Detecting Lines and Vanishing Points

Yongxin Wang
March 11, 2017

1 CANNY EDGE DETECTIOR

(a) img10

(b) img26

(c) img46

(d) img66

(e) img95

Figure 1.1: Results of Edge Detection. $minVal = 100, maxVal = 150$;

The Canny edge detector parameters we choose are $minVal = 100; maxVal = 150;$. When the $minVal$ is too low or $maxVal$ is too low, there are too many noises in the image. And when $minVal$ is too high or $maxVal$ is too high, there are too few edges detected.

(a) 100,120

(b) 70, 150

(c) 100,200

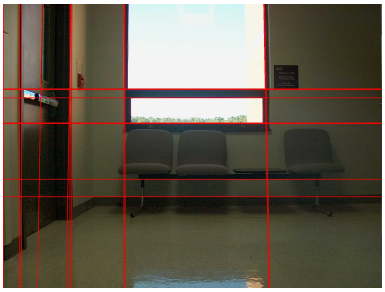
(d) 120, 150

(e) 100, 150; our setting

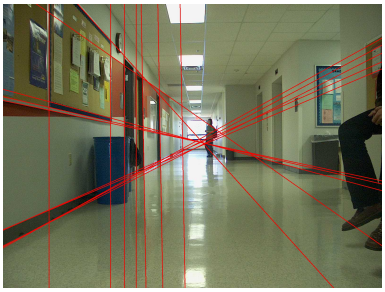
Figure 1.2: Results of Edge Detection. $minVal = 100, maxVal = 150$;

2 Standard Hough Transform

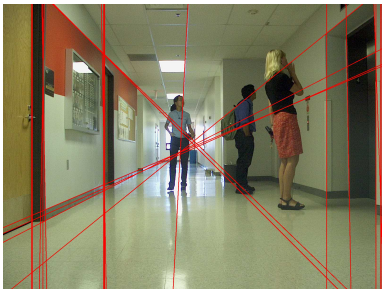
The Standard Hough Transform parameters we choose are $minVote = 230$;. When the $minVote$ is low, many "lines" detected by the detectors are not real lines. And when $minVote$ is high, there are too few lines candidate lines.



(a) img10

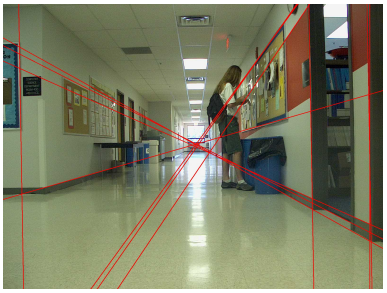


(b) img26



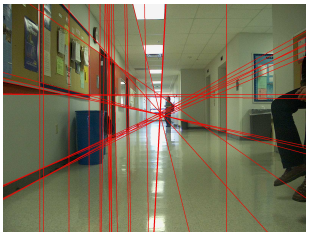
(c) img46

(d) img66

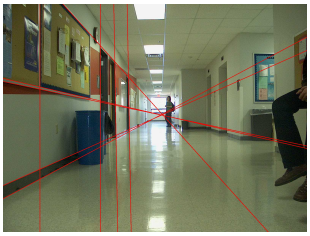


(e) img95

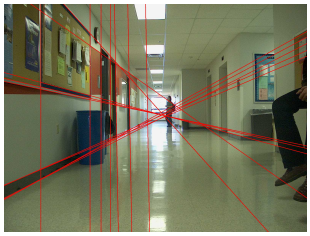
Figure 2.1: Results of Standard Hough Transform. $minVote = 230$



(a) $minVote = 200$

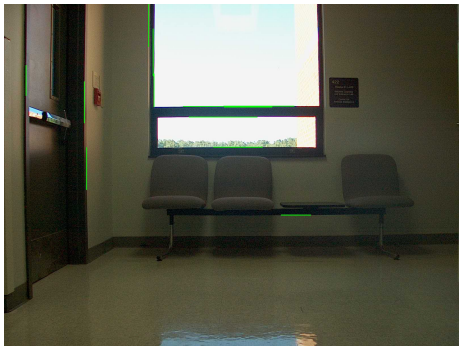


(b) $minVote = 270$

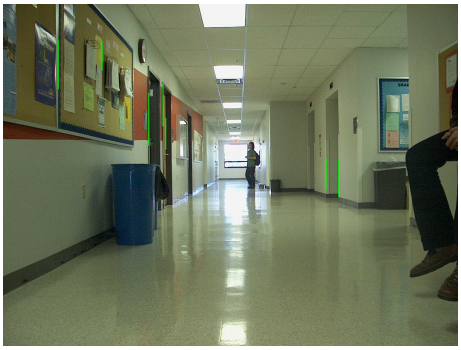


(c) $minVote = 230$, our setting

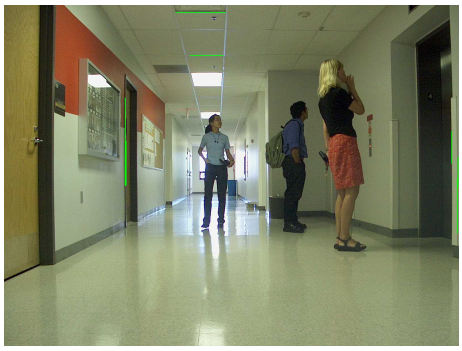
Figure 2.2: Results of Standard Hough Transform



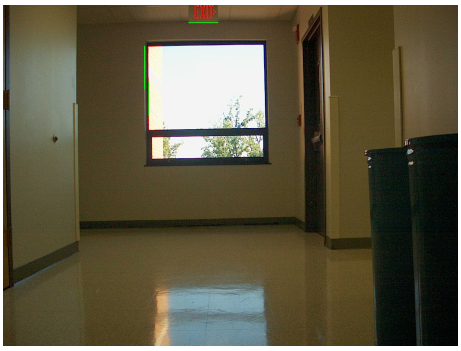
(a) img10



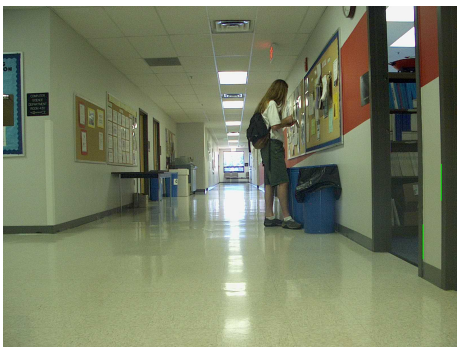
(b) img26



(c) img46



(d) img66

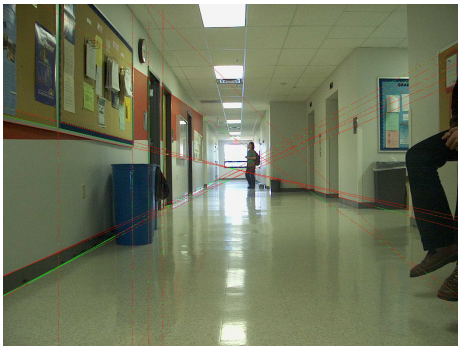


(e) img95

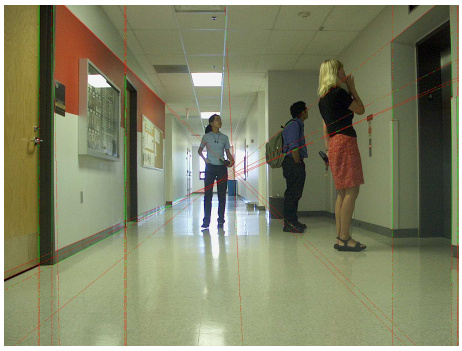
Figure 3.1: Results of Probabilistic Hough Transform

The Probabilistic Hough Transform parameters we choose are $minLineLength = 20$, $maxLineGap = 100$. When "$minLineLength$" is high, too many line segments are detected. When "$maxLineGap$" is high, some no-lines are detected as lines.. The Probabilistic Hough Transform uses sampled points to find lines, so it finds fewer lines than Standard Hough Transfrom. And it outputs line segments other than standard one.

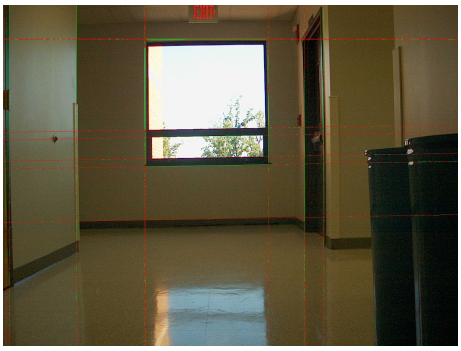
(a) img10



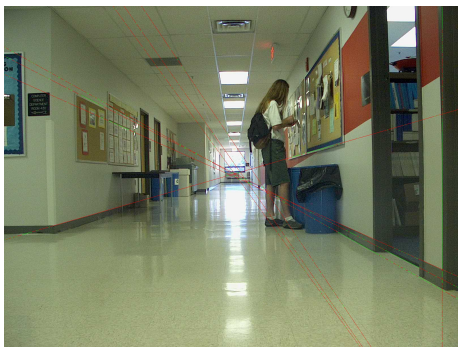
(b) img26



(c) img46



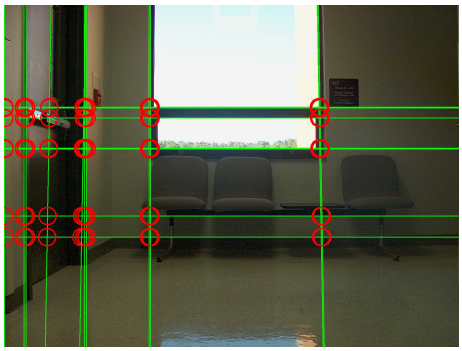
(d) img66



(e) img95

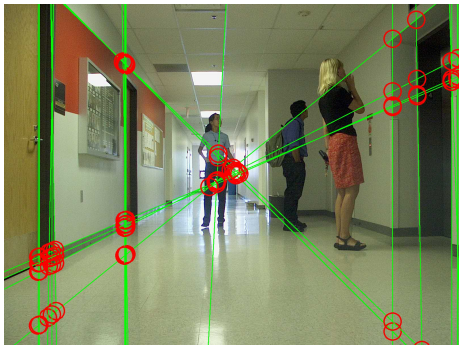
Figure 4.1: Results of Filterd Hough Transform

Filter the hough lines given gradient magnitude and orientation. The parameters we choose are $minGrad = 0.03, maxOriDiff = 0.2$. These parameters have the best results of detecting real lines. We use Bresenham algorithm in both x and y direction accordingly. And based on the gradient and orientation image we filter all the points that don't satisfy the "$minGrad = 0.03, maxOriDiff = 0.2$" requirement.

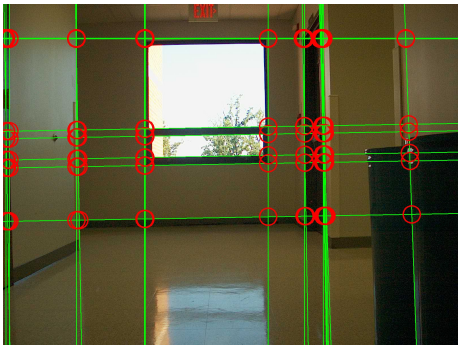


(a) img10

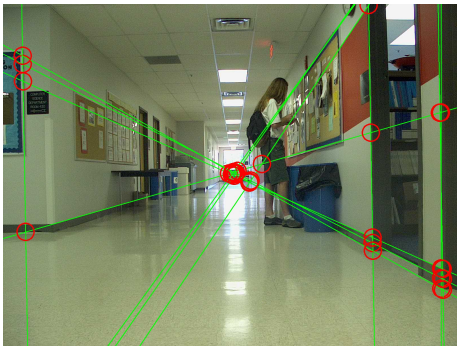
(b) img26



(c) img46



(d) img66

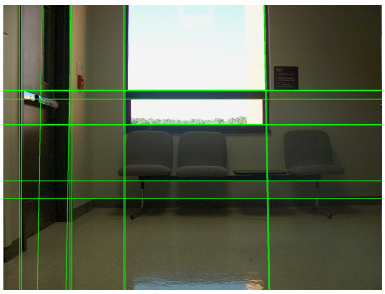


(e) img95

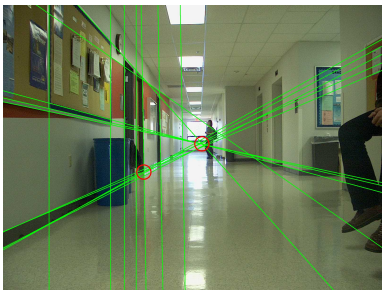
Figure 5.1: Results of Vanishing Points from All Pairs

Check every pairs of lines from the Hough Line Transform results. And in addition we store the orientations of the two lines together with intersection points for the next step("True vanishing ponits").

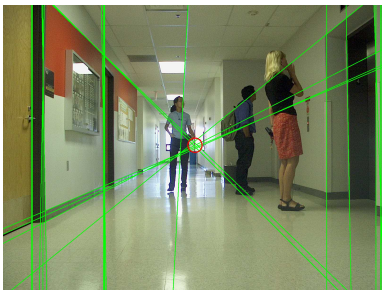
6 VANISHING POINTS FROM ALL LINES



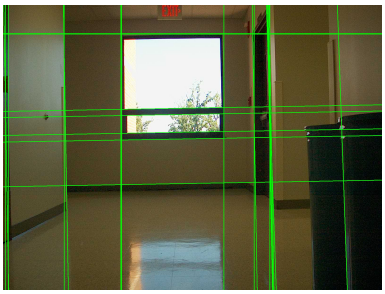
(a) img10



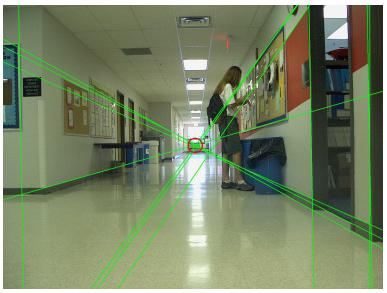
(b) img26



(c) img46



(d) img66



(e) img95

Figure 6.1: Results of Vanishing Points from All Lines

1. Find all the groups of intersection points with the number of points over "$minNum$" which are within the distance of "$maxDis$".
2. Use k-means to cluster all intersection points from step1.
3. Merge the centers from step 2 which are close enough to each other.