

03-05-H
-709.53

Echtzeitbildverarbeitung (6)

Prof. Dr. Udo Frese

Faltungsoperationen
Linien- bzw. Kantendetektion

Was bisher geschah

- ▶ **k-D Baum für m-Nearest Neighbour Farbsegmentierung**
 - ▶ Klassifiziere eine Farbe gemäß der absoluten Mehrheit der m nächsten Trainingsvektoren (oder weise sie zurück)
 - ▶ Bilde Binärbaum der entlang abwechselnder Dimensionen jeweils halbiert (Aufbau mit Sortieren, Median und Rekursion)
 - ▶ Suche rekursiv, steige zuerst in die nähere Hälfte ab, dann in die fernere, falls der m -t nächste bisher gefundene weiter entfernt ist als die Grenze
- ▶ **Tabelliere das Ergebnis der Klassifikation für alle Farben**
- ▶ **Vermeide Tabelle größer als Cache, dann lieber etwas rechnen**
- ▶ **Bildverarbeitung GermanTeam im Sony Four-legged RoboCup**
 - ▶ Tabellenbasierte Farbsegmentierung als Basis
 - ▶ Projektion auf den Boden durch bekannte Kamerapose
 - ▶ Lokalisation mit Partikelfiltern
 - ▶ Viele spezielle Tricks um Effizienz und Robustheit zu verbessern

Faltungsoperationen

Faltungsoperationen

Pixel mit seinen Nachbarn verknüpfen

- ▶ **Bisher: Pixel alleine (Schwellwert / Farbsegmentierung)**
- ▶ **Jetzt: Ein Pixel und seine Nachbarn**
 - ▶ ein Hilfsbild ausrechnen
 - ▶ Pixel im Hilfsbild hängt von Nachbarn im Originalbild ab
 - ▶ im Hilfsbild Pixel einzeln betrachten
- ▶ **Allgemeine Erfahrung:
Lineare Operationen sind**
 - ▶ strukturiert
 - ▶ leicht zu verstehen
 - ▶ oft die Basis nichtlinearer Operationen

Faltungsoperationen

Lineare Abbildung von Bildern

▶ Axiome:

- ▶ $f(p_1+p_2) = f(p_1)+f(p_2)$
- ▶ $f(\lambda p)=\lambda f(p)$

▶ Ergebnispixel $f(p)_{x',y'}$ ist gewichtete Summe der Eingangspixel $p_{x,y}$.

$$f(p)_{x',y'} = \sum_{x,y} \alpha_{x,y,x',y'} p_{x,y}$$

▶ zu allgemein, weil für 1024×768 Bild $(1024 \cdot 768)^2 =$ $6.2 \cdot 10^{11}$ Koeffizienten α

▶ Axiom: Translationsinvariant

- ▶ $f(\text{Trans}(p)) = \text{Trans}(f(p))$

▶ Wirkt an jedem Pixel gleich.

▶ Koeffizient $\alpha_{x,y,x',y'}$ hängt nur von der Relativposition $x-x'$, $y-y'$ ab.

$$\alpha_{x,y,x',y'} = \alpha_{x-x',y-y'}$$

$$f(p)_{x',y'} = \sum_{x,y} \alpha_{x-x',y-y'} p_{x,y}$$

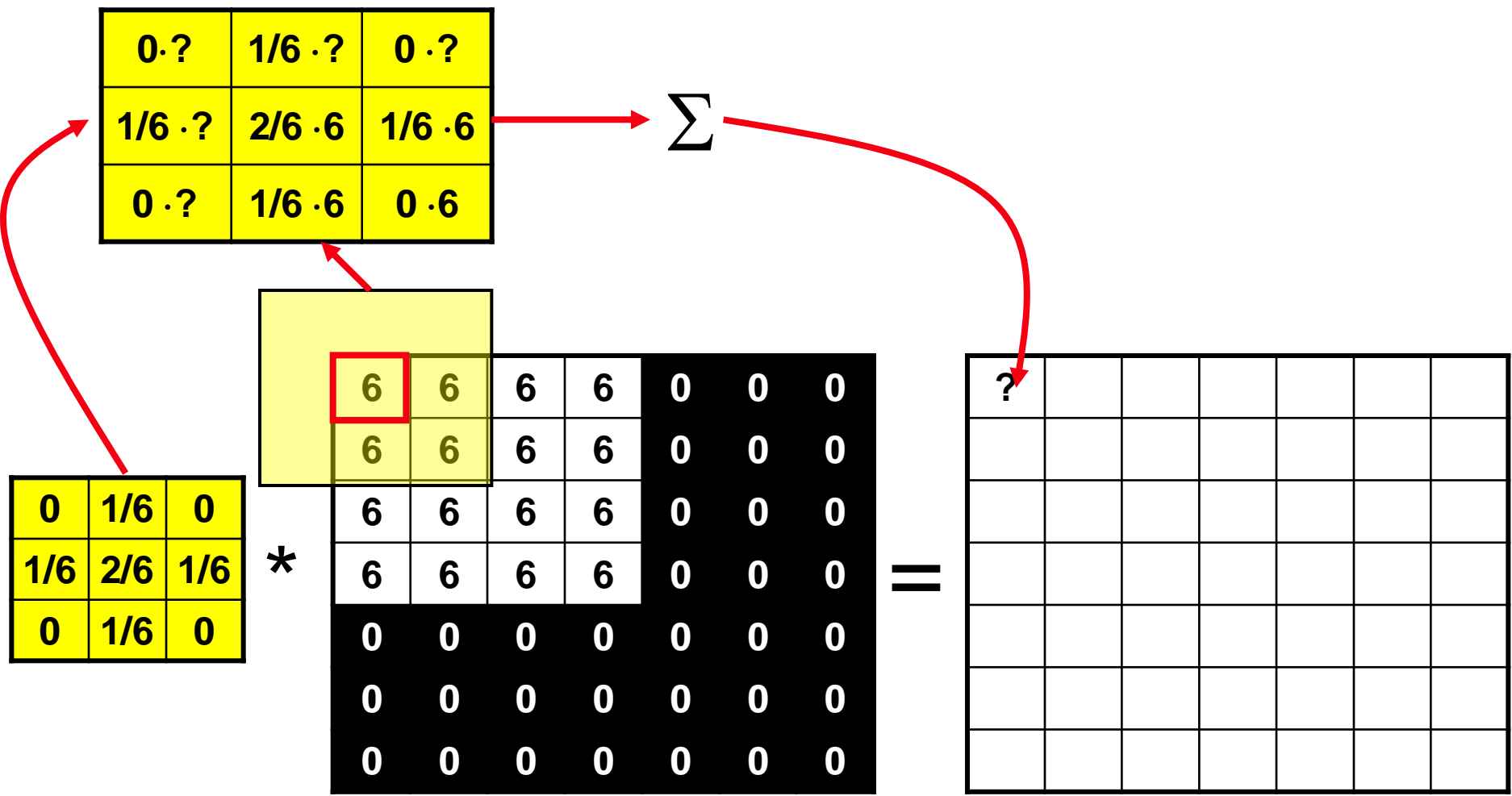
Faltungsoperationen

Lineare Abbildung von Bildern

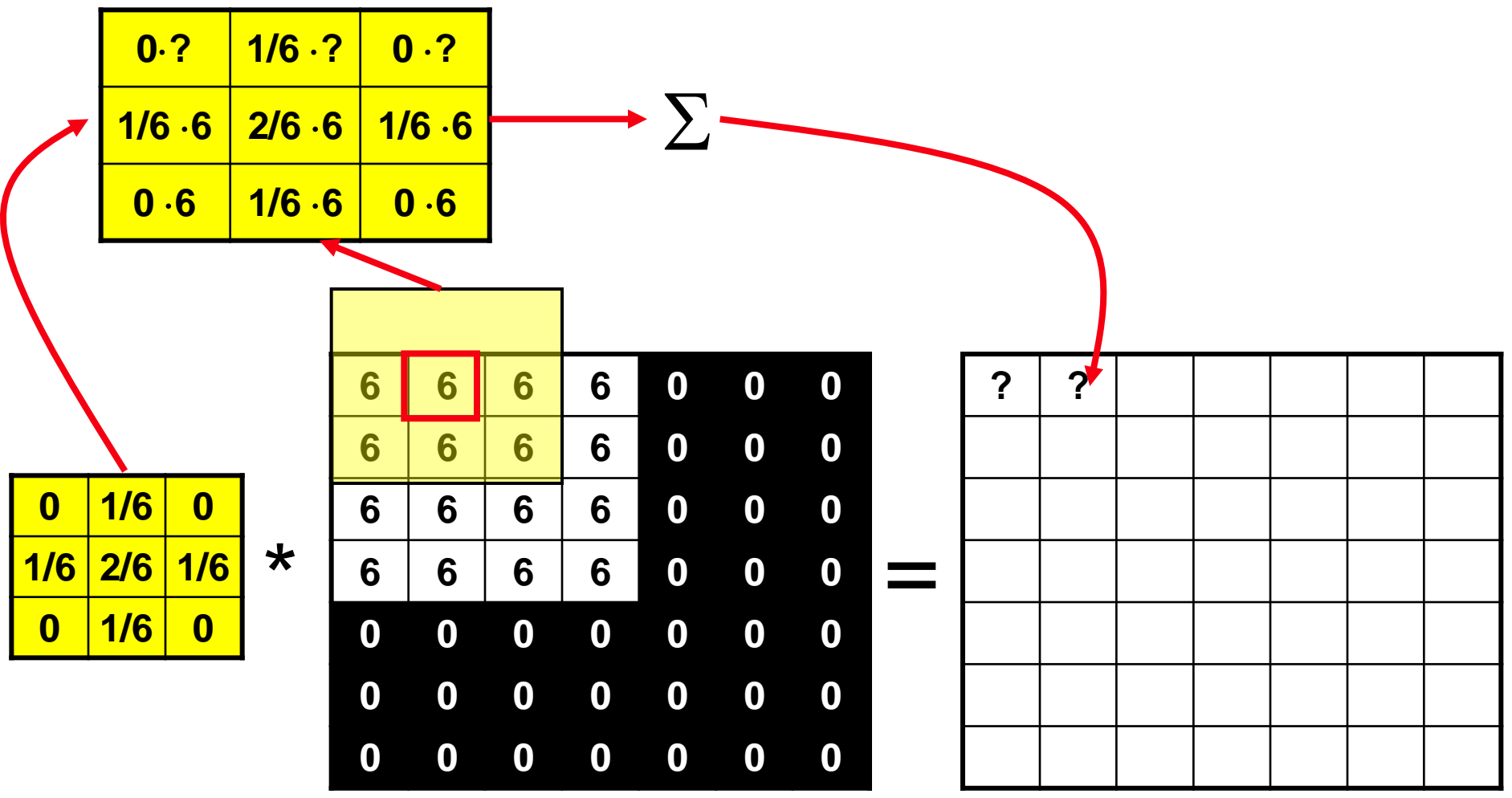
- ▶ **Faltung, Konvolution, Operator ***
- ▶ **α Filter, Filterkern oder Faltungsmaske**
- ▶ **Filter α ist wie ein Bild, * verknüpft also Bilder**
- ▶ **Oft ist α lokal**
 - ▶ 0 für große $x-x'$, $y-y'$
 - ▶ Ergebnispixel wird nur von kleiner Nachbarschaft des Eingangspixels beeinflusst

$$p' = \alpha * p$$

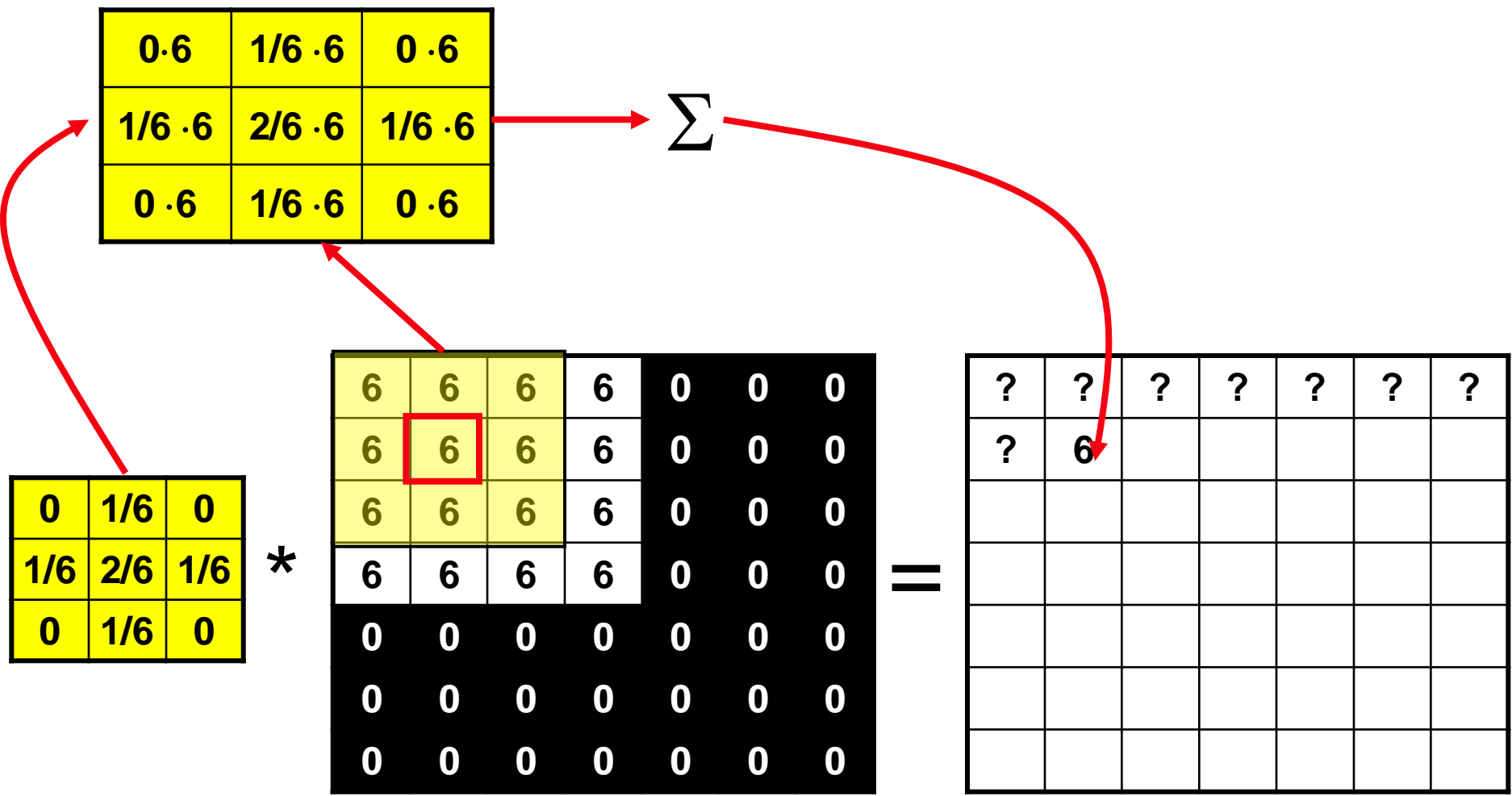
Faltungsoperationen



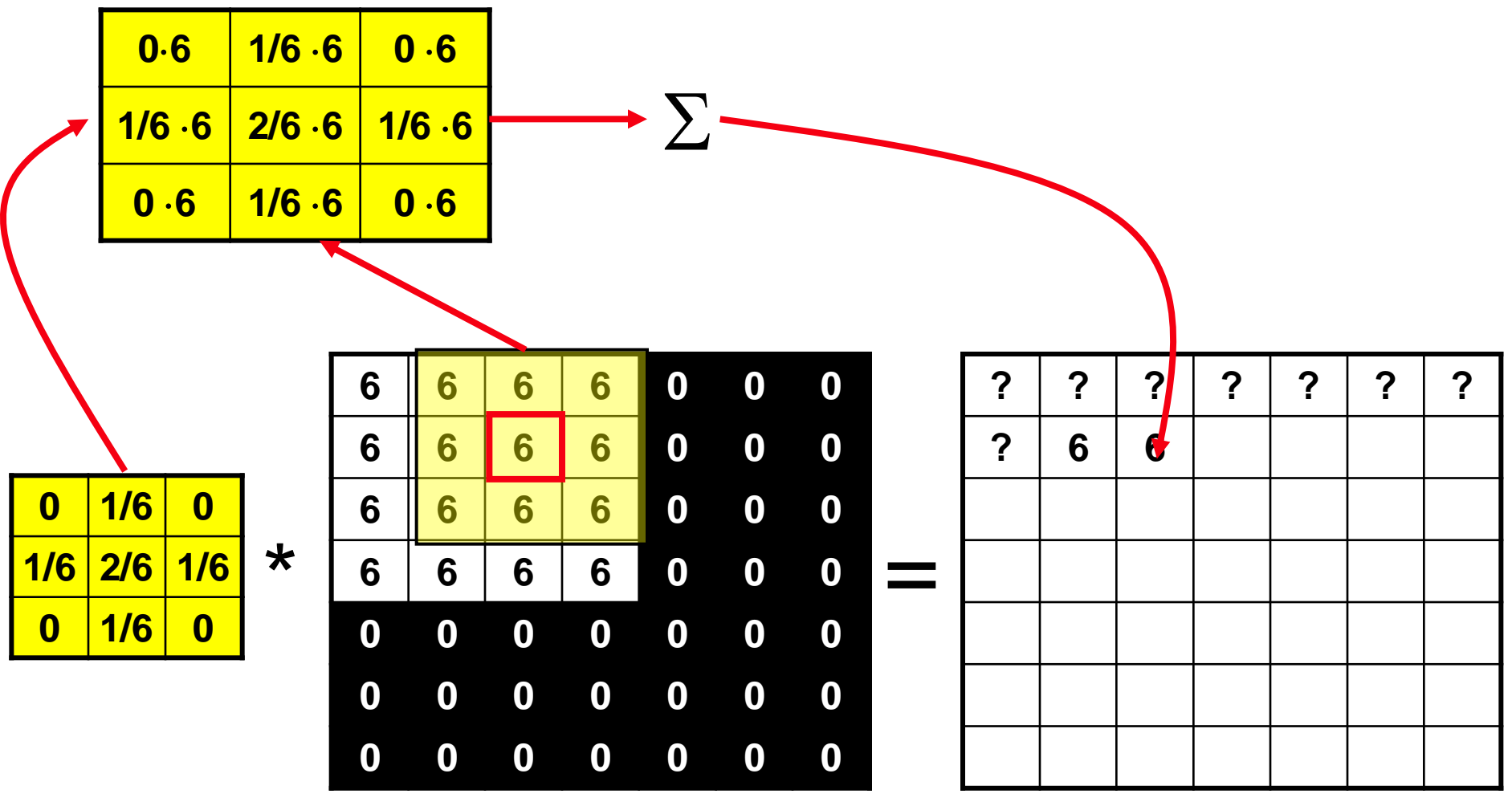
Faltungsoperationen



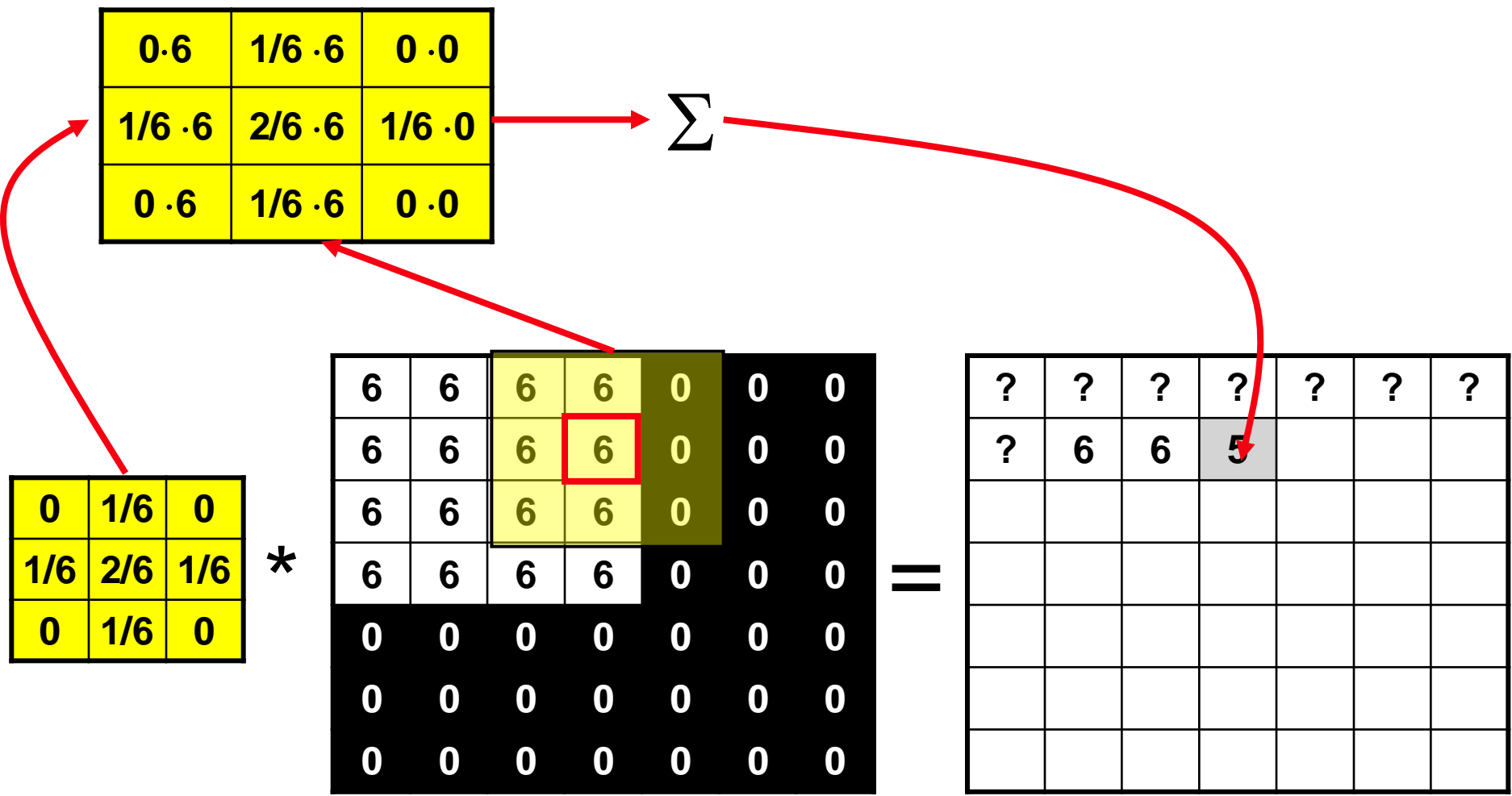
Faltungsoperationen



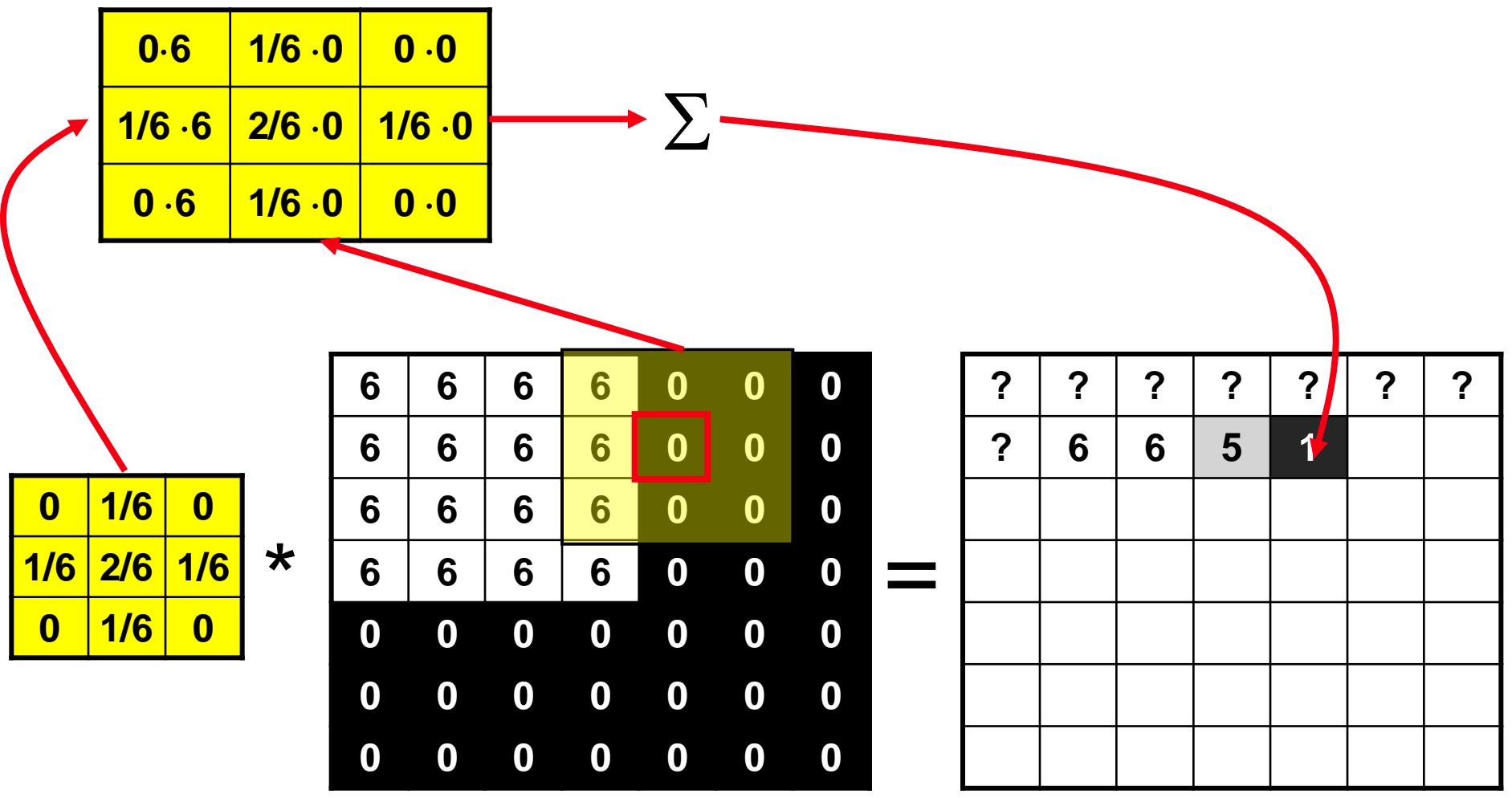
Faltungsoperationen



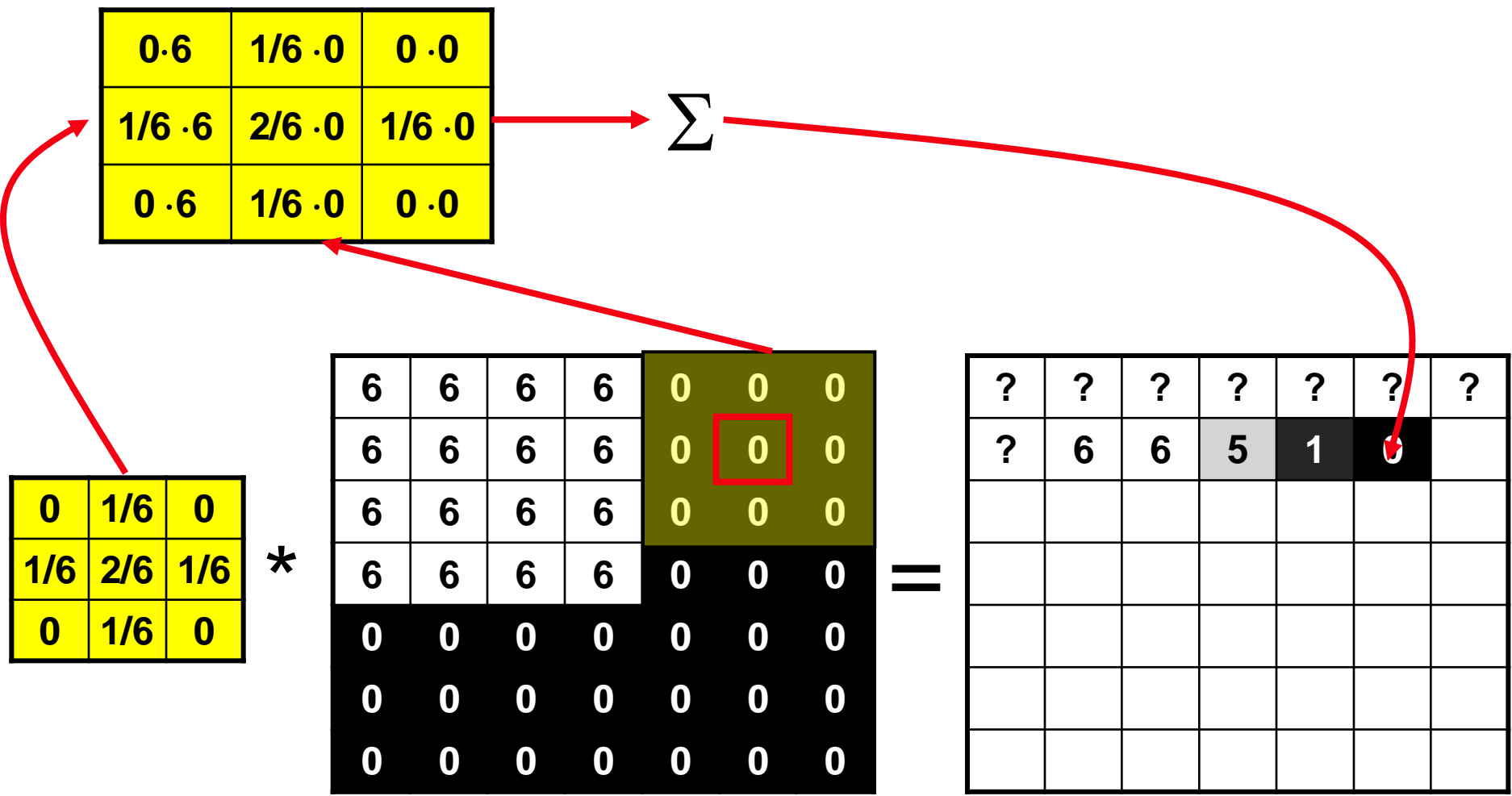
Faltungsoperationen



Faltungsoperationen

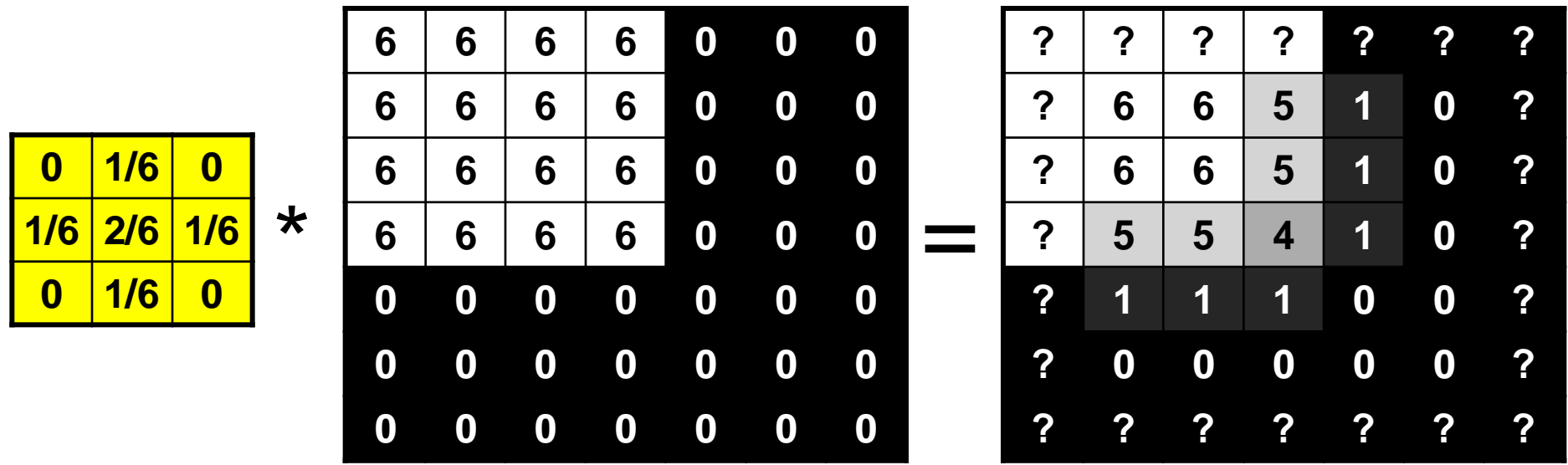


Faltungsoperationen



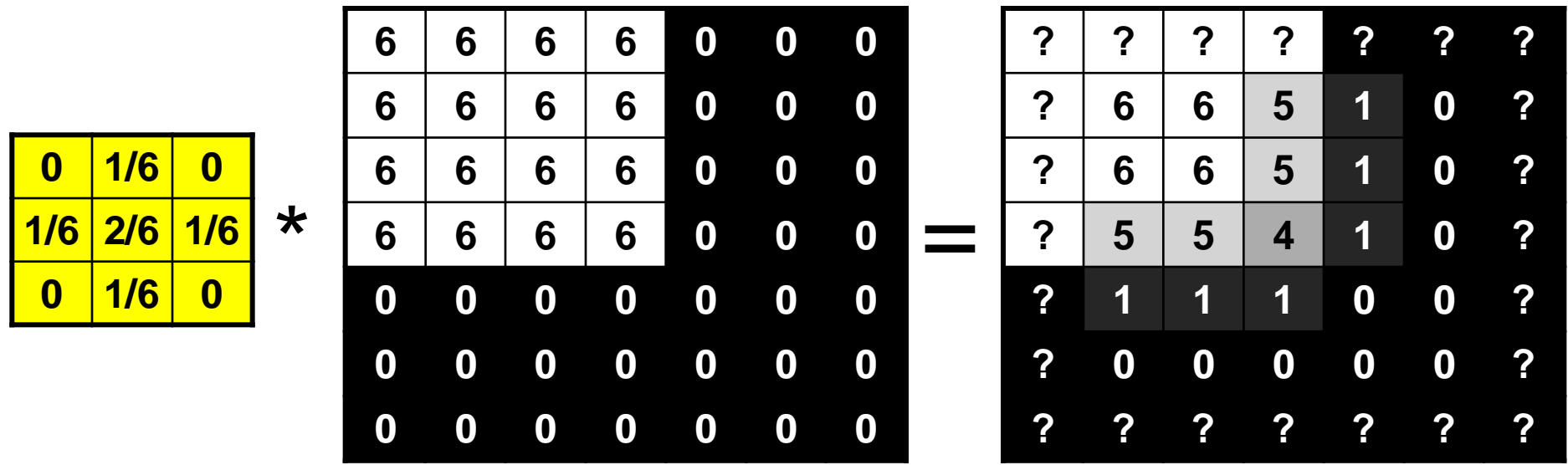
Faltungsoperationen

- Frage an das Auditorium: Wie würde man sprachlich die Wirkung des Filters beschreiben?

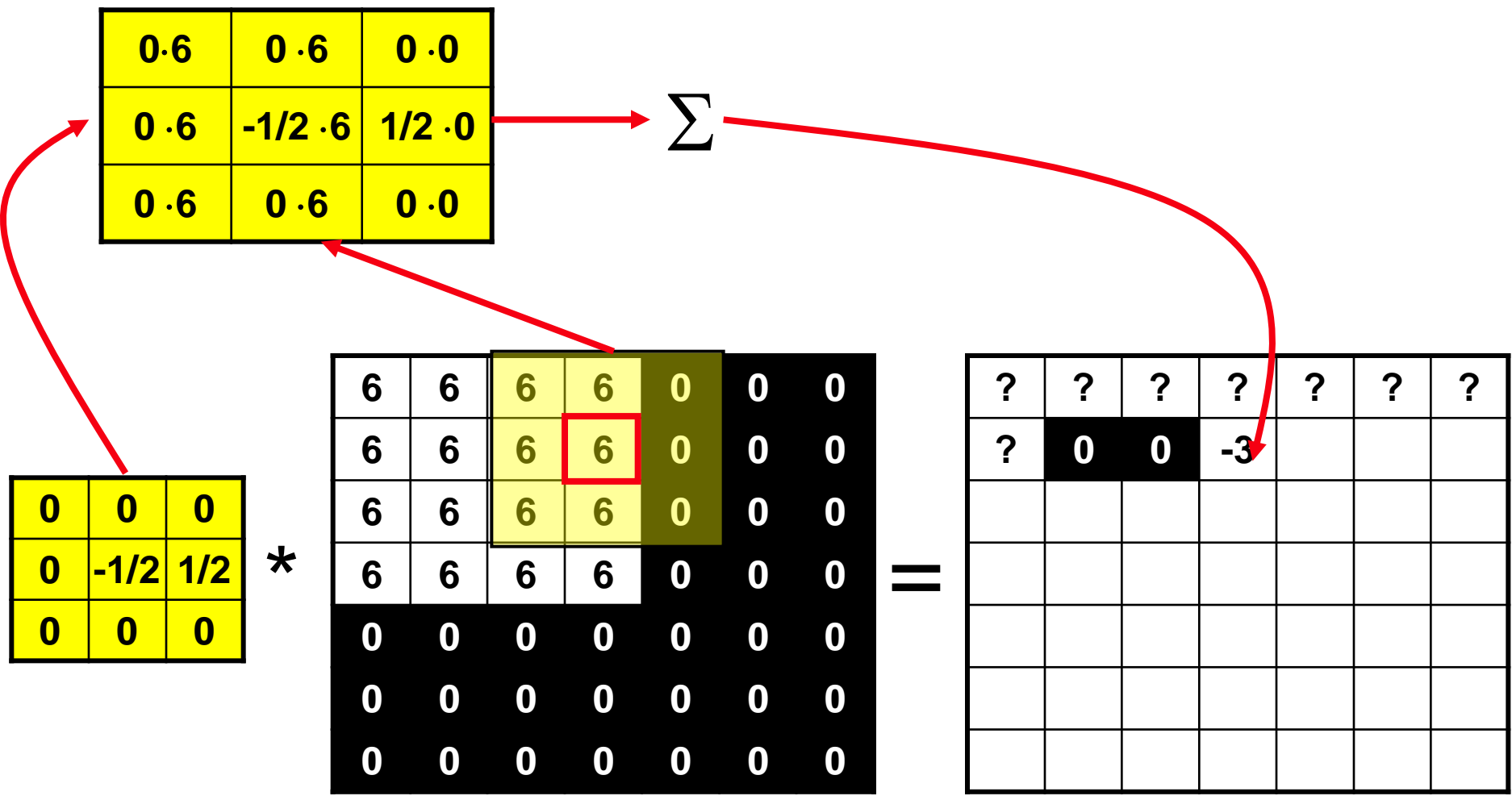


Faltungsoperationen

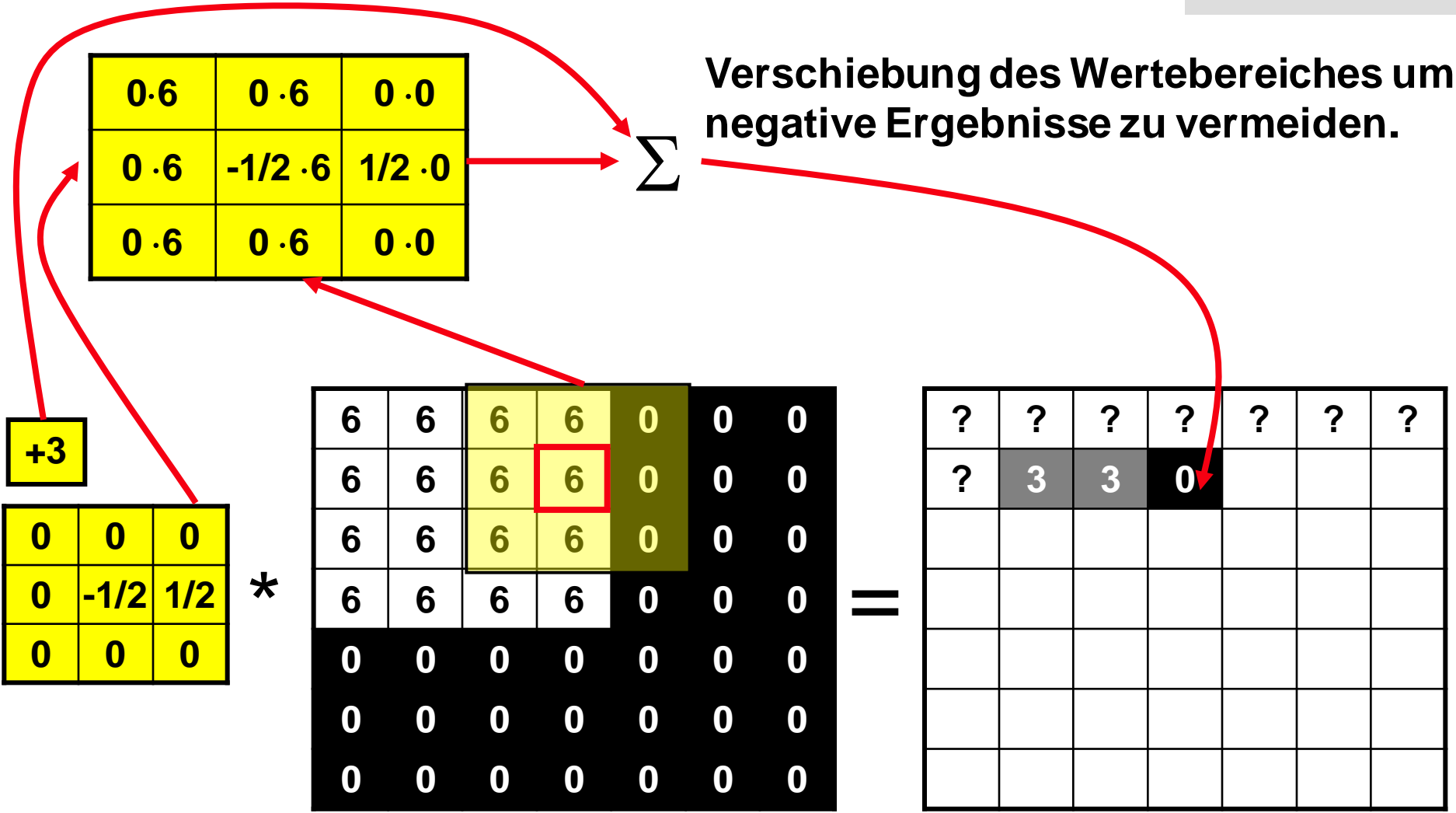
- ▶ Frage an das Auditorium: Wie würde man sprachlich die Wirkung des Filters beschreiben?
- ▶ Das Bild wird geglättet oder verwischt oder unscharf.



Faltungsoperationen



Faltungsoperationen



+3

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

*

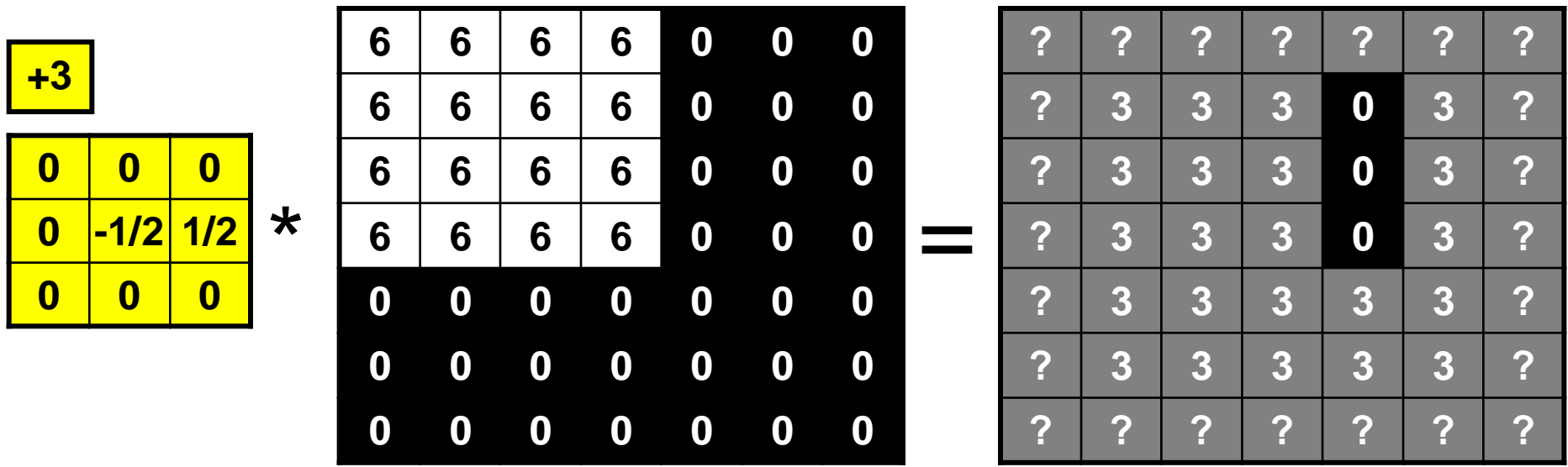
6	6	6	6	0	0	0
6	6	6	6	0	0	0
6	6	6	6	0	0	0
6	6	6	6	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

=

?	?	?	?	?	?	?
?	3	3	0			

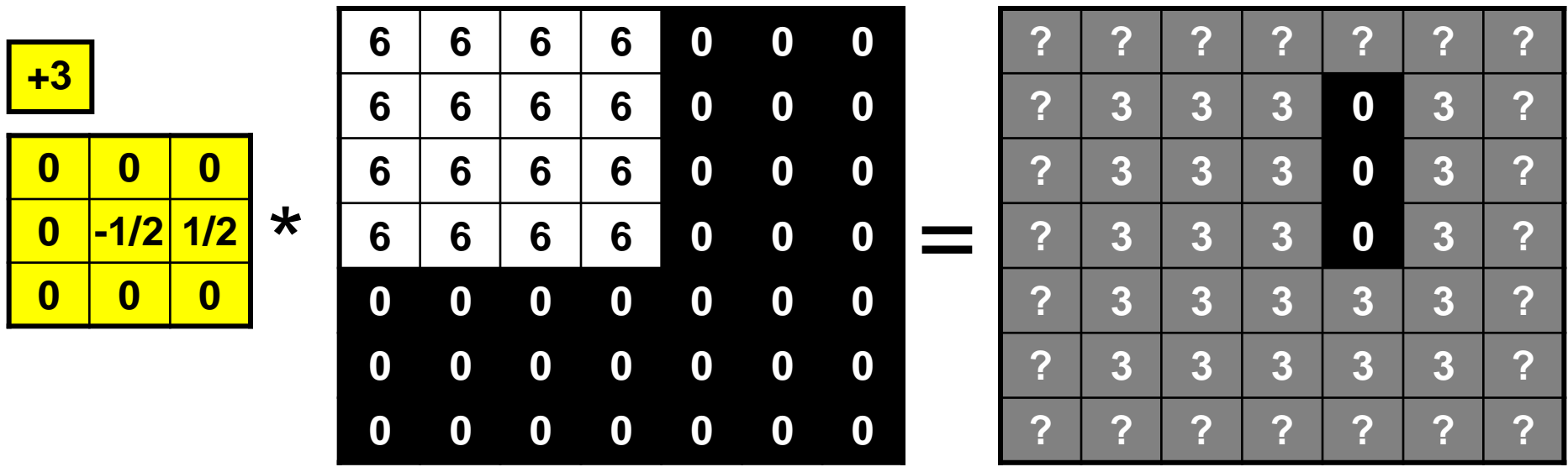
Faltungsoperationen

- Frage an das Auditorium: Wie würde man sprachlich die Wirkung des Filters beschreiben?



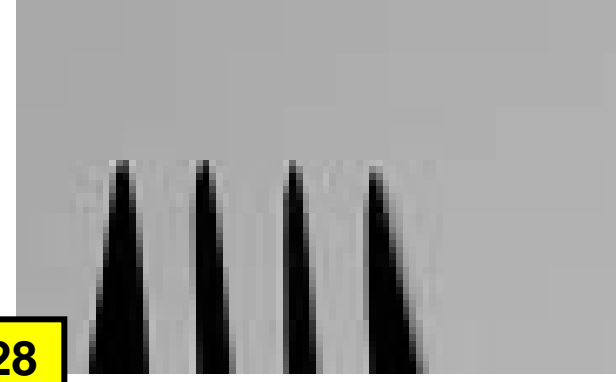
Faltungsoperationen

- ▶ Frage an das Auditorium: Wie würde man sprachlich die Wirkung des Filters beschreiben?
- ▶ Vertikale Kanten werden erkannt.



► Frage an das Auditorium:
Welches Bild gehört zu welchem Filter?

0	0	1/28	0	0
0	2/28	3/28	2/28	0
1/28	3/28	4/28	3/28	1/28
0	2/28	3/28	2/28	0
0	0	1/28	0	0



+128

+128

+128

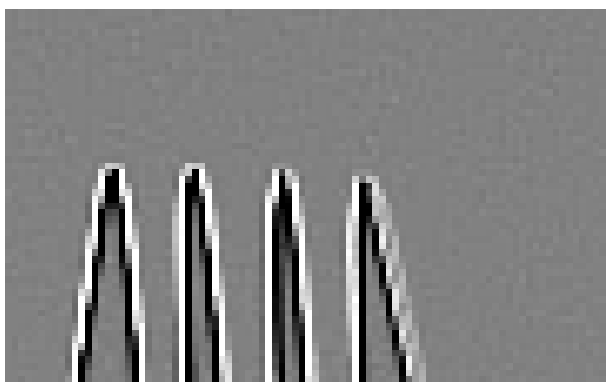
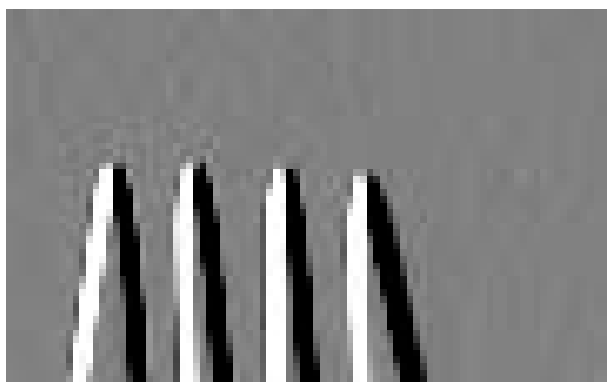
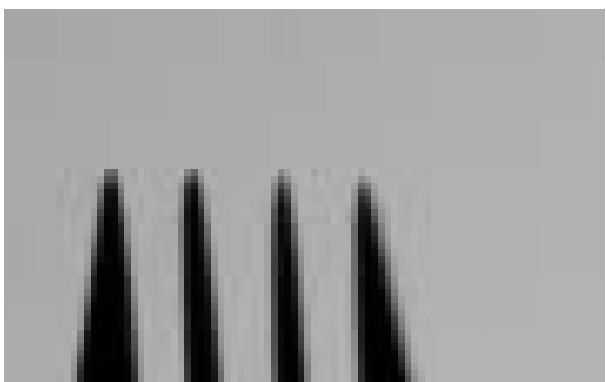
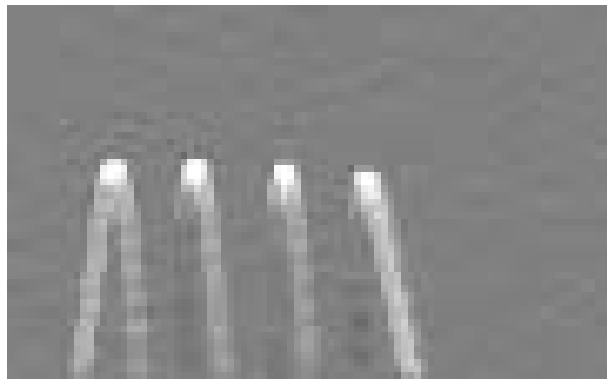
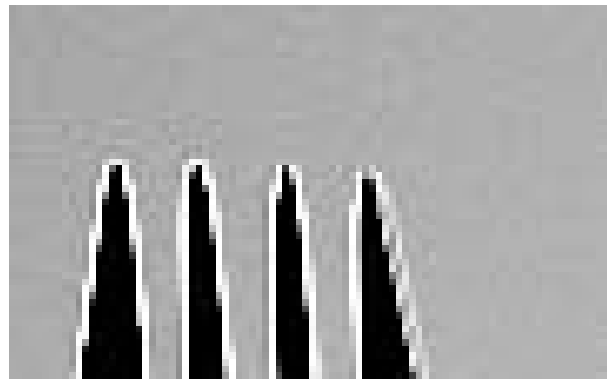
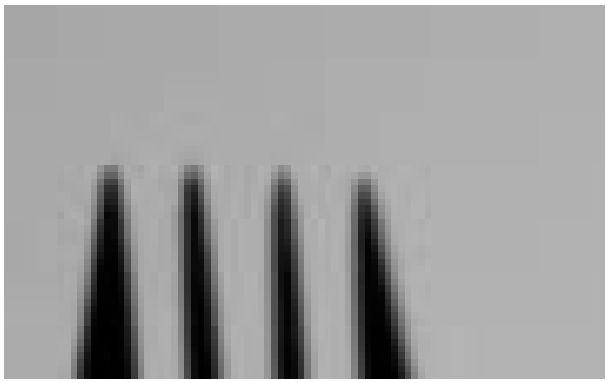
1	1	1
0	0	0
-1	-1	-1

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

0	1/8	0
1/8	4/8	1/8
0	1/8	0

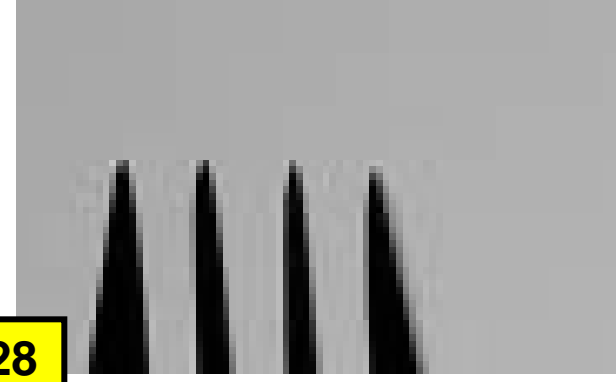
-1/2	-1/2	-1/2
-1/2	8/2	-1/2
-1/2	-1/2	-1/2

-1/2	-1/2	-1/2
-1/2	10/2	-1/2
-1/2	-1/2	-1/2

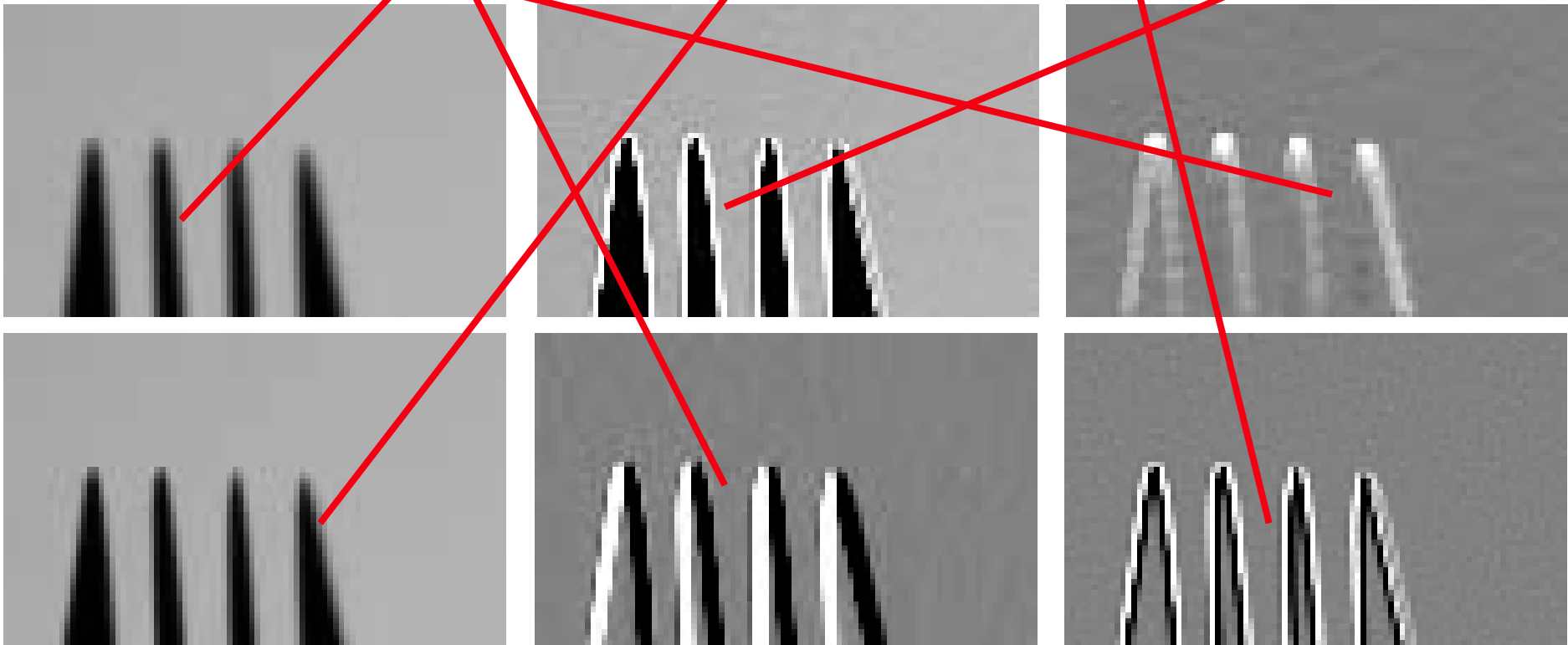


► Frage an das Auditorium:
 Welches Bild gehört zu welchem Filter?

0	0	1/28	0	0
0	2/28	3/28	2/28	0
1/28	3/28	4/28	3/28	1/28
0	2/28	3/28	2/28	0
0	0	1/28	0	0

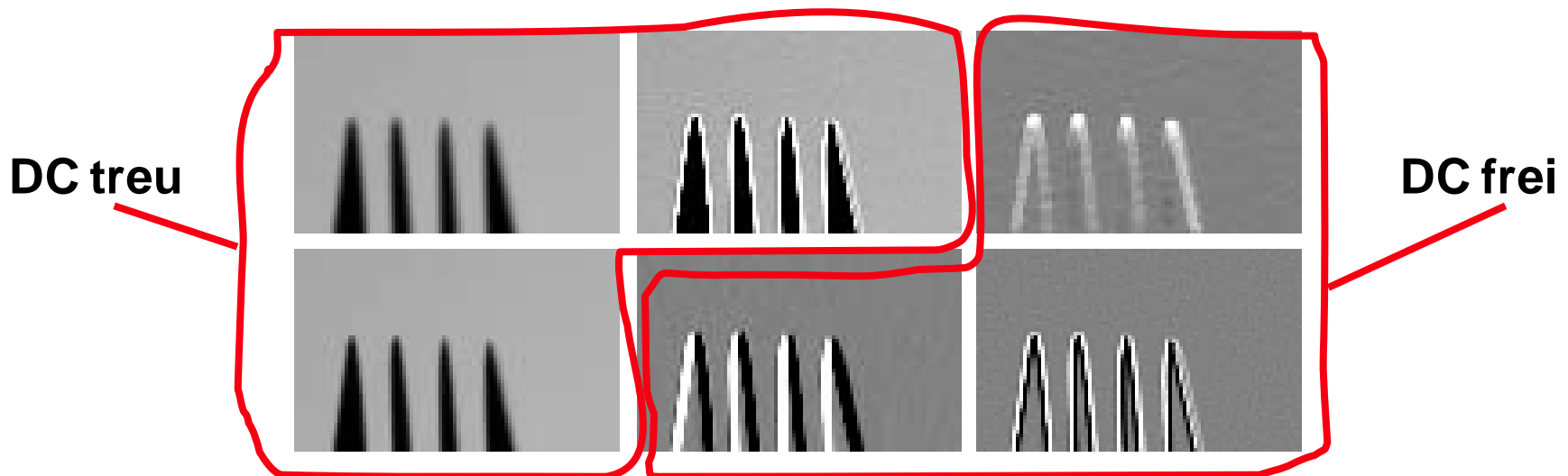


+128	+128		+128																																														
<table border="1"> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>-1</td><td>-1</td><td>-1</td></tr> </table>	1	1	1	0	0	0	-1	-1	-1	<table border="1"> <tr><td>1</td><td>0</td><td>-1</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> </table>	1	0	-1	1	0	-1	1	0	-1	<table border="1"> <tr><td>0</td><td>1/8</td><td>0</td></tr> <tr><td>1/8</td><td>4/8</td><td>1/8</td></tr> <tr><td>0</td><td>1/8</td><td>0</td></tr> </table>	0	1/8	0	1/8	4/8	1/8	0	1/8	0	<table border="1"> <tr><td>-1/2</td><td>-1/2</td><td>-1/2</td></tr> <tr><td>-1/2</td><td>8/2</td><td>-1/2</td></tr> <tr><td>-1/2</td><td>-1/2</td><td>-1/2</td></tr> </table>	-1/2	-1/2	-1/2	-1/2	8/2	-1/2	-1/2	-1/2	-1/2	<table border="1"> <tr><td>-1/2</td><td>-1/2</td><td>-1/2</td></tr> <tr><td>-1/2</td><td>10/2</td><td>-1/2</td></tr> <tr><td>-1/2</td><td>-1/2</td><td>-1/2</td></tr> </table>	-1/2	-1/2	-1/2	-1/2	10/2	-1/2	-1/2	-1/2	-1/2
1	1	1																																															
0	0	0																																															
-1	-1	-1																																															
1	0	-1																																															
1	0	-1																																															
1	0	-1																																															
0	1/8	0																																															
1/8	4/8	1/8																																															
0	1/8	0																																															
-1/2	-1/2	-1/2																																															
-1/2	8/2	-1/2																																															
-1/2	-1/2	-1/2																																															
-1/2	-1/2	-1/2																																															
-1/2	10/2	-1/2																																															
-1/2	-1/2	-1/2																																															



Faltungsoperationen

- ▶ **Summe der Filterkoeffizienten ist Antwort auf konstantes Bild**
- ▶ **DC Anteil, von Directed Current = Gleichstrom**
- ▶ **DC Anteil 0: Helligkeit „im Großen“ spielt keine Rolle (DC frei).**
- ▶ **DC Anteil 1: Helligkeit „im Großen“ bleibt gleich (DC treu).**
- ▶ **DC Anteil $\neq 0$: Filter skalierbar auf DC Anteil = 1**
- ▶ **Technisch: In Darstellung +128 (Grau) addieren.**



Faltungsoperationen

► Direkte Implementierung einer Faltung

```
void convolve (Image& dstImg, Image& srcImg, DoubleImage& filter)
{
    for (y=0; y<dstImg.height; y++)
        for (x=0; x<dstImg.width; x++) {
            double sum = 0;
            for (y2=0; y2<filter.height; y2++)
                for (x2=0; x2<filter.width; x2++) {
                    xSrc = x + x2 - filter.width/2;
                    ySrc = y + y2 - filter.height/2;
                    if (0<=xSrc && xSrc<srcImg.width &&
                        0<=ySrc && ySrc<srcImg.height)
                        sum += filter(x2, y2) * srcImg(xSrc, ySrc);
                    dstImg(x, y) = sum + filter.offset;
                }
        }
}
```

Faltungsoperationen

- ▶ Direkte Implementierung einer Faltung
- ▶ Frage an das Auditorium: Wie kann man die Routine beschleunigen? Wie für einen speziellen Filter?

```
void convolve (Image& dstImg, Image& srcImg, DoubleImage& filter)
{
    for (y=0; y<dstImg.height; y++)
        for (x=0; x<dstImg.width; x++) {
            double sum = 0;
            for (y2=0; y2<filter.height; y2++)
                for (x2=0; x2<filter.width; x2++) {
                    xSrc = x + x2 - filter.width/2;
                    ySrc = y + y2 - filter.height/2;
                    if (0<=xSrc && xSrc<srcImg.width &&
                        0<=ySrc && ySrc<srcImg.height)
                        sum += filter(x2, y2) * srcImg(xSrc, ySrc);
                    dstImg(x, y) = sum + filter.offset;
                }
        }
}
```


Faltungsoperationen

Beschleunigung der Faltung durch

▶ **Allgemeine Faltung**

- ▶ Zeigern statt Koordinaten
- ▶ Rand in äußeren 2 Schleifen berücksichtigen \Rightarrow innen kein Test
- ▶ rechnen mit `int`, Summe durch ggT teilen

▶ **Spezieller Filter**

- ▶ inneren 2 Schleifen durch Term ersetzen (!)
- ▶ 0 Koeffizienten auslassen
- ▶ gleiche Koeffizienten zusammenfassen
- ▶ Multiplikation / Division mit Zweierpotenzen durch `<<`, `>>` (autom.)
- ▶ konstanten Speicheroffset von 1 für `x` ausnutzen

▶ **Bei großen Filtern**

- ▶ Separieren (erst `x` dann `y` filtern)
- ▶ Fast Fourier Transformation



- ▶ **Frage an das Auditorium:
Was sind Merkmalen, die man in
Büroumgebungen, hier z.B. das MZH wohl mit
Bildverarbeitung erkennen könnte?**

Linien- und Kantendetektion

- ▶ **Helligkeitsschwellwert:**
 - ▶ unmöglich
- ▶ **Farbsegmentierung:**
 - ▶ rote Türen, blaue Türen, gelbe Schilder?
 - ▶ Schwierig wegen wechselnder Beleuchtung.
 - ▶ Vielleicht Farbe relativ zur Bodenfarbe bestimmen.
- ▶ **Kanten:**
 - ▶ Viele gerade Konturen in Büroumgebungen.
 - ▶ Oft senkrecht.
 - ▶ Ziemlich Beleuchtungsunabhängig.
 - ▶ Gutes Merkmal.
 - ▶ Aber Verwechslungsgefahr durch viele Kanten

Linien- und Kantendetektion

Linien- und Kantendetektion

▶ Bisher:

- ▶ Grauwert Schwellwert oder Farbsegmentierung
- ▶ Objekt muss eine durchgängige Region im Bild sein

▶ Probleme, wenn

- ▶ Verdeckung / Überlappung
- ▶ Innere Struktur (Muster, Aufdruck, ...)
- ▶ Reflexionen

▶ Lösung

- ▶ Merkmal des Randes nicht der Region
- ▶ geometrische Struktur toleriert Verdeckungen
- ▶ z.B. „Kanten sind gerade“



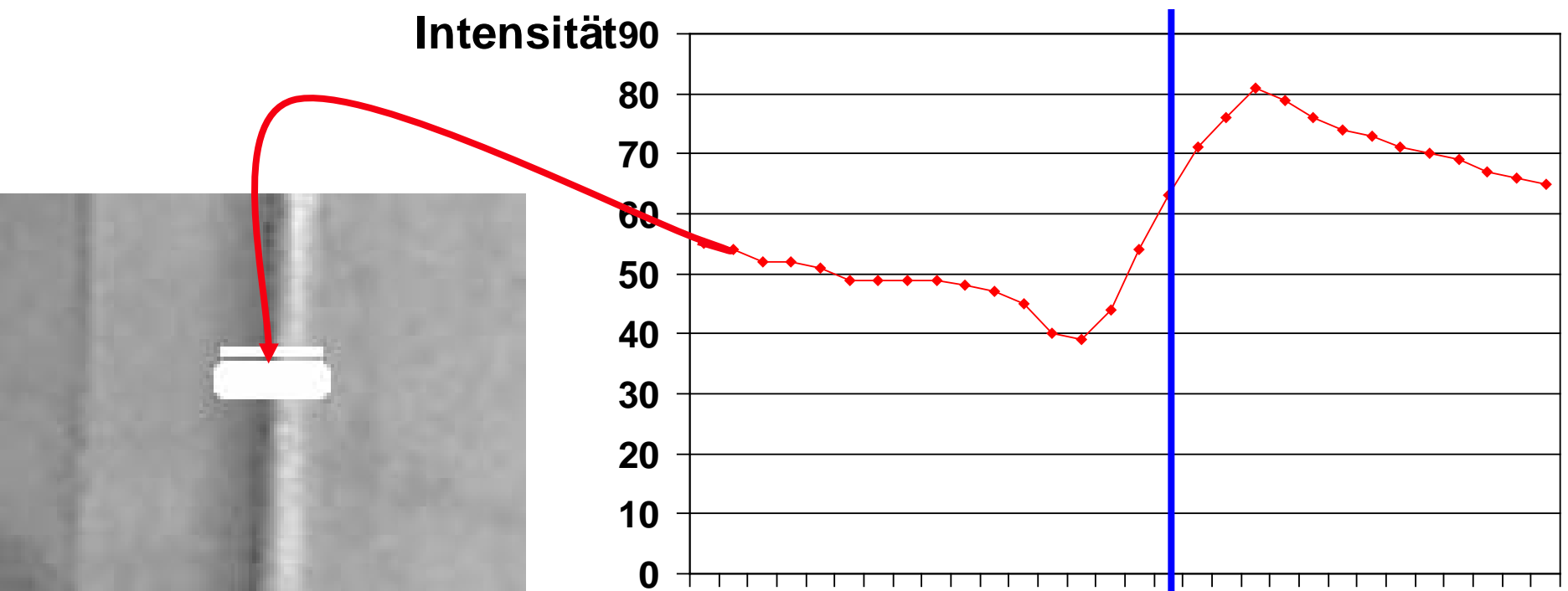


Linien- und Kantendetektion

- ▶ **Erster Schritt: Gehört ein Pixel zu einer Kante?**
 - ▶ Analog zu „gehört ein Pixel zum Objekt“ bei Schwellwert / Farbsegmentierung
 - ▶ Betrachte nicht nur Pixel alleine
 - ▶ Betrachte 3×3 Umgebung um Pixel
 - ▶ Harte Entscheidung Kante / Nicht-Kante vermeiden
 - ▶ Graduelles Ergebnis
- ▶ **Zweiter Schritt: Fasse die Kantenpunkte zu Kurven zusammen**
 - ▶ Hier: zu Geraden
 - ▶ Festes Modell dadurch Toleranz gegenüber Verdeckung
 - ▶ Geraden statt Strecken, dadurch nicht Bestimmung der Endpunkte

Linien- und Kantendetektion

Horizontaler Schnitt durch eine vertikale Kante



Idee: Eine Kante ist dort, wo sich die Intensität schnell ändert.

Linien- und Kantendetektion

Sobel-Filter

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

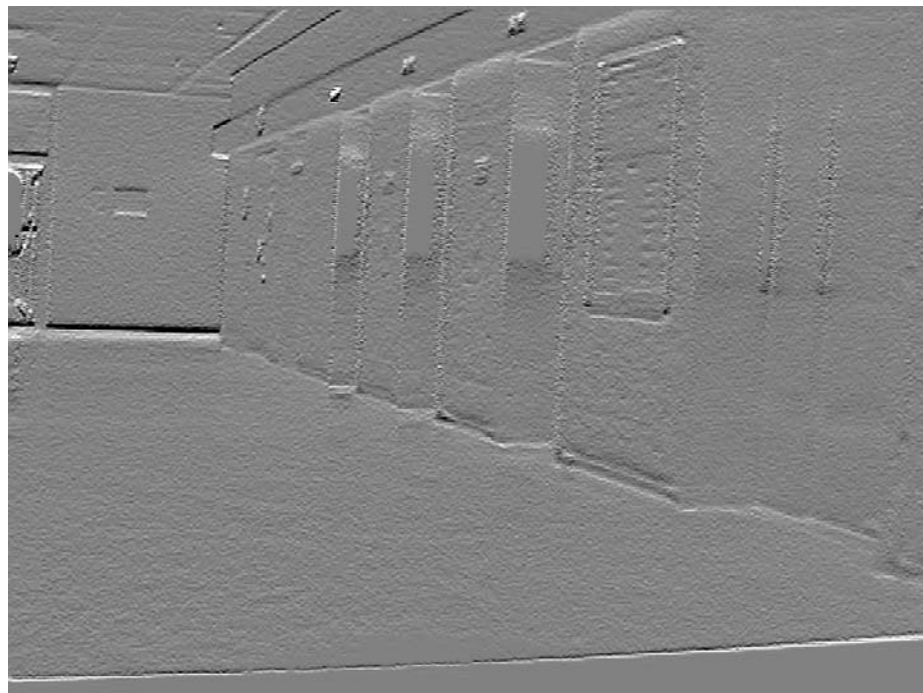
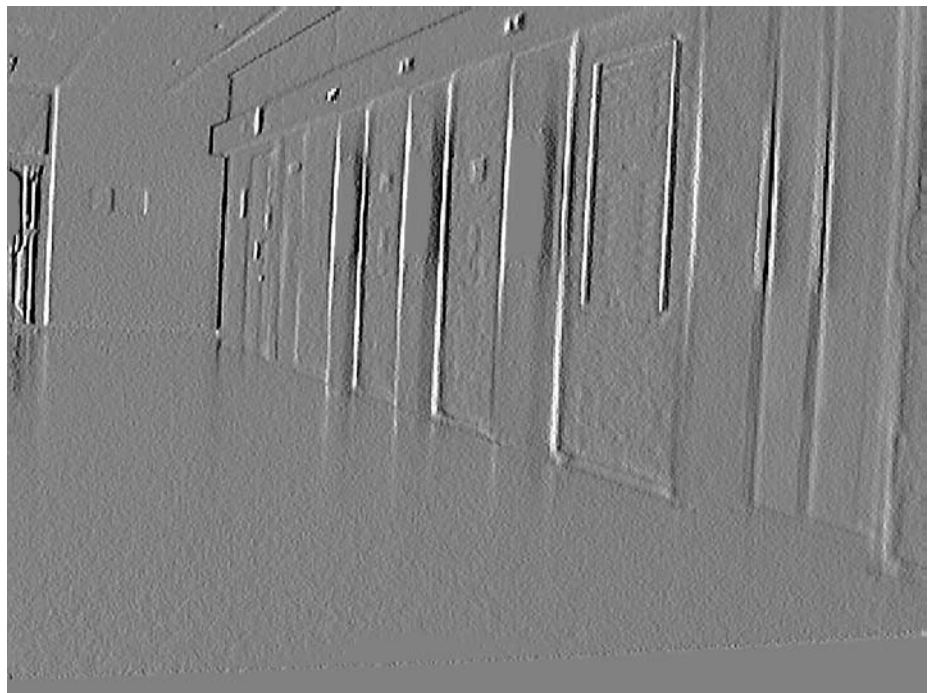
Sobel X
(vertikale Kanten)

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

Sobel Y
(horizontale Kanten)

- ▶ Kombination aus Differenz (quer) und Mittelwert (längs)
- ▶ Auch mit Faktor 1/8 üblich.
- ▶ Es gibt noch weitere ähnliche Filter.

Linien- und Kantendetektion



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

Sobel X
(vertikale Kanten)

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

Sobel Y
(horizontale Kanten)

Linien- und Kantendetektion

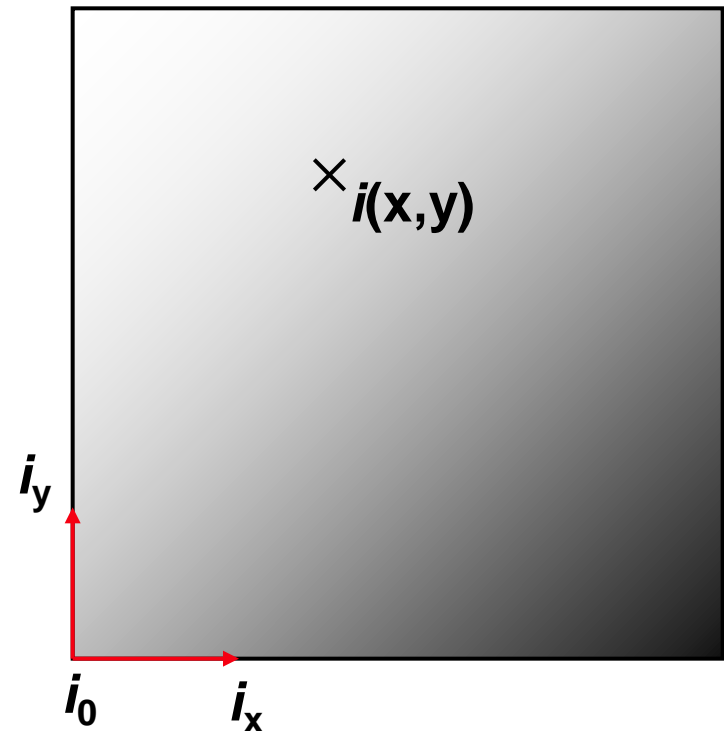
Was sagt der Sobel-Filter?

- ▶ Angenommen, wir haben ein Bild mit einem linearen Helligkeitsverlauf, was liefert der Sobel Filter?

- ▶ Helligkeit des Pixels x, y ist

$$i(x, y) = i_0 + i_x x + i_y y$$

- ▶ i_0 : Grundintensität am Pixel 0,0
- ▶ i_x : Änderung in X Richtung
- ▶ i_y : Änderung in Y-Richtung



Linien- und Kantendetektion

- ▶ Sobel-X (s_x) / Sobel Y (s_y) bestimmen für die Änderung (Gradient) der Helligkeit in X und Y Richtung

$$i(x, y) = i_0 + i_x x + i_y y$$

$$\begin{aligned}
 s_x &= -1(i_0 + i_x(x-1) + i_y(y-1)) + 1(i_0 + i_x(x+1) + i_y(y-1)) \\
 &\quad + -2(i_0 + i_x(x-1) + i_y(y)) + 2(i_0 + i_x(x+1) + i_y(y)) \\
 &\quad + -1(i_0 + i_x(x-1) + i_y(y+1)) + 1(i_0 + i_x(x+1) + i_y(y+1)) \\
 &= i_x + i_x + 2i_x + 2i_x + i_x + i_x \\
 &= 8i_x
 \end{aligned}$$

$$s_y = 8i_y$$

- ▶ Sobel X / Y liefert den Gradienten in X / Y-Richtung

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

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

Linien- und Kantendetektion

▶ Was passiert mit schrägen Kanten?

- ▶ Sowohl Sobel X als auch Sobel Y spricht an.
- ▶ Je horizontaler die Kante, je mehr Sobel Y
- ▶ Je vertikaler die Kante, je mehr Sobel X

▶ Sobel X (s_x) und Sobel Y (s_y) als Vektor zeigen die Richtung einer schrägen Kante an.

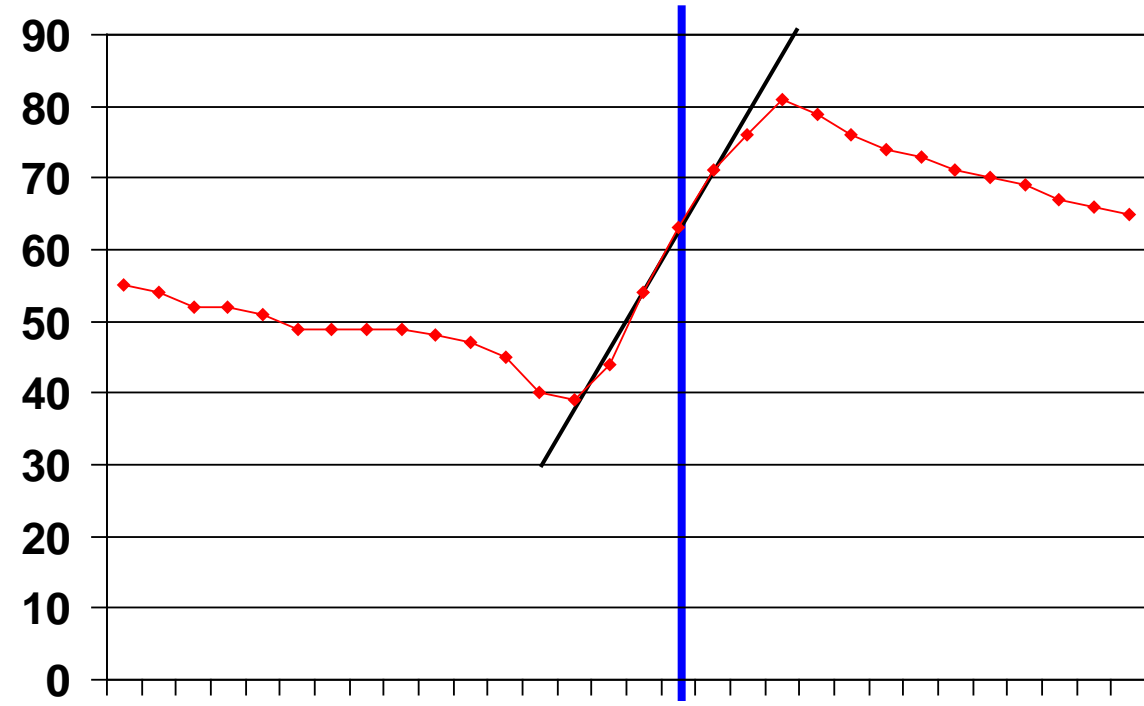
- ▶ Betrag des Vektors gibt Intensität der Kante an
- ▶ Richtung des Vektors gibt Winkel quer zur Kante an (von dunkel nach hell)

$$\sqrt{s_x^2 + s_y^2}$$
$$\arctan 2(s_y, s_x)$$

▶ Im folgenden Herleitung und Erklärung

Linien- und Kantendetektion

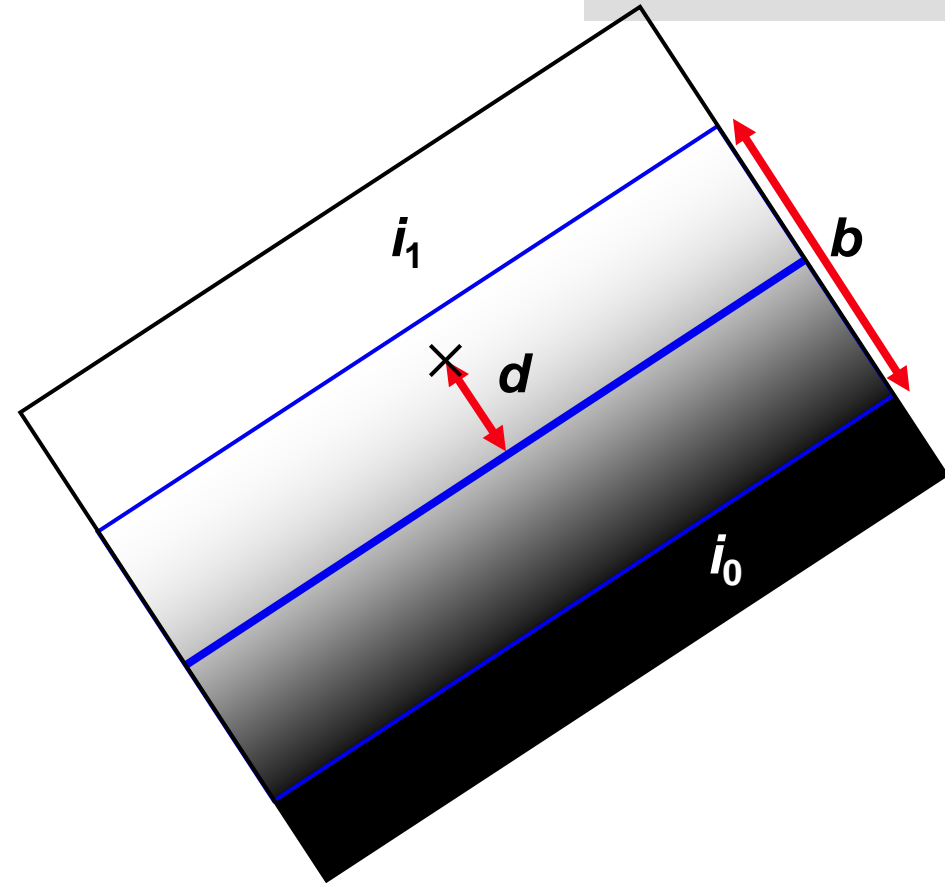
- ▶ **Beobachtung: An einer Kante ist ein (kleiner) Streifen linearen Helligkeitswechsels**



Linien- und Kantendetektion

- ▶ **Ideale Linie (dick blau) ist umgeben von einem Bereich linearen Helligkeitsverlaufs (dünn blau)**
- ▶ **Helligkeitsverlauf von i_0 nach i_1 hat Breite b**
- ▶ **Helligkeit eines Punktes mit vorzeichenbehaftetem Abstand d zur Kante ist**

$$\frac{i_0 + i_1}{2} + \frac{d}{b}(i_1 - i_0)$$



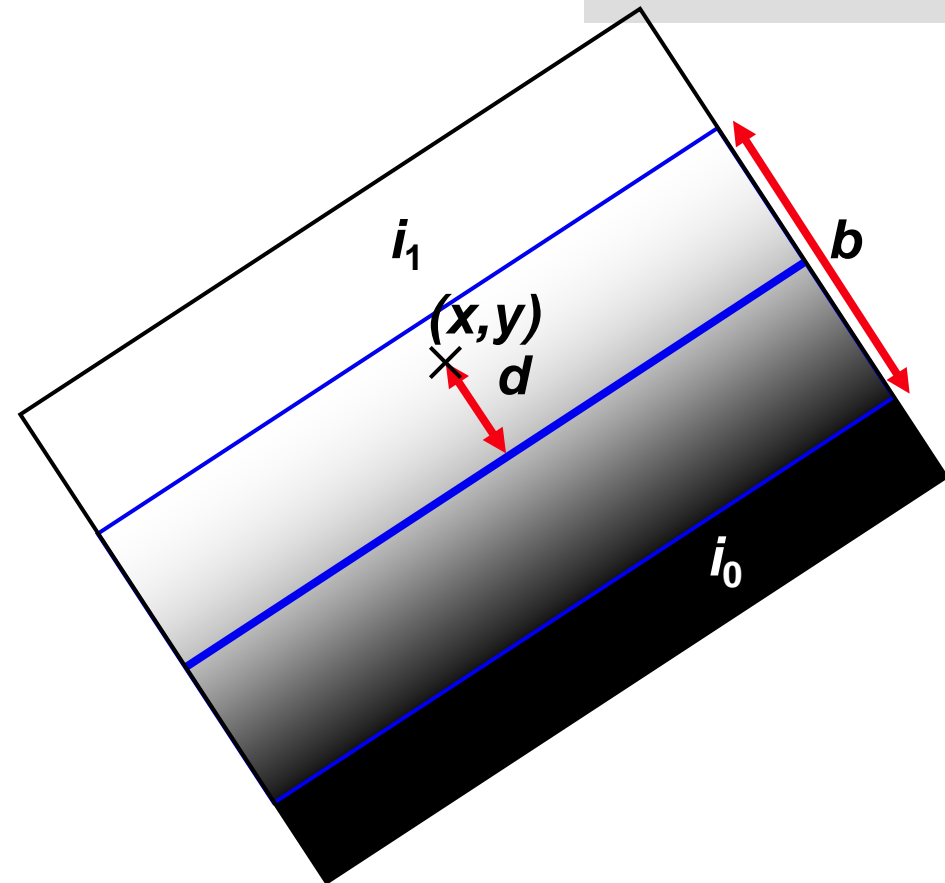
Linien- und Kantendetektion

- ▶ Abstand d ergibt sich aus Koordinaten (x,y) und Hessescher Normalform (p Abstand Kante zum Ursprung, α Richtung Normalenvektor auf i_1)

$$d = p + \cos \alpha \cdot x + \sin \alpha \cdot y$$

$$i = \frac{i_0 + i_1}{2} +$$

$$\frac{p + \cos \alpha \cdot x + \sin \alpha \cdot y}{b} (i_1 - i_0)$$



Linien- und Kantendetektion

$$i = \frac{i_0 + i_1}{2} + \frac{p + \cos \alpha \cdot x + \sin \alpha \cdot y}{b} (i_1 - i_0)$$

- ▶ Anwendung des vorherigen Ergebnisses ergibt

$$s_x = 8 \cos \alpha \frac{i_1 - i_0}{b} \quad s_y = 8 \sin \alpha \frac{i_1 - i_0}{b}$$

$$\sqrt{s_x^2 + s_y^2} = 8 \frac{i_1 - i_0}{b}, \quad \arctan 2(s_y, s_x) = \alpha$$

- ▶ Die Sobellänge liefert richtungsunabhängig die Intensität der Kante
- ▶ Die Sobelrichtung liefert die Richtung senkrecht zur Kante

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

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

Linien- und Kantenerkennung

- ▶ Betrag des Sobelvektors

$$\sqrt{s_x^2 + s_y^2} = 8 \frac{i_1 - i_0}{b}$$

- ▶ zeigt Kanten in alle Richtungen



Linien- und Kantendetektion

- ▶ **Sobel Operator findet Kanten / Linien**
- ▶ **Sobel X (s_x) und Sobel Y (s_y) als Vektor geben den Helligkeitsgradienten an**
 - ▶ Betrag des Vektors ist Maß für Stärke der Kante
 - ▶ Vorhandensein einer Kante mit Schwellwert
 - ▶ Richtung des Vektors gibt Richtung der Kante an
 - ▶ senkrecht zur Kante zur helleren Seite
- ▶ **Linien führen zu zwei hohen Sobelwerten mit umgekehrten Vorzeichen**
 - ▶ Daher (s_x, s_y) und $(-s_x, -s_y)$ gleich behandeln
 - ▶ Kantenorientierung: 360° , Linienorientierung 180°

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

Sobel X
(vertikale
Kanten)

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

Sobel Y
(horizontale
Kanten)

Zusammenfassung

▶ Faltungsoperationen

- ▶ Lineare, translationsinvariante Abbildung
- ▶ Ergebnispixel ist gewichtete Summe der Pixel in der Umgebung des Eingangspixels
- ▶ Bilder glätten, Kontrast vergrößern, Kanten detektieren

▶ Kantendetektion mit dem Sobel Filter (SobelX, SobelY)

- ▶ SobelX und SobelY geben als Ergebnis einen Vektor
- ▶ Betrag: Kantenstärke
- ▶ Richtung: Richtung der Kante (senkrecht zum Helleren)

▶ Kontrastnormalisierter Kantenfiter

- ▶ Sobel Vektor durch Kontrast dividiert
- ▶ Beleuchtungsinvariant
- ▶ Betrag²: Welcher Anteil des Bildkontrastes ist linearer Gradient
- ▶ Gradienten*reinheit* nicht -*stärke*

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

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