The Eyes of Iorek Byrnison

John Oberlin
Brown University, 2014

Abstract

State of the art techniques in object detection and pose estimation are powerful and general but usually run at a rate less than 1 Hz. This makes it difficult to employ such techniques in real-time human-computer interaction. This document outlines a simple, robust framework for object detection which trades a large memory overhead for improvements in latency and total throughput of detections. Included is a workflow for that framework which makes training and calibration first intuitive and then automatic.

1 Introduction

Our overall pipeline can be described as:

1. Collect RGB-D data for objects.
2. Train BoW model and kNN classifiers for objects.
3. Use the classifiers and additional logic to provide 3D detections pose estimates of objects.

2 Basic Techniques

Objectness try clustering the proposals for blueboxes.

Fast Keypoints

SIFT descriptors

KMeans

BoW

Color Histogram

Depth Histogram yet to implement

kNN

3 Detector Structure

Green Boxes

Blue Boxes

Red Boxes

4 Teaching the System

This is teaching not training because it is an interaction used to collect data not a script used to optimize a function.

4.1 Background Filtering

Teaching is carried out on a flat surface of uniform color. The background plane and color model are inferred by looking at parts of the image not contained in the blue boxes. These are used to keep with high probability only keypoints which are on the object. During inference (employment), background contributions will contribute to all examples approximately equally, and any confusion is likely to be understandable.

4.2 Manual Teaching

Teaching the system about an object involves showing it many different views of the object in a controlled environment. Once a session is initiated, the following process should be performed:

1. Repeat k times:
   (a) Adjust the object to expose a novel viewpoint and record pose data.
   (b) Remove extraneous objects in the scene to ensure a unique blue box is present.
   (c) Collect the view, which saves RGB-D data and the background data.

Where k might be around 100. Now the collected data can be used to train object class and pose classifiers.

4.3 Self Filtering

Suppose that we want to gather data while a robot is holding the object. It is possible to impose geomentric constraints on kept data to ensure that captured keypoints belong to the object being examined and not to our manipulating robot. This is called self filtering.

4.4 Semi-Automatic Teaching

1. Repeat k times:
   (a) The object is placed in a robot's manipulator in a known grasp.
   (b) A base pose for the grasp is provided.
   (c) The robot collects view from many different precisely known poses, together with models for self and background filtering.

Where k might be around 3.

4.5 Automatic Teaching

1. Place an object in front of the robot.
2. Robot repeats k times:
   (a) Infer a stateless grasp on the object.
   (b) if a pose model exists, infer the base pose for the grasp. else, start a new pose model.
   (c) Perform semi-automatic teaching with the object.
   (d) Replace the object.

Where k might be around 3.

5 Employing the System

Object Poses on Table or Floor

Single Target Tracking

6 Yet To Be Implemented

Background Filtering

Depth Models

Train feature weights to minimize the leave-one-out error rate on training data.