**Project 2 Essay**

Michael Nishida

CS 410 VC

3/14/14

I am doing the vehicle and pedestrian motion tracking and detection for my second project. I accomplished this by utilizing the VideoCapture, oVideoWriter and BackgroundSubtractorMOG2 objects to find and manipulate the video frames. The first and most important piece we need to utilize in order to track the motion in a video is the ability to go through a video file frame by frame. That is what the VideoCapture object does it reads in the video file and goes through the entire video sequentially frame by frame if we put into a while loop. This While Loop uses a Boolean value that will remain true as long as the VideoCapture is able to retrieve a frame.

The first part of this algorithm gets the static background of an entire video file. This static image is the accumulation of all the video frames combined together so that any moving objects in the video are blurred out and when this image is compared to any frame in the video file the moving objects will be clearly defined. The algorithm accomplished this by going through all of the frames in the video file using the while loop. For each frame I used the addWeight function to add each frame image to an average image whose weight or importance becomes greater and greater the more frames were added to it. The result to this is not perfect and the average image has so noticeable color streaks where there is a lot of heavy traffic and ghost images where a moving object remains stationary for a long period of time.

The next part of the algorithm does the actual moving object detection. It does this by finding all moving object in every frame of the video. Once the moving objects have been identified a box is drawn around each object in the frame and then the video frames are recompiled into a new video. In this section the algorithm once again uses the While Loop to go through the video frame by frame and uses the BackgroundSubtractorMOG2 object to find all of the moving objects. The BackgroundSubtractorMOG2 object takes a video frame by frame and calculates what objects have changed i.e. in the foreground and what parts of the image are static i.e. in the background based on the previous frames already processed. We can utilize this function to find out what objects have moved when compared with the previous frames and get a grayscale image outlining the moving object.

The BackgroundSubtractorMOG2 object is good for basic motion detection where a moving object is against a mostly static background but is not that great for vehicle and pedestrian motion tracking where the motion detector also needs to deal with a lot of background noise like trees swaying in the wind. This means that a lot of objects like pedestrians are almost completely overshadowed by this background noise.

The algorithm solves this issue by first using the average static background image that was created in the first part of the algorithm. By subtracting this background image from a frame that is being processed using the asbdiff function and then taking the threshold of the result the algorithm gets an outline of all objects that aren’t in the background image. This result is fairly good at identifying all of the moving objects in the frame and eliminating much of the background noise caused by movements in a small area like swaying branches.

While this technique may sound perfect to detect all the moving objects there is a flaw due to the color trails and ghost images left by the moving objects. This leaves certain areas where there is a high traffic of moving objects like crosswalks, street lines and curbs still visible in the subtracted image. How I deal with that issue is by taking that subtracted image for each frame and passing it through a BackgroundSubtractorMOG2 object which is being passed only those subtracted images. This means that the BackgroundSubtractorMOG2 will only detect the object outlines that are moving and return them as a result. The results of this are almost there but the objects still need to be filled in a little more. The algorithm accomplishes this by using a second BackgroundSubtractorMOG2 that gets sent the raw frames to detect the moving foreground images. The results from this second BackgroundSubtractorMOG2 are still noisy so it is passed through an erode function to eliminated the background noise. The results from the two BackgroundSubtractorMOG2 objects are then combined and the end result is blurred several times using the simple blur function to fill in as much of the outlines as possible.

With the outlines of the moving objects found the algorithm then finds the contours of each object outline using the findContours function. Using these contours the algorithm can then draw rectangles around the moving objects in the original frame. This frame is then saved to the new video file and the algorithm moves on to the frame in the video and repeating the process until all the moving objects in all the frames have been detected.