



Prof. B. Schiele  
<schiele@informatik.tu-darmstadt.de>

Micha Andriluka  
<andriluka@mis.informatik.tu-darmstadt.de>

## Exercise 1: Introduction to Matlab

(due April 16th 2008, 23:59)

### Question 1: Matlab Tutorial

- a) Create your working directory. Run matlab by clicking on matlab icon or by typing "matlab" in the command shell. Change the directory to your working directory.
- b) Read the Matlab help chapter "Getting Started with Matlab" (linked from the class webpage).

- Matrices and Arrays
  - Expressions
  - Working with Matrices
  - More About Matrices and Arrays
    - Linear Algebra
    - Arrays
    - Multivariate Data

#### Graphics

- Editing Plots

- Mesh and Surface Plots

- Images

#### Programming

- Flow Control

- if, else, and elseif

- for

- while

- return

- Other Data Structures

- Characters and Text

- Scripts and Functions

- Scripts

- Functions

- c) Walk through the following Matlab demos (<http://www.mathworks.com/products/matlab/demos.html>):

#### Mathematics

- Basic Matrix Operations

- Matrix Manipulation

#### Graphics

- 2-D Plots

- 3-D Plots

- Images and Matrices

#### Programming

- Manipulating Multidimensional Arrays

- Function Functions

- Reading Arbitrary Text Files with TEXTSCAN

## Question 2: Basic Image Processing

Download the file `exercise1.tar.gz` from the class web page and uncompress it in your working directory. This file contains two example images, which you will manipulate in the following. For the first exercises, we provide example code. Try it out and shortly explain what happens. For the later exercises, we only provide a code framework which you should complete by yourselves. Useful commands for this are:

```
imread, image, imshow, imagesc, colormap, imrotate.
```

- a) Read the image “graf.png” and display it in a window with suggested commands.

```
graf=imread('graf.png');  
image(graf);  
imagesc(graf);  
imshow(graf);
```

- b) Look at the colormap. Create and set a new colormap which varies from 1 to 0, 0 to 1, and several random maps. Display the image. Explain what happens.

```
map=colormap;  
map1=map(end:-1:1,:);  
map2=rand(64,3);  
colormap(map1);  
image(graf);  
colormap(map2);  
image(graf);  
colormap('default');  
image(graf);
```

- c) Brighten and darken the image.

```
bgraf=graf*2;  
image(bgraf);  
dgraf=graf/2;  
image(dgraf);
```

- d) Assign row 200 in the image to a variable. Display it with a plot. Repeat that for column 300.

```
row200=graf(200,:);  
plot(row200);  
column300=graf(:,300);  
plot(column300);
```

- e) Remove a rectangular region from the image and display the region. Crop a rectangular region from the image containing only the head of the karate man and display it as a new image.

```
graf1=graf;  
graf1(130:260,240:450)=0;  
imshow(graf1);  
graf2=graf(130:260,240:450);  
imshow(graf2);
```

- f) Crop the head of the chicken and rotate it by 90 degree. Try different methods (`nearest`, `bilinear`, `bicubic`) and bounding boxes (`loose`, `crop`). Explain the observations.

```
graf3=graf(130:260,240:450);  
graf4=imrotate(graf2,90,'default');  
imshow(graf4);
```

- g) Sample the image (reduce the size). Use different sampling step sizes (2,4,10). Explain what happens.

```
graf5=graf(1:2:end,1:2:end);
imshow(graf5);
```

- h) Create a mirror image. Concatenate 2 or 4 images.

```
graf6=graf(:,end:-1:1);
imshow(graf6);
graf7=[graf,graf;graf,graf];
imshow(graf7);
```

- i) Assign row 200 in the image to a variable. Display it with a plot. Create a filter which is a vector of 10 equal elements summing to 1. Convolve the row with the filter and plot the result. Explain what happens.

```
row200=graf(200,:);
plot(row200);
filt1=ones(1,10)/10;
newrow=conv(row200,filt1);
plot(newrow);
```

- j) Create a 2D blur filter of size  $10 \times 10$  which sums to 1. Convolve it with the image and display the new image. Create a 1D filter of size 10 which sums to 1. Convolve it with the lines of the image. Transpose the filter and convolve it with rows of the filtered image. Explain what happens.

```
filtersize=10;
filt2D=ones(filtersize,filtersize)/(filtersize*filtersize);
graf8=conv2(graf,filt2D);
imshow(graf8);
filt1D=ones(1,filtersize)/filtersize;
graf9=conv2(graf,filt1D);
graf10=conv2(graf9,filt1D');
imshow(graf10);
imshow(graf10-graf8);
```

For the following questions, we just provide a code framework. Try to fill in the holes in the Matlab code using the things you have learned above.

- k) (5 points) Implement a method which generates a Gaussian for a given variance and number of samples.

$$G = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{\sigma^2}\right) \quad (1)$$

where  $\sigma$  is the standard deviation and  $x$  is a vector of integer values.

Create a new file `gauss.m` with your preferred editor and begin the script:

```
function G=gauss(x,sigma)
...
...
end
```

- l) (5 points) Create a function called `basicimproc` which takes as arguments an image `inimg` and five numbers `x`, `y`, the `size` of a square, `rotationangle` and `filtersize`. The function should crop a square region out of the image, rotate it by the desired angle, blur the cropped region, and then return it as `outimg`. It should also display the intermediate result images.

Create a new file called `basicimproc.m` with your preferred editor and begin the script:

```
function outimg=basicimproc(inimg,x,y,size, rotationangle, filtersize)
...
...
end
```

m) (10 points) Implement a Gaussian filter in `gaussianfilter.m`. Width and height of the filter should be  $2 * (3 * \text{sigma}) + 1$ .

Create a new file called `gaussianfilter.m` with your preferred editor and begin the script:

```
function outimg=gaussianfilter(inimg,sigma)
...
...
end
```

*Hint: use the fact that 2-d Gaussian filter is separable to speed up computations.*

---

*Please turn in your solution to Question 2 by sending an email to Micha Andriluka ([andriluka@mis.informatik.tu-darmstadt.de](mailto:andriluka@mis.informatik.tu-darmstadt.de)) including all relevant m-files before Wednesday, April 16th, 23:59.*