Project #3: Optical Flow

Yang Xu (yxu71@vols.utk.edu)

Problem Description

Optical flow is the apparent motion of brightness patterns in image and can be applied to track movement of objects. The solution to optical flow problems are based on two key assumptions: 1) brightness is constant; 2) motion is small. In this project, Lukas-Kanade algorithm is used to solve optical flow equations. For better motion estimation, Harris corner detector is applied to avoid central pixels that may cause aperture problem.

Code Design

Function	Description
conv_de(img)	This function uses prewitt filter to calculate derivatives of the image. It returns derivatives of x and derivatives of y, separately.
lk_optcflow(frame1,frame2,window_size=5,k=0.05)	Use Lukas-Kanade algorithm to estimate optical flow. k is the threshold for corner regions. Within the window block, central pixels with R > 0.05 will be regarded as corners, otherwise edges or flat regions.

Assumptions

1. Brightness is constant;
2. Motion is small;
3. pixels and neighboring pixels have spatial coherence;
4. Solvability of optical flow equations rely on eigenvalues.

Running Code

To run the code, open the terminal and run the code below

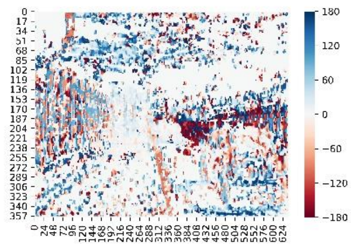
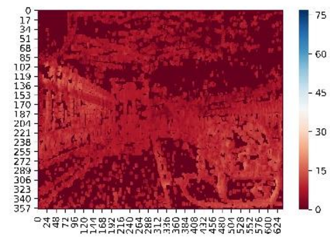
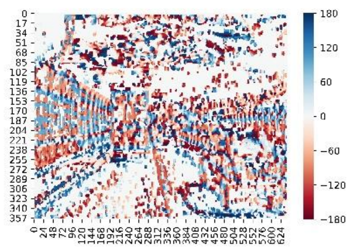
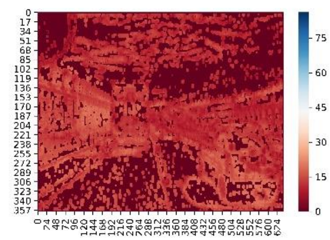
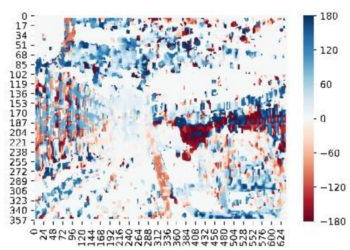
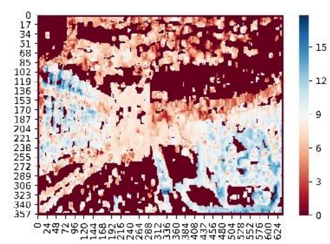
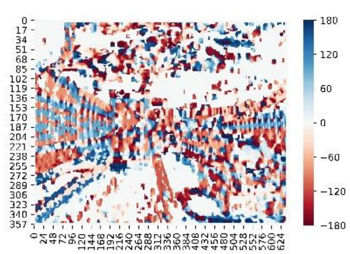
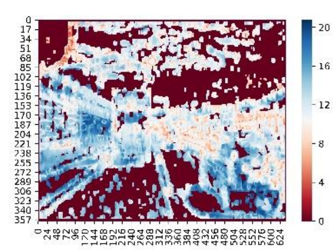
python optical_flow.py Daylight_00564.jpeg Daylight_00565.jpeg 5 0.05

The python script takes two image frames as inputs to estimate optical flow. The first image is considered earlier frame in time series, and the second image should be 1 or N frames later. The 3rd parameter is the window size, which is used for Harris corner detection and Lukas-Kanade algorithm. The final parameter is the threshold in Harris corner detector that defines regions as corner, edges or flat regions.

Example Results

The example results showed estimations of optical flows from frame 00567 to frame 00568 (1 frame apart) and from frame 00567 to frame 00726 (N frames apart). Angle of movement is set from -180 degree to

180 degree. The magnitude of movement is first log-transformed and then presented in heatmap. I set k = 0.05 as threshold for corner regions and use 5 and 9 two window sizes in this project.

	Angle	Magnitude
Window size = 5, 1 frame apart.		
Window size = 5, N frames apart.		
Window size = 9, 1 frame apart.		
Window size = 9, N frames apart.		

Conclusion

1. This experiment shows a larger window has a better estimation than the smaller window (5 and 9 in this case). When window size is 9, the difference of movement between the car (on the right side) and the cargo (in front) is distinct. In fact, the car has larger motion than the cargo. From the view of movement angle, window size 9 and window size 5 both can identify that the car has around 120 degrees movement (toward top left) and the cargo has around 30 degree (towards top right).
2. When two images are N frame apart, the estimation seems inconsistent. Both cargo and car moved towards the camera point. However, car moved away in fact. Interestingly, the two lines on the sides should have the direction of movement but the estimation shows the opposite.