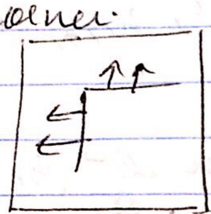
1 (a) Principle of corner detection:-

For a image, we define a window having neighbourhood points, where corners represent interest points, so goal is to identify these points.

The gradient in that window is seen, if there are more than one directions, it is said to have a corner.



window

(b) To find principal direction of gradient Orientation in the local patch,

we have

$\sum_{i=0}^{n} P_i P^T$ → as correlation matrix comprising of P_i = points in neighbourhood

here we find direction of minimum projection, a direction subject to be perpendicular to all previous directions.

Here,

Direction of → eigen vectors of Correlation matrix.
projections → proportional to eigen values.

(c) $\{(0,0)\ (0,1)\ (0,2)\ (0,3)$
$(0,4)\ (1,0)\ (1,1)\ (1,2)\ (1,3)\}$

correlation matrix --
$$\begin{bmatrix} \Sigma x_i^2 & \Sigma x_i y_i \\ \Sigma y_i x_i & \Sigma y_i^2 \end{bmatrix}$$

$$= \begin{bmatrix} 0+0+0+0+0 + 1^2+1^2+1^2+1^2 & 0+1+2+3 \\ 0+1+2+3 & 1^2+2^2+3^2+4^2+0 + 1^2+2^2+3^2 \end{bmatrix}$$

$$= \begin{bmatrix} 4 & 6 \\ 6 & 44 \end{bmatrix}$$

Scanned by CamScanner

d) Eigen vectors → λ_1, λ_2 ;
The eigen vector $\boxed{\lambda_1 \cdot \lambda_2 > \text{'}\tau\text{'} \rightarrow \text{threshold}}$.
then we detect a corner in that neighbourhood
here both λ_1, λ_2 are large enough.

e) Non- Maximum suppression helps in finding a unique corner for a location, when using multiple windows. As, for a point in image multiple windows will find multiple corners.

So steps are :-

a) compute λ_1, λ_2 for all

b) sort pixels based on ($\lambda_1 \cdot \lambda_2$) → select the top

c) start from top - select the strongest corner

d) Delete corners in the vicinity of the selected corners.

STOP when detecting x% of pixel.

f) In Harris Corner detection.,

$$\boxed{C(G) = \det(G) - K \cdot tr^2(G)}$$

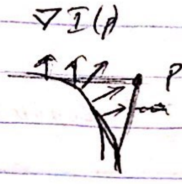
G is gradient correlation matrix

$\det(G) \rightarrow$ determinant of G

$tr(G) \rightarrow$ trace of G

thus we do not consider eigen values of the gradient correlation matrix directly instead we find determinant and trace.

(g)



we project gradient onto edge hypothesis and choose p minimum projection

Localization of point 'p'

$$P = C^{-1} v$$

$$= C^{-1} \sum_i \nabla I(P_i) \cdot \nabla I(P_i)^T \cdot P_i$$

where $C = \Sigma \nabla I(P_i) \cdot \nabla I(P_i)^T$

↳ correlation matrix

<u>conditions for solution to exist</u>

→ $\lambda_1, \lambda_2 > Z$ so that C is non-singular and we can inverse of that

(h) HOG

a) Take a window
b) split into blocks which can be overlapping.
c) Compute histogram of gradient orientations in each block
d) Concatenate histograms.
and we get feature vector

good characteristics of featurepoint are:-

→ Translation invarient
→ rotation invarient
→ scale invarient
→ illumination invarient

8)(g) SIFT

a) take a large window
b) split in to blocks
c) compute gradient vector in each block
d) Combine all the gradient vector, into orientation histogram over smaller subregion

Scanned by CamScanner

2) a LINE DETECTION.

a) The problem is that the range of both slope & y-intercept as the parameters vary from range [$-\infty$, ∞]

b) $\theta = 45$, $d = 10$

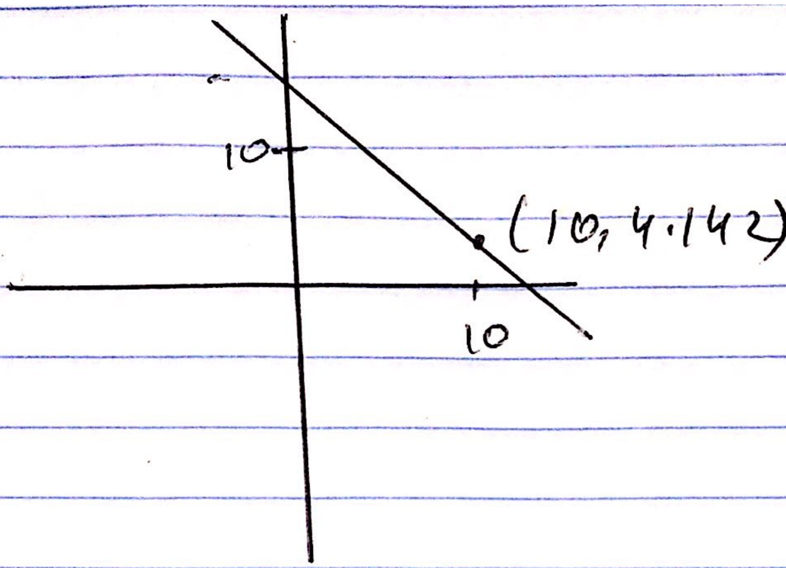
so equation is

$$x\cos\theta + y\sin\theta - d = 0$$

$$x\left(\frac{1}{\sqrt{2}}\right) + y\left(\frac{1}{\sqrt{2}}\right) - 10 = 0$$

multiply by $\sqrt{2}$ both side

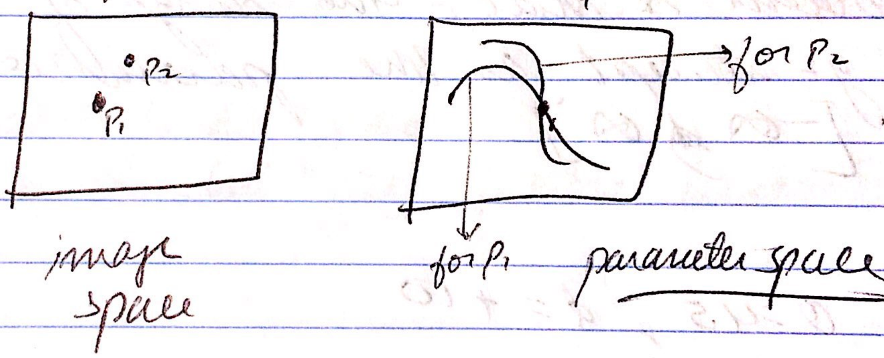
$$x + y - 10(\sqrt{2}) = 0$$



so point detected is (10, 4.142)
putting in equation we get,

$10 + 4.142 = 14.142$
$\approx 10\sqrt{2}$ thus result satisfied.

(C) when using polar representation of lines, vote for each point in image will look like a $\boxed{\text{sinusoidal curve}}$ in the parameter space (combination of sin & cos)



Scanned by CamScanner

d) parameter plane will give 2 value
d → distance from centre.
θ → θ for normal of the desired line in image plane.

so
the equation will be:
$x\cos\theta + y\sin\theta - d = 0$. → in image plane.

e) Trade off regarding bin-size is between efficiency and accuracy.

Bigger bin size → more efficient, less accuracy
Smaller bin size → less efficient, more accuracy.

(f) If the normal at each point is known, we can find ∇I at voting point & compute θ

→ now range will become $(\theta - \Delta \ldots \theta + \Delta)$

Thus voting is now more accurate, less sensitive to noise & faster.

(g) The number of dimensions of in Hough transform for circles is 3

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

so dimensions are ⇒ a, b, r

3a) — Using $y = ax+b$ model,

distance of neighbouring points to this line will be increased thus ~~does~~, will result in non-accurate fitting.

— Lines with higher slopes cannot be fitted accurately.

b) normal (1,2) distance, d = 2

so equation is $l^T x = 0$

when l^T has 3 coeff → a, b, c.

so.

$x + 2y + 2 = 0$

where a = 1

b = 2

c = 2.

so $l = [1, 2, 2]$

Scanned by CamScanner

(C) ∵ We use explicit line equation to minimize geometric distance,

$l^T x = 0$ · where all pts 'x' on line 'l' should satisfy this

Objective function to be minimised ($E(l)$)

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

$$= l^T \underbrace{\left(\sum_{i=1}^{n} (x_i \, x_i^T)\right)}_{C} l$$

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

$$l^* = \underset{l}{\operatorname{argmin}} \, E(l) \Rightarrow \Delta E(l) = 0$$

$$\Rightarrow 2Cl = 0 \quad \text{— (1)}$$

(1) solution will be solve for eigenvectors for corresponding zero eigenvalues.

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

(d) points ((0,1) (1,3), (2,6))

$C = D^T D$

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

$D^T D = \begin{bmatrix} 5 & 15 & 3 \\ 15 & 46 & 10 \\ 3 & 10 & 3 \end{bmatrix}$

$C = \begin{bmatrix} \Sigma x_i^2 & \Sigma x_i y_i & \Sigma x_i \\ \Sigma x_i y_i & \Sigma y_i^2 & \Sigma y_i \\ \Sigma x_i & \Sigma y_i & n \end{bmatrix} = \begin{bmatrix} 5 & 15 & 3 \\ 15 & 46 & 10 \\ 3 & 10 & 3 \end{bmatrix}$

Scanned by CamScanner

e) The input equation for conic curve is:-

$$a x^2 + b xy + c y^2 + dx + ey + f = 0 \quad (1)$$

→ all x, y should satisfy this (1)

constraints for it to be an ellipse.

→ $b^2 - 4ac < 0$

→ when $P_i = (x_i^2, x_i y_i, y_i^2, x_i, y_i, 1)$

$l^T P_i = 0$

$E(l) = \sum_{1}^{n} (l^T P_i)^2$ where $l \to (a, b, c, d, e, f)$ coeffs

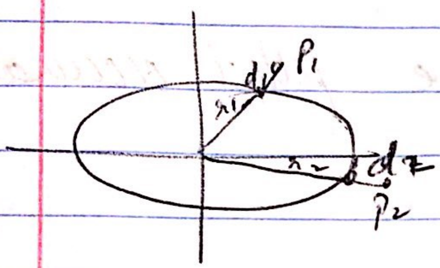
Scanned by CamScanner

f) Equation for fitting ellipse :-

$$\boxed{\sum_{i=1}^{n} (l^T p_i)^2} \rightarrow \text{where } l^T p_i \rightarrow \text{is algebraic distance.}$$

where $\boxed{q_i = l^T p_i}$ and $\boxed{q_i \propto \frac{d_i}{d_i + r_i}}$

$\downarrow$ algebraic distance



$\longrightarrow$ so $\boxed{q_1 > q_2 . \quad \frac{d_i}{d+r_1} > \frac{d}{d+r_2}}$

So points closer to P_1 ~~that short~~ i.e short-axis will affect fitting more and cause penalty.

3g) Using geometric distance.

Objective function $\rightarrow$ $E(\ell) = \frac{\sum_i f(P_i, \ell)}{|\nabla f(P_i, \ell)|}$

Here, dividing by gradient of algebraic distance is the additional complication involved.

3 h) Active contours:-

Objective function $\Rightarrow$

$$E[\phi(s)] = \int_{\phi s} E_{continuity}\, E_{curvature}\, E_{image}.$$

where :- $E_{continuity}$ $\rightarrow$ small is desired

$E_{curvature}$ $\rightarrow$ small is desired

E_{image} $\rightarrow$ hight integrated gradient

$$\therefore\ E[\phi(s)] = \int_{\phi s} \left(\alpha(s) E_{continuity} + \beta(s) E_{curvature} + \gamma_{(s)} E_{image}\right) ds$$

where $\left.\begin{matrix}\alpha(s) E_{continuity} \\ + \beta(s) E_{curvature}\end{matrix}\right\}$ ~~integral parts~~ internal parts

$\gamma_{(s)}\, E_{image}$ } Exernal

α, β, γ's are continuity, curvature &

3(i) Using Active contours,

continuity of discrete curve :-

$$E_{continuity} = \left|\frac{\partial \phi}{\partial s}\right|^2 \Rightarrow \sum \|\underbrace{P_i - P_{i-1}}_{\text{distance b/w neighbouring points}}\|^2$$

curvature of discrete curve :-

$$E_{curvature} = \left|\frac{\partial^2 \phi}{\partial s^2}\right|^2 \Rightarrow \sum \left|(P_{i+1} - P_i) - (P_i - P_{i-1})\right|$$

3(j) For sharp corners, continuity can be relaxed by placing snake point at the corner

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