COMPUTER VISION

CS-512 - ASSIGNMENT-3

RAKSHITH CHURCHAGUNDI AMARNATH
CWID: A20424771
Fall-2018

① Corner detection:-

a. Explain the basic principle of corner detection. How is the number of principle directions assessed?

Solution: Also Known as point of interest, corners are the points formed by the intersection of edges.

The basic principle is that the gradient of direction vector is high in more than one direction for the corner.

The steps of the algorithm to detect a corner in a local window are:-

(i) finding the correlation matrix of gradients in the local window.

(ii) find eigen values of correlation matrix.

(iii) Detect corner in window if eigen values are sufficiently large

if there are more than 1 orientations of the gradient vector in the local Neighbourhood, then we detect a corner, else if there is only one orientation then we detect the edge of the image ($\lambda_1 \cdot \lambda_2 > \tau$)

(b) ~~Ex~~ Explain how PCA is used to find Principal directions of gradient orientations in a local patch.

Solution :- PCA finds the principal directions of gradient orientations in a local path by finding the direction ~~of~~ to minimize projections of all points in the local path.

$$E(v) = \sum_{i=1}^{n} (g_i \cdot v)^2 = \sum_{i=1}^{n} (g_i^T v)\,(g_i^T v)$$

$$= \sum_{i=1}^{n} (v^T g_i)(g_i^T v) = \sum_{i=1}^{n} v^T g_i g_i^T v = v^T \Sigma (g_i g_i^T) v$$

$$= \underline{\underline{v^T C v}}$$

* the direction of minimum projection has to be found and also a direction subject to be perpendicular to all previous directions.
* the direction here the the eigen vector of correlation matrix and projection is proportional to eigen vectors

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

=> we know,

Correlation matrix, $C = \sum_{i=1}^{n} g_i g_i^T$

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

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

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

(d) For correlation matrix,

if the 2 eigen values are large, and dot Product i.e, $\lambda_1 . \lambda_2 > \tau$, we detect a corner.

where λ_1 and λ_2 are eigen value which are large, τ is the threshold value

(e) when using multiple windows, the non maximum suppression helps in finding a unique corner for a location.

Corner detection steps using non-maximum Support is as follows :-

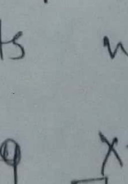
(i) compute $\lambda_1 . \lambda_2$ for all windows

(ii) Select windows with $\lambda_1 . \lambda_2 > \tau$ and sort in decreasing order

(iii) Select the top of the list as corner, and delete all other corners in its neighbourhood from the list.

(iv) stop once detecting x% of the Points as corners

corner



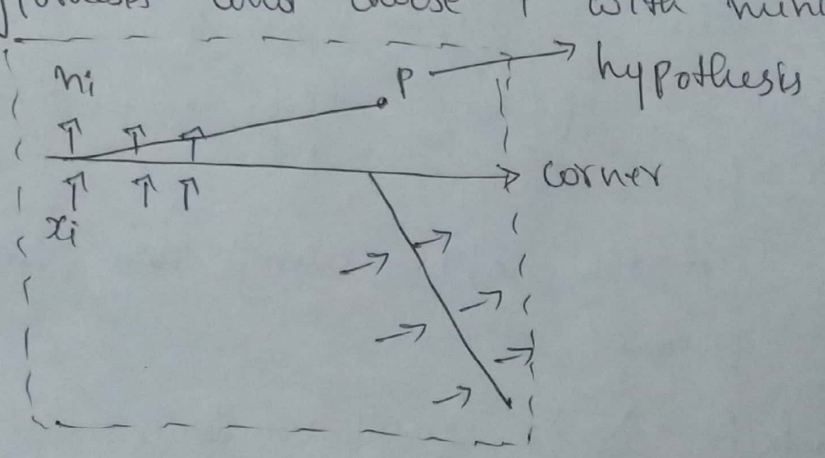
(f) In Harris corner detection, we have.

$$G(C) = \det(C) - k\,tr^2(C)$$

$\det(C) \to \lambda_1, \lambda_2$; $k\,tr^2(C) \to k(\lambda_1+\lambda_2)^2$

∴ we do not consider eigen values of the gradient correlation matrix directly instead, we find determinant to detect corner or not and trace to detect edge or not edge.

(g) For better localization of a corner, we try to find the best hypothesis P, by projecting gradients onto edge hypothesis and choose P with minimal projection.



Hence the object function is $E(P) = \sum_{i=1}^{n} \left((x_i - p) \cdot \nabla I(x_i)\right)^2$

i.e projection of $\nabla I(x_i)$ onto $(x_i - p)$ $i=1$

$p^* = \underset{p}{\operatorname{argmin}}\, E(p)$, so we solve $\nabla E(p) = 0$

thus derivative is

$$\nabla E(p) = 2\sum_{i=1}^{n} \left(\nabla I(x_i) \cdot \nabla I(x_i)^T\right)(x_i - p)$$

$$\underbrace{\sum_i \nabla I(x_i)\, \nabla I(x_i)}_{\substack{2\times 2 \\ C_1}} \underbrace{P}_{2\times 1} = \underbrace{\sum \nabla I(x_i) \cdot \nabla I(x_i)^T \cdot x_i}_{2\times 1}$$

Correlation Matrix

$$P^* = \left(\sum_i \nabla I(x_i) \nabla I(x_i)^T\right)^{-1} \sum \nabla I(x_i) \cdot \nabla I(x_i)^T \cdot x_i$$

Thus $V = CP$.

$P = C^{-1} V$

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

$\to$ correlation matrix

location of corner

$\therefore$ for the solution to exist, C^{-1} should exists

C^{-1} exists as C is singular matrix and λ_1, λ_2 are large as $\lambda_1 \cdot \lambda_2 > \tau$ where λ_1 & λ_2 are eigen values of C.

(h) Feature points can be characterized using histogram of oriented gradients (HoG)

<u>Steps</u>:-

* divide image into windows
* split them into blocks
* compute histogram of gradient orientation in local blocks.
* concatenate the histograms.

Requirements from a good characterization of feature points are:-

* feature point should be translation invariant
* it should be scale invariant
* it has to be illumination invariant.
* it should be scale invariant.

(i) SIFT features are computed by.

* Taking large neighbourhood of pixels.
* Breaking the neighbourhood into small blocks
* Gradient vector is computed for each pixel in block
* orientation histogram is computed for each block by accumulating the gradient of the pixel.
* Finally, the histograms are connected to compute the features

LINE - DETECTION:-

(a) The slope can get to infinity as the line is parallel to y-axis and also the y-intercept can range from $[-\infty, \infty]$. So, we cannot compute all parameters and know before, how much space to reserve to store these parameters

(b)

θ = slope = 45°

d = 10 from origin.

$\Rightarrow d = x\cos\theta + y\sin\theta$

$10 = x\cos 45^\circ + y\sin 45^\circ$

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

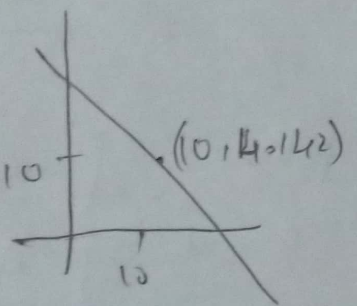
$x+y = 20/\sqrt{2}$

$x+y = 14.14$

$\therefore a = 1$

$b = 1$

$c = 14.14$



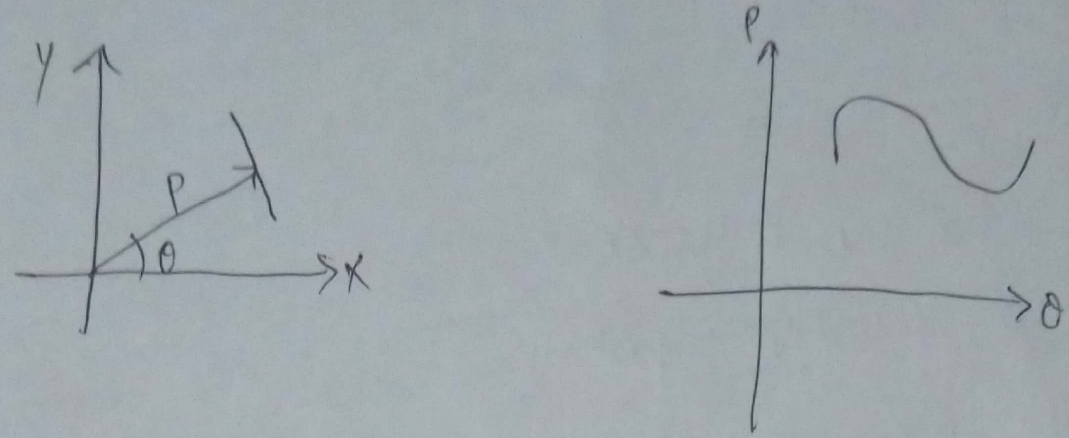
So, Point detected at (10, 4.142)

$\therefore$ Put in eqn,

$10 + 4.142 = 14.142$

$= 10\sqrt{2}$ (satisfied)

(c)



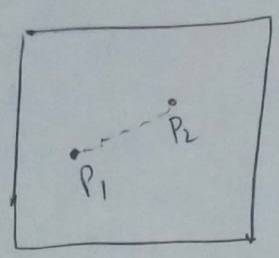
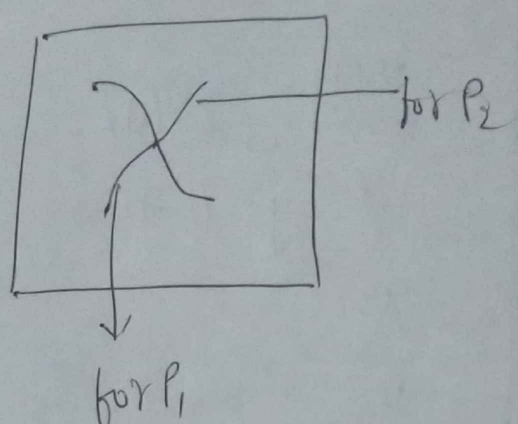


Image space



Parameter space

when using polar representation of lines, Vote for each point in image and that looks like a sinusoidal curve in the Parameter space

(2d) lines are detected by taking the point where all the vote intersect in the parameter plane.

this point defines the Parameter of the line a,b,c, using these, the line can be constructed. as $ax+by+c=0$

Steps:-

* Detect edges
* Map edge points to Hough space and store it.
* Yield stored points in lines of infinite length.
* Convert infinite lines to finite ones
* Find interception.

(e) Trade off with bin size in parameter plane.

Big Bin SIZE:-

* we get less votes and missout some information.
* less accurate
* less computation to be done
* less sensitive to noise.

Small Bin SIZE:-

* we get more votes and may be susceptible to noise.
* More computation has to be done
* Votes may not intersect because of more accurate information.

(f) if the normal at each voting point is known instead of using the range of θ from $0°$ to $180°$ $[\theta_{min}, \theta_{max}]$. it can improve the voting in the parameter plane. this can be achieved by finding θ using the ∇I at each voting point and scan over $(\theta - \nabla I, \theta + \nabla)$ which is computationally more efficient

(g) The number of dimension in Hough transform for circle is 3.

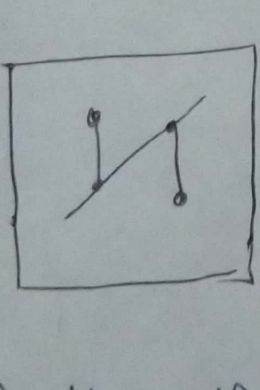
<u>example</u> $(x-a)^2 + (y-b)^2 = r^2$

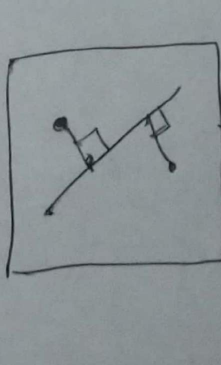
$\therefore$ a, b, r are the dimensions.

$\therefore$ The parameter space is 3 Dimensional.

(3) <u>MODEL FITTING:-</u>

(a)





Eq^n $y = ax+b$, for line fitting do not lead to an optimal solution. It only minimizes the algebraic distance of the actual points to the

Predicted points on the line.

Line with bigger slope cannot fit accurately with this equation.

(b) Normal $\Rightarrow (1,2)$

distance, $d \Rightarrow 2$

$l^T x = 0,$

l^T has 3 co-efficient $\rightarrow a, b, c.$

$ax + by + c = 0$

$a = 1,\ b = 2,\ c = 2$

$l = [1, 2, 2]$

(c) Explicit line equation is used to minimize geometric distance i.e $l^T x = 0$.

where all points, x on line 'l' should satisfy their objective function to be minimized $E(l)$.

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

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

$$= l^T C l$$

(6)

$l^* = \arg\min_l E(l) \rightarrow \nabla E(l) = 0$

$2Cl = 0$ —— ①

lets solve for eigen value vectors for corresponding zero eigen values.

where $C = \begin{bmatrix} \sum x_i^2 & \sum x_i y_i & \sum x_i \\ \sum x_i y_i & \sum y_i^3 & \sum y_i \\ \sum x_i & \sum y_i & n \end{bmatrix}$

(d) Given $\{(0,1), (1,3), (2,6)\}$

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

$= \begin{bmatrix} 0^2+1^2+2^2 & 0+3+12 & 0+1+2 \\ 0+3+12 & \cancel{0}+1^2+3^2+6^2 & 1+3+6 \\ 0+1+2 & 1+3+6 & 3 \end{bmatrix}$

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

(e) $P_i = (x_i^2, x_i y_i, y_i^2, x_i, y_i, 1)$

The explicit eqns of the conic curves is represented by.

$$S = \sum_{i=1}^{n} P_i . P_i^T$$

The implicit Eqn is

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

$b^2 - 4ac < 0$, guarentees that the model is an ellipse.

(f) we have to solve,

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

where $P_i = (x_i^2, x_i y_i, y_i^2, x_i, y_i, 1)$

Given, $\{x_i, y_i\}^n$

Solving above equation to fit an ellipse using algebraic distance $Sl = 0$, where.

$$S = \sum_{i=1}^{n} P_i P_i^T.$$

given the Soln, eigen vector corresponding to zero eigen values.

7

Points closer to the short axis of the ellipse have more effect on the fittings as these points get more weight considering, the algebraic distance of these are lesser than those closer to the long axis of the ellipse

(g) objective function to be minimized when fitting an ellipse using geometric distance.

$\Rightarrow E(l) = \sum_i \frac{|f(P_i, l)|}{|\nabla f(P_i, l)|}$ where, $|f(P_i, l)| = (l^T P_i)^2$
$P_i = (x_i^2, x_i y_i, y_i^2 x_i, y_i, 1)$

$\rightarrow$ the result will not be a quadratic equation, so we don't get an explicit solution, and a linear solver cannot be used. So iterative approach like gradient descent.

(h) objective function of active contours:-

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

$\alpha(s), \beta(s), \gamma(s)$ are co-efficients (variables).

$E_{continuity}$, $E_{curvature}$, E_{img} are energy terms.

$\alpha(s)E_{continuity} + \beta(s)E_{curvature}$ is Internal part and

$\gamma(s)E_{img}$ is the external part.

* we want $E_{continuity}$ and $E_{curvature}$ to be smaller and E_{img} to be high i.e, high integrated gradients and $E_{continuity} = \left|\frac{\partial \phi}{\partial s}\right|^2$,

$E_{curvature} = \left|\frac{\partial^2 \phi}{\partial s^2}\right|^2$ and $E_{img} = -|\nabla I|^2$

(i) when the curve is descrete and using active contours:-

$E_{continuity}$ is estimated as distance between neighbouring points i.e, $E_{continuity} = \Sigma|P_i - P_{i-1}|^2$

$E_{curvature}$ is estimated as difference of tangents at neighbouring points.

$$E_{curvature} = \Sigma|(P_{i+1} - P_i) - (P_i - P_{i-1})^2|$$

$$= \Sigma|P_{i+1} - 2P_i + P_{i-1}|^2$$

(j) The continuity of active contours may be relaxed or to allow discontinuity, we find high curvature points and $\beta_i = 0$, i.e, if $|P_{i+1} - 2P_i - P_{i-1}| > \tau$ then $\beta_i = 0$.