TOBIE YEFFERSON BIYIHA AFOUNG

COMPUTER VISION

ASSIGNMENT 1:

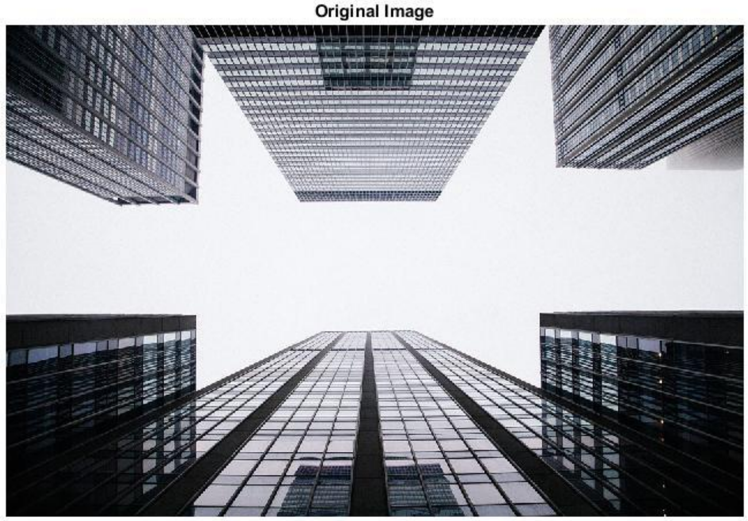
VANISHING POINT DETECTION

ANADOLU UNIVERSITY

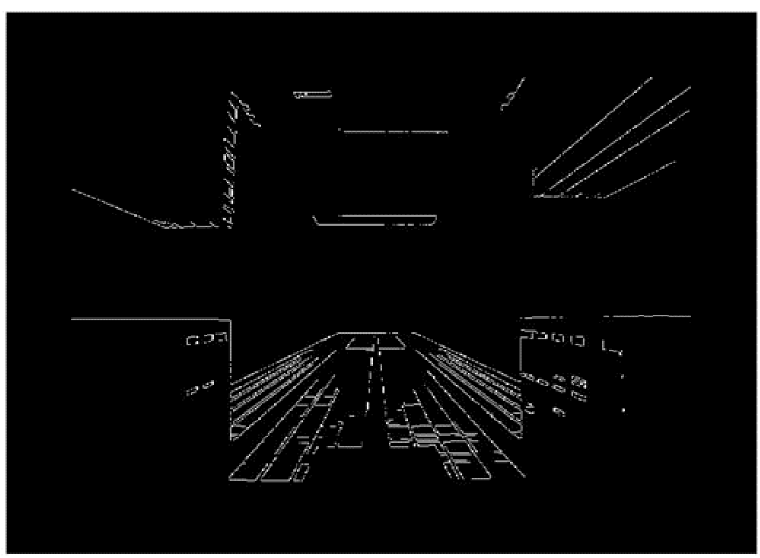
Vanishing point detection is an important aspect in computer vision. The following steps outline algorithm used to estimate the vanishing point in sample images. Note that, in some of these images, the vanishing point was outside of them. As such, extra pixels had to be added to these images in order to be able to view this point.

1. Consider the following image, computed with the following parameters:
   Number of pixels added N: 500, Canny Threshold CT: 0.6, Hough line minimum length HT: 60

Original Image

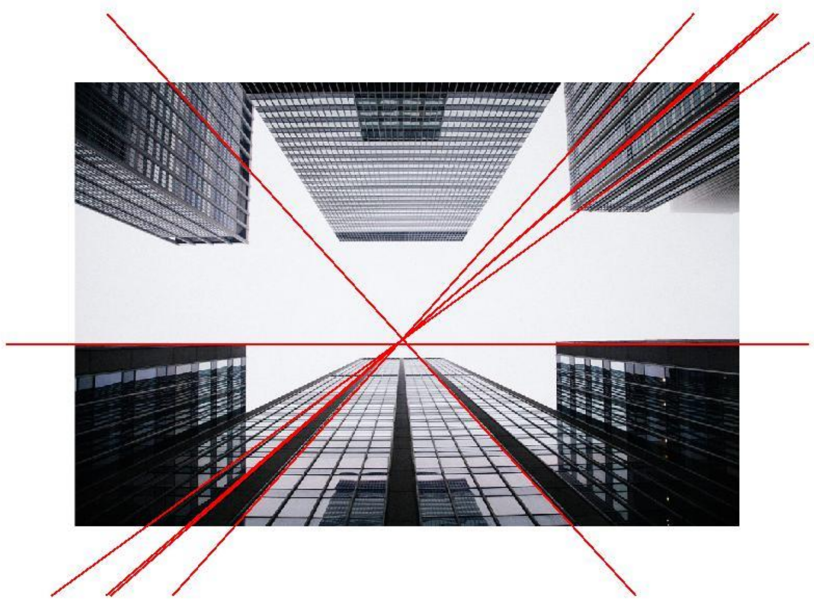


2. *"Canny" edge* detection was applied on the images. For some images, we had to vary the threshold in order to eliminate noisy lines.



3. Hough Transform was used to detect the lines in the image

4. Applying the equation of the line y = mx + c , the lines detected can be extended to the end of the image



5. The vanishing point is supposed to be a point where most lines intercept. Thus we implement an intercept-voting method.
6. **How the *"Intercept-voting method"* works:** For every point where 2 lines meet (intercept), one vote is given. Thus the vanishing point must have the most votes. The main problem with this method is that the intersection point between every 2 lines vary slightly. Thus to fixe to this issue, we rounded

up the intersect point between 2 lines to the nearest 10th, maximizing the chances of getting a point with most lines passing through it. Therefore detecting the vanishing point (**green dot**).

7.

interX | interX_rd

14x2 double

	1	2
1	715.4865	627
2	730.3147	613.9371
3	722.4780	620.8408
4	723.1820	620.2206
5	716	627
6	734.7603	610.7150
7	722.7870	621.1085
8	723.4588	620.5253
9	712.2911	627
10	729.5882	627
11	729.3394	627
12	721.7087	620.1744
13	722.5719	619.5488
14	728.4215	625.9893

Intersection points

interX_rd | interX

14x2 double

	1	2
1	720	630
2	730	610
3	720	620
4	720	620
5	720	630
6	730	610
7	720	620
8	720	620
9	710	630
10	730	630
11	730	630
12	720	620
13	720	620
14	730	630

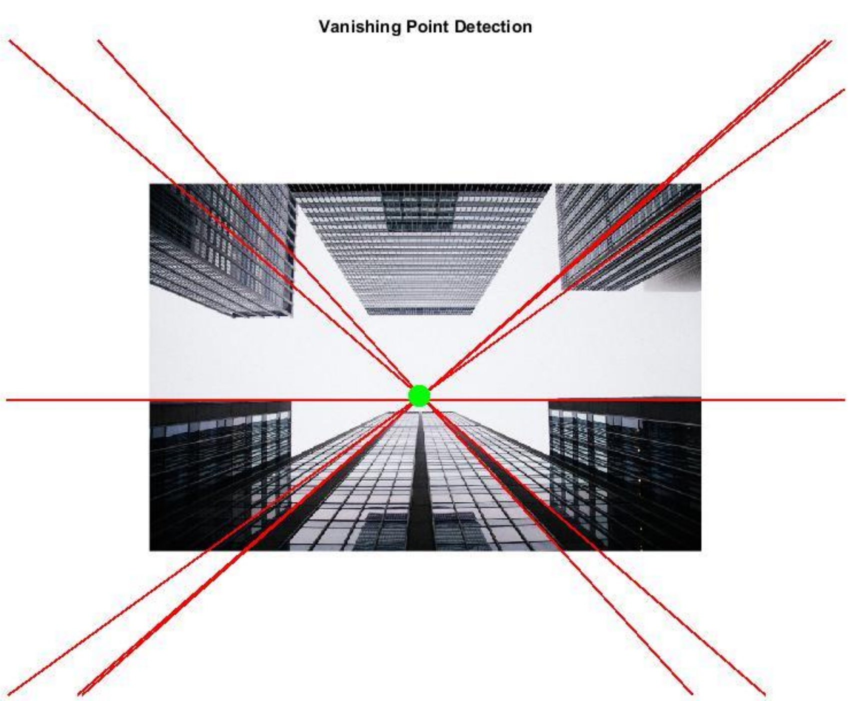
Rounded intersection points

vanishingPoint | interX | interX_rd

1x2 double

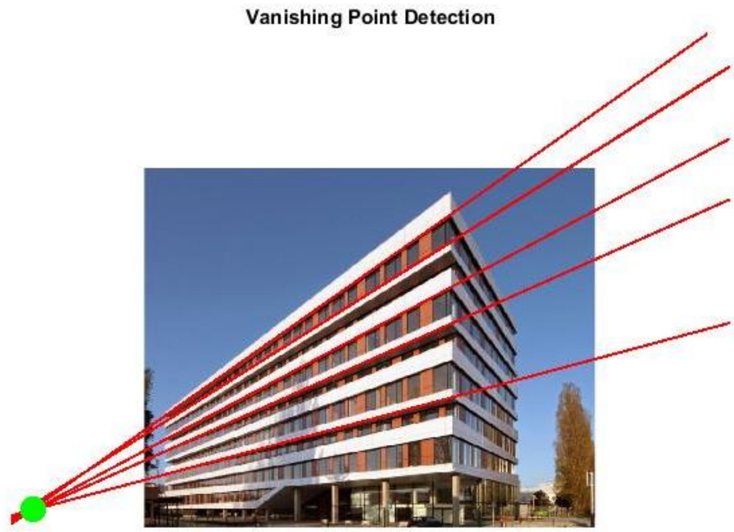
1	2	3	4
720	620		

Vanishing Point after voting

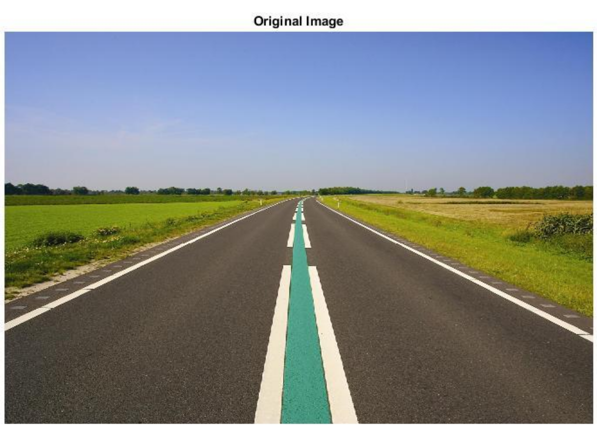


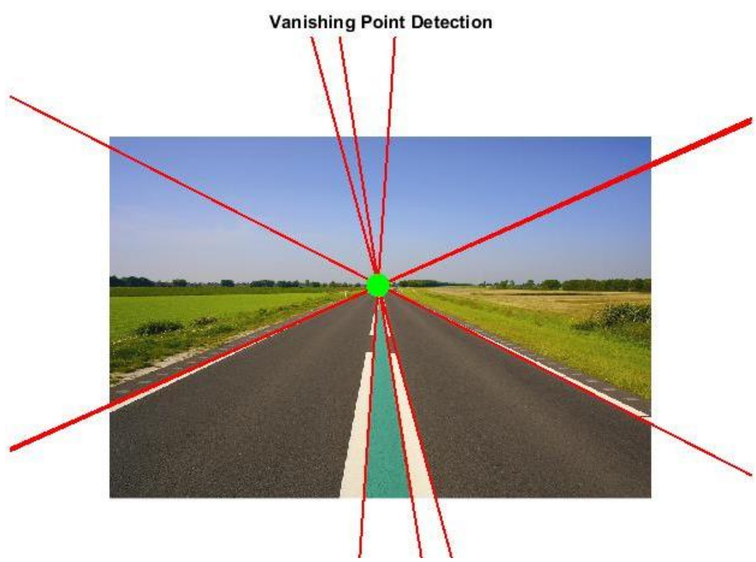
Vanishing Point plotted on image (Green Dot)

1. Now let’s consider other images:
2. Images 2: Number of added pixels N: 600, Canny Threshold CT: 0.6, Hough line mini length HT: 60

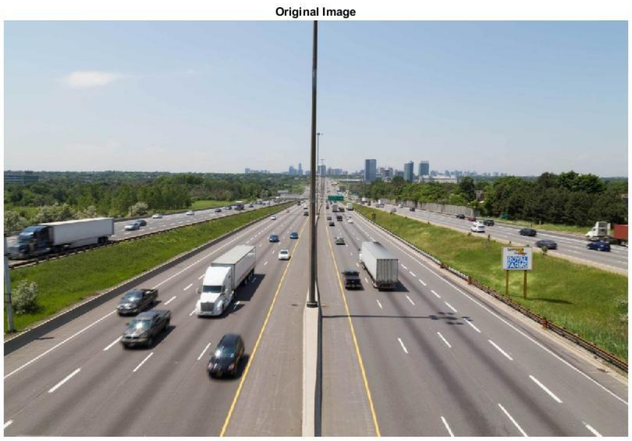


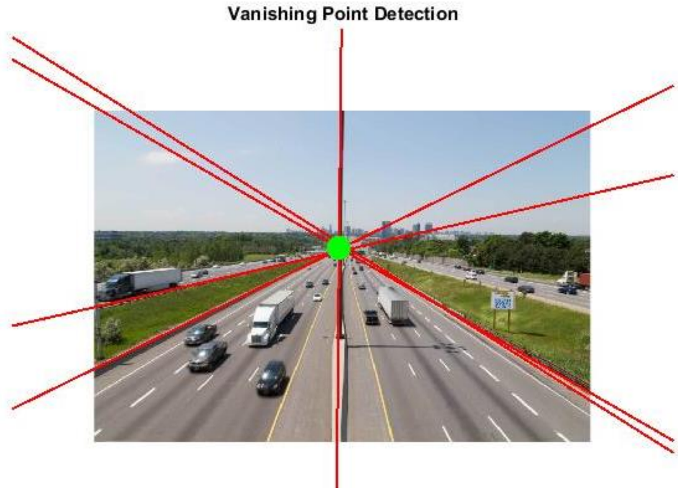
3. Images 3: Number of added pixels N: 500, Canny Threshold CT: 0.6, Hough line mini length HT: 60



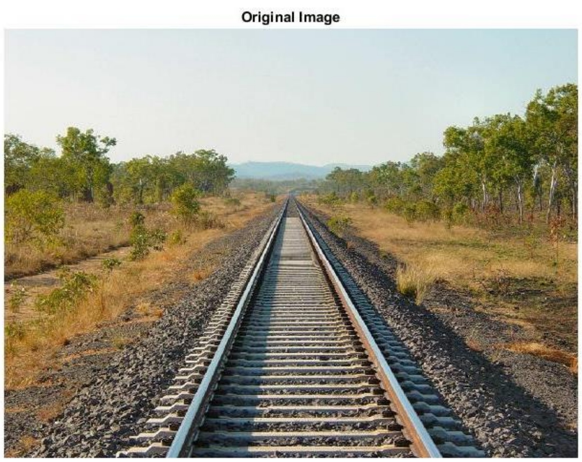


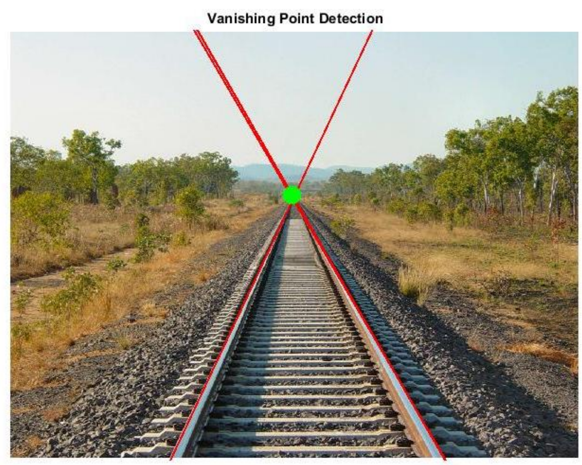
4. Images 4: Number of added pixels N: 400, Canny Threshold CT: 0.6, Hough line mini length HT: 60



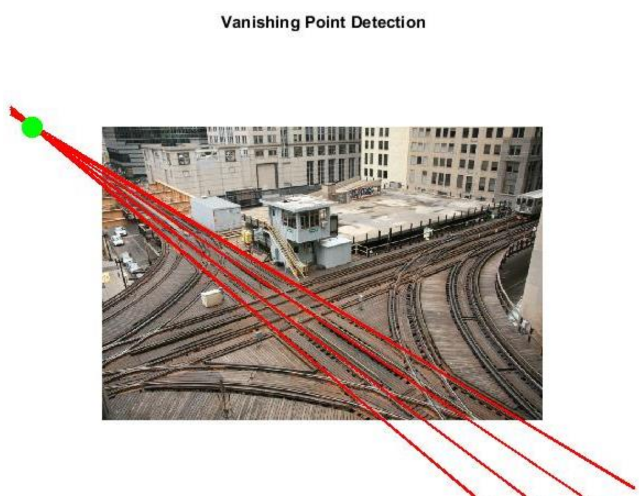


5. Images 5: Number of added pixels N: 10, Canny Threshold CT: 0.9, Hough line mini length HT: 80





6. Images 6: Number of added pixels N: 500, Canny Threshold CT: 0.6, Hough line mini length HT: 60



Conclusion:

The algorithm works fairly well for most images. Nevertheless its performance is greatly affected because it depends on a weak Hough transform to detect the lines in the image.

vanishingPointEstimation.m is the MATLAB code for the above algorithm.