
Gruppe N :

Table of Contents

BIV-Übung 1	1
Einführung in MATLAB	1
1 Arbeiten mit Vektoren und Matrizen	1
2 Farbbilder	2

1527985 Donkeng, Ferly Loic IMB

1611148 Warsame, Ahmed IB

BIV-Übung 1

Einführung in MATLAB

1 Arbeiten mit Vektoren und Matrizen

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

```
x=[ 1 2 3 4 ];
r = x';
r;
r = x*x';
r;
r = x.*x;
r;
r = [x x];
r;
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"
r = M>3; % vergleichen Sie die Ausgabe mit der Ausgabe der nächsten
Zeile
r;
M(M>3)
M(M>3)=17;
M;
```

ans =

6

ans =

2 4 6 8

ans =

3

6

3

6

ans =

2 4

1 2

2 4

ans =

4

4

6

6

4

8

4

8

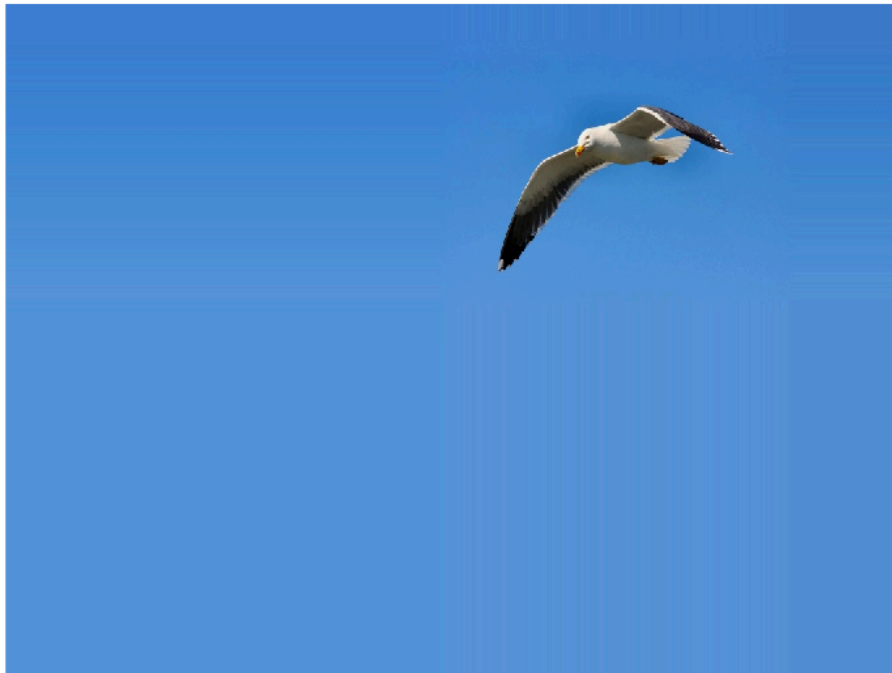
2 Farbbilder

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');`.

```
moewe = imread('moewex.jpg');
```

- öffnen Sie ein neues Fenster mit `figure` und zeigen Sie das Bild mit `imshow` an.

```
figure
imshow(moewe)
```



- 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).

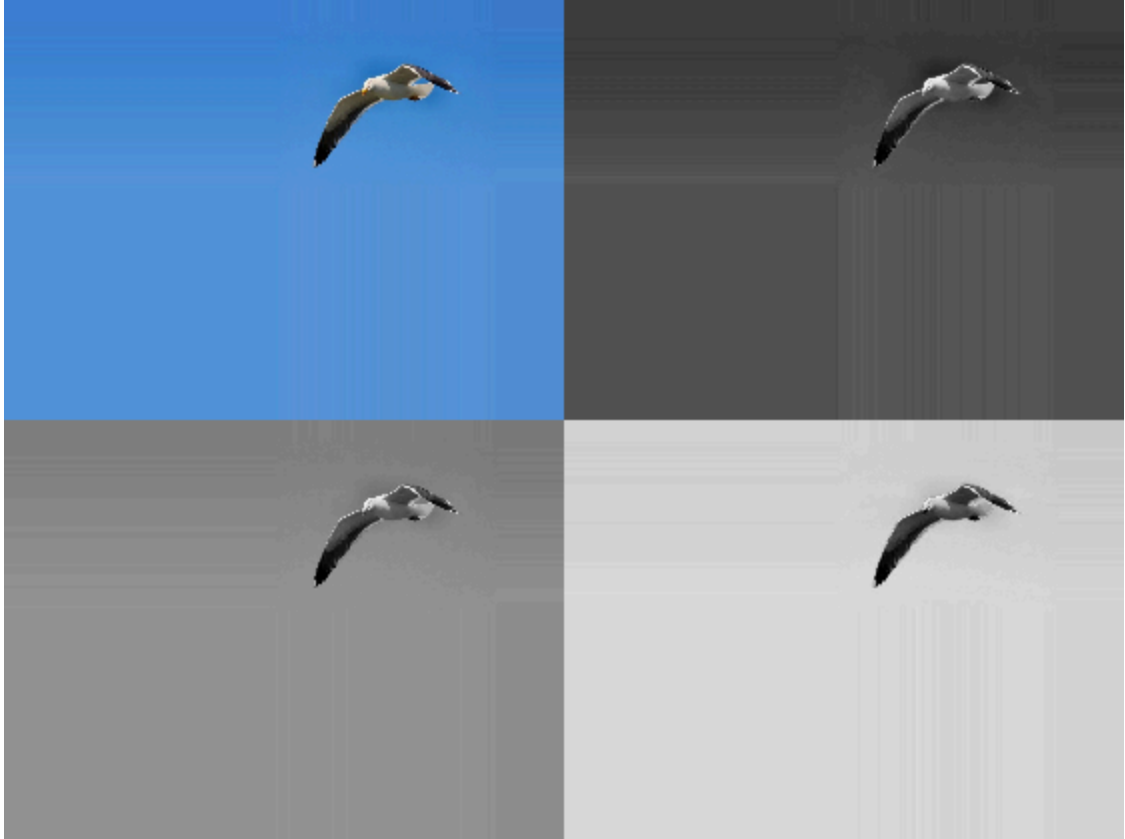
```
size(moewe)
```

```
ans =
```

```
       768       1024        3
```

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

```
moewe_red = moewe(:,:,1);  
moewe_green = moewe(:,:,2);  
moewe_blue = moewe(:,:,3);  
showQuadView(moewe,moewe_red,moewe_green,moewe_blue)
```



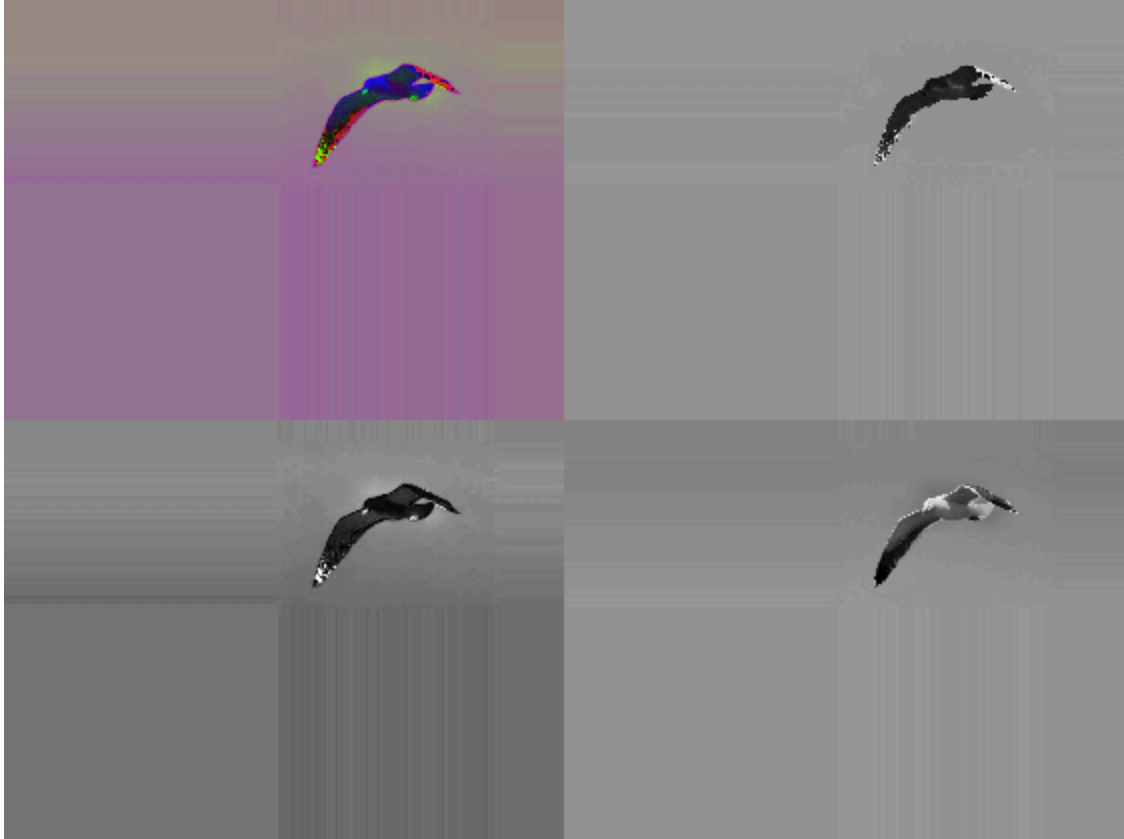
b) Konvertieren Sie das Bild in HSI mit dem Befehl `rgb2hsi`. Erzeugen Sie wiederum mit `showQuadView` eine vier-Bilder-Ansicht mit dem 3-Kanal-HSI-Bild als erstes und den einzelnen Kanälen in den übrigen drei Ansichten.

```
moewe2 = rgb2hsi(moewe);  
moewe2_red = moewe2(:, :, 1);  
moewe2_green = moewe2(:, :, 2);  
moewe2_blue = moewe2(:, :, 3);  
showQuadView(moewe2, moewe2_red, moewe2_green, moewe2_blue)
```



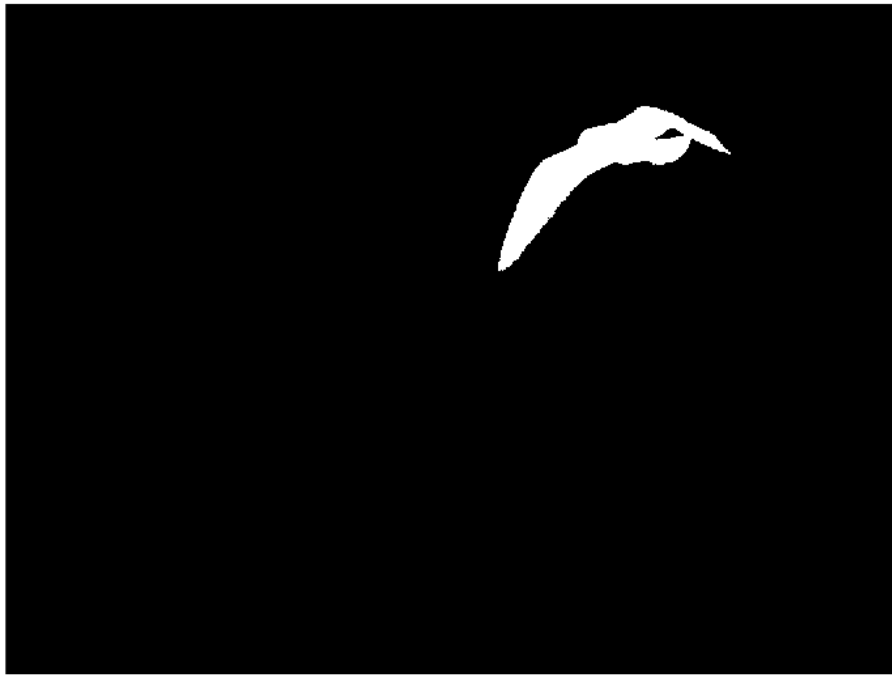
c) In welchem der Farbkanäle (RGB, HSI) lässt sich die Möwe besonders gut vom Hintergrund unterscheiden?

```
print('in HSI Kanal')
```



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.

```
moewe2_binary = ((moewe2_red < 0.4) + (moewe2_green < 0.2) +  
    (moewe2_blue < 0.3)) > 0;  
figure  
imshow(moewe2_binary)
```



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)`.

```
moewe_muster = repmat(uint8(moewe2_binary), [1 1 3]);  
moewe_only = moewe .* moewe_muster;  
figure  
imshow(moewe_only)
```



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.

```
bryce = imread('bryce.jpg');  
moewe2_binary_inversed = moewe2_binary == 0;  
moewe_muster2 = repmat(uint8(moewe2_binary_inversed), [1 1 3]);  
background = bryce .* moewe_muster2;  
  
figure  
final = background + moewe_only;  
imshow(final)
```




Published with MATLAB® R2019a