



PROF. DIPL.-INF. INGRID SCHOLL

KATHRIN PETTERS, M.ENG.

Praktikumsbuch Bildverarbeitung

BACHELOR INFORMATIK

WS 2018/19

Inhaltsverzeichnis

0 Hinweise und Vorbereitung	2
0.1 Allgemeine Hinweise	2
0.2 Ablauf des Praktikum	2
0.3 Arbeitsplattform	2
0.3.1 Matlab	3
0.3.2 OpenCV	3
1 Aufgabe: Grundlegende Operationen und Farbkonvertierungen	4
1.1 Anaglyphenbilder	4
1.2 Farbsysteme	5
1.3 Aufgabenstellung	7
2 Aufgabe: Bildverbesserung durch Histogrammänderungen mit GUI	8
2.1 Histogrammanalyse	8
2.2 Aufgabenstellung	11

0 Hinweise und Vorbereitung

0.1 Allgemeine Hinweise

1. Lesen Sie die Aufgabenstellungen vollständig und aufmerksam durch. Bearbeiten Sie die Aufgaben vollständig und in vorgegebener Reihenfolge.
2. Der Aufwand kann zwischen den Aufgaben deutlich variieren. Bei manchen Teilen reicht es den Anweisungen zu folgen. Bei anderen hingegen müssen Sie selbstständig recherchieren und die Lösung erarbeiten. Planen Sie daher für jede Aufgabe genügend Zeit ein.
3. Zur Abgabe benötigen Sie eine lauffähige Version Ihrer Lösung. Überlegen Sie, ob sich für Spielereien oder Experimente eine Versionskontrolle (git) lohnt.
4. Stellen Sie sicher, dass Sie jede Methode bzw. Funktion und deren Parameter erklären können, wenn Sie sie verwenden. Wenn Sie sich unsicher sind, schlagen Sie die Funktion in der jeweiligen Dokumentation nach.
5. Formulieren Sie Suchanfragen auf englisch. Das erleichtert die Suche und Sie erhalten schneller eine brauchbare Lösung.

0.2 Ablauf des Praktikum

Sie erhalten in diesem Dokument sechs Aufgaben als Tutorial. Diese sind Pflicht für alle. Die Bearbeitung sieht ca sechs Wochen vor, die genauen Termine entnehmen Sie bitte den Ilias Abgaben. Diese Aufgaben sollen Ihnen eine Einführung in die Bildverarbeitung, sowie die Verwendete Bibliothek und Entwicklungsumgebung geben.

Im Anschluss an diese Aufgaben können Sie zwischen weiteren drei Aufgaben oder einem Projekt wählen. Die Aufgaben sind dabei unabhängig von den bisherigen Aufgaben und bilden jede für sich ein Teilgebiet der Bildverarbeitung ab. Das Projekt können Sie frei wählen. Falls Sie etwas interessiert, was Sie immer schon mal umsetzen wollten, können Sie das hier tun.

Zum Ende des Semesters gibt es einen Abschlusstermin, der für alle Teilnehmer Pflicht ist. Hier werden die Projekte vorgestellt und in kurzen ca 10-15 minütigen Präsentationen erläutert.

0.3 Arbeitsplattform

Das Praktikum kann in Matlab oder OpenCV gelöst werden. Beide Möglichkeiten werden im Praktikum möglichst gleich verteilt unterstützt. Im Folgenden werden Ihnen zu beiden Varianten ein Überblick und Hinweise gegeben. Entscheiden Sie selber, was sie wählen möchten.

0.3.1 Matlab

Matlab ist eine Software, die Sie über die FH-Aachen kostenlos installieren können. Es steht eine sogenannte Campuslizenz auf der Homepage zur Verfügung. <https://www.fh-aachen.de/hochschule/datenverarbeitungszentrale/soft-und-hardware/matlab/> (Login nicht vergessen). Der Name leitet sich von MATrix LABoratory ab. Es integriert die Entwicklungsumgebung, ein Tool zu GUI-Erstellung und die Bildverarbeitungsfunktionen. Programmiert wird in einer C++ ähnlichen Sprache, Matlab verhält sich aber hier und da anders. Durch die gute Dokumentation ist es aber leicht, sich an diese Umgebung zu gewöhnen.

Matlab ist jedoch nicht sehr objektorientiert. Die kompletten Variablen-Deklarationen regelt es intern und man kann nur durch Überprüfung zur Laufzeit Typen erkennen und zur Kompatibilität miteinander vergleichen.

0.3.2 OpenCV

Die aktuelle Version ist OpenCV 3.4.3 und wird in diesem Praktikum unterstützt. Grundsätzlich sind Sie aber bei der Versionswahl frei. Das Dokument gibt Ihnen Hinweise basierend auf OpenCV 3.4.3, bei anderen Versionen kann es zu Fehlern kommen.

Neben der gewünschten OpenCV Version sollen Sie die Zusatzmodule „opencv-contrib“ installieren. Sie finden diese unter https://github.com/opencv/opencv_contrib/releases für die verschiedenen Versionen. Wählen Sie den passenden Release zu Ihrer OpenCV Version.

Für die Installation unter Windows finden Sie unter https://docs.opencv.org/master/d3/d52/tutorial_windows_install.html eine Anleitung. Für Ubuntu ist sie unter https://docs.opencv.org/master/d7/d9f/tutorial_linux_install.html zu finden.

Hinweise zur Installation unter Windows (für Ubuntu ggf auch relevant!):

- Sie können oben links auf der Tutorial Seite die richtige OpenCV Version wählen
- Beachten Sie alle relevanten Abhängigkeiten, z.B. CMake Version
- Fügen Sie `#define STRSAFE_NO_DEPRECATED` in die Datei `$OPENCVFolder$\sources\modules\videoio\src\cap_dshow.cpp`
- aktivieren Sie c++11 für die Installation
- deaktivieren Sie `WITH_MSMF`

Kommt es zu weiteren Fehlern, so könnten diese systembedingt sein. Bitte versuchen Sie mit Hilfe des Internets hierfür jeweils eine Lösung zu finden. Bei weiteren Fragen stehen wir Ihnen natürlich auch gerne zur Verfügung.

1 Aufgabe: Grundlegende Operationen und Farbkonvertierungen

Ein Farbbild besteht pro Pixel aus RGB-Werten. Jedes Pixel wird somit als ein (rot,grün,blau)-Tripel repräsentiert. Das Farbbild besteht also aus einer Pixelmatrix mit diesen Tripeln. Diese Matrix kann in 3 Matrizen (pro Farbkanal eine) zerlegt werden. Abbildung 1 zeigt, wie die drei Kanäle als Grauwertbild aussehen, Abbildung 2 zeigt wie ein RGB Tripel beispielhaft zu einer Pixelregion (region of interest) aussehen kann.

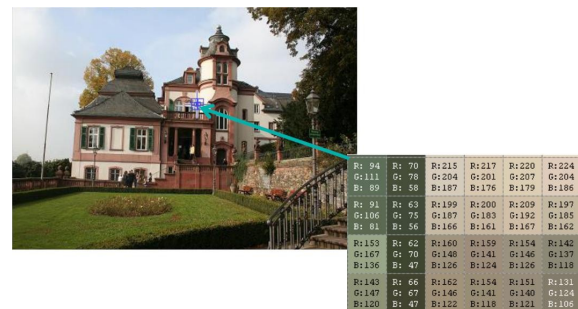


Abbildung 1: Farbbild RGB (drei Kanal-Bild)

Abbildung 2: Farbbild RGB (drei Kanal-Bild)

1.1 Anaglyphenbilder

Eine alte Technik, 3D-Bilder zu erzeugen, war die Erstellung von Anaglyphenbildern. Hierbei werden zwei Bilder in einem Abstand von ca dem Augenabstand aufgenommen und dann zu einem zusammengefasst. Aus dem linken Bild wird der Rot-Kanal genutzt und aus dem rechten grün und blau (zusammen cyan).

Beispiele für solche Bilder sind in Abbildung 3a und 3b zu sehen. Das zusammengebaute Bild ist in Abbildung 4 gezeigt.



Ausgabe des linken Bildes.



Ausgabe des rechten Bildes.



Abbildung 4: So sieht das Anaglyphenbild aus

1.2 Farbsysteme

HSV

HSV ist ein wahrnehmungsangepasstes Farbmodell mit den drei Kanälen **H**ue-**S**aturation -**V**alue, also Farbton-Sättigung-Intensität/Helligkeit. Man kann also die Helligkeit ohne Veränderung der Farbwerte anpassen.

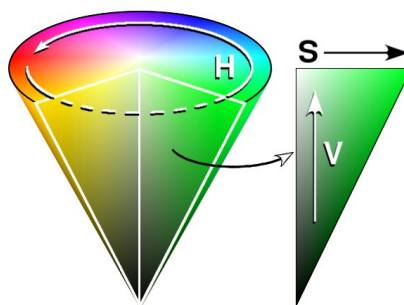


Abbildung 5: HSV als Zylinderkoordinaten

Die Umrechnung von RGB nach HSV wird nach folgenden Regeln berechnet:

$$V = \max(r, g, b) \quad S = \frac{(V - \min\{r, g, b\})}{V} \quad \begin{aligned} \max = R : H &= 2 + \frac{(g-b)}{\max-\min} \\ \max = G : H &= 2 + \frac{(b-r)}{\max-\min} \\ \max = B : H &= 2 + \frac{(r-g)}{\max-\min} \end{aligned}$$

CMY

CMY ist ein subtraktives Farbsystem mit den drei Grundfarben Cyan, Magenta und Gelb/Yellow. Der zugehörige Farbwürfel ist in Abbildung 6 dargestellt.

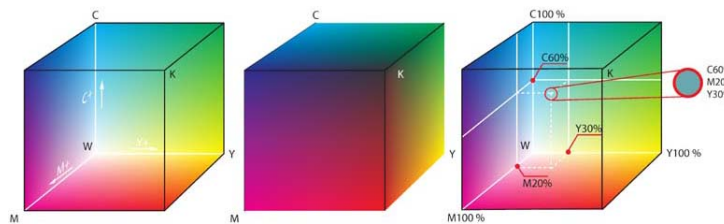


Abbildung 6: CMY-Farbwürfel

Die Umrechnung von RGB nach CMY wird bei 8 Bit folgendermaßen berechnet:

$$\begin{pmatrix} C \\ M \\ Y \end{pmatrix} = \begin{pmatrix} 255 \\ 255 \\ 255 \end{pmatrix} - \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Lab

Lab verwendet die CIE Spezifikation und skaliert die Helligkeit nicht linear aber wahrnehmungsangepasst. Zum Umwandeln von RGB nach Lab muss zunächst in CIE-XYZ und dann in Lab umgewandelt werden. Die entsprechende Tabelle hierfür entnehmen Sie bitte der Vorlesung.

Die Umwandlung von XYZ nach Lab wird folgendermaßen berechnet:

$$L^* = 116 \cdot h\left(\frac{Y}{Y_W}\right) - 16$$

$$a^* = 500 \cdot \left[h\left(\frac{X}{X_W}\right) - h\left(\frac{Y}{Y_W}\right) \right]$$

$$b^* = 200 \cdot \left[h\left(\frac{Y}{Y_W}\right) - h\left(\frac{Z}{Z_W}\right) \right]$$

$$h(q) = \begin{cases} \sqrt[3]{q} & \text{mit } q > 0.008856 \\ 7.78q + \frac{16}{116} & \text{mit } q \leq 0.008856 \end{cases}$$

YIQ

Bezeichnet Helligkeit Y, Farbdifferenzen I (Cyan-Orange Balance) und Q (Magenta-Grün Balance) und wird von NTSC-Farbfernsehsystemen verwendet. Es ist zur Umwandlung in ein Grauwertbild (Y-Kanal) geeignet. Die Umwandlung von RGB zu YIQ für eine spezifische NTSC-Version sieht folgendermaßen aus:

$$R, G, B, Y \in [0, 1], \quad I \in [-0.5957, 0.5957], \quad Q \in [-0.5226, 0.5226]$$

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.595716 & -0.274453 & -0.321263 \\ 0.211456 & -0.522591 & 0.311135 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

1.3 Aufgabenstellung

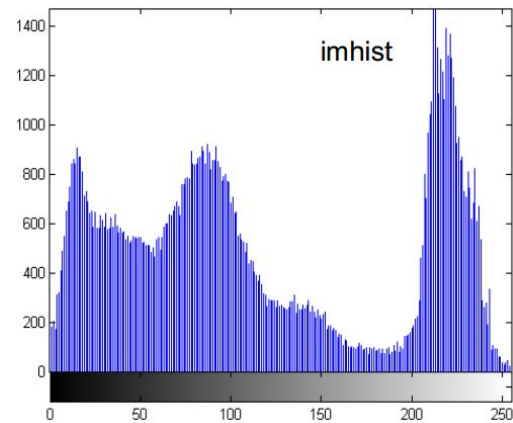
1. Lesen Sie ein beliebiges Bild ein (Matlab und OpenCV: imread)
2. Stellen Sie das Bild in einem Fenster dar (Matlab: imshow, imtool | OpenCV: imshow)
3. Zerlegen Sie das Bild in seine drei Kanäle und zeigen Sie diese an (Matlab: $r = I(:, :, 1)$ | OpenCV: split(I,channels))
4. Erzeugen Sie zwei Bilder in einem Abstand von ca. 6,5cm und generieren Sie daraus ein Anaglyphenbild.
5. Konvertieren Sie das Bild in folgende Farbräume und zeigen Sie jeweils das Ergebnis und die drei Kanäle an:
 - HSV - Nutzen Sie gegebene Funktionen (Matlab: rgb2hsv(...) | OpenCV: cvtColor(...))
 - CMY – Vergleichen Sie die gegebenen Funktionen mit einer selbst implementierten Funktion
 - YIQ (NTSC) – Verwenden Sie hier die Linearkombination, die in der vorangestellten Kapiteleinführung beschrieben wurde
 - Lab – Verwenden Sie hier die gegebenen Funktionen von Matlab und OpenCV
6. Bereiten Sie Ihre Ausgaben so vor, dass sie die verschiedenen Farbsysteme nacheinander zeigen können.

2 Aufgabe: Bildverbesserung durch Histogrammänderungen mit GUI

Histogramme sind Häufigkeitsverteilungen. Bei Bildern beschreiben sie die Häufigkeit der einzelnen Farb-/Grauwerte. In Abbildung 7a ist wieder das Bild aus Aufgabe 1 zu sehen, Abbildung 7b zeigt das zugehörige Histogramm zu diesem Grauwertbild.



(a) Grauwertbild mit 256 Grauwerten



(b) Histogramm vom Grauwertbild, x-Achse: Grauwertintensität g und y-Achse: Häufigkeit ($h(g)$)

Abbildung 7: Bild mit zugehörigem Histogramm

Ein Histogramm für alle Grauwerte mit $0 < g < G - 1$ hat i.d.R. $G = 2^8 = 256$ Häufigkeiten $h(g)$ (0..255). Bei Grauwertbildern gilt: $h(g)$ ist die Anzahl der Pixel mit Intensitätswert g .

Histogramme enthalten keine Informationen darüber, wo die Grauwerte im Bild auftreten. Sämtliche räumliche Informationen gehen verloren, einzig die Auftrittswahrscheinlichkeit wird abgebildet. Daher kann es sein, dass verschiedene Bilder das gleiche Histogramm haben.

Es gibt drei verschiedene Arten von Histogrammen:

- „Normales“ Histogramm entspricht der Häufigkeitsverteilung der Grauwerte.
- Normiertes Histogramm entspricht der Verteilung der Auftrittswahrscheinlichkeiten der Grauwerte. Dies wird auch relative Häufigkeit genannt und berechnet sich durch: $p(g) = h(g)/(M \cdot N)$, wenn das Bild $M \times N$ Pixel groß ist.
- Kumulatives Histogramm entspricht der summierten Auftrittswahrscheinlichkeit bis zu einem Grauwert. Berechnet wird es durch: $h_{cum}(g) = \sum_{i=0}^g p(i)$

Beispiele dieser drei Typen sind in Abbildung 8 gezeigt.

2.1 Histogrammanalyse

Bei der Histogrammanalyse können zwei Dinge betrachtet werden. Zum Einen der Intensitätsbereich (Sättigung). Bei Belichtungsfehlern kann es hier zu Histogrammen kommen, bei denen sich die Pixel

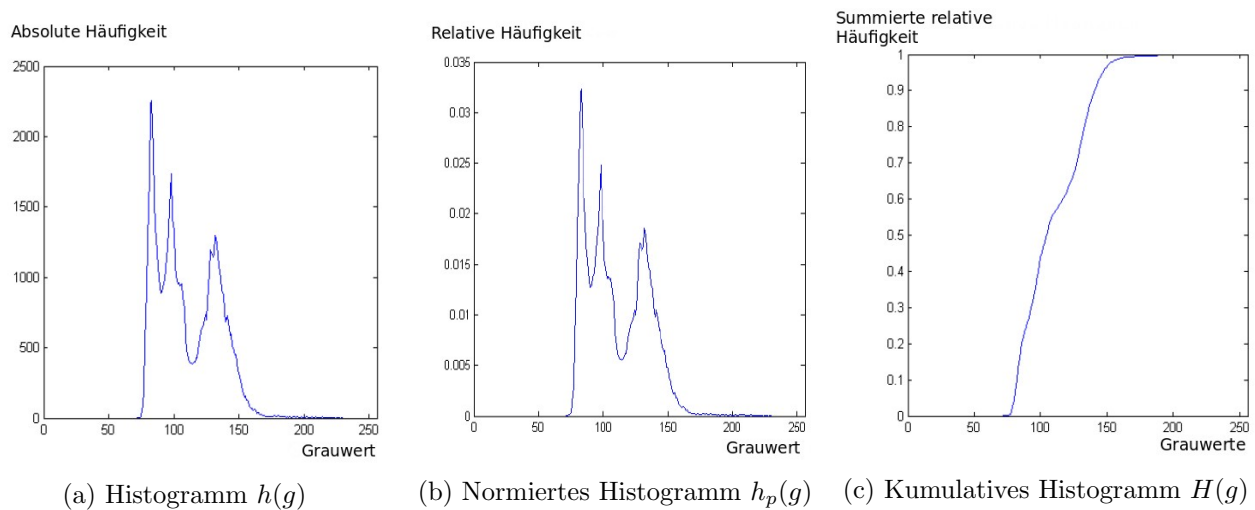


Abbildung 8: Histogrammvarianten

vorwiegend im dunklen oder hellen Bereich, sprich links oder rechts im Histogramm häufen.

Zum Anderen kann die Dynamik im Bild beobachtet werden. Sie beschreibt die Anzahl verschiedener Farb- bzw. Grauwerte im Bild. Eine weitere Eigenschaft ist der globale Kontrast, er ist der Bereich der genutzten Intensitätsstufen. Er wird durch die Differenz zwischen maximalen und minimalen vorkommenden Grauwerten berechnet: $C_{global} = g_{max} - g_{min}$. Ein voller Kontrast nutzt den gesamten Intensitätsbereich.

Wird bei einem der beiden Dinge ein Ungleichgewicht festgestellt, können ein Histogrammstretching, Gammakorrektur oder Histogrammlinearisation verwendet werden.

Histogramm-Stretching

Hierbei wird eine Grauwerttransformation durchgeführt. Dazu wird eine Funktion berechnet, die den Grauwert g auf einen neuen Grauwert g' abbildet. Zur Vergrößerung des globalen Kontrastets durch Grauwertspreizung wird folgende Funktion verwendet:

$$g'(g) = (w_{max} - w_{min}) \cdot \left(\frac{g - g_{min}}{g_{max} - g_{min}} \right) + w_{min}$$

wobei w_{min} und w_{max} den kleinsten bzw. größten neuen darstellbaren Grauwert definiert, z.B. bei 256 Grauwerten ist $w_{min} = 0$ und $w_{max} = 255$. g_{min} und g_{max} definiert den kleinsten bzw größten aktuell dargestellten Grauwert.

In Abbildung 9 ist das Ergebnis eines Histogramm-Stretching an einem Beispielbild gezeigt. Abbildung 9a zeigt das Originalbild und Abbildung 9b das gestretchte Bild.

Gamma-Korrektur

Hierbei wird eine Korrektur nichtlinearer Kamerakennlinien (Grauwerte sind nicht proportional zur Lichtintensität) durchgeführt. Die Formel vom Histogramm-Stretching wird hier lediglich um γ erweitert:



(a) Original-Bild



(b) Grauwertspreizung

Abbildung 9: Histogrammstretching Beispiel

$$g'(g) = (w_{max} - w_{min}) \cdot \left(\frac{g - g_{min}}{g_{max} - g_{min}} \right)^\gamma + w_{min}$$

$\gamma = 1$: Gamma hat keinen Einfluss und es wird ein normales Histogrammstretching gemacht

$\gamma < 1$: Spreizung der helleren und Stauchung der dunkleren Grauwerte

$\gamma > 1$: umgekehrt zu $\gamma < 1$, d.h. Spreizung der dunkleren und Stauchung der helleren Grauwerte

Die Gammakorrektur ist eine nicht-lineare Grauwertspreizung bei der mit $\gamma < 1$ eine Aufhellung und mit $\gamma > 1$ eine Verdunkelung durchgeführt wird.

Beispiel Aufhellung:

$$w_{min} = g_{min} = 0 \text{ und } w_{max} = g_{max} = 255$$

$$g = 15 \text{ und } \gamma = 0.5$$

$$g'(g = 15) = 255 \cdot \left(\frac{15 - 0}{255 - 0} \right)^{0.5} + 0 = 255 \cdot \left(\frac{15}{255} \right)^{0.5} = 61,84$$

Der Grauwert hat sich von $g = 15$ zu $g' = 61,84$ verändert \Rightarrow Aufhellung.

Beispiel Verdunklung:

$$w_{min} = g_{min} = 0 \text{ und } w_{max} = g_{max} = 255$$

$$g = 15 \text{ und } \gamma = 1.5$$

$$g'(g = 15) = 255 \cdot \left(\frac{15 - 0}{255 - 0} \right)^{1.5} + 0 = 255 \cdot \left(\frac{15}{255} \right)^{1.5} = 3,63 \Rightarrow \text{verdunkelt}$$

Histogrammlinearisierung

In der Kompression liefert die Entropie die durchschnittliche Anzahl der benötigten Bits einer Folge von Pixeln (max. Kompressionsfaktor):

$$E(h_p) = - \sum_{g=0}^{G-1} h_p(g) \cdot \log_2(h_p(g))$$

Die Entropie kann zur Kontrastverbesserung verwendet werden. Dazu muss eine Transferfunktion g' berechnet werden, bei der die Grauwerte g auf neue Grauwerte abgebildet werden, so dass die Entropie des Histogramms $h_p(g')$, der Anzahl der verwendeten Bits (i.d.R. 8) entspricht.

Ziel ist die Maximierung der Entropie. Dazu wird das kumulative Histogramm als Transferfunktion g' verwendet:

$$w_{max} = 255 : g'(g) = w_{max} \cdot H_{cum}(g) = w_{max} \cdot \sum_{i=0}^g p(i)$$

Auf Integerzahlen runden:

$$w_{max} = 255 : g'(g) = \lceil w_{max} \cdot H_{cum}(g) \rceil - 1 = \lceil w_{max} \cdot \sum_{i=0}^g p(i) \rceil - 1$$

Abbildung 10 zeigt ein Beispiel der Histogrammlinearisierung mit dem Originalbild in ?? und dem verbesserten Bild in 10b

Befehl Matlab: `histeq(image)`, Befehl OpenCV: `equalizeHist(InputArray src, OutputArray dst)`



(a) Vor Histogrammlinearisierung



(b) Nach Histogrammlinearisierung

Abbildung 10: Beispiel Histogrammlinearisierung

2.2 Aufgabenstellung

Entwickeln Sie ein Graphical User Interface (GUI), das ein Bild über eine Histogrammänderung verbessert. Dazu soll ein beliebiges Bild über einen Einlese-Dialog vom Dateisystem ausgewählt werden. Falls es sich nicht um ein Grauwertbild handelt, soll dieses automatisch in ein Grauwertbild umgewandelt

werden.

Stellen Sie das Bild selbst wie auch das dazugehörige Histogramm grafisch dar.

Die GUI soll über ein Menü zur Auswahl einer Histogramm-Methode verfügen. Hilfen zur GUI Programmierung finden Sie in den Handbüchern. Stellen Sie folgende Methoden zur Verfügung:

- Histogrammstretching (Folie 2-15) (Matlab: `imadjust+stretchlim` | OpenCV: nicht vorhanden)
- Gammakorrektur mit Eingabe von Gamma-Wert (Folie 2-19) (Matlab/OpenCV: Erweitern Sie die Funktion aus dem Stretching)
- Histogrammlinearisation (Folie 2-23) (Matlab: `histeq()` | OpenCV: `equalizeHist()`)
- Histogrammstretching + Gammakorrektur
- Histogrammstretching + Histogrammlinearisation
- Zurück setzen auf Original

Wählen Sie für diese Aufgabe ein eher zu dunkles Bild um einen Effekt sehen zu können. Das Beispiel aus der Vorlesung zeigt den Effekt beispielhaft:

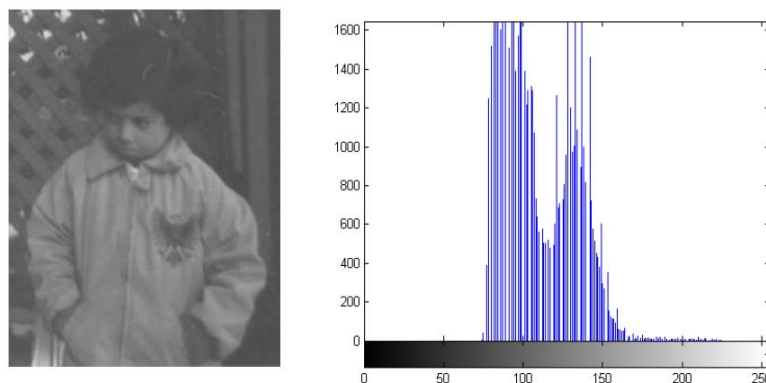


Abbildung 11: Originalbild mit Histogramm

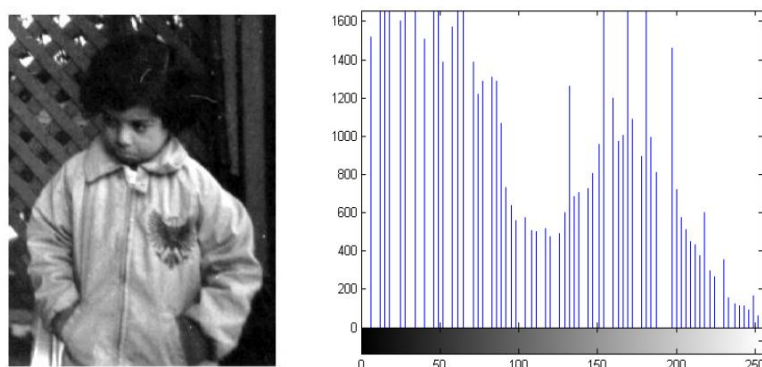


Abbildung 12: Bild nach Histogrammstretching