Informatik mit Matlab – Labor 8

Thema des Labors: Bildverarbeitung, RGB-Farbmodell, bmp-Files

Vorübung: Versuchen Sie die in der Vorlesung besprochenen Beispiele zum Laufen zu bringen. Die zu gehörigen M-Files finden Sie hier in diesem Verzeichnis.

Aufgabe 1: modifyColor()

Erstellen Sie die Funktion **modifyColor**, die aus einem farbigen Bild ein eingefärbtes, monochromes Bild erzeugt.
Als erstes wählt man über *uigetfile* den Namen einer bmp-Datei und liest die Bild-Daten ein. Dann startet der Farbdialog *uisetcolor*, mit dem der Anwender die Farbe wählt, mit der das Bild eingefärbt werden soll. Die Grauwerte zu den Bild-Daten werden berechnet und diese werden proportional zu der ausgewählten Farbe modifiziert. Das geänderte Bild wird auf dem Bildschirm angezeigt, mit dem korrekten Seitenverhältnis.

Aufgabe 2: createDetail()

Erzeugen Sie die Funktion **createDetail**, die nur die rechte Hälfte eines Bildes anzeigt.
Als erstes wählt man über *uigetfile* den Namen einer bmp-Datei und liest die Bild-Daten ein. Dann bestimmt man die Zahl der Zeilen und Spalten und erzeugt die Daten für ein neues Bild, das (als Detail) die rechte Hälfte des ausgewählten Bildes enthält, also alle Zeilen, aber nur die Spalten ab der Mitte des ursprünglichen Bildes. Zeigen Sie das geänderte Bild wird auf dem Bildschirm an, mit dem korrekten Seitenverhältnis.

Aufgabe 3: createImage()

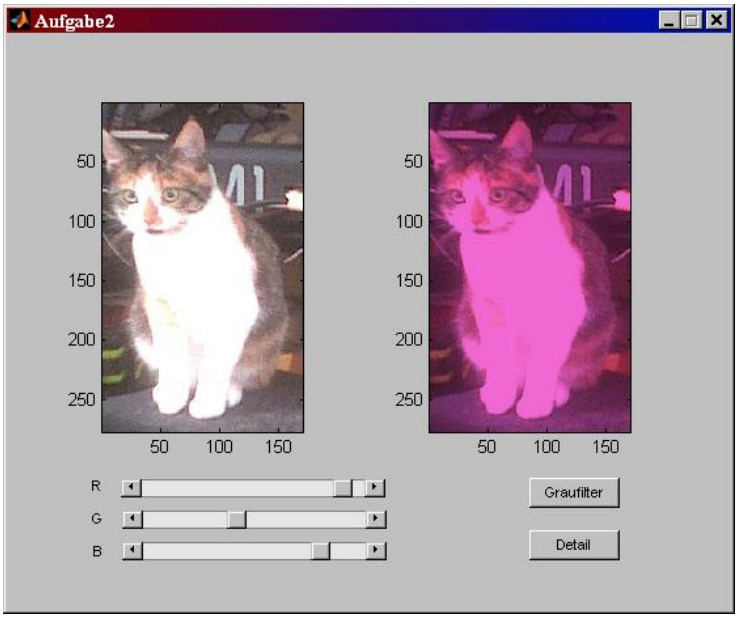
Die Funktion erzeugt ein RGB-Bild mit 255 Zeilen und 255 Spalten, also mit exakt 255x255 Pixeln, wobei der Rot-Wert eines Pixels $P(z,s)$ den Wert z der Zeile hat, der Grün-Wert den Wert s der Spalte und der Blau-Wert für alle Pixel 100 ist.
Das Bild wird angezeigt und anschließend abgespeichert - achten Sie dabei auf den korrekten Datentyp. Der Datei-Name zum Speichern wird über den GUI-Standard-Dialog *uiputfile* bestimmt, mit einem Test auf Abbruch des Dialogs durch den Anwender.

Aufgabe 4: p = calcArea()

Erstellen Sie die Funktion **calcArea**, die ähnlich arbeitet wie die in der Vorlesung besprochene Funktion *Schwerpunkt*, nur dass *calcArea* nicht den Schwerpunkt des Bildes berechnet, sondern seine relative Fläche, also die Anzahl der dunklen Pixel (Farbsumme kleiner als 100) in der Fläche dividiert durch die Gesamtzahl der Pixel des Bildes. Geben Sie den zugehörenden Prozentwert p zurück.

Zusatzaufgabe: Bildbearbeitung

In dieser Aufgabe sollen Sie ein bmp-Bild laden, z.B. *katze.bmp,* und seine Farben bzw. den Bildausschnitt bearbeiten. Hierzu können Sie die in der Vorlesung besprochenen Funktionen in *farben.m*, *grauFilter.m* und *detailBild.m* verwenden.



Erstellen Sie ein **GUI-Layout**
–> mit zwei Axes-Bereichen,
–> drei Slidern für die Farbmanipulation und
–> zwei Push-Buttons zum Aufruf von Graufilter bzw. Detail:

Im **linken** Axes-Bereich wird
–> das Originalbild dargestellt,
in der **rechten** Axes
die Modifikation, also
–> entweder eine Farbänderung über die Slider,
–> das Schwarz-Weiß-Bild, analog der Funktion *grauFilter.m*, oder
–> ein Detail, analog *detailBild.m*.

In der Opening-Funktion laden Sie am besten schon das Bild mit dem Befehl *imread* und zeichnen es mit *image* als Original in axes1. Die Wahl der aktuellen Axes zum Zeichnen erfolgt über

```
set( gcf, 'CurrentAxes', handles.axes1 ); % bzw. axes 2
```

Die aktuellen Daten merken Sie sich am besten im *handles*-struct des GUI-Fensters.

Farbänderungen: Standardmäßig lassen sich die Slider für die 3 Farbkanäle R, G und G zwischen den Werten 0 und 1 verschieben. Den aktuellen Wert des jeweiligen Sliders erhalten Sie im zugehörigen Callback über

```
val = get( hObject, 'Value' );
```

Multiplizieren Sie alle Pixel eines Farbkanals R, G oder B mit diesem Wert *val*, so erhalten Sie eine Farbmodifikation des Bildes für diesen Kanal.

Detail: Im Callback zum Detail-Push-Button starten Sie das Picken von 2 Punkten mittels *ginput*. Die ginput-Rückgabewerte legen die Grenzen des ausgewählten Details fest. Beachten Sie, dass der y-Wert die Zeilen und der x-Wert die Spalten der Farb-Matrix angibt. Sie brauchen also eine Bildpunkte-Zuordnung wie:

```
A2 = A1( ymin:ymax, xmin:xmax, : );
```