Introduction to MATLAB and image processing

MATLAB and images

- The help in MATLAB is very good, use it!
- An image in MATLAB is treated as a matrix
- Every pixel is a matrix element
- All the operators in MATLAB defined on
  matrices can be used on images: +, -, *, /, ^, sqrt, sin, cos etc.

Images in MATLAB

- Matlab can handle several image formats
  - BMP (Microsoft Windows Bitmap)
  - GIF (Graphics Interchange Files)
  - HDF (Hierarchical Data Format)
  - JPEG (Joint Photographic Experts Group)
  - PCX (Paintbrush)
  - PNG (Portable Network Graphics)
  - TIFF (Tagged Image File Format)
  - XWD (X Window Dump)

- Data types in MATLAB
  - Double (64-bit double-precision floating point)
  - Single (32-bit single-precision floating point)
  - Int32 (32-bit signed integer)
  - Int16 (16-bit signed integer)
  - Int8 (8-bit signed integer)
  - Uint32 (32-bit unsigned integer)
  - Uint16 (16-bit unsigned integer)
  - Uint8 (8-bit unsigned integer)
- Most commonly used data types are double and uint8

Images in MATLAB

- Indexed images : m-by-3 color map
- Intensity images : [0,1] or uint8, double etc.
- Binary images : {0,1}
- RGB images : m-by-n-by-3

Image import and export

- Read and write images in Matlab

```
>> I=imread('cells.jpg');
>> imshow(I)
>> size(I)
ans =   479   600     3                (RGB image)
>> Igrey=rgb2gray(I);
>> imshow(Igrey)
>> imwrite(Igrey, 'cell_gray.tif', 'tiff')
```

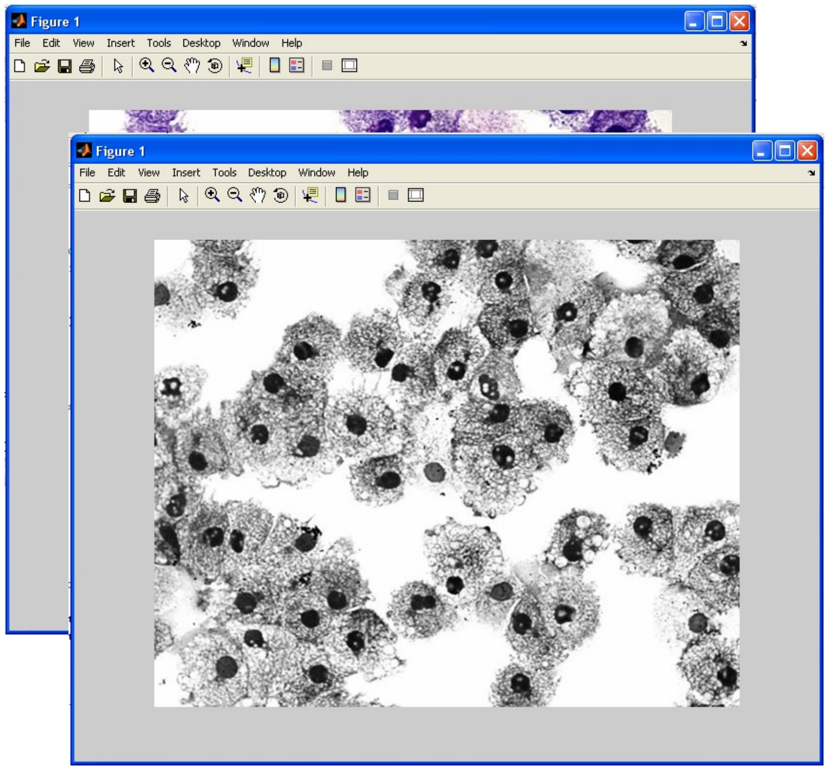
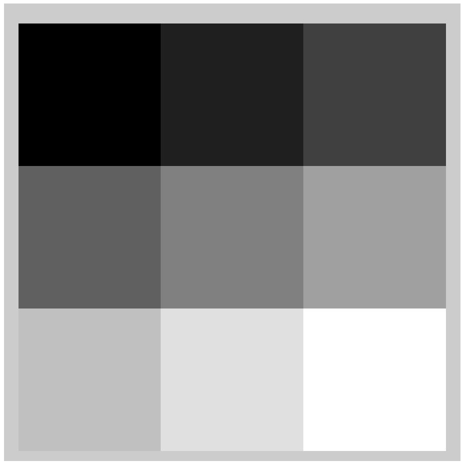


Figure 1
File Edit View Insert Tools Desktop Window Help
Figure 1
File Edit View Insert Tools Desktop Window Help

Images and Matrices

- How to build a matrix (or image)?

```
>> A = [ 1 2 3; 4 5 6; 7 8 9 ];
 A =   1     2     3
       4     5     6
       7     8     9
>> B = zeros(3,3)
 B =   0     0     0
       0     0     0
       0     0     0
>> C = ones(3,3)
C =    1     1     1
       1     1     1
       1     1     1
```



```
>>imshow(A)   (imshow(A,[]) to get automatic pixel range)
```

Images and Matrices

- Accesing image elements (row, column)

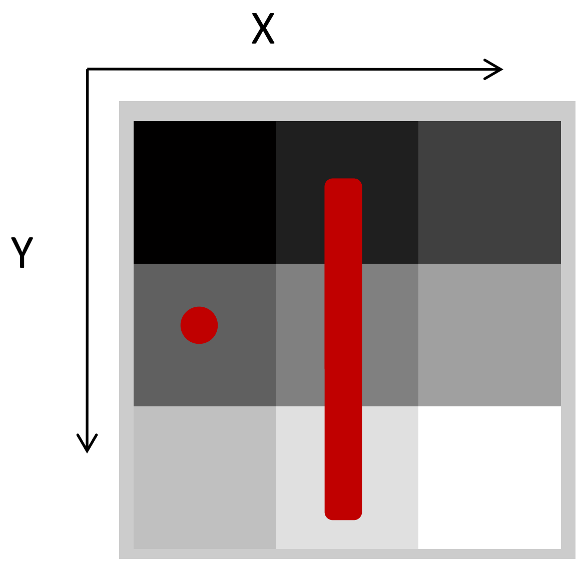
```
>> A(2,1)
ans = 4
```

- : can be used to a whole column or row

```
>> A(:,2)
ans =
2
5
8
```

- or a part of a column or row

```
>> A(1:2,2)
ans =
2
5
```

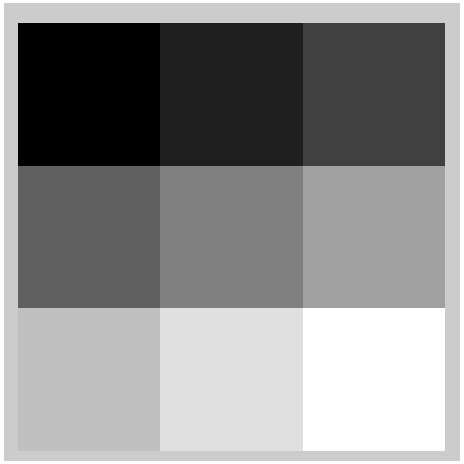


```
A =
1    2    3
4    5    6
7    8    9
```

Image Arithmetic

- Arithmetic operations such as addition, subtraction, multiplication and division can be applied to images in matlab
  - +, -, *, / peforms matrix operations

```
 >> A+A
ans =     2     4     6
           8    10    12
          14    16    18
 >> A*A
ans =    30    36    42
          66    81    96
         102   126   150
```



```
 A =
1    2     3
4    5     6
7     8     9
```

- To perform an elementwise operation use . (.*, ./, .*, .^ etc)

```
>> A.*A
ans =    1     4     9
         16    25    36
         49    64    81
```

Logical Conditions

- equal (==) , less than and greater than (< and >), not equal (~=) and not (~)
- find('condition') - Returns indexes of A's elements that satisfies the condition.

```
>> [row col]=find(A==7)
row =   3
col =    1
>> [row col]=find(A>7)
row =   3
         3
col =    2
         3
```

```
 A =
1    2    3
4    5    6
7    8    9
```

Flow Control

- Flow control in MATLAB

  - if, else and elseif statements

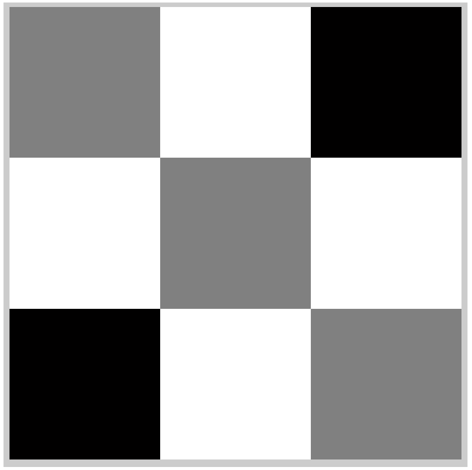
```
if row==col
        A(row, col)=1;
elseif abs(row-col)==1
        A(row, col)=2;
else
        A(row, col)=0;
end
```

Flow Control

- Flow control in MATLAB

  - for loops

```
for row=1:3
        for col=1:3
                if row==col
                        A(row, col)=1;
                elseif abs(row-col)==1
                        A(row, col)=2;
                else
                        A(row, col)=0;
                end
        end
end
```



```
A =

     1     2     0
     2     1     2
     0     2     1
```

- while expression statements end

```
Indx=1;
while A(Indx)<6
     A(Indx)=0;
     Indx=Indx+1;
 end
```

```
A =
1    2    3
4    5    6
7    8    9
```

```
 A =

    0    2    3
    0    5    6
    7    8    9
```

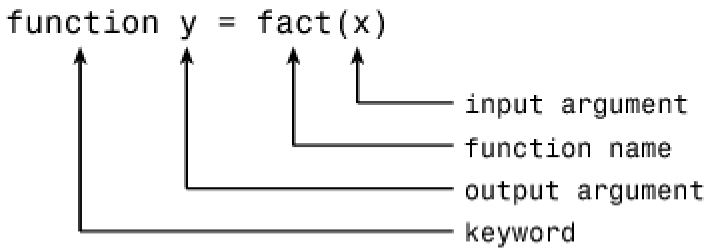
Working with M-Files

- M-files can be *scripts* that simply execute a series of MATLAB statements, or they can be *functions* that also accept input arguments and produce output.
- MATLAB functions:
  - Are useful for extending the MATLAB language for your application.
  - Can accept input arguments and return output arguments.
  - Store variables in a workspace internal to the function.

Working with M-Files

- Create a new empty m-file

```
function B=test(I)
[row col]=size(I)
for r=1:row
        for c=1:col
                if r==c
                        A(r, c)=1;
                elseif abs(r-c)==1
                        A(r, c)=2;
                else
                        A(r, c)=0;
                end
        end
end
B=A;
```


function y = fact(x)
input argument
function name
output argument
keyword