



TW793: DIGITAL IMAGE PROCESSING

WHAT IS A (GRAY SCALE) DIGITAL IMAGE?

Matlab instructions:

```
>> A = [0 128; 192 255]
```

ENTER

```
A =
```

```
    0    128
```

```
   192   255
```

```
>> colormap( gray ( 256 ) );
```

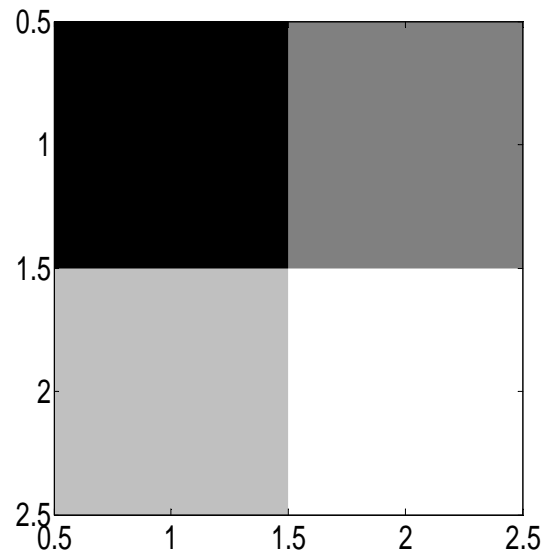
ENTER

```
>> image( A )
```

ENTER

```
>> axis image
```

ENTER





A gray scale digital image is a

- **matrix $f(x, y)$, where**
- **f is the intensity (grey scale value) of the image f at the coordinate (x, y)**
- **and each coordinate (x, y) is represented by a pixel, where**
- **the pixel is coloured black if $f(x, y) = 0$ and coloured white if $f(x, y) = L - 1$**
(for the example on the previous page we have that $L = 256$)

Matlab instructions:

```
>> A(2,1)
```

```
ENTER
```

```
ans =  
    192
```

```
>> A(1,2)
```

```
ENTER
```

```
ans =  
    128
```



When `colormap(gray (256))` is selected and an image (matrix) is **displayed** in Matlab with the `image` instruction, each entry (pixel) is internally “converted” to the `uint8` (unsigned 8-bit integer) data type. This implies that each pixel (matrix entry) is represented with an 8-bit byte, which implies $L = 2^8 = 256$ possible intensity values (gray scales). The instruction `imagesc` is the same as `image`, except that the data is scaled to use the full colormap, which can come in very handy.

In order to be able to **manipulate** an `uint8` matrix (image) in double precision arithmetic, the matrix has to be converted to the double data type with `B = double(A)`. The inverse operation is achieved with `A = uint8(B)`.

An image file, e.g. “lenna256.jpg” (JPEG format) can be **displayed** with the following Matlab instruction (IP Toolbox)

```
>> imshow( 'lenna256.jpg' )
```

ENTER



lenna256.jpg





To manipulate this image in Matlab, the following instructions are needed

```
>> X = imread( 'lenna256.jpg' );
```

ENTER

```
>> X = double( X );
```

ENTER

The variable X now represents a matrix that can be accessed/manipulated

```
>> min( min( X ) );
```

ENTER

```
ans =
```

```
0
```

```
>> max( max( X ) );
```

ENTER

```
ans =
```

```
239
```

```
>> size( X );
```

ENTER

```
ans =
```

```
256 256
```

```
>> X(100,150);
```

ENTER

```
ans =
```

```
173
```

The matrix X can again be displayed with the instructions

```
>> colormap( gray ( 256 ) );
```

ENTER

```
>> imagesc( X )
```

ENTER

```
>> axis image
```

ENTER



We can now, e.g., calculate and display the negative of the image with the instructions

```
>> Y = max( max( X ) ) - X;
```

ENTER

```
>> colormap( gray ( 256 ) );
```

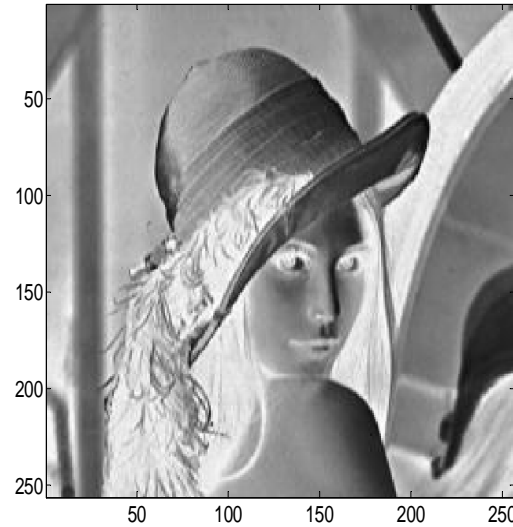
ENTER

```
>> imagesc( Y )
```

ENTER

```
>> axis image
```

ENTER



Any matrix can be saved to a disk. The following instruction saves the variable Y in the current directory as the Matlab file “lenna256neg.mat”:

```
>> save lenna256neg.mat Y
```

ENTER

During subsequent Matlab sessions this file can be retrieved and automatically reassigned to the variable Y with the instruction

```
>> load lenna256neg.mat
```

ENTER

The matrix Y can also be saved in a variety of common image file formats, like JPEG, BMP, etc., with instructions like

```
>> imwrite(Y,gray(256),'lenna256neg.bmp','bmp')
```

The file “lenna256neg.bmp” (BMP format) can be displayed with the Matlab instruction (IP Toolbox):

```
>> imshow('lenna256neg.bmp')
```



In order to obtain a list of all the available Matlab functions in the Image Processing (IP) Toolbox, give the instruction:

```
>> help images
```

E.g., the negative of an image can also be obtained with the `imcomplement` function, i.e.:

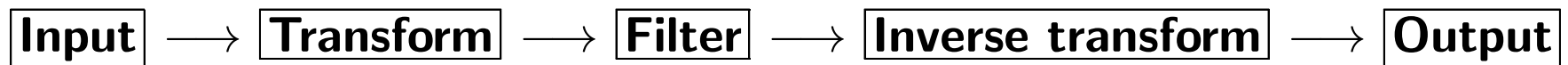
```
>> Y = imcomplement( X );
```

For a demonstration of the IP Toolbox, enter “demo” in the command window and double-click on “Toolboxes” and then “Image Processing”...

WHAT IS DIGITAL IMAGE PROCESSING?



- Image enhancement and restoration



Input image



Wiener filter



Output image






- Image compression

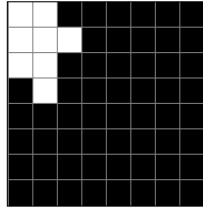


JPEG Demo: 8 DCT coefficients retained

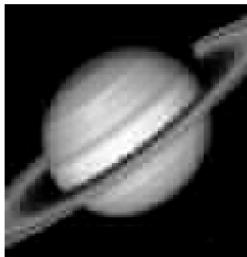
Original Saturn Image




DCT coefficients



Reconstructed Image



Error Image



1 64
8 Coefficients

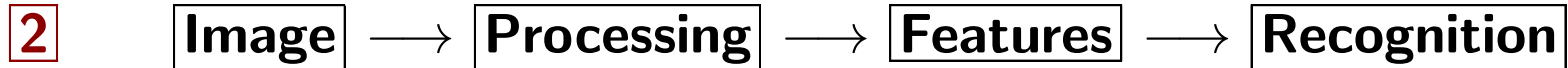
Apply

Select Image:

Info

Close

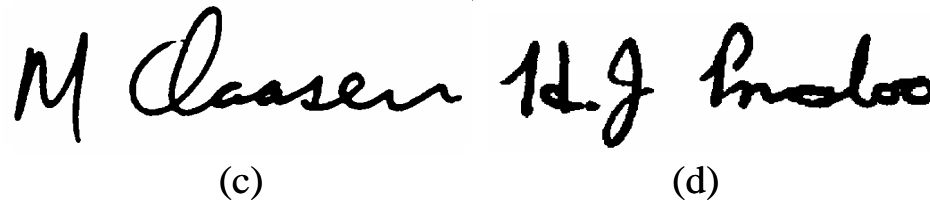
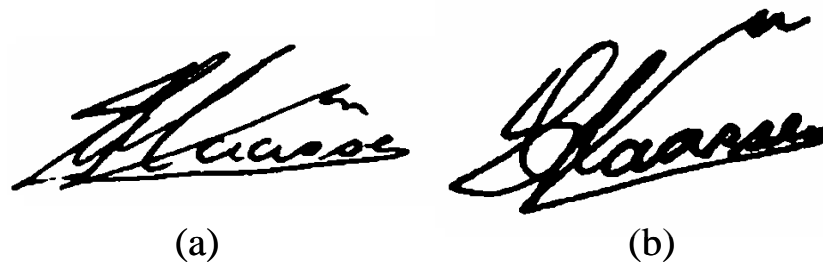
The MSE (with images normalized) is 0.000984 .



• Object recognition



Example: Signature verification



(the authentic signature is (a))



OVERVIEW OF COURSE

- **Fourier analysis**
 - **Image enhancement**
 - **Image restoration (Ch. 5)**
 - **Colour IP (Ch. 6)**
 - *Wavelets (Ch. 7)*
 - **Compression (Ch. 8)**
 - **Morphological IP (Ch. 9)**
 - **Pattern Recognition**
- **Spatial domain (Ch. 3)**
 - **Fourier domain (Ch. 4)**
- **Segmentation (Ch. 10)**
 - **Representation & description (Ch. 11)**
 - *Object recognition (Ch. 12)*