North South University

Department of Electrical & Computer Engineering

Homework 3

Course Instructor: Dr. Shahnewaz Siddique
Course Code: CSE495
Course Title: Introduction to Robotics
Section: 01
Semester: Summer 2023
Date of Submission: November 15, 2023

Submitted By:

Name	ID
Ahmed Kiser	2021280042

Ans to the Ques No 1

Given that,

$$G(i,j) = \sum_{u=0}^{k-1} \sum_{v=0}^{L-1} F(u,v) \cdot \bar{I}(i+u, j+v)$$

$$I = \begin{bmatrix} 7 & 4 & 1 \\ 8 & 5 & 2 \\ 9 & 6 & 3 \end{bmatrix}, \quad \bar{I} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 7 & 4 & 1 & 0 \\ 0 & 8 & 5 & 2 & 0 \\ 0 & 9 & 6 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(a) $F = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

$G = F \times I$ and $G \in \mathbb{R}^{3\times 3}$

according to, ~~F(0,0)~~ formula, F start from, (0, 0).

$$G(1,1) = F(0,0) \cdot \bar{I}(1,1) + F(0,1)\bar{I}(1,2) + F(0,2)\bar{I}(1,3)$$
$$+ F(1,0)\bar{I}(2,1) + F(1,1)\bar{I}(2,2) + F(1,2)\bar{I}(2,3)$$
$$+ F(2,0)\bar{I}(3,1) + F(2,1)\bar{I}(3,2) + F(2,2)\bar{I}(3,3)$$
$$= F(1,1)\bar{I}(2,2) = 7$$

as others value of f are 0 without f(1,1) (according to formula) so, we can calculate only F(1,1)

$G(1,2) = F(1,1) \cdot \bar{I}(2,3) = 4$

$G(1,3) = F(1,1) \cdot \bar{I}(2,4) = 1$

$G(2,1) = F(1,1) \cdot \bar{I}(3,2) = 8$

$G(2,2) = F(1,1) \cdot \bar{I}(3,3) = 5$

$G(2,3) = F(1,1) \cdot \bar{I}(3,4) = 2$

$G(3,1) = F(1,1) \cdot \bar{I}(4,2) = 9$

$G(3,2) = F(1,1) \cdot \bar{I}(4,3) = 6$

$G(3,3) = F(1,1) \cdot \bar{I}(4,4) = 3$

$$G = \begin{bmatrix} 7 & 4 & 1 \\ 8 & 5 & 2 \\ 9 & 6 & 3 \end{bmatrix}$$

(b) $F = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ $F(0,0) = 1$ (according to formula)

$G(1,1) = F(0,0) \cdot I(1,1) = 0$

$G(1,2) = F(0,0) \cdot I(1,2) = 0$

$G(1,3) = F(0,0) \cdot I(1,3) = 0$

$G(2,1) = F(0,0) \cdot I(2,1) = 0$

$G(2,2) = F(0,0) \cdot I(2,2) = 7$

$G(2,3) = F(0,0) \cdot I(2,3) = 4$

$G(3,1) = F(0,0) \cdot I(3,1) = 0$

$G(3,2) = F(0,0) \cdot I(3,2) = 8$

$G(3,3) = F(0,0) \cdot I(3,3) = 5$

$G = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 7 & 4 \\ 0 & 8 & 5 \end{bmatrix}$

(e) $F = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$

$$G(1,1) = F(0,0)\bar{I}(1,1) + F(0,1)\bar{I}(1,2) + F(0,2)\bar{I}(1,3)$$
$$+ F(2,0)\bar{I}(3,1) + F(2,1)\bar{I}(3,2) + F(2,2)\bar{I}(3,3)$$
$$= 0 + 0 + 0 + 0 - 8 - 5 = -13$$

$$G(1,2) = F(0,0)\bar{I}(1,2) + F(0,1)\bar{I}(1,3) + F(0,2)\bar{I}(1,4)$$
$$+ F(2,0)\bar{I}(3,2) + F(2,1)\bar{I}(3,3) + F(2,2)\bar{I}(3,4)$$
$$\cancel{= 0 + 0 + 0 - 5 - 2 + 0 = -7}$$
$$= 0 + 0 + 0 - 8 - 5 - 2 = -15$$

$$G(1,3) = F(0,0)\bar{I}(1,3) + F(0,1)\bar{I}(1,4) + F(0,2)\bar{I}(1,5)$$
$$+ F(2,0)\bar{I}(3,3) + F(2,1)\bar{I}(3,4) + F(2,2)\bar{I}(3,5)$$
$$= 0 + 0 + 0 - 5 - 2 = -7$$

$$G(2,1) = F(0,0)\bar{I}(2,1) + F(0,1)\bar{I}(2,2) + F(0,2)\bar{I}(2,3)$$
$$+ F(2,0)\bar{I}(4,1) + F(2,1)\bar{I}(4,2) + F(2,2)\bar{I}(4,3)$$
$$= 0 + 7 + 4 - 0 - 9 - 6 = -4$$

$$G(2,2) = F(0,0)\,I(2,2) + F(0,1)\,I(2,3) + F(0,2)\,I(2,4) + F(2,0)\,I(4,2) + F(2,1)\,I(4,3) + F(2,2)\,I(4,4)$$

$$= 7+4+1-9-6-3 = -6$$

$$G(2,3) = F(0,0)\,I(2,3) + F(0,1)\,I(2,4) + F(0,2)\,I(4,5) + F(2,0)\,I(4,3) + F(2,1)\,I(4,4) + F(2,2)\,I(4,5)$$

$$= 4+1+0-6-3-0 = -4$$

$$G(3,1) = F(0,0)\,I(3,1) + F(0,1)\,I(3,2) + F(0,2)\,I(3,3) + F(2,0)\,I(5,1) + F(2,1)\,I(5,2) + F(2,2)\,I(5,3)$$

$$= 0+8+5+0+0+0 = 13$$

$$G(3,2) = F(0,0)\,I(3,2) + F(0,1)\,I(3,3) + F(0,2)\,I(3,4) + F(2,0)\,I(5,2) + F(2,1)\,I(5,3) + F(2,2)\,I(5,4)$$

$$= 8+5+2-0-0-0 = \cancel{18}\ 15$$

$$G(3,3) = F(0,0)\,I(3,3) + F(0,1)\,I(3,4) + F(0,2)\,I(3,5) + F(2,0)\,I(5,3) + F(2,1)\,I(5,4) + F(2,2)\,I(5,5)$$

$$= 5+2+0+0-0 = 7$$

$$G = \begin{bmatrix} -13 & -15 & -7 \\ -4 & -6 & -4 \\ 13 & 15 & 7 \end{bmatrix}$$

(d) ~~The~~ The ~~m~~ Filter is shown masks are mirrored in convolution. Image ~~deriv~~ differentiation filters ~~taking the~~ derivative of an image can be used to identify certain

F is intended to detect horizontal edges in image. It reacts strongly to horizontal egdes when performing shifting operation from dark to light and light to dark. It emphasizes features like horizontal boundaries and edges.

(d) $F' = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$

Hence, $F(0,1)$, $F(1,1)$ and $F(2,1)$ are 0 [according to formula]

$$G(1,1) = F(0,0)\,I'(1,1) + F(0,2)\,I'(1,3) + F(1,0)\,I'(2,1) + F(1,2)\,I'(2,3) + F(2,0)\,I'(3,1) + F(2,2)\,I'(3,3)$$
$$= 4 + 5 = 9$$

$$G(1,2) = F(0,0)\,I'(1,2) + F(0,2)\,I'(1,4) + F(1,0)\,I'(2,2) + F(1,2)\,I'(2,4) + F(2,0)\,I'(3,2) + F(2,2)\,I'(3,4)$$
$$= -7 - 8 + 1 + 2 = -12$$

$$G(1,3) = F(0,0)\,I'(1,3) + F(0,2)\,I'(1,5) + F(1,0)\,I'(2,3) + F(1,2)\,I'(2,5) + F(2,0)\,I'(3,3) + F(2,2)\,I'(3,5)$$
$$= -4 - 5 = -9$$

$$G(2,1) = F(0,0)\,I'(2,1) + F(0,2)\,I'(2,3) + F(1,0)\,I'(3,1) + F(1,2)\,I'(3,3) + F(2,0)\,I'(4,1) + F(2,2)\,I'(4,3)$$
$$= 4 + 5 + 6 = 15$$

$$F(2,2) = F(0,0)\,I'(2,2) + F(0,2)\,I'(2,4) + F(1,0)\,I'(3,2) + F(1,2)\,I'(3,4) + F(2,0)\,I'(4,2) + F(2,2)\,I'(4,4)$$
$$= -7 + 1 - 8 + 2 - 9 + 3 = -18$$

$$G(2,3) = F(0,0)\,I'(2,3) + F(0,2)\,I'(2,5) + F(1,0)\,I'(3,3)$$
$$+ F(1,2)\,I'(3,5) + F(2,0)\,I'(4,3) + F(2,2)\,I'(4,5)$$
$$= -4 - 5 - 6 = -15$$

$$G(3,1) = F(0,0)\,I'(3,1) + F(0,2)\,I'(3,3) + F(1,0)\,I'(4,1)$$
$$+ F(1,2)\,I'(4,3) + F(2,0)\,I'(5,1) + F(2,2)\,I'(5,3)$$
$$= \cancel{8}\ 5 + 6 = 11$$

$$G(3,2) = F(0,0)\,I'(3,2) + F(0,2)\,I'(3,4) + F(1,0)\,I'(4,2)$$
$$+ F(1,2)\,I'(4,4) + F(2,0)\,I'(5,2) + F(2,2)\,I'(5,4)$$
$$= -8 - 9 + 3 + 2 = -12$$

$$G(3,3) = F(0,0)\,I'(3,3) + F(0,2)\,I'(3,5) + F(1,0)\,I'(4,3)$$
$$+ F(1,2)\,I'(4,5) + F(2,0)\,I'(5,3) + F(2,2)\,I'(5,5)$$
$$= -5 - 6 = -11$$

$$G = \begin{bmatrix} 9 & -12 & -9 \\ 15 & -18 & 15 \\ 11 & -12 & -11 \end{bmatrix}$$

Now in filter $F = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$ and

$F' = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$ have difference in edge

detection.

horizontal
F is intend'd to detect
It reacts to horizontal where
edges.
a shift from dark to light. F'
a transition
detect vertical changes where
from drak to light or vice versa. It
Looks the area in vertical direction.

F and F' are designed for edge
detection for defined operation. those are
detect edges in their direction and contributing
to different aspect of image analysis.

(e) $F = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

$$G(1,1) = F(0,0)\,I'(1,1) + F(0,1)\,I'(0,2) + F(0,2)\,I'(0,3)$$
$$+ F(1,0)\,I'(2,1) + F(1,1)\,I'(2,2) + F(1,2)\,I'(2,3)$$
$$+ F(2,0)\,I'(3,1) + F(2,1)\,I'(3,2) + F(2,2)\,I'(3,3)$$
$$= \frac{1}{16}(\cancel{8}\ 4\cdot 7 + 2\cdot 4 + 2\cdot \cancel{8} + 5)$$
$$= \frac{1}{16} \times 57 = 3.5$$

$$G(1,2) = F(0,0)\,I'(1,2) + F(0,1)\,I'(1,3) + F(0,2)\,I'(1,4)$$
$$+ F(1,0)\,I'(2,2) + F(1,1)\,I'(2,3) + F(1,2)\,I'(2,4)$$
$$+ F(2,0)\,I'(3,2) + F(2,1)\,I'(3,3) + F(2,2)\,I'(3,4)$$
$$= \frac{1}{16}(2\cdot 7 + 4\cdot 4 + 2\cdot 1 + 1\cdot 8 + 2\cdot 5 + 1\cdot 2)$$
$$= \frac{1}{16} \times 52 = 3.25$$

$$G(1,3) = F(0,0)\,I'(1,3) + F(0,1)\,I'(1,4) + F(0,2)\,I'(1,5)$$
$$+ F(1,0)\,I'(2,3) + F(1,1)\,I'(2,4) + F(1,2)\,I'(2,5)$$
$$+ F(2,0)\,I'(3,3) + F(2,1)\,I'(3,4) + F(2,2)\,I'(3,5)$$
$$= \frac{1}{16}(2\cdot 4 + 4\cdot 1 + 1\cdot 5 + 2\cdot 2)$$
$$= \frac{1}{16}\cdot 21 = 1.31$$

$$G(2,1) = F(0,0)\,I'(2,1) + F(0,1)\,I'(2,2) + F(0,2)\,I'(2,3)$$
$$+ F(1,0)\,I'(3,1) + F(1,1)\,I'(3,2) + F(1,2)\,I'(3,3)$$
$$+ F(2,0)\,I'(4,1) + F(2,1)\,I'(4,2) + F(2,2)\,I'(4,4)$$
$$= \frac{1}{16}(2.7 + 1.4 + 4.8 + 2.5 + 2.9 + 1.6)$$
$$= \frac{1}{16} \cdot 84 = 5.25$$

$$G(2,2) = F(0,0)\,I'(2,2) + F(0,1)\,I'(2,3) + F(0,2)\,I'(2,4)$$
$$+ F(1,0)\,I'(3,2) + F(1,1)\,I'(3,3) + F(1,2)\,I(3,4)$$
$$+ F(2,0)\,I(4,2) + F(2,1)\,I(4,3) + F(2,2)\,I'(4,4)$$
$$= \frac{1}{16}(7 + 2.4 + 1 + 2.8 + 4.5 + 2.2 + 1.9 + 2.6 + 1.3) = \frac{1}{16} \cdot 80 = 5$$

$$G(2,3) = F(0,0)\,I'(2,3) + F(0,1)\,I'(2,4) + F(0,2)\,I'(2,5)$$
$$+ F(1,0)\,I'(3,3) + F(1,1)\,I'(3,4) + F(1,2)\,I'(3,5)$$
$$+ F(2,0)\,I'(4,3) + F(2,1)\,I'(4,4) + F(2,2)\,I(4,5)$$
$$= \frac{1}{16}(4 + 2 + 10 + 8 + 6 + 6) = \frac{1}{16} \cdot 36$$
$$= 2.25$$

$$G(3,1) = F(0,0)\,I'(3,1) + F(0,1)\,I'(3,2) + F(0,2)\,I'(3,3)$$
$$+ F(1,0)\,I'(4,1) + F(1,1)\,I(4,2) + F(1,2)\,I'(4,3)$$
$$+ F(2,0)\,I'(5,1) + F(2,1)\,I'(5,2) + F(2,2)\,I'(5,3)$$
$$= \frac{1}{16}(2.8 + 1.5 + 4.9 + 2.5\cancel{7})$$
$$= \frac{1}{16} \cdot 69 = 4.31$$

$$G(3,2) = F(0,0)\,I'(3,2) + F(0,1)\,I'(3,3) + F(0,2)\,I'(3,4) + F(1,0)\,I'(4,2) + F(1,1)\,I'(4,3) + F(1,2)\,I'(4,4) + F(2,0)\,I'(5,2) + F(2,1)\,I'(5,3) + F(2,2)\,I'(5,4)$$

$$= \frac{1}{16}(8 + 10 + 2 + 2.9 + 4.6 + 2.3)$$

$$= \frac{1}{16} \cdot 68 = 4.25$$

$$G(3,3) = F(0,0)\,I'(3,3) + F(0,1)\,I'(3,4) + F(0,2)\,I'(3,5) + F(1,0)\,I'(4,3) + F(1,1)\,I(4,4) + F(1,2)\,I'(4,5) + F(2,0)\,I'(5,3) + F(2,1)\,I'(5,4) + F(2,2)\,I'(5,5)$$

$$= \frac{1}{16} \cdot (5 + 4 + 2.6 + 4.3)$$

$$= \frac{1}{16} \cdot 33 = 2.06$$

$$G = \begin{bmatrix} 3.5 & 3.25 & 1.31 \\ 5.25 & 5 & 2.25 \\ 4.31 & 4.25 & 2.06 \end{bmatrix}$$

The filter $F = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$ is

Gaussian smoothing filter.

It is working as smoothing and blurring tool. It is designed to smooth image by averaging each pixel. It reduce high frequency noise. and create a less detailed appearance. The normalization factor $\frac{1}{16}$ ensure the smoothing is done in a way which preserve the overall brightness of input image.

(f) $F = \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$

$$G(1,1) = F(0,0)\,\bar{I}(1,1) + F(0,1)\,\bar{I}(1,2) + F(0,2)\,\bar{I}(1,3)$$
$$+ F(1,0)\,\bar{I}(2,1) + F(1,1)\,\bar{I}(2,2) + F(1,2)\,\bar{I}(2,3)$$
$$+ F(2,0)\,\bar{I}(3,1) + F(2,1)\,\bar{I}(3,2) + F(2,2)\,\bar{I}(3,3)$$
$$= \frac{1}{9}(0 + 0 + 0 + 0 + 7 + 4 + 8 + 5)$$
$$= \frac{1}{9} \cdot 24 = 2.67$$

$$G(1,2) = F(0,0)\,\bar{I}(1,2) + F(0,1)\,\bar{I}(1,3) + F(0,2)\,\bar{I}(1,4)$$
$$+ F(1,0)\,\bar{I}(2,2) + F(1,1)\,\bar{I}(2,3) + F(1,2)\,\bar{I}(2,4)$$
$$+ F(2,0)\,\bar{I}(3,2) + F(2,1)\,\bar{I}(3,3) + F(2,2)\,\bar{I}(3,4)$$
$$= \frac{1}{9}(7 + 4 + 1 + 8 + 5 + 2) = \frac{1}{9} \times 27 = 3$$

$$G(1,3) = F(0,0)\,\bar{I}(1,3) + F(0,1)\,\bar{I}(1,4) + F(0,2)\,\bar{I}(1,5)$$
$$+ F(1,0)\,\bar{I}(2,3) + F(1,1)\,\bar{I}(2,4) + F(1,2)\,\bar{I}(2,5)$$
$$+ F(2,0)\,\bar{I}(3,3) + F(2,1)\,\bar{I}(3,4) + F(2,2)\,\bar{I}(3,5)$$

$$= \frac{1}{9}(4 + 1 + 5 + 2) = \frac{1}{9} \times 12$$

$$= 1.33$$

$G(2,1) = F(0,0)\,I'(2,1) + F(0,1)\,I'(2,2) + F(0,2)\,I'(2,3)$
$+ F(1,0)\,I'(3,1) + F(1,1)\,I'(3,2) + F(1,2)\,I'(3,3)$
$+ F(2,0)\,I'(4,1) + F(2,1)\,I'(4,2) + F(2,2)\,I'(4,3)$

$= \frac{1}{9}(7 + 4 + 8 + 5 + 9 + 6) = \frac{1}{9} \times 39$

$= 4.33$

$G(2,2) = F(0,0)\,I'(2,2) + F(0,1)\,I'(2,3) + F(0,2)\,I'(2,4)$
$+ F(1,0)\,I'(3,2) + F(1,1)\,I'(3,3) + F(1,2)\,I'(3,4)$
$+ F(2,0)\,I'(4,2) + F(2,1)\,I'(4,3) + F(2,2)\,I'(4,4)$

$= \frac{1}{9}(7 + 4 + 1 + 8 + 5 + 2 + 9 + 6 + 3)$

$= \frac{1}{9} \cdot 45 = 5$

$G(2,3) = F(0,0)\,I'(2,3) + F(0,1)\,I'(2,4) + F(0,2)\,I'(2,5)$
$+ F(1,0)\,I'(3,3) + F(1,1)\,I'(3,4) + F(1,2)\,I'(3,5)$
$+ F(2,0)\,I'(4,3) + F(2,1)\,I'(4,4) + F(2,2)\,I'(4,5)$

$= \frac{1}{9}(4 + 1 + 5 + 2 + 6 + 3) = \frac{1}{9} \times 21$

$= 2.33$

$G(3,1) = F(0,0)\,I'(3,1) + F(0,1)\,I'(3,2) + F(0,2)\,I'(3,3)$
$+ F(1,0)\,I'(4,1) + F(1,1)\,I'(4,2) + F(1,2)\,I'(4,3)$
$+ F(2,0)\,I'(5,1) + F(2,1)\,I'(5,2) + F(2,2)\,I'(5,3)$

$= \frac{1}{9}(0 + 8 + 5 + 0 + 9 + 6 + 0 + 0 + 0)$

$= \frac{1}{9} \times 28 = 3.11$

$$G(3,2) = F(0,0)I(3,2) + F(0,1)I(3,3) + f(0,2)I(3,4)$$
$$+ f(1,0)I(4,2) + F(1,1)I(4,3) + f(1,2)I(4,4)$$
$$+ f(2,0)I(5,2) + f(2,1)I(5,3) + f(2,2)I(5,4)$$
$$= \frac{1}{9}(8+5+2+9+6+3+0+0+0)$$
$$= \frac{1}{9} \times 33 = 3.67$$

$$G(3,3) = f(0,0)I(3,3) + F(0,1)I(3,4) + f(0,2)I(3,5)$$
$$+ f(1,0)I(4,3) + f(1,1)I(4,4) + F(1,2)I(4,5)$$
$$+ f(2,0)I(5,3) + F(2,1)I(5,4) + F(2,2)I(5,5)$$
$$= \frac{1}{9}(5+2+0+6+3+0+0+0+0)$$
$$= \frac{1}{9} \times 16 = 1.77$$

$$G = \begin{bmatrix} 2.67 & 3 & 1.33 \\ 4.33 & 5 & 2.33 \\ 3.11 & 3.67 & 1.77 \end{bmatrix}$$

~~Hene~~ the filter of (e) is gaussian smoothing filter and (f) is ~~aven~~ Average moving filter.

Gaussian smoothing filter used for smoothing or bluring image. And Average moving filter are used for basic smoothing through ~~basic~~ pixel averaging. Gaussian smoothing emphasizes the central pixel, gradually diminishing weights toward the edges in a gausiian distribution. that filter ~~effective~~ for reduce noise and maintain image details. Basic smoothing simplifies the smoothing and less focus on preserving details. It applies uniform weights to all pixels and results a straightforward average of neighboring values without a selective emphasis.

Ans to the Ques No 2

(a)

According to formula, the correlation defined as,

$$G(i,j) = \sum_{u=0}^{k-1} \sum_{v=0}^{l-1} F(u,v) \cdot \bar{I}(i+u, j+v)$$

where, F is the filter, $\bar{I} \in \mathbb{R}^{(m+k-1) \times (n+l-1)}$ is the original image, I, padded with zeros along it's edges.

Now, we can define filter F as a vector representation of f,

$$f = \text{vector}(F)$$

Also, we can represent $\bar{I}(i,j)$ as $t(i,j)$ which is the vector representation of neighborhood patch of images.

$$t(i,j) = \text{vector}(\bar{I}(i-1 : i+2,\ j-1 : j+2))$$

We have to show that. We can write correlation as a vector dot product,

$$G(i,j) = f^T t_{ij}$$

As we apply vectorization which turns a matrix into a single column, we have to apply dot product. For performing that dot product we need to use the neighborhood ~~path~~ patch as the dot product of a row vector and a column vector can be expressed as a multiplication. ~~of~~ The correlation operation involves corresponding element in neighborhood and summing the result.

$G(i,j) = f^T t_{ij}$ reshaping f and neighborhood path $\bar{I}(i,j)$. finally we can say that, correlation operation can perform as a vector dot product.

$$f = [f(0,0) \ldots f(u,v)] \qquad t_{ij} = \begin{bmatrix} \bar{I}(i,j) \\ \bar{I}(i,j+1) \\ \vdots \\ \bar{I}(i+u, j+v) \end{bmatrix}$$

2(b)

In [1]:
```
#import library function
import numpy as np
from skimage import io
import matplotlib.pyplot as plt
```

Vector Representation of Filter

In [2]:
```
# Given filter F
F = np.array([1, 1, 1, 0, 0, 0, -1, -1, -1])
# Flatten the filter
F_flat = F.flatten()
```

Vector Representation of the Neighborhood Patch

In [3]:
```
# Given image I' which is maintaining zero-padding
image_patch = np.array([
    [0, 0, 0, 0, 0],
    [0, 7, 4, 1, 0],
    [0, 8, 5, 2, 0],
    [0, 9, 6, 3, 0],
    [0, 0, 0, 0, 0]
])

```

In [4]:
```
# Initialize a matrix to store correlation results
correlation_matrix = np.zeros((3, 3))
```

Calculate correlation for every valid position

Extract the neighborhood patch at position (i, j)

Flatten the neighborhood patch and calculate the dot product

In [5]:
```
for i in range(1, 4):
    for j in range(1, 4):

        neighborhood_patch = image_patch[i - 1:i + 2, j - 1:j + 2]
        correlation_matrix[i - 1, j - 1] = np.dot(F_flat, neighborhood_

# Print the correlation matrix
print("Correlation matrix:")
print(correlation_matrix)
```

```
Correlation matrix:
[[-13. -15.  -7.]
 [ -4.  -6.  -4.]
 [ 13.  15.   7.]]
```

Test code using the provided image and the filters.

In [6]:

```

def correlation_filter_image_dot_product(F, I):
    # implement zero-padding to maintain correct sizes
    padded_I = np.pad(I, ((1, 1), (1, 1)), mode='constant', constant_va

    # reshape filter F to a column vector
    f = F.flatten()

    # extract patches from the image using array slicing
    patches = np.lib.stride_tricks.sliding_window_view(padded_I, (3, 3)

    # reshape patches to 1D arrays
    patches_1d = patches.reshape(-1, 9)

    # computing G(i,j) for all patches using dot product
    G = np.dot(patches_1d, f)

    # reshape G to the original image shape
    G = G.reshape(I.shape)

    return G

# Load the image
image_path = 'parrot.png'
image = io.imread(image_path)

# Define filters
a = np.array([[0, 0, 0], [0, 1, 0], [0, 0, 0]])
b = np.array([[1, 0, 0], [0, 0, 0], [0, 0, 0]])
c = np.array([[1, 1, 1], [0, 0, 0], [-1, -1, -1]])
d = np.array([[-1, 0, 1], [-1, 0, 1], [-1, 0, 1]])
e = (1/16) * np.array([[1, 2, 1], [2, 4, 2], [1, 2, 1]])
f = (1/9) * np.array([[1, 1, 1], [1, 1, 1], [1, 1, 1]])

# Filter a
result_a = correlation_filter_image_dot_product(a, image)
# Filter b
result_b = correlation_filter_image_dot_product(b, image)
# Filter c
result_c = correlation_filter_image_dot_product(c, image)
# Filter d
result_d = correlation_filter_image_dot_product(d, image)
# Filter e
result_e = correlation_filter_image_dot_product(e, image)
# Filter f
result_f = correlation_filter_image_dot_product(f, image)
# Display original image and the results for all filters

plt.figure(figsize=(40, 30))
plt.subplot(3, 3, 1)
plt.imshow(image, cmap='gray')
plt.title('Original Image', fontsize = 40)

plt.subplot(3, 3, 2)
plt.imshow(result_a, cmap='gray')
plt.title('Filter (a) - No change', fontsize = 40)

plt.subplot(3, 3, 3)
plt.imshow(result_b, cmap='gray')
plt.title('Filter (b) - Shift', fontsize = 40)

```

```python
plt.subplot(3, 3, 4)
plt.imshow(result_c, cmap='gray')
plt.title('Filter (c) - Horizontal Edge', fontsize = 40)

plt.subplot(3, 3, 5)
plt.imshow(result_d, cmap='gray')
plt.title('Filter (d) - Vertical Edge', fontsize = 40)

plt.subplot(3, 3, 6)
plt.imshow(result_e, cmap='gray')
plt.title('Filter (e) - Gaussian Smoothing', fontsize = 40)

plt.subplot(3, 3, 7)
plt.imshow(result_f, cmap='gray')
plt.title('Filter (f) - Average Moving', fontsize = 40)

plt.show()
```