CS 512 - HW 3
SUDIPTA SWARNAKAR
CWID: A20377210

1)

a)

Corner Detection:

A corner can be defined as a point for which there are two dominant and different edge directions in a local neighborhood of the point.
So corners are can be found as the local image features characterized by locations where variations of intensity function f(x, y) in both X and Y directions are high.

b)

PCA to Find Principle Direction of Gradient Orientation:

PCA computes a new orthogonal base given a multi-dimensional data set such that the variance of the projections on the axes of this new base is subsequently maximized. It turns out that the base is formed by the eigenvectors of the auto-covariance matrix.
Applying PCA to the auto-covariance matrix of the [Gx Gy]^T gradient vectors provides the 2-dimensional Gaussian joint probability density function of these vectors. From this function, the main direction of the gradients can be calculated

c)

c) The correlation matrix can be given as,

$$M = \begin{bmatrix} \sum x_i^2 & \sum x_i y_i \\ \sum x_i y_i & \sum y_i^2 \end{bmatrix} = \begin{bmatrix} 4 & 6 \\ 6 & 44 \end{bmatrix}$$

d)

For corner detection correlation matrix can be given as,

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

so the equation would be,

$$R = \det(M) - k(\operatorname{trace}(M))^2$$

For above equation, $\det(M) = \lambda_1 \lambda_2$

$$\operatorname{trace}(M) = \lambda_1 + \lambda_2$$

where λ_1 and λ_2 are the Eigen values of M

So the values of these Eigen values decide whether a region is corner, edge or flat.

When $|R|$ is small, which happens when λ_1 and λ_2 are small, the region is flat.
When $R < 0$, which happens when $\lambda_1 >> \lambda_2$ or vice versa, the region is edge.
When R is large, which happens when λ_1 and λ_2 are large and $\lambda_1 \sim \lambda_2$, the region is a corner.

e)
Non-Maximum Suppression:
In corner detection, in order to pick up the optimal values to indicate corners, we find the local maxima by deleting corners in the vicinity of selected corners. The process of selecting non-maximum suppression can be stopped when deleting x% of pixels are completed in that detected corner window.

f) We know for Harris corner detection, the correlation matrix equation can be given as,

$$R = \det(M) - k(\operatorname{trace}(M))^2$$

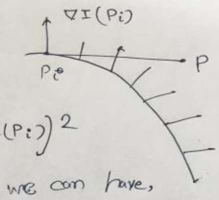
Where $\det(M) = \lambda_1 \lambda_2$

$$\operatorname{trace}(M) = \lambda_1 + \lambda_2$$

So from the above equation it's quite clear that Harris corner detection avoids computing Eigen values of the gradient correlation matrix by computing determinant of two Eigen values rather than computing those Eigen values directly.

g)
Better Localization of Corners:
For better localization, we project gradient onto edge hypothesis and choose P with minimal projection.

Here the objective function for minimizing the projection would be,



$$E(P) = \sum_i \left((P_i - P) \cdot \nabla I(P_i)\right)^2$$

To minimize the projection we can have,

$$\cancel{\nabla E(P)=0} \quad p^* = \underset{P}{\operatorname{argmax}}\, E(P)$$

$$\Rightarrow \text{Solve } \nabla E(P) = 0$$

Now $E(P)$ can be written as,

$$E(P) = \sum_i \left((P_i - P)^T \nabla I(P_i)\right)\left(\nabla I(P_i)^T (P_i - P)\right)$$

$$= \sum_i (P_i - P)^T \nabla I(P_i) \nabla I(P_i)^T (P_i - P)$$

So, $$\nabla E(P) = 2 \sum_i \nabla I(P_i) \nabla I(P_i)^T (P_i - P) = 0$$

or, $$\underbrace{\sum_i \nabla I(P_i) \nabla I(P_i)^T P_i}_{V} = \underbrace{\sum_i \nabla I(P_i) \nabla I(P_i)^T}_{\text{Corner Matrix } (C)} P$$

So, $V = CP$

where $P = C^{-1} V = C^{-1} \sum \nabla I(P_i) \nabla I(P_i)^T P_i$

Here P is the location of the corner.

h)

Histogram of Oriented Gradients(HOG):

The steps for HOG can be given as,

i) Take window
ii) Split into Blocks
iii) Compute histogram of gradient orientation in each block
iv) Concatenate the histograms

i)

SIFT Features Detection:

For SIFT, interest points are detected in the image using the Difference-of-Gaussian (DOG) operator. The points are selected as local extrema of the DOG function. At each interest point, a feature vector is extracted. Over a number of scales and over a neighborhood around the point of interest, the local orientation of the image is estimated using the local image properties to provide invariance against rotation. Next, a descriptor is computed for each detected point, based on local image information at the characteristic scale. The SIFT descriptor builds a histogram of gradient orientations of sample points in a region around the key-point, finds the highest orientation value and any other values that are within 80% of the highest, and uses these orientations as the dominant orientation of the key-point.

2)

a)

When using the Hough transform the problem with using the slope and y-intercept as line parameters is value of slope and y-intercept (a and b) can range between +infinite to –infinite. So while scanning image from left to right and top to bottom for each pixel will vote and this can range from +infinite to –infinite for a and b.

b)

b) $\theta = 45^\circ \quad d = 10$

$a \cdot x + b \cdot y + c \cdot x = 0$

$a = \cos\theta = \cos 45^\circ = 1/\sqrt{2}$

$b = \sin\theta = \sin 45^\circ = 1/\sqrt{2}$

$c = d = 10$

$ax + by + c = 0$

$1/\sqrt{2}\, x + \frac{1}{\sqrt{2}} y - 10 = 0$

or, $0.41 \simeq 0$

c)
Polar representation of a line can be given as,
. x*cos(theta)+ y*sin(theta)-d=0

From the above equation it looks like a sinusoidal curve in the parameter plane.

d)
In order to detect lines in parameter place, plot all the points on the same plane and then check the intersection of the lines. This intersection gives the theta at the x-axis and distance at y-axis, Then, verify this value with the actual value so that we can get accurate points.

e)
Trade off regarding bin-size in parameter plane:
If big bin size is used, then there will be fewer votes and where process will be fast.

Eventually it would be less sensitive to noise. However, if smaller bin size is used then there will be lot more votes and we may not get the intersection since each line will be little off and there will not be any exact intersection point and peak.

f)

If normal at each voting point is known, then it will be more efficient ass we will be able to scan the small part of angle and instead of tasking the theta bin of [0,180],we can compute delta(I) at voting points will get the theta and then scan the smaller plane in the range [theta-delta,theta+delta] where delta is same threshold value.

g)

Numbers of dimension of parameters space in circle using Hough transform is 3. The circle equation can be expressed as,

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

where a=x-r*cos(theta) and b=y-r*sin(theta)

3)

a)

The line fitting equation can be given as,

Y=a*x+b

If we get all the parameters for this equation, then we will get the best solution. At each point we chose the line that predicts the best measured of y for x coordinates.

But this model is very poor as we get a optimal solution but it leads to greater least square error. Mainly, the vertical lines lead to larger value of error.

b)

Value of l can be give as, [1 2 2]

c)

c) To fit a line using ellipse line equation we use

$$= \sum_{i=1}^{n} (l^T X_i)^2$$

or $l^T \rightarrow \begin{bmatrix} a \\ b \\ c \end{bmatrix}$

$X_i \rightarrow \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$

$$= \sum_{i=1}^{n} l^T X_i X_i^T l$$

$$= l\left(\sum_{i=1}^{n} X_i X_i^T\right.$$

Here, $E(l) = l^T C l$

$$l^* = \operatorname{argmin} E(l)$$

$$\nabla E(l) = 0 \rightarrow 2Cl = 0$$

These equation are to be solved for line parameters.

So, here,

$$C = \begin{bmatrix} \sum x_i & \sum x_i y_i & \sum x_i \\ \sum x_i y_i & \sum y_i^2 & \sum y_i \\ \sum x_i & \sum y_i & n \end{bmatrix} = D$$

where Data Matrix,

$$D = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix}$$

d)

d) $P_1 \rightarrow (0,1)$

$P_2 \rightarrow (1,3)$ and $P_3 \rightarrow (2,6)$

$$C = \begin{bmatrix} \Sigma x_i & \Sigma x_i y_i & \Sigma x_i \\ \Sigma x_i y_i & \Sigma y_i^2 & \Sigma y_i \\ \Sigma x_i & \Sigma y_i & n \end{bmatrix}$$

$$= \begin{bmatrix} 3 & 15 & 3 \\ 15 & 46 & 10 \\ 3 & 10 & 3 \end{bmatrix}$$

and Data Matrix $\rightarrow$

$$D = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 3 & 1 \\ 2 & 6 & 1 \end{bmatrix}$$

Here $C = D^T D$

e) Equation for conic curve can be given as,

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

The constraint on the parameters that guarantees the model will be ellipse would be,

$$\boxed{b^2 - 4ac < 0}$$

f) Equation for fitting an ellipse using algebraic distance can be given as,

$$q_i = l^T p_i$$

Where, q_i & d_i or $d_i + r_i$

The short axis point on the ellipse affect the more fitting.

d) Objective function using geometric distance →

$$E(\lambda) = \sum \frac{|f(P_i, \lambda)|}{|\nabla f(P_i, \lambda)|}$$

Here in geometric fitting, we need to find gradient distance or need to do relative fitting to get the ellipse parameters by Geometric fitting.

i) In discrete space →

$$E_{continuity} : \left|\frac{\delta \phi}{\delta s}\right|^2 \rightarrow \sum |P_i - P_{i-1}|^2$$

where $\phi(s) \rightarrow \{P\}_{i=1}^{n}$

Here n continuous points define the active contour curve.

$$E_{continuity} : \left|\frac{\delta^2 \phi}{\delta s^2}\right|^2 \rightarrow \sum (P_{i+1} - P_i) - (P_i - P_{i-1})$$

J) Continuity of active contours may be relaxed to allow for sharp corners as it will be helpful to place a point at the corner

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