h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Visual Computing
Digitale Bilder

Hochschule Darmstadt

Prof. Dr. Elke Hergenröther
Björn Frömmer
Prof. Dr. Benjamin Meyer

KAPITEL 12

Bildverarbeitung – Lokale Bildoperationen

Bildverarbeitung – Lokale Bildoperationen

Punktoperatoren im Vergleich zu lokalen Bildoperatoren

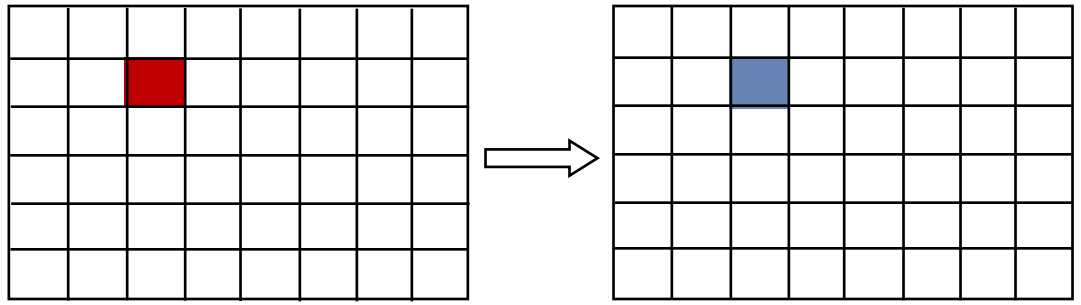
Lokale Bildoperatoren

- Weichzeichner: Mittelwertoperator, Gauß-Filter
- Kantendetektoren: Differenzfilter und Sobel-Operator, Laplace-Operator
- Filter zur Kontrastverbesserung
- Rangfolgeoperatoren: Erosion, Dilation, Median sowie Opening und Closing
- Segmentierungsverfahren

Bildverarbeitung – Lokale Bildoperatoren

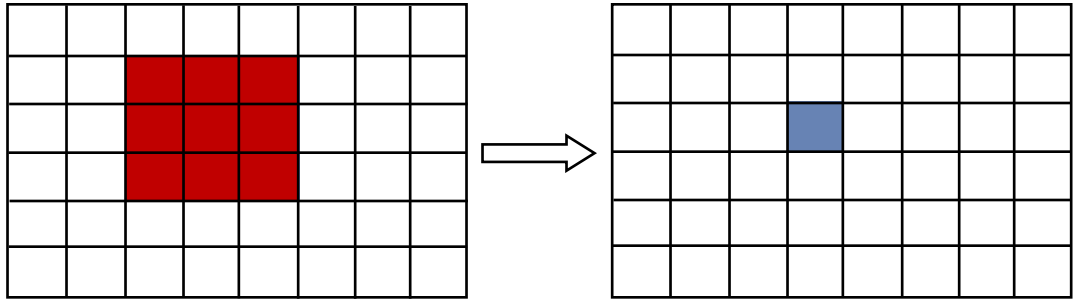
Punktoperatoren im Vergleich zu lokalen Bildoperatoren

Punktoperator: Pixel für Pixel wird transformiert, ohne das die benachbarten Pixel in die Transformation einfließen



Beispiele:
- Helligkeitsänderungen,
- Kontraständerungen,
- Gamma-Korrektur,
- Farbtransformation, …

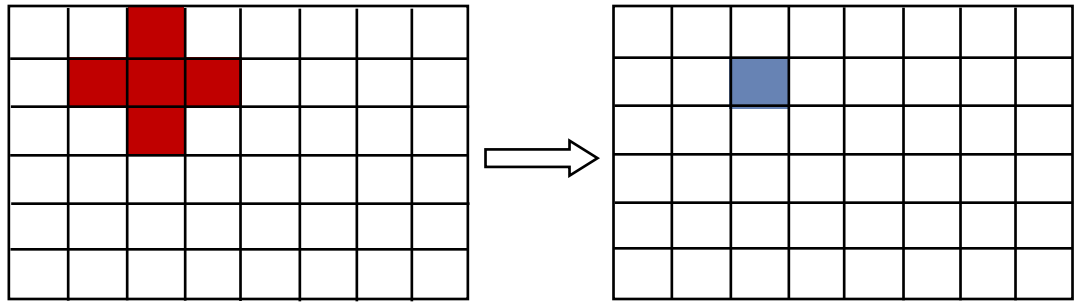
Lokale Bildoperatoren: Jeder Pixel wird in Abhängigkeit zu seinen benachbarten Pixeln transformiert



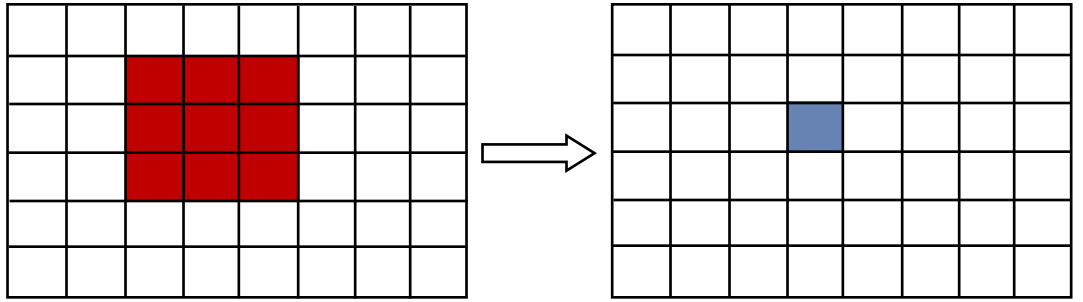
Beispiele:
- Bilder weichzeichnen / verschmieren,
- Kanten detektieren, …

Bildverarbeitung – Lokale Bildoperationen

Lokale Bildoperationen lassen die benachbarten Pixel durch Verwendung unterschiedlicher Kernels in die Berechnung der Pixelformation (Faltung) einfließen.



4er-Nachbarschaft

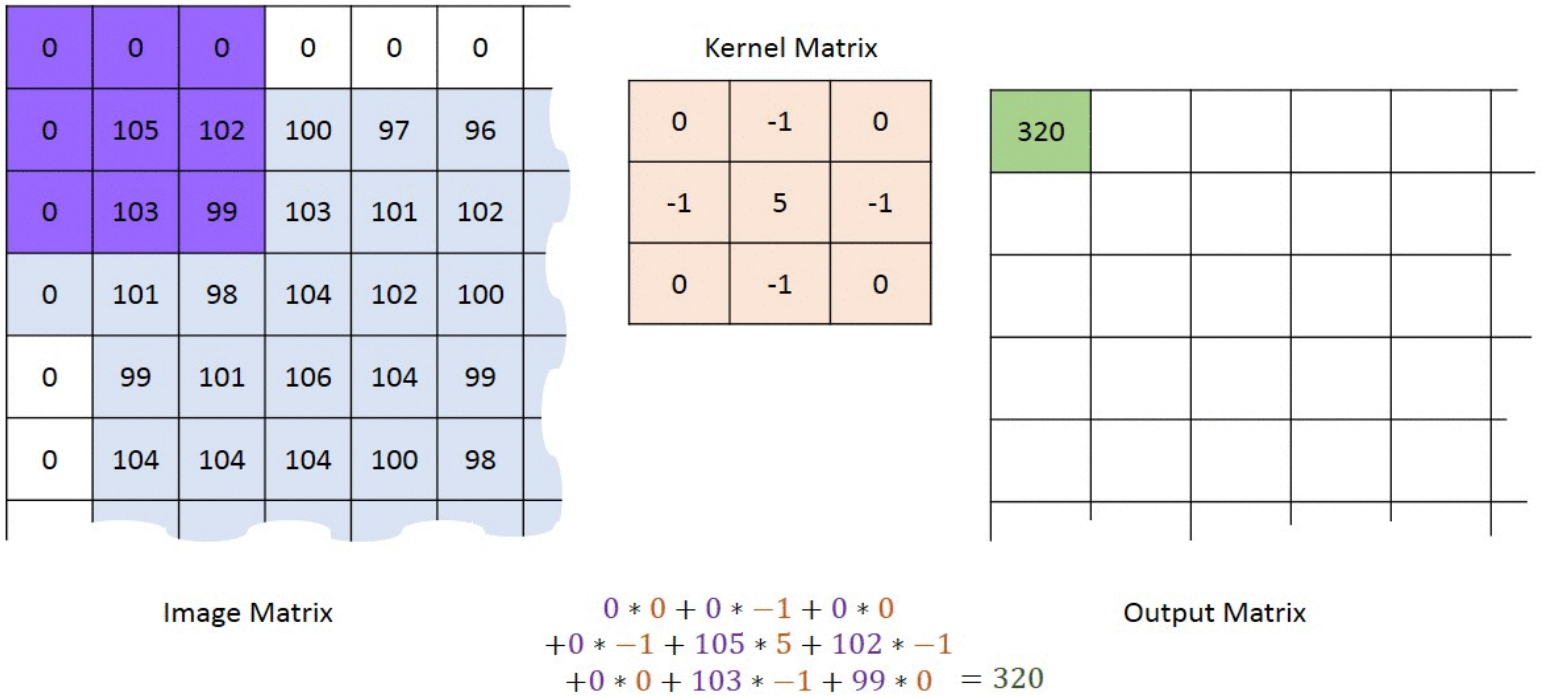


8er-Nachbarschaft

Faltungsmatrix / Kernel

Bildverarbeitung – Lokale Bildoperatoren

Beispiel einer Faltung mit einem lokalen Bildoperator, der die N8-Nachbarschaft nutzt



aus: https://i.stack.imgur.com/9OZKF.gif

Bildverarbeitung – Lokale Bildoperatoren

Beispiel einer Faltung mit einem lokalen Bildoperator, der die N8-Nachbarschaft nutzt

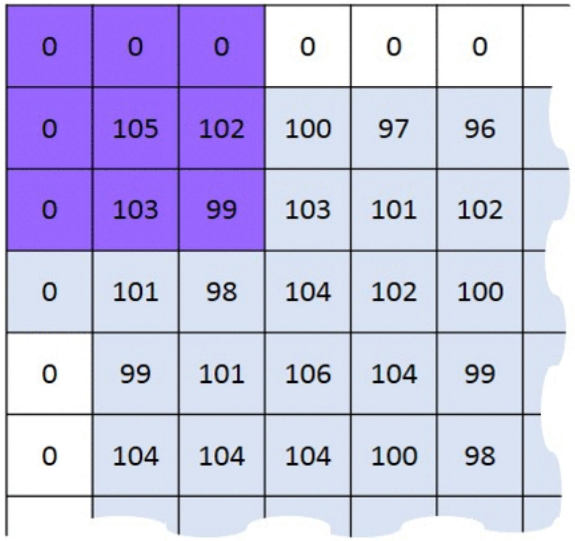
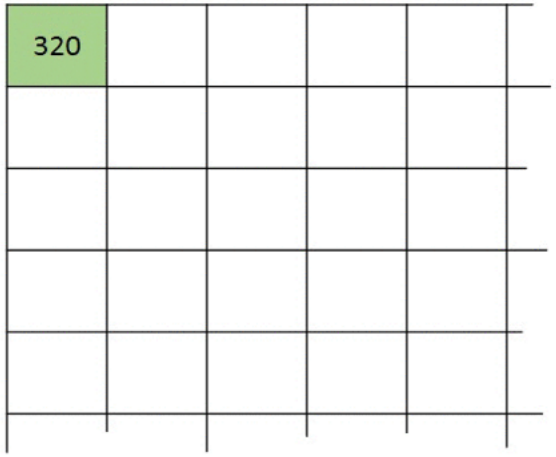


Image Matrix

Kernel Matrix

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



Output Matrix

Allgemeine Formulierung der Faltungsoperation: $e(i,j) = \sum_{l=0}^{2} \sum_{k=0}^{2} \{g(i-1+k, j-1+l) * f(k,l)\}$

aus: https://i.stack.imgur.com/9OZKF.gif

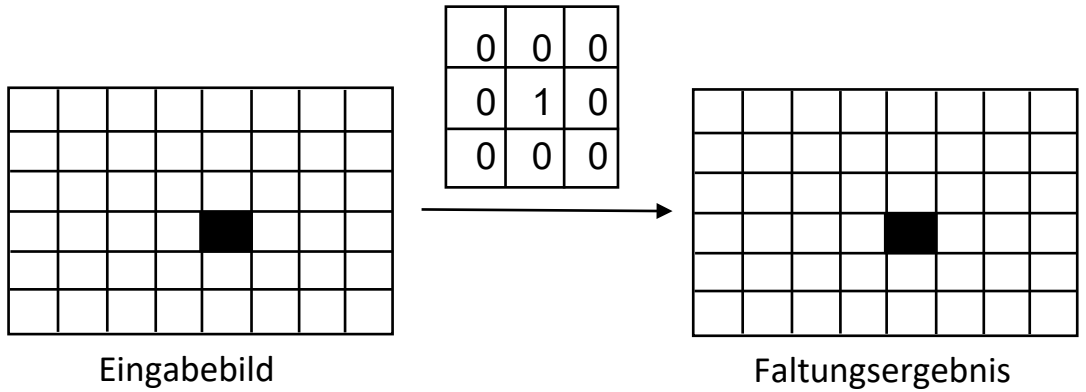
Bildverarbeitung – Lokale Bildoperatoren

Faltung: **Identitätsoperator**

Eingabebild und Faltungsergebnis sind identisch.

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

Kernel / Faltungsmatrix



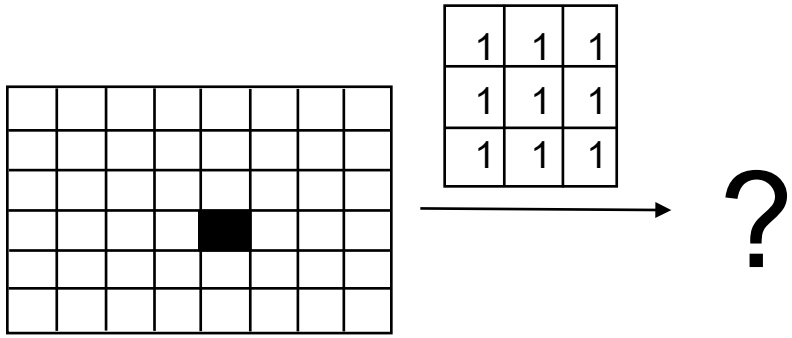
Bildverarbeitung – Lokale Bildoperatoren

Faltung: **Mittelwertoperator**

Bildet den Mittelwert/Durchschnittwert aus den benachbarten Pixelwerten.

$$F = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Kernel oder Faltungsmatrix



Eingabebild ist
8*6 Pixel groß

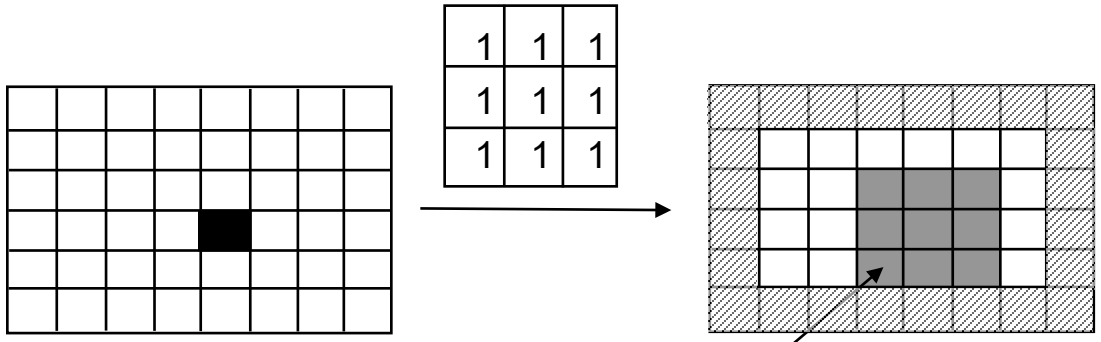
Bildverarbeitung – Lokale Bildoperatoren

Faltung: **Mittelwertsoperator**

Bildet den Mittelwert/Durchschnittwert aus den benachbarten Pixelwerten.

$$F = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Kernel oder Faltungsmatrix



Eingabebild ist
8*6 Pixel groß

g‘(i,j)= 255*8/9 = 227

Bildverarbeitung – Lokale Bildoperatoren

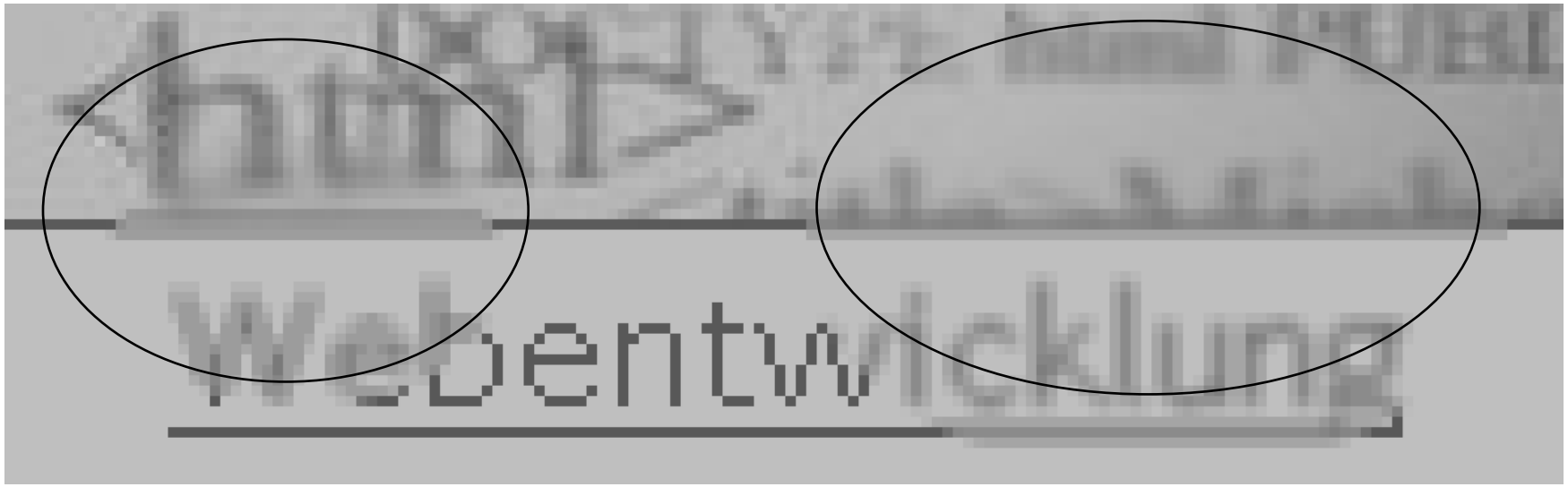
Faltung: **unterschiedliche Weichzeichner – Mittelwertoperator & Gauß-Filter**

$F_{Mittelwert} =$

1	1	1
1	1	1
1	1	1

$F_{Gauß} =$

1	2	1
2	4	2
1	2	1



Bildverarbeitung – Lokale Bildoperatoren

Durch die Berechnung mit dem Mittelwertoperator bzw. dem Gauß-Filter ergeben sich Faltungsergebnissen e(i, j), die außerhalb des Grauwertbereich [0, … 255] liegen.

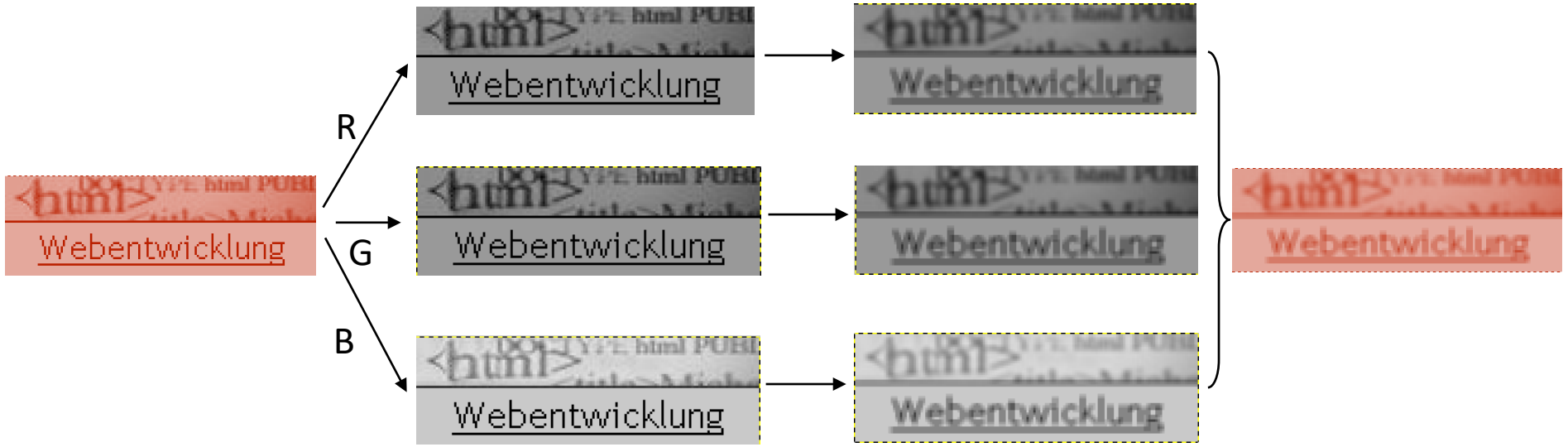
Um die Faltungsergebnisse wieder in den Grauwertbereich zu transformieren, muss eine lineare Abbildung auf den Wertebereich [0, … 255] durchgeführt werden:

- Transformation nach Faltung mit dem Mittelwertoperator: $g'(i, j) = 1/9 \cdot e(i, j) + 0$
- Transformation nach Faltung mit dem Gauß-Filter: $g'(i, j) = 1/16 \cdot e(i, j) + 0$

Bildverarbeitung – Lokale Bildoperatoren

Faltung farbiger Bilder / RGB-Bilder

1. Trennen der unterschiedlichen RGB-Farbkanäle
2. Für jeden Farbkanal: entsprechendes Grauwertbild filtern
3. Gefilterte Bilder der Farbkanäle zum Farbbild kombinieren



Bildverarbeitung – Lokale Bildoperatoren

Welchen Effekt haben die folgenden Filterkerne?

Mittelwertfilter in y-Richtung

$$\frac{1}{3}\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Mittelwertfilter in x-Richtung

$$\frac{1}{3}\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Aus: „**Multimediale Signalverarbeitung Bildverarbeitung: Filter“,** Thorsten Thormählen https://www.mathematik.uni-marburg.de/~thormae/lectures/mmk/mmk_6_1_ger_web.html#1

Bildverarbeitung – Lokale Bildoperatoren

Filterkerne können auch unterschiedliche Größen haben. Die verwendete Kernelgröße ist abhängig von der Auflösung des Eingabebilds und dem gewünschten Effekt.

Bsp. Gauß-Filter:

$$\frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix} \qquad \frac{1}{256}\begin{bmatrix}1 & 4 & 6 & 4 & 1\\ 4 & 16 & 24 & 16 & 4\\ 6 & 24 & 36 & 24 & 6\\ 4 & 16 & 24 & 16 & 4\\ 1 & 4 & 6 & 4 & 1\end{bmatrix} \qquad \dots$$

Aus: „**Multimediale Signalverarbeitung Bildverarbeitung: Filter“,** Thorsten Thormählen https://www.mathematik.uni-marburg.de/~thormae/lectures/mmk/mmk_6_1_ger_web.html#1

Bildverarbeitung – Lokale Bildoperationen

Punktoperatoren im Vergleich zu lokalen Bildoperatoren

Lokale Bildoperatoren

- Weichzeichner: Mittelwertoperator, Gauß-Filter
- Kantendetektoren: Differenzfilter und Sobel-Operator, Laplace-Operator
- Filter zur Kontrastverbesserung
- Rangfolgeoperatoren: Erosion, Dilation, Median sowie Opening und Closing
- Segmentierungsverfahren

Bildverarbeitung – Lokale Bildoperatoren

Unser Wahrnehmungssystem ist auf „Kantenerkennung“ optimiert: **Mach-Band Effekt**

Gleichbleibende Reize der Flächen werden gedämpft und Kontraste werden überzeichnet.

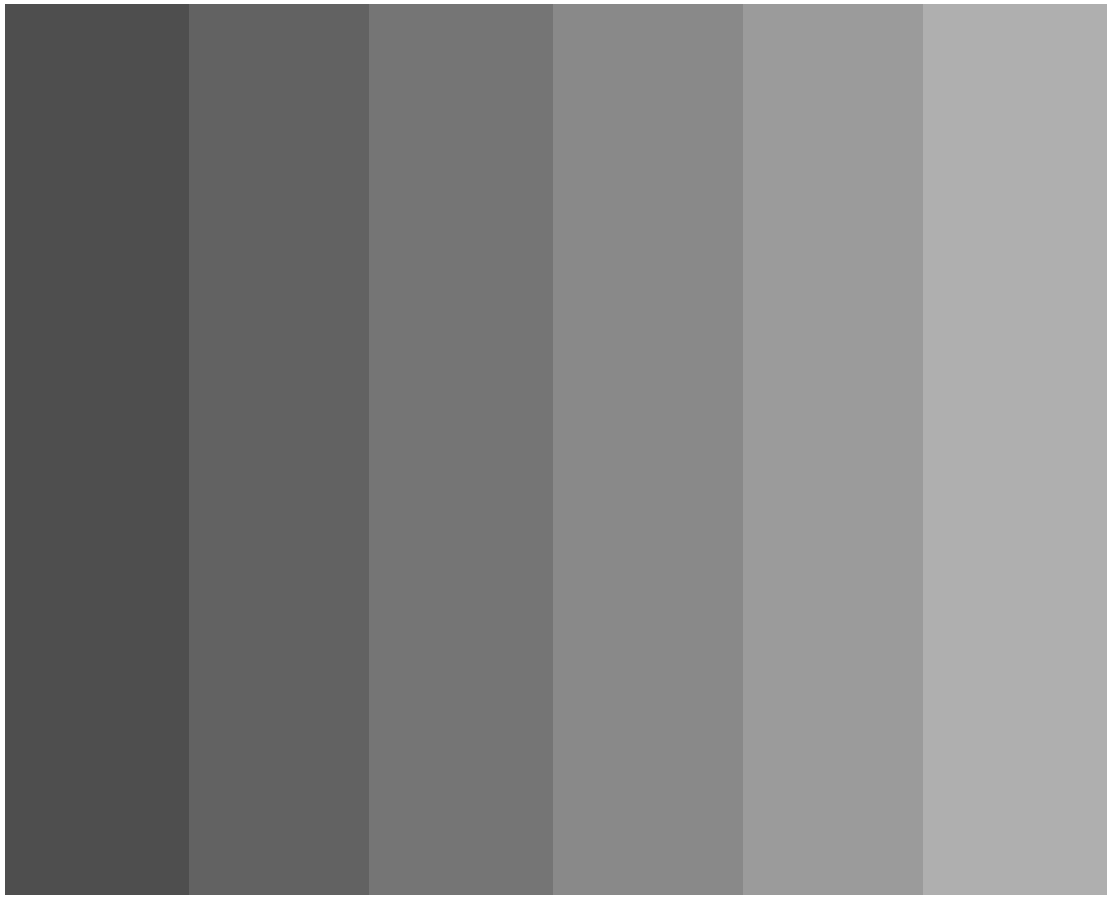


Bild aus:
https://upload.wikimedia.org/wikipedia/commons/9/97/Bandes_de_mach.PNG

Bildverarbeitung – Lokale Bildoperatoren

Unser Wahrnehmungssystem ist auf „Kantenerkennung“ optimiert: **Mach-Band Effekt**

Gleichbleibende Reize der Flächen werden gedämpft und Kontraste werden überzeichnet.

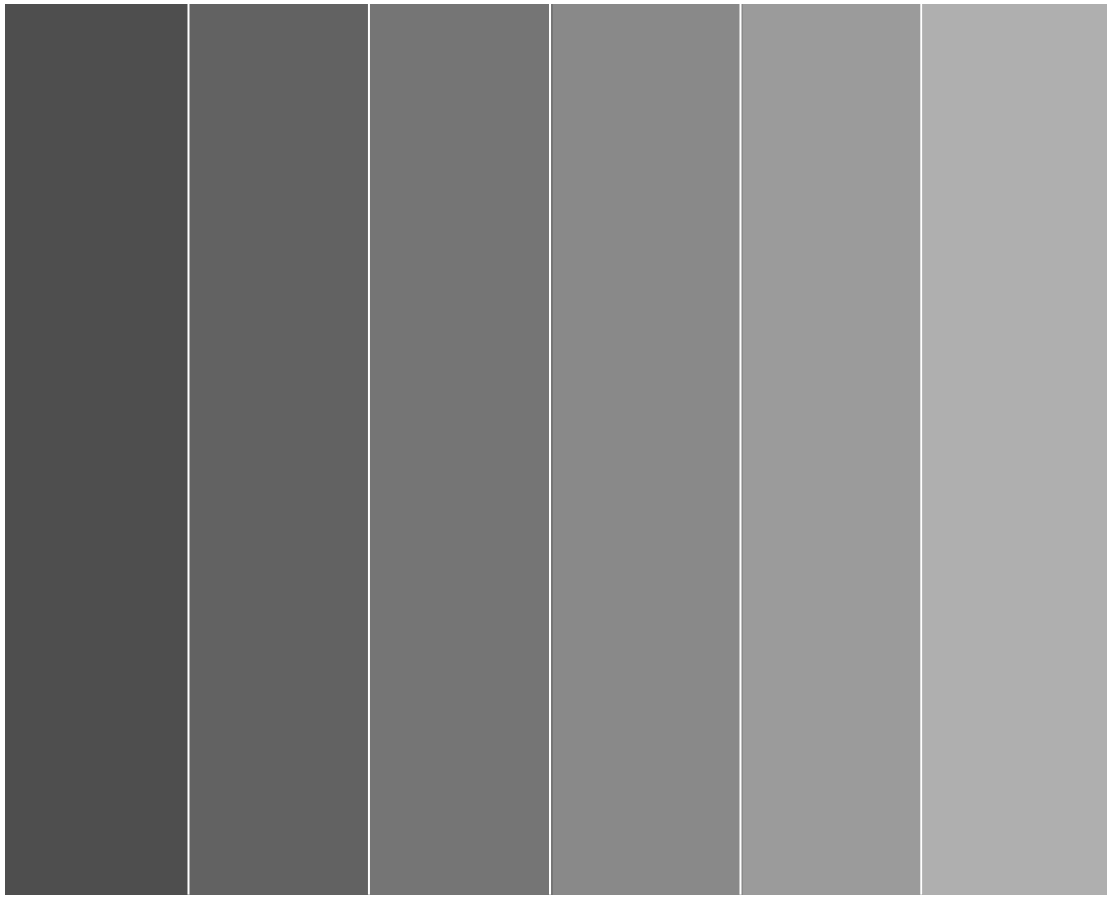


Bild aus:
https://upload.wikimedia.org/wikipedia/commons/9/97/Bandes_de_mach.PNG

Bildverarbeitung – Lokale Bildoperatoren

Unser Wahrnehmungssystem ist auf „Kantenerkennung“ optimiert: **Mach-Band Effekt**

Mach-Band Effekt kann zu Fehlinterpretationen bei einer radiologischen Befundung führen. Hell-Dunkel-Kontraste werden verstärkt und bspw. als Karies interpretiert.

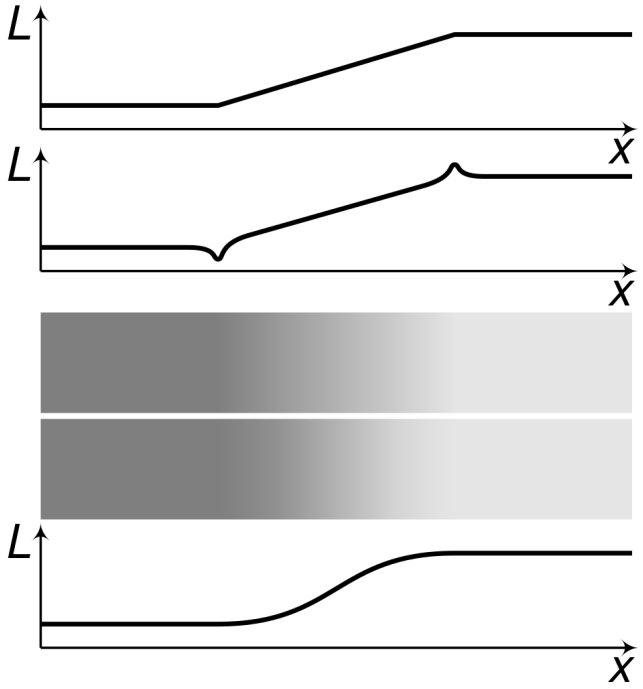
Der Effekt wurde 1865 von Ernst Mach entdeckt.

An dieser Stelle zeichnet sich ein dunkler Fleck ab, der als Karies interpretiert werden könnte.

Bild aus: Radiografische Projektionen der Objekte, Folie 11 https://slideplayer.org/slide/1330107/

Bildverarbeitung – Lokale Bildoperatoren

Mach-Band Effekt: Helligkeitsübergänge werden mit einem höheren Kontrast wahrgenommen, als er tatsächlich vorhanden ist. Die Abbildung beschreibt den Effekt als Funktion der Leuchtdichte L, die der wahrgenommenen Helligkeit ohne Mach-Band Effekt entspricht.



1. Leuchtdichtenprofil von sich abrupt ändernden Grauwerte (siehe A)

2. Wahrgenommenes Helligkeitsprofil von A – unter Einbeziehung des Mach-Band Effekts

A. Grauwertkeil mit sich abrupt ändernden Grauwerten

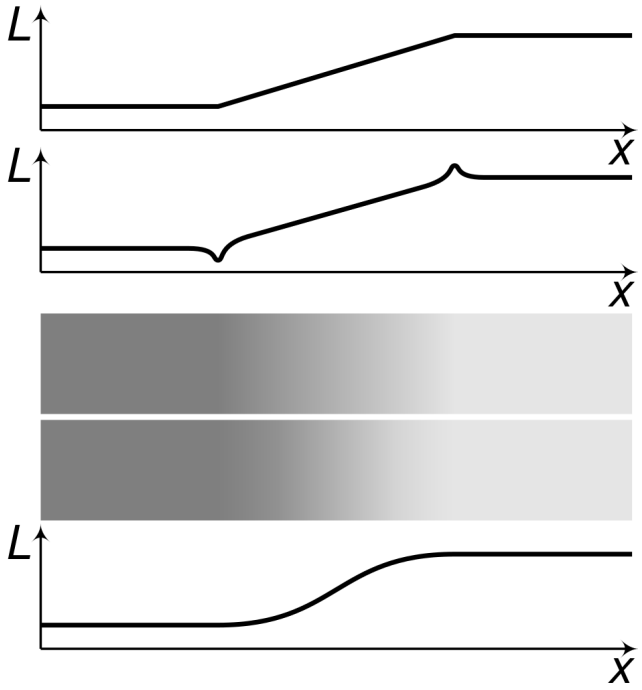
B. Grauwertkeil mit sich kontinuierlich ändernden Grauwerten

3. Leuchtdichtenprofil der sich kontinuierlich ändernden Grauwerte (siehe B)

Bild und Erklärungen aus: https://en.wikipedia.org/wiki/Mach_bands#/media/File:Mach_bands_gradient_overshoot.svg

Bildverarbeitung – Lokale Bildoperatoren

Mach-Band Effekt: Helligkeitsübergänge werden mit einem höheren Kontrast wahrgenommen, als er tatsächlich vorhanden ist. Die Abbildung beschreibt den Effekt als Funktion der Leuchtdichte L, die der wahrgenommene Helligkeit ohne Mach-Band Effekt entspricht.



Dieser Effekt wurde mathematisch übersetzt, um damit einen Faltungs-operator zur Kontrastverstärkung zu entwickeln. Also einen Faltungsoperator, der Kanten so verstärkt, wie wir es von unserer Wahrnehmung gewohnt sind.

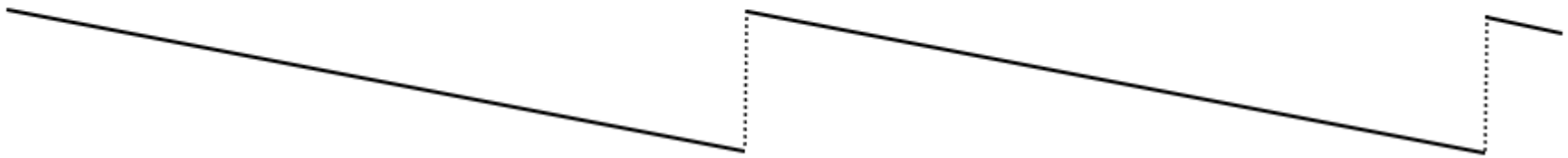
Bild und Erklärungen aus: https://en.wikipedia.org/wiki/Mach_bands#/media/File:Mach_bands_gradient_overshoot.svg

Bildverarbeitung – Lokale Bildoperatoren

Faltung: **Differenzoperator** – erkennt Kanten.

1. Schritt zur Herleitung des Differenzoperators: Statt eines Bildes wird nur eine Bildzeile betrachtet!

Grauwertprofil der Bildzeile

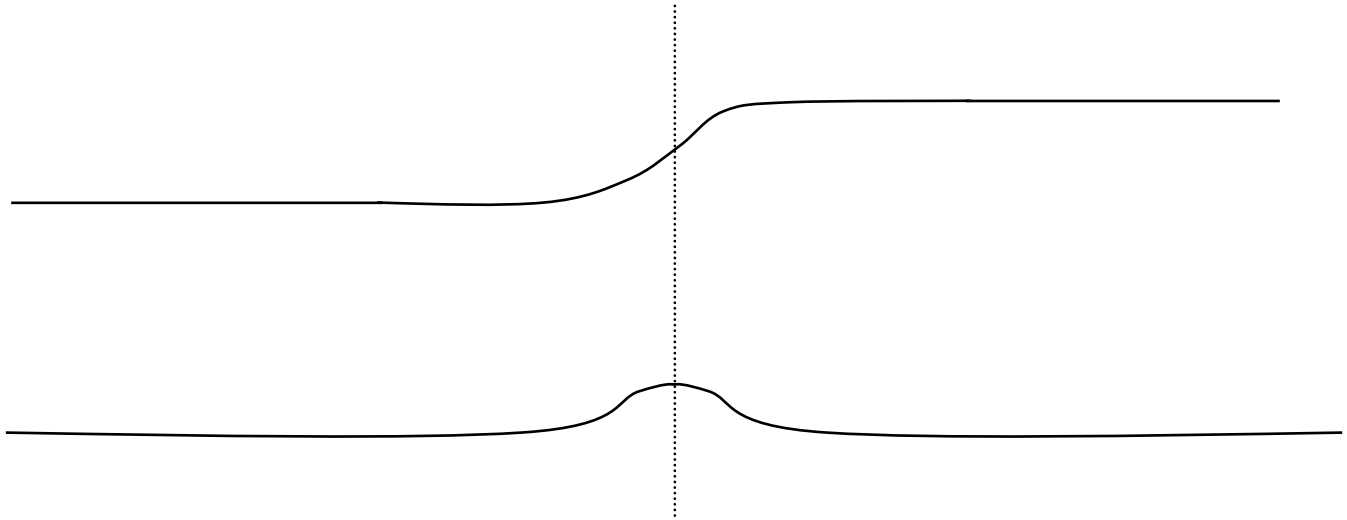


Funktion s(x), die den Grauwertverlauf entlang der Bildzeile darstellt.

Bildverarbeitung – Lokale Bildoperatoren

Faltung: **Differenzoperator** – erkennt Kanten.

2. Schritt zur Herleitung des Differenzoperators: Ableiten der Funktion s(x) , um die Kante zu finden.



Grauwertverlauf einer Bildzeile s(x)

Berechnen der 1. Ableitung s´(x)

In der 1. Ableitung zeigen die Werte ungleich Null die Positionen der Kanten in der Bildzeile an.

Bildverarbeitung – Lokale Bildoperatoren

Faltung: **Differenzoperator** – erkennt Kanten.

2. Schritt zur Herleitung des Differenzoperators: Ableiten der Funktion s(x) , um die Kante zu finden.

Die 1. Ableitung (Steigung) ist definiert durch: $$g'(x) = \lim_{\Delta x \to 0} \frac{g(x+\Delta x) - g(x)}{\Delta x}$$

mit g(x) = Grauwert des Pixel an Position x

Bildverarbeitung – Lokale Bildoperatoren

Faltung: **Differenzoperator – erkennt Kanten.**

2. Schritt zur Herleitung des Differenzoperators: Ableiten der Funktion s(x) , um die Kante zu finden.

Die 1. Ableitung (Steigung) ist definiert durch:

$$g'(x) = \lim_{\Delta x \to 0} \frac{g(x+\Delta x) - g(x)}{\Delta x}$$

mit g(x) = Grauwert des Pixel an Position x

Übertragen auf ein Bild, welches durch die Pixel-Darstellung einen diskreten Raum bildet, bedeutet dies Δx, der Abstand zwischen zwei Pixeln, wird nicht unendlich klein sondern nimmt im minimalen Fall den Wert 1 an.

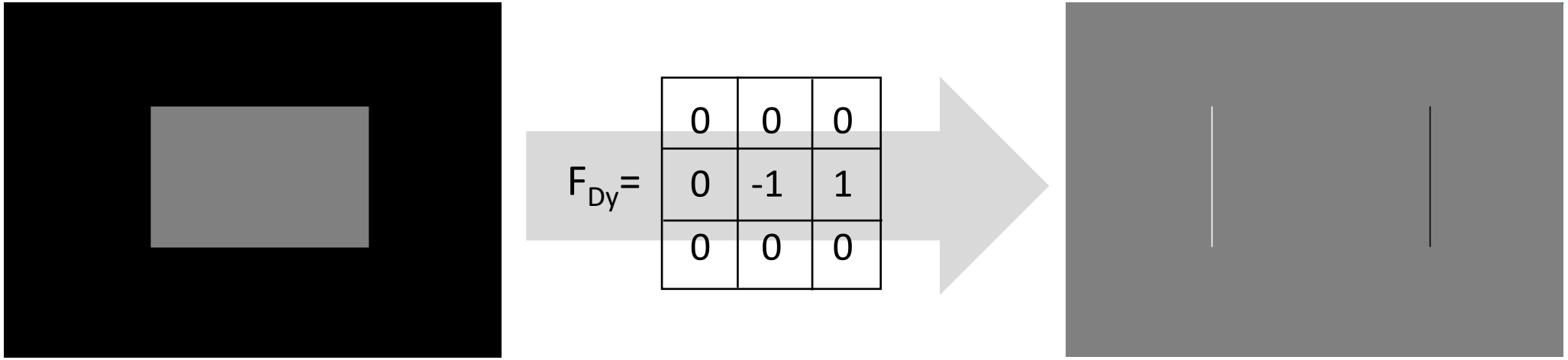
Die 1. Ableitung entspricht also der Differenz zweier benachbarter Grauwerte:

$$\frac{g(x+1) - g(x)}{1} = g(x+1) - g(x)$$

Bildverarbeitung – Lokale Bildoperatoren

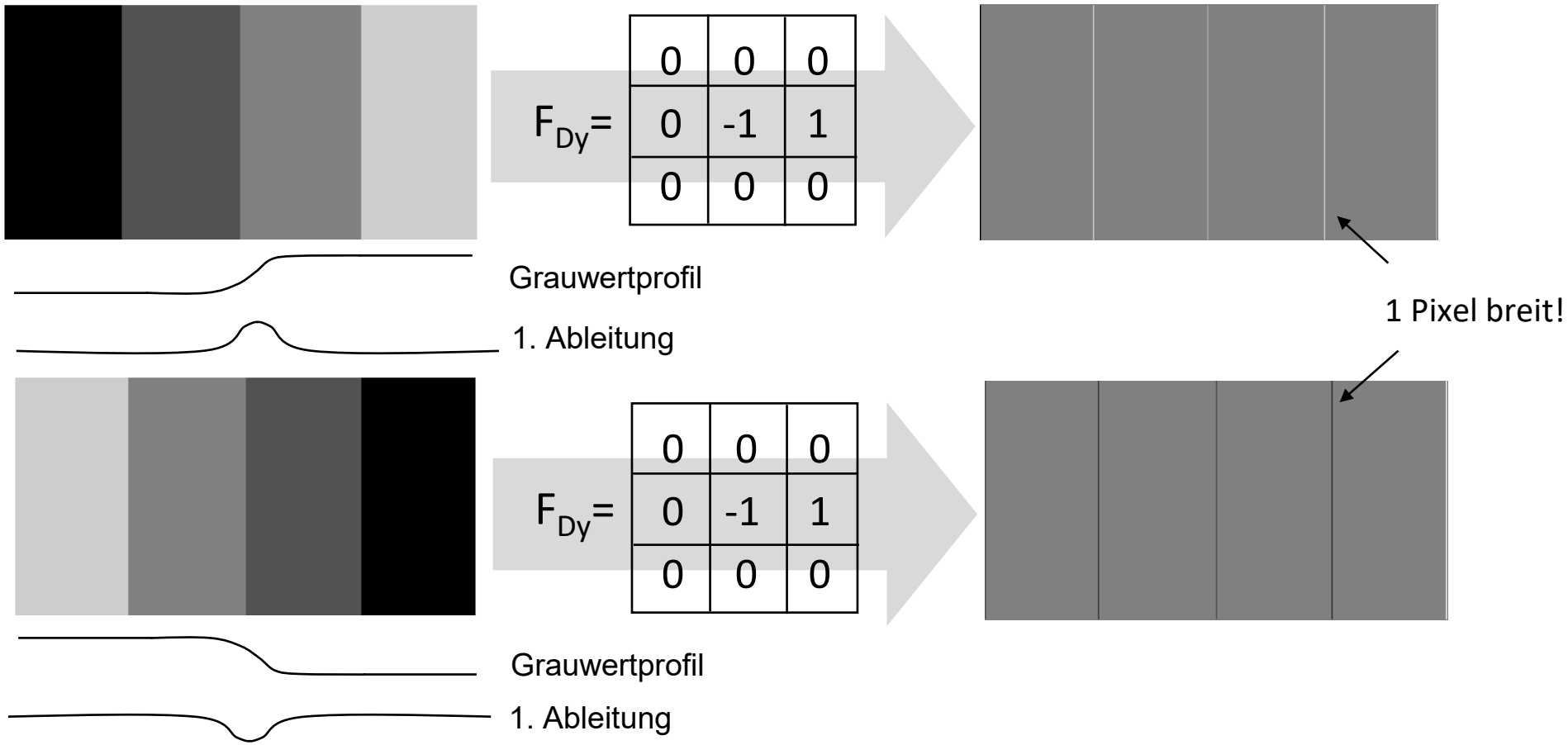
Faltung: **Differenzoperator**

Erkennt Kanten indem er die Differenz der Grauwerte zweier benachbarter Pixel berechnet.



Bildverarbeitung – Lokale Bildoperatoren

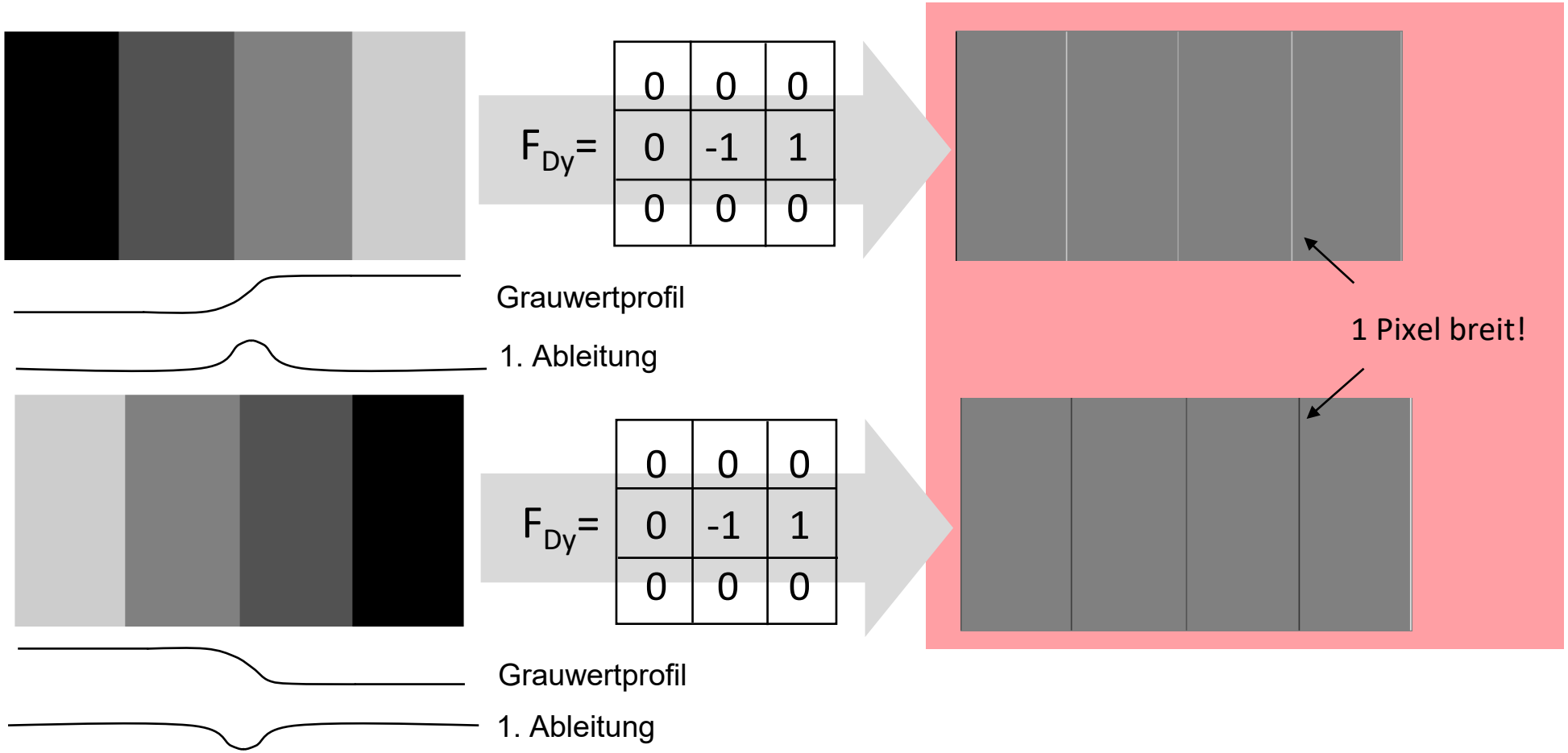
Faltung: **Differenzoperator**



Bildverarbeitung – Lokale Bildoperatoren

Wie entstehen diese Kantenbilder?
Der Differenzoperator bildet auf einen Wertebereich e(i, j) ∈ {-255, … , 255} ab

Faltung: **Differenzoperator**

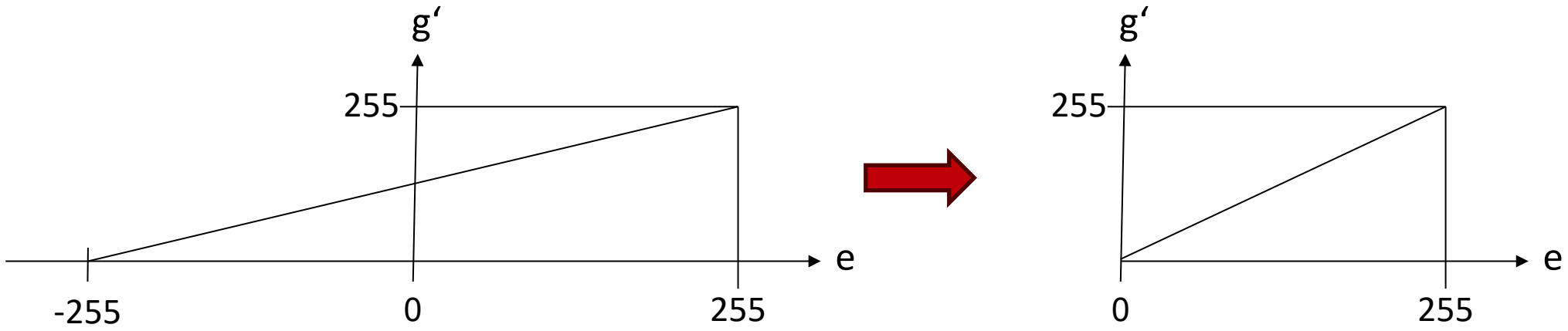


Bildverarbeitung – Lokale Bildoperatoren

Faltung: **Differenzoperator**

Was tun, wenn die Ergebnisse der Faltung teilweise negativ sind?

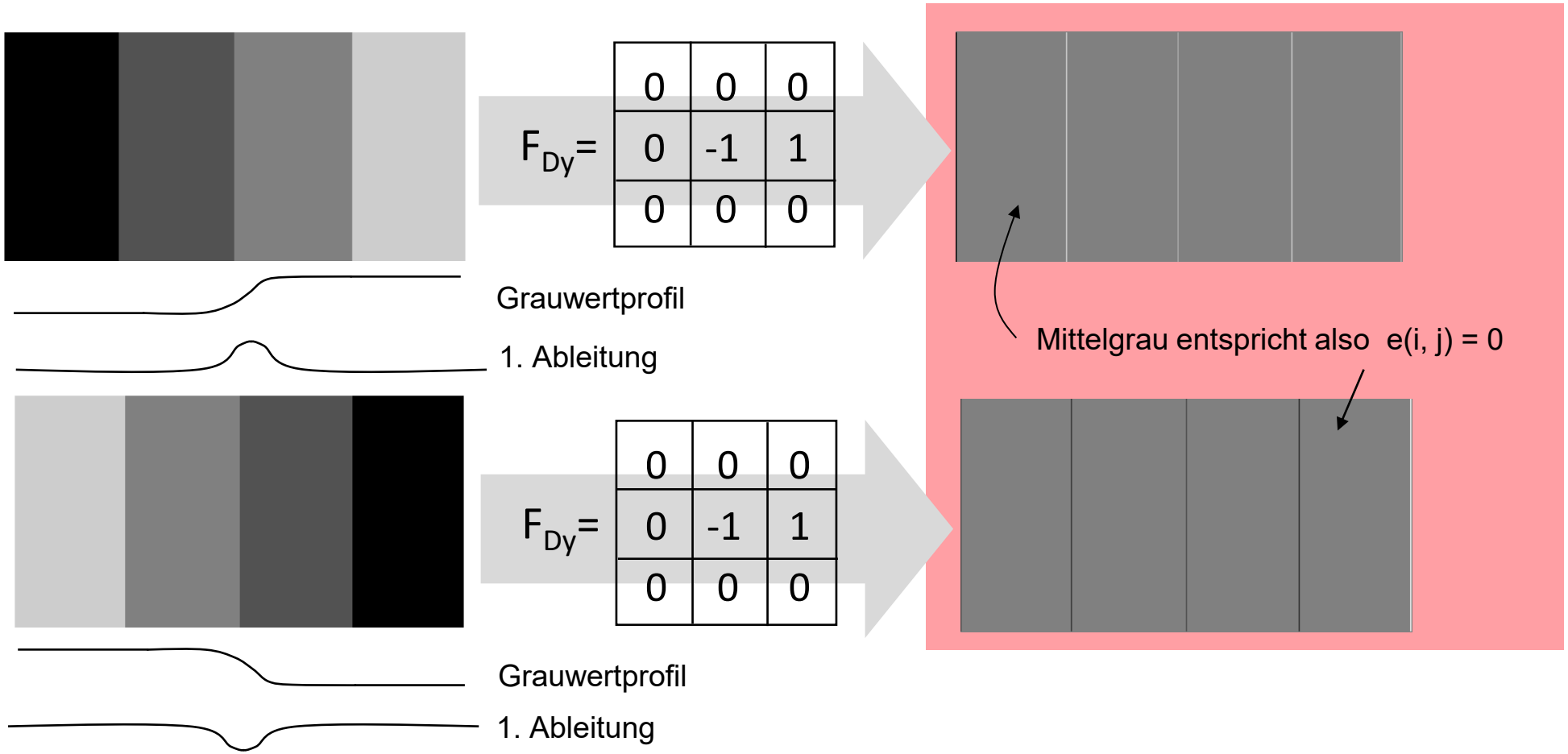
- Nach der Faltung mit dem Differenzoperator erhält man Ergebniswerte e(i, j) ∈ {-255, … , 255}
- Es wird also eine lineare Abbildung benötigt, die den Wertebereich von e(i, j) auf den Bereich g‘(i, j) ∈ {0, … , 255} abbildet
- Diese lineare Abbildungsfunktion lautet: g‘(i,j)= e(i,j*) · 0,5 + 127*



Bildverarbeitung – Lokale Bildoperatoren

Faltung: **Differenzoperator**

Durch die Anwendung der linearen Abbildungsfunktion $g'(i,j) = e(i,j) \cdot 0{,}5 + 127$ werden die Faltungsergebnisse in den darstellbaren Grauwertbereich transformiert.



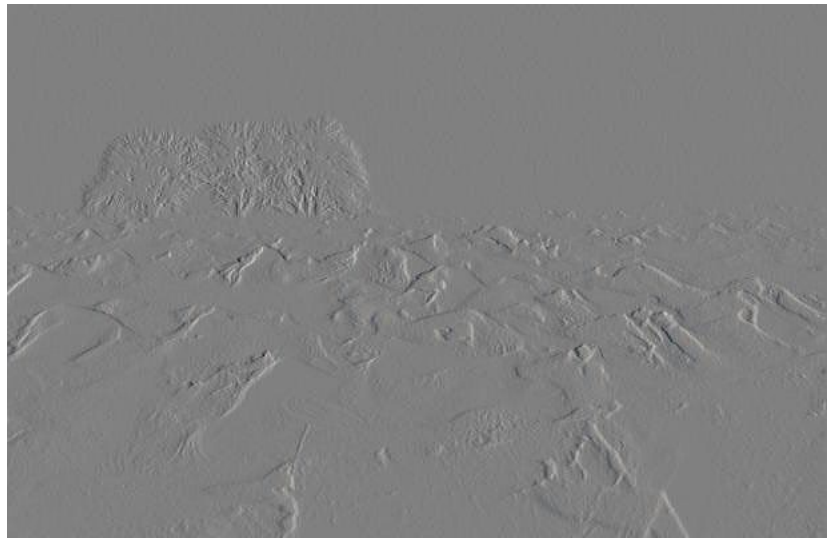
Bildverarbeitung – Lokale Bildoperatoren

Faltung: **Differenzoperator**

Erkennt Kanten indem er die Differenz der Werte zweier benachbarter Pixel berechnet.

F_{Dy} =

0	0	0
0	-1	1
0	0	0



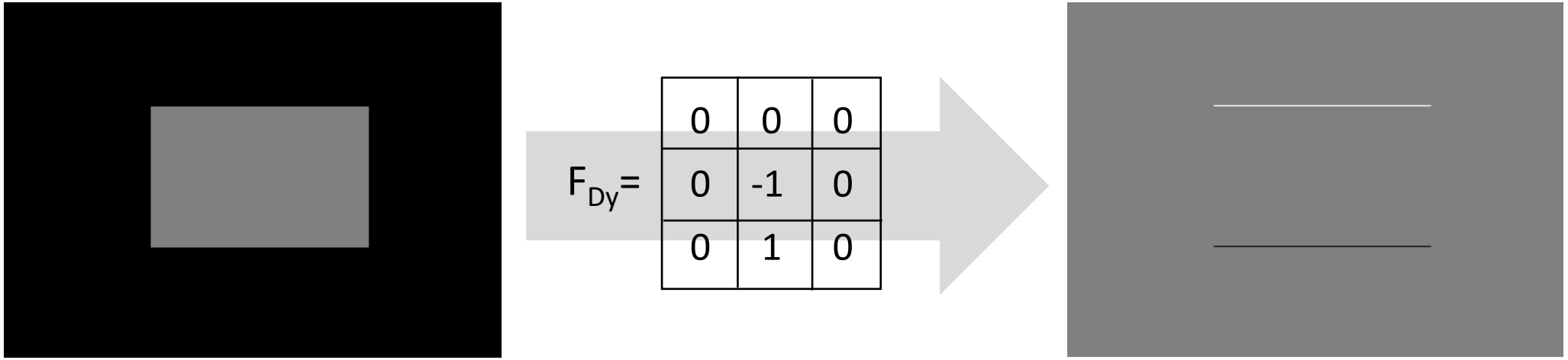
In diesem Beispiel wurden nur die vertikal verlaufenden Kanten angezeigt.

Wie muss ein Differenzoperator aussehen, der die horizontal verlaufenden Kanten findet?

Bildverarbeitung – Lokale Bildoperatoren

Faltung: **Differenzoperator**

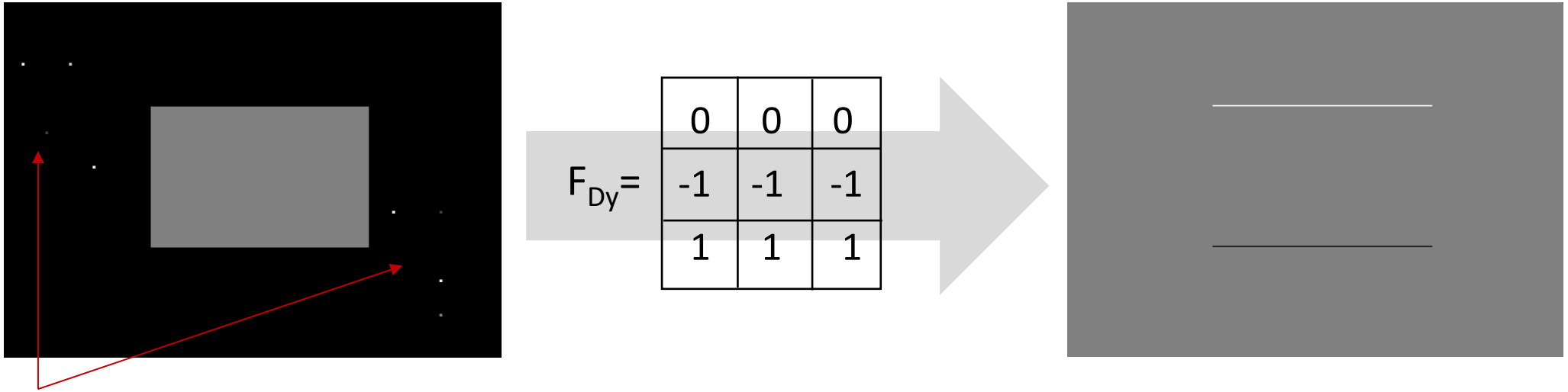
Erkennt die horizontal verlaufenden Kanten indem er die Differenz der Werte zweier benachbarter Pixel in einer Bildspalte berechnet.



Bildverarbeitung – Lokale Bildoperatoren

Faltung: **Differenzoperator**

Um die Kantenerkennung robust zu machen und nicht bei jedem „fehlerhaften“ Pixel eine Kante anzuzeigen, werden die benachbarten Pixel in die Berechnung mit einbezogen.



Fehlerhafte Pixel machen sich durch Rauschen im Bild bemerkbar.

Bildverarbeitung – Lokale Bildoperatoren

Faltung: **Differenzoperator**

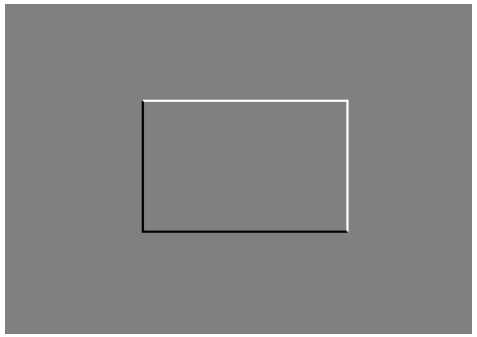
Um die Kantenerkennung noch robuster zu machen wird eine „Puffer“-Zeile oder –Spalte in den Operator eingefügt. Diese Art von Differenzoperatoren heißen Sobel-Operator.

Faltungsergebnisse der unterschiedlichen Sobel-Operatoren

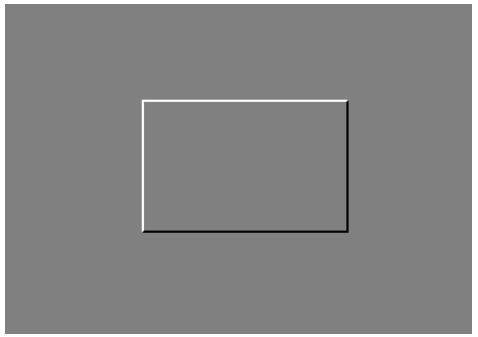
-1	0	1
-1	0	1
-1	0	1

-1	-1	-1
0	0	0
1	1	1

0	-1	-1
1	0	-1
1	1	0



-1	-1	0
-1	0	1
0	1	1



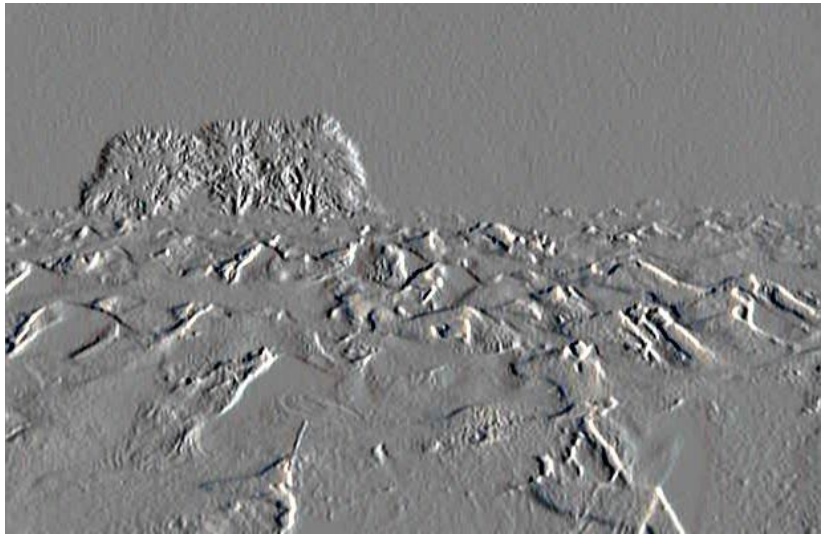
Bildverarbeitung – Lokale Bildoperatoren

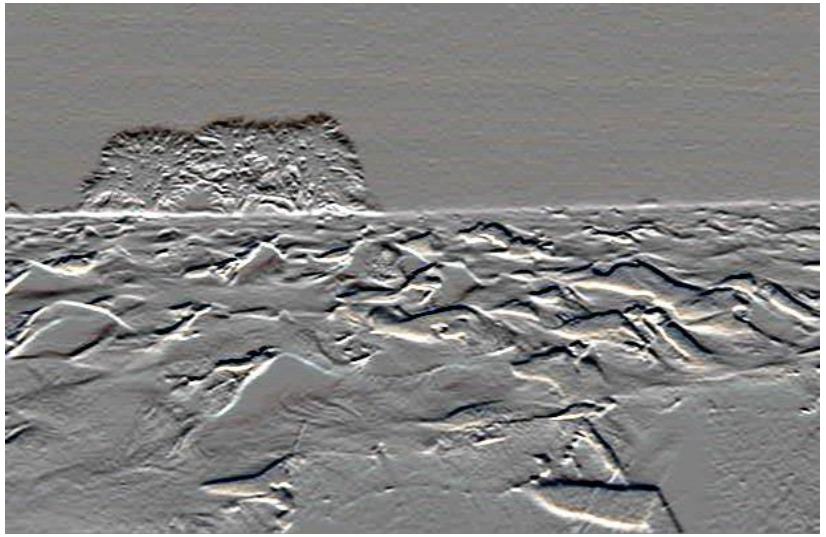
Faltung: **Differenzoperator**

Um die Kantenerkennung noch robuster zu machen wird eine „Puffer“-Zeile oder –Spalte in den Operator eingefügt. Diese Art von Differenzoperatoren heißen Sobel-Operator.

Durch die „Puffer“-Spalte bzw. –Zeile markiert der Sobel-Operator die Kanten mit mehreren Pixeln.

-1	0	1
-1	0	1
-1	0	1





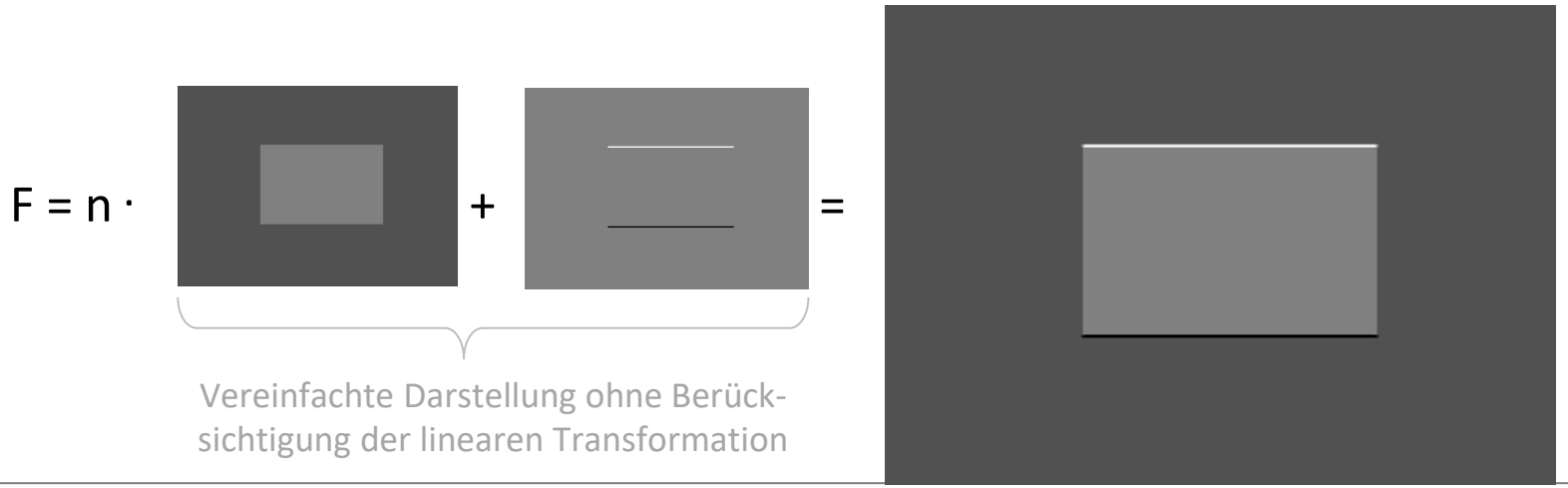
-1	-1	-1
0	0	0
1	1	1

Bildverarbeitung – Lokale Bildoperatoren

Faltung: Ergebnis des **Differenzoperators** ins Eingabebild einfügen:

Nach Anwendung eines Kantenoperators sind im Ergebnisbild nur Kanten sichtbar. Durch die Addition des Originalbildes erhält man einen Filter F, der Kanten im Eingabebild erzeugt.

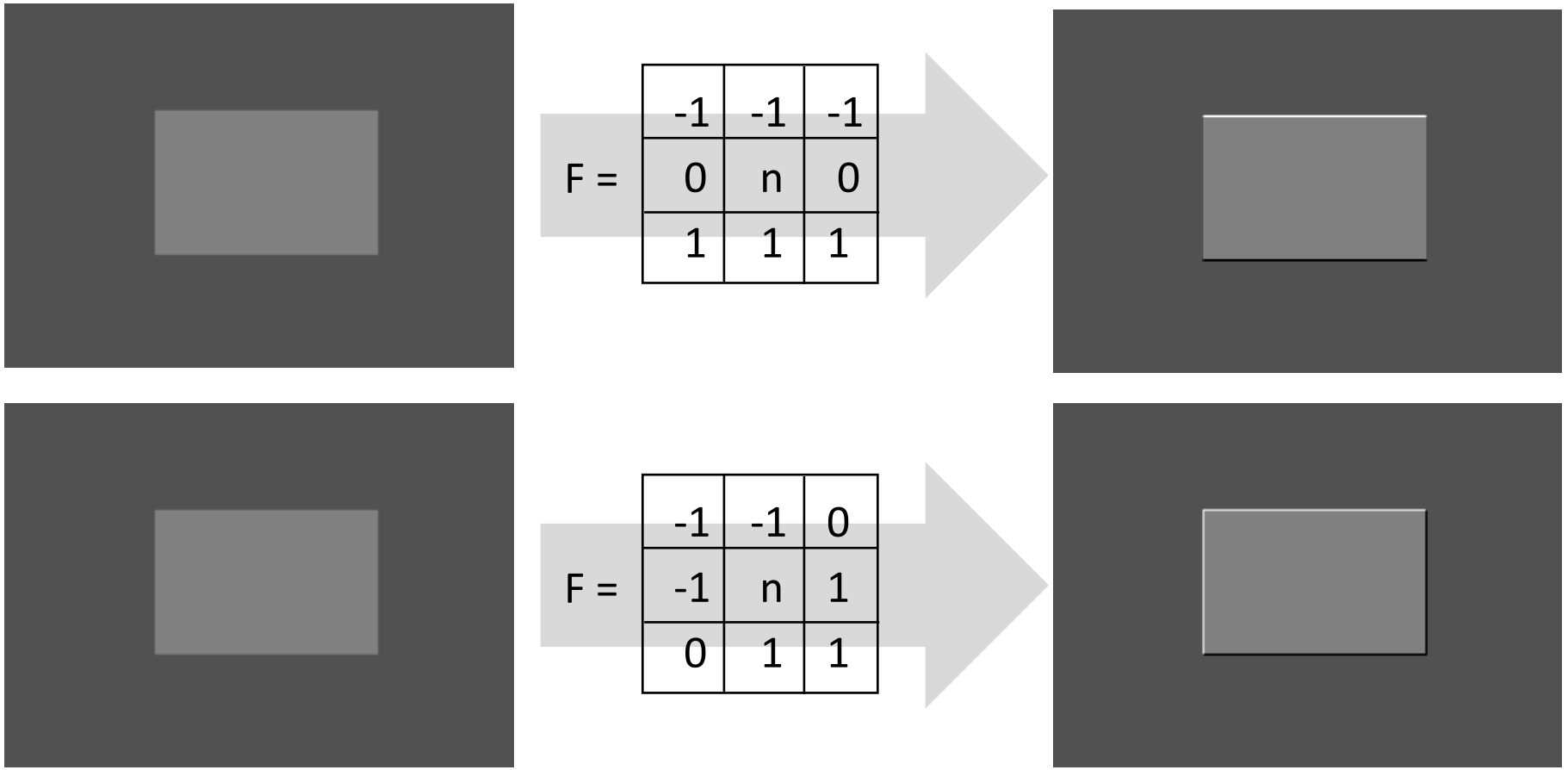
$$F = n \cdot \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & n & 0 \\ 1 & 1 & 1 \end{bmatrix}$$



Vereinfachte Darstellung ohne Berücksichtigung der linearen Transformation

Bildverarbeitung – Lokale Bildoperatoren

Faltung: Ergebnis des **Differenzoperators** ins Eingabebild einfügen:



Bildverarbeitung – Lokale Bildoperatoren

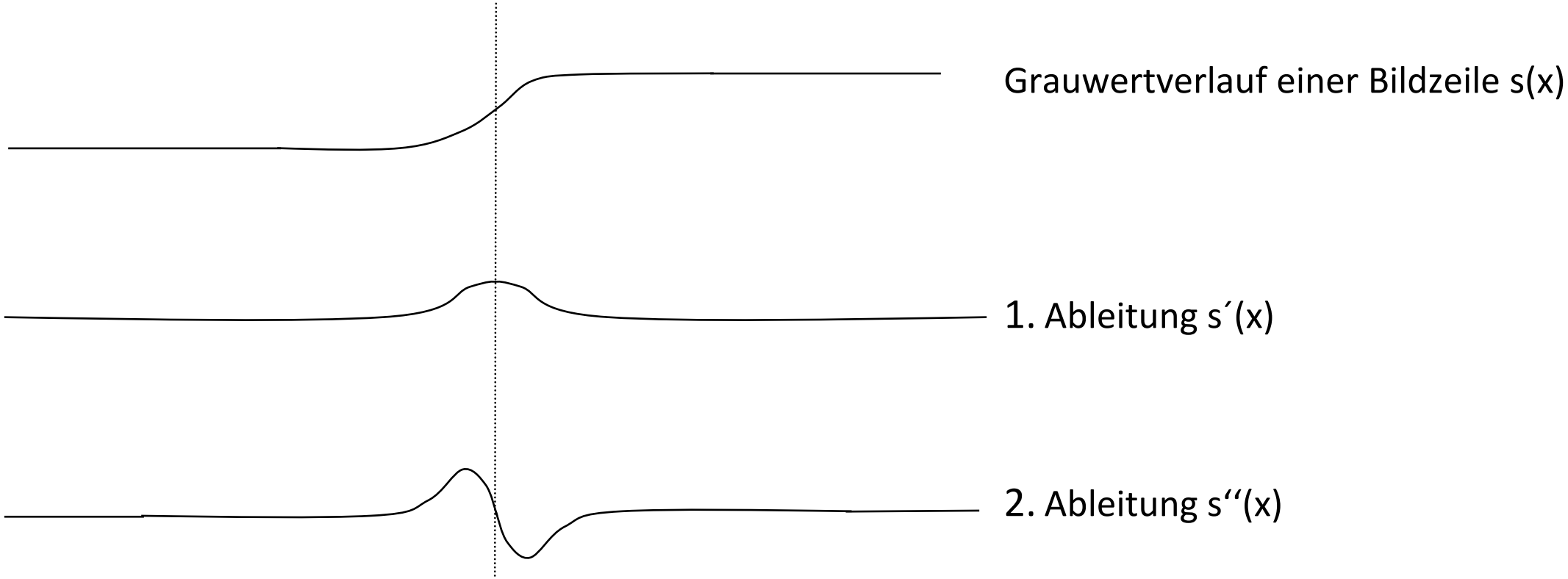
Anwendung des Filters F mit unterschiedlichem n:

F =

-1	-1	0
-1	n	1
0	1	1

Bildverarbeitung – Lokale Bildoperatoren

Faltung: **Laplace-Operator** – erkennt Kanten anhand der 2. Ableitung.



Bildverarbeitung – Lokale Bildoperatoren

Faltung: **Laplace-Operator** – erkennt Kanten anhand der 2. Ableitung.

Die 2. Ableitung (Steigung) ist definiert durch: $$g''(x) = \lim_{\Delta x \to 0} \frac{(g(x+\Delta x) - g(x)) - (g(x) - g(x-\Delta x))}{\Delta x}$$

mit $g(x)$ = Grauwert des Pixel an Position x

Für Rasterbilder gilt: Δx nimmt im minimalen Fall den Wert 1 an.

Die 2. Ableitung in einer Bildzeile oder Spalte: $$g''(x) = g(x+1) - 2 \cdot g(x) + g(x-1)$$

Bildverarbeitung – Lokale Bildoperatoren

Faltung: **Laplace-Operator** – erkennt Kanten anhand der 2. Ableitung.

Die 2. Ableitung in einer Bildzeile oder Spalte: $g''(x) = g(x+1) - 2 \cdot g(x) + g(x-1)$

Laplace-Operator =

0	0	0
1	-2	1
0	0	0

Kantendetektion in der Bildzeile

+

0	1	0
0	-2	0
0	1	0

Kantendetektion in der Bildspalte

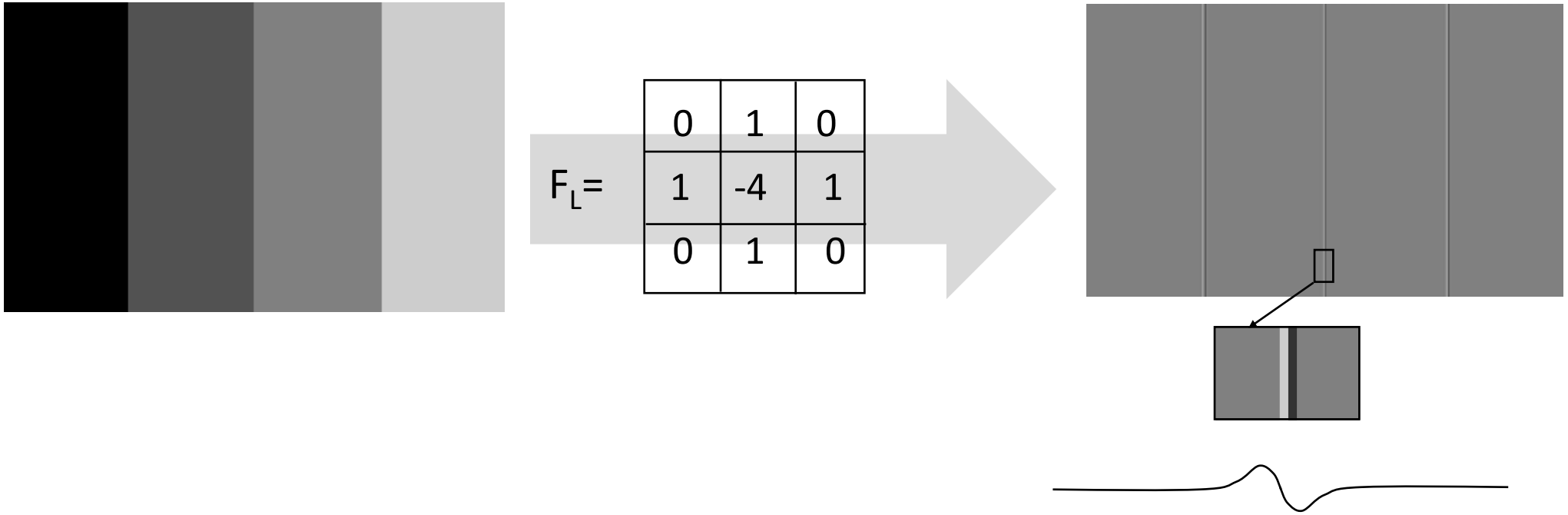
=

0	1	0
1	-4	1
0	1	0

Kantendetektion sowohl in der Bildspalte als auch Bildzeile

Bildverarbeitung – Lokale Bildoperatoren

Faltung: **Laplace-Operator** – erkennt Kanten anhand der 2. Ableitung.

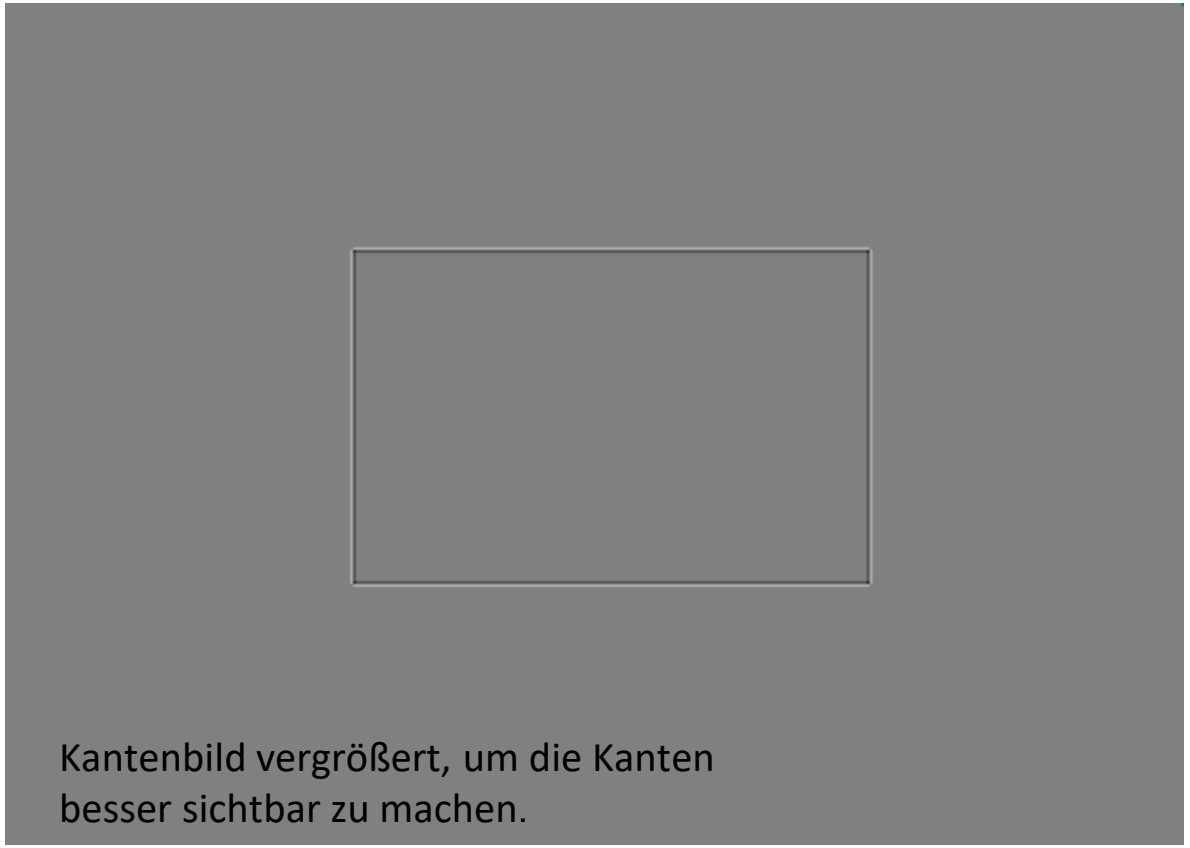


Bildverarbeitung – Lokale Bildoperatoren

Faltung: **Laplace-Operator** – erkennt Kanten anhand der 2. Ableitung.

$F_L =$

0	1	0
1	-4	1
0	1	0



Kantenbild vergrößert, um die Kanten besser sichtbar zu machen.

Bildverarbeitung – Lokale Bildoperatoren

Faltung: **Laplace-Operator** – erkennt Kanten anhand der 2. Ableitung.

$F_L =$

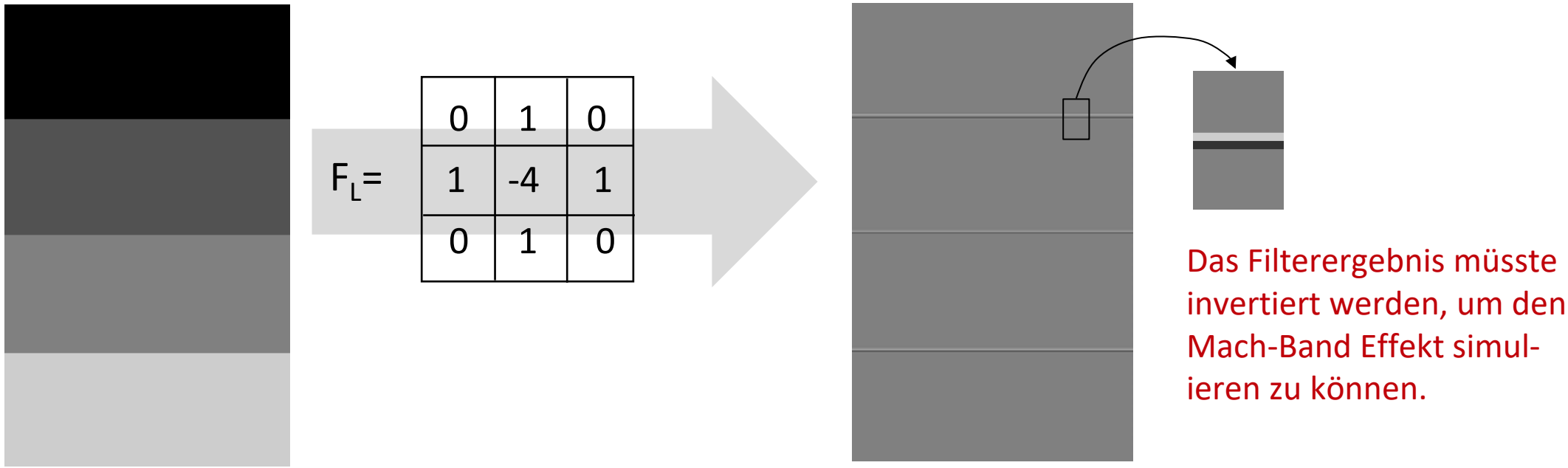
0	1	0
1	-4	1
0	1	0

Kantenbild vergrößert, um die Kanten besser sichtbar zu machen.

Bildverarbeitung – Lokale Bildoperatoren

Faltung: vom **Laplace-Operator** zum **Kontrastverbesserungsfilter**

Ein Kontrastverbesserungsfilter betont die Kante auf der hellen Seite der Kante durch hellere und auf der dunklen Seite der Kante durch dunklere Pixel – entspricht dem Mach-Band Effekt.

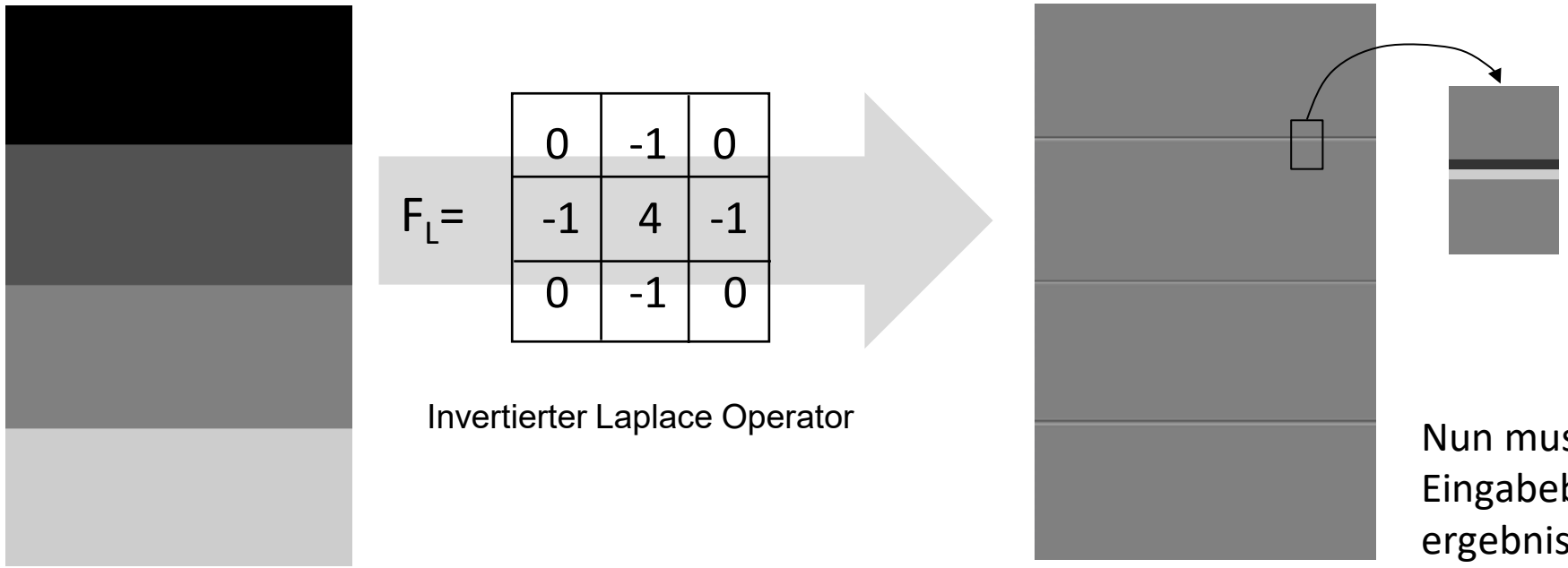


Das Filterergebnis müsste invertiert werden, um den Mach-Band Effekt simulieren zu können.

Bildverarbeitung – Lokale Bildoperatoren

Faltung: vom **Laplace-Operator** zum **Kontrastverbesserungsfilter**

Ein Kontrastverbesserungsfilter betont die Kante auf der hellen Seite der Kante durch hellere und auf der dunklen Seite der Kante durch dunklere Pixel – entspricht dem Mach-Band Effekt.



Nun muss nur noch das Eingabebild ins Faltungs-ergebnis integriert werden.

Bildverarbeitung – Lokale Bildoperatoren

Faltung: vom **Laplace-Operator** zum **Kontrastverbesserungsfilter**

Ein Kontrastverbesserungsfilter betont die Kante auf der hellen Seite der Kante durch hellere und auf der dunklen Seite der Kante durch dunklere Pixel – entspricht dem Mach-Band Effekt.

Der Trick zur Integration des Eingabebildes ins Faltungsergebnis ist schon bekannt:

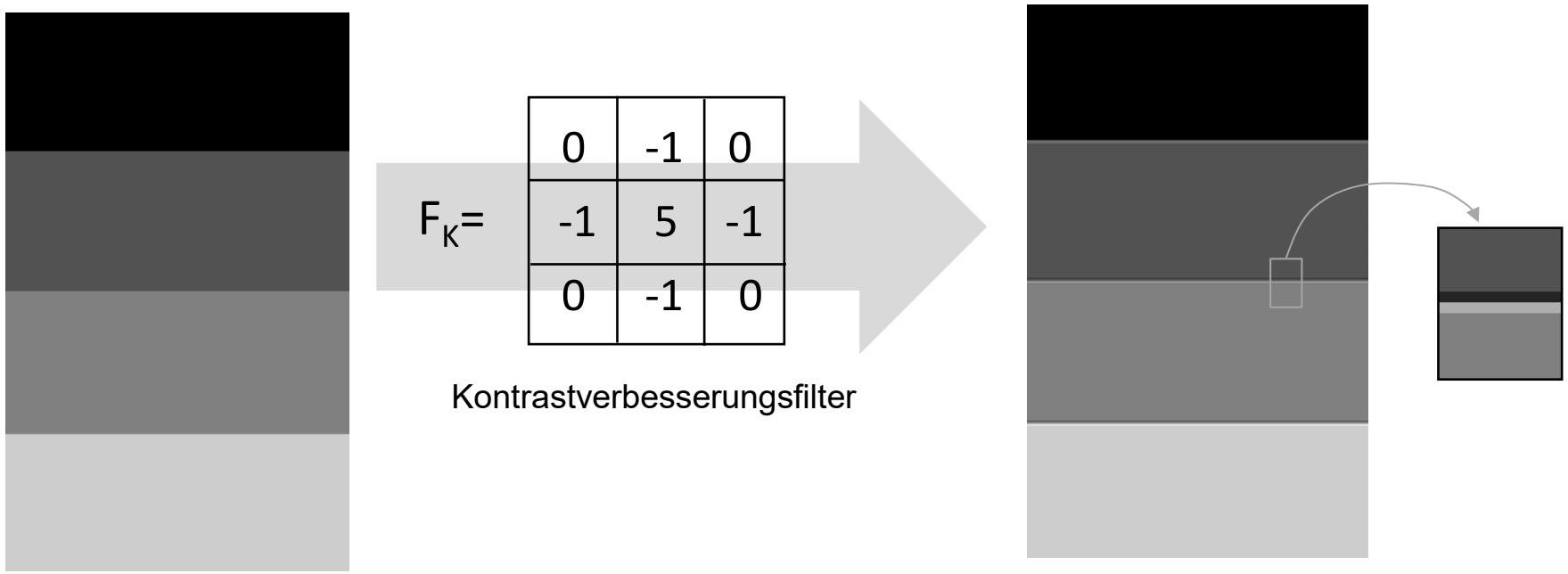
$$F_K = n \cdot \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & n+4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Kontrastverbesserungsfilter

Bildverarbeitung – Lokale Bildoperatoren

Faltung: vom **Laplace-Operator** zum **Kontrastverbesserungsfilter**

Ein Kontrastverbesserungsfilter betont die Kante auf der hellen Seite der Kante durch hellere und auf der dunklen Seite der Kante durch dunklere Pixel – entspricht dem Mach-Band Effekt.



Bildverarbeitung – Lokale Bildoperatoren

Faltung: vom **Laplace-Operator** zum **Kontrastverbesserungsfilter**

Ein Kontrastverbesserungsfilter betont die Kante auf der hellen Seite der Kante durch hellere und auf der dunklen Seite der Kante durch dunklere Pixel – entspricht dem Mach-Band Effekt.

$F_K =$

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

Kontrastverbesserungsfilter

Bildverarbeitung – Lokale Bildoperationen

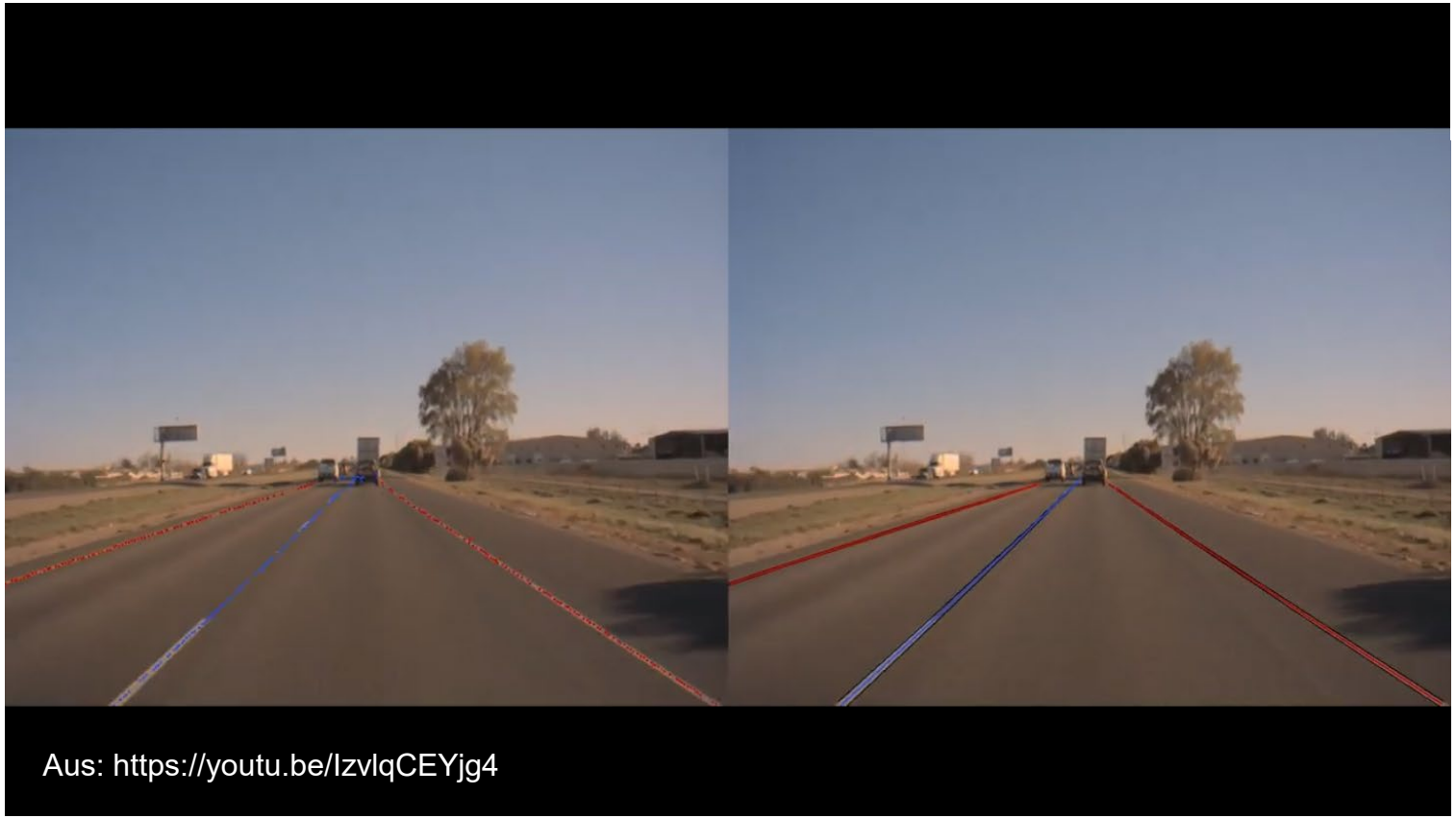
Punktoperatoren im Vergleich zu lokalen Bildoperatoren

Lokale Bildoperatoren

- Weichzeichner: Mittelwertoperator, Gauß-Filter
- Kantendetektoren: Differenzfilter und Sobel-Operator, Laplace-Operator
- Filter zur Kontrastverbesserung – bereits mit den Kantendetektoren besprochen
- Rangfolgeoperatoren: Erosion, Dilation, Median sowie Opening und Closing
- Segmentierungsverfahren

Bildverarbeitung – Lokale Bildoperationen

Wie werden die Filter eingesetzt? Beispiel Fahrspurerkennung.



Aus: https://youtu.be/IzvlqCEYjg4

Bitte auf den schwarzen Rand klicken, um das Video zu starten.

Bildverarbeitung – Lokale Bildoperationen

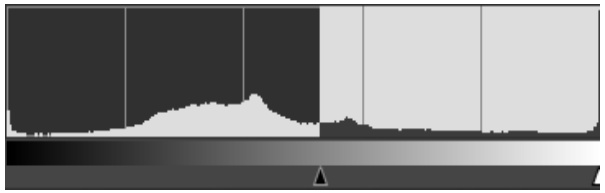
Wie werden die Filter eingesetzt? Beispiel Fahrspurerkennung.

- Ziel ist die im JPEG-Bild (links) eingezeichneten roten Fahrbahnmarkierungen zu finden
- Die bekannten Filter werden typischerweise zur Vorverarbeitung des Bilds oder Videos eingesetzt.

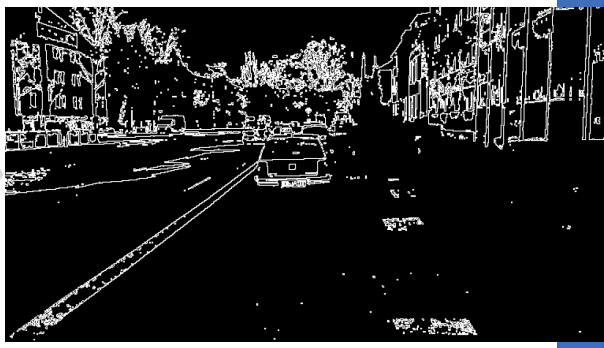
Mit JPEG komprimiertes Bild

1. Umwandlung in ein Grauwertbild durch die Extraktion des Helligkeitskanals

2. Biniarisierung



Im Beispielbild liegt der optimale Schwellwert bei ca. 135



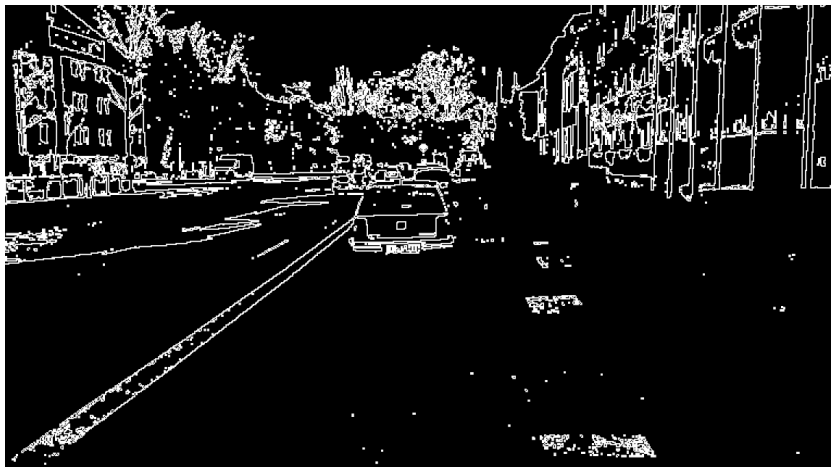
3. Kantenextraktion
   Hier wurde der Sobel-Operator verwendet

Bild aus https://davidgruenewald.de/category/darmstadt/

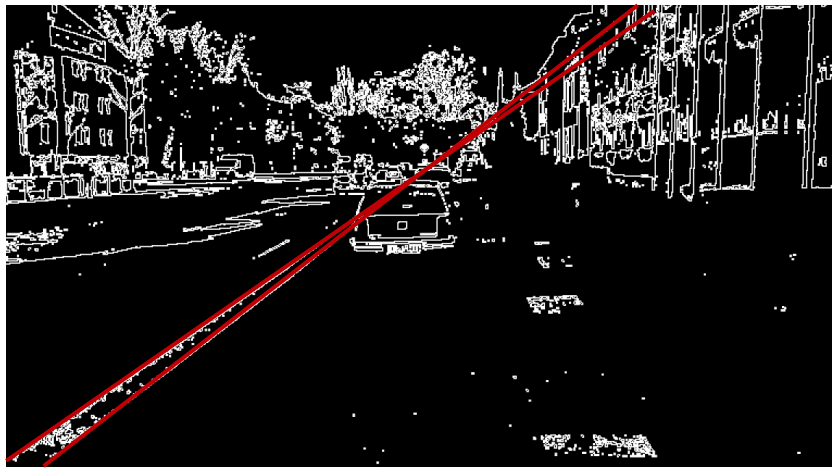
Bildverarbeitung – Lokale Bildoperationen

Wie werden die Filter eingesetzt? Beispiel Fahrspurerkennung.

- Im nächsten Schritt wird den Pixeln eine inhaltliche Bedeutung zugeordnet.
- Im Beispiel für die Fahrspurerkennung interessieren nur die Kanten.
- Genauer: die Linien, die nicht senkrecht verlaufen und die die meisten Pixel in sich vereinen.
- für diese Aufgabe eignet sich die **Hough Transformation**



Eingabe: **Kantenbild**



Ergebnis: **extrahierte Geraden, die nicht senkrecht verlaufen und die meisten Pixel in sich vereinen**

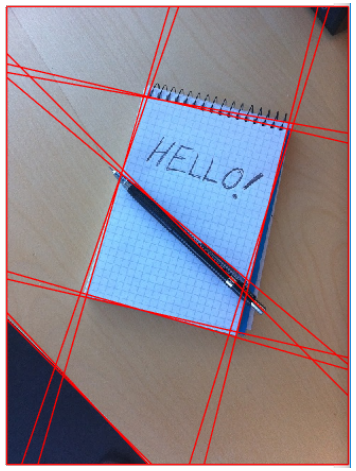
Originalbild aus https://davidgruenewald.de/category/darmstadt/

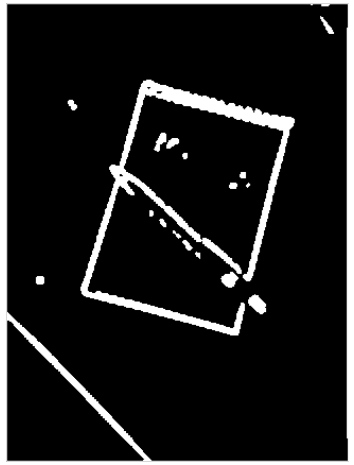
Bildverarbeitung – Lokale Bildoperationen

Wie werden die Filter eingesetzt? Beispiel Fahrspurerkennung.

Die Hough Transformation sucht nach Geraden in einem Kantenbild.

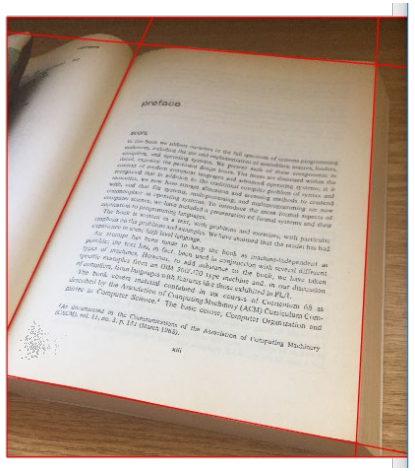
A.) Ergebnis der Hough Transformation auf Basis eines Kantenbilds, dessen Kanten mehrere Pixel breit sind.

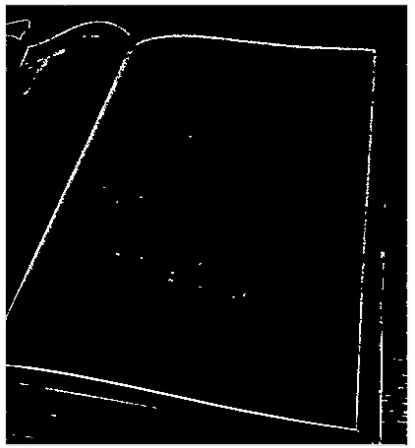




Es werden pro Kante zwei Geraden erkannt, die paarweise ungefähr gleich viele Pixel in sich vereinen.

B.) Ergebnis der Hough Transformation auf Basis eines Kantenbilds, dessen Kanten nur ein Pixel breit sind.





Es wird jeweils nur eine Geraden erkannt.

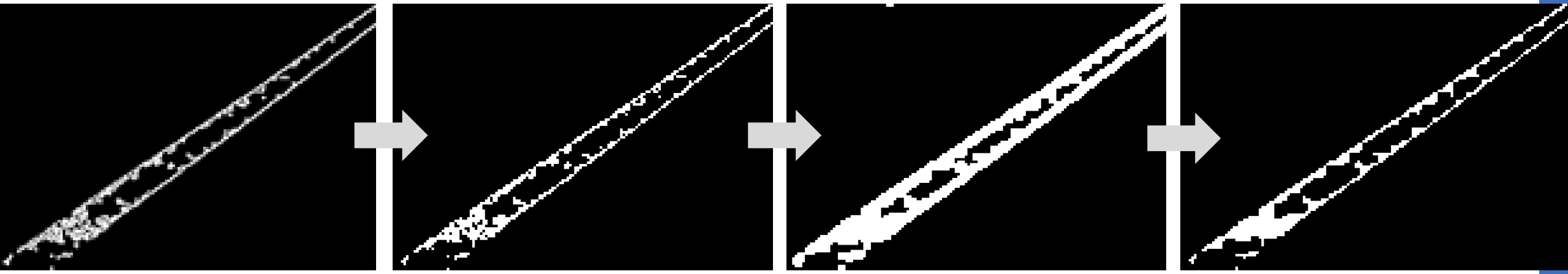
Fazit: In der Vorverarbeitung muss darauf geachtet werden, dass die Kante nur ein Pixel breit ist.

Bilder aus: https://stackoverflow.com/questions/41474719/find-corners-of-a-page-after-applying-hough-transformation

Bildverarbeitung – Lokale Bildoperationen

Wie werden die Filter eingesetzt? Beispiel Fahrspurerkennung.

Wie können Kanten ausgedünnt werden? Beispielsweise durch Rangfolgeoperatoren.



Kantenbild

Biniarisierung des Kantenbilds (optional, nicht unbedingt nötig)

Dilatation = Erweiterung der Vordergrundobjekte. Die weißen Pixel dehnen sich um eine Schicht aus.

Erosion = Die weißen Pixelbereiche schrumpfen um eine Pixelschicht.

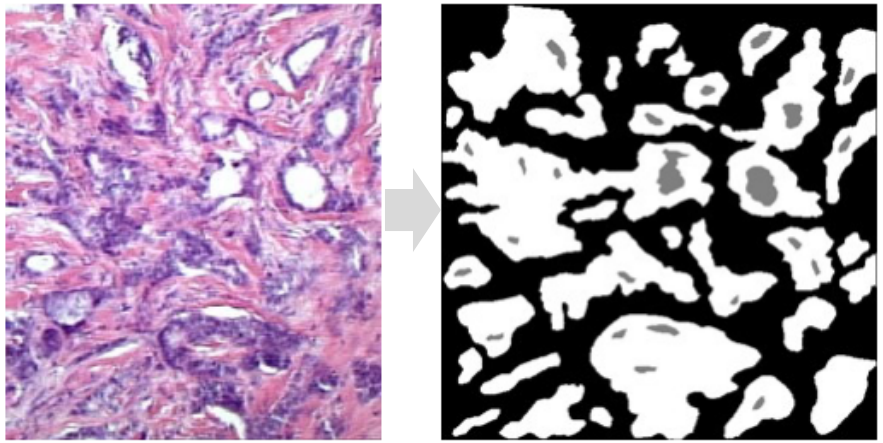
In der Praxis wird der Canny Edge Detector genutzt. Er dünnt nahezu alle Kanten auf eine ein pixelbreite Dicke aus. **Die Rangfolgeoperatoren benutzt man oft zur Vorbereitung der Segmentierung**.

Bildverarbeitung – Lokale Bildoperationen

Begriffsdefinition: Segmentierung

Die Segmentierung ist die inhaltliche Interpretation eines Bildes auf Pixelebene [1]. Die Bildsegmentierung erkennt anhand der Eigenschaften der Pixel, welche Pixel zu einem Objekt gehören. Farben bzw. Grauwerte und die Nachbarschaft der Pixel zueinander sind Beispiele für Eigenschaften, die einen Pixel beschreiben. Die Segmentierung fasst also Pixel mit homogenen Eigenschaften zu einer Gruppe zusammen [2].

Medizinisches Beispielbild „Brustkrebsgewebe“ [2]



Originalbild segmentiertes Bild

Segmentierung einer Luftaufnahme, aus [2]

Originalbild segmentiertes Bild

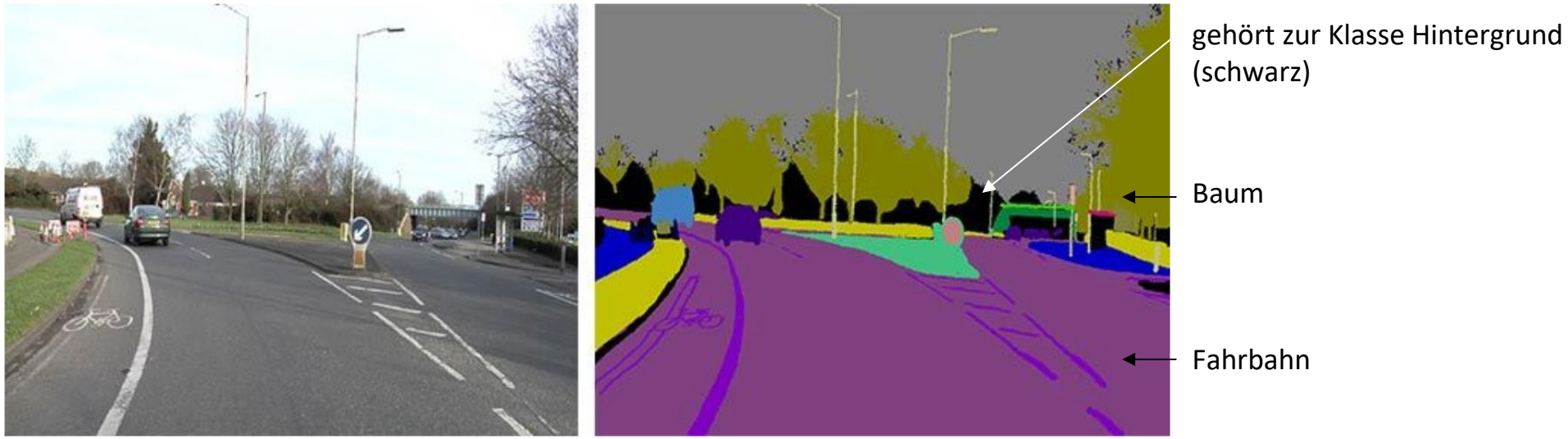
[1] https://de.mathworks.com/solutions/image-video-processing/semantic-segmentation.html
[2] https://www.mathematik.uni-ulm.de/stochastik/lehre/ws05_06/seminar/ausarbeitung_sauter.pdf

Bildverarbeitung – Lokale Bildoperationen

Bedeutet „Segmentierung“ und „semantische Segmentierung“ das gleiche?

Beides sind inhaltliche Interpretationen eines Bildes auf Pixelebene [1]. Auch die semantische Segmentierung teilt das Bild in Gruppen (Klassen) auf. Allerdings sind nicht die Eigenschaften eines Pixels für die Zuordnung zu einer Klasse ausschlaggebend, sondern die Zugehörigkeit des Pixels zu einer Objektklasse, wie Baum, Auto oder ähnliches.

Beispiel für eine semantische Segmentierung aus [1]



[1] https://de.mathworks.com/solutions/image-video-processing/semantic-segmentation.html

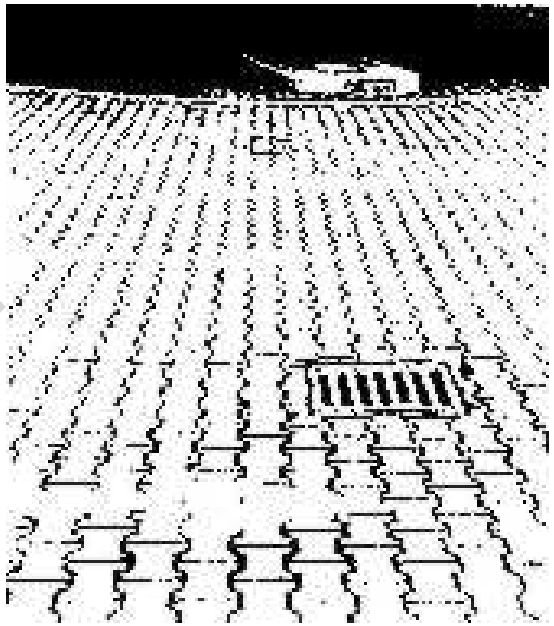
Bildverarbeitung – Lokale Bildoperationen

Segmentierung: Aufgabe ist die Pixel des großen Pfeils im Bildvordergrund einem Segment zuzuordnen und alle anderen Pixel als Hintergrundsegment zusammen zu fassen.

Bildverarbeitung – Lokale Bildoperationen

Segmentierung: Aufgabe ist die Pixel des großen Pfeils im Bildvordergrund einem Segment zuzuordnen und alle anderen Pixel als Hintergrundsegment zusammen zu fassen.

1. Schritt:

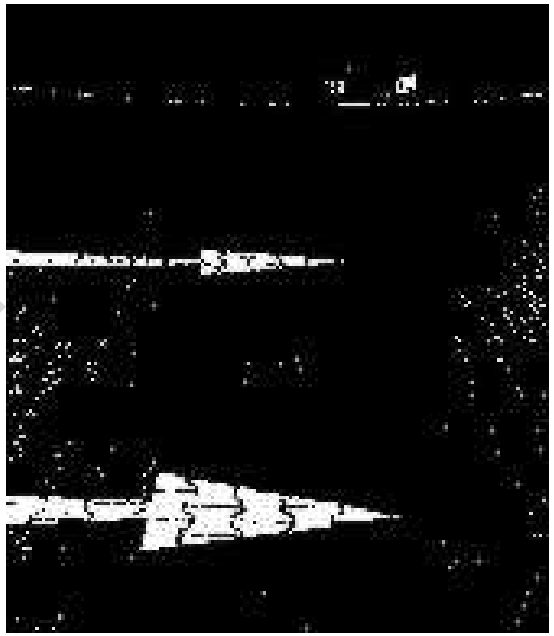


Biniarisierung mit dem Schwellwert 127

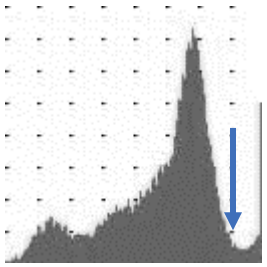
Bildverarbeitung – Lokale Bildoperationen

Segmentierung: Aufgabe ist die Pixel des großen Pfeils im Bildvordergrund einem Segment zuzuordnen und alle anderen Pixel als Hintergrundsegment zusammen zu fassen.

1. Schritt:



Histogramm



Biniarisierung mit dem Schwellwert 225, welcher auf Basis des Histogramms ermittelt wurde. 225 ist der Grauwert der nach dem ersten Minimum im Histogramm von weiß ausgehend kommt.

Bildverarbeitung – Lokale Bildoperationen

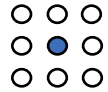
Segmentierung: Aufgabe ist die Pixel des großen Pfeils im Bildvordergrund einem Segment zuzuordnen und alle anderen Pixel als Hintergrundsegment zusammen zu fassen.

Anmerkung:
Die Auflösung des biniarisierten Bildes wurde reduziert, um die Funktionsweise der Rangfolgeoperatoren besser zeigen zu können.

Frage: Wie werden Rangfolgeoperatoren berechnet?

Bildverarbeitung – Lokale Bildoperationen

Berechnung der Rangfolgeoperatoren:

- das Grauwertbild wird, ähnlich wie bei der Faltung, Pixel für Pixel mit einem Kernel abgetastet, der Kernel heißt jetzt aber Strukturelement und kann alle möglichen Formen annehmen.
- Berechnung erfolgt beispielhaft durch das Strukturelement, das der N8-Nachbarschaft entspricht: 
- der zu transformierende Pixel *g(i,j)* liegt in der Mitte des Strukturelements (blau markiert)

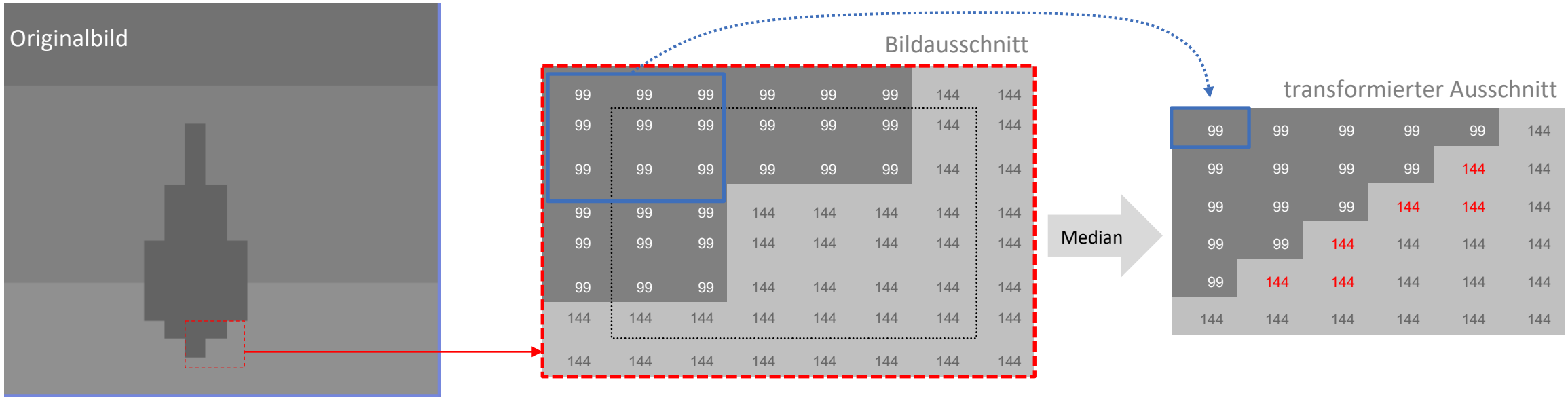
Für jede Pixeltransformation mit einem Rangfolgeoperator gilt:

- die Grauwerte der Pixel welche vom Strukturelement abgedeckt werden, werden nach Größe sortiert: $g_0 \leq g_1 \leq \ldots \leq g_n$
- Je nach Rangfolgeoperator wird der Grauwert *g(i,j)* mit einem Grauwert auf einer bestimmten Positionen der Rangfolge ersetzt...
  - Beim Medianoperator ist das der mittlere Grauwert. Für die N8-Nachbarschaft gilt: $g'(i,j) = g_4$
  - Bei der Dilatation ist das der größte Grauwert. Für die N8-Nachbarschaft gilt: $g'(i,j) = g_8$
  - Bei der Erosion ist das der kleinste Grauwert. Für die N8-Nachbarschaft gilt: $g'(i,j) = g_0$

Bildverarbeitung – Lokale Bildoperationen

Berechnung der Rangfolgeoperatoren: **Medianoperator**

Die Grauwerte der Pixel, welche vom Strukturelement (N8-Nachbarschaft) abgedeckt werden, werden der Größe nach sortiert. Der Medianoperator ersetzt den Grauwert des zu transformierende Pixel mit $g'(i,j) = g_4$.

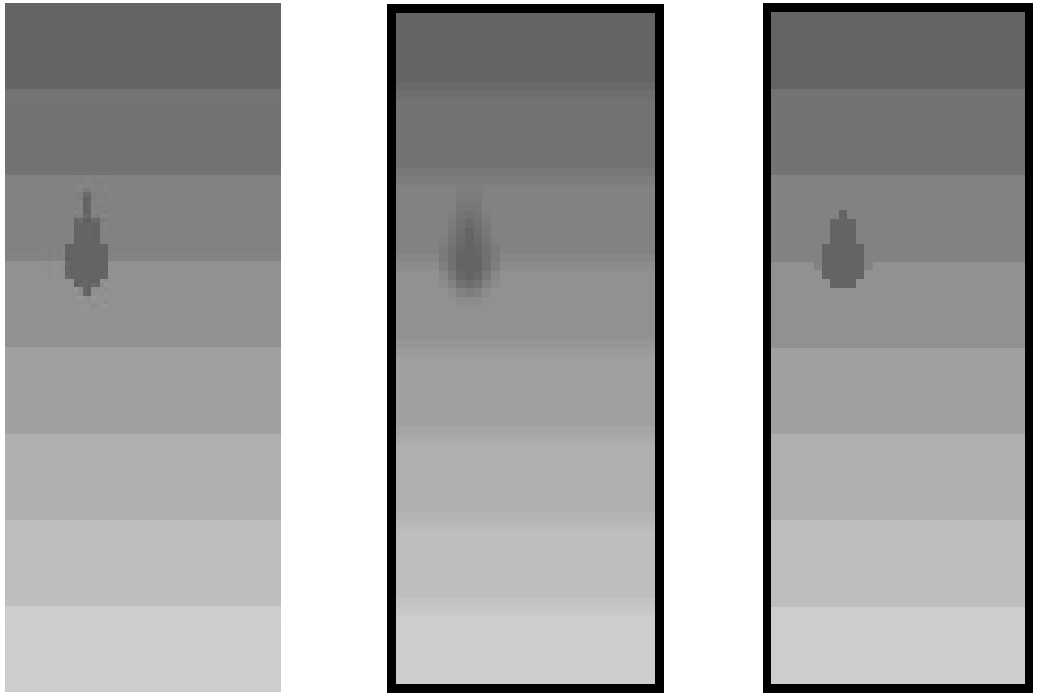


- Verbesserung von verrauschten Bildern: Eliminiert isolierte, fehlerhafte Bildpunkte
- Kanten werden jedoch nicht verwaschen (vergleiche Mittelwertoperator)

Bildverarbeitung – Lokale Bildoperationen

Vergleich: **Medianoperator und Mittelwert**

- Der Medianoperator verbessert verrauschte Bilder: Eliminiert isolierte, fehlerhafte Bildpunkte
- Kanten werden jedoch nicht verwaschen (vergleiche Mittelwertoperator)

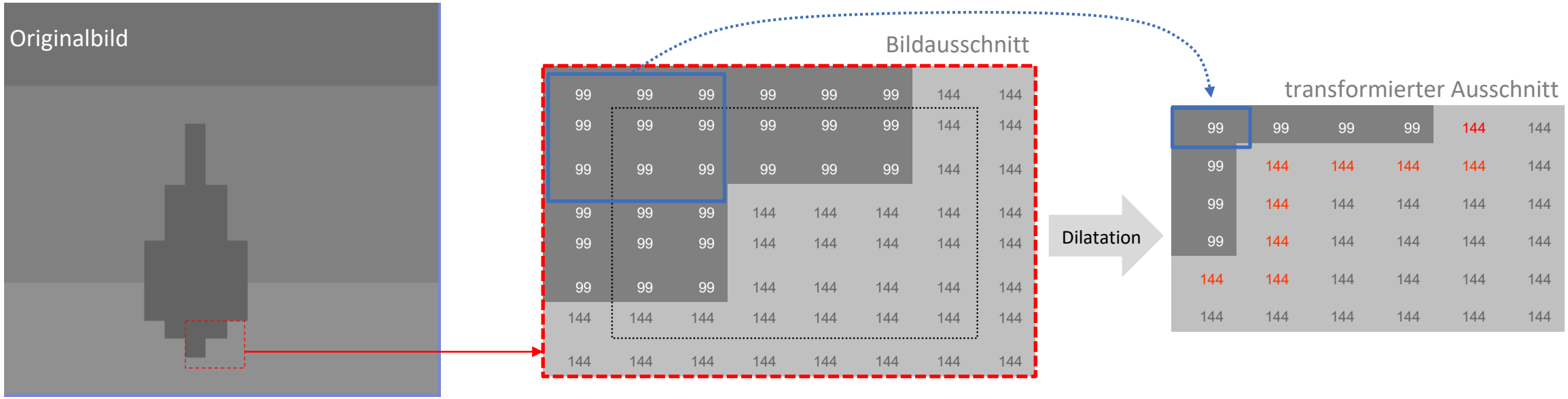


Original | 3x3 Mittelwert | 3x3 Median

Bildverarbeitung – Lokale Bildoperationen

Berechnung der Rangfolgeoperatoren: **Dilatation**

Die Grauwerte der Pixel, welche vom Strukturelement (N8-Nachbarschaft) abgedeckt werden, werden der Größe nach sortiert. Die Dilatation ersetzt den Grauwert des zu transformierende Pixel mit $g'(i,j) = g_8$, dem größten Grauwert.

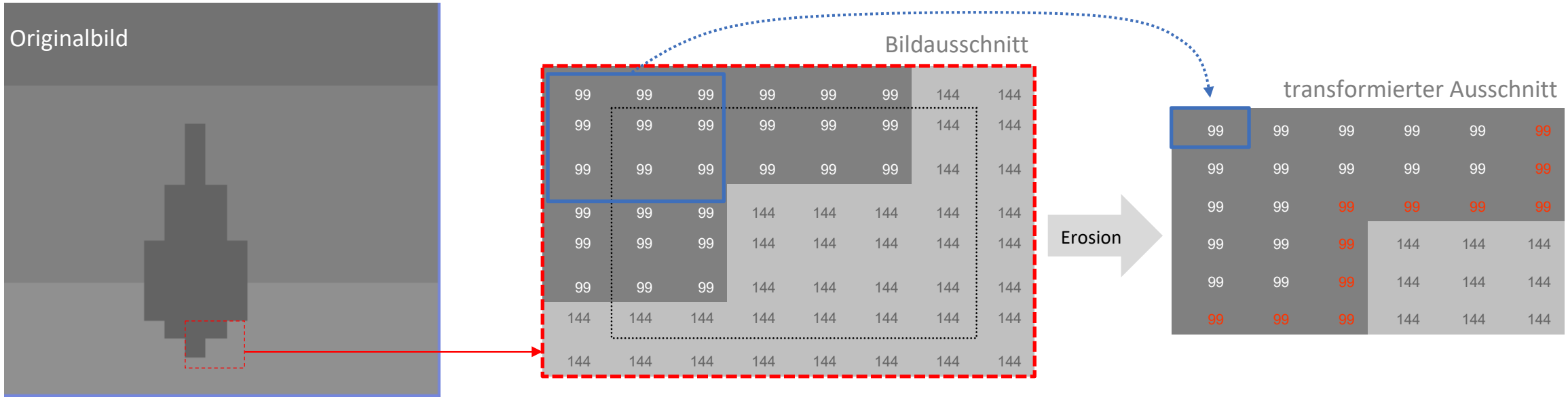


- Die hellen Bildbereiche dehnen sich aus.
- Allgemein gilt: $dil(x,y) = \max_{i,j}\{s_e(x+i, y+j) + k(i,j)\}$ mit: i und j laufen über den Geltungsbereich des Strukturelements.

Bildverarbeitung – Lokale Bildoperationen

Berechnung der Rangfolgeoperatoren: **Erosion**

Die Grauwerte der Pixel welche vom Strukturelement (N8-Nachbarschaft) abgedeckt werden, werden der Größe nach sortiert. Die Erosion ersetzt den Grauwert des zu transformierenden Pixels mit $g'(i,j) = g_0$, dem kleinsten Grauwert.



- Die „dunkleren“ Bildbereiche dehnen sich aus.
- Allgemein gilt: $ero(x,y) = \min_{i,j}\{s_e(x+i, y+j) + k(i,j)\}$ mit: i und j laufen über den Geltungsbereich des Strukturelements.

Bildverarbeitung – Lokale Bildoperationen

Segmentierung: Aufgabe ist die Pixel des großen Pfeils im Bildvordergrund einem Segment zuzuordnen und alle anderen Pixel als Hintergrundsegment zusammenzufassen.

Die Frage: „Wie werden Rangfolgeoperatoren berechnet?“ ist nun beantwortet.

Also zurück zur eigentlichen Aufgabe

Bildverarbeitung – Lokale Bildoperationen

Segmentierung: Aufgabe ist die Pixel des großen Pfeils im Bildvordergrund einem Segment zuzuordnen und alle anderen Pixel als Hintergrundsegment zusammenzufassen.

Die Aufgaben kann in die folgenden Teilaufgaben unterteilt werden:

1. Rauschen eliminieren
2. Lücken innerhalb des Pfeils schließen

Idee zur Elimination des Rauschens:

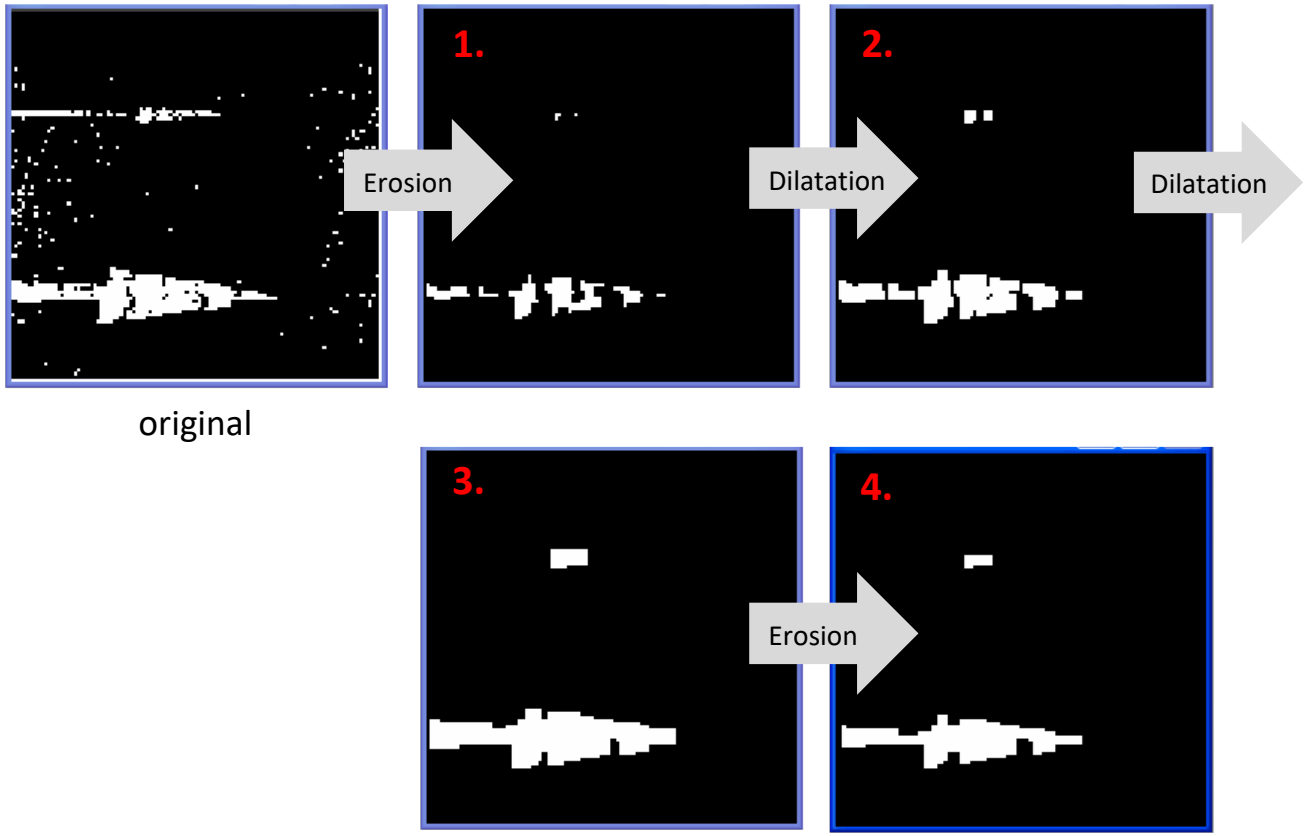
- 1 x Erosion
- 1 x Dilatation

Idee zum Schließen der Pfeillücken:

- 1 x Dilatation
- 1 x Erosion

Bildverarbeitung – Lokale Bildoperationen

Segmentierung: Aufgabe ist die Pixel des großen Pfeils im Bildvordergrund einem Segment zuzuordnen und alle anderen Pixel als Hintergrundsegment zusammenzufassen.



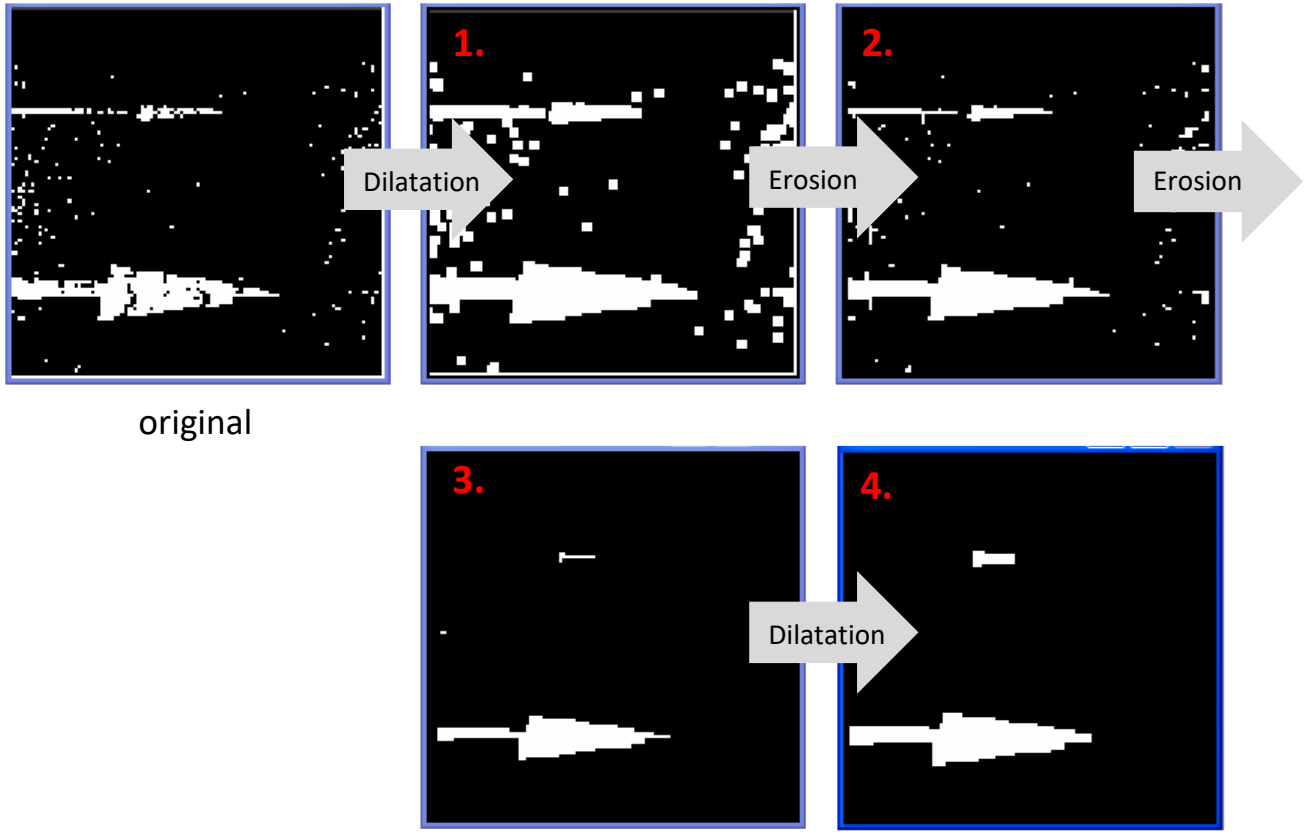
Variante A zur Lösung der Aufgabe:

1. Rauschen eliminieren:
- 1 x Erosion
- 1 x Dilatation
- n · Erosion + n · Dilatation = Opening

2. Lücken innerhalb des Pfeils schließen:
- 1 x Dilatation
- 1 x Erosion
- n · Dilatation + n · Erosion = Closing

Bildverarbeitung – Lokale Bildoperationen

Segmentierung: Aufgabe ist die Pixel des großen Pfeils im Bildvordergrund einem Segment zuzuordnen und alle anderen Pixel als Hintergrundsegment zusammenzufassen.



Variante B zur Lösung der Aufgabe:

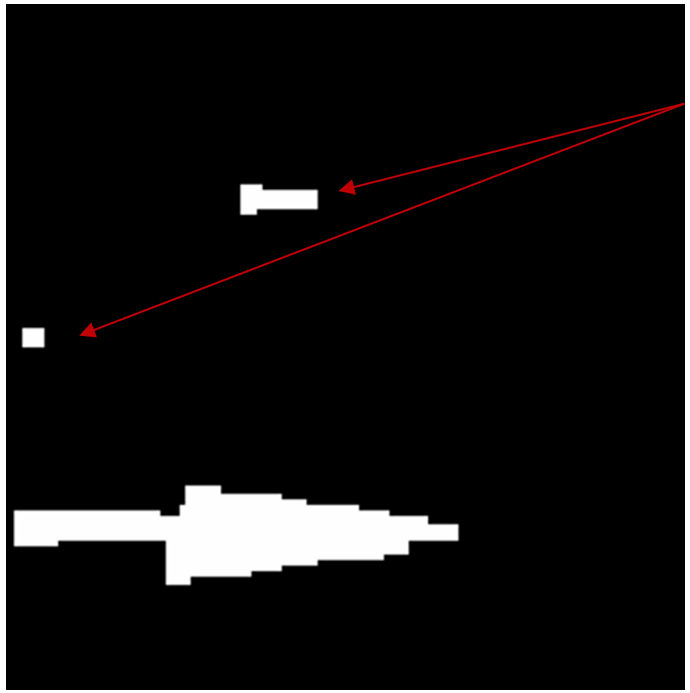
1. Lücken innerhalb des Pfeils schließen:
   - 1 x Dilatation
   - 1 x Erosion
   - n · Dilatation + n · Erosion = Closing
2. Rauschen eliminieren:
   - 1 x Erosion
   - 1 x Dilatation
   - n · Erosion + n · Dilatation = Opening

Diese Variante ist eindeutig die bessere!

Wie bekommt man die Pixelgruppe oben weg?

Bildverarbeitung – Lokale Bildoperationen

Segmentierung: Aufgabe ist die Pixel des großen Pfeils im Bildvordergrund einem Segment zuzuordnen und alle anderen Pixel als Hintergrundsegment zusammenzufassen.



Ergebnis nach Anwendung der Rangfolgeoperatoren

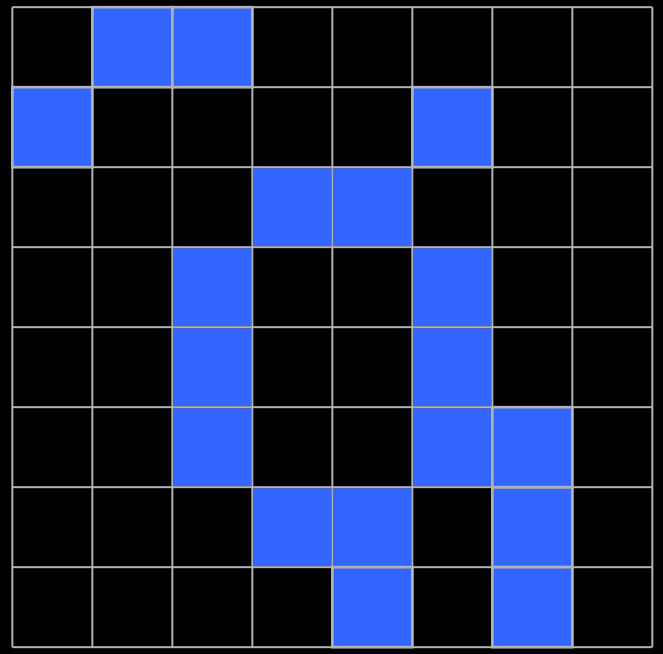
Frage: Wie bekommt man diese beiden Pixelhaufen weg?

Antwort: Man segmentiert das Bild in ein Hintergrund- und drei unterschiedliche Vordergrundobjekte. Nur das größte Vordergrundobjekt behält man. Die Pixel aller anderen Vordergrundobjekte fließen in den Hintergrund ein, werden also schwarz.

Ein Grauwertbild wird segmentiert indem Zusammenhangskomponenten gebildet werden.

Bildverarbeitung – Lokale Bildoperationen

Segmentierung: Was sind Zusammenhangskomponenten (ZHK)?



Der Vordergrund (blau) enthält:

7 N4 ZHK

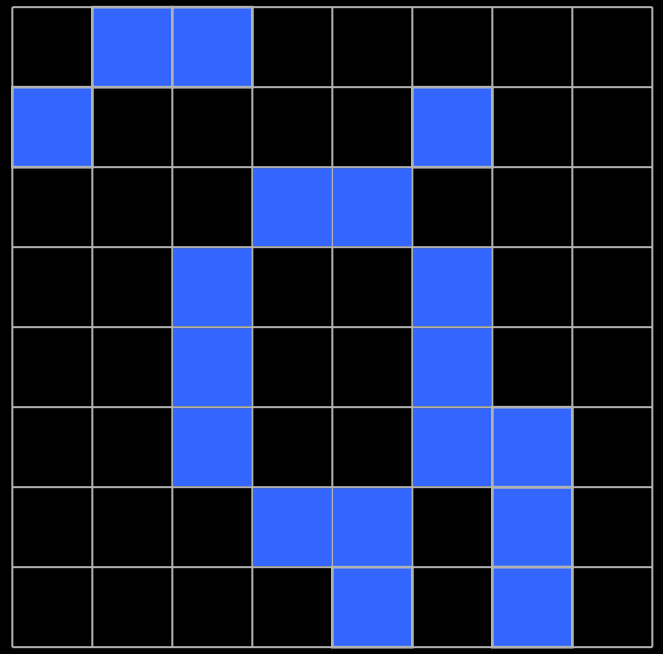
2 N8 ZHK

Vorder- und Hintergrund müssen immer mit entgegengesetztem Nachbarschaftsverhältnis ermittelt werden.

Werden die Vordergrundobjekte durch N4-Nachbarschaften gebildet, so sind die Ecken der Pixel nicht miteinander verbunden und der Hintergrund kann hindurch fließen. Was bedeutet, dass die Hintergrund-pixel über eine N8-Nachbarschaft miteinander verbunden sind.

Bildverarbeitung – Lokale Bildoperationen

Segmentierung: Was sind Zusammenhangskomponenten (ZHK)?



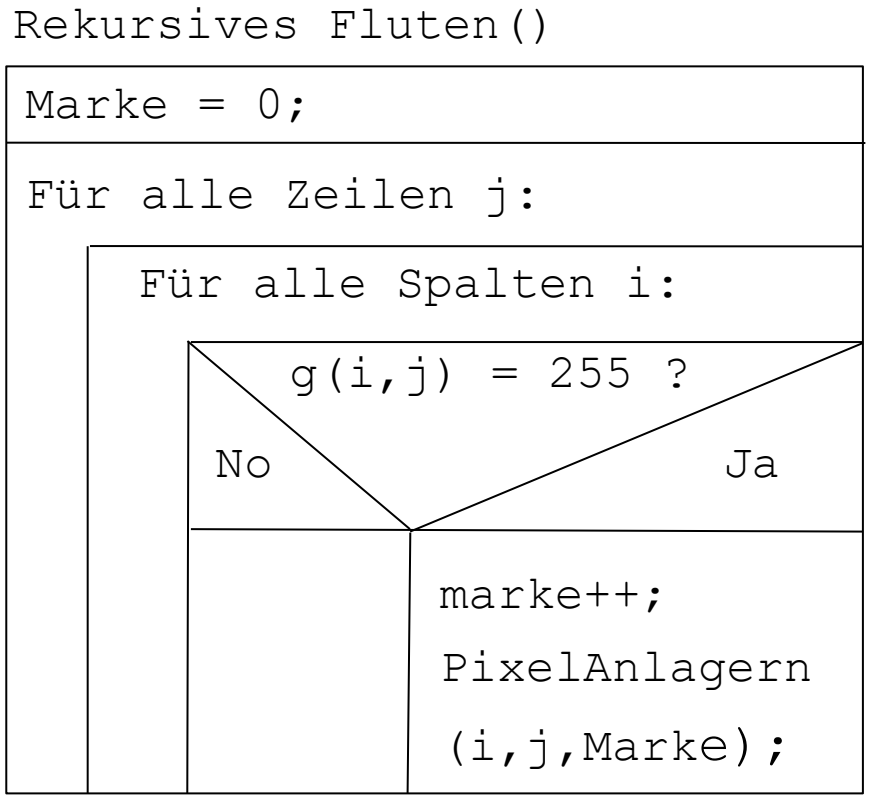
Der Hintergrund (schwarz) enthält:

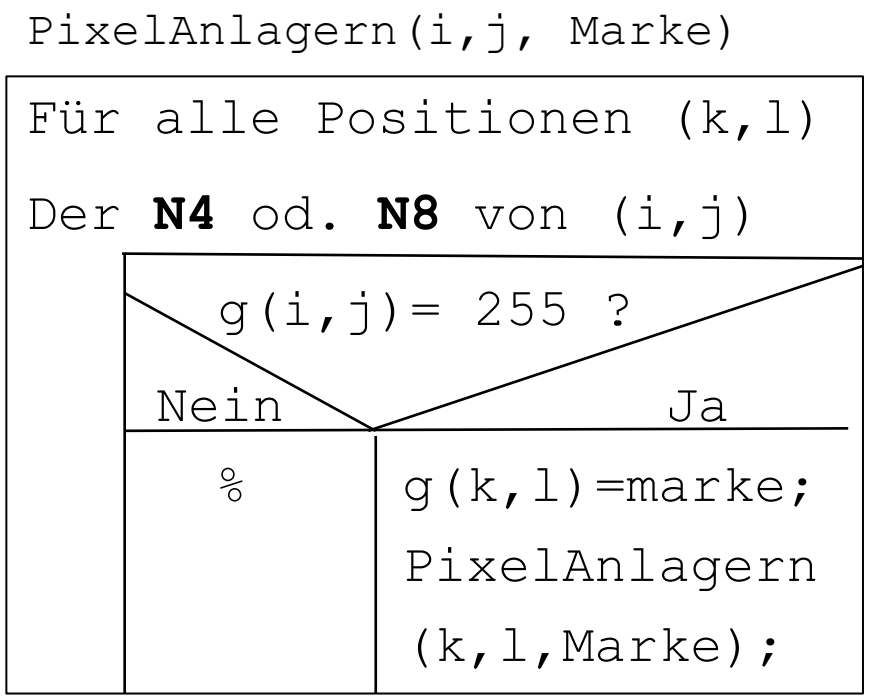
4 N4 ZHK

1 N8 ZHK

Bildverarbeitung – Lokale Bildoperationen

Segmentierung: Rekursives Fluten (Flood Fill Algorithm) – eine Möglichkeit zur Bildung einer Zusammenhangskomponente





Bildverarbeitung – Lokale Bildoperationen

Segmentierung: Rekursives Fluten (Flood Fill Algorithm) – eine Möglichkeit zur Bildung einer Zusammenhangskomponente

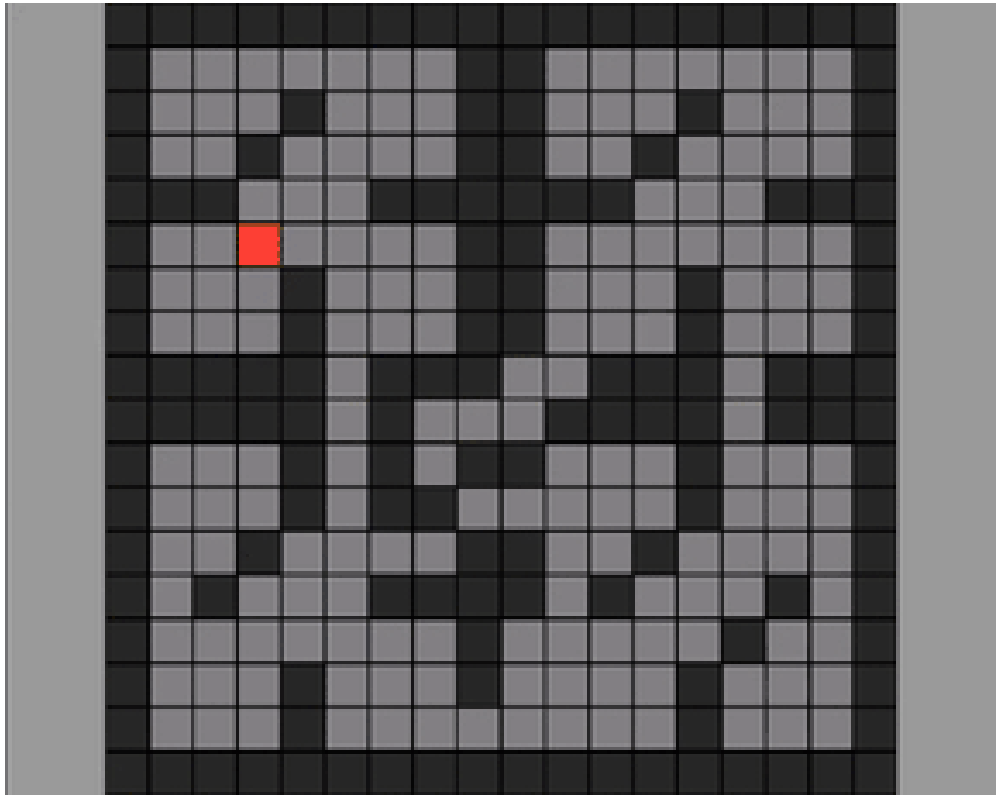


Bild aus: https://vidhu-verma.medium.com/everything-about-flood-fill-algorithm-graphs-28c133e516cf

Bildverarbeitung – Lokale Bildoperationen

Segmentierung: Aufgabe ist die Pixel des großen Pfeils im Bildvordergrund einem Segment zuzuordnen und alle anderen Pixel als Hintergrundsegment zusammenzufassen.

Ergebnis nach Anwendung der Rangfolgeoperatoren

Das Grauwertbild wird durch die Bildung von Zusammenhangskomponenten segmentiert:

- Es gibt ein Hintergrundsegment und drei Vordergrundobjekte.
- Die zwei kleinen Vordergrundobjekte werden gelöscht.
- Es bleibt das größte Vordergrundobjekt übrig. Die Pixel dieses Segments gehören zum großen Pfeil, der gesucht wurde. Die Aufgabe ist gelöst

Bildverarbeitung – Lokale Bildoperationen

Zusammenfassung: Musterbeispiel einer Bildvorverarbeitung

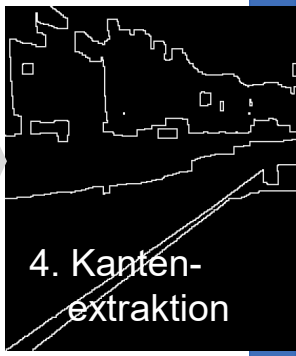
- passiert vor der inhaltlichen Analyse eines Bildes,
- also bevor Bildinhalte als Linien, Segmente usw. klassifiziert werden.
- die Segmentierung und die Hough Transformation sind zwei Beispiele zur Interpretation von Bildinhalten (siehe 4. bzw. 5. Schritt: Hier beginnt die inhaltliche Bildanalyse)

Vorverarbeitung findet entweder auf dem Helligkeitskanal oder auf allen Farbkanälen statt.

1. Bild weichzeichnen, um Grauwertbereiche homogener zu machen. Achtung: Kanten nicht zu sehr verwischen! Hier wurde ein selektiver Weichzeichner verwendet, der die Kanten erhält.

2. Schwellwert ermitteln, um das Bild zu biniarisieren

3. Durch Rangfolgeoperatoren das Bild für die Kantenextraktion oder für die Segmentierung vorbereiten.



5. Hough Transformation zur Erkennung von Geraden

4. Verfahren zur Segmentierung

Bildvorlage aus https://davidgruenewald.de/category/darmstadt/

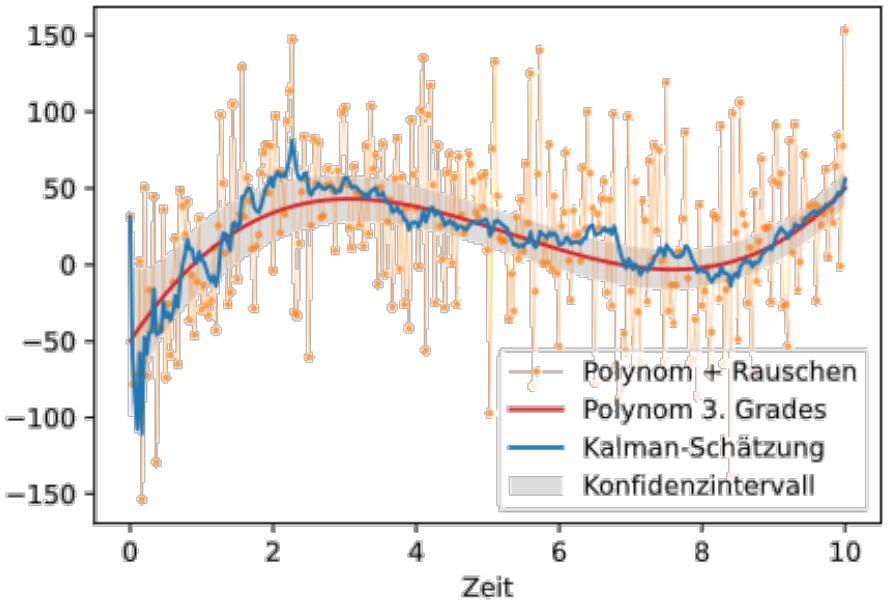
Bildverarbeitung – Lokale Bildoperationen

Kleines Extra: Wie kann man die Pixel, die die kurvige Straße markieren, zu einer Linie verschmelzen?

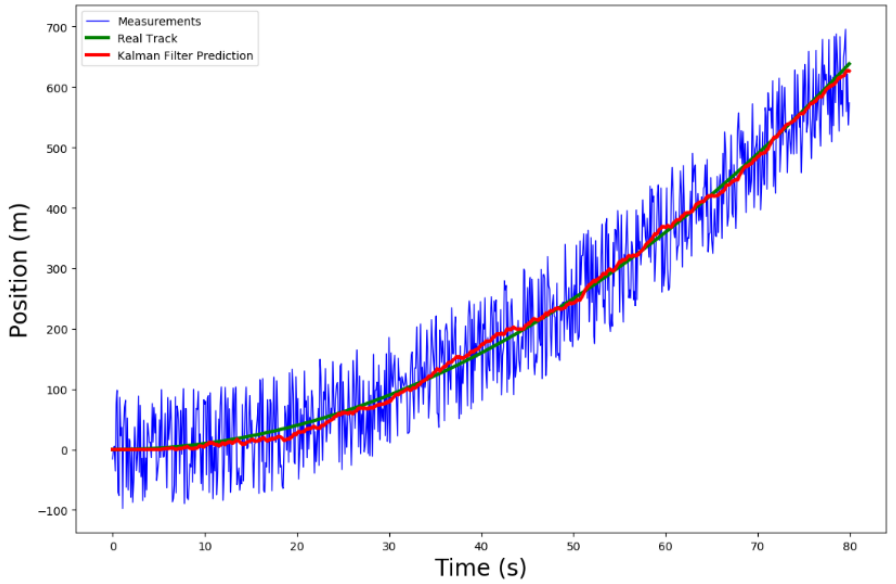
Bildverarbeitung – Lokale Bildoperationen

Kleines Extra: Wie kann man die Pixel, die die kurvige Straße markieren, zu einer Linie verschmelzen?

Eine Möglichkeit besteht darin den Kalman-Filter zu verwenden.
Der Kalman-Filter ist ein mathematisches Verfahren zur iterativen Schätzung von Parametern auf Basis fehlerbehafteter Beobachtungen.



Beispiel aus: https://de.wikipedia.org/wiki/Kalman-Filter#/media/Datei:Kalman_Polynom_Test.svg



Beispiel aus: https://machinelearningspace.com/object-tracking-python/

Bildverarbeitung – Lokale Bildoperationen

Kleines Extra: Der Kalman-Filter ist ein mathematisches Verfahren zur iterativen Schätzung von Parametern auf Basis fehlerbehafteter Beobachtungen.

Aus: https://youtu.be/mwn8xhgNpFY

Bitte auf den schwarzen Rand klicken, um das Video zu starten.

Bildverarbeitung – Lokale Bildoperationen

Punktoperatoren im Vergleich zu lokalen Bildoperatoren

Lokale Bildoperatoren

- Weichzeichner: Mittelwertoperator, Gauß-Filter
- Kantendetektoren: Differenzfilter und Sobel-Operator, Laplace-Operator
- Filter zur Kontrastverbesserung
- Rangfolgeoperatoren: Erosion, Dilatation, Median sowie Opening und Closing
- Segmentierungsverfahren

Bildverarbeitung – Lokale Bildoperationen

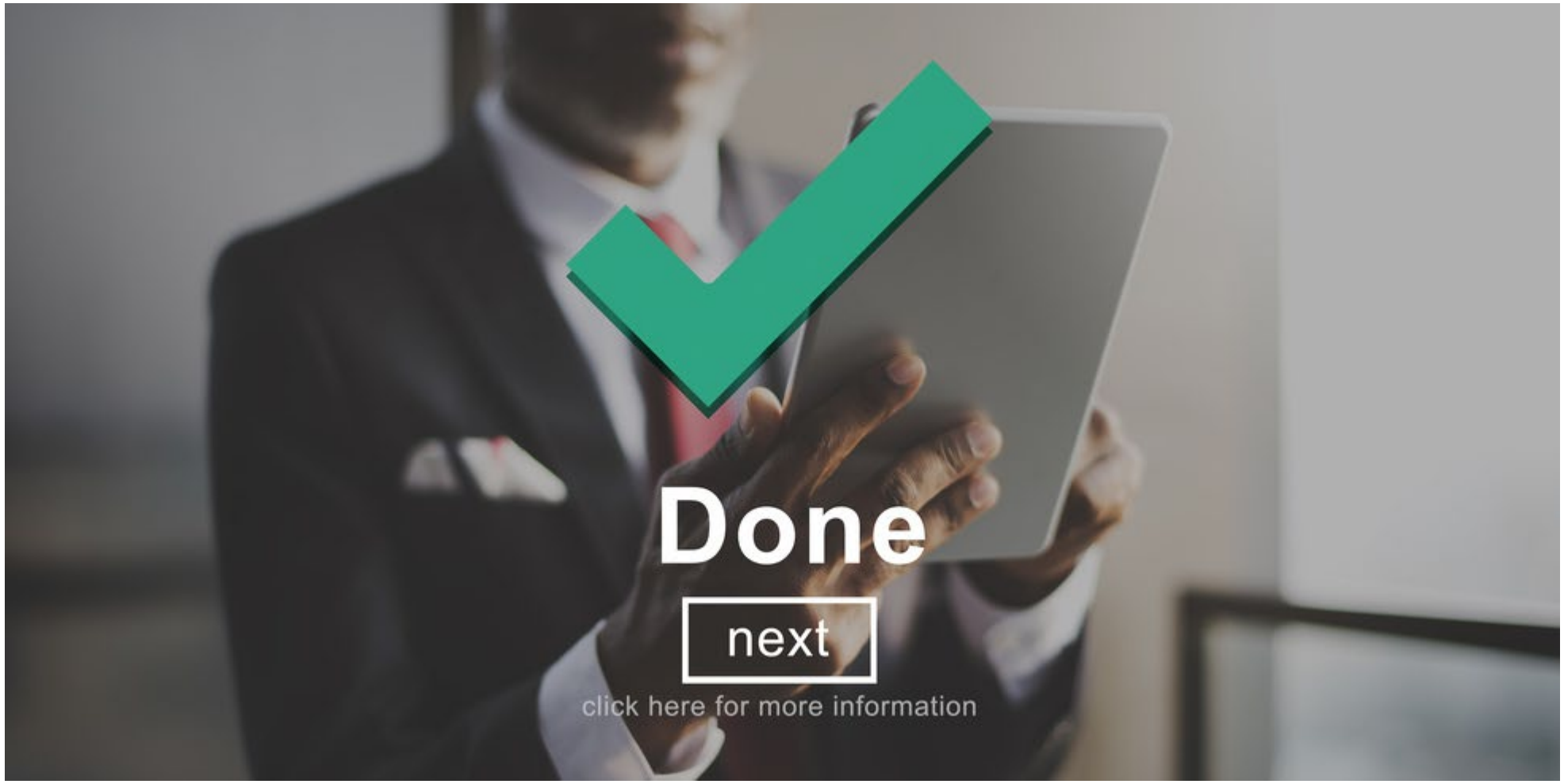


Bild übernommen von: https://social-systems.de/blog/denkanstoesse/die-strategie-steht/