

8. JUNI 2023

VISUAL COMPUTING SoSe 2023

AUFGABE 5

5.1 Lineare Grauwert-Transformation

Laden Sie das Bild "schrott.png" ein. Mittels der im Moodle zu findenden Funktion `plotHistogram` können Sie sich das Grauwert-Histogramm des Bildes anschauen. Was ändert sich, wenn Sie `cumulative` auf `true` setzen?

Auf diesem Bild soll nun eine lineare Grauwert-Transformation durchgeführt werden, um den Kontrast künstlich zu verringern. Hierfür benötigen Sie zuerst die Werte des maximalen und minimalen Grauwerts. Skalieren Sie dann das Bild derart, dass es einen eingeschränkten Kontrastumfang von 100 bis 150 besitzt (also der minimale Helligkeitswert 100 beträgt und der maximale 150).

Im nächsten Schritt soll durch erneute Skalierung des kontrastarmen Bildes der Kontrastumfang maximiert werden (also den Wertebereich $[0...255]$ abdecken) und so sogar den des Originalbildes übersteigen! Vergleichen Sie die Histogramme des resultierenden Bildes und des Originalbildes, was fällt auf und wie lässt sich dieser Unterschied erklären?

5.2 Histogram Equalization

Aus der Vorlesung (und der Aufgabe 5.1) kennen Sie eine Möglichkeit, den Kontrastumfang eines Bildes zu maximieren. Dies ist häufig erwünscht, da so Bilddetails besser erkannt werden können. Allerdings kann eine einfache Skalierung des Bildes eine Ungleichverteilung von hellen und dunklen Pixeln nicht verhindern. Das Bild schrott.png beispielsweise enthält verhältnismäßig viele Pixel in mittlerem Graubereich, der "lokale" Bildkontrast wird daher durch eine Skalierung nur minimal verändert, Bilddetails werden nicht verstärkt.

Eine Weiterentwicklung der Skalierung ist die Histogrammentzerrung (eng. Histogram Equalization), die dafür sorgt, dass alle möglichen Pixelintensitäten ungefähr gleich häufig im Bild vorkommen. Implementieren Sie den folgenden Algorithmus und testen Sie ihn am schrott.png (Sie können hierfür die Hilfsmethode `plotHistogram` anpassen, wenn Sie möchten).

- Schritt 1: Berechnung des Histogramms
- Schritt 2: Berechnung des akkumulierten Histogramms
- Schritt 3: Berechnung der kumulativen Verteilungsfunktion (eng. CDF): Teilen des akkumulierten Pixelzählers durch die Bildauflösung (also max. Pixelanzahl)
- Schritt 4: Multiplizieren der Verteilungsfunktion mit der höchstmöglichen Pixelintensität des Zielbildes (normalerweise 255) und Abrunden auf den nächsten Integer
- Schritt 5: Wenden Sie die resultierende Abbildung auf jeden Pixel des Input-Bildes an.

Ein Beispiel für die Berechnung und die Funktionsweise des Algorithmus' finden Sie in Abbildung 1, für ein 4-Bit Inputbild.

Wenn Ihre Implementierung funktioniert, werfen Sie einen Blick auf das resultierende Histogramm. Was passiert im akkumulierten Histogramm? Wie verhält sich der Algorithmus bei dem angepassten Bild mit verringertem Kontrast aus Aufgabe 5.1?

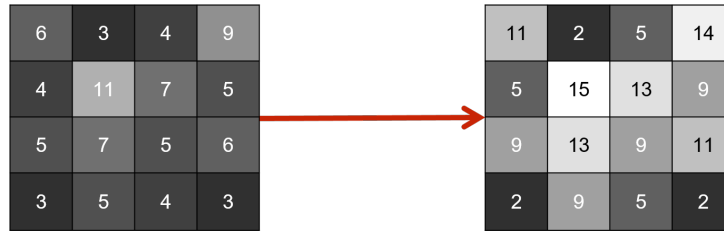


Abbildung 1: Beispielumwandlung für ein Inputbild mit 4-Bit Dynamikumfang

Pixel Intensität	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Pixelanzahl	0	0	0	3	3	4	2	2	0	1	0	1	0	0	0	0
Kumulativ	0	0	0	3	6	10	12	14	14	15	15	16	16	16	16	16
Kumulative Verteilung	0	0	0	0,1 875	0,3 75	0,6 25	0,7 5	0,8 75	0,8 75	0,9 375	0,9 375	1	1	1	1	1
Abbildung	0	0	0	2	5	9	11	13	13	14	14	15	15	15	15	15

Abbildung 2: Beispielberechnung für ein Inputbild mit 4-Bit Dynamikumfang

5.3 Faltung von Bildern

Bevor wir uns der Faltungsberechnung in OpenCV widmen, hilft es, sich die zugrundeliegende Berechnung einmal genauer vor Augen zu führen. Das Bild "kante.png" soll als Beispiel dienen, die Pixelwerte hierfür finden Sie in der rechten Tabelle.

Definieren Sie sich selber eine sinnvolle 3x3 Faltungsmatrix für einen Gaußschen Weichzeichner (siehe Vorlesung) und falten Sie per Hand (auf Papier) das ursprüngliche kante.png mit der Matrix.

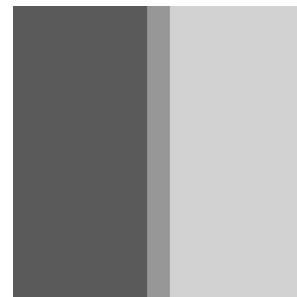
Überlegen Sie sich hierfür, wie Sie mit fehlenden Rand-Werten umgehen wollen. Welche Alternativen gibt es hierfür?

Bewerten Sie das resultierende Bild, was genau ist passiert? Was können Sie über den lokalen Bildkontrast sagen (also die Helligkeitsdifferenz benachbarter Pixel)?

Anschließend wollen wir die integrierte Faltung von OpenCV testen:

```
C++:
cv::filter2D (InputArray src,
              OutputArray dst,
              int ddepth,
              InputArray kernel,
              Point anchor = Point(-1,-1),
              double delta = 0,
              int borderType = BORDER_DEFAULT
              )
```

```
Python:
cv2.filter2D( src,
              ddepth,
              kernel[,
              dst[,
              anchor[,
              delta[,
              borderType]]]] ) -> dst
```



90	90	90	150	210	210	210
90	90	90	150	210	210	210
90	90	90	150	210	210	210
90	90	90	150	210	210	210

Abbildung 3: Kante.png

Gehen Sie dafür wie folgt vor:

- Laden Sie das Bild "kante.png" und führen Sie eine Faltung mit der Faltungsmatrix F1 durch. (Setzen Sie ddepth auf -1, dann entspricht die Bildtiefe des Ergebnisbildes dem Eingabebild.)
- Die Ergebniswerte liegen hierdurch nahe null (da die summierten Gewichte der Faltungsmatrix null ergeben, nicht eins). Addieren Sie daher zur Faltung den Wert 128, dies geht über den Parameter **delta**.
- Wenden Sie auf das Ergebnis die Faltungsmatrix F2 an (wieder mit delta=128).
- Wenden Sie auf eine Kopie des ursprünglichen Bildes die Faltungsmatrix F3 an (delta=128).

$$F_1 = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$F_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$F_3 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$F_4 = \begin{bmatrix} 0 & 0 & 0 \\ 0,333 & 0,333 & 0,333 \\ 0 & 0 & 0 \end{bmatrix}$$

$$F_5 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$F_6 = \begin{bmatrix} 0 & 0 & 0 \\ 0,333 & -0,666 & 0,333 \\ 0 & 0 & 0 \end{bmatrix}$$

Fragen:

- 1) Was liefert der Vergleich der beiden zuletzt erzeugten Ergebnisse?
 - 2) Die beiden Faltungsmatrizen F1 und F2 differenzieren das Bild (d.h. sie bilden jeweils die Ableitung) in Zeilenrichtung. Was bewirkt demzufolge die Faltungsmatrix F3 bzgl. des Begriffes "Ableitung"?
- Falten Sie eine Kopie des ursprünglichen Bildes mit der Faltungsmatrix F4.
 - Falten Sie unabhängig davon eine weitere Kopie des ursprünglichen Bildes mit der Faltungsmatrix F5.
 - Wenn Sie jetzt das mit F5 erzeugte Bild von dem mit F4 erzeugten Bild subtrahieren würden, so entspräche das einer Faltung des ursprünglichen Bildes mit der Faltungsmatrix F6, die eine Subtraktion dieser zwei Faltungsmatrizen ist: $F_6 = F_4 - F_5$. Führen Sie die Faltung mit F6 durch (delta=128).

Fragen:

- 3) Was liefert der pixelweise Vergleiche des Ergebnisses der Faltung mit F6 mit dem Ergebnis der Faltung mit F3? Begründen Sie die Unterschiede.
- 4) Was bewirken die beiden Faltungen F4 bzw. F5?
- 5) Was erhält man, wenn man die Faltungsmatrix F3 um 90 Grad dreht und zu F3 selbst addiert? Wie heit dieser Operator und was bewirkt er bzgl. des Begriffes "Ableitung"?