Dense Object Detection in Indoor Scenes Using Depth Information

Jinchao Ye
Stanford University
Stanford CA 94305
jcye@stanford.edu

Second Author
Institution2
First line of institution2 address
http://www.author.org/~second

Abstract

RGB-D sensors, such as Microsoft Kinect sensor, can provide us depth information which might be useful to do dense object detection. Moreover, internet images, such as images from ImageNet, have millions photos for various kinds of objects. Internet images are useful for training while the depth image can provide valuable information for detection. Using models trained from the internet images and a depth image acquired by Kinect, we want to do dense object detection on indoor scenes.

1. Introduction

1.1. Dense Object Detection

Object detection is a fundamental problem in computer vision. Recent years, Bag of Words Model and Part Based Model [1] have showed significant improvement in object classification or detection. However, these methods can only deal well with the scenario when there is only one kind or only a small number of kinds of objects in the image.

In natural indoor images, there are various kinds of objects, such as monitor, desk, keyboard, chair, books etc. Our goal is to detect almost every object in indoor images. The ideal result is that for every pixel in the foreground in the image, it belongs to a bounding box which is detected and labeled by our algorithm. The background means floor, wall and ceiling, which we usually don't care about. The ideal result is shown in figure 1. This is a very difficult task due to occlusion, low resolution, intra-class variance, etc.

1.2. Images from ImageNet

ImageNet [5] is an image database organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images. This is an excellent image dataset for training.

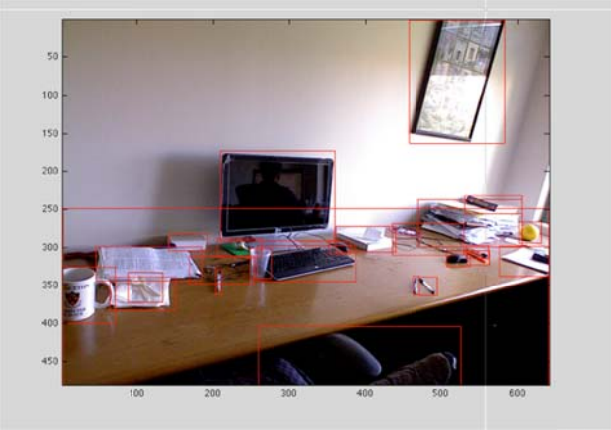

Figure 1 Ideal Result of Dense Object Detection

1.3. Depth Images from Kinect

Kinect sensor uses stereo techniques and can provide a depth image of the same scene. The effective depth sensor range of Kinect is between 0.8m and 3.5m and some pixels lack depth information. We need to do interpolation to get the useful depth image.

1.4. Problem Statement

Our goal is to do dense object detection on indoor images. Moreover, during the training process, we use only the ordinary 2D images from ImageNet. During detecting process, apart from the test image, we also have an additional depth image to help improve the detection performance.

1.5. Dataset

During training process, for each kind of object, we use the images from the corresponding subset in ImageNet. There are about 500 hundreds images per node in ImageNet so we can use the 200 hundreds images to train a classifier for each object.

For testing, we have collected 200 indoor images along with corresponding depth images. I have also labeled all the ground truth bounding boxes. There are 146 different kinds of objects as well as some undefined objects in the testing

images. Some objects appear a lot of times in the testing images, such as monitor, microwave, chair, desk, sink, faucet, etc. Some objects such as tennis appear only once or twice. There are also some kinds of objects which do not have corresponding subsets in ImageNet, such as chopping board. We neglect such kinds of objects during training and testing.

1.6. Expected Result and Evaluation

We expect our algorithm can achieve good results on the testing images. The ideal result is shown as in figure 1. Our algorithm might not detect every object but it should detect as many objects as possible. For each kinds of object, we can get a detection rate and false alarm rate. We can also calculate the overall detection rate and false alarm rate. We will run part based models [1] on our dataset and compare the result with the result of our own algorithm.

2. Related Work

2.1. Part Based Models

Deformable Part Based Models [1] is the state-of-art technique for detecting single object. Our approach is largely based on this work. This technique does not need depth information either in training process or detecting process. This technique is built on pictorial structures framework, which represent objects by a collection of parts and the spring-like connections between certain pairs of parts. There is a root filter for the whole object and a part filter for each part of the model. The score of detection is the sum of filter scores plus deformation scores. It also train multiple models for a kind of object according to different aspect ratio

While Deformable Part Based Models is the state-of-art technique of detection and perform quite well on VOC dataset. However, it only takes advantage of 2D images which limit the performance, especially on real-world image datasets. Our approach adds depth information during testing process and gets better result.

2.2. Part Based Models

Koppula et al. have tried detecting indoor objects using 3D information [2]. They over-segment the input image and label each object as bed, table top, wall, etc. Their approach is based on a Markov Random Field model and explicitly models the geometric configuration between different objects. For example, a monitor is always on-top-of a table and chairs are usually near a table. They applied this algorithm on a mobile robot for the tasking of finding an object and got quite good results.

However, unlike our approach, their training process is based on 3D images (images with 3D information) while we use web images to train classifiers. It is obviously that web images are more easily obtained and there are almost infinite categories of images on the internet.

2.3. Coherent Object Detection

Bao et al. [3] use support planes for coherent object detection and scene understanding. Their intuition is that many objects all usually on a plane. For example, mugs are usually on a table, which is a plane. In this way, they can rule out the false candidates which are not on the supporting plane or supporting planes. Once they get the planes, they further assume that the planes are horizontal and get the camera parameters.

This method does not need 3D information as input, and has got quite good performance on certain dataset. However, when detecting many categories of objects, many objects are not on horizontal a few numbers of supporting planes. For example, pictures hanging on the wall.

3. Approach

3.1. Baseline Approach Using Part Based Models

We detect each object independently using part based models and latent SVM, just as Felzenszwalb did. We briefly describe the algorithm in the following.

It first computes HOG feature pyramid. A filter is a rectangular template defining weight for features and the response of a filter F at a position (x, y) in a feature map G is defined as the dot product,

$$\sum_{x',y'} F(x',y') \cdot G(x+x', y+y').$$

The score of a candidate (or hypothesis) is the sum of the scores of each part filter at its own position and scale minus the deformation cost,

$$score(p_0, p_1, \dots, p_n) = \sum_{i=0}^{n} F_i' \cdot \phi(H, p_i) - \sum_{i=1}^{n} d_i \cdot \phi_d\,(dx_i, dy_i) + b$$

The first term is the data term (the response of part filters). The second term is the deformation cost which punishes the displacement of each part relative to its anchor position. The bias term is introduced to make scores comparable between different models. Here p_0 is the root filter.

For matching, we pick the candidates which maximize the score. p_0 is the root filter here.

$$score(p_0) = \max_{p_1, p_2, \dots p_n} score(p_0, p_1, \dots, p_n)$$

As for latent SVM, we need to maximize the function,

$$f_\beta(x) = \max_{p_1, \dots p_n} \beta \cdot \Phi(x, z).$$

Here z is a latent variable and for part based models, the latent variable is parts. The maximization problem can be transformed into a semi-convex optimization problem and therefore can be solved.

The model of a monitor is shown as figure 2,

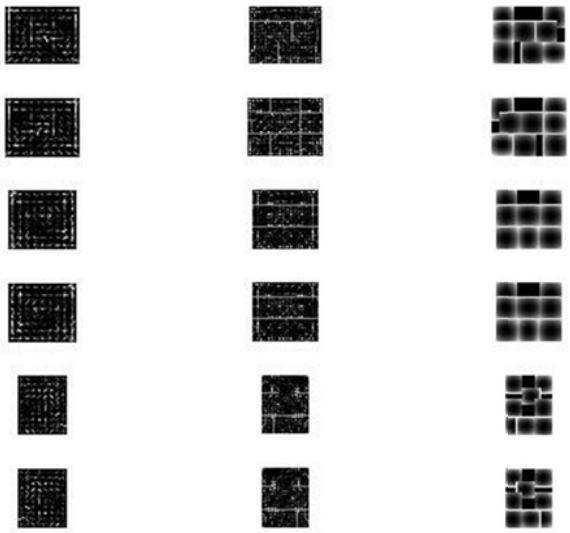

Figure 2 6 Component Part Based Model—Monitor

The picture shows the model of monitor, a 6 component part based model.

3.2. Posterior Handling using Depth Image

Because we do not have depth images during training process, we cannot incorporate the depth information during training. However, we can use depth images to remove the false alarms by a certain detection method.

The depth of objects from camera is continuous. Moreover, When we look at indoor objects, such as monitors, microwaves, keyboards, books, mouses or even mugs, we can find that many objects are planar (monitors, microwaves) or within a small volume (mouses, mugs). We can therefore assume that the depth of an indoor objects changes linearly with the coordinates of the other two dimensions. Denote the depth by d, then

$$\mathrm{d}(\mathrm{x},\mathrm{y}) = Ax + By + C.$$

If a candidate contains a large depth gap, then the bounding box must be removed, just as the green boxes in figure 3 shows.

So, for each candidate bounding boxes using part based models, we first do non-maximum suppression. Then we compute a plane for each of the remaining bounding boxes. It is a linear regression problem. In order to use data more efficiently, when computing the plane, we first narrow the bounding boxes because the boundaries of bounding boxes are likely to be part of other objects and there might be a depth gap on the boundary. Secondly, the depth information we got from Kinect sensor are cluttered because there are many pixels which do not have a depth value. So we need to rule these pixels out when computing the plane of the bounding box. At last, the linear regression problem can be performed by the minimization problem,

Figure 3 remove false positives using depth information

$$[A,B,C] = \min_{A,B,C} \sum_{x,y} \big(d(x,y) - (Ax + By + C)\big)^2$$

Here, (x, y) must be a valid pixel in the bounding box.

Once we got the plane for a bounding box, we compute the outliers of the plane. By "outlier", we mean the point of which depth deviates from Ax+By+C more than a certain threshold. We calculate the number of such outliers, and if the number is larger than a threshold, we remove this bounding box.

4. Experiment

4.1. Dataset

As said before, the training dataset is from Image-Net. For each category (each node in Image-Net), we used about 200 hundreds images for training.

As for the testing data, we collect 200 indoor images along with its depth information using Kinect sensors. I labeled the data set manually, trying hard to satisfying that any foreground pixel belongs to a bounding box. One testing image with ground truth is shown as figure 1. The ground truth is stored using VOC2007 format.

There are 146 kinds of known objects in the testing dataset. Each category corresponds to a certain node in Image-Net. There are also some undefined objects in the testing data which cannot find corresponding nodes in the Image-Net, such as chopping board.

Some popular categories are shown as table 1,

Table 1 Popular Categories in the testing dataset

Object Category	Number of Times	Corresponding Node
Monitor	103	n03085219
Microwave	31	n03761084

keyboard	71	n03085013
mug	97	n03797390
Chair	151	n04373704

4.2. Result Using Baseline Approach

We train and test on three popular kinds of categories, Monitor, Microwave and Chair. During training process, we train a 6-componenet model for each of the category. During testing process, we test each category on each testing image independently.

Some of the testing results are as following, the green bounding boxes are monitors, the red bounding boxes are chairs, the blue bounding boxes are microwaves.

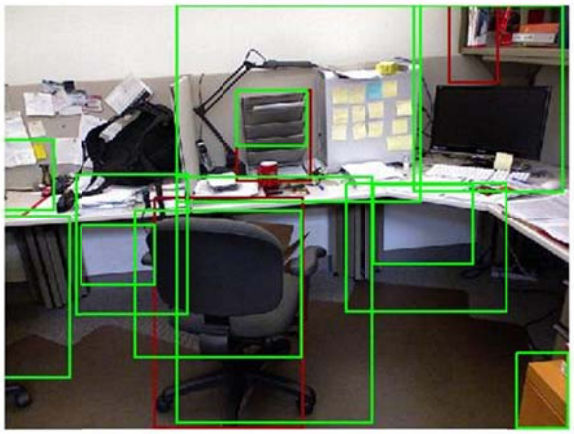

Figure 4 Baseline Result on Image 1

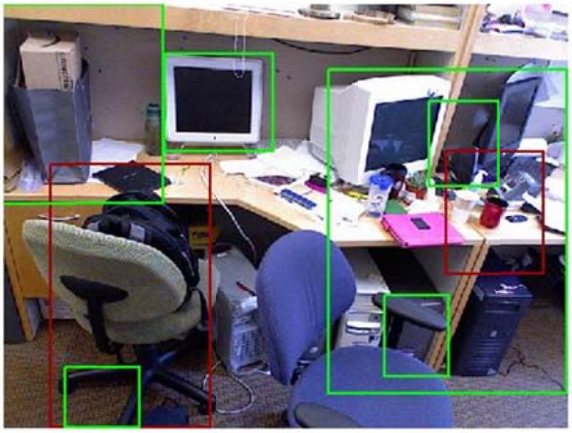

Figure 5 Baseline Result on Image 2

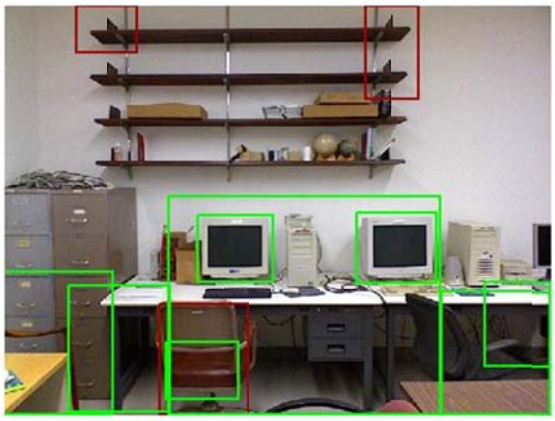

Figure 6 Baseline Result on Image 3

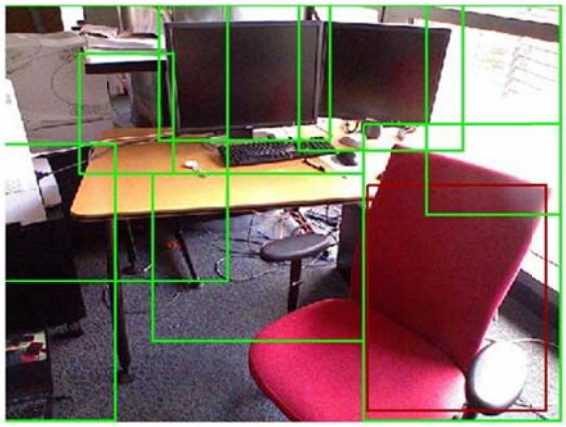

Figure 7 Baseline Result on Image 4

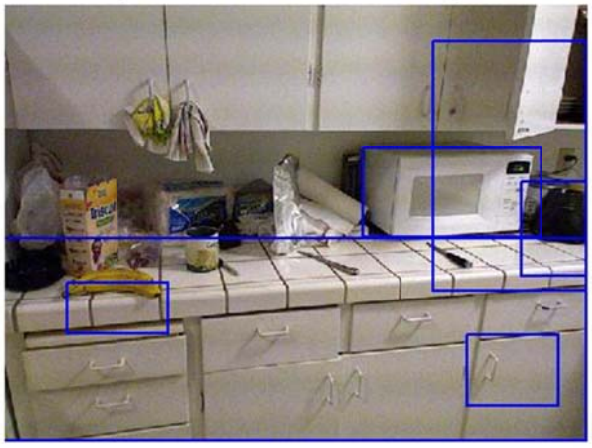

Figure 8 Baseline Result on Image 5

Figure 9 Baseline Reulst on Image 6

Apparently there are two many false alarms in these images, especially for the category of monitors.

4.3. Result Using Depth Information

We use depth information to remove some false alarms of the result of part based models. We test each category on each testing image independently. This method improves the performance.

Some of the testing results are as following, the green bounding boxes are monitors, the red bounding boxes are chairs and the blue bounding boxes are microwaves.

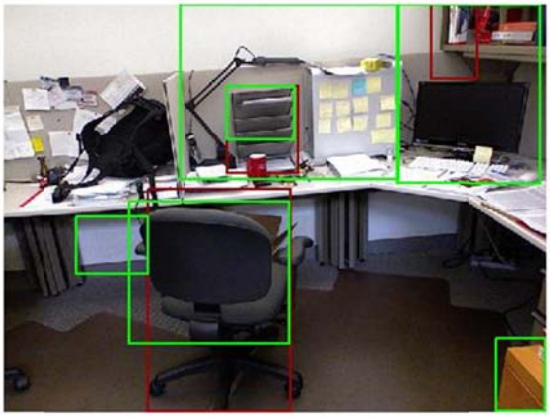

Figure 10 Result using depth on Image 1

Figure 11 Result using depth on Image 2

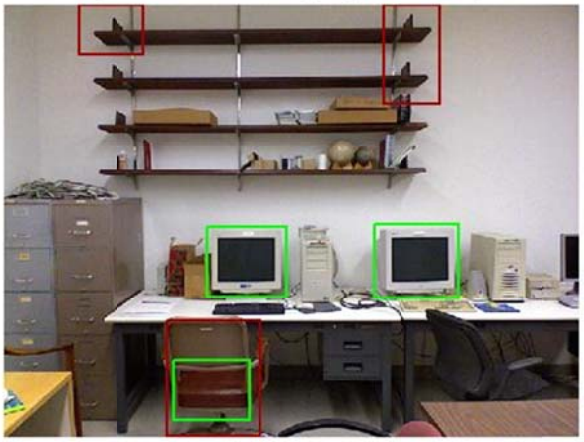

Figure 12 Result using depth on Image 3

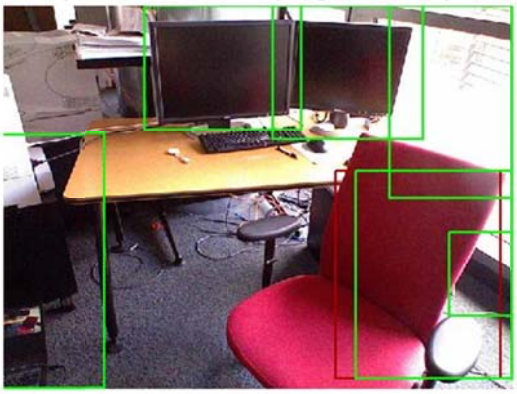

Figure 13 Result using depth on Image 4

Figure 14 Result using depth on Image 5

Figure 15 Result using depth on Image 6

Comparing these images with results using baseline approach, we can conclude that our approach acquired better results using depth information, despite that there are still some false alarms in the results.

4.4. Precision Recall Curve

Because we detect each category independently, therefore we can compute a precision recall curve for each category independently.

The curves are shown below, the red line is the baseline result and the blue line is the result using depth information. This further confirms that our approach using depth information outperforms using part-based models alone.

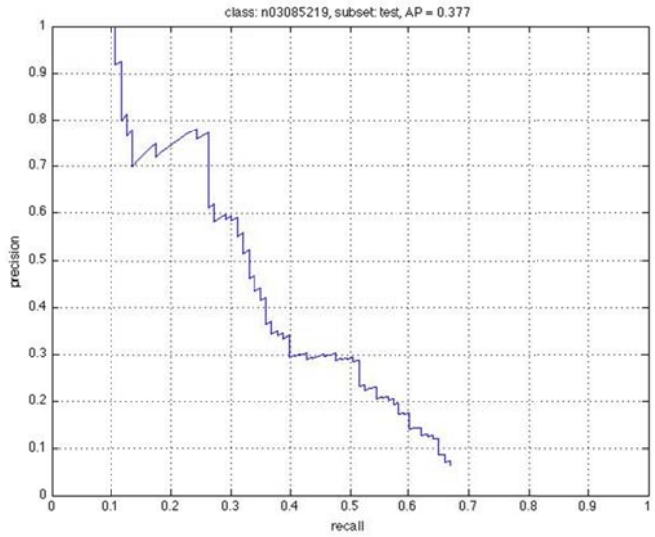

Figure 16 Precision_Recall_Monitor_Baseline

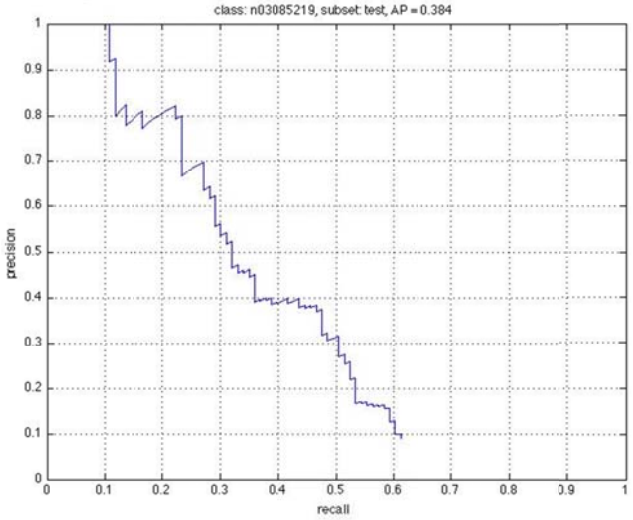

Figure 17 Precision_Recall_Monitor_Depth

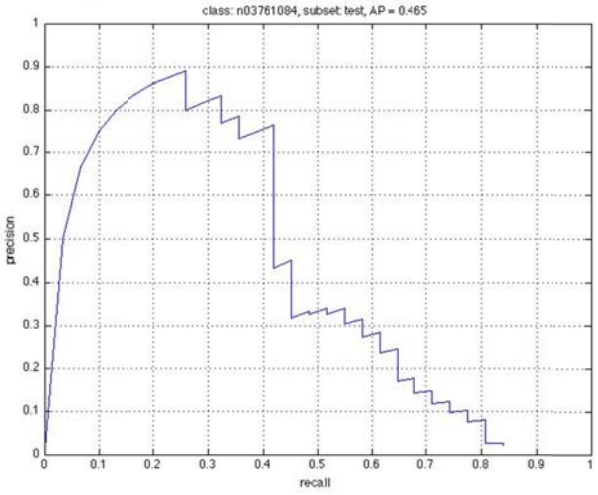

Figure 18 Precision_Recall_Microwave_Baseline

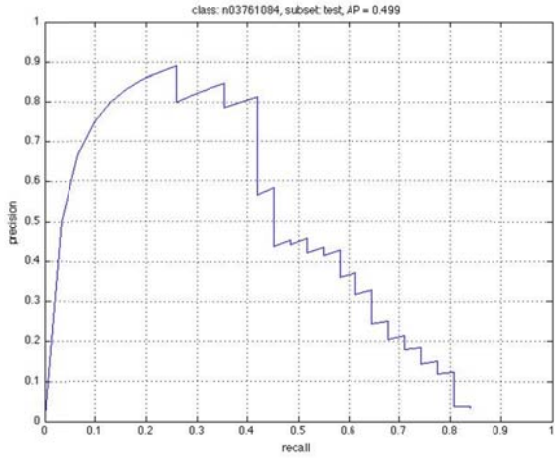

Figure 19 Precision_Recall_Microwave_Depth

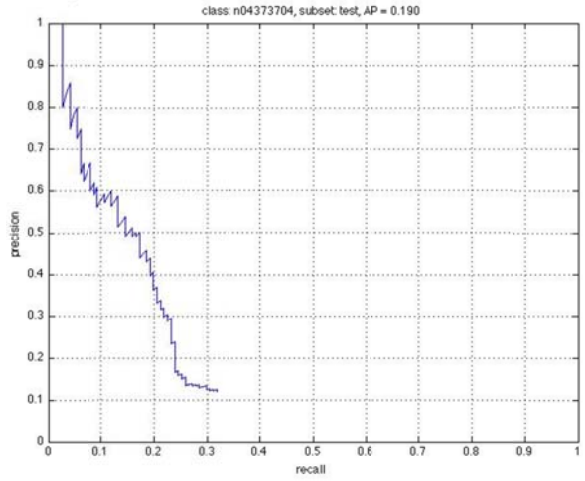

Figure 20 Precision_Recall_Baseline_Chair

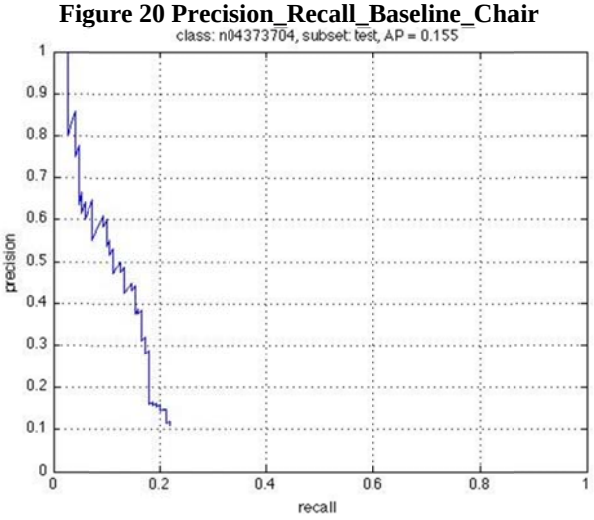

Figure 21 Precision_Recall_Depth_Chair

For the first two categories, monitor and microwave, the average precision score is higher if we use depth information instead of using part-based model alone. Despite that the recall might decrease a little if we remove false alarms using depth information, the precisions are higher and therefore the average precisions are also higher. Therefore we can see that result using depth information generally do a little better than result without using depth information. As for the chairs, the performances of both methods are quite bad. And the performance of our approach is even worse than baseline. It might be because chair cannot be assumed as a plane. We think the overall bad performance for chair is due to that the variance in shape and scales between training images and testing images is really large.

5. Conclusion

5.1. Summary

The most interesting part of this project is it incorporates recognition problem and a little 3D knowledge. We don't use 3D information during training and we use the depth information only in the testing process. Since we don't need 3D input during training, this makes the training images available almost everywhere. In fact, another interesting part of this project is that it uses the web images as training images. Specifically, we use pictures from Image-Net.

During the project, I have learned a lot about Part-Based Models and latent SVM. The Part-Based Model is an elegant framework for object detection. Unlike Bag of Words techniques, it provides information of the geometric configuration of between certain parts of a certain object and provides more insight for more advanced tasks such as action detection.

I have also learned a little about how to incorporate depth information into recognition problem. Despite that I have not proposed a model which incorporates depth information during training, the posterior handling using depth images still needs some insight and is a little useful.

This project also needs patience as it is a time-consuming thing to label testing data as well as running part based models.

5.2. Future Work

There is still a lot to be done. Due to lack of time, I am not able to propose a nice model as well as implement it. In the future, there are several things I want to try.

Firstly, detect objects jointly. We can model this dense object detection problem using a conditional Markov random field. In this way, we can jointly detect the multiple objects and this might will improve the performance.

Secondly, incorporate depth information during training. In this project, we have not used depth information during training and therefore we can hardly learn anything for the 3rd dimension during training process. Moreover, the training images from the web and the testing data are quite different. In fact we do not need depth images for every

image during training, we only need depth images for a small fraction of images. If we put some labeled testing data during training process and test on the rest, this will solve the problem and also make the training data more "familiar" with the testing data. This would definitely improve the performance.

6. References

[1] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan. Object Detection with Discriminatively Trained Part Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32, No. 9, September 2010.

[2] Hema Swetha Koppula, Abhishek Anand, Thorsten Joachims, Ashutosh Saxena. Labeling 3D scenes for Personal Assistant Robots. In *RSS workshop on RGB-D cameras*, 2011

[3] Yingze Bao, Min Sun, Silvio Savarese. Toward Coherent Object Detection And Scene Layout Understanding. In CVPR, 2010.

[4] Krystian Mikolajczyk, Bastian Leibe, Bernt Schiele. Multiple Object Class Detection with a Generative Model. In CVPR, 2006.

[5] http://www.image-net.org

7. Appendix

My course project is part of a larger project in vision lab. Thanks for the instruction of Prof. Fei-fei Li and Ph.D. Jia Deng.

8. Supplementary Materials

My code is attached in the email as Ye_Code.zip. I also use P. Felzenszwalb's code for part based models. The link for this code is
http://www.cs.brown.edu/~pff/latent/voc-release4.01.tgz.