

Praktikum 1

Lena und Matlab

M.Thaler, 8/2014, ZHAW

1 Einführung

Ziel dieses Praktikums ist es, Sie mit den grundlegenden Bildverarbeitungsfunktionen in Matlab vertraut zu machen. Sie lernen Bilddaten aus Files einzulesen, zu verarbeiten, auszugeben und in verschiedenen Fileformaten abzuspeichern. Gleichzeitig machen Sie sich mit einigen grundlegenden Eigenschaften von Bilddaten wie Auflösung, Datentypen, Farbe, etc. vertraut.

Benötigte Bilder finden Sie im Praktikumsverzeichnis auf dem WEB. Speichern Sie diese Files an einem leicht zugänglichen Ort, z.B. in einem Ordner mit Namen `m:\private\bv\p01\imgs` (der gesamte Pfadname muss beim Einlesen mit Matlab angegeben werden).

2 Lernziele

- Sie können ein Bild aus einem File mit gängigem Format (`tif`, `jpg`, `gif`, `png`, `bmp`), in einen Datenarray einlesen, verarbeiten und in einem File mit einem der gängigen Formate abspeichern.
- Sie können mit den wichtigsten Datentypen (`uint8`, `double` and `logical`) umgehen und Bilder zwischen den Formaten konvertieren
- Sie wissen was eine Bit-Ebene ist, Sie können die einzelnen Bit-Ebenen eines Bildes darstellen und den jeweiligen Informationsgehalt diskutieren.
- Sie können die Auflösung von Bildern verändern und die resultierenden Effekte diskutieren.
- Sie können Bildregionen auswählen, weiterverarbeiten und wieder zusammensetzen.
- Sie wissen wie die Daten von Farbbildern organisiert sind.

3 Aufgaben

Arbeiten Sie die folgenden Aufgabenstellungen Schritt um Schritt durch und notieren Sie sich Ihre Beobachtungen und Resultate. Informationen zu den einzelnen Funktionen finden Sie in der Kurzanleitung "Image processing basics using Matlab" und im online Help zu Matlab (`doc NameDerFunktion`) oder der Dokumentation zur Image Toolbox.

3.1 Ein- / Ausgabe, Bildinformation und Datentypen

1. Bestimmen Sie die Informationen zum Bild im File `lena.tif`.
2. Lesen Sie das File `lena.tif` in den Bildarray `f` ein, geben Sie das Bild auf dem Bildschirm aus und schreiben Sie das Bild in Files mit den Formaten `jpg`, `gif` und `bmp`. Geben sie die Bildinformation zu den einzelnen Files aus und betrachten Sie die Information zum Bildarray `f`.

3. Konvertieren Sie das Bild `f` in den Datentyp `double` und geben Sie das Bild auf dem Bildschirm mit `imshow(f)` aus. Konvertieren Sie das Bild zurück in den Datentyp `uint8` und geben Sie das Bild wiederum auf dem Bildschirm aus.
4. Lesen Sie das Bild `lenaM.tif` ein und geben Sie es auf dem Bildschirm aus. Konvertieren Sie das Bild in den Datentyp `double`, skalieren Sie die Grauwerte auf den Bereich `0.0...1.0` und geben Sie das skalierte Bild aus.

3.2 Bit-Ebenen

Die Darstellung der einzelnen Bit-Ebenen¹ gibt anschaulich Auskunft wie viel Information in den einzelnen Bit-Ebenen überhaupt vorhanden ist und wieso i.A. ein Bild mit 8-Bit Genauigkeit dargestellt werden kann.

1. Stellen Sie die einzelnen Bit-Ebenen von `lena.tif` dar. Auf die einzelnen Bits eines Pixels können Sie mit Hilfe der Funktion `bin2dec()` zugreifen, z.B.:

```
bitplane3 = bin2dec('00001000'); % select bitplane 3
g = bitand(f, bitplane3);         % select bit 3 of all pixels
```

Skalieren Sie nun `g` und geben Sie das Bild als Binärbild aus.

2. Stellen Sie sämtliche Bit-Ebenen des Bildes einzeln dar und diskutieren Sie den Informationsgehalt der einzelnen Bit-Ebenen.
3. Wie viele der Bildebenen ab b_0 können Sie ausblenden, bis Effekte sichtbar werden?
4. Vergleichen Sie die Bilder `lenaB.tif` und `lena.tif` visuell. Stellen Sie einen Unterschied fest? Schauen Sie sich nun Bit-Ebene 0 von `lenaB.tif` an. Was lässt sich aus diesem Resultat schließen? Lassen sich Ihre Überlegungen beliebig verallgemeinern oder ist eher Vorsicht geboten?

3.3 Bildauflösung

In dieser Aufgabe werden wir die Auflösung (Abtastrate) eines Bildes verändern und die resultierenden Effekte diskutieren.

1. Geben Sie `lena.jpg` in verschiedenen Auflösungen aus. Für die Reduktion der Auflösung stellen wir Ihnen die Funktion `reduceRes(img, sz)` zu Verfügung. Die Funktion teilt das Bild in Blöcke der Grösse $sz \times sz$ auf und setzt alle Pixel eines Blockes gleich dem Mittelwert der ursprünglichen Pixel in diesem Block.

Ab welchem Wert von sz wird die Reduktion der Auflösung sichtbar? Welchen Wert (qualitativ) erhalten Sie, wenn Sie sz gleich gross wie das Bild wählen?

2. Lesen Sie nun das File `rose.tif` ein und wählen Sie wiederum verschiedene Auflösungen. Was stellen Sie fest? Vergleichen Sie das Resultat mit obigem Punkt.

¹als Bit-Ebene bezeichnet man alle gleichen Bits (z.B. Bit Nr. 4) der Grauwerte der einzelnen Pixel, Bit-Ebenen können als Schwarz/Weissbilder dargestellt werden

3.4 Bildregionen

Bringen Sie das Bild `lenaV.tif` in Ordnung. Schneiden Sie dazu die entsprechenden Regionen aus und fügen Sie sie wieder korrekt zusammen. Hinweis: alle Rechtecke sind gleich gross.

Beliebige rechteckige Regionen eines Bildes können in Matlab mit der *Bereichsfunktion* ausgeschnitten und wieder zusammengefügt werden. Z.B. kann ein rechteckiger Bereich ab Pixel $f(sx, sy)$ der Grösse $m \times n$ wie folgt ausgeschnitten werden (Annahme sy, sy, m und n sind bekannt:

```
g = f(sx:sx+m-1, sy:sy+n-1);
```

Vergessen Sie nicht, dass Arrays in Matlab bei Index 1 beginnen.

Hinweis: auf eine Bildzeile oder Bildspalte kann folgendermassen zugegriffen werden:

```
zeile = f(3, :);    % Zugriff auf Zeile 3  
spalte = f(:, 2);   % Zugriff auf Spalte 4
```

3.5 Farbbilder

1. Lesen Sie das Bild `lenaC.tif` und geben Sie die Informationen zum Bildarray und zum File aus.
2. Setzen Sie jeweils zwei Ebenen des 3-dimensionalen Arrays auf 0 und geben Sie den Bildarray aus. Welche Schlüsse bezüglich Organisation der Bilddaten können Sie ziehen?
3. Kopieren Sie jeweils eine Ebene des 3-dimensionalen Arrays in einen 2-dimensionalen Array und geben sie diesen als Graustufenbild aus.