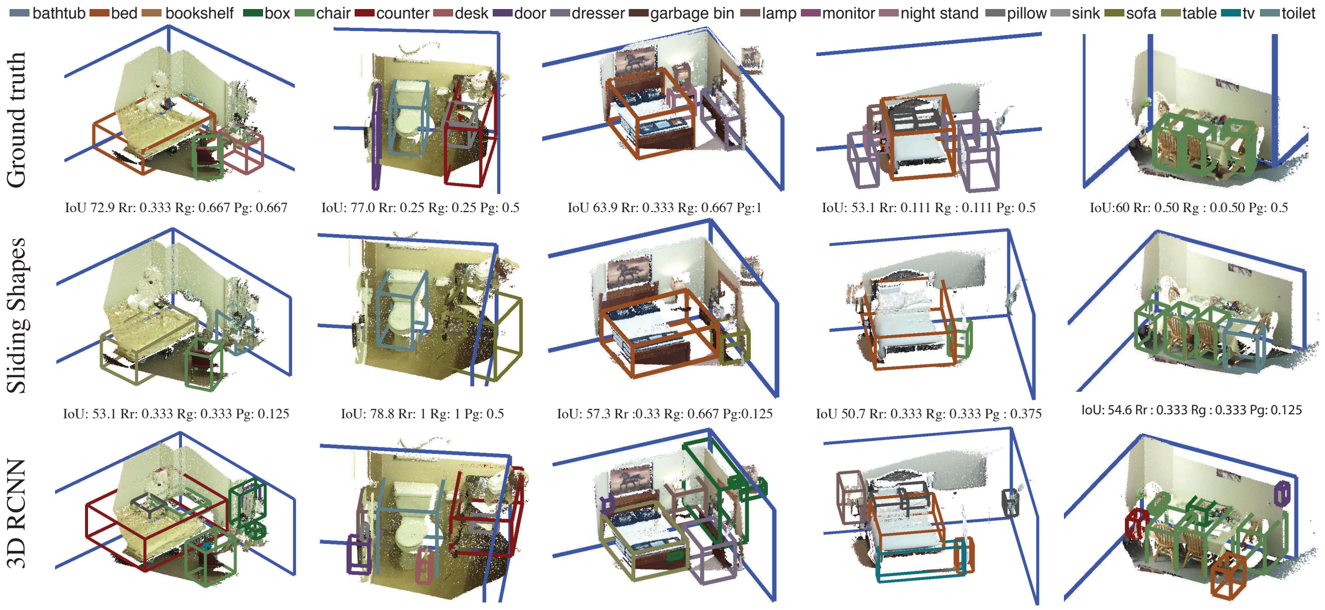

bathtub
bed
bookshelf
box
chair
counter
desk
door
dresser
garbage bin
lamp
monitor
night stand
pillow
sink
sofa
table
tv
toilet
Ground truth
Sliding Shapes
3D RCNN
IoU 72.9 Rr: 0.333 Rg: 0.667 Pg: 0.667
IoU: 77.0 Rr: 0.25 Rg: 0.25 Pg: 0.5
IoU 63.9 Rr: 0.333 Rg: 0.667 Pg:1
IoU: 53.1 Rr: 0.111 Rg : 0.111 Pg: 0.5
IoU:60 Rr: 0.50 Rg : 0.0.50 Pg: 0.5
IoU: 53.1 Rr: 0.333 Rg: 0.333 Pg: 0.125
IoU: 78.8 Rr: 1 Rg: 1 Pg: 0.5
IoU: 57.3 Rr :0.33 Rg: 0.667 Pg:0.125
IoU 50.7 Rr: 0.333 Rg: 0.333 Pg : 0.375
IoU: 54.6 Rr : 0.333 Rg : 0.333 Pg: 0.125

Figure 11. **Visualization of total scene understanding results.**

of the 3D points along the gravity direction, and computes the convex hull of the point projection onto the floor plane to estimate the walls. Our stronger baseline (named Manhattan Box) uses plane fitting to estimate a 3D rectangular room box. We first estimate the three principal directions of the point cloud based on the histogram of normal directions (see Section 2.5). We then segment the point cloud based on the normal orientation and look for the planes with furthest distance from center to form a box for the room layout. To compare with the color-based approach, we run Geometric Context [22] on the color image to estimate the room layout in 2D. We then use the camera tilt angle from gravity direction estimation and the focal length from the sensor to reconstruct the layout in 3D with single-view geometry, using the estimated floor height to scale the 3D layout properly. Figure 10 shows examples of the results of these algorithms. Average IoU for Geometric Context is 0.442, Convex Hull is 0.713, and Manhattan Box is 0.734 performs best.

Total Scene Understanding We use RGB-D RCNN and Sliding Shapes for object detection and combine them with Manhattan Box for room layout estimation. We do non-maximum suppression across object categories. For RGB-D RCNN, we estimate the 3D bounding boxes of objects from the 2D detection results. To get the 3D box we first project the points inside the 2D window to 3D. Along each major direction of the room we build a histogram of the point count. Starting from the median of the histogram, we set the box boundary at the first discontinuous location. We also set a threshold of detection confidence and maximum number of objects in a room to further reduce the number of detections. With the objects and room layout in hand we propose four simple ways to integrate them: (1) directly combines them; (2) remove the object detections that fall outside the estimated room layout; (3) adjust room to encompass 90 % the objects; (4) adjust the room according to majority of objects and remove the out-of-room objects. Figure 11 and Table 6 show the results.

	Train / Test	Kinect v2			Xtion			Percent drop (%)		
		rgb	d	rgbd	rgb	d	rgbd	rgb	d	rgbd
chair	Kinect v2	18.07	22.15	24.46	18.93	22.28	24.77	-4.76	-0.60	-1.28
	Xtion	12.28	16.80	15.31	15.86	13.71	23.76	29.22	-18.39	55.23
table	Kinect v2	15.45	30.54	29.53	16.34	8.74	18.69	-5.78	71.38	36.70
	Xtion	8.13	24.39	28.38	14.95	18.33	24.30	45.64	-33.05	-16.79

Table 7. **Cross-sensor bias.**

Cross sensor Because real data likely come from different sensors, it is important that an algorithm can generalize across them. Similar to dataset bias [68], we study sensor bias for different RGB-D sensors. We conduct an experiment to train a DPM object detector using data captured by one sensor and test on data captured by another to evaluate the cross-sensor generality. To separate out the dataset biases, we do this experiment on a subset of our data, where a Xtion and a Kinect v2 are mounted on a rig with large overlapping views of the same places. From the result in Table 7, we can see that sensor bias does exist. Both color and depth based algorithms exhibit some performance drop. We hope this benchmark can stimulate the development of RGB-D algorithms with better sensor generalization ability.

5. Conclusions

We introduce a RGB-D benchmark suite at PASCAL VOC scale with annotation in both 2D and 3D. We propose 3D metrics and evaluate algorithms for all major tasks towards total scene understanding. We hope that our benchmarks will enable significant progress for RGB-D scene understanding in the coming years.

Acknowledgement. This work is supported by gift funds from Intel Corporation. We thank Thomas Funkhouser, Jitendra Malik, Alexi A. Efros and Szymon Rusinkiewicz for valuable discussion. We also thank Linguang Zhang, Fisher Yu, Yinda Zhang, Luna Song, Zhirong Wu, Pingmei Xu, Guoxuan Zhang and others for data capturing and labeling.

References

[1] A. Aldoma, F. Tombari, L. Di Stefano, and M. Vincze. A global hypotheses verification method for 3d object recognition. In *ECCV*. 2012. 2

[2] A. Anand, H. S. Koppula, T. Joachims, and A. Saxena. Contextually guided semantic labeling and search for three-dimensional point clouds. *IJRR*, 2012. 2

[3] B. I. Barbosa, M. Cristani, A. Del Bue, L. Bazzani, and V. Murino. Re-identification with rgb-d sensors. In *First International Workshop on Re-Identification*, 2012. 2

[4] J. T. Barron and J. Malik. Intrinsic scene properties from a single rgb-d image. *CVPR*, 2013. 1

[5] J.-Y. Bouguet. Camera calibration toolbox for matlab. 2004. 3

[6] C. S. Catalin Ionescu, Fuxin Li. Latent structured models for human pose estimation. In *ICCV*, 2011. 2

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2

[8] N. Erdogmus and S. Marcel. Spoofing in 2d face recognition with 3d masks and anti-spoofing with kinect. In *BTAS*, 2013. 2

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 1, 2

[10] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Van Gool. Random forests for real time 3d face analysis. *IJCV*, 2013. 2

[11] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 2010. 7

[12] S. Fothergill, H. M. Mentis, P. Kohli, and S. Nowozin. Instructing people for training gestural interactive systems. In *CHI*, 2012. 2

[13] D. F. Fouhey, A. Gupta, and M. Hebert. Data-driven 3d primitives for single image understanding. In *ICCV*, 2013. 2

[14] D. F. Fouhey, A. Gupta, and M. Hebert. Unfolding an indoor origami world. In *ECCV*. 2014. 2

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1

[16] D. Gossow, D. Weikersdorfer, and M. Beetz. Distinctive texture features from perspective-invariant keypoints. In *ICPR*, 2012. 2

[17] R. Guo and D. Hoiem. Support surface prediction in indoor scenes. In *ICCV*, 2013. 1

[18] R. Guo and D. Hoiem. Support surface prediction in indoor scenes. In *ICCV*, 2013. 2

[19] S. Gupta, P. Arbelaez, and J. Malik. Perceptual organization and recognition of indoor scenes from RGB-D images. In *CVPR*, 2013. 1, 2

[20] S. Gupta, R. Girshick, P. Arbelaez, and J. Malik. Learning rich features from RGB-D images for object detection and segmentation. In *ECCV*, 2014. 1, 2, 7

[21] A. Handa, T. Whelan, J. McDonald, and A. Davison. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *ICRA*, 2014. 2

[22] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *ICCV*, 2009. 5, 8

[23] V. Hedau, D. Hoiem, and D. Forsyth. Thinking inside the box: Using appearance models and context based on room geometry. In *ECCV*. 2010. 5

[24] V. Hedau, D. Hoiem, and D. Forsyth. Recovering free space of indoor scenes from a single image. In *CVPR*, 2012. 5

[25] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *ACCV*. 2013. 2

[26] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *PAMI*, 2014. 2

[27] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In *UIST*, 2011. 1

[28] 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*, 2011. 2, 4

[29] A. Janoch, S. Karayev, Y. Jia, J. T. Barron, M. Fritz, K. Saenko, and T. Darrell. A category-level 3d object dataset: Putting the kinect to work. 2013. 2

[30] Z. Jia, A. Gallagher, A. Saxena, and T. Chen. 3d-based reasoning with blocks, support, and stability. In *CVPR*, 2013. 1

[31] H. Jiang. Finding approximate convex shapes in rgbd images. In *ECCV*. 2014. 2

[32] H. Jiang and J. Xiao. A linear approach to matching cuboids in RGBD images. In *CVPR*, 2013. 1, 2

[33] M. Kepski and B. Kwolek. Fall detection using ceiling-mounted 3d depth camera. 2

[34] H. S. Koppula, A. Anand, T. Joachims, and A. Saxena. Semantic labeling of 3d point clouds for indoor scenes. In *NIPS*, 2011. 2

[35] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from rgb-d videos. *IJRR*, 2013. 2

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 7

[37] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *ICRA*, 2011. 2

[38] D. Lin, S. Fidler, and R. Urtasun. Holistic scene understanding for 3d object detection with RGBD cameras. In *ICCV*, 2013. 1, 2, 6

[39] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing: Label transfer via dense scene alignment. In *CVPR*, 2009. 7

[40] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *PAMI*, 2011. 6, 7

[41] L. Liu and L. Shao. Learning discriminative representations from rgb-d video data. In *IJCAI*, 2013. 2

[42] M. Luber, L. Spinello, and K. O. Arras. People tracking in rgb-d data with on-line boosted target models. In *IROS*, 2011. 2

[43] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011. 7

[44] J. Mason, B. Marthi, and R. Parr. Object disappearance for object discovery. In *IROS*, 2012. 2

[45] O. Mattausch, D. Panozzo, C. Mura, O. Sorkine-Hornung, and R. Pajarola. Object detection and classification from large-scale cluttered indoor scans. In *Computer Graphics Forum*, 2014. 2

[46] S. Meister, S. Izadi, P. Kohli, M. Hämmerle, C. Rother, and D. Kondermann. When can we use kinectfusion for ground truth acquisition? In *Proc. Workshop on Color-Depth Camera Fusion in Robotics*, 2012. 2

[47] A. Mian, M. Bennamoun, and R. Owens. On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. *IJCV*, 2010. 2

[48] R. Min, N. Kose, and J.-L. Dugelay. Kinectfacedb: A kinect database for face recognition. 2

[49] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 1, 2, 4, 7

[50] B. Ni, G. Wang, and P. Moulin. Rgbd-hudaact: A color-depth video database for human daily activity recognition. 2011. 2

[51] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 2001. 5, 7

[52] F. Pomerleau, S. Magnenat, F. Colas, M. Liu, and R. Siegwart. Tracking a depth camera: Parameter exploration for fast icp. In *IROS*, 2011. 2

[53] X. Ren, L. Bo, and D. Fox. Rgb-(d) scene labeling: Features and algorithms. In *CVPR*, 2012. 1, 2, 6, 7

[54] A. Richtsfeld, T. Morwald, J. Prankl, M. Zillich, and M. Vincze. Segmentation of unknown objects in indoor environments. In *IROS*, 2012. 2

[55] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *IJCV*, 2008. 4

[56] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, et al. Efficient human pose estimation from single depth images. *PAMI*, 2013. 1

[57] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *CVPR*, 2013. 2

[58] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 2013. 1

[59] A. Shrivastava and A. Gupta. Building part-based object detectors via 3d geometry. In *ICCV*, 2013. 2

[60] N. Silberman and R. Fergus. Indoor scene segmentation using a structured light sensor. In *Proceedings of the International Conference on Computer Vision - Workshop on 3D Representation and Recognition*, 2011. 2

[61] A. Singh, J. Sha, K. S. Narayan, T. Achim, and P. Abbeel. Bigbird: A large-scale 3d database of object instances. In *ICRA*, 2014. 2

[62] S. Song and J. Xiao. Sliding Shapes for 3D object detection in RGB-D images. In *ECCV*, 2014. 2, 6, 7

[63] L. Spinello and K. O. Arras. People detection in rgb-d data. In *IROS*, 2011. 2

[64] S. Stein and S. J. McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, 2013. 2

[65] S. Stein and S. J. McKenna. User-adaptive models for recognizing food preparation activities. 2013. 2

[66] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *IROS*, 2012. 2

[67] J. Sung, C. Ponce, B. Selman, and A. Saxena. Human activity detection from rgbd images. *Plan, Activity, and Intent Recognition*, 2011. 2

[68] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011. 8

[69] Z. Wu, S. Song, A. Khosla, X. Tang, and J. Xiao. 3D ShapeNets: A deep representation for volumetric shape modeling. *CVPR*, 2015. 2

[70] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 5, 7

[71] J. Xiao, J. Hays, B. C. Russell, G. Patterson, K. Ehinger, A. Torralba, and A. Oliva. Basic level scene understanding: Categories, attributes and structures. *Frontiers in Psychology*, 4(506), 2013. 6

[72] J. Xiao, A. Owens, and A. Torralba. SUN3D: A database of big spaces reconstructed using SfM and object labels. In *ICCV*, 2013. 1, 2, 4

[73] B. Zeisl, K. Koser, and M. Pollefeys. Automatic registration of rgb-d scans via salient directions. In *ICCV*, 2013. 2

[74] J. Zhang, C. Kan, A. G. Schwing, and R. Urtasun. Estimating the 3d layout of indoor scenes and its clutter from depth sensors. In *ICCV*, 2013. 2, 5, 7

[75] B. Zhou, J. Xiao, A. Lapedriza, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014. 5, 7