CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS

BRUNO NASCIMENTO DAMACENA

In collaboration with

GABRIEL DUTRA DIAS
RAMON GRIFFO COSTA

Problem Set 4

Problem Set proposed by Flávio Luis Cardeal Pádua for the Computer Vision course from Computer Department of CEFET-MG

Belo Horizonte
June 2019

1 Variations of the Horn–Schunck Algorithm

The Horn–Schunck method is a global method of estimating optical flow which introduces a global constraint of smoothness to solve the aperture problem. The aperture problem is a motion perception problem that states the motion direction of a contour is ambiguous, because the motion component parallel to the line cannot be inferred based on the visual input. The Horn-Schunck algorithm assumes smoothness in the flow over the whole image. Thus, it tries to minimize distortions in flow and prefers solutions which show more smoothness. The flow is estimated by the following function:

$$E = \iint \left[(I_x u + I_y v + I_t)^2 + \alpha^2 (\|\nabla u\|^2 + \|\nabla v\|^2) \right] \mathrm{d}x \mathrm{d}y$$

Figure 1: Horn-Schunck optical flow algorithm

For calculating the optical flow, I choose a video of a road camera, capturing traffic. Then, I calculated the optical flow based upon the video. In the program, you can choose the video mode, which will calculate the optical flow for each frame along the video, and show it in real time to the user, or you can choose the image mode, which will capture the first frame of a video, calculate the motion for the next 24 frames and them show the image. You can run the program to see the video mode in action. For the image mode, the result is as follows:

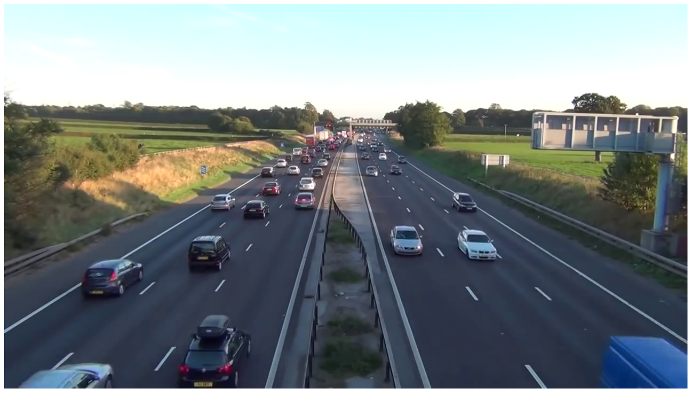

Figure 2: First frame of the video

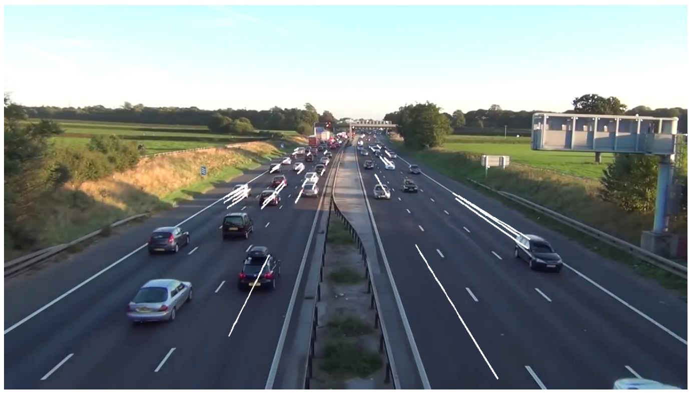

Figure 3: Frame of a second later, with the optical flow drawn

2 Mean-Shift Segmentation in OpenCV

Mean-shift is a analysis technique for locating stationary points in a density function. In Computer Vision application, it depends on the spatial radius, color radius and image pyramid. The spatial radius determine the size of the window in which you will be calculating the mean-shift. It could be a circle, a square, etc. The color radius change the sensibility of the algorithm to color changes.
Regarding the image pyramid, in my opinion the most significant parameter, it determines the number of levels that will be used in the pyramid. Choosing this parameter correctly is crucial, due to the impact on accuracy of the algorithm, and on the execution time.
But, for some cases, is not that trivial to calculate the mean-shift segmentation. This is due usually to a large color radius, that can interfere on the final result.

For instance, this image:

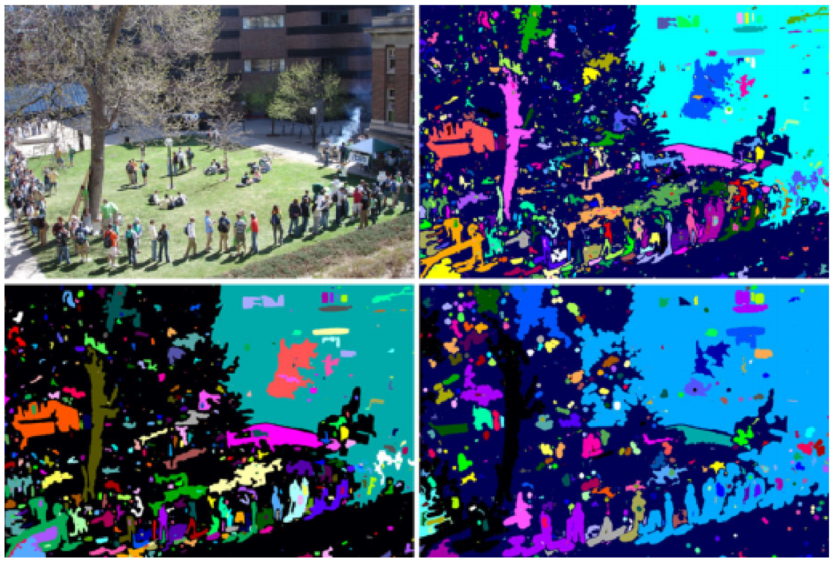

Figure 4: Upper left: The original colour image Spring. Upper right: The segmentation result with a spatial radius = 5 and five iterations using pyramidal Belief Propagation segmentation. Lower left: After 10 iterations, also for spatial radius = 5. Lower right: Also after 10 iterations but for spatial radius = 10. All three mean-shift uses the same color radius

Here we can see how much of a difference the parameters presented before do on the result image.

3 Detection of Calibration Marks

Two major distortions applied to images by cameras are radial distortion and tangential distortion. The radial distortion makes straight lines appear to be curved the further away it is from the center. The tangential distortion occurs due to the camera lens is not aligned parallel to the plane, and make some areas in the image to look nearer than expected.
So, in order to run the calibration, Ramon recorded a video in which he shows a checkerboard pattern on the screen, from several angles. The checkerboard pattern used is 7x9, so in order to detect the inner corners, we need to pass the kind of pattern we are looking at. This pattern consist in how many corners are inside the checkerboard. For any MxN checkerboard, the number of corners inside is (M-1)x(N-1), so in this case the kind of pattern will be 6x8. Once we find the corners, we will increase their accuracy at subpixel level. The detected corners are showned below:

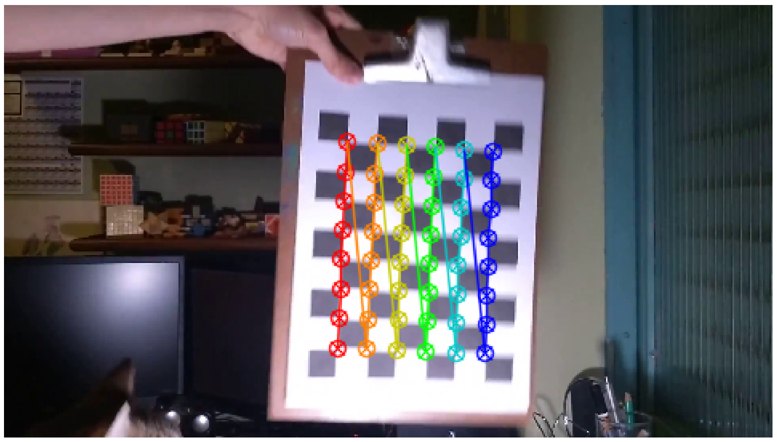

Figure 5: Detected corners, front view

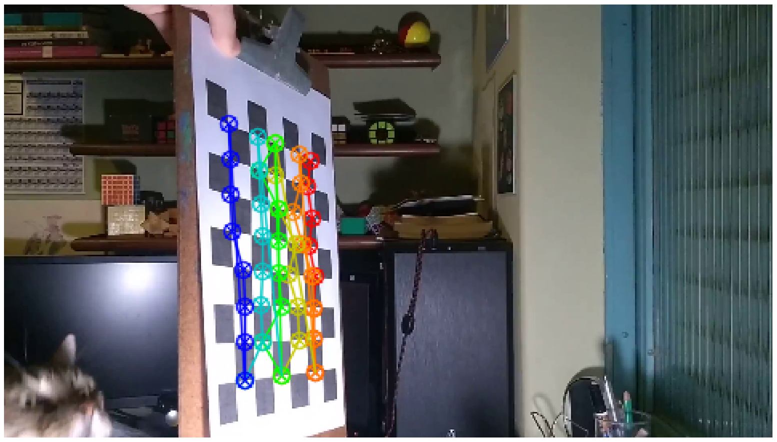

Figure 6: Detected corners, side view

Now that we have our object and image points, we can calibrate the camera. With all the information gathered, we can calculate the camera matrix, distortion coefficients and the rotation and translation vectors. Then, we need only to undistort the image, store the camera matrix and distortion coefficients, and then we can use it later!