

Abdo Babukr • Abaco Systems • Boston Center • 6-1-17

Finding Lane Lines on the Road

In this project we use python and OpenCV tools to identify lane lines on the road. We develop a pipeline on a series of individual images, and later apply the result to a video stream.

Beginning with the original image, my pipeline consisted of 7 steps.



Step 1: Convert the images to grayscale.



Step 2: Apply Gaussian (blur) filter prior to edge detection



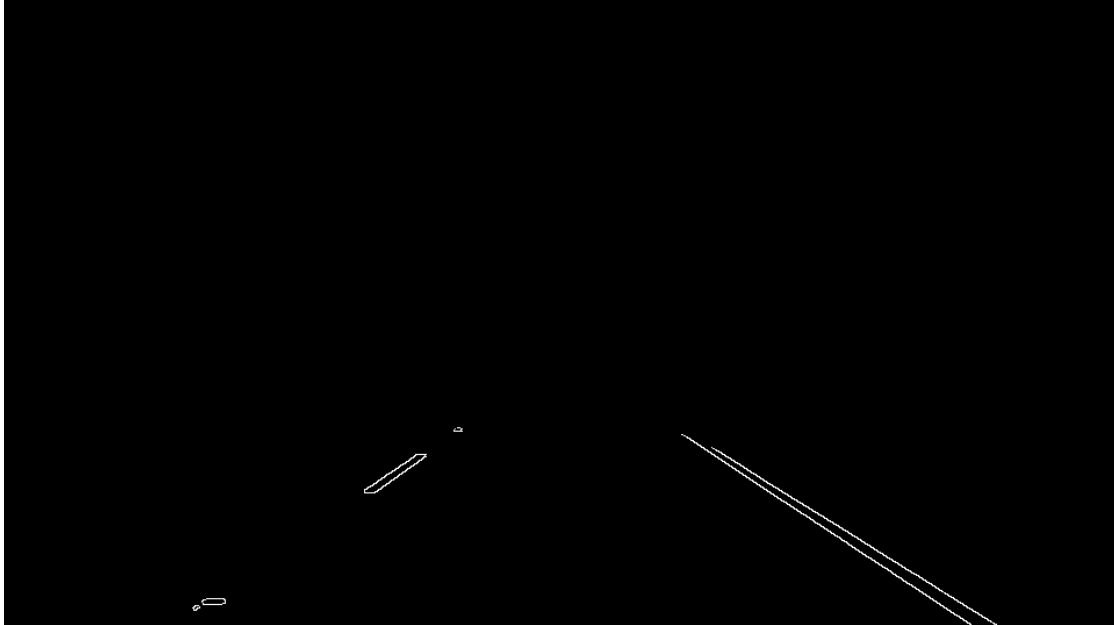
Step 3: Apply canny edge detector (gradient filter with thresholds).



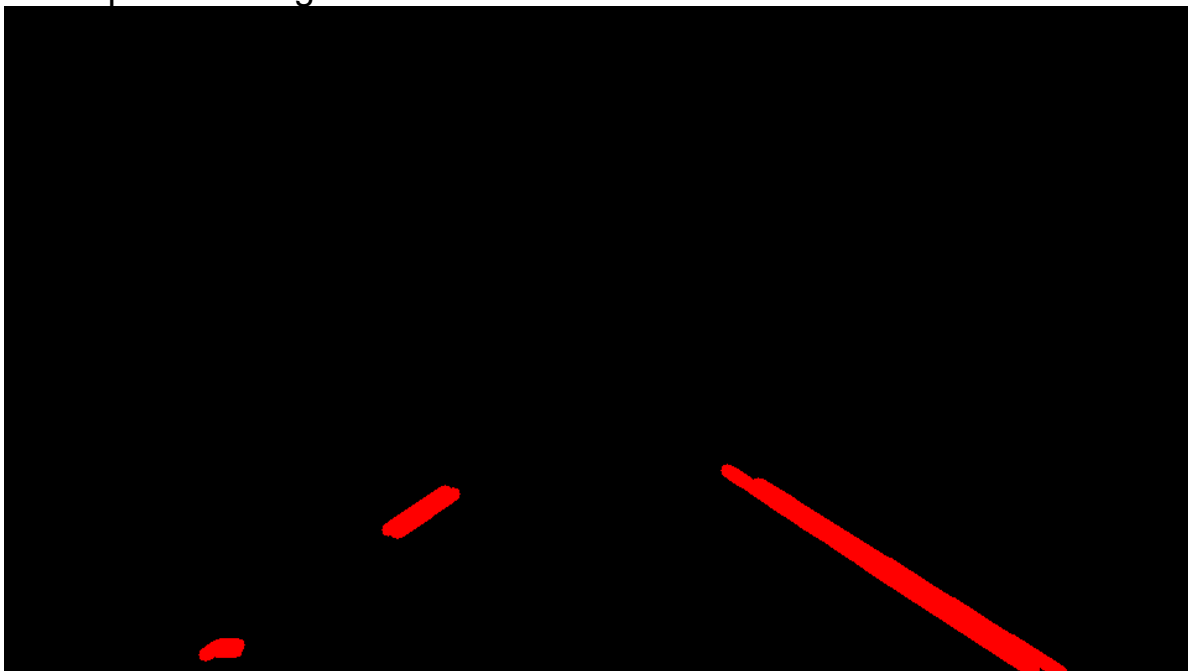
Step 4: Create a masked ROI (region of interest).



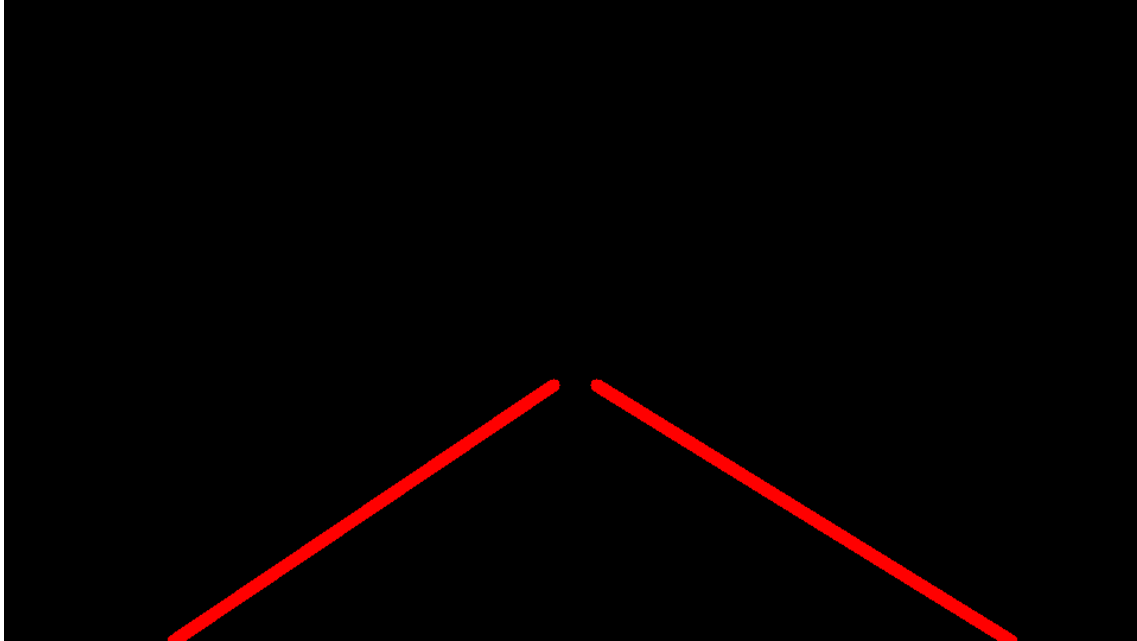
Step 5: Apply the masked ROI to the edge image using binary AND.



Step 6: This is the point where we identify lane lines. We do so using the Hough transform for detecting arbitrary shapes. In this case we detect a line, $y = mx + b$. The goal is to transform the image from the (x, y) space to the (m, b) space. Since the slope is rise/run the run can potentially be zero and therefore the slope cannot be computed. The image thus gets transformed to the hough space where the parameters are rho and theta. The OpenCV houghline function detects a series of lines as shown.



Step 7: Each line has a pair of (x,y) points in image space. Representing them as scatter points we use linear regression to compute a single line in the right side and a single line in the left side.



The original image is fused with the linear regression lines to display the results visually over the lane lines. Apply the pipeline over a video stream we see there can be more challenges to consider such as other lines in the ROI that interfere with the lane lines. In a future project we will apply advanced techniques to overcome these challenges.

