Name - Shivashish Suman

Roll no. - CS17BTECH11037.

1. (a) After zero padding I will be : $\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 1 \\ 0 & 1 & -1 & 2 \end{bmatrix}$

$$F * I = \begin{bmatrix} 0*1+0*-1+0*-1+2*1 & 0*1+0*-1+0*1+2*-1 & 0*-1+0*1+0*-1+1*1 \\ 0*-1+0*-1+2*1+1*1 & -1*2-1*1+1*-1+0*-1 & 1*1+2*1+0*-1+-1*-1 \end{bmatrix}$$

$$F * I = \begin{bmatrix} 2 & -2 & 1 \\ 3 & -4 & 4 \end{bmatrix}$$

$\therefore F * I = \begin{bmatrix} 2 & -2 & 1 \\ 3 & -4 & 4 \end{bmatrix}$

(b) $F_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ $F_2 = \begin{bmatrix} 1 & -1 \end{bmatrix}$ $F_1 F_2 = \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} = F$

For computing $(F_1 * I)$ we first use zero padding so

I will become : $\begin{bmatrix} 0 & 0 & 0 \\ 2 & 0 & 1 \\ 1 & -1 & 2 \end{bmatrix}$

Scanned with CamScanner

$$\therefore\ F_1 * I = \begin{bmatrix} 0*1+2*1 & 0*1+0*1 & 0*1+1*1 \\ 1*1+2*1 & 0*1+-1*1 & 1*1+2*1 \end{bmatrix}$$

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

Before convolving $F_1 * I$ with F_2, we first use zero padding to get:

$$\begin{bmatrix} 0 & 2 & 0 & 1 \\ 0 & 3 & -1 & 3 \end{bmatrix}$$

$$\therefore\ F_2 * (F_1 * I) = \begin{bmatrix} -1*0+2*1 & 1*0+2*-1 & 1*1+-1*0 \\ -1*0+1*3 & -1*3+1*-1 & -1*-1+1*3 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & -2 & 1 \\ 3 & -4 & 4 \end{bmatrix}$$

(c) $$(F * I)[i,j] = \sum_{k,l} I[i-k, j-l]\, F[k,l]$$

$$= \sum_{k,l} I[i-k, j-l]\, F_1[k] F_2[l] \quad \left\{ \because F[k,l] = F_1[k] \cdot F_2[l] \right\}$$

$$= \sum_{l} F_2[l] \left(\sum_{k} I[i-k, j-l]\, F_1[k] \right)$$

$$= \sum_{l} F_2[l] \cdot (F_1 * I)[i, j-l] \quad \left\{ \because (F_1 * I)[i,j] = \sum_k I[i-k, j] \cdot F_1[k] \right.$$

$$= F_2 * (F_1 * I)$$

$$\left\{ \because F_2 * I[i,j] = \sum_l F_2[l]\, I[i, j-l] \right\}$$

(d) In part (a) there are 6 elements in which each takes 4 multiplication, So in total 24 multiplications in part (a).

Scanned with CamScanner

In part (b), $F_1 * I$ takes 12 multiplication operation and then $F_2 * (F_1 * I)$ takes 12. So in total it takes 24 multiplication

$\therefore$ Both requires same number of operations.

(e) (i) For each $M_1 \times N_1$ positions we have to perform $M_2 \times N_2$ multiplications.

$\therefore$ Number of multiplications $= M_1 \times N_1 \times M_2 \times N_2$

(ii) 1D convolution on rows will take N_2 multiplications for each $M_1 \times N_1$ positions.

Similarly for columns it will take M_2 for each $M_1 \times N_1$ position

$\therefore$ Total number of multiplications $= M_1 \times N_1 \times M_2 + M_1 \times N_1 \times N_2$

$= M_1 \times N_1 \times (M_2 + N_2)$

(iii) We know that $O(M_2 + N_2)$ is much more efficient as compared to $O(M_2 \cdot N_2)$ {$O(n)$ is much more efficient than $O(n^2)$ as n grows}.

$\therefore$ Direct convolution is $O(M_1 N_1 (M_2 \cdot N_2))$ and two successive 1D convolution is $O(M_1 N_1 (M_2 + N_2))$.

$\therefore$ Two successive 1-D convolution is more efficient in general as compared to direct 2D convolution

2. (a) After rotation, $x' = x\cos\theta$, $y' = x\sin\theta$.

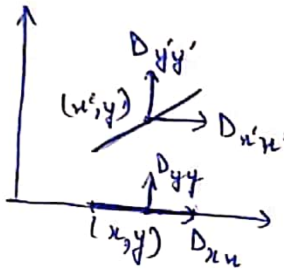
The only part of the algorithm which can change is the magnitude of the derivative.

For original edge, edge strength $= \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$

$$= \sqrt{D_{xx}^2} \quad \left\{ \because \frac{\partial f}{\partial y} = 0 \right\}$$

$$= |D_{xx}| \qquad \text{for horizontal edge.}$$

For rotated edge, ~~edge~~ magnitude of derivative $= \sqrt{D_{x'x'}^2 + D_{y'y'}^2}$



$$= \sqrt{D_{xx}^2\cos^2\theta + D_{xx}^2\sin^2\theta}$$

$$= \sqrt{D_{xx}^2(\cos^2\theta + \sin^2\theta)}$$

$$= \sqrt{D_{xx}^2} = |D_{xx}|$$

$\therefore$ Magnitude of derivative remains same. Hence rotated edge will be detected using the same Canny edge detector.

(b) In the first case, "low" value should be decreased so that edges are not broken and gap between low and high increases so that more points are marked as edge point.

In the second case, "high" value should be increased in order to reduce the ~~sp~~ spurious edges and also to decrease the edge points.

In both case, "low" and "high" value should be changed by small ~~margin~~ delta so as to reach best solution which do exist as per the assumption.

Scanned with CamScanner