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 File)
  - 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
  ```matlab
  >> imread('tulip.jpg');
  >> imshow()
  ans = 479 600 3  [RGB image]
  >> imshow(rgb2gray());
  >> imshave(gray)
  >> imantsv(gray, 'cell_gray.tif', 'tiff')
  ```

Images and Matrices

- How to build a matrix (or image)?
  ```matlab
  >> 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)
  
  \[
  \text{A}(2,1) \]
  
  ans = 4
  
  : can be used to a whole column or row
  
  \[
  \text{A}(:,2) \]
  
  ans =
  
  2
  5
  8
  
  or a part of a column or row
  
  \[
  \text{A}(1:2,2) \]
  
  ans =
  
  2
  5

- Arithmetic operations such as addition, subtraction, multiplication and division can be applied to images in MATLAB
  
  \[
  \text{A} + \text{B} \]
  
  ans =
  
  2 4 6
  8 10 12
  14 16 18
  
  \[
  \text{A} \times \text{B} \]
  
  ans =
  
  30 36 42
  66 81 96
  102 126 150

- To perform an elementwise operation use \( \cdot, \div, \div, \cdot, \cdot \) etc
  
  \[
  \text{A} \cdot \text{B} \]
  
  ans =
  
  1 4 9
  16 25 36
  49 64 81

Logical Conditions

- \text{equal} (=), \text{less than} (<), \text{greater than} (>), \text{not equal} (\neq) and \text{not} (\not=)

- \text{find(condition)} - Returns indexes of A's elements that satisfies the condition.

\[
\begin{align*}
\text{row} & = 3 \\
\text{col} & = 1 \\
\text{row} & = 3 \\
\text{col} & = 2 \\
\text{row} & = 3 \\
\text{col} & = 3
\end{align*}
\]

Flow Control

- Flow control in MATLAB - if, else and elseif statements

\[
\text{if row==col} \\
\text{A(row, col)=1;} \\
\text{else if abs(row-col)==1} \\
\text{A(row, col)=2;} \\
\text{else} \\
\text{A(row, col)=0;} \\
\text{end}
\]

- While expression statements end

\[
\text{while A(index)>0} \\
\text{A(index)=0;} \\
\text{index=index+1;} \\
\text{end}
\]

\[
\text{A} =
\begin{bmatrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{bmatrix}
\]

- Flow control in MATLAB - for loops

\[
\begin{align*}
\text{for row}=1:3 \\
\text{for col}=1:3 \\
\text{if row==col} \\
\text{A(row, col)=1;} \\
\text{else if abs(row-col)==1} \\
\text{A(row, col)=2;} \\
\text{else} \\
\text{A(row, col)=0;} \\
\text{end} \\
\text{end} \\
\text{end}
\end{align*}
\]

\[
\text{A} =
\begin{bmatrix}
1 & 2 & 0 \\
2 & 1 & 2 \\
0 & 2 & 1
\end{bmatrix}
\]
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.

```matlab
function B=test(t)
    [row col]=size(t)
    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;
```

- Create a new empty m-file

```matlab
function y = fact(x)
    output argument
    input argument
    function name
    y= x = 0
    for n=1 to x
        y=y*n;
    end
    return y;
```

```matlab
function y = fact(x)
    output argument
    input argument
    function name
    y= x = 0
    for n=1 to x
        y=y*n;
    end
    return y;
```