Assignment 2: Tracking Feature Points with Kalman Filter

Daniel Sawyer
University of South Florida
danielsawyer@mail.usf.edu

Abstract

Using Harris detector for feature detector by finding Eigen vectors from correlating an image with itself. Then taking the top K feature points as x,y values of pixel location. We then use a Kalman filter with a constant velocity in order to get the minimum value for the covariance patch by correlating with the difference squared of the intensities from the original image and current image summed up within a set window area. We then update the state every optimal amount of frames f(x) based off some function dependent on the domain, velocity, etc factors.

1. Algorithm

The algorithm is broken into four parts. The convolution to get dx and dy gradients, feature detection, Kalman filter, and model update function. For convolution you use the basic 1st order one dimensional discrete Gaussian kernel which you can easily calculate for a predetermined window size. You then autocorrelate the image with itself in order to calculate the Eigen feature points that we want to track. Then run the Kalman filter. I used all of the formulas from the slides posted by Dr. Sarkar and it worked quite well with the original P1 formula without the Identity matrix. My main problem was the update function which was originally just 1, meaning it updated the model every frame. After switching this to 5 frames that increments by 5, it was able to track the feature points fairly well.

2. Related Work

I used the Gaussian kernel Dr. Sarkar used in class along with along with the Kalman Filter. I didnt use any online resources.

3. Examples

I ran the algorithm on two different videos. One with a moon traveling through the sky at night and the other was the Test1 provided by the class with a man walking from right to left. The algorithm was able to follow both pretty well with the models and parameters set in the PDF put up on canvas.

Figure 1. Moon Dx

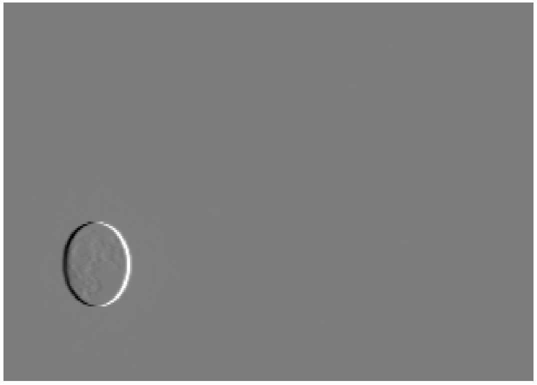
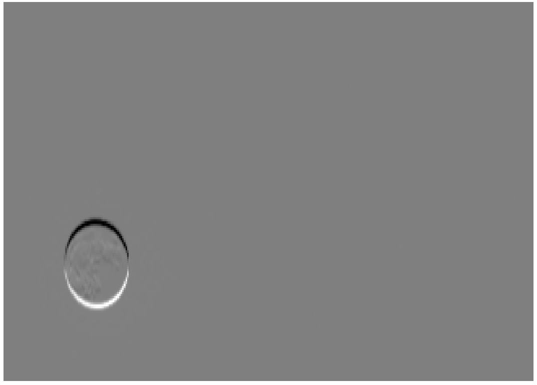


Figure 2. Moon Dy



4. Lessons Learned

I learned quite a few lessons with this project. The major lesson is how you can use Eigen values to find feature points by convolving the image into an Dx and Dy images, then autocorrelating the images into Eigen values. We then take the location in x,y coordinates and track it using the Kalman Filter. A major lesson I learned was with the Kalman Filter, you dont want to update the points, previous frame, and model every iteration/frame but every X frames where X can be calculated by either trial and error or by writing a

Figure 3. Moon Eigen Feature Points

Figure 4. Moon Tracking, Red Points

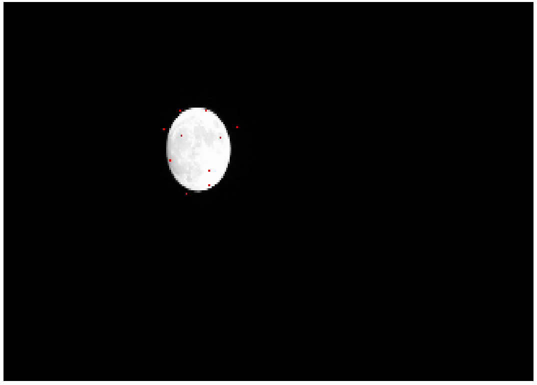
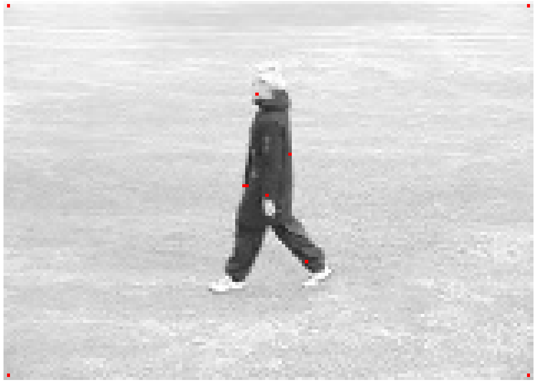


Figure 5. Walking Tracking, Red Points



function to calculate it based off Kalman Gain and/or the P covarariance matrix. This was a hard project and spent many a sleepless nights working on it and think I did pretty well.

References