Assignment 2: Feature Tracking

Sudheer Nadella
U93511802

Abstract

Feature tracking algorithm allows us to track the feature points in a video by just calculating the feature points in the first frame and mapping them throughout the rest of the frames. The tracking is done finding Eigenvectors from correlating the image and then finding the K highest feature points considering the pixel locations. We now take a frame over these feature points in the next image and perform SSD(Sum of Square Distance) considering the minimum displacement pixels to be our feature points in the next image. We then update the feature point values till will perform the correlation on all the frames of the video.

1. Algorithm

The Algorithm has three parts. FirstOrderDetection to get dx and dy gradients. ZeroOrderDetection to get the feature points of the first frame of the video. Correlate to find the feature points of the next frame based out of feature points of the previous frame.
For dx and dy we use the first order 1-D Gaussian kernel. For feature points we take dx and dy and use zero order 2-D Gaussian kernel to get the eigenvectors and considering the top K eigens as feature points. Now considering the locations of the top K feature points of the previous frame, we consider a window large enough to trace out the displacement of our target. In that window we put a sliding frame over which we calculate the SSD by first finding the difference between currentImagePatch and previousImagePatch, squaring the difference and multiplying the squared difference with our 2-D Gaussian kernel.We now calculate the sum of the values in the multiplied result and assign the value to that pixel.By repeating it till the sliding frame covers the main window we get that SSD values.The minimum of the SSD's will be considered and the location of that particular pixel is our new feature point. We repeat the same process by updating the previous with current for next frames.

2. Outputs

Fig 1:dx moon

Fig 2: dy moon

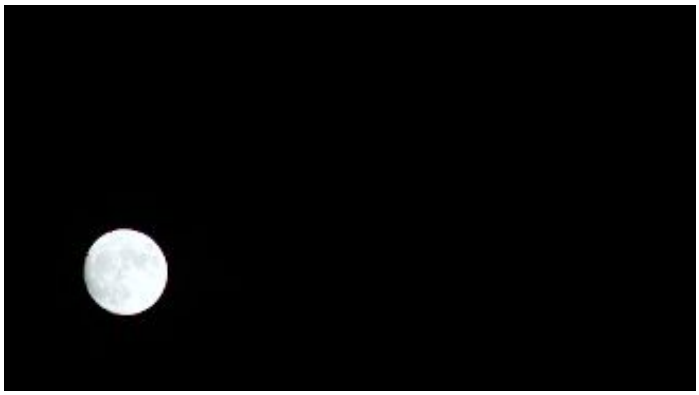

Fig 3: Feature points in first frame 0.png

Fig 4: Feature points in 54 frame 54.png

Fig 5: Feature points in first frame 22.png

Fig 6: Feature points in 90 frame 90.png

As we can clearly see how the feature points are changed according to the target in the image.
Consider the moon images, we have the 1st and 54th images from the output. The feature points are changed with respect to the target in the frame. We can also see what the dx and dy images of an image look like. Now consider the walking frames, look at 22 and 90. The number of feature points on the target in 22.png is 5 and in 90.png is only 3 because the person is walking out of the frame and part of his body can't be seen. So there are only three points on the body.

3. Issues Faced

While considering the frames to compute the feature points we have to check the boundary. And for walking frames, the feature points are not clearly visible if the file is in ".jpg" format. So for clear visibility I have saved the outputs in ".png" format.