Steven Brown
4/04/14

CSE408: Honors Project, Foreground Separation

Abstract:

This project is a program written in C to separate the foreground of a video from the background. Multiple history schemes are examined as are multiple adaptive background schemes. Schemes are chosen through command line flags for easy comparison. The accuracy of each scheme is examined in this report. This application is useful for tracking moving objects in a static frame.

.

Keywords:

Foreground-Background Separation, History, Object Tracking

Terminology:

- ffmpeg: a library to record, convert, and stream audio and video. For the purposes of this project, only the video conversion features are used and relied upon.

Goal:

The goal is to accurately class pixels from a video feed into one of two categories, foreground or background, in as close to real-time as possible.

Assumptions:

It is assumed that all pixels fall into one of the two possible categories, foreground or background. It is also assumed that the video feed is in a format recognized and readable by the decoding library, ffmpeg.

Description:

The first task of the program is to open up the video file or stream and determine the codec. This is handled through the ffmpeg library. Then the video frames are loaded and converted to rgb frames. RGB frames were chosen due to the luminosity channel not being sufficient information for the separation and to have a common format regardless of the input's encoding. As the frames are read, they are stored in a circular buffer for processing, with the head representing the most recent frame, f_0. At this point depending on the options the program is compiled with, one of a few things will happen. If the first adaptive background scheme is chosen, the background pixel data will be updated each frame regardless of its previous classification using the following formula:

$$f_b = (0.95)*(f_b) + (0.05)*(f_0)$$

where f_0 is defined as the current frame and f_b is what the program is maintaining as the background.

Next, after either updating the background information or not (depending on chosen scheme), the program will check the current frame vs. the maintained background for differences. Adaptive thresholding, the next option comes into play during this check. If it is off then a static threshold value (chosen by trial and error inspection) is used. If adaptive thresholding is used, the average difference between the previous 10 frames is used to compare each pixel, with a minimum value (also chosen by trial and error inspection).

If the differences are high enough, the pixels are then classed as either moving foreground or stationary foreground using temporal differences. Five different temporal differencing schemes were tested (average relative increased overhead value per frame in parenthesis): one previous frame (0 ms), one previous frame and one future frame (2 ms), two previous frames (2 ms), two previous frames and one future frame (4 ms), and finally two previous frames and one future frame (6 ms). The results were insignificant between the schemes, so the fastest, just one previous frame, was chosen. At this point, if the second adaptive background scheme is chosen, the background information is updated using the same formula listed above, but instead of updating all pixels, just those that are identified as stationary foreground pixels are updated.

Next, if the erode/dilate option is chosen, the program will perform an erode and a dilate using a 3x3 square as the filter. This helps remove a lot of small noise from the end result. Then the output is written to disk for debugging purposes and the next frame is fetched and the process repeated. When the video stream or file ends, the program cleans up and frees used resources and exits.

The results comparison between options may be found in Appendix A.

System Requirements:

Linux (written and tested on Ubuntu 12.04 LTS)

ffmpeg

Execution Instructions:

A makefile is included for building the project with the default options (B1F0, ADAPTIVE_THRESHOLD, ADAPTIVE_BG2). It can be run using the command line by invoking the resulting executable and passing it the video location. For example:

./main ./location/to/video.mp4

The resulting output will be written to the output folder located in the same directory as main (main will not create the folder if it does not exist). To run with non-default options enabled, uncomment their specific lines in the source (main.c) before compilation.

Appendix A: Option Result Comparison

Figure 1

Adaptive background off, adaptive threshold off, erode/dilate off, history scheme 1

As can be seen from figure 1, due to the hue change from much earlier in the video, the results are almost the reverse of what they should be, presenting a good argument for adaptive backgrounding for this specific algorithm.

Figure 2

Adaptive background scheme 1, adaptive threshold off, erode/dilate off, history scheme 1

Figure 3

Adaptive background scheme 1, adaptive threshold off, erode/dilate off, history scheme 4

Figures two and three show the insignificant differences between the history schemes of the fastest with only one previous frame, and the slowest with two previous and two future frames.

Figure 4

Adaptive background scheme 1, adaptive threshold on, erode/dilate off, history scheme 1

Figure 5

Adaptive background scheme 1, adaptive threshold off, erode/dilate on, history scheme 1

Figure 6

Adaptive background scheme 2, adaptive threshold off, erode/dilate off, history scheme 1

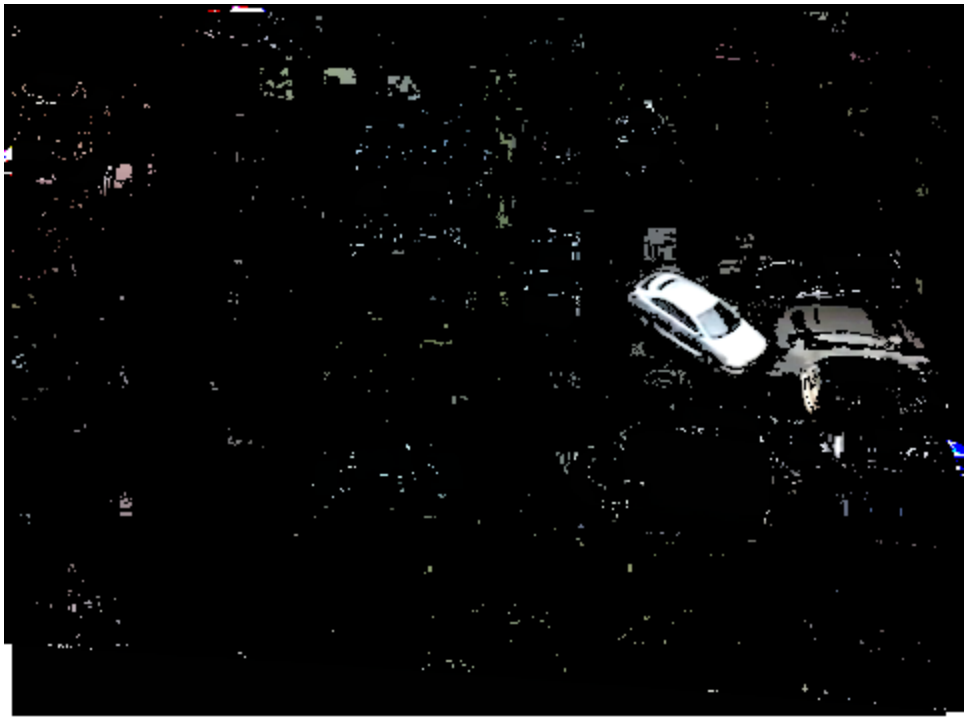

Figure 7

Adaptive background scheme 2, adaptive threshold on, erode/dilate off, history scheme 1

Figure 8

Adaptive background scheme 2, adaptive threshold off, erode/dilate on, history scheme 1

Figure 9

Adaptive background scheme 2, adaptive threshold on, erode/dilate on, history scheme 1

Figure 10

Adaptive background scheme 3, adaptive threshold off, erode/dilate off, history scheme 1

Figure 11

Adaptive background scheme 3, adaptive threshold on, erode/dilate off, history scheme 1

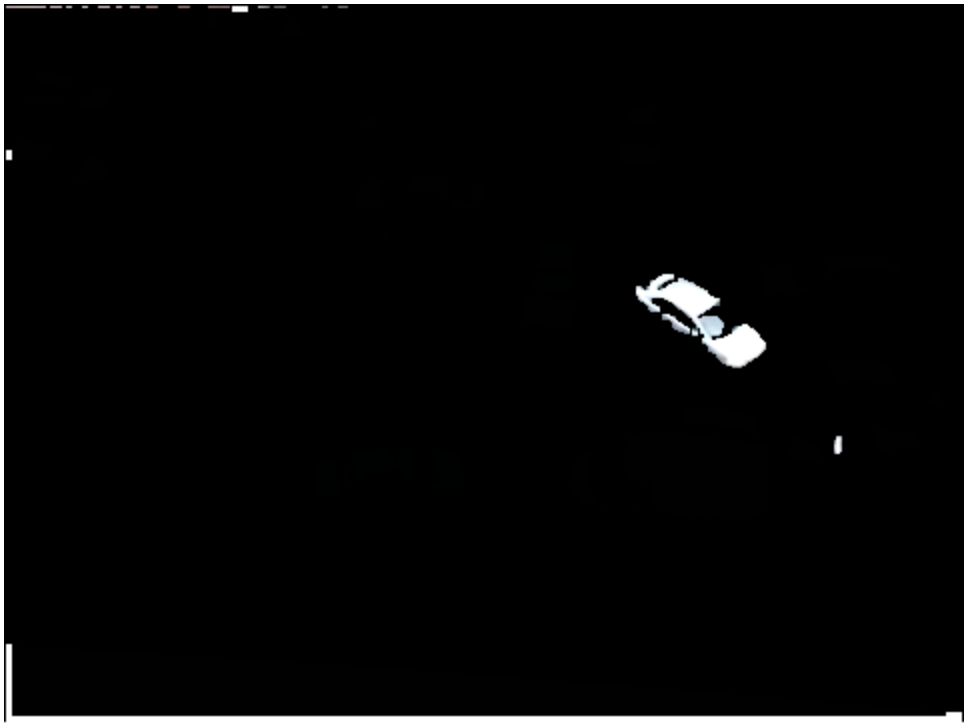

Figure 12

Adaptive background scheme 3, adaptive threshold off, erode/dilate on, history scheme 1

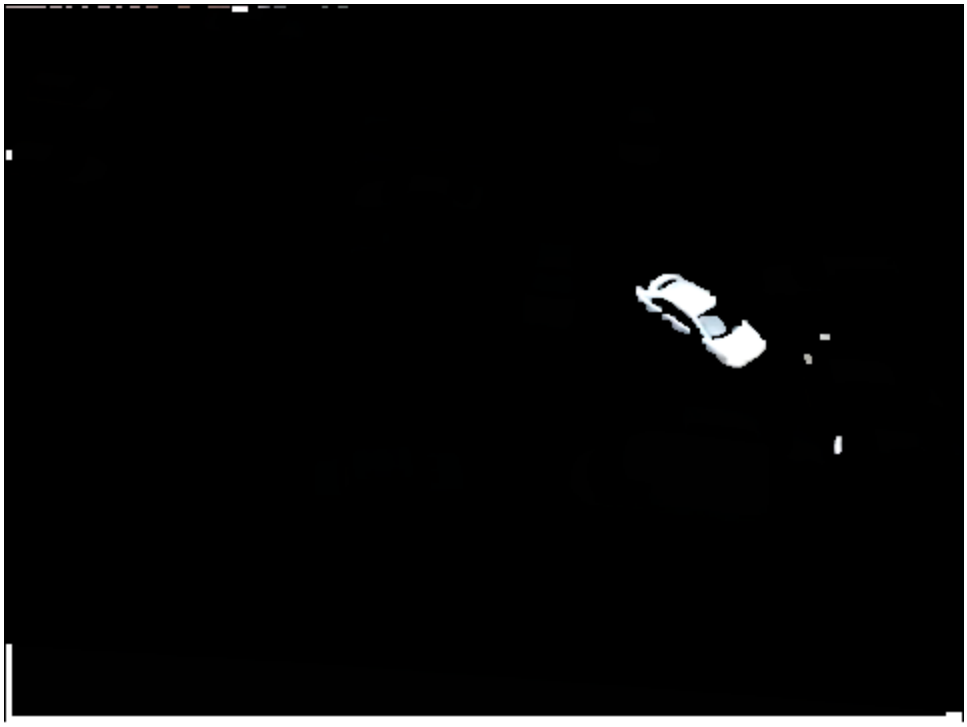

Figure 13

Adaptive background scheme 3, adaptive threshold on, erode/dilate on, history scheme 1

Figure 14

Original frame

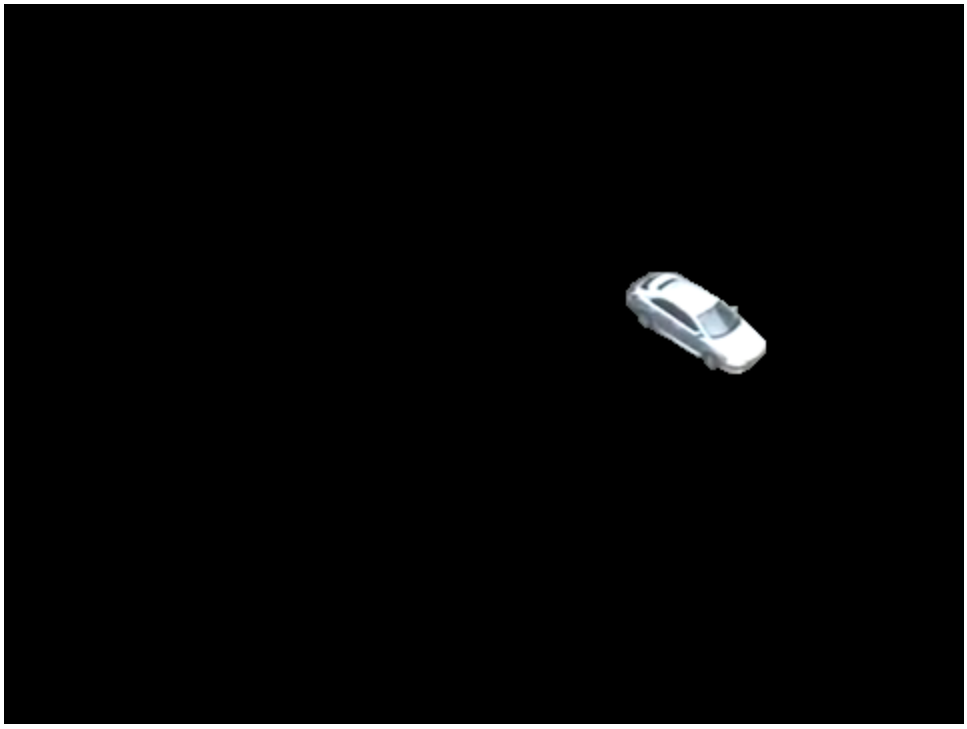

Figure 15

Correct Result by Hand

Figures 4-13 show various combinations of the options available. Figure 14 is the original frame and figure 15 is a frame with the correct results marked by hand. Of the various options, the ones to generate figure 12 consistently outperform the others. As a review, those options are: adaptive background scheme 3, adaptive thresholding off, erode and dilate on, and history scheme 1.