

CSE 473/573

Object Detection and Tracking Pipeline

Final Project Proposal

Pranav Jain

Person Number : 50208349, UBIT : pjain4

Manvijay Lather

Person Number : 50200437, UBIT : manvijay

Muthuvel Palanisamy

Person Number : 50246815, UBIT : muthuvel

CSE 473/573

Object Detection and Tracking Pipeline

Final Project Proposal

Introduction

The purpose of this project is to test and create an optimal object tracking and detection pipeline. This document will highlight the approaches that will be taken to form the pipeline and list briefly the reasons for taking those approaches and end with Bibliography and a rough time line.

Scope of the Problem: We feel that this would be a really good starting point to acquaint ourselves to the existing tracking algorithms. The ability to use open source libraries makes this project feasible while at the same time exposing us to new API's ¹

Methodology

This section explains the stages required to form an optimal pipeline. We plan to use OpenCV with Python for testing out the code

1. Object Detection Only

In this phase object detection is run on an input video for all the frames. For object detection Google TensorFlow's Object Detection API will be used that was recently released by Google.

2. Intialization with Object Detection followed by Tracking

In this phase, The TensorFlow API is just used on the initial frame to detect objects and a tracking algorithm like Lucas-Kanade will be used to keep track of the objects in the subsequent frame.

¹ Object Detection API by Google's Tensorflow

3. Hybrid Approach

For this phase, The initialization will happen just like in part 2, However after every few frames the object detection algorithm will be called to strengthen the current belief.

Expected Results

The First approach where only Object Detection is used is expected to be Naive and slowest of all the approaches as it requires more processing to detect objects rather than tracking as in the case of tracking we already have a decent idea about the appearance of the object that we are tracking. As a result of this the second approach intuitively should be the fastest. The first approach should however be the most accurate as it is easy to accumulate error while using tracking on frames. A way to make the second approach more robust is to reinitialize the belief after every few frames and is proposed in the hybrid approach

We expect to include a comparison of the running times of all the mentioned approaches. We also plan to include a comparison of the accuracy of the methods which will imply their ability to track objects in a video sequence.

Timeline and Responsibilities

1. Installing OpneCV with TensorFLow - November 15th, 2017 (To be done by all team members)
2. Object Detection Code - November 23rd, 2017. (Primarily by Manvijay and Muthuvel)
3. Object Tracking Code (Primarily by Manvijay and Pranav) and Hybrid Approach (Primarily by Muthuvel and Pranav) - December 7th, 2017.
4. Report Writing and Submission – December 15th, 2017 (Report will be made jointly by inputs from all team members)

Bibliography

1. <https://research.googleblog.com/2017/06/supercharge-your-computer-vision-models.html>
2. <http://16720.courses.cs.cmu.edu/hw/hw3.pdf>
3. <https://towardsdatascience.com/building-a-real-time-object-recognition-app-with-tensorflow-and-opencv-b7a2b4ebdc32>