Lucas-Kanade and Horn-Schunck Optical Flow

Both the Lucas-Kanade and Horn-Schunck methods are coded using OpenCV for convenient image processing functions. The optical flow algorithms themselves are coded explicitly for this assignment.

Lucas Kanade Results

This particular algorithm uses a brute-force method to evaluate the entire Gaussian-smoothed image in small kernel intervals for optical flow. For a faster code, it would be desired to limit evaluation to a finite number of edges, which contain enough information to avoid the aperture problem.

The results below demonstrate optical flow vectors observed for a given image sequence. Optical flow appears reasonable for the sphere, though weighted velocities seem incorrect. The rubic's cube seems susceptible to noise, as the background registers some flow:

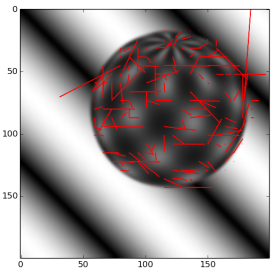

Figure 1 – W/ Gaus. Weight

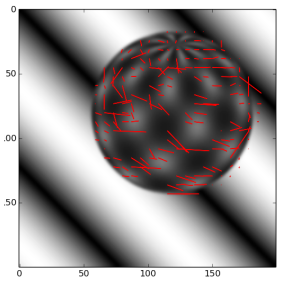

Figure 2 – W/out Gaus. Weight

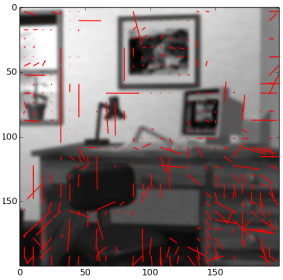

Figure 3 - Office LK

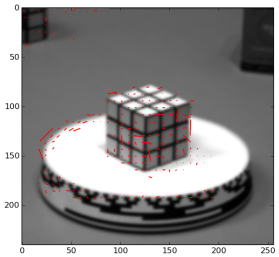

Figure 4 - Rubic LK

Horn-Schunck Results

The Horn-Schunck algorithm assumes uniform flow throughout the image and estimates velocity through an iterative approach. Because of this assumption, we no longer need to evaluate small kernels of the image under the assumption that optical flow is uniform within the cell. Results appear much more along the lines of what we would expect. For reasonable iteration numbers, real-time control is a possibility. Reducing the image resolution is one potential solution to reinforce this.

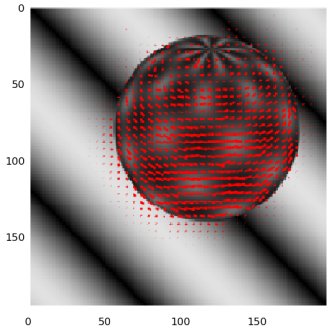

Figure 5 - Sphere HS

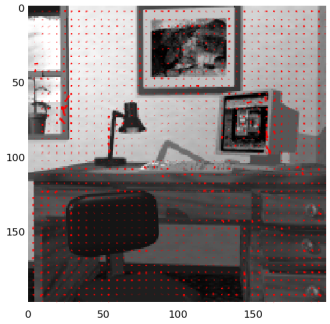

Figure 6 - Office HS

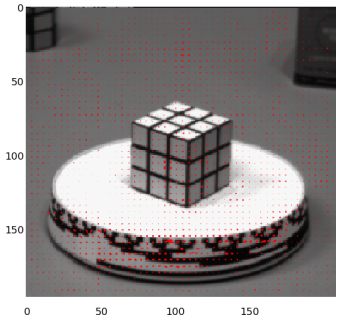

Figure 7 - Rubic HS