Assignment 2 Report

Ashwin Rao (2019101049)

Harris and Shi Tomasi Corner Detectors

Implementation

- Both these methods are used to detect corners in an image.
- The main difference is that they use different metrics to find corners. In Harris corner detection, we use the formula $det(H) - tr(H)^2$. In Shi-Tomasi corner detection, we directly use the lowest eigenvalue of H.

Observations and analysis

- Shi-Tomasi corner detection results in a denser output, since it returns more points than Harris corner detection. If we want to make the output of Shi-Tomasi sparse, we can always add a constraint to return only the top n points.
- Shi-Tomasi is computationally more expensive than Harris corner detection. This is because we need to compute the eigenvalues of H at each point. On the other hand, both $det(H)$ and $tr(H)$ are easy to compute.
- For direct comparison purposes, I ran both methods with the same threshold ($= 0.005$) and evaluated them using the same error metric, on the same three images, the first of each sequence.
- Increasing the window size lead to better results, but lower efficiency due to more computation. Decreasing the threshold also lead to better results.

Lucas Kanade Algorithm

Observations and analysis

- I first obtained and plotted the results for the first two (consecutive) images of each sequence.
- The algorithm is computationally expensive and took a significant amount of time to run on my local machine. This is because the computation of eigenvalues of a matrix is an expensive operation.

- I found that the algorithm worked better for the RubberWhale sequence, while it performed relatively worse on the other sequences.
    - In the RubberWhale sequence, the flow is more organized, less noisy and similar over larger regions. An image in this sequence can be divided into four parts based on flow, and the algorithm performs well.
    - On the other hand, the Grove3 sequence, for example, although having more corners and edges, is highly unorganized, with flow varying from point to point. The estimated flow is hence noisier, and the algorithm performed worse.
- Increasing the window size and using a reasonably small threshold lead to better results. I used a threshold of 0.005 and a window size of 5×5.
- I also observed that the results are much less dense than the ground truth flow.
- The analysis of flow videos showed the overall direction of motion of flow in the sequence of images.