Textured Object Detection in Stereo Images using Visual Features

Michael Levin (005529603)

December 12, 2008

Introduction

A robot can be easily modified to gather depth data alongside optical images. When combined with an effective object detection framework, this allows the robot to not only identify potentially interesting objects but locate their position and orientation within the world. This information is useful for obstacle avoidance and navigation, locating landmarks for positioning, and for locating a target object in the world. If the detection is accurate enough, the robot can also use the information to locate a grasping point.

The goal of this project is to produce a trainable object detector suitable for this setting. During training, visual features from highly textured areas of the image are extracted and stored along with their world-space position to form a stored "view" of the object. When presented with a novel scene the detector will try to match highly textured points with the database of stored object descriptions to find consistent formations of features that would indicate the presence of a known object.

This project is supervised by Dr. Gary Bradski at Willow Garage. Willow Garage provided the necessary equipment to capture dense stereo images. Upon completion of the project, the resulting system is planned to be integrated into the Robot Operating System and tested on the Personal Robotics platform currently under development at Willow Garage.

Stereo Image Dataset

For this project, stereo images were collected using a Videre binocular camera. Depth data is determined from the image pair using block correspondence as described in [3]. Because this method works by finding similar patches in left and right images, it does not work well on poorly textured surfaces. In order to overcome this problem, a second set of images is taken in addition to the initial images while a random noise pattern is projected onto the scene. The second set of images is then used for depth calculations while the left image of the first set of images is used to extract visual features (see Figure 1).

The dataset (Figure 2) consisted of 111, 640×480 stereo color images of ten toy objects. The background was known and kept fixed throughout training and testing and was used in foreground segmentation. A scene could be composed of one or more known and unknown objects and multiple instances of one object type. Objects were placed in novel position and orientations and allowed to occlude each other to present a challenging detection problem.

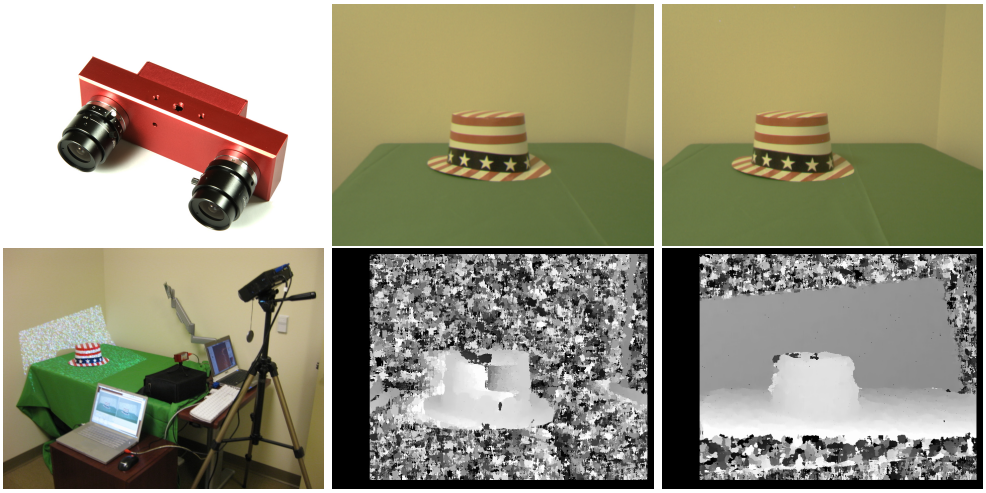

Figure 1: *On the first row is the camera used and the output 640 by 480 stereo images. In the second row is the dense stereo setup and the disparity map without and with the projector.*

Training

The detector must first be trained on the set of objects that are to be detected. Using one of the images from the stereo pair, SURF [1] feature points (or alternatively SIFT [6]) are first extracted along with the depth calculated from the dense stereo image pair. These points build up a sparse point cloud description of the object. To robustly recognize an object, enough views must be collected so that no area of the object has not been captured.

After all object descriptions have been stored, a point classifier is trained to classify feature point vectors as belonging to one or more of the trained objects. The classifier does not have to precisely partition the points but needs only to determine if a point could be a plausible member of an object class. Random forest, k-nearest neighbors, and k-means clustering were investigated for the point classifier.

The random forest classifier was chosen for feature classification for it's efficiency and straightforward parallelism. Random forests have been shown to have recognition performance matching multi-way SVM and superior computational speed in image classification tasks [2, 5] and will perform well even when built in a completely random fashion [4].

However, it was found that both the random forest classifier and the k-nearest neighbors classifiers tended to overfit the data. Better results were obtained by using k-means clustering (with k proportional to the data size) to build a model of where in feature space the points were distributed for each object. To test against this, the distance between the test point and the nearest cluster was compared to the cluster's standard deviation. This kind of per-object classification was better suited for the purposes of object detection.

Object Detection

When scanning a novel scene, depth and visual features are extracted in the same fashion as during training. The following algorithm is then run over each of the object views collected during training:

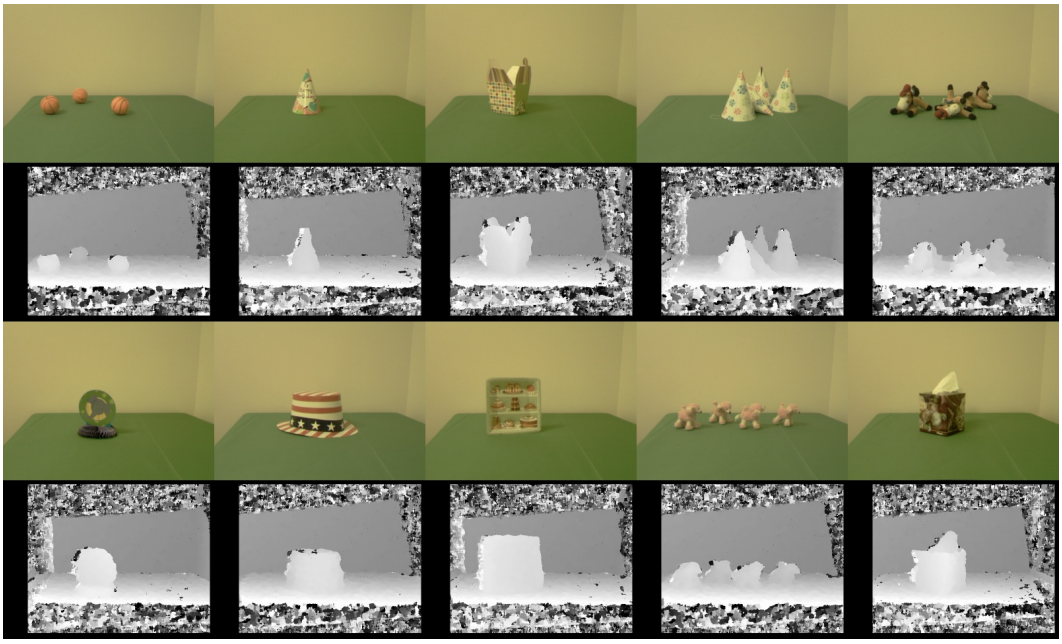

Figure 2: *Each of the ten classes of object used in the dataset with the corresponding dense stereo disparity map shown below the left camera image.*

1. Random sample consensus (RANSAC) is used to find a translation and rotation of the view into the test space. Only very good feature matches are used as potential inliers.

2. After a potential position in test space is found, validation is performed. The volume around the points composing the view after transformation into test space are used as a hull and test points that are classified as belonging to the current view's object class that fall within this volume are collected. If enough points are collected, all collected points are removed from the test scene and the process repeats again for the current view. This allows the detector to match multiple instances of one object.

3. If less than three points pairs were found for RANSAC, if RANSAC failed to find three good inliers, or if validation did not collect enough points, the process moves on to the next view.

The results for some of the test images are shown in Figure 4. The detector worked well on larger, well-textured objects that produced large feature point clouds but worked poorly on smaller objects. Part of the reason for this was the low resolution of the camera. A large portion of the feature points produced for small objects contained pieces of the background and would not match if the object was moved in the scene or occluded.

One of the advantages of this detector against a purely feature-based approach is the ability to match non-planar objects through limited rotations. If depth information were not available the detector would have been forced to treat the object as a plane and would not have found the correct orientation adjustment for the feature point cloud.

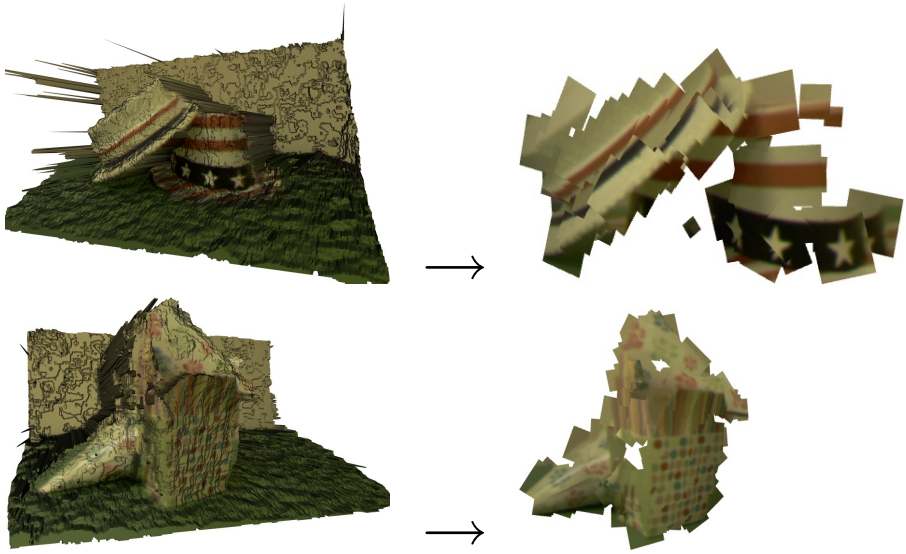

Figure 3: *A visualization of the depth maps for two scenes and the extracted feature point clouds.*

Conclusion

The object detector produced for this project would be ideally suited to detection tasks with a limited "vocabulary" of objects such as landmark detection or finding a target object in clutter. The relative computational efficiency of the detector also allow it to run in real-time. Overall the approach has promising applications and would perform even better with more accurate equipment.

References

[1] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Proceedings of the ninth European Conference on Computer Vision*, May 2006.

[2] A. Bosch, A. Zisserman, and X. Muoz. Image classification using random forests and ferns. *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct. 2007.

[3] Gary Bradski and Adrian Kaehler. *Learning OpenCV*, pages 415–453. O'Reilly Media, Inc., 1st edition, Oct. 2008.

[4] Pierre Geurts. Extremely randomized trees. In *Machine Learning*, page 2006, 2006.

[5] Vincent Lepetit and Pascal Fua. Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:2006, 2006.

[6] D.G. Lowe. Object recognition from local scale-invariant features. *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, 2:1150–1157 vol.2, 1999.

Figure 4: *Object detection results for test images. Query images and their extracted feature point clouds. The feature point clouds from the matched view are overlaid with colored borders.*