Introduction to MATLAB and image processing

Amin Allalou
amin@cb.uu.se

Centre for Image Analysis
Uppsala University

Computer Assisted Image Analysis
April 4 2008
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 import/export 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)
  - MATLAB can also load raw-data or other types of image data

- 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)
Images in MATLAB

- Binary images: \{0,1\}
- Intensity images: \[0,1\] or uint8, double etc.
- RGB images: m-by-n-by-3
- Indexed images: m-by-3 color map
- Multidimensional images m-by-n-by-p (p is the number of layers)
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')

Alternatives to imshow
  >>imagesc(I)
  >>imtool(I)
  >>image(I)
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

• Accessing image elements (row, column)
  \[ A(2,1) \]
  \[ \text{ans} = 4 \]

• : can be used to extract a whole column or row
  \[ A(:,2) \]
  \[ \text{ans} = \]
  \[ 2 \]
  \[ 5 \]
  \[ 8 \]

• or a part of a column or row
  \[ A(1:2,2) \]
  \[ \text{ans} = \]
  \[ 2 \]
  \[ 5 \]
Image Arithmetic

• Arithmetic operations such as addition, subtraction, multiplication and division can be applied to images in MATLAB.

\[ A + A \]
ans =

\[
\begin{array}{ccc}
2 & 4 & 6 \\
8 & 10 & 12 \\
14 & 16 & 18 \\
\end{array}
\]

\[ A \times A \]
ans =

\[
\begin{array}{ccc}
30 & 36 & 42 \\
66 & 81 & 96 \\
102 & 126 & 150 \\
\end{array}
\]

• To perform an elementwise operation use . (\(+, -, \times, \div\) etc).

\[ A \times A \]
ans =

\[
\begin{array}{ccc}
1 & 4 & 9 \\
16 & 25 & 36 \\
49 & 64 & 81 \\
\end{array}
\]
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.

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

```matlab
A =
12  3
45  6
 7  8
 9
```

```matlab
>> [row col]=find(A==7)
row =  3
col =  1
>> [row col]=find(A>7)
row =  3
    3
col =  2
    3
>> Indx=find(A<5)
Indx =  1
    2
    4
    7
```
Flow Control

- Flow control in MATLAB
  - if, else and elseif statements
  (row=1,2,3 col=1,2,3)

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

```
A =
1 2
0
2 1
2
0 2
1
```
Flow Control

- Flow control in MATLAB
  - for loops

```matlab
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
0     2     1
2     0     2
1
```
Flow Control

- **while**, **expression**, **statements**, **end**

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

```
A =
12  3
45  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

```matlab
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;
```