1.

a) A corner is a local place in the image where the gradient vectors take more than one orientation.

The steps to detect a corner in a local window are:

1) Find the correlation matrix in the window
2) Compute eigenvalues of the matrix
3) Check if $\lambda_1, \lambda_2 > \tau$

b) We want find the direction that minimizes the projection of the points. PCA is a statistical procedure that uses an orthogonal transformation to convert a set of observation of possibly correlated variables into a set of linearly uncorrelated variables. The number of principal components is equal or less than the smaller number of original values.

c) $$C = \sum_{i=1}^{n} P_i P_i^t = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \begin{bmatrix} x_i & y_i \end{bmatrix} = \begin{bmatrix} \sum x_i^2 & \sum x_i y_i \\ \sum x_i y_i & \sum y_i^2 \end{bmatrix} = \begin{bmatrix} 4 & 6 \\ 6 & 44 \end{bmatrix}$$

d) We can use $\lambda_1 \cdot \lambda_2 > \tau$ or $\lambda_1 \cdot \lambda_2 - k(\lambda_1 + \lambda_2)^2 > \tau$

e) 1. Compute the eigenvalues λ_1, λ_2 for the correlation matrix
2. Sort the pixels based on eigenvalues
3. Start from the top selecting the highest value.
4. Delete corners in vecinity of selected corner
5. Stop when you have selected a percentaje of the total.

f) Because it use $C(G) = \det(G) - k \cdot tr^2(G)$ where $\det(G) = \lambda_1 \cdot \lambda_2$ and $tr(G) = \lambda_1 + \lambda_2$

g) Project the points onto the edges hypothesis and choose the one with minimal projection. $\sum_i \nabla I(P_i) \nabla I(P_i)^T P_i = \sum_i \nabla I(P_i) \nabla I(P_i)^T P$

h) Local object appearance and shape in an image can be described by the distribution, intensity of gradients or edge directions.

To perform HOG:

1. Divide image in windows.
2. Compute a histogram of gradient direction for each pixel in each window
3. Compare HOG of each window and concatenate the histograms.

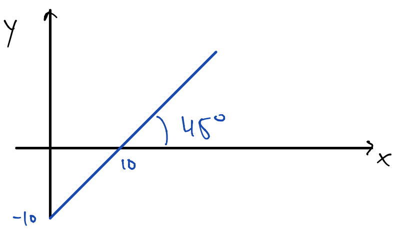
As it operates in local windows it is invariant to geometric and photometric transformations but not to object orientation.

i) 1. Create internal representations of the original image to ensure scale invariance
2. Use LOG to find keypoints
3. Eliminate bad keypoints (edges, low contrast regions, ...)
4. Compute an orientation for each keypoint,
5. Finally, scale and rotation invariance in place

2.

a) If you use the slope and y-intercept to calcute a and b in $y = ax + b$ you obtain values $a \in [-\infty, \infty]$ and $b [-\infty, \infty]$ what it's a problem and that is why the implicit line equation is used instead.

b)



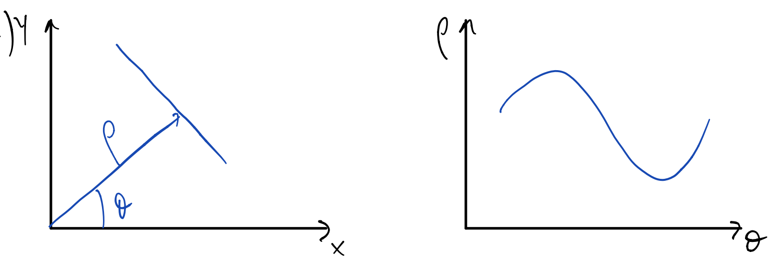
$ax + by + c = 0$

$x = 10;\ y = 0 \rightarrow 10a + c = 0$
$x = 0;\ y = -10 \rightarrow -10b + c = 0$
$\Big\}\ b = -a$

$a = 10;\ b = -10;\ c = -100$

$10x - 10y - 100 = 0$

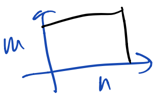
c)



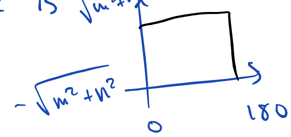
d) Each point defines a line in the parameter space. Points on the same line in the x, y space define lines. in the parameter space which all intercept in one point. This point define the parameters for the line in x, y space we are looking for.

The algorithm is as follows:

1. Detect edges
2. Map edge points to Hough space and store it.
3. Yield stored points in lines of infinite lenght
4. Convert infinite lines to finite ones
5. Find interception

e) Instead of be $m \times n$ 

the bin size in the parameter plane is $\sqrt{m^2+n^2}$



f) Instead of scanning from 0 to 180 we can scan from $\theta - \Delta$ to $\theta + \Delta$

g) The parameter space is tridimensional (a, b, r)

3.

a) It doesn't fit all data with the same precision. Lines with big slopes cannot be well fitted.

b) $\begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix} [x \;\; y \;\; 1] \Rightarrow x + 2y + 2 = 0$

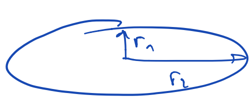
c) $E(l) = \sum_{i=1}^{n} (l^t x_i)^2$. We have to solve $\nabla E(l) = 0$. We have to compute the data matrix and fit the points to the equation.

d) $$\begin{bmatrix} \sum x_i^2 & \sum x_i y_i & \sum x_i \\ \sum x_i y_i & \sum y_i^2 & \sum y_i \\ \sum x_i & \sum y_i & n \end{bmatrix} = \begin{bmatrix} 5 & 15 & 3 \\ 15 & 46 & 10 \\ 3 & 10 & 3 \end{bmatrix}$$

e) $ax^2 + bxy + cy^2 + dx + ey + f = 0$

To be an ellipse: $b^2 - 4ac < 0$

f) $$\left.\begin{aligned} & f(x^*) = 0 \\ & (p - x^*)^t \begin{pmatrix} -df/dy \\ df/dx \end{pmatrix} = 0 \end{aligned}\right\}$$ We need to solve these equations



$\frac{d_1}{d_1 + r_1} > \frac{d_2}{d_2 + r_2}$

Points close to the short axis affect more the fitting

g) $E(l) = \sum_{i=1}^{n} \frac{|f(P_i, l)|}{|\nabla f(P_i, l)|}$

h) $E[\phi(s)] = \int_{\phi(s)} (\alpha(s) E_{continuity} + \beta(s) E_{curvature} + \gamma(s) E_{image})\, ds$

$E_{continuity}$, $E_{curvature}$, E_{image} are energy terms

$\alpha(s)$, $\beta(s)$, $\gamma(s)$ are coefficients

$\alpha(s) \cdot E_{continuity} + \beta(s) E_{curvature}$ are internal parameters

$\gamma(s) E_{image}$ is external parameter

i) Continuous space:

$E_{continuity} = \left| \frac{\partial \phi}{\partial s} \right|^2$

$E_{curvature} = \left| \frac{\partial^2 \phi}{\partial s^2} \right|^2$

$E_{image} = -|\nabla I|^2$

Discrete space

$E_{continuity} = \sum_{i=1}^{n} |P_i - P_{i-1}|^2$

$E_{curvature} = \sum_{i=1}^{n} |(P_{i+1} - P_i) - (P_i - P_{i-1})|$

$E_{image} = \sum |P_{i+1} - 2P_i - P_{i-1}|^2$

j) Using a threshold if $|P_{i+1} - 2P_i + P_{i-1}| > \tau \Rightarrow \beta = 0$