CSC 249 Homework 5

Chris Dalke

April 28, 2017

1 Introduction

For this homework assignment, we were tasked with testing two tracking algorithms and comparing their performance. I recorded a video of my face moving around, which I fed through open-source implementations of the Kalman filter tracker and the TLD tracker. I then analyzed the results, comparing the filter performance. When evaluating the performance, I compared the accuracy, stability, and computational intensity for the two filters. I also analyzed the differences in how the two algorithms calculate their results.

2 Procedure

2.1 Recording Video

I started by recording footage of my face moving around on a webcam. In the video, I made several movements that I thought would trick up the filters. I moved forwards and backwards relative to my display, making my face larger and smaller. I also tilted my face to the side, which obscured part of my face. Lastly, I moved my face around at different speeds.

2.2 Kalman Filter Tracker

For the Kalman Filter Tracker, I found an implementation of the algorithm that used the OpenCV library. The algorithm uses Hue blob tracking and smooths out the movement of the detected region using the Kalman filter.

To get the software to track my face in the video, I configured it to load my video and I selected a bounding box that covered my face in the first frame of the video. I recorded the display from the software, which can be found in `Output_Kalman.mp4`

2.3 TLD Tracker

For the TLD tracker, I found an open-source C++ implementation of the OpenTLD algorithm. This algorithm also uses the OpenCV library. Similarly to the setup for the Kalman filter, I set up the OpenTLD software to run on my video and passed in the initial bounding box for the face tracking. I recorded the display from the software, which can be found in `Output_TLD.mp4`

2.4 A Note on Running Tests

Since the software packages were built for different platforms and each require manual configuration to run, I've chosen to record videos of the results instead of attempting to set up the programs in a way that would run on any computer. I've also included the two software packages in the `Programs` folder, and I've cited the repositories from which I downloaded them at the end of this document.

3 Results & Analysis

3.1 Kalman Filter

The core of the Kalman Filter is an algorithm that keeps track of the estimated state of a system and the amount of uncertainty of the estimate. The filter receives a series of measurements over time, and produces estimates that are more accurate than an estimate would be using a single measurement.

In the context of face detection and tracking, the Kalman Filter takes in a bounding box that has been detected using a primitive algorithm on the current frame, and uses that noisy data to produce a more accurate bounding box prediction.

In the case of the software that I used, each frame is analyzed based on the image hue. We know that my face will have a consistent hue, which makes it easy to track against a background. The software finds the areas with hues close to what was originally specified using the bounding box, finds the connected region around the group, and produces a bounding box. This box is subject to noise and other inaccuracies, so feeding the box into the Kalman filter smooths it into a stable and usable tracker.

3.2 TLD Tracker

The TLD Algorithm consists of three parts which the algorithm is continually balancing: Tracking, Learning, and Detecting. The algorithm is initially given a bounding box which defines an object region.

In the Tracking and Detecting phases, it tries to find that object in the image and predict its path. In the Learning phase, the algorithm looks at its detection of the object and tries to develop a better model to minimize the future error.

3.3 Comparison

There are several main differences that I noticed between the two algorithms. The algorithms differed in performance, accuracy, and stability.

3.3.1 Performance

The first main difference was that the Kalman filter seemed to be more performant. The comparison is not entirely reliable since I had to run the two softwares on different computers, but the Kalman filter seemed more capable of realtime performance. When the TLD filter was in Learning mode, actively trying to improve its model, it put considerable load on my computer. Once

learning mode was off, the performance increased, but still did not match that of the Kalman filter.

I would suspect that this performance difference is due to the fact that the Kalman filter is relatively simple; it simply computers the connected regions and then performs prediction on the bounding box position.

3.3.2 Accuracy

In general, it seemed that the TLD filter was better at accurately finding my face in the image.

The Kalman filter was incapable of tracking just my face, and immediately expanded the bounding box to include my neck. This happened because the hue of my neck is almost identical to the hue of my face, and so it appears when the algorithm performs image analysis and finds the connected blob around my face.

In contrast, the TLD filter does not have this error. It does not expand its bounding box since it has been given a template from which it compares. The TLD filter, when it did find the face, very accurately outlined the correct bounding box. There were false negatives but never false positives in the detection.

3.3.3 Stability

When discussing stability, I am referring to the ability of the algorithms to continually track faces even when it hits corner cases, such as a fast moving face or the face turned to one direction. The Kalman filter has better stability, since its prediction features allows it to still return a reasonable bounding box even when it does not detect a reliable region in the image.

When the TLD filter loses tracking for the face, it has no way of predicting where the face would be, so the detection fails and cannot return a result. This is shown in frames in the video which have no bounding box.

In contrast, the Kalman filter is, at its core, built on estimations, and when it cannot find the face it is still able to make a prediction as to where the new bounding box will be based on the previous data. In the example shown below, I am rotating my face in an unpredictable manner. The TLD tracker fails in this case, and briefly stops detecting a face.

The Kalman filter, however, is able to predict where the bounding box should be based on previous results, which removes much of the random variation as the unexpected movement occurs.

This is evident in the example and in the video. The green dot represents where the image filtering has detected the center of the region is, and the red dot and rectangle represents the bounding box calculated by the Kalman filter.

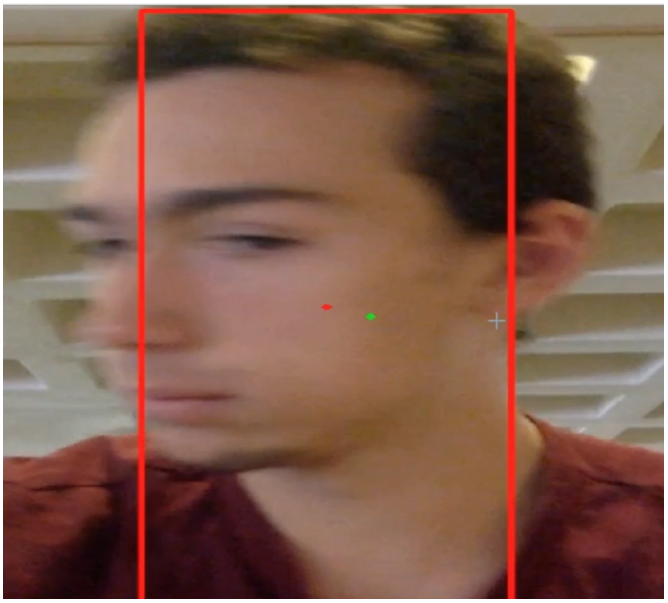

Figure 1: Kalman Filter response to Rapid Movement

3.3.4 A Final Note

These algorithms are not mutually exclusive, you could apply a Kalman filter to the result of a TLD algorithm, which could likely give a result better than both of the algorithms individually.

4 Citations

4.1 Kalman Filter Tracker

Github Project: camshiftKalman, `https://github.com/shaoguangcheng/camshiftKalman`

4.2 TLD Tracker

Github Project: open_tld, `https://github.com/pandora-auth-ros-pkg/open_tld`

5 Academic Honesty

I did not collaborate on this assignment and all work is my own.