

## Blatt 1 : Einführung in Matlab und die Image-Processing-TB

### Einloggen:

Eine Beschreibung zum Einloggen unter Linux oder Windows und den Zugriff auf die Praktikumsverzeichnis finden Sie bei den Vorlesungsbeschreibungen auf der Homepage.

Alles weitere entnehmen Sie dem Übungsblatt und ggf. begleitenden Notebooks für Mathematica oder Matlab

### Anmerkungen

Dieses Praktikum dient

- 1a) der allgemeinen Einführung in Matlab sowie
- 1b) der Einarbeitung in die Image Processing Toolbox.

Es folgen jeweils Listen grundlegender Kapitel, die für uns besonders relevant sind. Danach sollen Sie jeweils Fragen beantworten und abgeben.

Zu Beginn noch 2 Tips:

- Das LRZ bietet Matlab sehr preiswert für Studenten an.
- Im Internet gibt es eine Fülle von Tutorials, von denen man ein paar einmal überfliegen sollte.
- Sie können Ihre Notebooks (Matlab-Dokumente) mit allen Ausgaben und Ergebnisdarstellungen als html-Dateien exportieren. So geben Sie bitte generell Ihre Ergebnisse ab.
- Sollten in Ihrer Matlab-Version Probleme mit den Dateinamen der Notebooks bestehen (Zahl am Anfang, '-', '+'-Zeichen im Dateinamen), so ändern Sie diese einfach ab

Die Kapitelbezeichnungen im Help-Tool können in der aktuellen Version der installierten Version von Matlab im Labor etwas abweichen.

Noch eine Bitte: Sollten Ihnen Fehler oder Unstimmigkeiten in den Übungsblättern oder Notebooks des Praktikums auffallen, nennen Sie diese bitte.

### Blatt 1a) Einführung in Matlab

Starten Sie Matlab: (In Linux per `matlab &` oder) in Windows per Doppelklick auf das Matlab-Symbol.

CI, Schnör

Hochschule München, FK Informatik und Mathematik

Starten Sie rechts oben: Help → Matlab Help.

Machen Sie sich mit dem Help-Tool vertraut:

- links oben die Registerkarten Contents, Index, Search, Demos;
- Geben Sie unter „search for“ z.B. einmal „exit“ ein.

Arbeiten Sie dann folgende Kapitel unter „Contents“ durch, indem Sie zunächst die Kapitel **kurz überfliegen**, um einen Gesamtüberblick zu bekommen, und vertiefen Sie sie dann, indem Sie die ein oder anderen Befehle selbst im Command-Window ausprobieren<sup>1</sup>. Dafür sollte etwa eine halbe Stunde reichen:

„Matlab → Getting Started →“

- Introduction → Starting and Quitting Matlab;
- Matrices and Arrays; (Matrices and Magic Squares → „The magic function“ auslassen)
- Graphics → Basic Plotting Functions →
  - Creating a Plot,
  - Figure Windows,
  - Saving Figures;
- Graphics → Images;
- Programming →
  - Flow Control,
  - Other Data Structures → Multidimensional Arrays,
  - Other Data Structures → Characters and Text,
  - Scripts and Functions;

Nun zur Vertiefung noch folgende Kapitel (30 Minuten):

- Matlab → Mathematics → Matrices and Linear Algebra → Function Summary;
- Matlab → Programming → Data Structures →
  - Creating and Concatenating Matrices;
  - Accessing Elements of a Matrix;
  - Getting Information About a Matrix;
  - Resizing and Reshaping Matrices;
  - Multidimensional Arrays;
- Matlab → Programming → Data Types, →
  - Overview;
  - Numeric Types;

<sup>1</sup> auch hier zu finden: <http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.html>

CI, Schnör

Hochschule München, FK Informatik und Mathematik

- Logical Types;
- Matlab → Programming → M-File Programming →
  - M-File Scripts and Functions;
  - Function Arguments;
- Matlab → Programming → Programming Tips;

**Aufgaben zur Bearbeitung und Abgabe:**

Geben Sie jeweils den Befehl, das Symbol oder ein kurzes (Zahlen-)Beispiel für folgende Operationen an:

1. Kommentarzeichen
2. Unterdrückung der Befehlsausgabe
3. Listing aller Variablen einer Sitzung
4. Suchpfad für Matlab-Funktionen (m-Files) anzeigen
5. Löschen einer Variablen `a`
6. Löschen aller Variablen und im Cache gehaltenen Funktionen einer Arbeitsitzung
7. 3x4-Matrix mit Nullen per spezieller Matlab-Funktion
8. 1x3-Zeilenvektor
9. 3x1-Spaltenvektor
10. 3-te Spalte einer Matrix `A`
11. zweite Zeile einer Matrix `A`
12. Matrix-Transposition
13. vertikale Aneinanderreihung 2er Matrizen `A` und `B`
14. horizontale Aneinanderreihung 2er Matrizen `A` und `B`
15. Größe einer Matrix `[1,2,3]`:
16. Dimensionen einer Matrix `[1,2,3]`: Was fällt auf?
17. Aneinanderhängen 2er Strings `a='123'; b='456';`
18. Setzen aller negativen Elemente einer Matrix auf null: Testen Sie das!
19. eine Liste von Zahlen im Abstand 10 von 1 bis 100 (1, 11, 21, ..., 91)
20. Zahl der Übergabeparameter einer Funktion
21. Beginn eines Abschnittes, der im Editor separat ausgeführt werden kann (Stichwort Cell-Mode)
22. Ort aufrufen, wo eine Funktion definiert ist (um z.B. zu sehen, welche Funktion tatsächlich „angezogen“ wird oder um den Sourcecode zu studieren)

23.
  - Wechseln Sie in Ihr Arbeitsverzeichnis (cd);
  - legen Sie ein Unterverzeichnis „`itv`“ an und fügen Sie dieses in den Suchpfad von Matlab ein;
  - schreiben Sie nun eine Funktion „`myAddProd( arg1, arg2 )`“, welche Sie in „`itv`“ ablegen.
  - Die Funktion gibt die Summe und das Produkt der Argumente zurück:  
`.summe, produkt = myAddProd( arg1, arg2 );`  
Eine Übergabe von mehr oder weniger als 2 Argumenten soll als Fehler erkannt werden. Diese Funktion soll außerhalb des „`itv`“-Verzeichnisses aufrufbar sein.

**Blatt 1b) Die „Image Processing Toolbox“**

Die Image Processing Toolbox ist ein Erweiterungspaket von Matlab (neben vielen anderen), die viele nützliche Funktionen für die Bildverarbeitung bereithält.

Eine Befehlsübersicht erhalten Sie mit `help images`.

Überfliegen Sie folgende Kapitel im Help-Tool: <sup>2</sup>.

Contents → Image Processing Toolbox → Getting Started →

- What is the IP-TR?
- Example 1
- Introduction → Image Types in the Toolbox
- Introduction → Multiframe Image Arrays
- Introduction → Reading and Writing Image Data
- Displaying and Exploring Images →
  - Overview
  - Using `imshow` to Display Images
  - Getting Information about an Image
  - Viewing Multiple Images

**Hinweise:**

- Bilder werden als  $n \times m$  Matrizen (Grau- und Binärbilder) oder als  $n \times m \times 3$  Arrays (RGB-Farbbilder) gespeichert und interpretiert. Der erste Index ist der Zeilenindex (y-Bildkoordinate), der zweite Index ist der Spaltenindex (x-Bildkoordinate), wobei der Punkt (1,1) dem linken oberen Bildpunkt entspricht, und der dritte Index ist der Farbkanal.
- Laden, Ansehen und Speichern: `imread`, `imshow`, `imwrite`;

<sup>2</sup> auch hier zu finden: <http://www.mathworks.com/access/helpdesk/help/toolbox/images/images.shtml>

```
T = imread('myimage.jpg'); % Farbbild einlesen
[m, n, k] = size(T) % Groesse ansehen
imshow(I); % Anzeigen
T(1:10, 1:10, :) = 0; % schwarzes Quadrat oben links
imshow(I); % Anzeigen
```

- 4 Bildarten: Index, Grauwert (Intensity), Binärbild (Binary) und Farbbild (RGB).
- Bild umwandeln: ind2gray, ind2rgb, gray2ind, rgb2gray, rgb2ind
- 3 Bildklassen: double, uint8 und uint16
- Bildarithmetik: Für Bilder der Klasse double können die standardmäßigen arithmetischen Operationen angewendet werden. Binärbilder werden mit 1 – double(I) zunächst umgewandelt.

Beispiele allgemeiner Bildverarbeitungsoperationen:

Testen Sie folgende Befehle, sofern unklar, aus:

```
[I,map]=imread('trees.tif'); % read an indexed tiff image
size(I) , ndims(I) % 2 Dimensionen
figure % display it as indexed image w/colormap
imshow(I,map) % convert it to grayscale
I2=ind2gray(I,map);
figure
imagesc(I2,[c l]) % scale data to use full colormap
% for values between 0 and 1
colormap('gray') % use gray colormap
axis('image') % make displayed aspect ratio
% proportional to image dimensions
I=imread('greens.jpg'); % read a RGB-jpeg image into 3D array
size(I) , ndims(I) % 3 Dimensionen
class(I) % datentyp
figure
imshow(I) % select rectangle
rect=getrect; % crop
I2=imcrop(I,rect); % convert cropped image to grayscale
I2=rgb2gray(I2); % scale data to use full colormap
imagesc(I2) % between min and max values in I2
colormap('gray')
colorbar % turn on color bar
pixval % display pixel values interactively
trueSize % display at resolution of one screen pixel
% per image pixel
trueSize(2*size(I2)) % display at resolution of two screen pixels
```

```
I3=imresize(I2,0.5,'bilinear'); % per image pixel
% resize by 50% using bilinear
% interpolation
I3=imrotate(I2,45,'bilinear','crop'); % rotate 45 degrees and crop to
% original size
I3=double(I2); % convert from uint8 to double, to allow
imagesc(I3.^2) % math operations
% display squared image (pixel-wise)
imagesc(log(I3)) % display log of image
```

Aufgaben zur Bearbeitung und Abgabe:

Von Ihrem Arbeitsverzeichnis ausgehend in „allg/Bilder“ liegt das Bild peppers.png. Bearbeiten Sie die Aufgaben im vorbereiteten Notebook DBV-Ex-IP-Toolbox.m. Darin werden Sie

- ein Bild einlesen,
- es abspeichern;
- die Bildinformationen ermitteln (Größe, Pixel-Tiefe),
- es als JPEG-Datei abspeichern,
- es in die einzelnen Farbkanaäle auftrennen,
- eine Zeile herausgreifen und als Grauwertfunktion  $g(x)$  darstellen und
- eine Zeile in einem Bild markieren.

**Abgabe:** Exportieren Sie abschließend das Notebook (File->Publish...), packen es zusammen mit dem hml-Ordner ein (zip oder tar.gz) und schicken es per email zu (eine Abgabe pro Gruppe). Je nach Betriebssystem kann es Probleme mit Filenamen geben, entfernen Sie ggf. \_ aus dem Namen des Notebooks.

Nun sind Sie vorbereitet, komplexere Aufgaben an Bildern in Matlab vorzunehmen!