Object Detection Using Cascaded Classification Models

Chau Nguyen

December 17, 2010

Abstract

Building a robust object detector for robots is alway essential for many robot operations. The traditional problem of object detection and recognition has been tackled by computer vision researchers for years [4, 5, 6, 7, 8, 13, 15]. There are many different approaches ranging from location, geometric context, cultural context [6, 8] to functionalities and categories of the objects. Most of these methods were developed based on very sophisticated computer vision algorithm and machine learning techniques with a single source of visualization - camera images. The other source of visualization such as 3D sensor has not been well taken into account. The recent technique of Cascaded Classification Models is applied for less sophisticated machine learning models and some of them gives good results [20]. This projects employs Cascaded Classification Models (CCM) to build a robust object detector given two main sources of visualization usually available on a robot: camera images and 3D point cloud data from a depth sensor.

1 Introduction

Object detection and recognititon problem is usually classified as the problem of computer vision rather than robotics. The traditional approach is to build a template of the object that needs to be detected and use the sliding window technique with or without scaling to search for the object in the given images. In recent years, the field of robotics has proven the possibility of having certain autonomous systems - robots - to perform specific operations. In order to perform operations on objects, the robot first must locate and recongnize objects in unconstraint environments. Object detection becomes increasingly important for robotics. Object detector for robots emphasizes on robustness. Computer vision researchers developed many robust object detectors by taking into account the context, especially geometric context of the object appearance. Although a substaintial number of computer vision algorithms can estimate 3 dimensional world with a single image [9, 11, 12], not many of these algorithms demonstrate to work well with both indoor and outdoor environments. On the other hand, most robots are equiped with cameras and 3D depth sensors. 3D depth sensors information can provide geometric context of the object appearance without the need to implement very sophisticated algorithms and can work well with both indoor and outdoor environments. Unfortunately, 3D sensor data is mostly ignored by many object detection algorithms. Obtaining both 3D data and camera images to build Cascaded Classification Models (CCM) object detector promises better results.

2 Related Works

Context based object detection is not a new concept in computer vision. An empirical study of context in object detection by Divvala et al. [6] has proven that context understanding is a critical part of object detection problem. The authors give many sources of context for object detection such as 3D geometric context, weather context, geographic context, and cultural context. Among these type of contexts, 3D geometric context attracts most of researchers because it is easier for computer to recognize with a single image than the other types of contextual information. 3D geometric context estimator segments images into geometrically consistent regions where the world could be simplified as the set of planar surfaces, for instance, the ground is consider as support (horizontal) surfaces and buildings are vertical surfaces, etc... Hoeim et al. great work in estimating surface layout [11, 12] bases on the theory of James Gibson (Perception of the Visual World, Gibson, 1950) that *"The elementary impressions of visual world are those of surface and edge."* The authors make an important claim that is applied for this project: an object tends to correspond to a certain type of surface such as the road corresponds to a supporting surface and a pedestrian corresponds to a vertical, non-planar solid surface. Their work demonstrates the application of 3D surfaces layout in object detector performs much better than purely local detector. Malisiewicz and Efros article on improving spatial support of objects [9] once again reinforces the relations between objects and the 3D surfaces from the environments containing them. Many works from Gould et al [5, 14] segment images into geometric and semantically consistent regions to build a region-based object detector.

From the viewpoint of robotics, object detection involves multiple sources of visualization rather than a single camera since most robots are equipped with one or more cameras and at least one depth sensor. Given those available resources, Gould et al. [7] propose taking into account range data (or 3D data from sensor) to build the object detector on a robotic platform. The method combines 2D image and 3D sensor modalities to enhance the object detection in cluttered real-world environment. Their work relies on a camera, and both low and high dimensional 3D laser sensors which can provide them very accurate 3D information. The object detector's feature vector comprises the image features and 3D features of the object which gives it the distinct advantage of 3D information. Another object detection work in robotic platform by Coates and Ng [13] shows that their probabilitic method for combining multiple camera views can significantly improve the acculary of the object detector.

3 Approaches

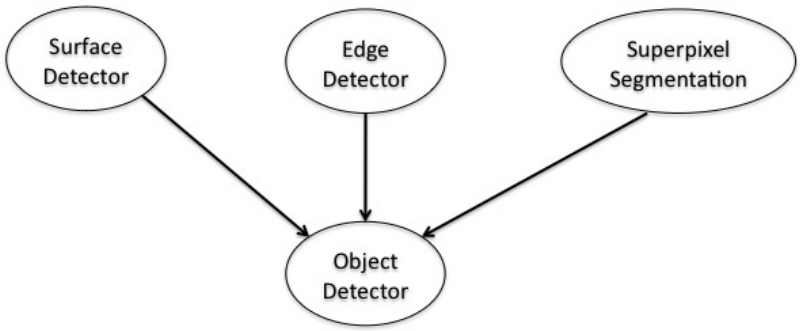
When looking for objects, people tend to look for objects in specific places. Some objects are more likely to appear on certain types of surfaces than the others. For example, monitors, keyboards, and mouse are more likely to appear on the tables whereas clocks are more likely to be on the walls and shoes are more likely to be on the ground. Tables and ground are both horizontal surfaces and walls are vertical surfaces. This project takes the approach of geometric context based object detector, especially the correspondence of objects and surfaces, on robotic platform that segments the images into geometrically consistent regions. We use 3D stereo sensor data to segment environment into regions where each region belongs to a planar surface, then align them with images obtained from camera. The relative locations of objects and horizontal or vertical surfaces are expected to be consistent across many different images.

4 Hardware Specifications

Data obtained for this project uses Miscrosoft Kinect sensor. The device produces aprroximately 240,000 points in one single scan which is extracted from depth depth sensor. Kinect has 57 degree horizontal field of view and 43 degree vertical field of view. Kinect depth sensor ranges from 1.2 meters to 3.5 meters. Kinect camera produces about 30 frames of 640 by 480 per second.
Kinect was released by Microsoft in early November 2010 and is demonstrated to effectively and accurately track human motion. Kinect is chosen to be the main sensor and camera because it produces much denser depth infomation than the available sensors in the lab do. It is easier for one to obtain a Kinect than look for any kind of stereo sensors and cheaper than laser sensors. Another advantage of Kinect is that it has both camera and depth sensor in a very close distance, approximately 4 cm, which make calibration between these two devices easier than separate camera and depth sensor.

5 Cascaded Classification Models (CCM) Object Detector

Although surface detector segments 3D point cloud into planar surfaces, it will not be able to segment regions that have similar planar surface type but different textures. For example, wall and black board both appear as vertical surface but they have different colors as well as textures. Edge detector draws the distinction between regions that are geometrically similar but texturally different. Superpixel segmentation gives different colors to these surfaces. So, three classification models: surface detector, edge detector, and superpixel segmentation are used as first layer of the CCM model. The combination of these 3 models is fed into an object detector.



5.1 Surface Detector on Point Cloud Data

Hybrid 3D surface detector segments 3D point cloud into clusters where each cluster seems to belong to the same planar surface. First, given a set of unorganized 3d points, the program puts

the points into 3D grid cells. Then, the program applies Orthogonal Distance Regression Plane with Single Value Decomposition method to points in a cell to find the best 3D plane equation $ax + by + cz + d = 0$ where (a, b, c) is the normalized normal vector and d is the offset for the plane of that cell.

Orthogonal Distance Regression Plane Method

1. Find the a, b, c, d that minimizes $J(a, b, c, d) = \min_{a,b,c,d}(\sum_{i=0}^{n} \frac{|ax_i+by_i+cz_i+d|^2}{a^2+b^2+c^2})$
2. Set partial derivative $\frac{\partial J(a,b,c,d)}{\partial d} = 0$ and solve for d, we get $d = -(ax_0 + by_0 + cz_0)$
3. Substitute d back into $J(a, b, c, d)$, we get new formula for J:
   $J(a, b, c) = \min_{a,b,c}(\sum_{i=0}^{n} \frac{|a(x_i-x_0)+b(y_i-y_0)+c(z_i-z_0)|^2}{a^2+b^2+c^2})$
4. Define $v^T = [a\ b\ c]$ and M be n×3 matrix where $M_{i1} = x_i - x_0$, $M_{i2} = y_i - y_0$, and $M_{i3} = z_i - z_0$ for $i \in \{1, 2, .., n\}$. We can rewrite $J(a, b, c)$ as:
   $J(v) = \frac{(v^T M^T)(Mv)}{v^T v} = \frac{v^T (M^T M)v}{v^T v}$.
5. Let $A = M^T M$, so, $J(v)$ is minimized by the eigenvector of A that corresponds to its smallest eigenvalues
6. Apply Single Value Decomposition computation on A and set (a, b, c) to the eigenvector that corresponds to the smallest eigenvalue of A, then normalize (a, b, c).

We choose the plane equation $ax + by + cz + d = 0$ because vertical planes with zero coefficent for z can still be represented using this equation. After 3D plane equations are computed, merging step gets executed. Two adjacent cells are merged into one plane if the angle between their normal vectors and the difference between their offsets are sufficiently small. Cells are continuously merged together until there does not exist any two adjacent cells that have similar plane equations. After the merging step is done, cells are grouped into segments. 3D plane equation of each segment is recomputed. Finally, these 3D plane equations are compared with equation of horizontal and vertical surfaces. If a normal vector makes an angle of approximately 0 degree with the normal vector $(0, 0, 1)$, the corresponding segment is considered as a horizontal surface. Similarly, if a normal vector makes an angle of approximately 90 degree with the normal vector $(0, 0, 1)$ then the corresponding segment is a vertical surface. Otherwise, the segment is neither vertical nor horizontal.

5.2 Edge Detector on camera image

We use MATLAB implementation of Canny method for edge detector [21] on Kinect camera images. Not much time is devoted to select the which method and threshold should be most appropriate, but Canny method with $threshold = 0.095$ and $sigma = 0.105$ seems to work fine.

5.3 Superpixel Segmentation on camera image

Superpixel Segmentation [1] helps eliminate edges from Edge Detector that are generated from the set of pixels that seems to belong to one region but different lighting condition makes their

color become different. We run Superpixel Segmentatiom model with $sigma = 0.8$, $k = 300$, and $min = 5$ on camera images.

5.4 Object Detector on the combined image

The object detector for this CCM model is based on the implementation of Simple Object Detector with Gentle Boosting with 120 week classifiers [2, 3, 4]. We used LabelMe to annotate images from Kinect camera and used those annotation on the combine images from 3 models (surface detector, edge detector, and superpixel segmentation). The features for training this CCM model is the template of the objects and the entire scene that contains the object including position of vertical and horizontal surfaces.

Gentle AdaBoost

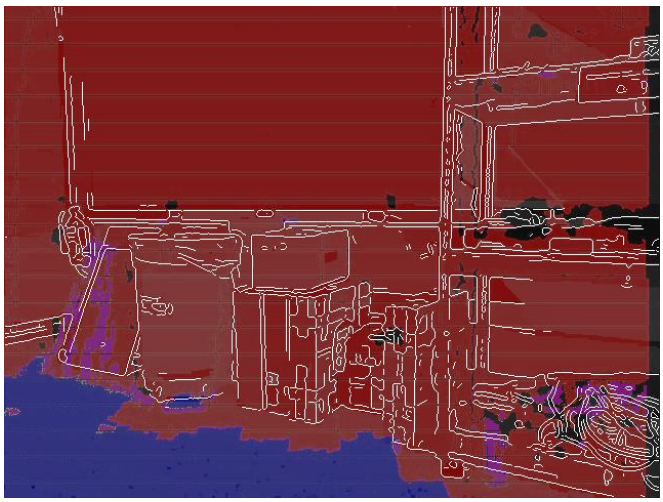
Training Data $(x_1, y_1), ..., (x_N, y_N)$ with x_i is a vector valued feature and $y_i \in \{-1, 1\}$ and M weak classifiers.

1. Start with weigths $w_i = 1/N, i = 1, 2, ..., N, F(x) = 0$.
2. Repeat for $m = 1, 2, ..., M$
   (a) Fit the regression function $f_m(x)$ by weighted least-squared of y_i to x_i with weights w_i
   (b) Update $F(x) \leftarrow F(x) + f_m(x)$
   (c) Update $w_i \leftarrow w_i e^{y_i f_m(x_i)}$
3. Output the classifier $sign[F(x)] = sign[\sum_{m=1}^{M} f_m(x)]$

The following image is an example of a camera image then the outputs from 3 classifiers of the first layer of CCM model.

The left most among 3 pictures below is computed using surface detector from 3D point cloud, then the point cloud segments are projected into the camera field of view. We use blue to label horizontal surfaces, red to label vertical surfaces, and purple to label other random surfaces that neiter vertical nor horizontal.. The middle picture is generated from edge detector, and the right most computed from superpixel segmentation.

And the type of the image below is the alignment of 3 classifiers that will be fed into the object detector both for training and testing purposes.



6 Experiments

Data needed for this project includes both real 3D point clouds and camera images. Due to insufficent time for data collection and labelling, the dataset only has a size of 100 images and the corresponding point clouds. About 80 images are use as training data and remaining images are used for testing. Training on such a small size dataset is subject to overfitting, thus, the goal of the experiment not to show how accurate our CCM object detector is but to compare our CCM model with the Non-CCM model is solely based on camera images and the object detector in the second layer of CCM [2]. Due small dataset, it is expected that precision and recall rate of both CCM and Non-CCM object detectors are very low. CCM and Non-CCM object detectors are trained and tested on 9 different objects: computer, monitor, keyboard, mouse, chair, bottle, cup, plate, and box in every scene. We use the statistic formula of F_1 *Score* to measure the test's accuracy: $F = 2 \times \frac{precision \times recall}{precision + recall}$. The average precision, recall, and F_1 *Score* accuracy for all objects is recorded in Table 1. The presicion rates of both models are very low because the size of objects varies across images and some objects appear less than 10 times in the entire data set which is very difficult for both models to generalize the templates of objects.

This experiment achieves the goal of showing that this CCM object detector performs better than the Non-CCM in precision and recall, F_1 Score Accuracy rate. However, it is not yet sufficient to prove a significant improvement of our CCM model because the precision and recall rate of our

CCM model is no where comparing to many state-of-the-art object detectors that rely on single image. In addition, training and testing on a small set of data do not give us accurate results since one misdetected or misclassified object is counted toward large percentage of precision and recall rate.

Table 1: CCM vs. Non-CCM Object Detector

	CCM Model	Non-CCM Model
Precision	4.5659%	1.8006%
Recall	43.6111%	26.7007%
F_1 Score Accuracy	8.2663%	3.3737%

7 Discussion and Future Works

One of our main future works for this project is the calibration of Kinect camera images and its depth information. Because Kinect was recently released for about one month, not many works has done to study the technical details of this device including the transformation from depth sensor coordinate system to camera coordinate system and conversion from depth information to 3D point cloud data. Since the Kinect depth sensor is steoreo sensor, its depth information is subject to noise. The next step after calibration is apply noise filtering on 3D point cloud data.
Rebuilding surface detector is the most important work needs to be done. The main goal of this project is build object detector with dependencies on planar surfaces. The current surface detector misclassifies overhangings as vertical surfaces.
To achieve better, recall, and accuracy rate, a clear probabilistic graphical model is needed to define the dependencies between object appearance and its location relative to the location of major vertical and horizontal surfaces of the environment.
Data collection and labelling process is time-consuming and require much effort, especially with this project. A data set of around 2000 images will perform much better than the current data set of 100 images. This is essential to verify the improvement of this method comparing other methods. Although computer generated images can be easily obtained, it is impossible to have compute generated 3D point clouds. Thus, real 3D data and images play a vital role in this project.

8 Conclusion

Due to time constraint, many works has not been done, thus, the experiment might not provide a convincing results. However, it is true that our CCM Object Detector performs better than Non-CCM without the implementation of very sophisticated machine learning models. Our CCM model promises a more robust object detector for robots.

9 References

[1] Pedro F. Felzenszwalb and Daniel P. Huttenlocher, Efficient Graph-Based Image Segmentation, International Journal of Computer Vision, 59(2) September 2004.

[2] Simple Object Detector with Boosting, ICCV 2005 short course on Recognizing and Learning Object Categories.
[3] Friedman, J. H., Hastie, T. and Tibshirani, R., Additive Logistic Regression: a Statistical View of Boosting. (Aug. 1998).
[4] A. Torralba, K. P. Murphy and W. T. Freeman. (2004). Sharing features: efficient boosting procedures for multiclass object detection. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp 762- 769.
[5] Stephen Gould, Tianshi Gao, Daphne Koller, Region-based Segmentation and Object Detection.
[6] Santosh K. Divvala, Derek Hoiem, James H. Hays, Alexei A. Efros, Martial Hebert, An Empirical Study of Context in Object Detection.
[7] Stephen Gould, Paul Baumstarck, Morgan Quigley, Andrew Y. Ng, Daphne Koller, Integrating Visual and Range Data for Robotic Object Detection.
[8] Antonio Toralba, Kevin P. Murphy, William T. Freeman, Mark A Rubin, Context-based vision system for place and object recognition.
[9] Tomasz Melisiewicz and Alexei A. Efros, Improving Spatial Support for Objects via Multiple Segmentations.
[10] Geremy Heitz, Daphne Koller, Learning Spatial Context: Using Stuff to Find Things.
[11] Derek Hoiem, Alexei A. Efros, Martial Hebert, Recovering Surface Layout from an Image, International Journal of Computer Vision 75(1), 151-172, 2007.
[12] Derek Hoiem, Alexei A. Efros, Martial Hebert, Closing the Loop in Scene Interpretation.
[13] Adam Coates, Andrew Y. Ng, Multi-Camera Object Detection for Robotics.
[14] Stephen Gould, Richard Fulton, Daphne Koller, Decomposing a Scene into Geometric and Semantically Consisten Regions.
[15] Stephen Gould, Joakim Arfvidsson, Adrian Kaehler, Benjamin Sapp, Marius Messner, Gary Bradski, Paul Baumstarck, Sukwon Chung, Andrew Y. Ng, Peripheral-Foveal Vision for Real-time Object Recognition and Tracking Video.
[16] P. Dorninger, C. Nothegger, 3D Segmentation of Unstructured Point Clouds for Building Modelling.
[17] Federic Bretar, Michel Roux, Hybrid Image Segmentation Using Lidar 3D Planar Primitives.
[18] Xuchun Li, Lei Wang, Eric Sung, AdaBoost with SVM-based Component Classifiers.
[19] Hao Zhang, Chunhui Gu, Support Vector Machine versus Boosting.
[20] Congcong Li, Adarsh Kowdle, Ashutosh Saxena, and Tsuhan Chen, Towards Holistic Scene Understanding: Feedback Enabled Cascaded Classification Models, Neural Information Processing Systems (NIPS), 2010.
[21] Canny, John, "A Computational Approach to Edge Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence,Vol. PAMI-8, No. 6, 1986, pp. 679-698.