Abstract:

Mike Wu

Position and Vector Detection of Blind Spot Motion with Horn-Schunck Optical Flow

The goal was to create a cost-effective way to detect the presence of a vehicle in the blind spot in real time by computing its position, velocity and acceleration.

Our project is based off of the Horn-Schunck Optical Flow method, which uses convulsion by comparing a grouping of pixels within two subsequent frames. Since each specific pixel has a unique luminescence, Optical Flow traces pixels onto the next frame and draw vector arrows representing the movement of the pixels. We then split the vectors into car and background vectors.

To detect the position, we created a box to surround the car. The center of the box is the average of the (x,y) coordinate of the car vectors while the size of the box is directly proportional to the ratio of car to background vectors. Further advances like standard deviation filter and protection thresholds on movement and vector magnitude were used to prevent miscalculation.

Since our Optical Flow method relies on apparent car motion, a static car in the blind spot would not be detected. To overcome this problem, a memory system was developed that would look at past frames to determine whether or not a car is still present.

Besides assisting in static detection, the memory system was also used to save time lag by searching for the presence of a car faster. Basically, if a car wasn’t detected in previous frames, then we only at the left and right sides of a picture frame, because that is where a car initially appears.

A unique add-on was the use of stereo-cameras that basically use duo cameras positioned at converging angles to visualize depth of motion in specific cases of multiple object differentiation in the blind spot.

Besides the Horn-Schunck optical flow method, we also incorporated circle detection, which detects the presence of a car by looking for shapes resembling a circle. An edge detection was used before the circle detection to draw the layouts of the vehicle. A distance threshold was applied to the stereo data to remove background objects.

Velocity could be estimated from the movement of the center of the box throughout the frames and acceleration is easily calculated through a simple derivative.

In the trials that we ran, the program was able to accurately detect the vehicle and track its motion throughout the video. The code developed also extended to bicyclists, pedestrians and motorcyclists. Furthermore, the code was not depended on sunlight; the headlights of the incoming car were sufficient for vector generation. Thus, if applied to the real world, the real-time capabilities could be revolutionary in promoting safer driving conditions.