Example of Motion Estimation:

To achieve Motion Estimation, we first need to obtain two elements: Frames and Optical Flow. In this demonstration, we will be using two frames from a traffic camera footage[1].

To obtain optical flow in this demonstration, we will be using the OpenCV module with Python. Optical flow is calculated using the distance travelled between two pixels with the dimensions and time. The equation would be the following:

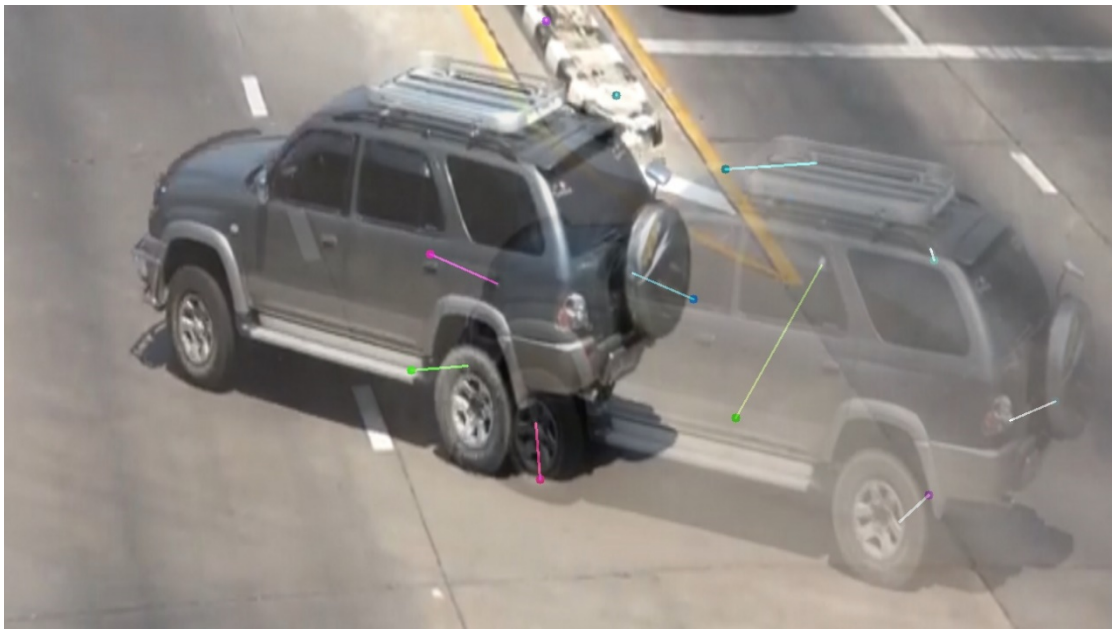
[1] https://www.youtube.com/watch?v=jjlBnrzSGjc , 4K camera example for Traffic Monitoring (Road), Panasonic Security

$$I(x, y, t) = I(x + dx, y + dy, t + dt)$$

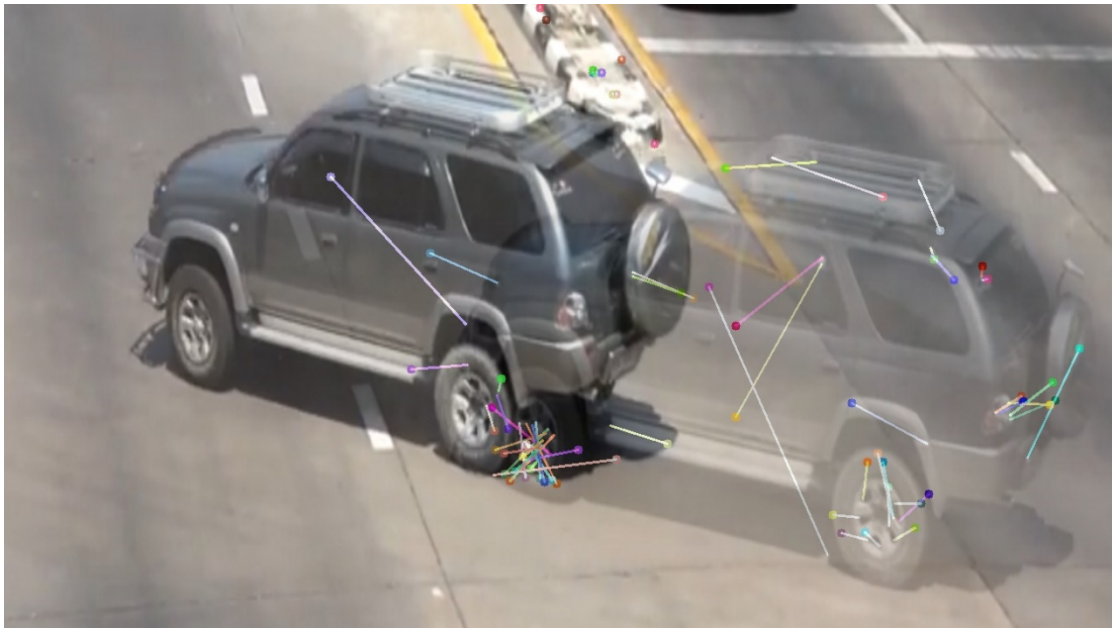
$$f_x u + f_y u + f_t = 0$$

This is called the optical flow equation. In this equation, we must assume that the intensity of pixels is consistent within the frames. As we need to track down the moving pattern of the same object. The x, y represent the coordinates of the pixels and t is representing the time. We will use the Lucas-Kanade function within the OpenCV module to calculate the optical flow within this demonstration.

Then, we will obtain the optical flow between the frames. The optical flow can show us the moving pattern of the moving object with a stationary background. Below we will show two examples of obtaining optical flow within two RGB frames that are converted to grayscale during the process.



(Figure ME1)



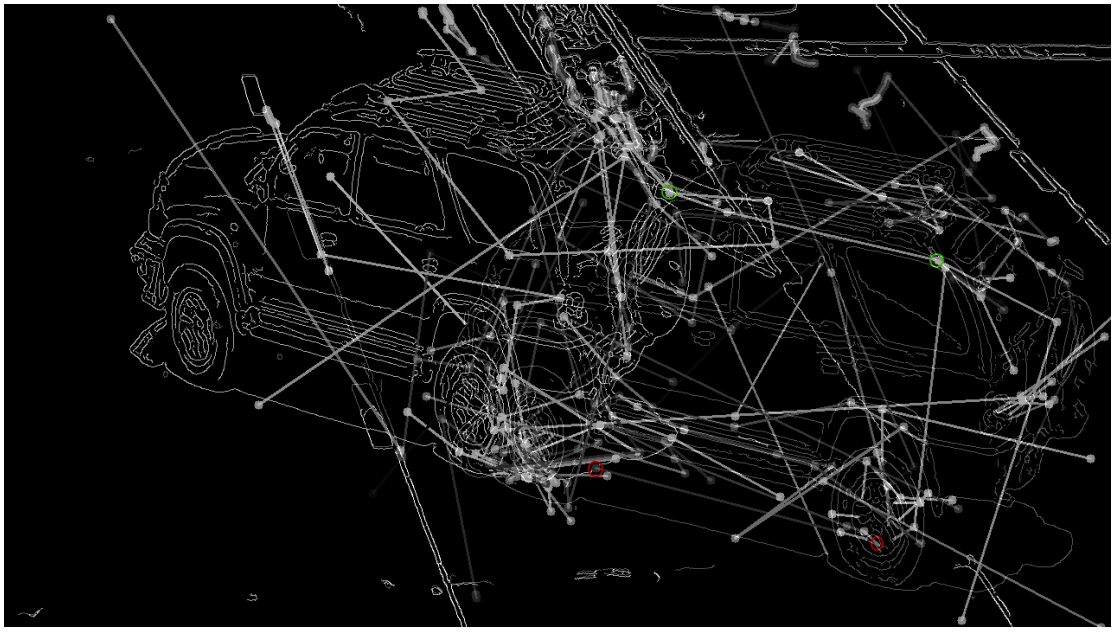
(Figure ME2)

We can see that if the optical flow was obtained within frames that were only converted to grayscale only, no matter how many reference points were added onto the frames. The result will not be accurate. This is mainly because even though we gave some points for it calculating the optical flow, the moving object's intensity was not consistent throughout the frames.

Therefore, we need to further enhance the accuracy of the result to achieve higher video quality. We can do that by applying some image processing techniques. Apart from using the grayscale frames to obtain optical flow, a more consistent way must be used.

As the intensity is playing a crucial role when calculating optical flow, the best for us to obtain optical flow will be using two unified frames while the object will have consistent intensity throughout the process. We can first blur the grayscale frames by using the Gaussian filter to eliminate as much noise as possible within the frames for edge detection. The edge detection we are using here is the Canny Edge Detector. When we are using edge detection, the object can then be highlighted consistently with just some white lines.

In this way, the optical flow can be calculated easier. Now if we used the same calculation within (Figure ME2) we will obtain the following result.



(Figure ME3)

From this result, we can see that there are more data being shown. Some points within the result are highlighted. These are part of the optical flows that can be used in motion compensation. However, there are still many errors within the result. This is mainly because we are calculating the optical flow using a sparse feature set and the two frames are not really two consecutive frames.