

# INDEX

- ① Introduction to Matlab
- ② Intro with hold on, hold off and subplot.
- ③ Read the image and perform the TCC and FCC
- ④ Perform the checkerboard with & without function
- ⑤ Flip the image with and without function
- ⑥ Perform transformation functions.
- ⑦ Perform ROI / gray level thresholding.
- ⑧ perform bit plane slicing
- ⑨ Perform alternate pixel 0 and alternate row and column 0
- ⑩ Perform Average, weighted Average and Laplacian filter
- ⑪ perform Order Statistics
- ⑫ perform Histogram equalization
- ⑬ perform Morphology
- ⑭ perform Line filter
- ⑮ Perform Sobel filter

## EXPERIMENT-1: INTRODUCTION TO MATLAB

Matlab is a software package

# commands to create 1-D array:-

$a = [1, 2, 3, 4, 5]$        $b = [1, 2, 3, 4, 5]$

$c = [1:5]$

output of a, b, c matrix:  $a = 1 \ 2 \ 3 \ 4 \ 5$   
 $b = 1 \ 2 \ 3 \ 4 \ 5$   
 $c = 1 \ 2 \ 3 \ 4 \ 5$

$e = [1:10:2]$  ,  $c = [1, 3, 5, 7, 9]$

# command to create 1-D array using step-wise:-

$d = \text{start} : \text{step} : \text{end}$

Example:-  $d = 1 : 0.2 : 10$

output:-  $d = [1, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6, 2.8, 3.0, 3.2, 3.4, 3.6, 3.8, 4.0, 4.2, 4.4, 4.6, 4.8, 5.0]$

$e = 1 : 0.5 : 5$

Output:- 1.0000 1.5000 2.0000 2.5000 3.0000 3.5000  
4.0000 4.5000 5.0000

# command to create 2-D array:-

1.  $a = [1, 2, 3; 4, 5, 6]$

output:-  $a = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$

2. Magic command:- Magic(a) command is used to create a matrix of size  $a \times a$  with random numbers.

Example:-  $\text{mag-magic}(2)$

output:-  $\begin{bmatrix} 1 & 5 \\ 7 & 10 \end{bmatrix}$



Zero command  $\rightarrow$  Zero command is used to create a matrix with 0's

Example:- zeros(2)

output:  $\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$

zeros(2,3)

output:  $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

ones command  $\rightarrow$  one command is used to create a matrix with 1's.

Example:- ones(2)

output:  $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$

ones(2,3)

output:  $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

command to multiply two matrices:-

$C = A * B$  where  $A$  and  $B$  are two matrices of size  $m \times n$  and  $n \times p$  respectively. then, we get a matrix of size  $m \times p$ .

Example:  $A = \text{ones}(2)$  ;  $B = \text{ones}(2)$  ,  $C = A * B$

output:  $\begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$

Scalar operation:- It is used to multiply index wise

Ex:  $A = \text{ones}(2)$

$B = \text{ones}(2)$

$C = A .* B$

output:  $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$

# command to square a matrix:- Let  $A$  be a matrix of size  $m \times n$

(1)  $A * A$  or  $A^2 \rightarrow$  used to square index wise element.

Example:- let  $A = [1, 2; 3, 4]$

$A * A$

output:  $\begin{bmatrix} 5 & 10 \\ 15 & 16 \end{bmatrix}$

# command to read an image  $\rightarrow$  `ii = imread('Path. format');`

# command to show an image  $\rightarrow$

`imshow(ii);`

`clc` = command to clear screen clear space all  $\Rightarrow$  clear the workspace  
clear (space  $C$ )  $\Rightarrow$  Clear the variable 'c' from workspace

## Intro with hold on, hold off and subplot

hold  $\rightarrow$  Retain current plot when adding new plots.

hold on  $\rightarrow$  hold on retains plots in the current axes. so that new plots added to the axes do not delete existing plots.

hold off  $\rightarrow$  hold off sets the hold state to off so that new plots added to the axes clear existing plots and reset all axes properties.

```
x = linspace(-pi, pi);
```

```
y1 = sin(x);
```

```
plot(x, y1);
```

```
hold on
```

```
y2 = cos(x);
```

```
plot(x, y2);
```

```
hold off
```

Subplot function subplot(m,n,p) divides the current figure into an m by n grid and creates axes in the position specified by p.

```
i1 = imread('cameraman.tif');
```

```
subplot(2,2,1);
```

```
imshow(i1);
```

```
subplot(2,2,2);
```

```
imshow(i1);
```

Read the image and perform the TCC and FCC

TCC → True colour composite

Red band-Red, Green band- Green, blue band- blue

FCC → false colour composite

Any other combination of colours.

Cat command: It is used to concatenate two or more commands.

```
i1 = imread('C:\users\CL2351\pictures\saved  
pictures\glau.jpg');
```

```
b1 = i1(:, :, 1);
```

```
b2 = i1(:, :, 2);
```

```
b3 = i1(:, :, 3);
```

```
subplot(2,2,1);
```

```
imshow(i1);
```

```
subplot(2,2,2);
```

```
imshow(b1);
```

```
subplot(2,2,3);
```

```
imshow(b2);
```

```
subplot(2,2,4);
```

```
imshow(b3);
```

```
tcc = cat(3, b1, b2, b3);
```

```
imshow(tcc);
```

```
fcc = cat(3, b3, b1, b2);
```

```
imshow(fcc);
```

Implement the checkerboard effect

```
i1 = zeros (256, 256);  
for i = 1:256  
    for j = 1:256  
        if (i == j)  
            i1(i,j) = 0;  
        elseif (mod(j,2) == 0) & (mod(i,2) == 0)  
            i1(i,j) = 0;  
        elseif (mod(j,2) == 0 || mod(i,2) == 0)  
            i1(i,j) = 1;  
        end  
    end  
end  
i1 = cast(i1, "uint8");  
imshow(i1);
```



Flip the image with function and without function  
with function →

```
i1 = imread('cameraman.tif');  
i2 = flip(i1, 1);  
i3 = flip(i1, 2);  
subplot(1, 3, 1);  
imshow(i1);  
subplot(1, 3, 2);  
imshow(i2);  
subplot(1, 3, 3);  
imshow(i3);
```

without function →

```
i1 = imread('cameraman.tif');  
i2 = zeros(256);  
for i = 1:256  
    i2(257-i, :) = i1(i, :);  
end  
i2 = cast(i2, 'uint8');  
subplot(1, 2, 1);  
imshow(i1);  
subplot(1, 2, 2);  
imshow(i2);
```


## Perform transformation functions

```
i1 = imread('cameraman.tif');  
i2 = i1  
subplot(1,4,1);  
imshow(i1);  
for i = 1:256  
    for j = 1:256  
        i2(i,j) = 255 - i2(i,j);  
    end  
end  
subplot(1,4,2);  
imshow(i2);  
i1 = cast(i1, 'double');  
i2 = i1;  
for i = 1:256  
    for j = 1:256  
        i2(i,j) = 100 * log(1 + i2(i,j));  
    end  
end  
i2 = cast(i2, 'uint8');  
subplot(1,4,3);  
imshow(i2);  
i2 = i1;  
for i = 1:256  
    for j = 1:256  
        i2(i,j) = 20 * i2(i,j)^0.8;  
    end  
end  
i2 = cast(i2, 'uint8');  
subplot(1,4,4);  
imshow(i2);
```



Perform ROI / gray level thresholding

```
i1 = imread('cameraman.tif');  
i2 = i1;  
for i = 1:256  
    for j = 1:256  
        if i2(i,j) < 10 || i2(i,j) > 20  
            i2(i,j) = 255;  
        end  
    end  
end  
subplot(1,2,1);  
imshow(i1);  
subplot(1,2,2);  
imshow(i2);
```



## Perform Bitplane slicing

```
i1 = imread('cameraman.tif');  
subplot(1, 9, 1);  
imshow(i1);  
p1 = bitget(i1, 1);  
p2 = bitget(i1, 2);  
p3 = bitget(i1, 3);  
p4 = bitget(i1, 4);  
p5 = bitget(i1, 5);  
p6 = bitget(i1, 6);  
p7 = bitget(i1, 7);  
p8 = bitget(i1, 8);  
subplot(1, 9, 2);  
p1 = cast(p1, 'logical');  
imshow(p1);  
subplot(1, 9, 3);  
p2 = cast(p2, 'logical');  
imshow(p2);  
subplot(1, 9, 4);  
p3 = cast(p3, 'logical');  
imshow(p3);  
subplot(1, 9, 5);  
p4 = cast(p4, 'logical');  
imshow(p4);  
subplot(1, 9, 6);  
p5 = cast(p5, 'logical');  
imshow(p5);  
subplot(1, 9, 7);  
p6 = cast(p6, 'logical');  
imshow(p6);  
subplot(1, 9, 8);  
p7 = cast(p7, 'logical');  
imshow(p7);
```

```
subplot (1,9,9);
```

```
p8 = cast (p8, 'logical');
```

```
imshow (p8);
```



Perform alternate Pixel 0 and alternate row  
and column 0

```
il = imread('cameraman.tif');
```

```
for i = 1:256
```

```
    for j = 1:256
```

```
        if mod(i,2) == 0
```

```
            il(i,j) = 0;
```

```
        end
```

```
        if mod(j,2) == 0
```

```
            il(i,j) = 0;
```

```
        end
```

```
    end
```

```
end
```

```
imshow(il);
```

## Perform Average Filter

```
i1 = imread('cameraman.tif')
i2 = cast(i1, 'double');
f = ones(3);
i3 = i2
[m, n] = size(i1);

for i = 2:m-1
    for j = 2:n-1
        sum = 0
        for k = 1:3
            for l = 1:3
                sum = sum + i2(i-2+k, j-2+l) *
                    f(k, l)
            end
        end
        i3(i, j) = sum/9;
    end
end

i3 = cast(i3, 'uint8');
subplot(1, 2, 1); imshow(i1);
subplot(1, 2, 2);
imshow(i3);
```

## Perform Weighted Average Filter

```
i1 = imread('cameraman.tif')
f = [1,2,1; 2,4,2; 1,2,1];
i2 = cast(i1, 'double');
i3 = i2; [m,n] = size(i1);
for i = 2:m-1
    for j = 2:n-1
        sum = 0;
        for k = 1:3
            for l = 1:3
                sum = sum + i2(i-2+k, j-2+l) *
                    f(k,l);
            end
        end
        i3(i,j) = sum/16;
    end
end
i3 = cast(i3, 'uint8');
subplot(1,2,1);
imshow(i1);
subplot(1,2,2);
imshow(i3);
```



## Perform Laplacian Filter

```
i1 = imread('cameraman.tif');  
f = [0, -1, 0; -1, 4, -1; 0, -1, 0];  
i2 = cast(i1, 'double');  
[m, n] = size(i1);  
for i = 2:m-1  
    for j = 2:n-1  
        sum = 0  
        for k = 1:3  
            for l = 1:3  
                sum = sum + i2(i-2+k, j-2+l) *  
                    f(k, l)  
            end  
        end  
        i3(i, j) = sum;  
    end  
end  
i3 = cast(i3, 'uint8');  
subplot(1, 2, 1);  
imshow(i1);  
subplot(1, 2, 2);  
imshow(i3);
```

## Perform Order Statistics

### 1. Min Filter:-

```
i1 = imread('cameraman.tif');  
[m,n] = size(i1);  
i3 = zeros(m,n);  
for i = 2:255  
    for j = 2:255  
        i3(i,j) = min(min(i1(i-1:i+1, j-1:j+1)));  
    end  
end  
subplot(1,2,1); imshow(i1);  
subplot(1,2,2); imshow(i3);
```

### 2. Max Filter

```
i1 = imread('cameraman.tif');  
[m,n] = size(i1);  
i3 = zeros(m,n);  
for i = 2:255  
    for j = 2:255  
        i3(i,j) = max(max(i1(i-1:i+1, j-1:j+1)));  
    end  
end  
subplot(1,2,1);  
imshow(i1);  
subplot(1,2,2);  
imshow(i3);
```

### 3. Median Filter

```
i1 = imread('cameraman.tif');  
[m,n] = size(i1);  
i3 = zeros(m,n);  
for i = 2:255  
    for j = 2:255  
        i3(i,j) = median(median(i1(i-1:i+1, j-1:j+1)));  
    end  
end  
subplot(1,2,1);  
imshow(i1);  
subplot(1,2,2);  
imshow(i3);
```



## Histogram

```
i1 = imread('cameraman.tif');  
for i = 1:256  
    his(i,1) = i-1;  
end  
for k = 1:256  
    C = 0;  
    for i = 1:256  
        for j = 1:256  
            if i1(i,j) == k-1  
                C = C+1;  
            end  
        end  
        his(k,2) = C;  
    end  
    for i = 1:256  
        his(i,3) = his(i,2) / 65536;  
    end  
    his(1,4) = his(1,3)  
    for i = 2:256  
        his(i,4) = his(i-1,4) + his(i,3)  
    end  
    for i = 1:256  
        his(i,5) = his(i,4) * 255  
    end  
    for i = 1:256  
        his(i,6) = round(his(i,5))  
    end  
    for i = 1:256  
        his_eq(i,1) = i-1;  
    end  
    for i = 1:256  
        CC = 0  
        for j = 1:256  
            if his_eq(i,1) == his(j,6)  
                CC = CC + his(j,2);  
            end  
        end  
    end
```

```

        hiseq(i,2) = cc;
    end
    subplot(2,2,1);
    bar(his(:,1), his(:,2), 0.255);
    subplot(2,2,2);
    bar(hiseq(:,2), hiseq(:,2));
    temp2 = i1;
    for i = 1:256
        for j = 1:256
            m = temp2(i,j);
            for k = 1:256
                if m == his(k,2)
                    temp2(i,j) = his(k,6);
                end
            end
        end
    end
    subplot(2,2,3);
    imshow(i1);
    subplot(2,2,4);
    imshow(temp2);

```

# Morphology

```
i1 = imread('cameraman.tif');  
i2 = imshow(i1);  
SC = strel('line', i1, 90);  
eroded = imerode(i2, SC);  
dilated = imdilate(i2, SC);  
open = imdilate(eroded, SC);  
closing = imdilate(dilated, SC);  
for i = 1:255  
    for j = 1:255  
        boundary(i,j) = i2(i,j) - eroded(i,j);  
    end  
end  
num = zeros(size(i2));  
i2invert = invertim(i2);  
w = ones(size(SC));  
[m,n] = size(SC);  
for i = 1:m  
    for j = 1:n  
        w(i,j) = w(i,j) - SC(i,j);  
    end  
end  
iminver = imerode(i2invert, w);  
for i = 1:255  
    for j = 1:255  
        if eroded(i,j) == iminver(i,j)  
            num(i,j) = eroded(i,j);  
        end  
    end  
end  
end
```



```
subplot (1,7,1);  
imshow (i2);  
subplot (1,7,2);  
imshow (eroded);  
subplot (1,7,3);  
imshow (dilated);  
subplot (1,7,4);  
imshow (open);  
subplot (1,7,5);  
imshow (closing);  
subplot (1,7,6);  
imshow (boundary);  
subplot (1,7,7);  
imshow (num);
```

## Line filter

```
i1 = imread('cameraman.tif');
```

```
i2 = im2bw(i1);
```

```
f1 = [-1,-1,-1; 2,2,2; -1,-1,-1];
```

```
f2 = [-1,-1,2; -1,2,-1; 2,-1,-1];
```

```
f3 = [-1,2,-1; -1,2,-1; -1,2,-1];
```

```
f4 = [2,-1,-1; -1,2,-1; -1,-1,2];
```

```
[m,n] = size(i2);
```

```
i3 = zeros(m+2, n+2);
```

```
i3(2:m+1, 2:n+1) = i2;
```

```
for i = 2 : m-1
```

```
    for j = 2 : n-1
```

```
        sum1 = 0;
```

```
        sum2 = 0;
```

```
        sum3 = 0;
```

```
        sum4 = 0;
```

```
        for k = 1 : 3
```

```
            for l = 1 : 3
```

```
                a1 = i2(i-2+k, j-2+l) * f1(k,l);
```

```
                sum1 = sum1 + a1;
```

```
                a2 = i2(i-2+k, j-2+l) * f2(k,l);
```

```
                sum2 = sum2 + a2;
```

```
                a3 = i3(i-2+k, j-2+l) * f3(k,l);
```

```
                sum3 = sum3 + a3;
```

```
                a4 = i4(i-2+k, j-2+l) * f4(k,l);
```

```
                sum4 = sum4 + a4;
```

```
            end
```

```
        end
```

```
        sum = [sum1 sum2 sum3 sum4];
```

```
        i3(i,j) = max(sum);
```

```
    end
```

```
subplot(1,2,1); imshow(i2);
```

```
subplot(1,2,2); imshow(i3);
```

## Sobel Filter

```
i1 = imread('cameraman.tif');  
i2 = im2bw(i1);  
sn = [-1, -2, -1; 0, 0, 0; 1, 2, 1];  
sv = [-1, 0, 1; -2, 0, -2; -1, 0, 1];  
[m, n] = size(i1); i3 = zeros(m+2, n+2);  
for i = 2:m-1  
    for j = 2:n-1  
        sum1 = 0;  
        sum2 = 0;  
        for k = 1:3  
            for l = 1:3  
                a1 = i2(i-2+k, j-2+l) + sn(k, l);  
                sum1 = sum1 + a1;  
                a2 = i2(i-2+k, j-2+l) + sv(k, l);  
                sum2 = sum2 + a2;  
            end  
        end  
        sum = abs(sum1) + abs(sum2);  
        if (sum > 3)  
            i3(i, j) = sum;  
        end  
    end  
end  
end  
subplot(1, 2, 1);  
imshow(i2);  
subplot(1, 2, 2);  
imshow(i3);
```