CS 512 FALL18

VISMAYA VEERAMANJU KALYAN

A20423189

1(a) Basic principle of corner detection

1. Find correation matrix of gradients in local window
2. Find the Eigen values of correation matrix
3. Check if the Eigen values are large.

ie $\lambda_1 \lambda_2 > \tau$

The number of principle direction is assessed using PCA
More than one principle direction are assessed in case of a corner.

(b) PCA to find principle direction of gradient orientation in a local patch

objective func. $E(v) = \sum (g_i \cdot v)^2$

$= \sum (g_i^T v)(g_i^T v)$

$= \sum_i v^T g_i g_i^T v$

$= v^T \sum g_i g_i^T v$

$= v^T C v$

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

Argument of that will minimize the function

$\nabla E(v) = 0$

$2Cv = 0$

$Cv = 0$

By solving this equation, the solution is the eigen vectors belonging to smallest eigen values

$\lambda_1 \lambda_2 > \tau$ detect corner

(c) $\sum_i g_i g_i^T = \begin{bmatrix} \sum x_i^2 & \sum x_i y_i \\ \sum x_i y_i & \sum y_i^2 \end{bmatrix} = \begin{bmatrix} 1+1+1+1 & 1+2+3 \\ 1+2+3 & 1+4+9+16+1+4+9 \end{bmatrix}$

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

(d) $\lambda_1 \lambda_2 > \tau$

Both the Eigen values must be large for a corner to be detected and the product of the Eigen values must be larger than threshold.

(e) non-maximum supression.

1. compute λ_1, λ_2 for all windows
2. Select window with $\lambda_1 \lambda_2 > \gamma$ and sort in decreasing order.
3. Select the top of the list as corners and delete all other corners in its neighbourhood from the list.
4. Stop once detecting x% of the points as corners

(f) Harris corner detection.

1. Compute co-rrelation matrix C for windows
2. Compute cornerness measure
   $C'(C) = \det(C) - K\, tr^2(C)$
3. Detect corners where $C'(C)$ is high.

K ranges from 0 to 0.5.
K = 0 detects corners
K = 0.5 detects edges.

★ We can avoid computing eigen value by use of det(c) to do pure corner detection.
K = 0

$$C'(C) = \det(C) = \lambda_1 \lambda_2$$

(g) To determine if P is the corner connect each point x_i to p and project the project the gradient at x_i onto $(x_i - p)$

$$E(p) = \sum_i ((\nabla I(x_i) \cdot (x_i - p))^2$$

$$= \sum_i (x_i - p)^T (\nabla I(x_i) \nabla I(x_i)^T)(x_i - p)$$

Find p that minimizes the sum.

$$\nabla E(p) = 0$$

$$2(x_i - p) \sum_i (\nabla I(x_i) \nabla I(x_i)^T) = 0$$

$$\sum_i \nabla I(x_i) \nabla I(x_i)^T P = \sum \nabla I(x_i) \nabla I(x_i)^T x_i$$

location of corner

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

$$= C^{-1} \sum \nabla I(x_i) \nabla I(x_i)^T x_i$$

← Correlation matrix

The condition that needs to be ~~checked~~ satisfied is the co-relation matrix must be invertable (As we have already detected if the window contains a corner or not using Harris or other method. correlation matrix will be invertable

h) HOG - Histogram of Oriented Gradients (HOG)
1) split each patch into cells (possibely overlapping.
2) Create orientation histogram in each cells (using edge or gradient direction possibly weighted by distance from center or gradient magnitude.
3) Concatinate orientation histogram.

Requirements of a good characterization of feature points.
1) translation invariance - local window
2) rotation invariance - histogram
3) Scale invariance - pyramids.
4) Illumination invariance - gradients.

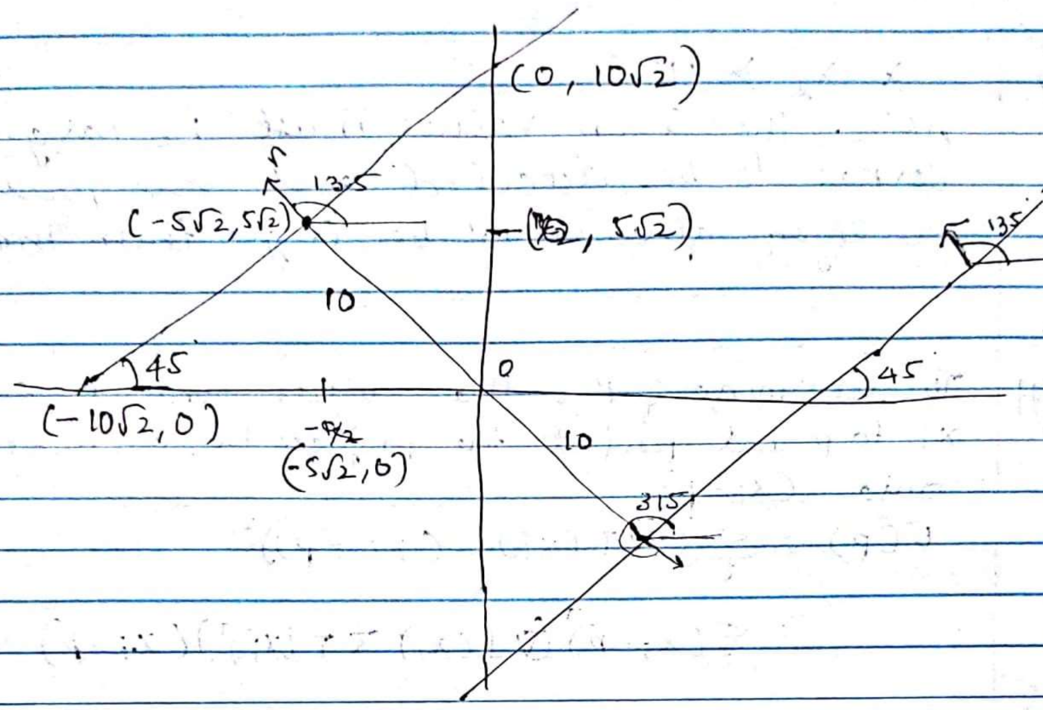
i) SIFT -
Use weighted sum % to create orientation histogram in cells, then concatenate.

Align histogram based on dominant direction (rotation invariance

2(a) Probleem with $y = ax+b$ model

The size of the parameter space is not determined (ie deciding the possible range of a?) or How to represent vertical line. The value can range till infinity.
$a \in [-\infty, \infty]$ and $b [\infty, \infty]$, its a problem hence we use the implicit line equation.

2 b



$ax + by + c = 0$

$\cos 135\, x + \sin 135\, y - 10 = 0$

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

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

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

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

$a = -1$

$b = 1$

$c = -10\sqrt{2}$

Satisfy the equation of line

ie $x = 0$: $y = 10\sqrt{2}$

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

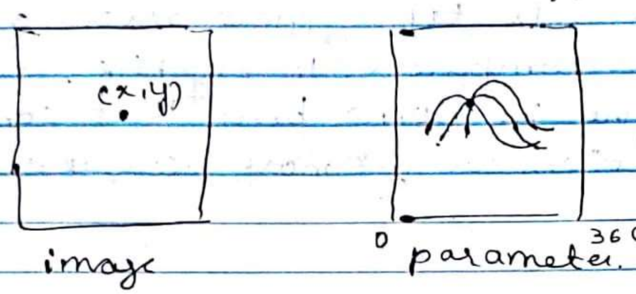
$x = -5\sqrt{2}$ $y = 5\sqrt{2}$

Scanned by CamScanner

c) The vote for the point (x, y)

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

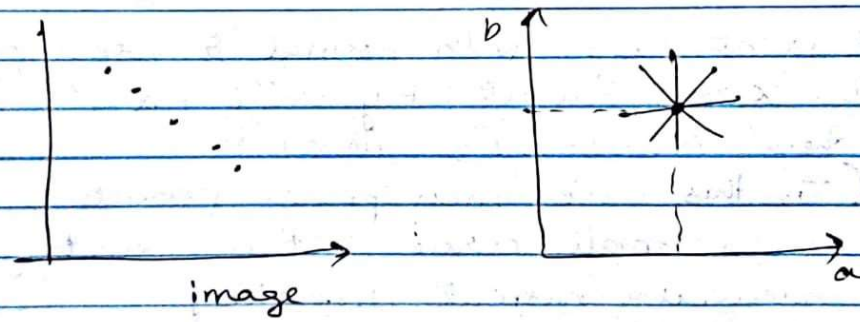
scan $\theta \in [0, 360]$ and compute d



For each θ : vote $d_i = x\cos\theta + y\sin\theta$

The vote of each point is sinusoidal in the parameter space.

d)



Each point in the image, votes for a line in the parameter space. Hence the parametric space is filled lines. If there are intersection of line in the parametric space it indicates a line with common parameter.

The intersection point in parameter space represent the parameter (a, b) of a line on which the set of points lie.

e) Big bin : It takes fewer vote but its much faster. however there is a compromise on the accuracy. It also reduces the use of memory.
Small bins : It takes more votes and its much slower compared to big bins.
There might be some bins that contain no intersection. We get more accurate results.

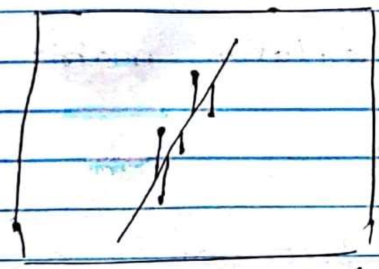
f) If the normal at each voting point is known, we could be more efficient,
Instead of using $\theta \in (0, 180)$
we can take $\theta \in (\theta_{min}, \theta_{max})$
ie
A point (x,y) with normal θ votes for
$d = x\cos(\theta + \alpha\Delta\theta) + y\sin(\theta + \alpha\Delta\theta)$
where $\alpha \in [-1, 1]$ (10 %)
(In this case each pixels provide a vote which is a small curve section accounting for inaccurate normal direction)

g) Given (x,y) vote in each r plane using a, b.
where : $a = x - r\cos\theta$. $\theta \in [0, 360)$
$b = y - r\sin\theta$

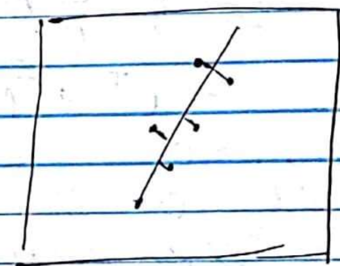
Hence we need 3 dimensions ~~for~~ for the parameter space as it contains 3 parameter (a, b, r)

3a Disadvantage of using equation $y = ax + b$ for line fitting, it doesnt ~~the~~ consider the geometric distance between the prediction and the real point ~~and~~ [illegible]

The solution obtained is optimal in the sense of minimizing the objective but doesnt consider the geometric distace.



$y = ax + b$ model



geometric distance

Vertical or close to vertical lines cannot be fitted accurately using this equation.

b $l = 3 \times 1$

$$l^T = 1 \times 3 = \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ -2 \end{bmatrix}$$

$$n = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad d = 2$$

$$l^T x = \begin{bmatrix} 1 \\ 2 \\ -2 \end{bmatrix} [x, y, 1] = 0$$

$$L = [1, 2, -2]$$

c) $l^T p = p.n - d$ (considering geometric distance)

$E(l) = \sum_i (l^T p_i)^2$

$l^* = \arg\min_l E(l)$

$E(l) = \sum l^T p_i p_i^T l$

$= l^T \sum_i p_i p_i^T l$

$= l^T S l$ ← corelation matrix

$\nabla E(l) = 0$ ← minimizing

$Sl = 0$

l = Eigen vectors of S belonging to zero Eigen value

d) $S = \sum p_i p_i^T$

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

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

$$\begin{bmatrix} 1+4 & 0+3+12 & 1+2 \\ 0+3+12 & 1+9+36 & 1+3+6 \\ 1+2 & 1+3+6 & 3 \end{bmatrix}$$

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

Scanned by CamScanner

~~d~~e. Explicit equation of conic curves

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

~~which sal~~

Constraint to satisfy as an ellipse

$b^2 - 4ac < 0$

f. $l^T p = 0$

$l = (a, b, c, d, e, f)$

$p = (x^2, xy, y, x, y, 1)$

Equation that needs to be solved for fitting an ellipse is

$E(l) = \sum_{i=1}^{m} (l^T p_i)^2$ such that $b^2 - 4ac < 0$ constrain

$= l^T S l + \lambda (l^T C l + 1)$

$l^* = \arg\min_l E(l)$

$\nabla E(l) = 0$

$2Sl = 2\lambda C l$

$l = \lambda S^{-1} C l$

l^* is the eigen vector of $S^{-1}C$ belonging to negative eigen value.

$\Rightarrow$ The points close to short axis are effected more.

$q = l^T x_i \simeq \frac{d_i}{d_i + r_i}$

$d_1 = d_2 = d, \quad r_2 > r_1$

$\frac{d}{d + r_1} > \frac{d}{d + r_2}$

$q_1 > q_2$

3 g) Geometric objective

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

where $f(P_i, \ell) = \ell^T p_i$

$\ell^* = \underset{\ell}{\arg\min}\, E(\ell)$ such that $b^2 - 4ac < 0$

The complication involved when trying to determine the ellipse parameter is that the above equation is not linear.

3(h) Objective function:

$$E[\phi(s)] = \int_{\phi(s)} \Big(\underbrace{\alpha(s) E_{cont} + \beta(s) E_{curv}}_{\text{internal energy}} + \underbrace{\gamma(s) E_{img}}_{\text{External Energy}} \Big) ds$$

$\alpha(s)$, $\beta(s)$, $\gamma(s)$ are co-efficients of the different energy terms.

Continuity energy: $E_{cnt} = \left| \frac{d\phi}{ds} \right|^2$

curvature energy: $E_{curv} = \left| \frac{d^2\phi}{ds^2} \right|^2$

Scanned by CamScanner

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

3 i) $E_{cont} = \left| \frac{d\phi}{ds} \right|^2 = |P_{i+1} - P_i|^2$

$E_{curv} = \left| \frac{d^2\phi}{ds^2} \right|^2 = |(P_{i+1} - P_i) - (P_i - P_{i+1})|^2$

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

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

3 j) The continuity of active contours will be $|P_i - P_{i-1}| - d$ to allow for sharp corner. d = average distance between points to prevent shrinking.

Scanned by CamScanner