

INDEX

<u>Exp no.</u>	<u>Experiment Name</u>	<u>Page no.</u>
1.	Intro with MATLAB command	1-2
2.	Intro with hold on, hold off & subplot	3
3.	Read the image & perform Tcc & Ecc	4
4.	perform alternate pixel col alt. row & col.	5
5.	perform Checkboard effect	6
6.	flip the image function	7
7.	perform transformation function	8
8.	perform ROI / grey level slicing thresh.	9
9.	perform Bit plane slicing	10
10.	Average filter, Weighted Average filter	11-13
11.	Laplacian filter - Derivative	
12.	Min, Max & Median filter	14-16
13.	morphology	17-18
14.	Histogram	19-20
15.	Line filter	20-21
16.	Sobel filter	22-23

#5

201500280(35)

Introduction

MATLAB is a software package for high-performance mathematical computation, visualization and programming environment. It provides an interactive environment with hundreds of built-in functions for technical computing, graphics & animation.

Basic Commands

1. `clc` - Clears command window
2. `clear` - Removes variables from workspace.
3. `exist` - checks for existence of file or variable
4. `global` - Declares variable to be global.
5. `help` - Searches for a help topic
6. `lookfor` - searches help entries for a keyword
7. `quit` - stops MATLAB.
8. `who` - lists current variables.
9. `whos` - list current (leg display).

Command to Create 1-D Array -

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

$b = [1, 2, 3, 4, 5, 6, 7]$

$c = [1:5]$

Read the Image

Syntax -

 $A = \text{imread}(\text{file-name})$

→ It reads the image from the file specified by file-name; inferring the format of the file from its contents.

Ex - `I1 = imread('camera man.tif');`

2015002000(33)
Read the image and perform the TCC & FCC
FCC - False color Composite
Any other combination of colors.

TCC - True color composite
Red band-Red, Green band-Green, Blue band-Blue

cat command - It is used to concatenate two or more commands.

```
ii = imread('c:\users\c1235\pictures\glam.jpeg');  
b1 = ii(:, :, 1);  
b2 = ii(:, :, 2);  
b3 = ii(:, :, 3);  
subplot(2, 2, 1);  
imshow(ii);  
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);
```

Perform alternative pixel 0 and alternate row & column 0

Alternate row and column 0

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

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

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

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

```
            i1(i, j) = 0;
```

```
        end
```

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

```
        i1(i, j) = 0;
```

```
    end
```

```
end
```

```
end
```

Alternate pixel 0.

201500200(35)

Implement the checkerboard effect

```
il = zeros (256, 256);
```

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

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

```
        if (i==j)
```

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

```
        else if (mod(j,2) == 0) && (mod(i,2) == 0)
```

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

```
        else if (mod(j,2) == 0) || (mod(i,2) == 0)
```

```
            il(i,j) = 1;
```

```
        end
```

```
    end
```

```
end
```

```
il = cast(il, "uint8");
```

$20150 \text{ oad} \rightarrow$

i/ = 'unread ('camera man. +if')'

$$12 = \text{Hip } 11, 13,$$
$$18 = \text{flip}(11, 2)_{\wedge}$$

Subplot (2,2,1), imshow(ii),

Subplot (2,2,2);

im show (12);

Subplot (2, 2, 3):

ins how (13).

Without function

it = imread('camera man.tif')

$$i_2 = \text{LWS} (256, 256)$$

for $1 = 1.256$

$$i^2(256 - 1 + 1, -) = i^1(i, -);$$

end

$j_2 = \text{cast}(i_2, \text{'uint8'})$,

subplot (1,2,1)

1 mushroom (11);

Subplot (1,2,2)';

1ms how (12)

Perform ROI / Gray level thresholding 201500200 (35)

img = imread('cameraman.tif');

t = input('enter the limit');

for i = 1:256

for j = 1:256

if (img(i,j)) < t

imgt(i,j) = 0;

else

imgt(i,j) = 255;

end

end

end

subplot(1,2,1); imshow(img);

subplot(1,2,2); imshow(imgt);

Perform bit-plane slicing

Using Bitgate function

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

```
b1 = bitget(img, 1);
```

```
b2 = bitget(img, 2);
```

```
b3 = bitget(img, 3);
```

```
b4 = bitget(img, 4);
```

```
b5 = bitget(img, 5);
```

```
b6 = bitget(img, 6);
```

```
b7 = bitget(img, 7);
```

```
b8 = bitget(img, 8);
```

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

```
imshow(img);
```

```
subplot(3,3,2); imshow(logical(b1));
```

```
subplot(3,3,3); imshow(logical(b2));
```

```
subplot(3,3,4); imshow(logical(b3));
```

```
subplot(3,3,5); imshow(logical(b4));
```

```
subplot(3,3,6); imshow(logical(b5));
```

```
subplot(3,3,7); imshow(logical(b6));
```

```
subplot(3,3,8); imshow(logical(b7));
```

```
subplot(3,3,9); imshow(logical(b8));
```


perform transformation function

201500080(35)

Negative transformation

```
i1 = imread('cameraman.tif');  
[m,n] = size(i1);  
i2 = zeros(m,n);  
for i = 1:m  
    for j = 1:n  
        i2(i,j) = 255 - i1(i,j);  
    end  
end  
i2 = cast(i2, 'uint8');  
subplot(2,2,1);  
imshow(i1);  
subplot(2,2,2);  
imshow(i2);
```

Logarithmic transformation

```
i1 = imread('cameraman.tif');  
[m,n] = size(i1);  
i2 = zeros(m,n);  
i1 = cast(i1, 'double');  
c = input('enter the value of c');  
for i = 1:m  
    for j = 1:n  
        i2(i,j) = c * log(1 + i1(i,j));  
    end  
end  
i1 = cast(i1, 'uint8');  
i2 = cast(i2, 'uint8');  
subplot(1,2,1); imshow(i1);
```


201500280 (35)

Average filter

$i_1 = \text{imread}('cameraman.tif');$

$i_2 = \text{zeros}(258:258);$

$i_2 = (2:257, 2:257) \rightarrow i_2;$

$i_3 = \text{zeros}(256:256);$

$i_3 = \text{double}(i_3);$

$f_k = [1, 1, 1; 1, 1, 1; 1, 1, 1];$

for $i = 2:256$

for $j = 2:256$

Sum = 0

for $k = 1:3$

for $l = 1:3$

Sum = Sum + $i(i-2+k, j-2+l) \times f(k, l);$

end

end

$i_3(i, j) = \text{Sum} / 9;$

end

end

$i_3 = \text{cast}(i_3, 'uint8');$

$\text{imshow}(i_3);$

$\text{subplot}(1, 2, 1); \text{imshow}(i_1);$

$\text{subplot}(1, 2, 2); \text{imshow}(i_3);$

2. Weighted average filter

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

```
i2 = zeros(258; 258);
```

```
i2 = (2:257, 2:257) = i1;
```

```
i3 = zeros(256:256);
```

```
i3 = double(i3);
```

```
f = [1, 2, 1; 2, 4, 2; 1, 2, 1];
```

```
for i = 2:255
```

```
    for j = 2:255
```

```
        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 / 6;
```

```
    end
```

```
end
```

```
i3 = cast(i3, 'uint8');
```

```
imshow(i3);
```

```
subplot(1, 2, 1); imshow(i1);
```

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

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

3.) Laplacian filter

$i1 = \text{imread}('cameraman.tif');$

$i2 = \text{zeros}(258, 258);$

$i2 = [2:257, 2:257] = i1;$

$i3 = \text{zeros}(256, 256);$

$i3 = \text{double}(i3);$

$f = [0, -1, 0, -1, 1, -1, 0, -1, 0];$

for $i = 2:255$

for $j = 2:255$

sum = 0

for $k = 1:3$

for $l = 1:3$

sum = sum + $i2(i-2+k, j-2+l) * f(k, l);$

end

end

$i3(i, j) = \text{sum};$

end

end

$i3 = \text{cast}(i3, 'uint8');$

$\text{imshow}(i3);$

$\text{subplot}(1, 4, 1); \text{imshow}(i1);$

$\text{subplot}(1, 2, 2); \text{imshow}(i3);$

Order Statistics• Min filter

```

i1 = imread('cameraman.tif');
[m,n] = size(i1);
i3 = zeros(m,n);
s = input('enter the size');
for (i = 1:m-s+1)
    for (j = 1:n-s+1)
        i3(i,j) = min(min(i1(i-c+1:i-c+c, j-c+s))));
    end
end
subplot(1,2,1); imshow(i1);
subplot(1,2,2); imshow(uint8(i3));

```

• Max filter

```

i2 = imread('cameraman.tif');
[m,n] = size(i2);
i3 = zeros(m,n);
s = input('enter the size');
for (i = 1:m-s+1)
    for (j = 1:n-s+1)
        i3(i,j) = max(max(i2(i-c+1:i-c+c, j-c+s)));
    end
end

```

```

end
subplot(1,2,1); imshow(i2);
subplot(1,2,2); imshow(uint8(i3));

```

Median

```

I1 = imread('cameraman.tif');

```

```

[m,n] = size(I1);

```

```

I3 = zeros(m,n);

```

```

C = 5;

```

```

for i = 1:m-C+1

```

```

    for j = 1:n-C+1

```

```

        I3(i,j) = median(median(I1(i-C+1:i-C+5, j-C+1:j-C+5)))

```

```

    end

```

```

end

```

```

subplot(1,2,1); imshow(I1);

```

```

subplot(1,2,2); imshow(I3);

```

Morphology (Ex-13)

201500280101

```
I1 = imread('cameraman.tif');
I2 = im2bw(I1);
SC = strel('line', 11, 90);
eroded = imerode(I2, SC);
dilated = imdilate(I2, SC);
open = imdilate(eroded, SC);
closing = imerode(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));
I2_invert = invert(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
I_minut = imerode(I_invert, w);
for i = 1:255
    for j = 1:255
        if eroded(i,j) == I_invert(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 (hnm);

```

i1 = imread('cameraman.tif');
for i = 1:256
    ms(i,1) = i-1;
end
for k = 1:256
    c = 0;
    for l = 1:256
        for j = 1:256
            if i1(i,j) == k-1
                c = c+1;
            end
        end
        ms(k,2) = c;
    end
end
for i = 1:256
    ms(i,3) = ms(i,2) / 65536;
end
ms(1,4) = ms(1,3)
for i = 2:256
    ms(i,4) = ms(i-1,4) + ms(i,3)
end
for i = 1:256
    ms(i,5) = ms(i,4) * 255
end
for i = 1:256
    ms(i,6) = round(ms(i,5))
end
for i = 1:256
    ms(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
his eq (i,2) = cc;
end
subplot (2,2,1);
bar (his (i,1); his (i,2), 0, 255);
subplot (2,2,2);
bar (his eq (i,2), his eq (i,2));
temp 2 = ii;
for i = 1:256
    for j = 1:256
        m = temp 2 (i,j);
        for k = 1:256
            if m = his (k,2)
                temp 2 (i,j) = his (k,6);
            end
        end
    end
end
subplot (2,2,3);
imshow (i,1);
subplot (2,2,4);
imshow (temp 2);

```


line filter (Ex-18) 201500280(3)

```
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 = i3(i-2+k, j-2+l) * f1(k,l);
```

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

```
                a2 = i3(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 = i3(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
```

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

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

Sobel - filter (Ex-6) 201500280(35)

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