Computer Vision
Assignment 2 Report

Roll: 2019101017
Name: Shaurya Dewan

2. Here, we are tasked with implementing and experimenting with the Harris and Shi-Tomasi corner detectors. The primary difference between the two is the metric used to find corners. In Harris, we make use of the formula |H| - tr(H) ^ 2 while we simply use the lowest eigenvalue of H for Shi-Tomasi. In my experiments and analysis of both, I found that Shi-Tomasi returns far more points than Harris making it more dense. In fact, we can constrain Shi-Tomasi to return the top k points in order to make it sparse. However, Shi-Tomasi also takes longer and is computationally more expensive as we need to explicitly compute the eigenvalues of the H matrix at each point. In order to compare both, I made use of the same image from all 3 given sequences with similar thresholds applied on the metrics used.
   In my experiments, I also found that increasing the window size, decreasing the threshold and giving more weight for the determinant term (in the Harris detector) gave the best results. However, the increase in window size came at the cost of increased time/decreased efficiency which was not worth it considering that the quality of results did not vary as much. Decreasing the threshold does give a few unwanted points but a high threshold gives very sparse results which is also undesirable.
   For the single-scale Lukas-Kanade algorithm, I first obtained the results for a pair of images from all 3 sequences. I found that this algorithm is time-consuming as this also involves the computation of eigenvalues of a matrix. I observed that the algorithm performed best for the 'RubberWhale' sequence which can be explained by the organization of the sequence where we can easily roughly divide each image into 4 non-overlapping subparts based on their direction of flow. It performed worse for the other two as they are not as well-organized. Moreover, the shift is much higher

for the other two sequences and this algorithm is best suited for smaller shifts. The algorithm performs worse as the shift between two frames increases.
During my experiments, I observed that increasing the window size with a relatively low threshold gave the best results with the same cost of time. Also, the results we obtain are far more sparse than desired such as compared to the groundtruth.

3. I found that the multi-scale version outperforms the single-scale version as expected. I primarily find that the magnitudes of shift are far better predicted by the multi-scale version while the direction prediction remains largely similar. More importantly, it does far better on the ‘Grove3’ and ‘Urban2’ sequences which have larger shifts between frames. However, it does take longer as it repeats the algorithm multiple times at different scales. Moreover, increasing the number of layers/scales/levels leads to further improvements compared to lesser levels.