

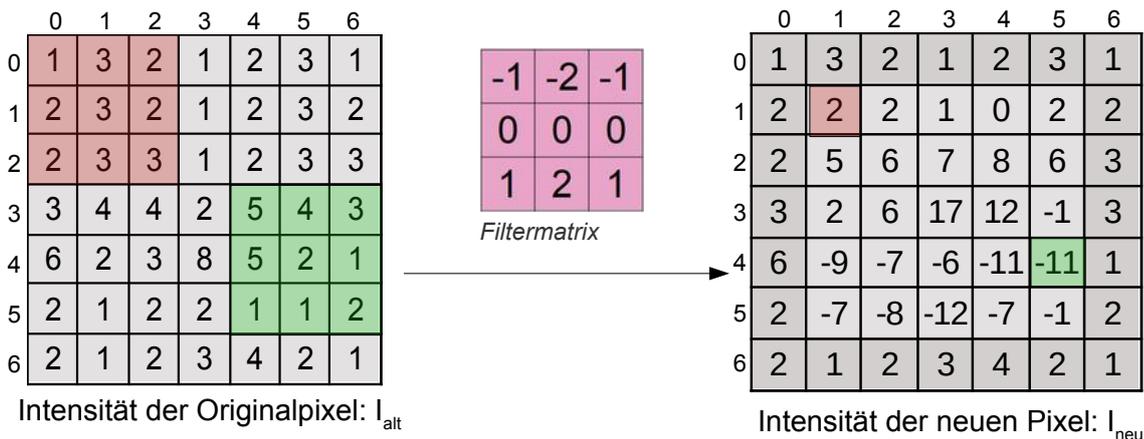


Grundidee: Die umliegenden Pixel beeinflussen die Farbe eines Pixels

Für die Berechnung der Intensität eines neuen Pixels werden die Intensitäten der umliegenden Pixel mit einem festgelegten Faktor multipliziert und dann alle addiert. Dies wird als Faltung bezeichnet.

Hinweis: auf den Rand kann man den Filter nicht so einfach anwenden, wir lassen ihn daher so wie er ist.

Beispiel:



Auftrag:

1. Berechne die restlichen Werte des oben angegebenen Bildes.
2. Öffne die Datei 03_faltung.ods. Das Beispiel aus Aufgabe 1 ist ein Teil des Bsp. 1 (zufällige Verteilung).
3. Kontrolliere Deine Berechnung, indem du die Filtermatrix einträgst.
4. Untersuche, wie sich diese Filtermatrix auf die anderen Beispiele auswirkt. Beschreibe, wozu diese Filtermatrix verwendet werden könnte. Untersuche, welchen Effekt es hat, wenn du jeden Wert der Matrix verdoppelst.

Die Matrix verstärkt Änderungen in vertikaler Richtung. Dadurch können horizontale Kanten erkannt werden.

Wenn man die Werte verdoppelt, dann wird der Effekt noch größer.

1/9	1/9	1/9	0	-1	0	0	-1	0
1/9	1/9	1/9	-1	4	-1	-1	5	-1
1/9	1/9	1/9	0	-1	0	0	-1	0

5. Untersuche genauso auch folgende Filtermatrizen:
 Die erste Matrix gleicht die Werte in der Umgebung aneinander an. Die zweite erkennt Unterschiede zwischen benachbarten Werten. Bei gleichen Werten ergibt sich überall 0. Die dritte verstärkt die Unterschiede. Bei gleichen Werten wird das Originalbild erhalten.
6. Die Summe der Einträge der Matrix ist in der Regel 0 oder 1. Beschreibe, welchen Effekt es auf das einfarbige Bild hat, wenn die Summe 0 ist. Beschreibe, was passiert, wenn die Summe 1 ist. Untersuche, was passiert, wenn die Summe zwischen 0 und 1 liegt bzw. größer als 1 ist.
 Wenn die Summe 0 ist, dann wird ein einfarbiges Bild komplett schwarz. Man kann also



bestimmte Bereiche aus einem Bild herausuchen. Bei der Summe 1 bleibt die Gesamthelligkeit erhalten. Bei einer Summe größer als 1 wird sie erhöht, bei Summen kleiner als 1 wird das Bild abgedunkelt.

Faltungsfilter

Glätten/Weichzeichnen

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Kantenfinden

-1	-2	-1	0	-1	0	-1	0	1
0	0	0	-1	4	-1	-2	0	2
1	2	1	0	-1	0	-1	0	1

Schärfen

0	-1	0
-1	5	-1
0	-1	0

Relief

-2	-1	0
-1	1	2
0	1	2

Bildbearbeitungsalgorithmen 4: Mehrpixeloperationen (Faltung)

Ziel: Es soll eine Methode `public Picture faltung(Picture originalbild, double[][] filter)` implementiert werden, die den Filter auf ein Bild anwendet. Durch Aufruf mit geeigneten Filtern können dann die verschiedenen Bildbearbeitungsalgorithmen realisiert werden. Die Größe des Filters muss dabei nicht unbedingt 3x3 sein.

Programmierauftrag:

7. Du erhältst ein Puzzle mit dem Quelltext der Methode. Ordne die Bauteile richtig an.
 Siehe Quelltext in Lösung
8. Kommentiere die einzelnen Abschnitte hinter den `"/` - Kommentarzeichen. Folgende Fragen können dabei helfen. Du solltest auf jeden Fall Antworten auf die Fragen haben.
 - Welches Pixel hat die Koordinaten $(x|y)$? Welches Pixel hat die Koordinaten $(xx|yy)$?
 - Was speichern die Variablen `laenge` und `halb`?
 Hinweis: Teilt man zwei ganze Zahlen durcheinander, werden die Nachkommastellen abgeschnitten: $3/2 = 1$.
 - Was gibt die Position $(i | j)$ an?
 - Warum wird `rot` auf `0.0` gesetzt, wenn es kleiner als `0.0` ist? Warum auf `255.0`, wenn `rot` größer als `255.0` ist?
 - Warum müssen die Variablen `rot`, `gruen` und `blau` als `double` definiert werden?
 - Warum beginnt die Schleife mit der Zählvariable `x` nicht bei `0`, sondern bei `halb`?
 - Warum wird `pixelNeu` mit `original.getPixelArray()` initialisiert?