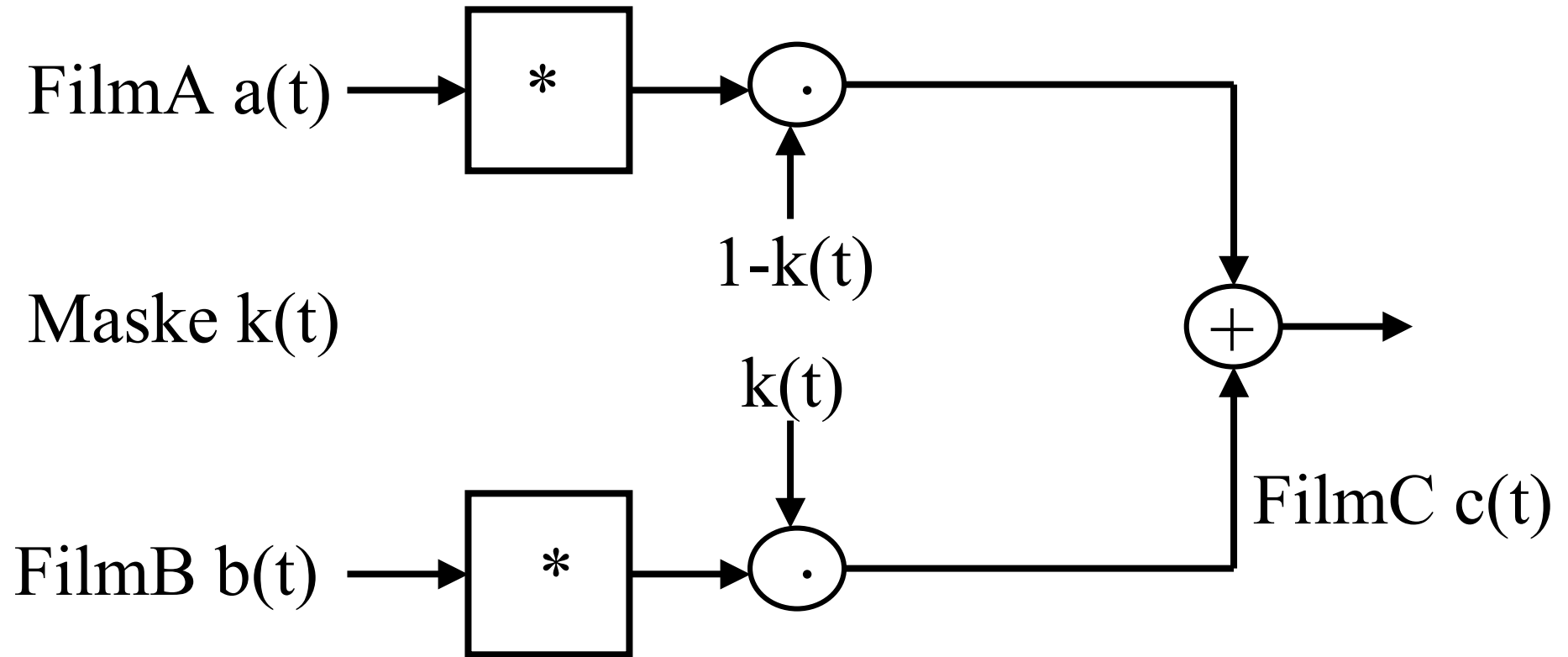


---

# Bildverarbeitung 1

Dipl.-Ing. Guido Heising

# Überblendung



$$c(t, m, n) = k(t, m, n) \cdot b(t, m, n) + [1 - k(t, m, n)] \cdot a(t, m, n)$$

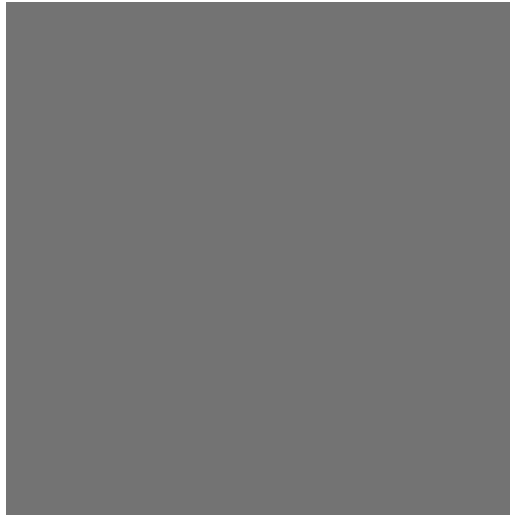
\*) = optionale Filterung, Transformation, etc.

# Überblendungen

---



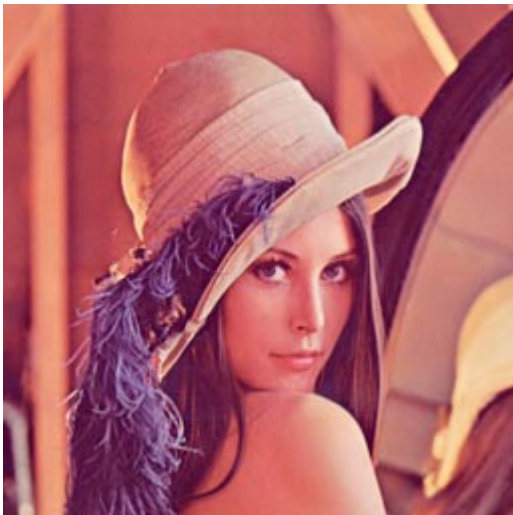
•



=



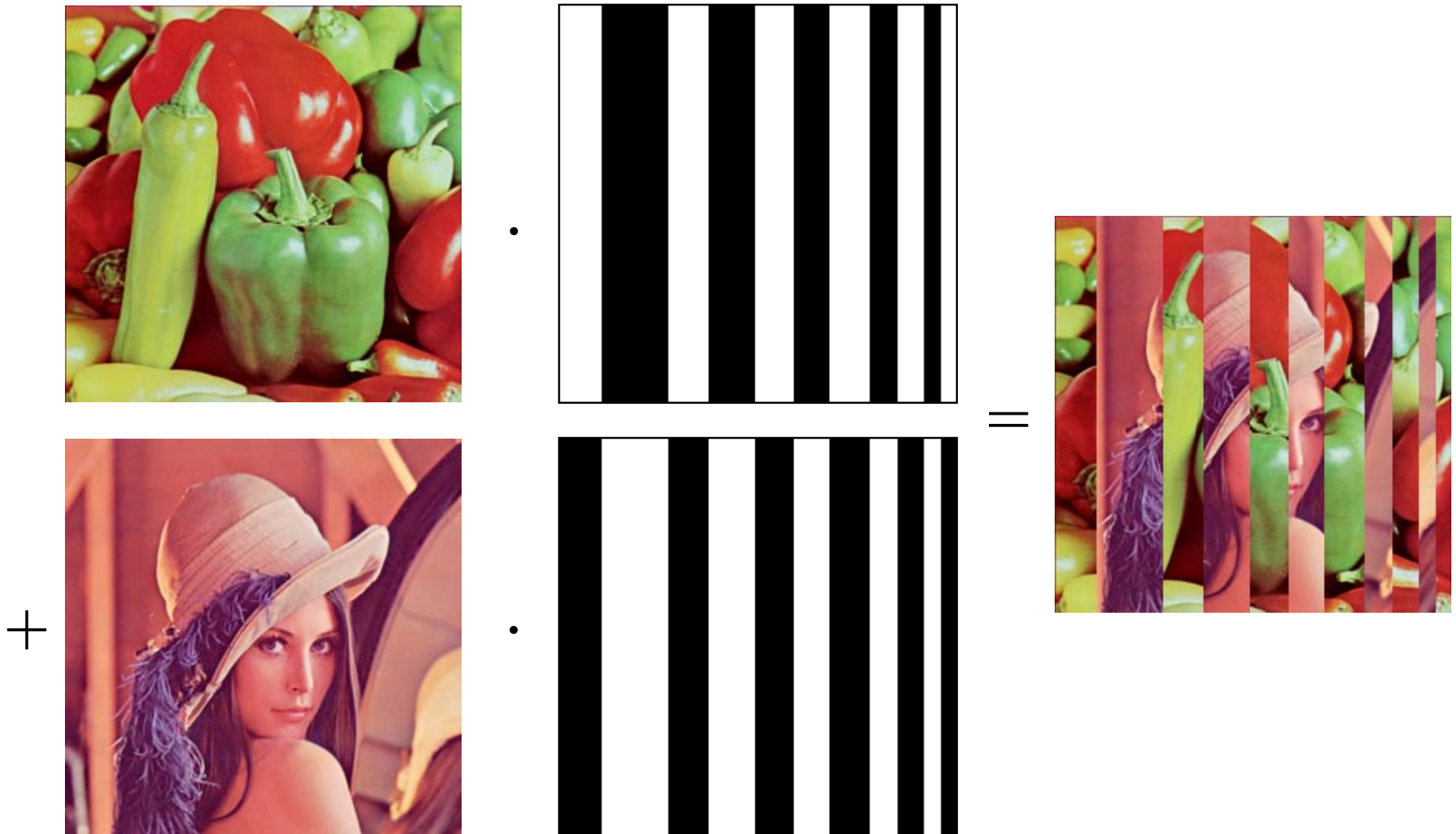
+



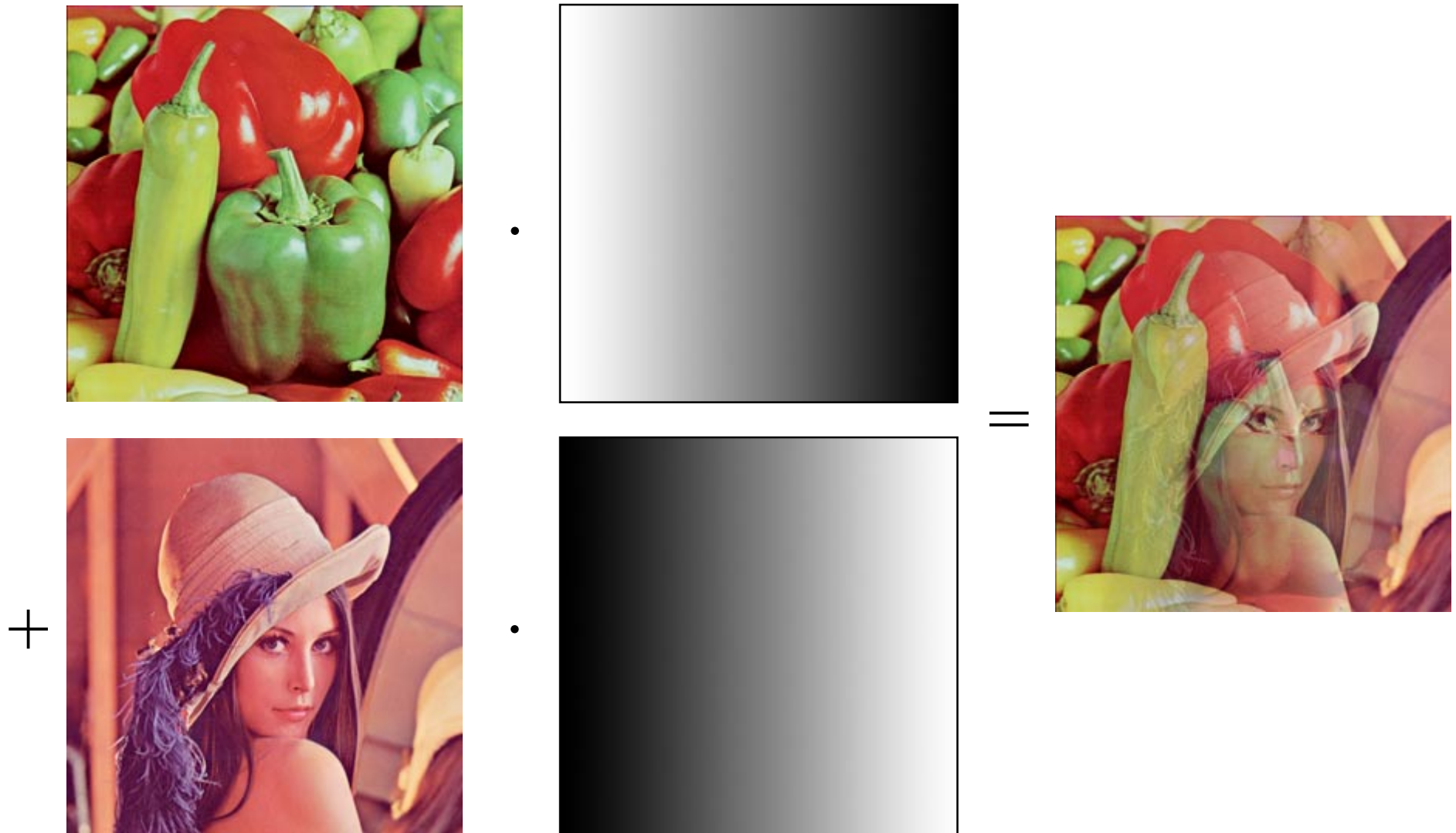
•



# Überblendungen



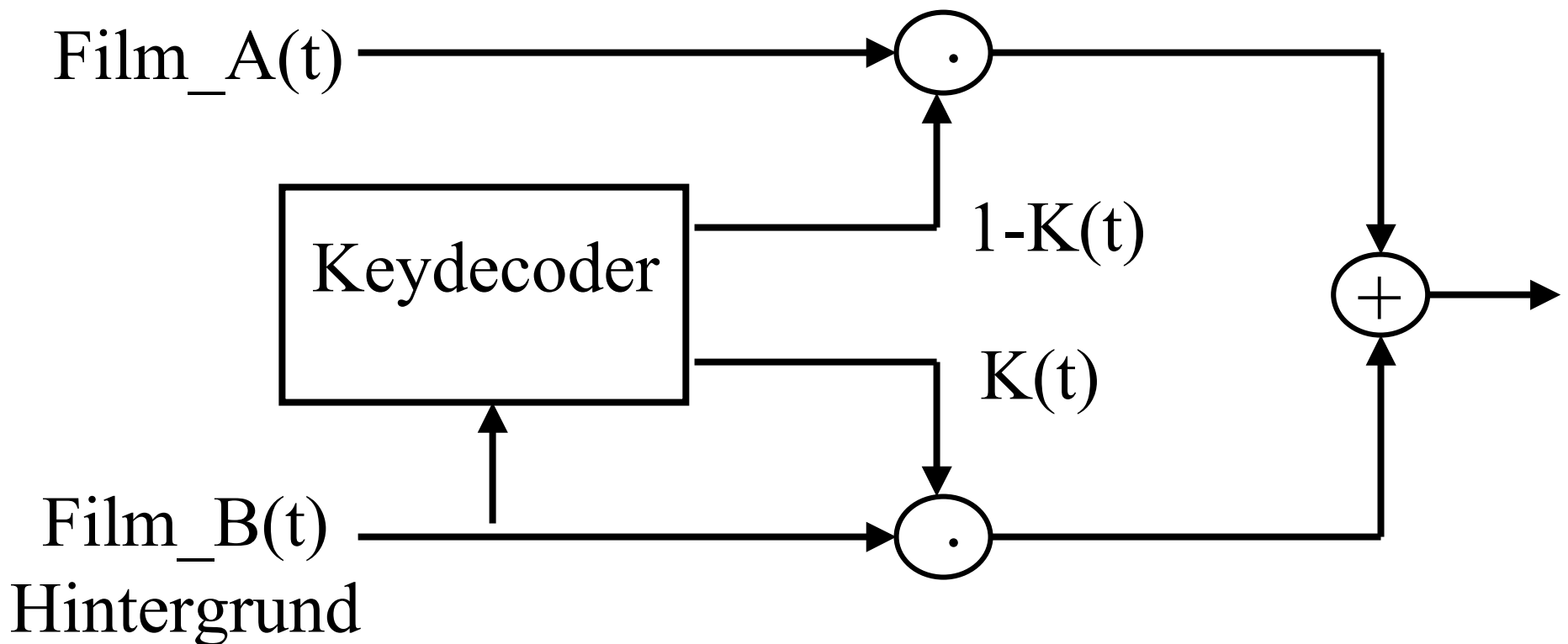
# Überblendungen



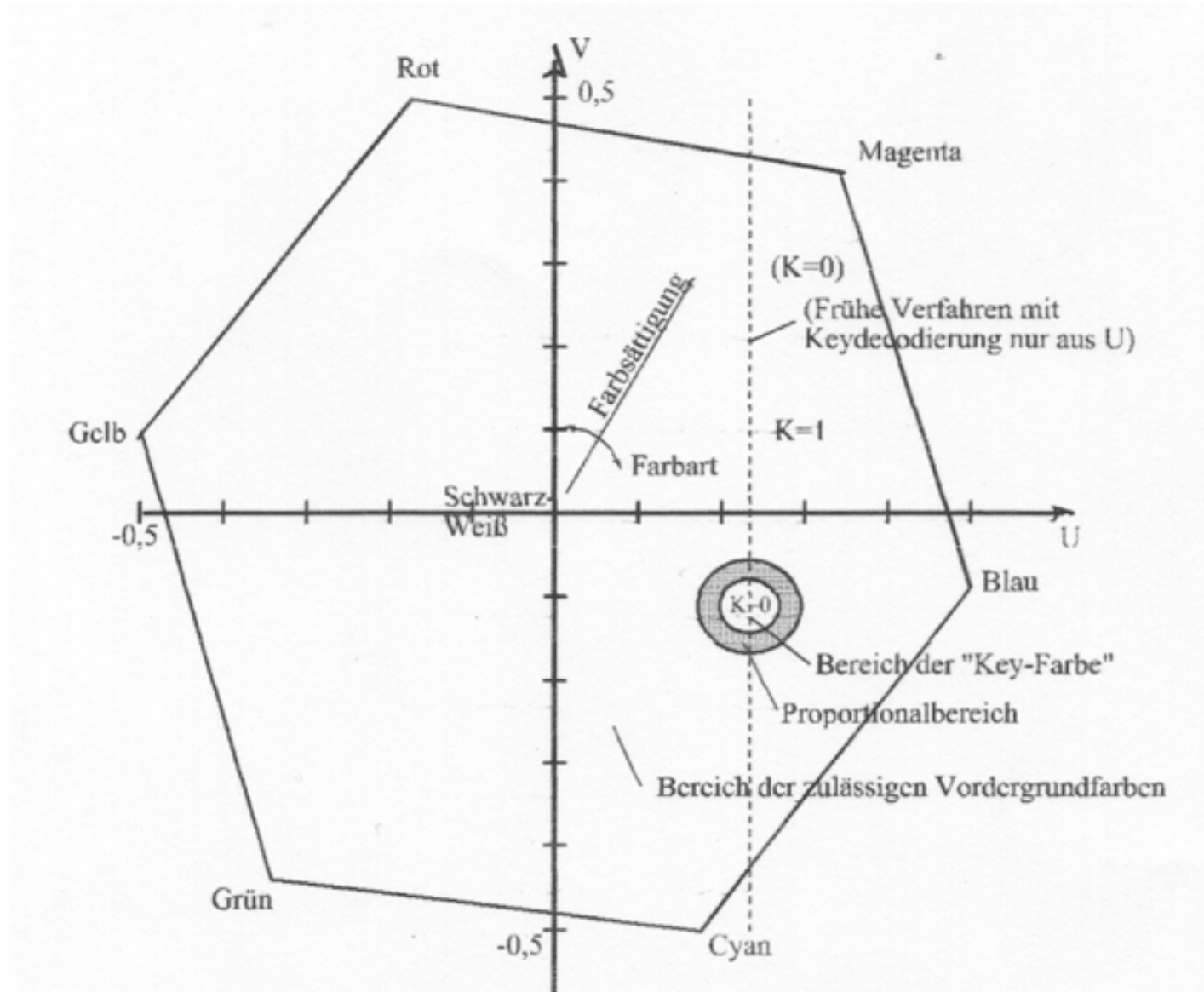
# Keying

---

Bildoperation ist abhängig vom Eingangsbild



# Keying (Blue screen)



# Keying



•



=

+



•



# Keying

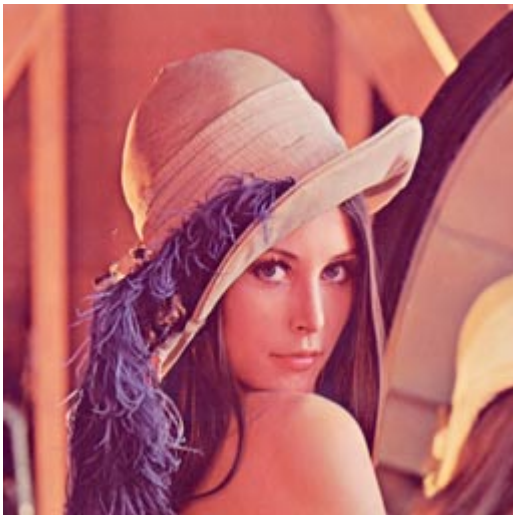


•

=



+



•

# Bildüberlagerung

---

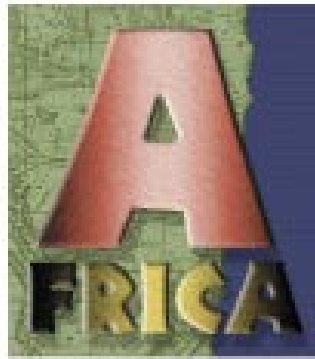


# Bildüberlagerung

---



Original



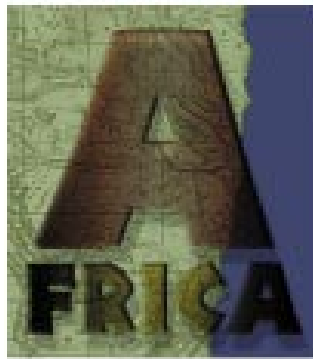
Normal

Keying

$$c(m, n) = k(m, n) \cdot b(m, n) + [1 - k(m, n)] \cdot a(m, n)$$



Original



Multiply

Multiplikation

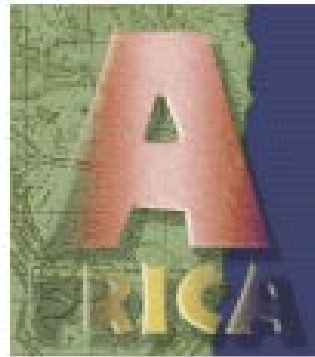
$$c(m, n) = b(m, n) \cdot a(m, n) / 255 \quad (\text{nach Keying})$$

# Bildüberlagerung

---



Original



Lighten

Maximum

$$c(m, n) = \max(b(m, n), a(m, n))$$



Original



Darken

Minimum

$$c(m, n) = \min(b(m, n), a(m, n))$$

# Bildüberlagerung

---



Original



Color

## Farbigkeit

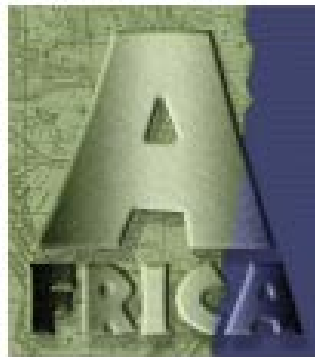
$$c_Y(m, n) = a_Y(m, n)$$

$$c_U(m, n) = b_U(m, n)$$

$$c_V(m, n) = b_V(m, n)$$



Original



Luminosity

## Luminanz

$$c_Y(m, n) = b_Y(m, n)$$

$$c_U(m, n) = a_U(m, n)$$

$$c_V(m, n) = a_V(m, n)$$

# Bildmanipulation

---



- Bildpunktoperationen
- Nachbarschaftsoperationen
- geometrische Transformationen

# Bildpunktoperatoren

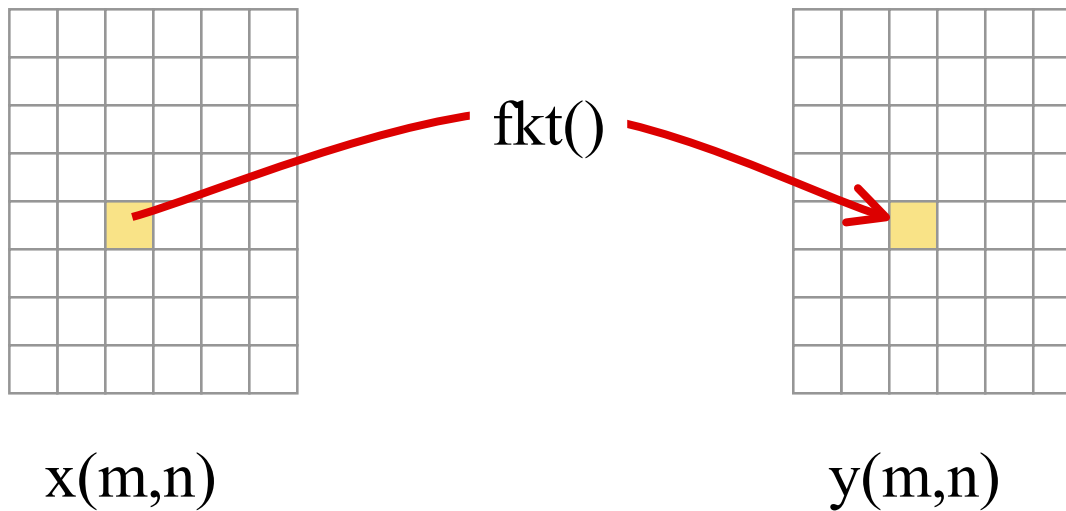
---

- Amplitudentransformationen  
 $y(m, n) = fkt(x(m, n))$
- Helligkeits- und Kontraständerung
- Histogrammanpassungen
- Gammakorrektur
- Farbkorrekturen
- Quantisierung

# Bildpunktoperatoren

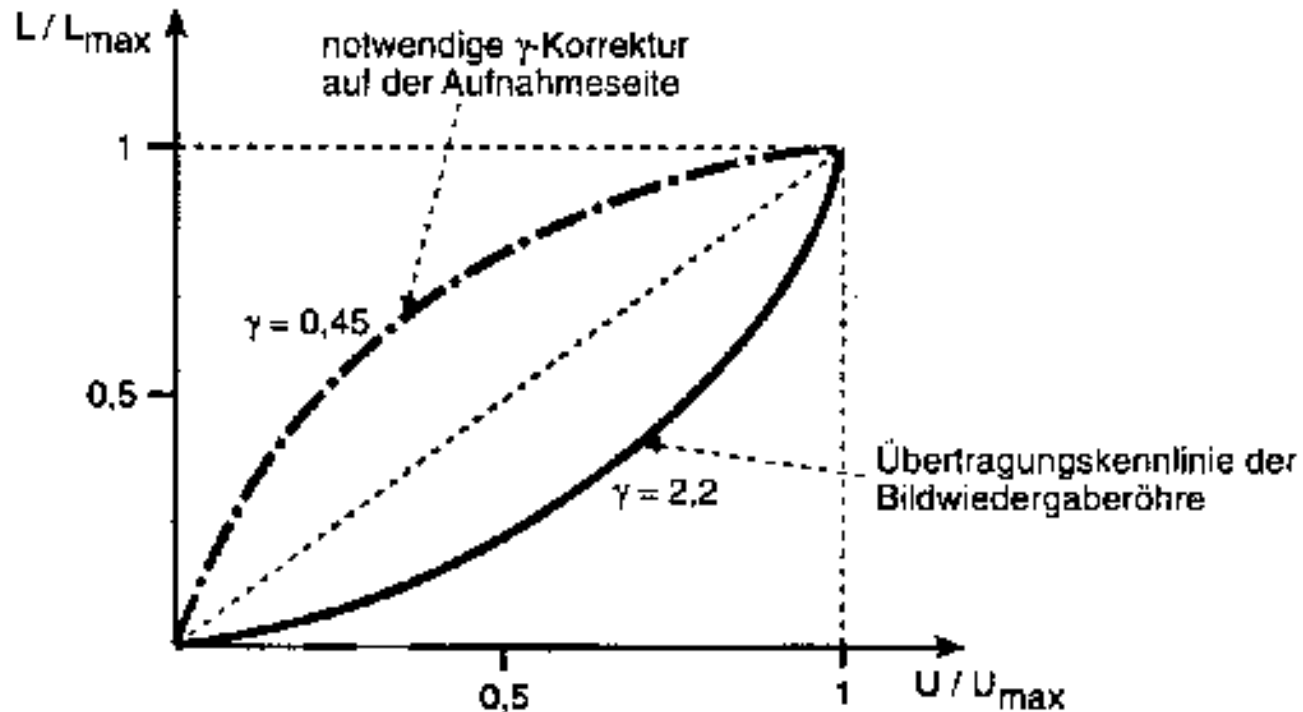
---

$$y(m, n) = fkt(x(m, n))$$





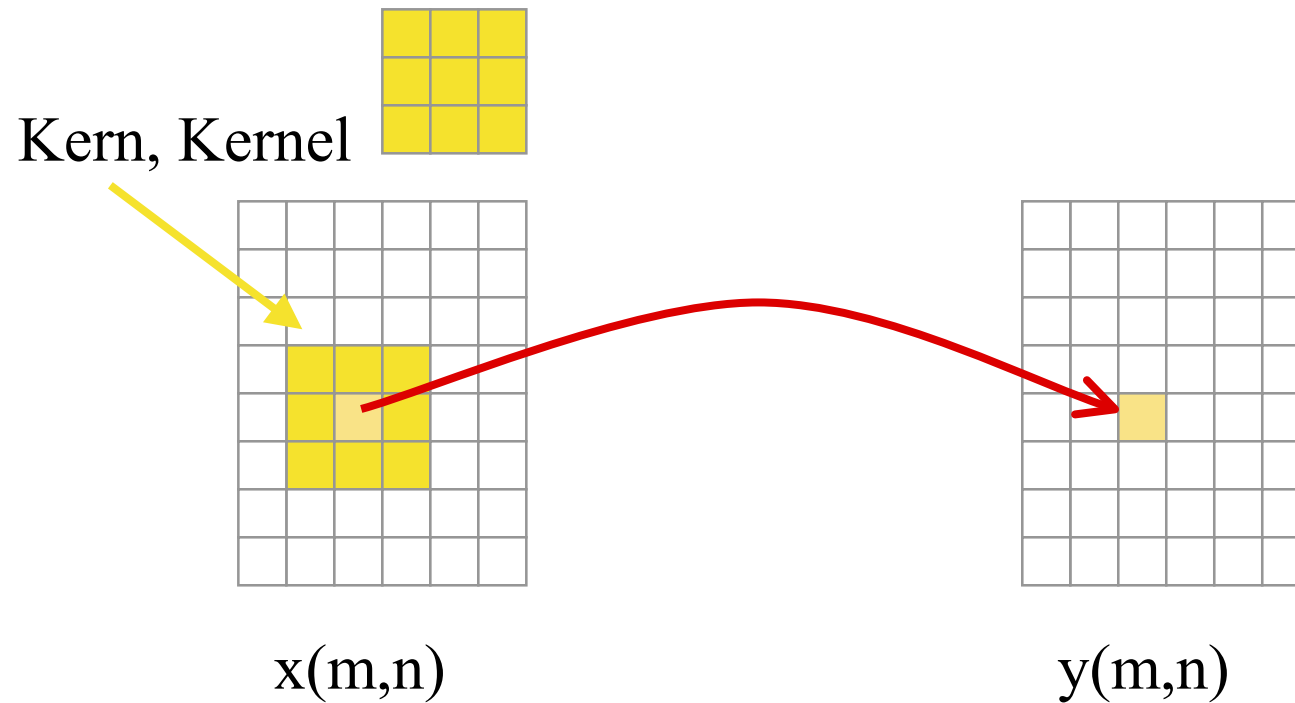
# Gammakorrektur



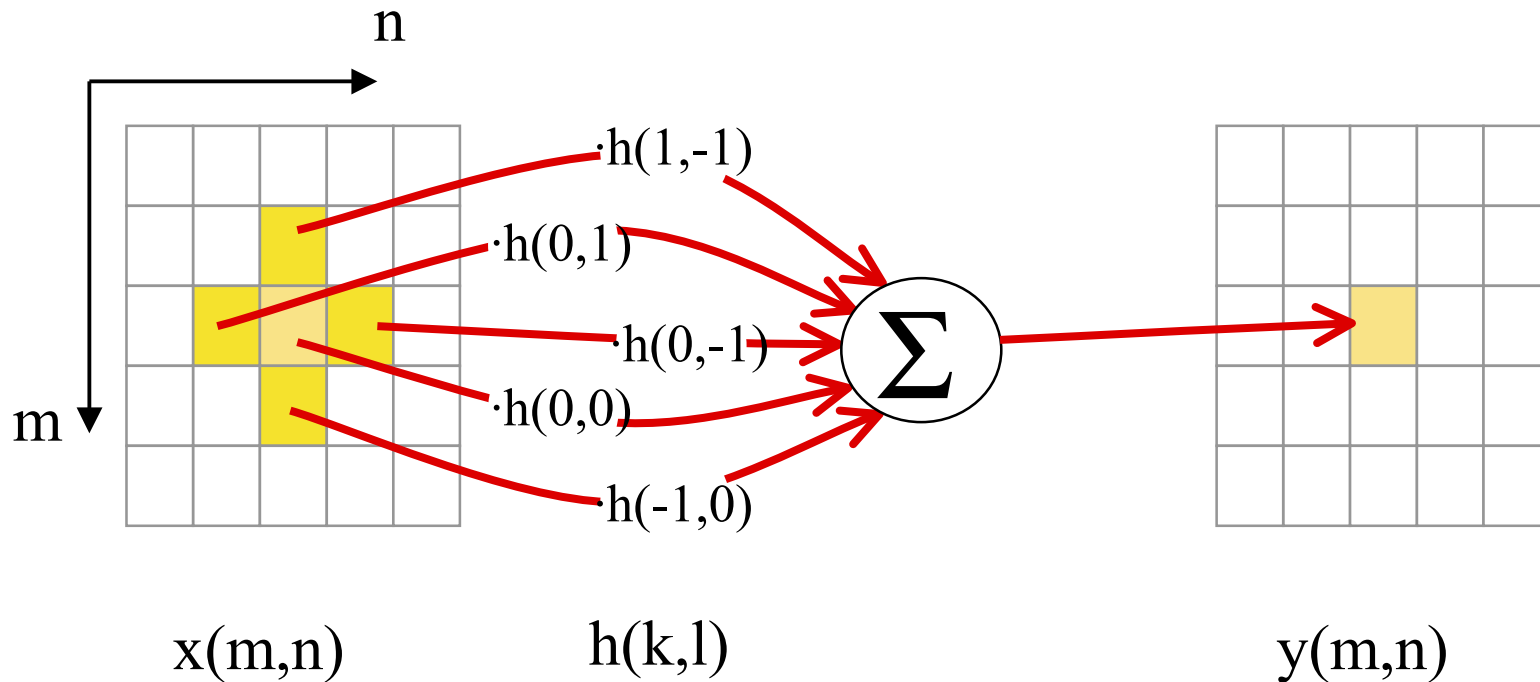
Anpassung der Luminanz  $L$  des Bildsignals bei der Bildaufnahme an die nichtlineare Wiedergabecharakteristik der Bildröhre

# Nachbarschaftsoperatoren

---



# Nachbarschaftsoperatoren



Lineare 2 - dimensionale Filterung :

$$y(m,n) = h(m,n) ** x(m,n) = \sum_{k=-P_m/2}^{P_m/2} \sum_{l=-P_n/2}^{P_n/2} h(k,l) \cdot x(m-k, n-l)$$

mit Faltungsoperator  $**$  und Filterordnung  $P_m$  und  $P_n$

# Nachbarschaftsoperatoren, Faltung

Mit Filterordnung  $P_m = 2$  und  $P_n = 2$ :

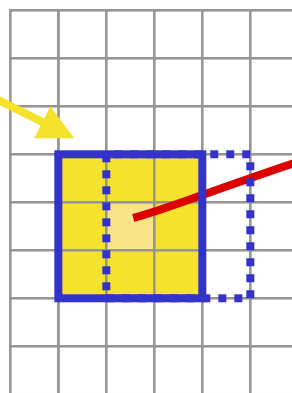
$$y(m,n) = \sum_{k=-1}^1 \sum_{l=-1}^1 h(k,l) \cdot x(m-k, n-l)$$

$$= h(1,1)x(m-1, n-1) + h(1,0)x(m-1, n) + h(1,-1)x(m-1, n+1)$$

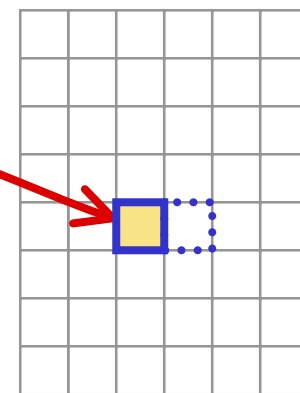
$$+ h(0,1)x(m, n-1) + h(0,0)x(m, n) + h(0,-1)x(m, n+1)$$

$$+ h(-1,1)x(m+1, n-1) + h(-1,0)x(m+1, n) + h(-1,-1)x(m+1, n+1)$$

Kern, Kernel  
wird über jeden  
Bildpunkt gelegt



$x(m,n)$



$y(m,n)$

# Nachbarschaftsoperatoren, Faltung

---

- Lineare Verknüpfung  
Tiefpassfilter (Mittelwertbildung)  
Hochpassfilter (Differenzbildung)
- Nichtlineare Verknüpfung  
Morphologische Filter

Siehe z.B. Photoshop:

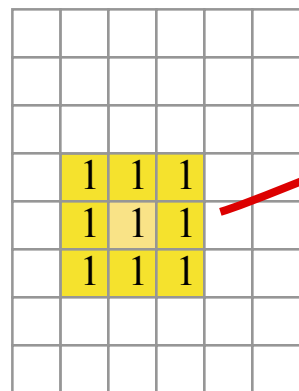
Lineare Filter: Filter-Other-Custom,

Median-Filter: Filter-Noise-Median (mit Radius=1 für 3x3 Maske, 2 für 5x5,...)

Max, Min-Filter: Filter-Other-Maximum/Minimum

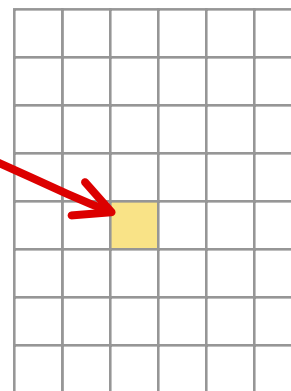
# Beispiele für Tiefpassfilter und Kantendetektion

Glättung durch Mittelwertbildung (Tiefpassfilter) :



$x(m,n)$

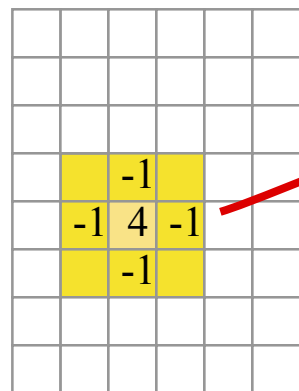
/9



$y(m,n)$

$$h_{TP}(k,l) = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

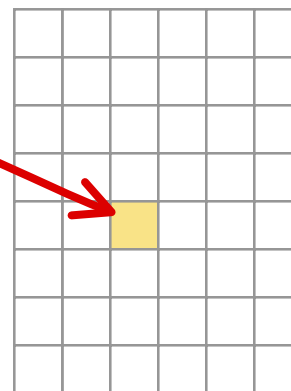
Kantendetektion durch Differenzbildung (Hochpassfilter):



$x(m,n)$

+128

Offset zur  
besseren  
Darstellung



$y(m,n)$

$$h_{HP}(k,l) = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

# Beispiele für Tiefpassfilter und Kantendetektion

---

Glättung durch Mittelwertbildung (Tiefpassfilter) :

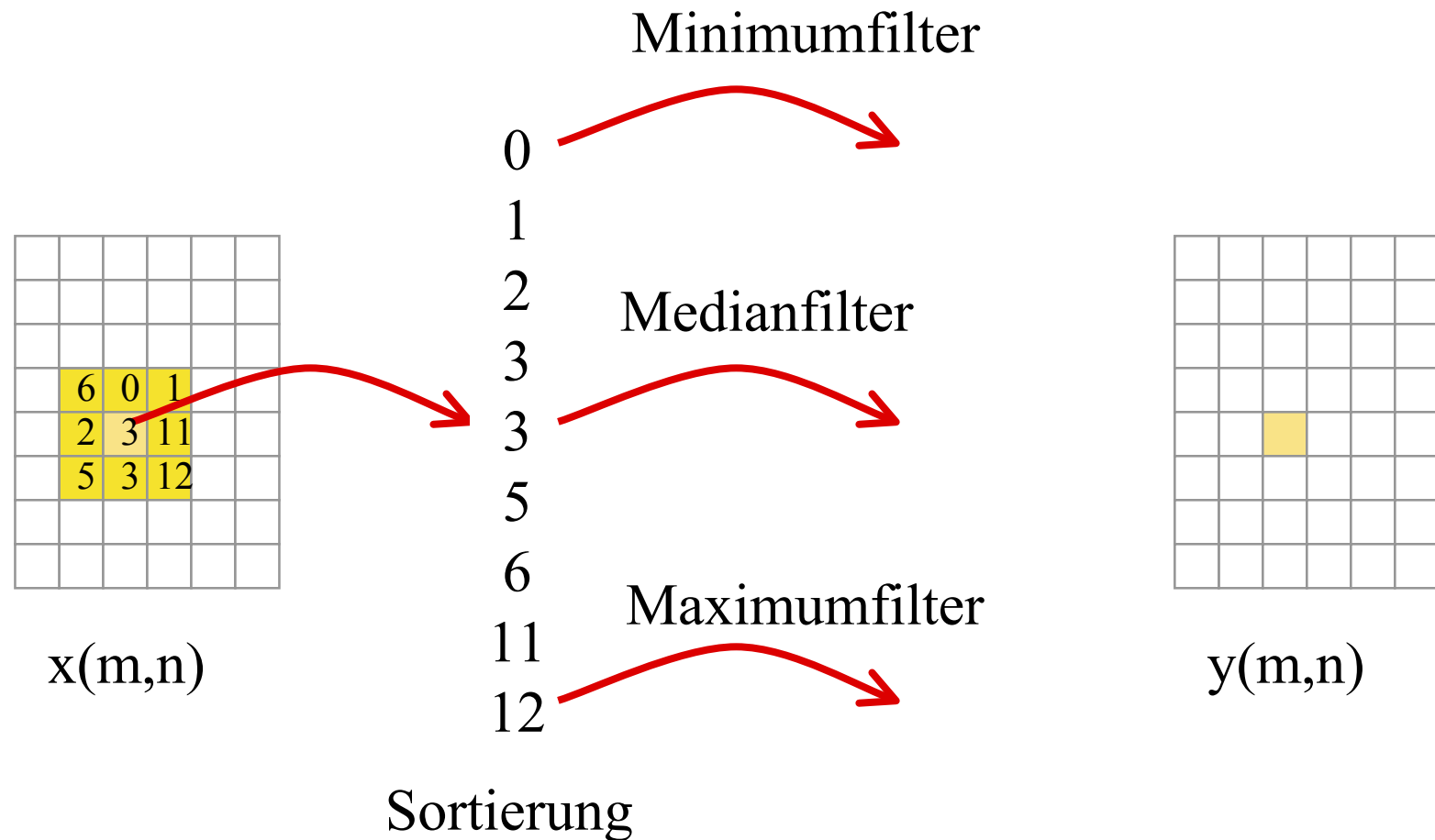
$$y(m,n) = x(m,n) ** h_{TP}(k,l) = \begin{pmatrix} 1 & 10 & 30 & 30 & 30 \\ 1 & 10 & 30 & 30 & 30 \\ 1 & 10 & 30 & 30 & 30 \\ 1 & 10 & 30 & 30 & 30 \\ 1 & 10 & 30 & 30 & 30 \end{pmatrix} ** \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} r & r & r & r & r \\ r & 14 & 23 & 30 & r \\ r & 14 & 23 & 30 & r \\ r & 14 & 23 & 30 & r \\ r & r & r & r & r \end{pmatrix}$$

Kantendetektion durch Differenzbildung (Hochpassfilter):

$$y(m,n) = x(m,n) ** h_{HP}(k,l) = \begin{pmatrix} 1 & 10 & 30 & 30 & 30 \\ 1 & 10 & 30 & 30 & 30 \\ 1 & 10 & 30 & 30 & 30 \\ 1 & 10 & 30 & 30 & 30 \\ 1 & 10 & 30 & 30 & 30 \end{pmatrix} ** \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} = \begin{pmatrix} r & r & r & r & r \\ r & -11 & 20 & 0 & r \\ r & -11 & 20 & 0 & r \\ r & -11 & 20 & 0 & r \\ r & r & r & r & r \end{pmatrix}$$

(Bildrandpunkte  $r$  erfordern spezielle Randberechnung)

# Morphologische Filter (Rangordnungsfilter)



Der Medianfilter eliminiert Störungen geringer Ausdehnung im Bild (z.B. Rauschen) ohne die Schärfe zu sehr zu beeinträchtigen.



# Morphologische Filter (Rangordnungsfiler)

---

Median-Filter der Größe 3x3:

$$y(m, n) = \text{median}_{3 \times 3}(x(m, n)) = \text{median}_{3 \times 3} \begin{pmatrix} 1 & 10 & 30 & 30 & 30 \\ 1 & 10 & 30 & 30 & 30 \\ 1 & 10 & 30 & 30 & 30 \\ 1 & 10 & 30 & 30 & 30 \\ 1 & 10 & 30 & 30 & 30 \end{pmatrix} = \begin{pmatrix} r & r & r & r & r \\ r & 10 & 30 & 30 & r \\ r & 10 & 30 & 30 & r \\ r & 10 & 30 & 30 & r \\ r & r & r & r & r \end{pmatrix}$$

(Bildrandpunkte  $r$  erfordern spezielle Randberechnung)

Median-Filterung eines gestörten Bildes

$$y(m, n) = \text{median}_{3 \times 3}(x(m, n)) = \text{median}_{3 \times 3} \begin{pmatrix} 1 & 10 & 30 & 30 & 30 \\ 1 & 10 & 30 & 30 & 30 \\ 1 & 10 & 30 & 30 & 30 \\ 1 & 10 & \mathbf{100} & 30 & 30 \\ 1 & 10 & 30 & 30 & 30 \end{pmatrix} = \begin{pmatrix} r & r & r & r & r \\ r & 10 & 30 & 30 & r \\ r & 10 & 30 & 30 & r \\ r & 10 & 30 & 30 & r \\ r & r & r & r & r \end{pmatrix}$$

Aufgabe: Welches Bild ergibt sich durch die Tiefpassfilterung des gestörten Bildes