

# BIV-Übung 1

## Einführung in MATLAB

MATLAB ist eine Vektor/Matrix-orientierte Sprache, die den Umgang mit Datenfeldern erheblich vereinfacht. MATLAB erlaubt eine *interaktive Befehlseingabe*, was besonders ein schnelles Ausprobieren erleichtert. Auf der Moodle-Seite finden Sie Links zu Einführungen in MATLAB.

MATLAB ist wegen der möglichen interaktiven Eingabe aller Befehle ein Interpreter. Besonders bei Schleifenkonstruktionen ist MATLAB daher langsam. Da aber mächtige Array-Befehle in MATLAB verfügbar sind und auch viele Operatoren mit Arrays arbeiten, können Schleifen oft durch Vektorisierung vermieden werden. Nutzen Sie die Möglichkeiten der Sprache! Für dieses Aufgabenblatt wird *keine Schleife* benötigt!

## 1 Arbeiten mit Vektoren und Matrizen

Probieren Sie die folgenden Befehle *nacheinander* (!) aus und vollziehen Sie die Ergebnisse nach:

```
x=[ 1 2 3 4 ]
x'
x*x'
x.*x

[x x]
M=[x;2*x];
M=[M;M];

M
M(2,3)
M(2,:)    % hier bedeutet der ":" alle Spalten
M(:,3)    % hier bedeutet der ":" alle Zeilen
M(2:4,1:2) % "2:4" bedeutet "2 bis 4"

M>3      % vergleichen Sie die Ausgabe mit der Ausgabe der nächsten Zeile
M(M>3)
M(M>3)=17
M
```

## 2 Farbbilder

Entpacken Sie die Datei **BIV\_P1-Dateien.zip** in ein neues Verzeichnis und setzen Sie das MATLAB-Arbeitsverzeichnis auf dieses Verzeichnis (Zeile unterhalb der Icons; üblicherweise werden dann im Bereich links die Dateien des Verzeichnisses angezeigt („Current Folder“), allerdings können Sie das auch ändern).

a) Echtfarben-Bilder werden in MATLAB als *dreidimensionale Matrix* behandelt, wobei die letzte Dimension für die Farbkanäle steht.

Hilfe zu den angegebenen Funktionen können Sie durch den Befehl `doc <Funktionsname>` aufrufen oder einfach durch Positionieren des Cursors auf dem Befehl und drücken von **F1**.

- Laden Sie das Bild `moewex.jpg` mit dem Befehl `moewe=imread('moewex.jpg');`

- Öffnen Sie ein neues Fenster mit `figure` und zeigen Sie das Bild mit `imshow` an.
- Rufen Sie die Funktion `size` auf dem Bild auf und vollziehen Sie die Rückgabe nach. Die Angaben finden Sie auch im Workspace Fenster (üblicherweise rechts oben).
- Die Funktion `showQuadView` (aus der oben erwähnten zip-Datei) dient der Anzeige von vier Bildern in einem Fenster, entsprechend hat die Funktion vier Parameter (wenn Sie die Implementierung interessiert, einfach die Datei öffnen).

Lassen Sie sich vier Bilder in folgender Reihenfolge anzeigen: Erste Zeile: Farbbild, Rot-Kanal; zweite Zeile: Grün-Kanal, Blau-Kanal.

Tipp: Um einen Kanal aus der dreidimensionalen Matrix herauszuholen, brauchen sie alle Zeilen des Bildes, alle Spalten des Bildes und den gewünschten Kanal. Für die Auswahl von „allen Zeilen“ ggf. nochmal die Aufgabe 1 anschauen!

Die einzelnen Kanäle können Sie entweder schwarz-weiß oder (Zusatzaufgabe) eingefärbt mit der Farbe des Kanals anzeigen lassen. Zum „Einfärben“ die jeweils anderen Farbkanäle entsprechend ändern (wie?).

- b) Konvertieren Sie das Bild in HSI mit dem Befehl `rgb2hsi`. Erzeugen Sie wiederum mit `showQuadView` eine vier-Bilder-Ansicht mit dem 3-Kanal-HSI-Bild als erstes und den einzelnen Kanälen in den übrigen drei Ansichten.

- c) In welchem der Farbkanäle (RGB, HSI) lässt sich die Möwe besonders gut vom Hintergrund unterscheiden?

- d) Benutzen Sie den „Data Cursor“ (siehe Icon-Leiste des Bild-Fensters) um herauszufinden, wie die Werte innerhalb der Möwe sich von denen außerhalb unterscheiden.

Können Sie einen Schwellwert finden, mit dem Sie die Möwe vom Hintergrund (in etwa) trennen können? (Oder sogar eine Kombination der Farbkanalwerte, die dann eine gute Trennung ermöglicht?)

Probieren Sie verschiedene Werte aus: Sie können in MATLAB auf ganzen Bildern boolsche Operationen wie „>“ so durchführen, wie Sie es in Java bei Zahlen gewohnt sind. Das Ergebnis ist ein binäres Bild. Lassen Sie sich das Bild anzeigen.

- e) „Schneiden“ Sie mit Hilfe des binären Bild die Möwe aus dem RGB-Bild aus: Im ausgeschnittenen Bild soll (möglichst) nur die Möwe zu sehen sein, alles andere soll schwarz sein. (Das Bild soll also *nicht* kleiner werden.) Mit dem Befehl `repmat(binaerbild, [1 1 3])` können Sie aus dem einen Binärbild ein Binärbild mit 3 „Farbkanälen“ machen, dass Sie dann zum Ausschneiden verwenden können. Eventuell müssen Sie den Datentyp konvertieren: `uint8(binaerbild)`.

- f) Laden Sie zusätzlich das Bild `bryce.jpg`. Setzen Sie die Möwe in das Bild.

Tipp: Dazu zunächst ein „Loch“ an der Stelle des Möwe in das Bild schneiden.