# theory

September 16, 2021

[ ]:

```
```

## 0.1 Hough Transform - Assignment 1

`Instructor: Kris                              TAs: Rawal (Lead), Yan, Zen, Wen-Hsuan, Paritosh, Qich`

In this assignment you will be implementing a Hough Transform based line detector.

---

## 0.2 Theory Questions (25 points)

**Grading**: - Each question is 5 points. - Please add your answers to the writeup. Insert images whenever necessary. - Show all your work to obtain full credit.

**Q1**: Show that using $x\cos\theta + y\sin\theta - \rho = 0$, each image point $(x, y)$ results in a sinsuoid in $(\rho, \theta)$ hough space.
Use the formulation, $a\sin\theta + b\cos\theta = A\sin(\theta + \phi)$ for the sinsuoid.
Write the amplitude $A$ and phase $\phi$ of the sinusoid as a function of $(x, y)$.

Consider the following formula

$$a\sin(\theta) + b\cos(\theta) = A\sin(\theta + \phi) \tag{0}$$

Expanding the RHS

$$a\sin(\theta) + b\cos(\theta) = A\cos(\phi)\sin(\theta) + A\sin(\phi)\cos(\theta) \tag{1}$$

Thus, on comparing RHS and LHS, we get

$$a = A\cos(\phi) \tag{2}$$
$$b = A\sin(\phi) \tag{3}$$

Divding (3) by (2), we get

$$\frac{b}{a} = \frac{\sin(\phi)}{\cos(\phi)} = \tan(\phi)$$

$$\therefore \phi = \tan^{-1}(\frac{b}{a}) \tag{4}$$

Squaring and adding (2) and (3)

$$a^2 + b^2 = A^2\cos^2(\phi) + A^2\sin^2(\phi) = A^2[\cos^2(\phi) + \sin^2(\phi)] = A^2$$

$$\therefore A = \sqrt{a^2 + b^2} \tag{5}$$

Now, consider the normal form of line equation at any point $(x_1, y_1)$ on the line

$$x_1\cos(\theta) + y_1\sin(\theta) = \rho$$

Based on equation (0), we can change this line equation to

$$A\sin(\theta + \phi) = \rho \tag{6}$$

where, using (3) and (2) respectively,

$$x_1 = A\sin(\phi)$$
$$y_1 = A\cos(\phi)$$

Using (4), we get

$$\phi = \tan^{-1}(\frac{x_1}{y_1}) \tag{7}$$

Using (5), we get

$$A = \sqrt{x_1^2 + y_1^2} \tag{8}$$

Thus, from (6), (7) and (8), we can see a point $(x_1, y_1)$ in the normal form of line equation forms a sinusoid ($\rho = A\sin(\theta + \phi)$) in parameter space with an amplitude, $A = \sqrt{x_1^2 + y_1^2}$ and phase, $\phi = \tan^{-1}(\frac{x_1}{y_1})$.

**Q2**: Why do we parameterize the line in terms of $\rho, \theta$ instead of slope and intercept $(m, c)$?
Also write the slope $m$ and intercept $c$ as a function of $\rho$ and $\theta$.

m (slope) ranges from -∞ to ∞. c (y-intercept) too has a huge bound. With both the parameters having such huge bounds, the accumulator array (H) would have to be huge. Therefore, very high memory usage and computation cost. Also, performing computation for every combination of m and c would not be possible and thus, we might miss out on some important values.
However, parameterizing in terms of $\rho$ and $\theta$ is better as they have finite range. $0 \leq \rho \leq \rho_{max}$ and

$0 \leq \theta \leq 2\pi$. Thus, calculating for all combinations would be possible with less computation and memory cost.

Writing m and c as function of $\rho$ and $\theta$ -
$x\cos(\theta) + y\sin(\theta) = \rho$

Rearranging above equation
$y\sin(\theta) = (-\cos(\theta))x + \rho$

Dividing both sides by $\sin(\theta)$
$y = (-\frac{\cos(\theta)}{\sin(\theta)})x + \frac{\rho}{\sin(\theta)}$

Therefore,

$m = -\frac{\cos(\theta)}{\sin(\theta)}$ and $c = \frac{\rho}{\sin(\theta)}$

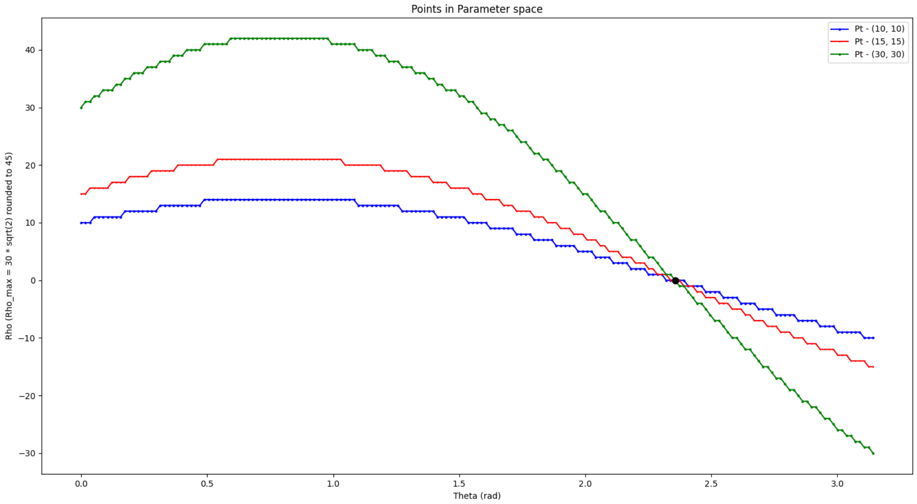
**Q3**: Assume the image points $(x, y)$ are in an image of width $W$ and height $H$, $x \in [1, W], y \in [1, H]$.
What is the maximum absolute value of $\rho$ and what is the range of $\theta$?

$\rho$ is the distance of the line from the origin. Thus, the farthest line possible would be at a distance same as the length of the diagonal.

Therefore, $|\rho_{max}| = \sqrt{W^2 + H^2}$

If we consider the range of $\rho$ to be positive - $0 \leq \rho \leq \rho_{max}$, then $\theta$ lies in the range - $0 \leq \theta \leq 2\pi$. If we allow $\rho$ to take negative values - $-\rho_{max} \leq \rho \leq \rho_{max}$, then $\theta$ lies in the range - $0 \leq \theta \leq \pi$.

**Q4**: For points $(10, 10)$, $(15, 15)$ and $(30, 30)$ in the image, plot the corresponding sinusoid waves in Hough space $(\rho, \theta)$.
Also visualize how their intersection point defines the line (what is $(m, c)$ for this line?).
Please add the plot as image.



The sinusoids intersect at ($\rho$, $\theta$) = (0, 2.3561945). This is shown by the black dot in the above plot.

By rearranging the normal form of line, we get

$$y = (-\frac{\cos(\theta)}{\sin(\theta)})x + \frac{\rho}{\sin(\theta)}$$

Therefore,

$$m = -\frac{\cos(\theta)}{\sin(\theta)} \text{ and } c = \frac{\rho}{\sin(\theta)}$$

Using the above equations,
$m = -\frac{\cos(2.3561945)}{\sin(2.3561945)} = 1.0000000196153103 \approx 1$

$$c = -\frac{0}{\sin(2.3561945)} = 0$$

Thus, the single intersection point helps us understand that all 3 points lie on the same line. Using the corresponding $\rho$ and $\theta$, we can calculate the slope (m) and y-intercept (c).

Therefore, the line is -
$y = 1.x + 0 = x$

**Q5**: How does the dimension of parameter space affects Hough Transform method? What would you do when the parameter space is high, i.e., 3D or 4D instead of 2D? Briefly explain your method.

As the dimension of the parameter space increases, the dimensions of the corresponding accumulator array of the Hough Transform algorithm increases as well. As the parameter space becomes 3-D, the accumulator needed is a 3-D array. Similarly for 4-D and so on. So, as the dimension of the parameter space increases, the overall computation and memory requirement increases. Thus, the runtime for Hough Transform method increases, making it slower.

Consider the equation of a circle of radius r and centered at (a, b) -

$(x - a)^2 + (y - b)^2 = r^2$

Here the parameter space would be 3-D - a, b, r. A circle in the cartesian coordinate system would be a point in the Parameter space.

Consider 2 points on that circle - (x1, y1) and (x2, y2). There could be an infinite number of circles passing through this points with varied a, b and r.

Therefore, in the parameter space, this point (x1, y1) would be represented by

$(a - x1)^2 + (b - y1)^2 = r^2$ where a,b,r are variables.

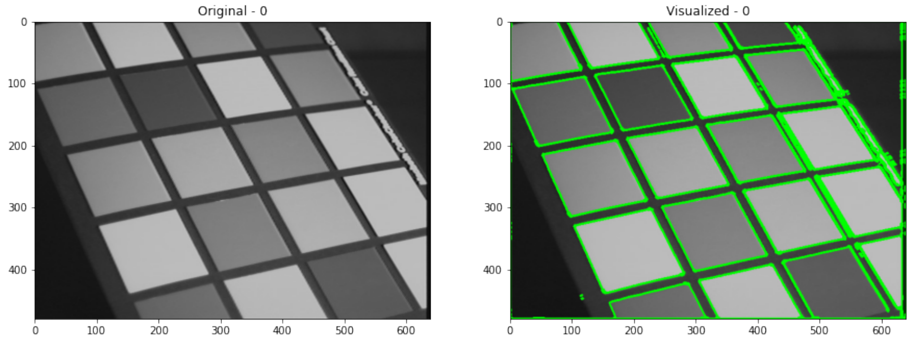
Similarly (x2, y2) would be represented by

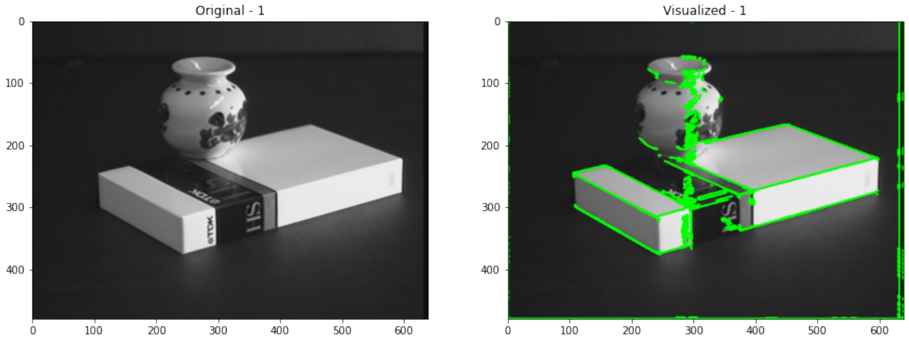
$(a - x2)^2 + (b - y2)^2 = r^2$ where a,b,r are variables.

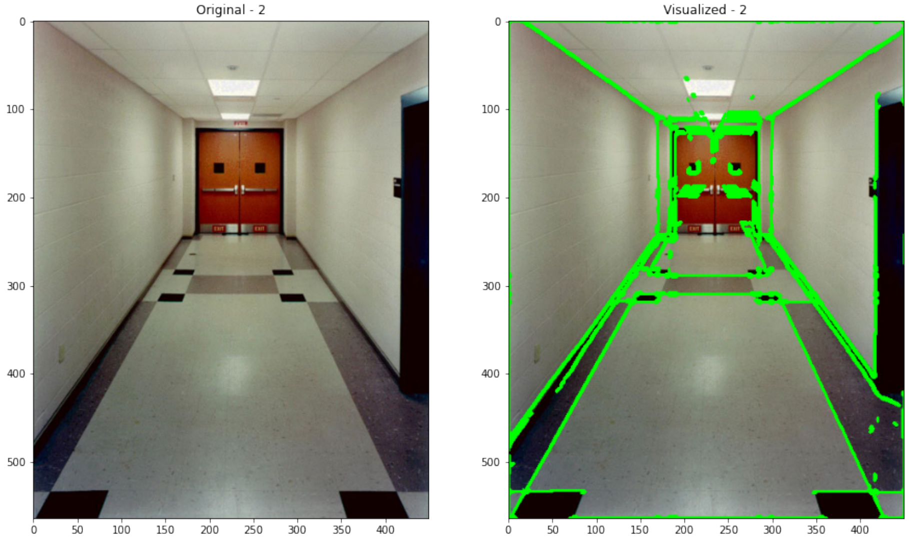
In order to solve this, we would create a 3-D accumulator with some fixed range of a, b, r along the 3 dimensions of the array. We could loop over every combination of a and b in the accumulator and calculate $r'$. We could then round this off to the nearest accumulator r value and increment its vote count.

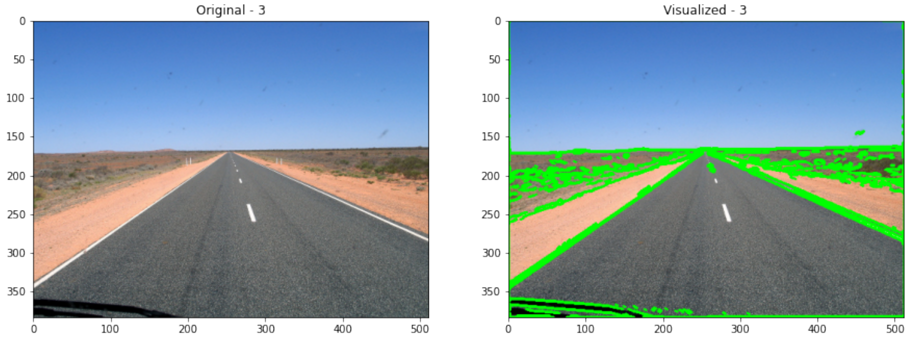
We'll do this for both the points and the a,b,r combination with the highest vote (in this case 2) would represent the a, b, r for the circle containing both these points.

**Visualizations from P4.2**

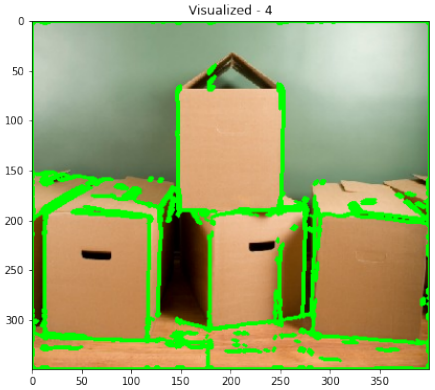



Original - 1
Visualized - 1
0
100
200
300
400
0
100
200
300
400
500
600
©TDK
SIIL

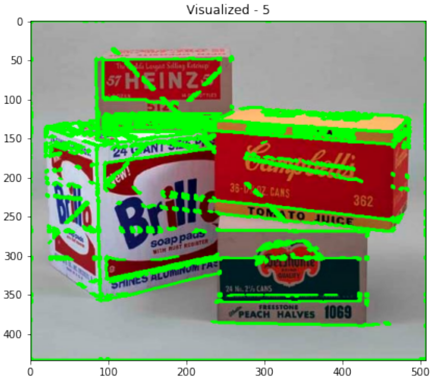

Original - 2
Visualized - 2
0
100
200
300
400
500
0
50
100
150
200
250
300
350
400
EXIT


Original - 3
Visualized - 3
0
50
100
150
200
250
300
350
0
100
200
300
400
500

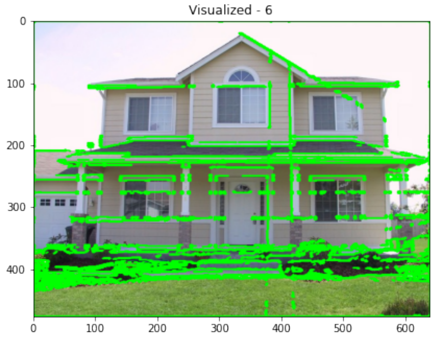
Original - 4
0
50
100
150
200
250
300
0
50
100
150
200
250
300
350

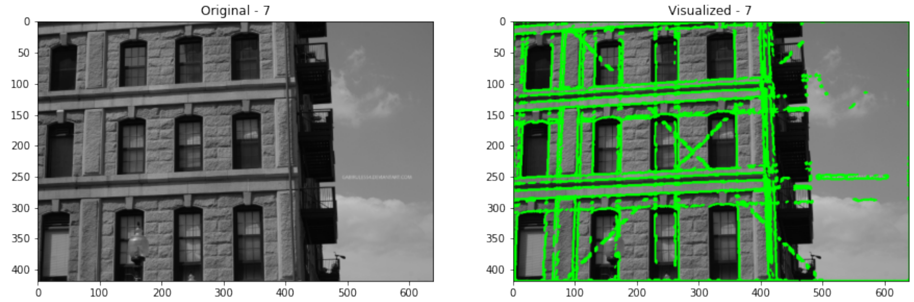

Visualized - 4
0
50
100
150
200
250
300
0
50
100
150
200
250
300
350

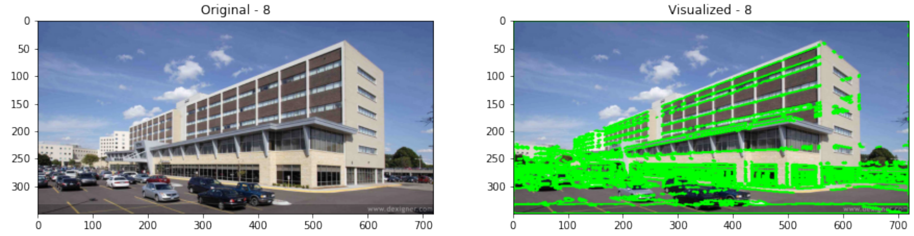
Original - 5
0
50
100
150
200
250
300
350
400
0
100
200
300
400
500

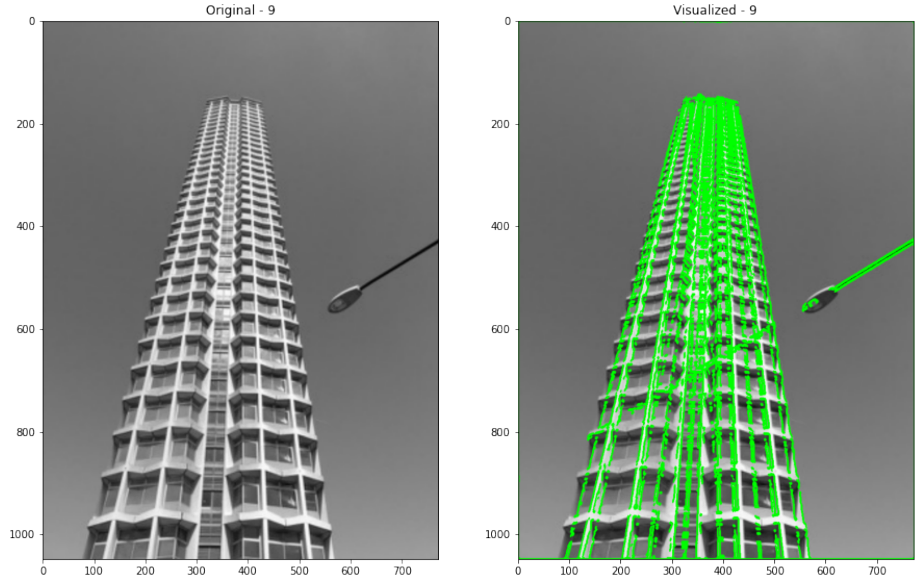

Visualized - 5
0
50
100
150
200
250
300
350
400
0
100
200
300
400
500

Original - 6
0
100
200
300
400
0
100
200
300
400
500
600


Visualized - 6
0
100
200
300
400
0
100
200
300
400
500
600


Original - 7
Visualized - 7
0
50
100
150
200
250
300
350
400
0
100
200
300
400
500
600


Original - 8
Visualized - 8
0
50
100
150
200
250
300
0
100
200
300
400
500
600
700


Original - 9
Visualized - 9
0
200
400
600
800
1000
0
100
200
300
400
500
600
700

```
[ ]:
```