# Messen optischer Größen, Messen aus Bildern
# Übersicht

## Optische Strahlung, Sensorik

## Geometrie, Photogrammetrie

- Kamerakalibrierung
- Stereo

## Menschliche Wahrnehmung

- Neurophysiologie
- Kognitive Psychologie

## Digitale Bildanalyse

- Digitales Rasterbild, Kenngrößen
- Bildverarbeitungsoperationen, Segmentation
- Salient point detection + description

# Bildverarbeitungsoperationen

Eingabebild *E(x,y)*, Ausgabebild *A(x,y)*

Funktion *f: E(x,y) → A(x,y)*

Bildkoordinaten *(x,y) … „Ortsbereich“*

Transformation *T: E(x,y) → F(u,v)*

*T* transformiert in einen anderen „Bereich“

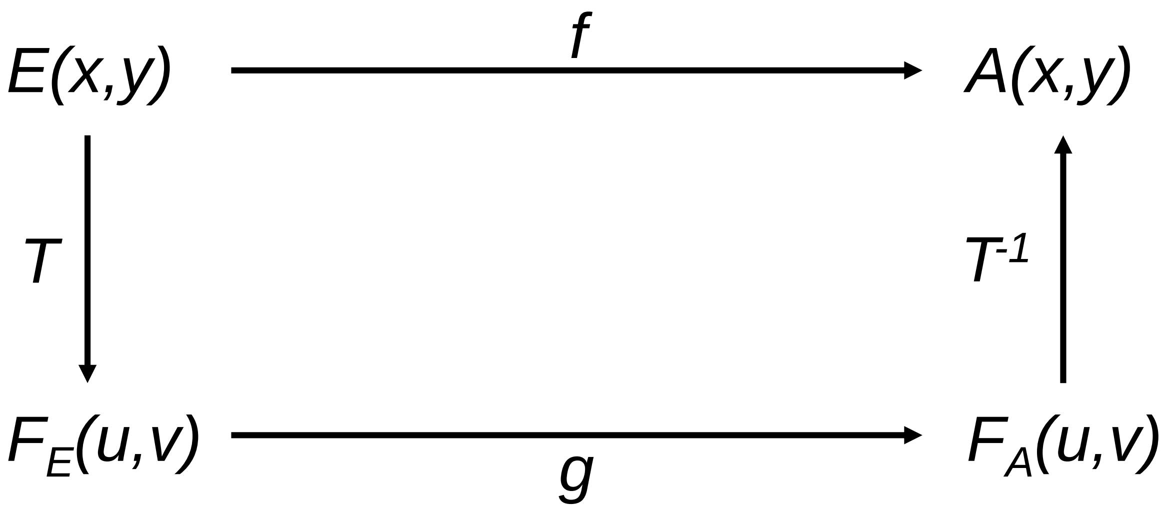
z.B.: *„Frequenzbereich“ (u,v)* bei der Fourier-Transformation

Wozu Transformationen?

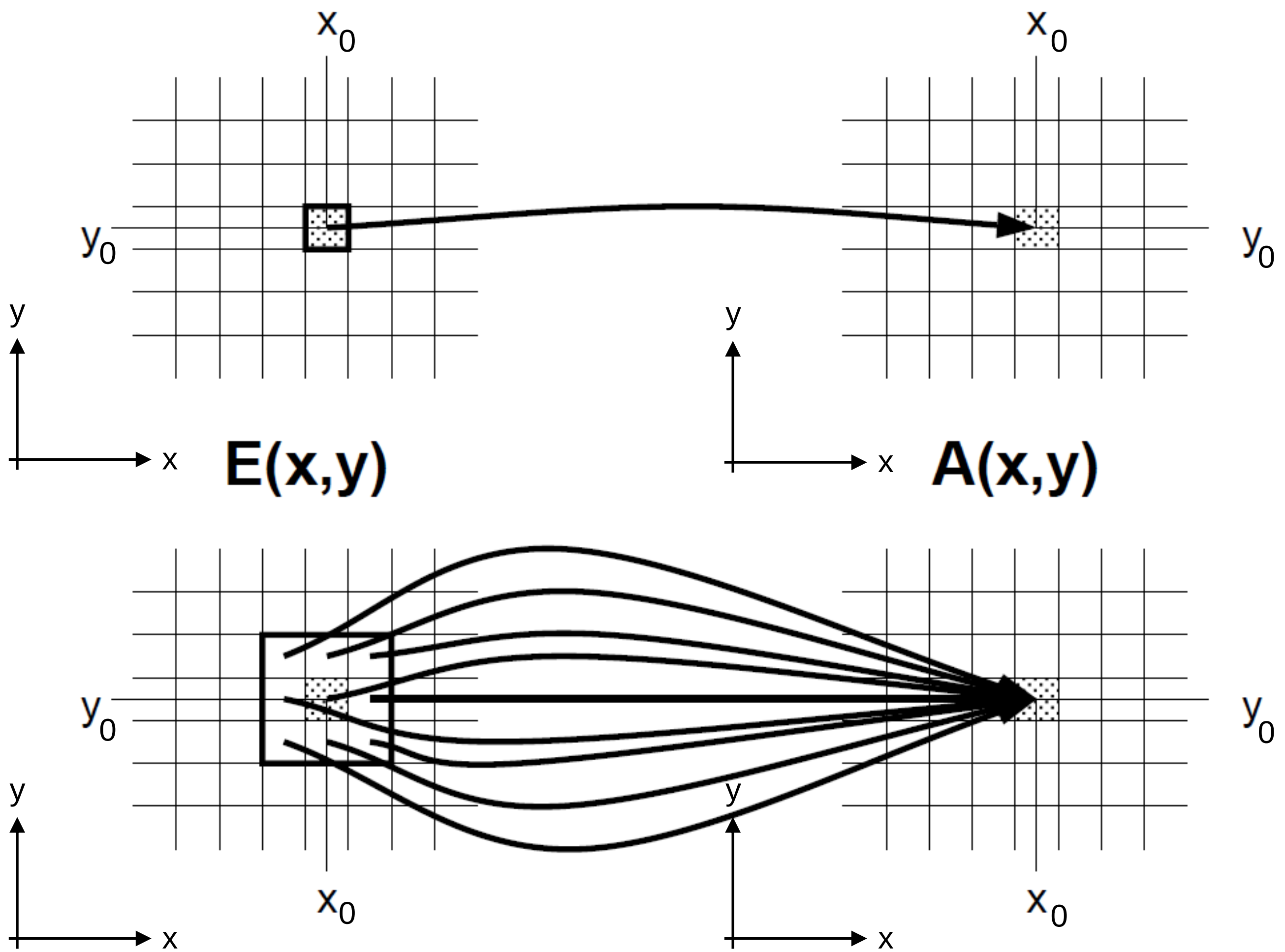
z.B.: Komplexität, geschlossene Formulierung e. Problems, etc.

# Ein Resultat, viele mögliche Lösungswege

```
>> E=imread('peppers.jpg');
>> G5=fspecial('gaussian',15,5);
>> A=imfilter(E,G5);
```



# Punkt- und „lokale“ Operationen

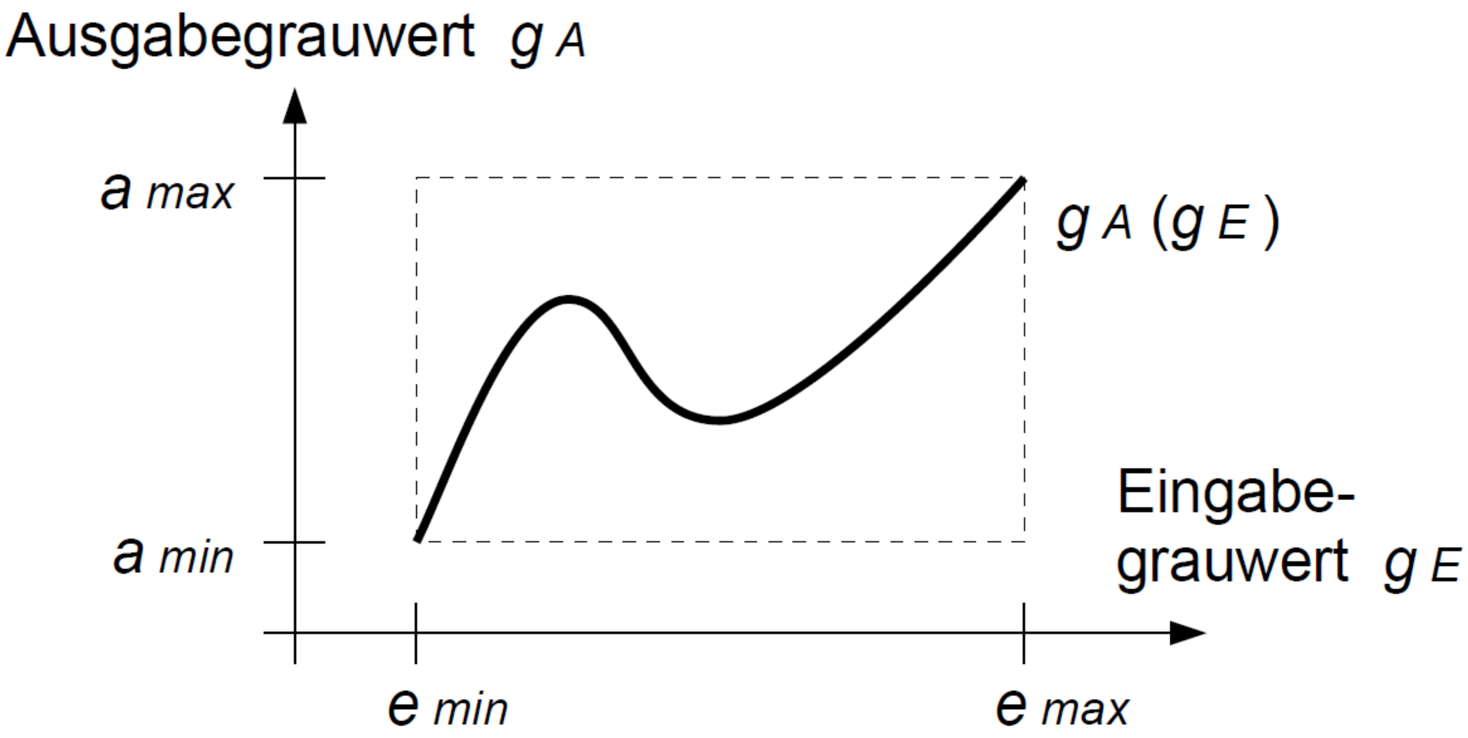



**Punkt-operation**

**Lokale Operation**

# Punktoperationen $A(x,y)=f(E(x,y))$

Allgemein: beliebige Grauwertübertragungsfunktion



# Punktoperationen – Beispiele (1)

Schwellwert *S*:

$$A(x,y) = \begin{cases} 0 & \dots E(x,y) < S \\ 1 & \dots E(x,y) \geq S \end{cases}$$

*E(x,y)*

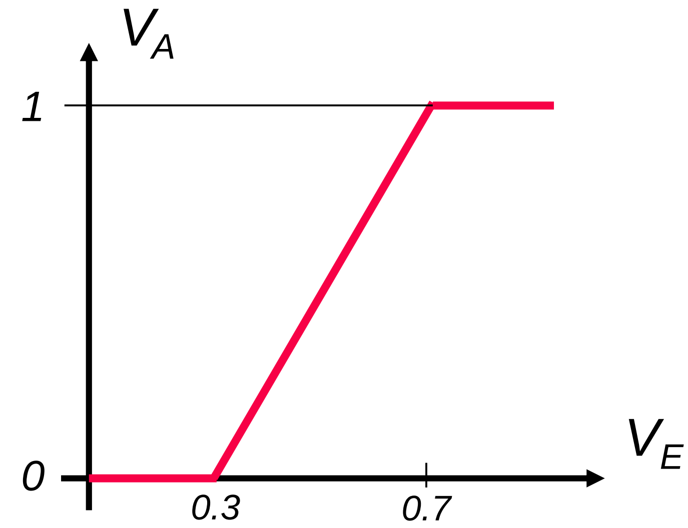
*A(x,y)*

# Punktoperationen – Beispiele (2)

Kontraständerung:

$$A(x,y) = kE(x,y) + d$$

(cropping auf [min,max])




*E(x,y)*

*A(x,y)*

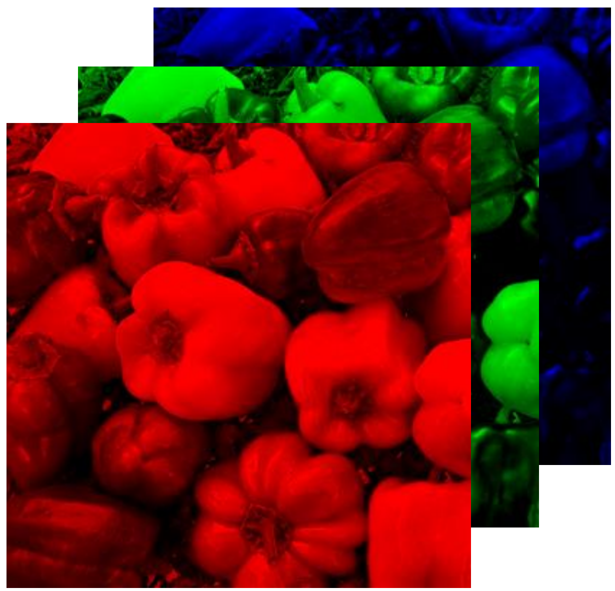
# Punktoperationen – Beispiele (3)

## Punktoperation auf mehreren Bildern: Farbe → monochrom

$$A(x,y) = \frac{1}{3}E_r(x,y) + \frac{1}{3}E_g(x,y) + \frac{1}{3}E_b(x,y)$$

```
>> IN=imread('peppers.jpg');
>> BW=IN(:,:,1)/3+IN(:,:,2)/3+IN(:,:,3)/3;
>> imwrite(BW,'peppers_BW.jpg');
```

$E(x,y)$



$E_r, E_g, E_b$

$A(x,y)$

# Punktoperationen – Beispiele (4)

Punktoperation auf mehreren Bildern: Bewegung in Video

$$A(x,y) = |E_1(x,y) - E_2(x,y)|$$

$E_1(x,y)$

$E_2(x,y)$

$A(x,y)$

Anmerkung: idealisierte, vereinfachte Darstellung aus [Pinz, Bildverstehen, 1994]

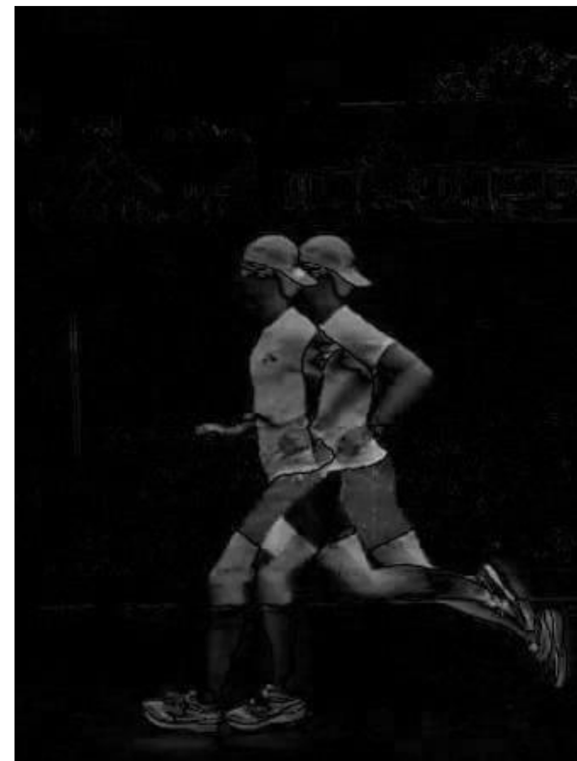
# Bewegung in Video (1)

```
>> frame1=imread('jogger_0.jpg');
>> frame2=imread('jogger_1.jpg');
>> f1_hsv=rgb2hsv(frame1);
>> f2_hsv=rgb2hsv(frame2);
>> f1_v=f1_hsv(:,:,3);
>> f2_v=f2_hsv(:,:,3);
>> f_diff=abs(f1_v-f2_v);
>> imshow(f_diff);
```

jogger_0

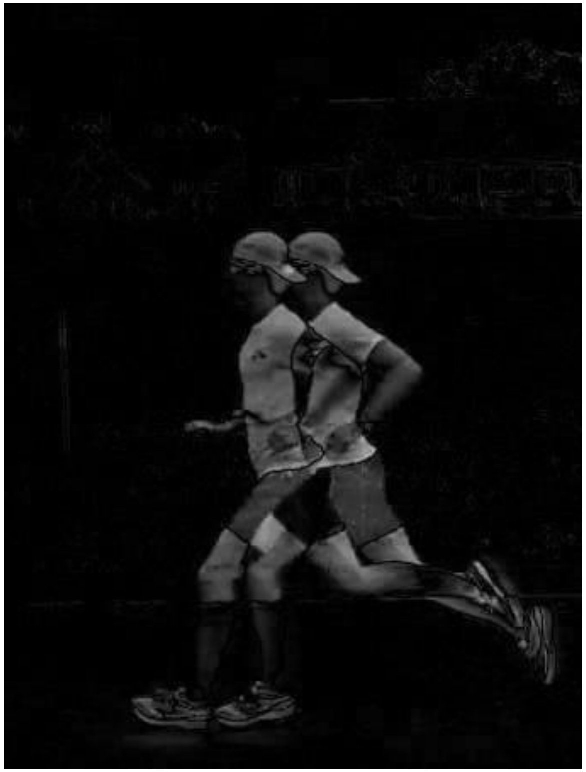
jogger_1

f_diff

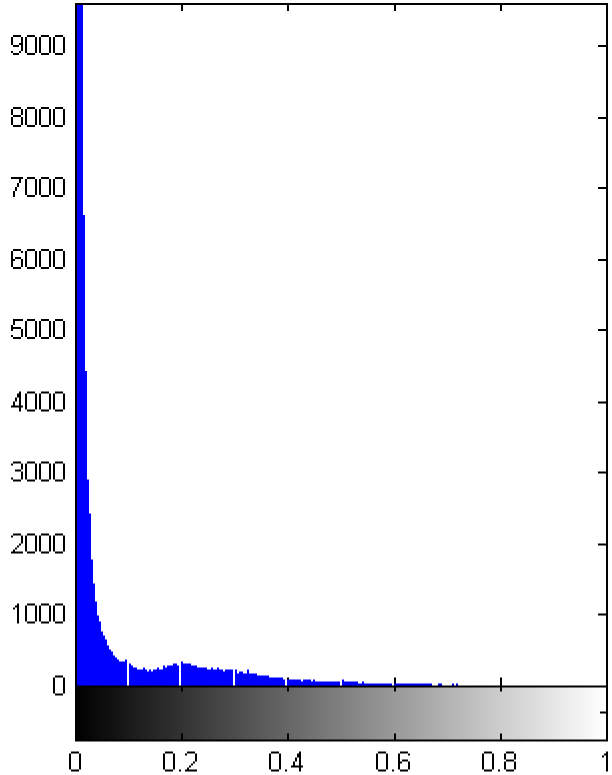


# Bewegung in Video (2)

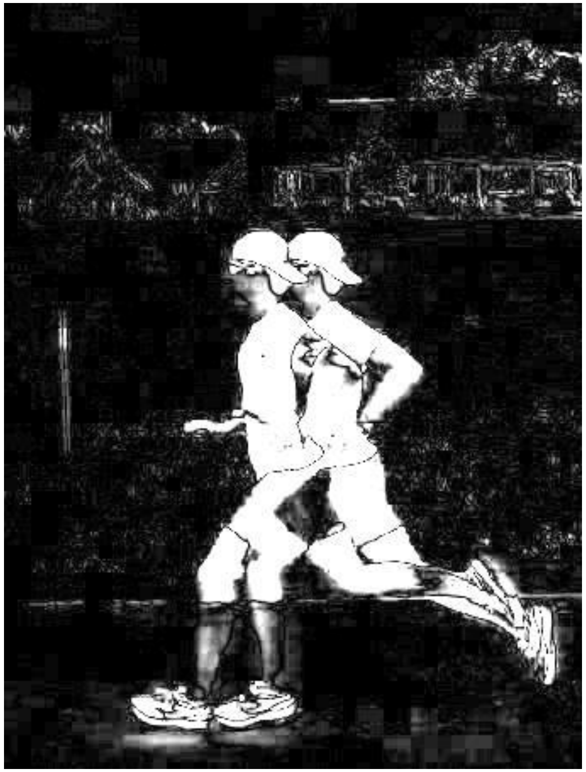
f_diff



imhist(f_diff)

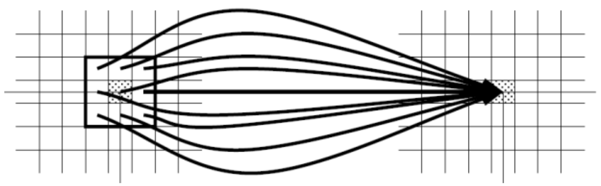



8*f_diff



Auch Bewegung im Hintergrund!

→ Kamera-Pan




# Lokale Fensteroperation

Allgemein: lokales Fenster $W_E$ von *E:*

$$W_E(x,y,\delta) = \begin{pmatrix} E(x-\delta,y-\delta) & \cdots & E(x+\delta,y-\delta) \\ \vdots & E(x,y) & \vdots \\ E(x-\delta,y+\delta) & \cdots & E(x+\delta,y+\delta) \end{pmatrix}, \quad A(x,y) = f(W_E(x,y,\delta))$$

Insbesondere: *Faltung* $A = W * E$

$$A(x,y) = \sum_{s=-\delta}^{\delta} \sum_{t=-\delta}^{\delta} W(s,t) \cdot E(x+s,y+t)$$

Achtung: Behandlung des Bildrandes definieren!

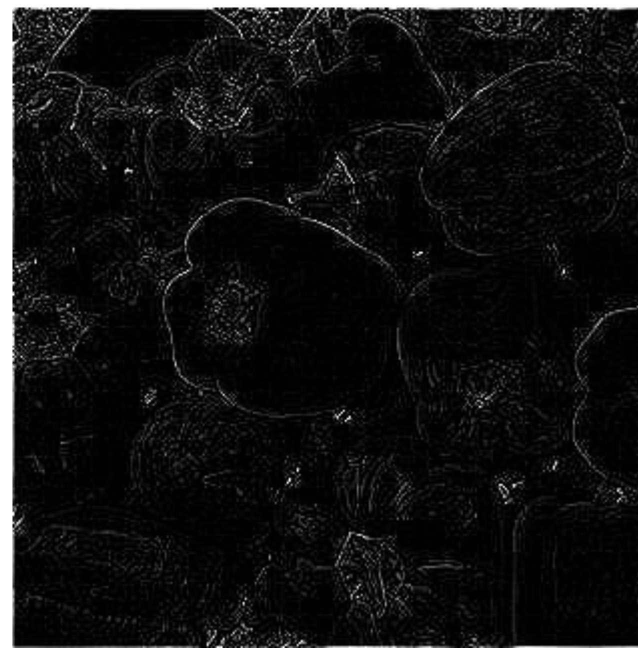
# Beispiele zur Faltung – Hochpass

$$W = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

```
>> E=im2double(imread('peppers_val.jpg'));
>> W=[0,-1,0;-1,4,-1;0,-1,0];
>> A=imfilter(E,W,'conv');
>> imshow(A);
```

*E(x,y)*

*A(x,y)*



# Beispiele zur Faltung – Tiefpass

$$W = \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

```
>> E=imread('peppers_val.jpg');
>> W=[1,1,1;1,1,1;1,1,1];
>> A=imfilter(E,W,'conv');
>> imshow(A/9);
```

*E(x,y)*

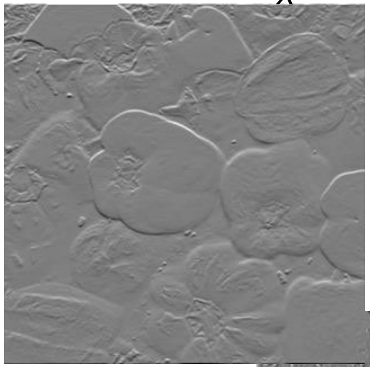
*A(x,y)*

# Beispiele zur Faltung – Kantendetektion mit dem Sobeloperator

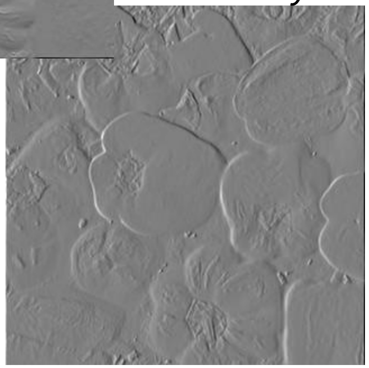
$$S_x = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix},\ S_y = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix},\ S_A = \sqrt{(S_x * E)^2 + \left(S_y * E\right)^2}$$

*E(x,y)*

*E*S$_x$*



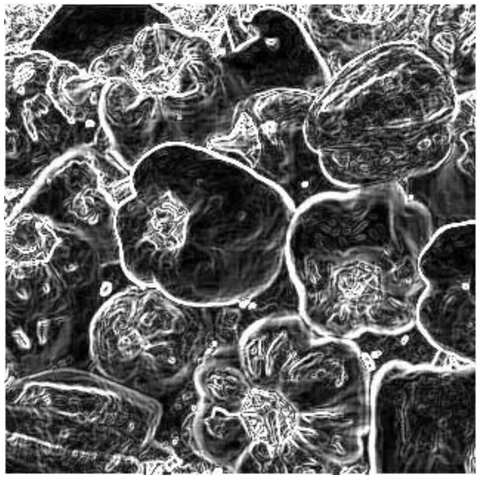
*E*S$_y$*



[-4,4]

→

*A(x,y)*



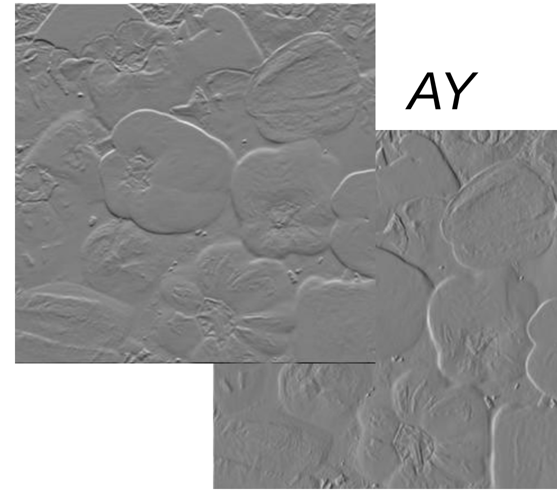
# Matlab Code zum Sobeloperator

```
>> E=im2double(imread('peppers_val.jpg'));
>> SX=[1,2,1;0,0,0;-1,-2,-1];
>> SY=[1,0,-1;2,0,-2,;1,0,-1];
>> AX=imfilter(E,SX,'conv');
>> AY=imfilter(E,SY,'conv');
>> imwrite(AX,'sobel_X.jpg');
>> imwrite(AY,'sobel_Y.jpg');
>> A=abs(AX)+abs(AY);
>> imshow(A);
>> imwrite(abs(AX),'abs_sobel_X.jpg');
>> imwrite(abs(AY),'abs_sobel_Y.jpg');
>> imwrite(A,'sobel_amplitude.jpg');
```

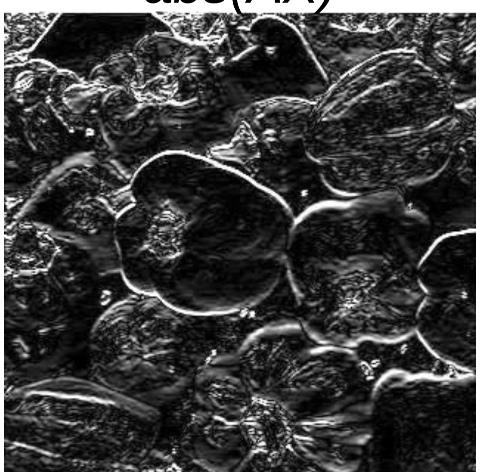
*E*

*AX*

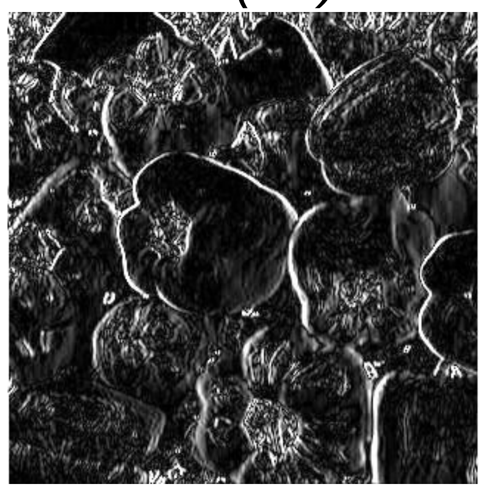
*AY*



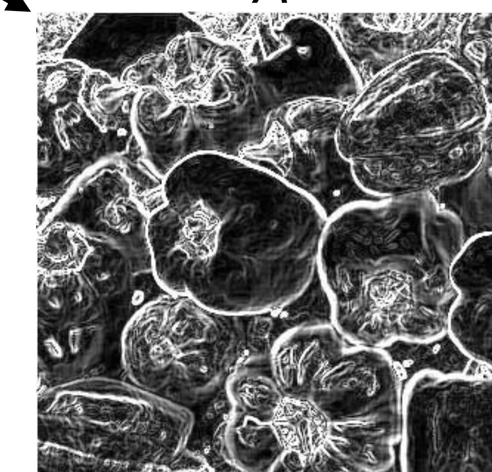
*abs(AX)*



*abs(AY)*



*A*



# Globale Operationen *A(u,v)=f(E)*
# Beispiel Fouriertransformation

$$\mathcal{F}\{f(x,y)\} = F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-2\pi i(ux+vy)}\,dx\,dy$$

$$\mathcal{F}^{-1}\{F(u,v)\} = f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)e^{2\pi i(ux+vy)}\,du\,dv$$

$$\begin{aligned} e^{i\phi} &= \cos\phi + i\sin\phi \\ e^{-i\phi} &= \cos\phi - i\sin\phi \end{aligned}$$

$$\begin{aligned} F(u,v) &= |F(u,v)|e^{j\Phi(u,v)} \\ |F(u,v)| &\ \ldots\ \text{Fourierspektrum} \\ \Phi(u,v) &\ \ldots\ \text{Phase} \end{aligned}$$

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)e^{-2\pi i\left(\frac{ux+vy}{N}\right)}$$

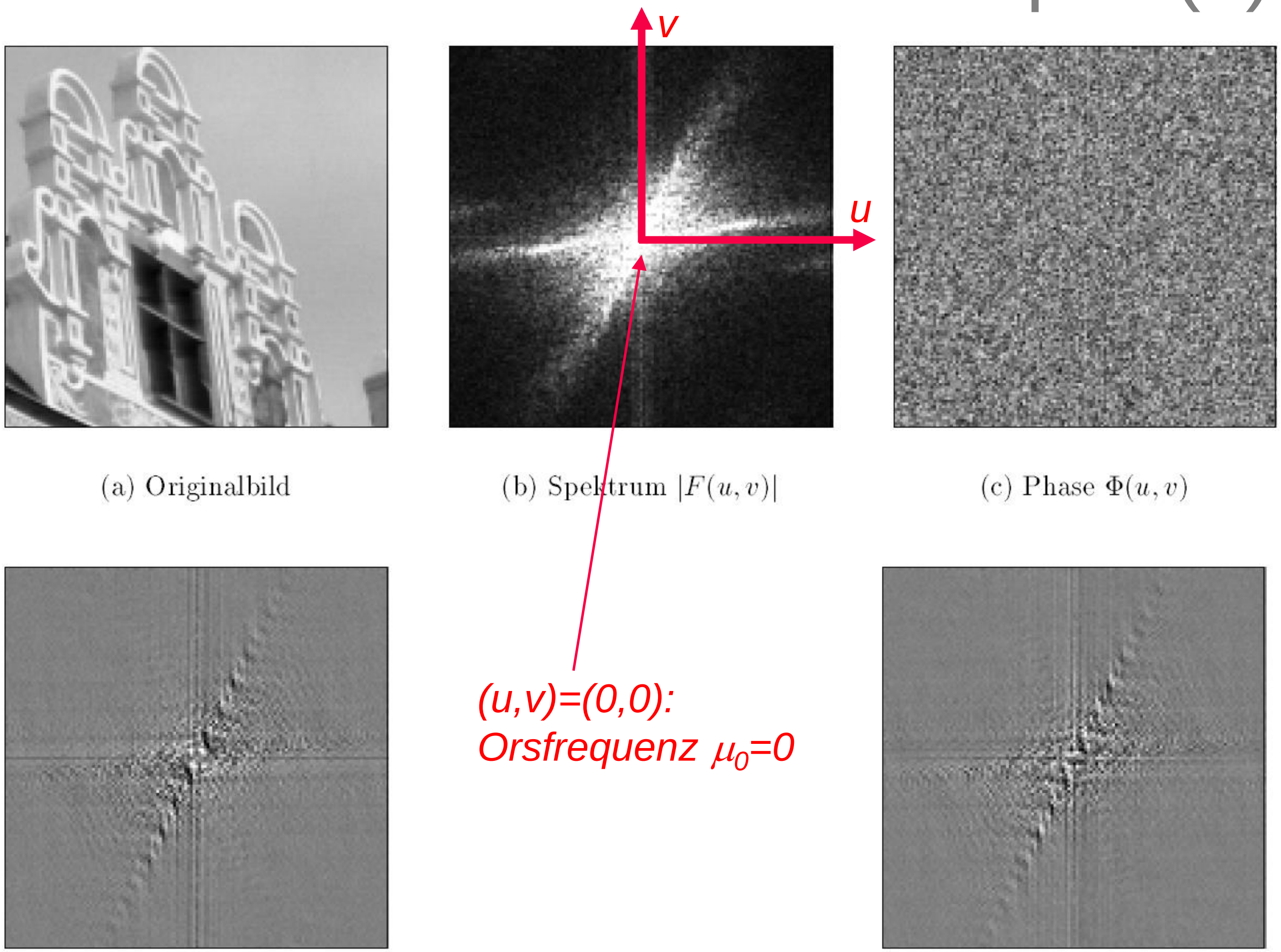
$$f(x,y) = \frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} F(u,v)e^{2\pi i\left(\frac{ux+vy}{N}\right)}$$

Filterung:

$$f(x,y) * g(x,y) = \mathcal{F}^{-1}\{F(u,v)G(u,v)\}$$

$$f * g \quad \circ\!\!-\!\!\bullet \quad F \cdot G$$

# Fouriertransformation – Beispiel (1)

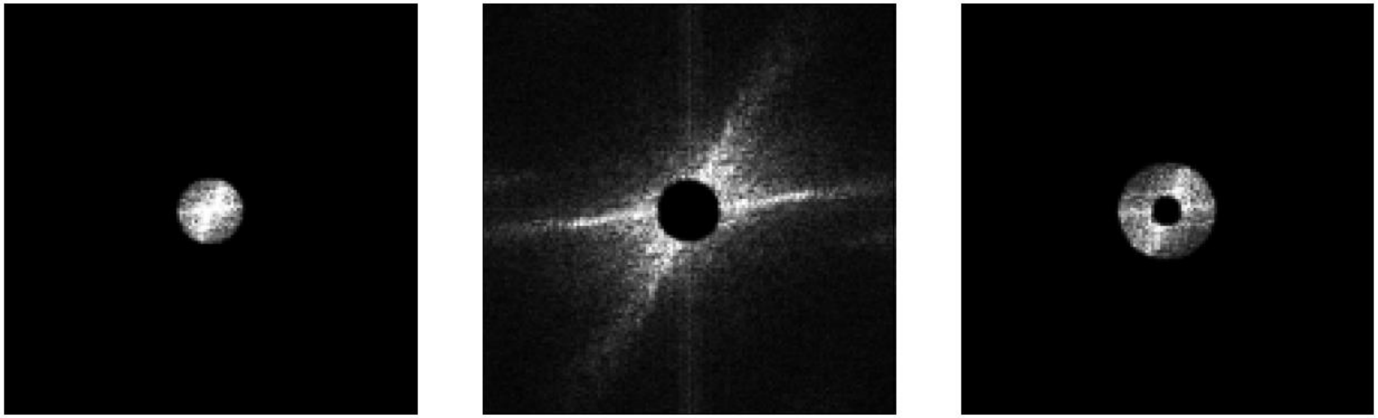



(a) Originalbild (b) Spektrum $|F(u,v)|$ (c) Phase $\Phi(u,v)$

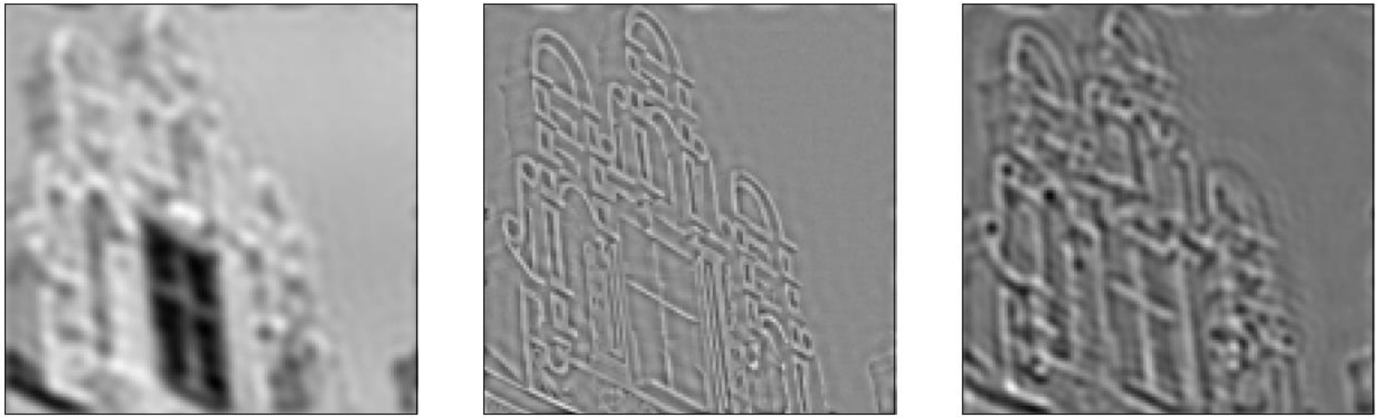
(d) Realteil (e) Imaginärteil

[Pinz, Bildverstehen, 1994]

# Fouriertransformation – Beispiel (1) [Pinz, 1994]



(a) Tiefpaß (b) Hochpaß (c) Bandpaß



(d) Tiefpaß (e) Hochpaß (f) Bandpaß

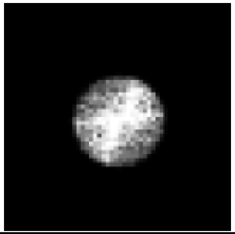
# Ein Resultat, viele mögliche Lösungswege

```
>> E=imread('peppers.jpg');
>> G5=fspecial('gaussian',15,5);
>> A=imfilter(E,G5);
```

$E(x,y) \xrightarrow{f} A(x,y)$

$T$: $E(x,y) \rightarrow F_E(u,v)$

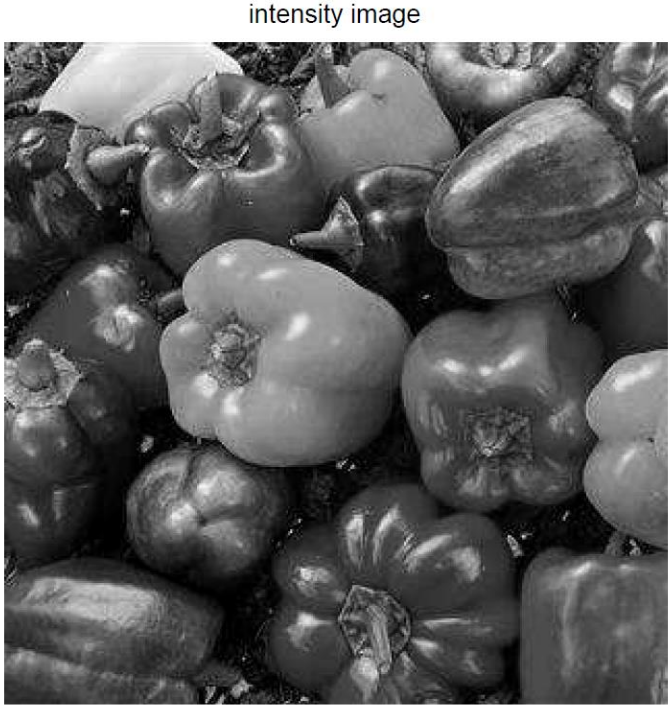
$T^{-1}$: $F_A(u,v) \rightarrow A(x,y)$

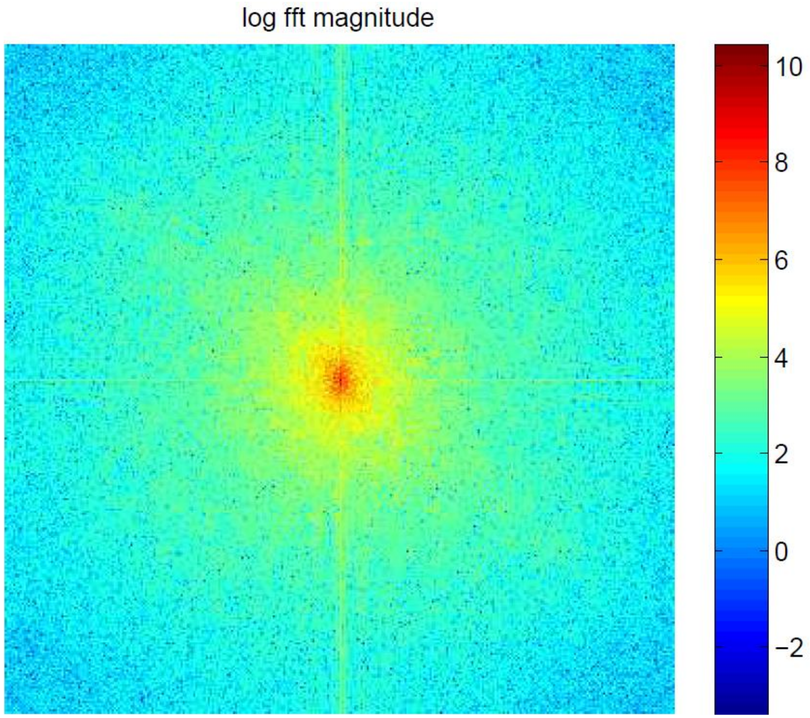


$F_E(u,v) \xrightarrow{g} F_A(u,v)$

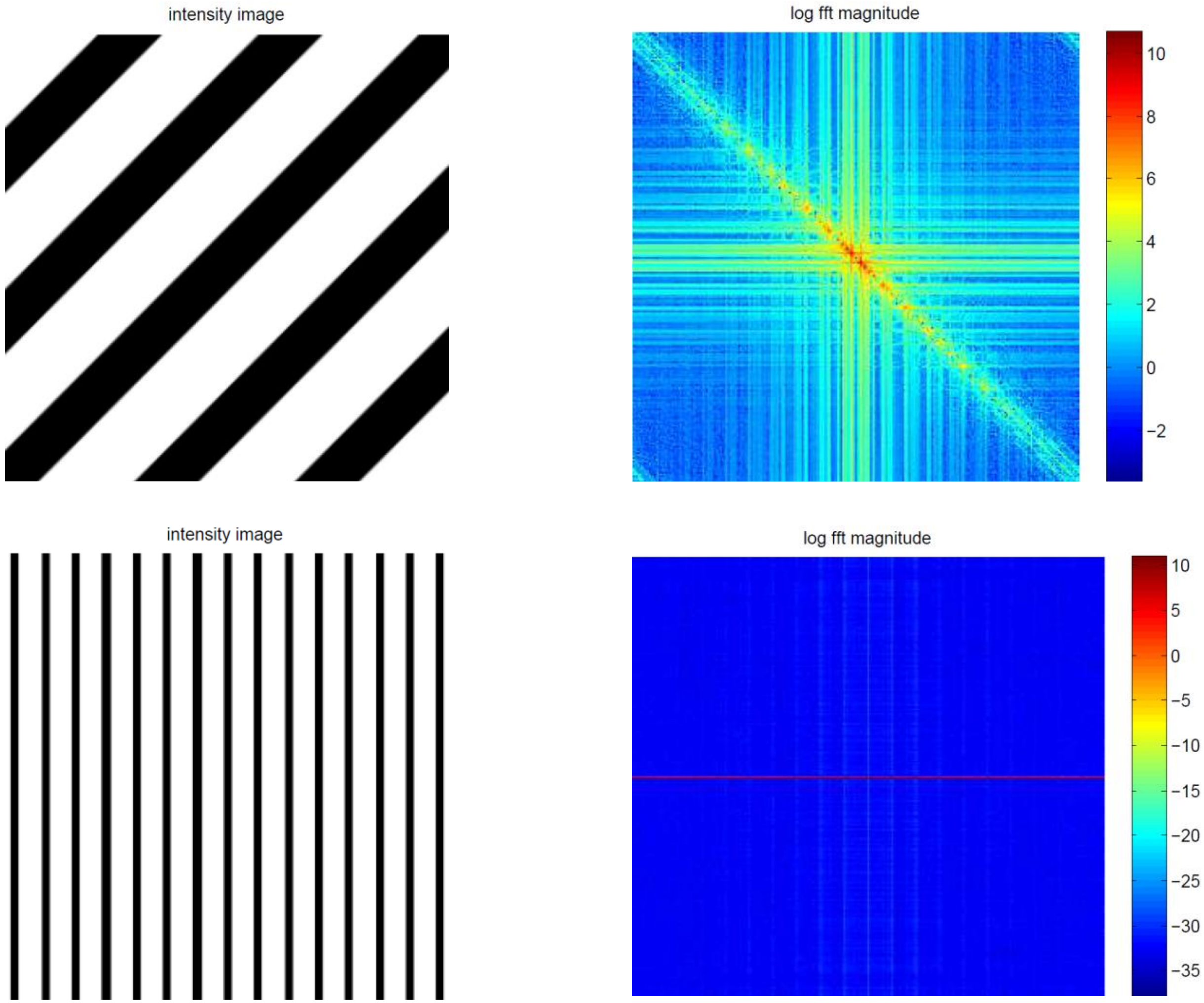
# Fouriertransformation in Matlab

```
>> E=im2double(imread('peppers_BW.jpg'));
>> F=fft2(E);
>> F=fftshift(F);
>> figure(1), hold off, imagesc(E), axis off, colormap gray, axis image, title('intensity image');
>> figure(2), hold off, imagesc(log(abs(F))), axis off, colormap jet, axis image, colorbar, title('log fft magnitude');
```



# Fouriertransformation – Beispiel (2)




*Vgl: rezeptives Feld von komplexen Zellen im visuellen Cortex!*

# Segmentation

Unterteilen eines Bildes *B* in disjunkte Regionen $R_i$

Formale Definition von Segmentation benötigt ein Homogenitätskriterium *H*:

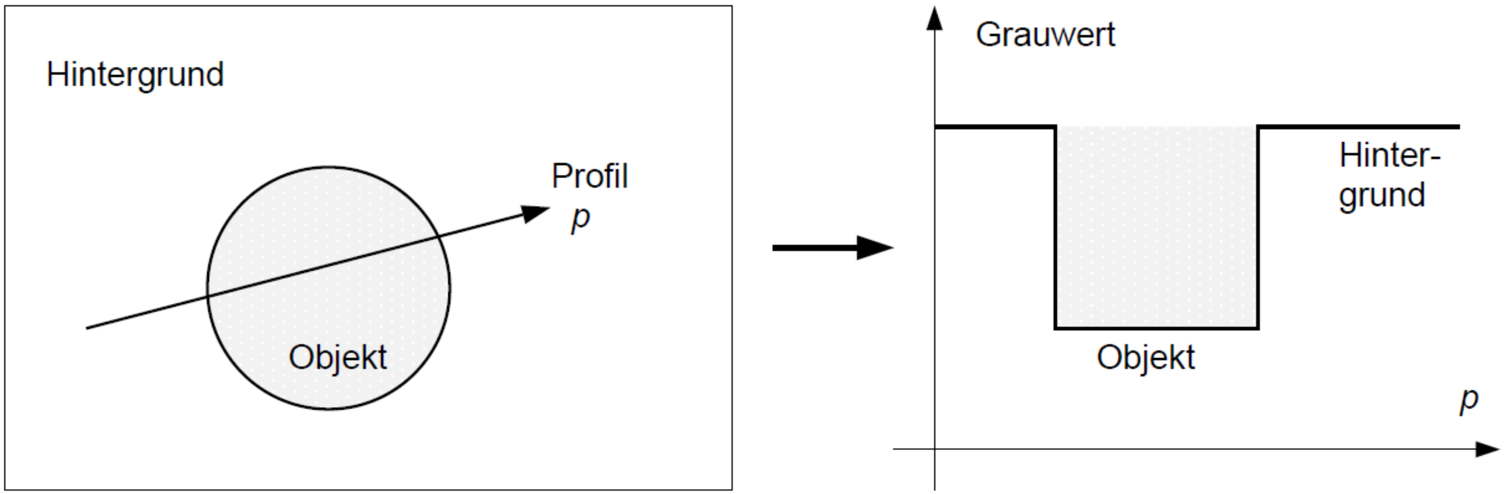
$$\bigcup_{i=1}^{N_R} R_i = B, \quad wobei \; \forall i : R_i \subseteq B$$

$$\forall i \neq j : R_i \cap R_j = \emptyset$$

$$H(R_i) = true$$

$$H(R_i \cup R_j) = false, \quad wenn \; R_i \; Nachbar \; von \; R_j \; ist.$$

# Flächen- / kantenbasierte Segmentation



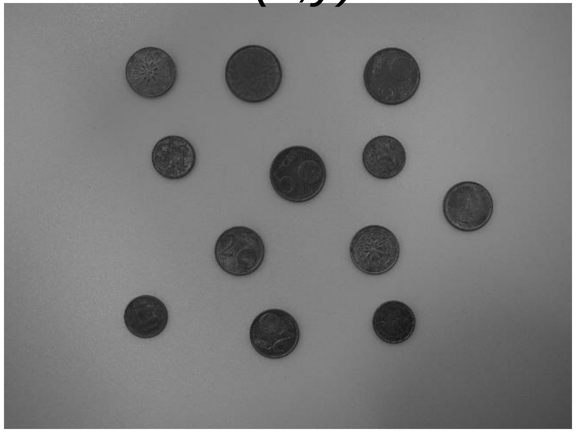

Segmentation durch:

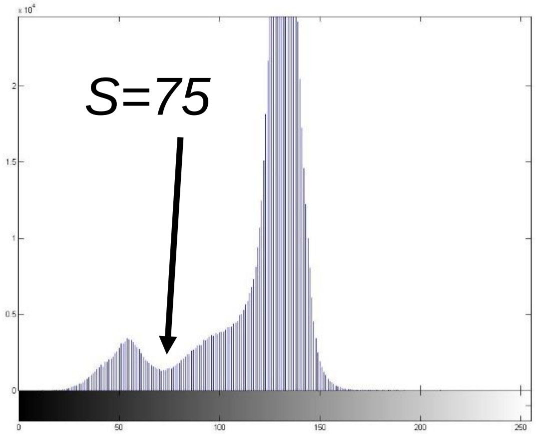
1. Finden der Objekte als homogene *Flächen*
2. Finden der *Grenzen* zwischen Objekt und Hintergrund.

[Pinz, Bildverstehen, 1994]

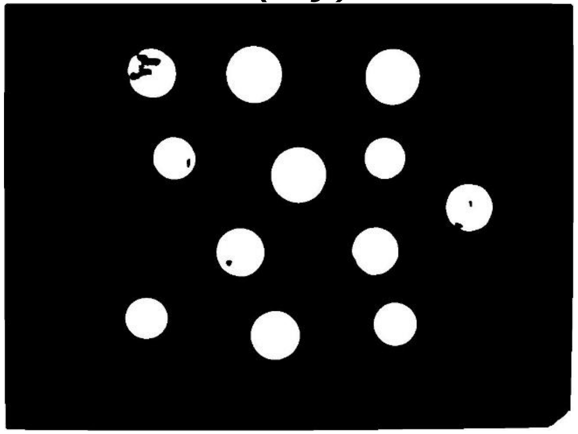
# Beispiel: Regionenbasierte Segmentation mittels Schwellwert

*E(x,y)*

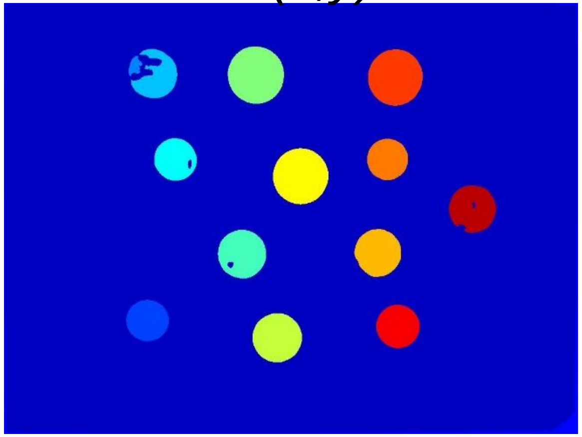





*A(x,y)*



*A2(x,y)*



1. Glätten (Gauß)
2. Schwellwert
3. Region labelling
4. Segmentation

```
>> G5=fspecial('gaussian',15,5);
>> EG=imfilter(E,G5);
>> A=1-im2bw(EG, 75/255);
>> A1=bwconncomp(A);
>> A2=labelmatrix(A1);
```

# Messen optischer Größen, Messen aus Bildern
# Übersicht

Optische Strahlung, Sensorik

Geometrie, Photogrammetrie
- Kamerakalibrierung
- Stereo

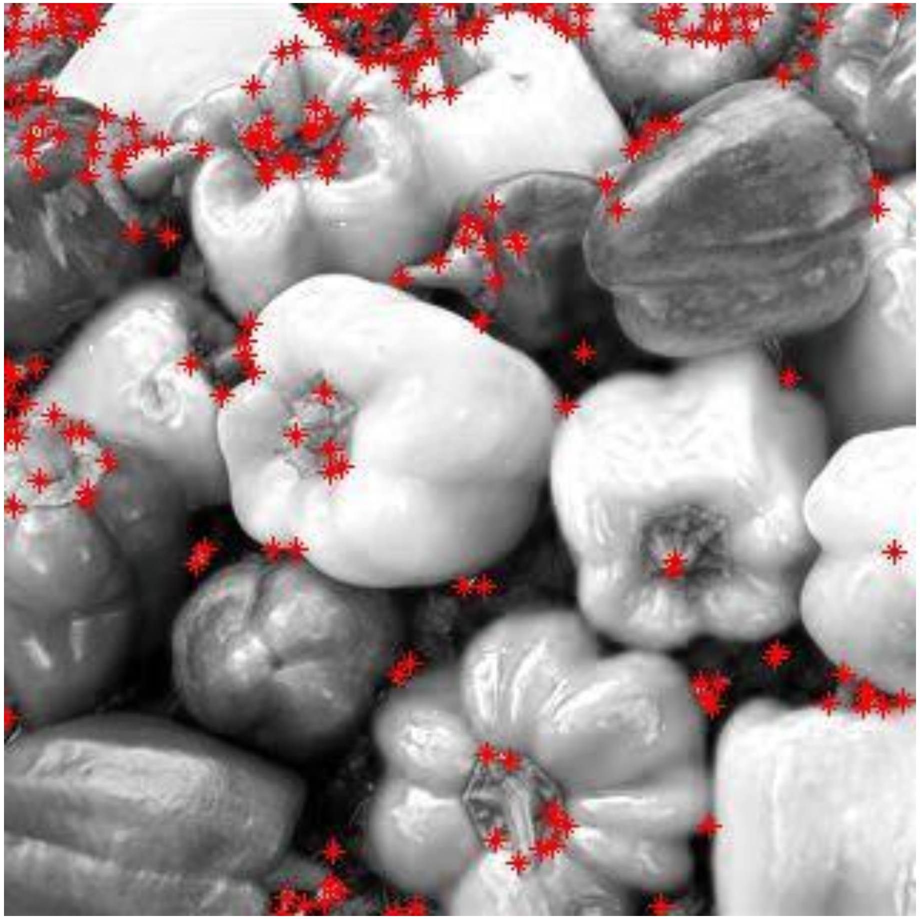
Menschliche Wahrnehmung
- Neurophysiologie
- Kognitive Psychologie

Digitale Bildanalyse
- Digitales Rasterbild, Kenngrößen
- Bildverarbeitungsoperationen, Segmentation
- Salient point detection + description

# Detektion und Beschreibung auffälliger Punkte “salient points” / “corners” / „Ecken“

## Beispiel 1:

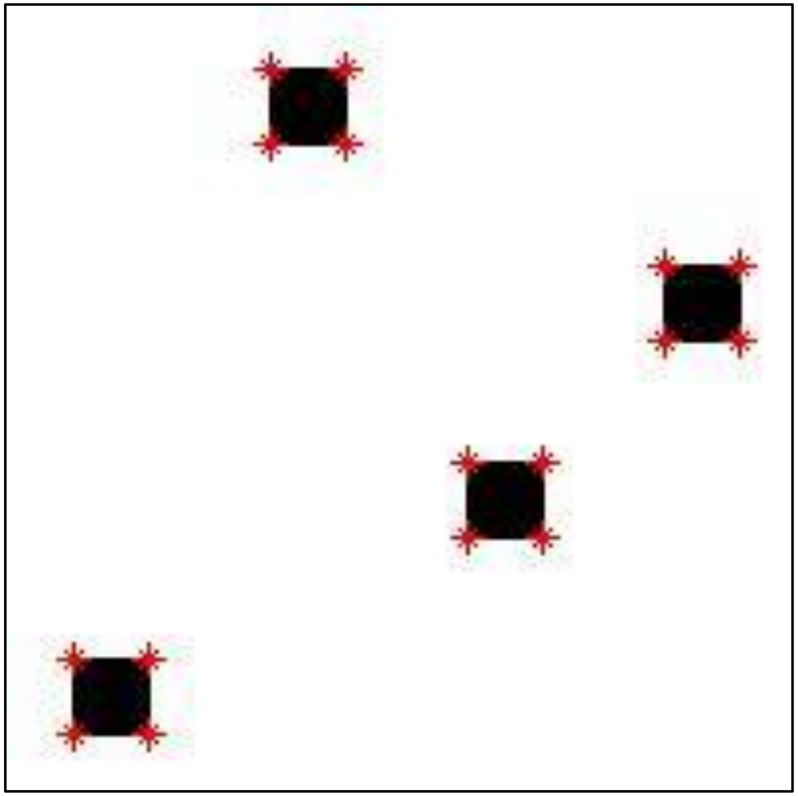
```
>> E=imread('peppers_val.jpg');
>> C=corner(E, 'Harris');
>> imshow(E);
>> hold on;
>> plot(C(:,1), C(:,2),'r*');
```



# Detektion und Beschreibung auffälliger Punkte “salient points” / “corners” / „Ecken“

## Beispiel 2:

```
>> E=imread('squares.jpg');
>> C=corner(E, 'Harris');
>> imshow(E);
>> hold on;
>> plot(C(:,1), C(:,2),'r*');
```

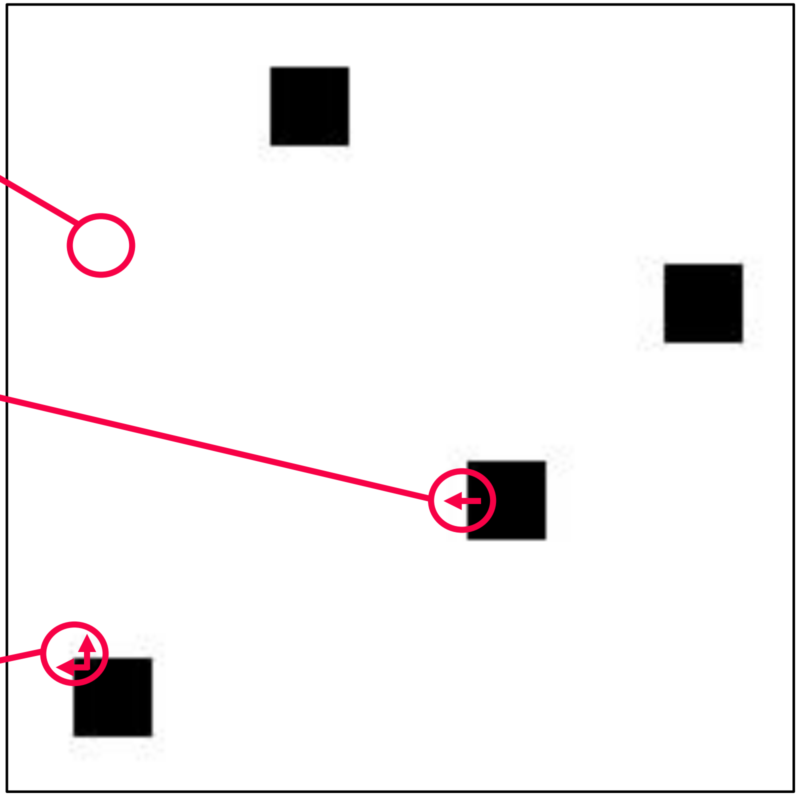


# Was definiert eine „Ecke“?

Homogene Bildregion: Kein Gradient

Kante: Gradient in 1 Richtung

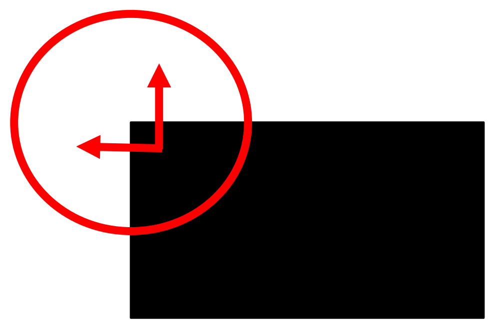
Ecke: Mindestens 2 Gradienten-Richtungen im „rezeptiven Feld“



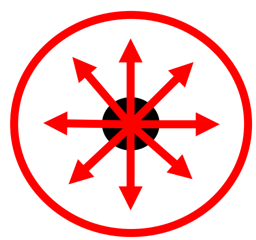
Gradient ↔ erste Ableitung

# Ecke ↔ "salient point"

Ecke:  Gradienten in 2 Richtungen



Auch das ist ein "salient point": Gradienten in *alle* Richtungen



*Vgl: "On-Off" und "Off-On" Zellen !*

# Point Correspondences - Example

Structure and motion from
“natural” landmarks [Schweighofer]

→ Stereo reconstruction of Harris corners

# Salient points (corners) based on 1st derivatives

- Autocorrelation of 2D image signal [Moravec]
  - Approximation by sum of squared differences (SSD)
  - Window *W*
  - Differences between grayvalues in *W* and a window shifted by (Δ*x*,Δ*y*)

$$f(x,y) = \sum_{x_w=-\delta}^{\delta} \sum_{y_w=-\delta}^{\delta} [I(x_w,y_w) - I(x_w+\Delta x, y_w+\Delta y)]^2$$

$$= \sum_{(x_w,y_w)\in W} [I(x_w,y_w) - I(x_w+\Delta x, y_w+\Delta y)]^2$$

  - Four different shift directions $f_i(x,y)$: $f_{\mathrm{Moravec}} = \sum_{i=1}^{4} f_i$
  - A corner is detected, when $f_{\mathrm{Moravec}} > th$

# Salient points (corners) based on 1st derivatives

- Autocorrelation (*second moment)* matrix:
  - Avoids various shift directions
  - Approximate $I(x_w+\Delta x, y_w+\Delta y)$ by Taylor expansion:

$$I(x_w+\Delta x, y_w+\Delta y) \approx I(x_w, y_w) + \begin{pmatrix} I_x(x_w, y_w) & I_y(x_w, y_w) \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

Gradient in x-Richtung

  - Rewrite $f(x,y)$:

$$f(x,y) = \sum_{(x_w,y_w)\in W} [\begin{pmatrix} I_x(x_w, y_w) & I_y(x_w, y_w) \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}]^2$$

$$= \sum_{(x_w,y_w)\in W} \begin{pmatrix} \Delta x & \Delta y \end{pmatrix} \begin{pmatrix} I_x(x_w, y_w) \\ I_y(x_w, y_w) \end{pmatrix} \begin{pmatrix} I_x(x_w, y_w) & I_y(x_w, y_w) \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

$$= \begin{pmatrix} \Delta x & \Delta y \end{pmatrix} \mathbf{M} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

"second moment matrix **M**"

# Salient points (corners) based on 1st derivatives

- Autocorrelation (*second moment*) matrix:

$$\mathbf{M} = \mu = \begin{pmatrix} \sum\limits_{(x_w,y_w)\in W} I_x^2(x_w,y_w) & \sum\limits_{(x_w,y_w)\in W} I_x(x_w,y_w)I_y(x_w,y_w) \\ \sum\limits_{(x_w,y_w)\in W} I_x(x_w,y_w)I_y(x_w,y_w) & \sum\limits_{(x_w,y_w)\in W} I_y^2(x_w,y_w) \end{pmatrix}$$

  - M can be used to derive a measure of “cornerness”
  - Independent of various displacements ($\Delta x$,$\Delta y$)
  - Corner: significant gradients in >1 directions ➔ rank **M** = 2
  - Edge: significant gradient in 1 direction ➔ rank **M** = 1
  - Homogeneous region ➔ rank **M** = 0
- Several variants of this corner detector:
  - KLT corners, Förstner corners

# Salient points (corners) based on 1st derivatives

- Harris corners
  - Most popular variant of a detector based on **M**
  - Local derivatives with “derivation scale” $\sigma_D$
  - Convolution with a Gaussian with “integration scale” $\sigma_I$
  - $\mathbf{M}_{\text{Harris}}$ for each point $\mathbf{x}$ in the image

$$\mathbf{M}_{\text{Harris}}(\mathbf{x},\sigma_I,\sigma_D)=\mu(\mathbf{x},\sigma_I,\sigma_D)=\sigma_D^2 G(\sigma_I)*\begin{pmatrix} I_x^2(\mathbf{x},\sigma_D) & I_x(\mathbf{x},\sigma_D)I_y(\mathbf{x},\sigma_D) \\ I_x(\mathbf{x},\sigma_D)I_y(\mathbf{x},\sigma_D) & I_y^2(\mathbf{x},\sigma_D) \end{pmatrix}$$

  - Cornerness $c_{\text{Harris}}$ does not require to compute eigenvalues

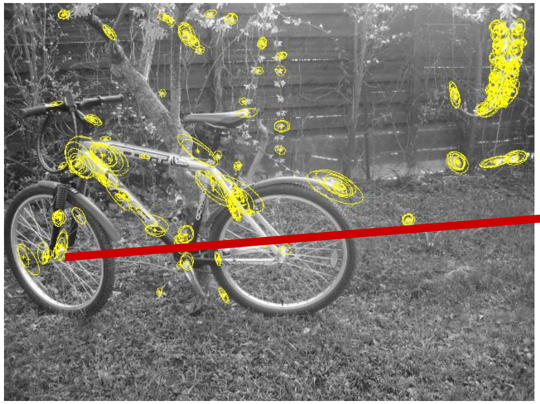
$$c_{\text{Harris}}=\det\mathbf{M}-\alpha\ \operatorname{trace}\mathbf{M}$$

  - Corner detection: $c_{\text{Harris}} > t_{\text{Harris}}$

$$\sigma_I=2,\quad \sigma_D=0.7$$
$$\alpha=0.05,\quad t_{\text{Harris}}=25000$$

# Descriptors (1)

- Representation of salient regions
- “descriptive” features → feature vector
- There are *many* possibilities !
- Matching, calibration, specific object recognition:
  - Search for high descriptive power
  - Tolerate viewpoint variations



$$\mathbf{f}_n = (f_{n,1}, \cdots, f_{n,j})^T$$

feature vector extracted from patch $P_n$

# Descriptors (2)

- Grayvalues
  - Raw pixel values of a patch *P*
  - “local appearance-based description”

- General moments of order *p*+*q*:

$$m_{pq} = \sum_{x \in P} \sum_{y \in P} x^p y^q I(x,y)$$

- Moment invariants:
  - Central moments $\mu_{pq}$: invariant to *translation*

$$\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}}: \quad \mu_{pq} = \sum_{x \in P} \sum_{y \in P} (x - \bar{x})^p (y - \bar{y})^q I(x,y)$$

# Descriptors (3)

additional (optional) material

- Moment invariants:
  - Normalized central moments

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}, \quad \text{with } \gamma = 1 + \frac{p+q}{2}$$

  - Translation, rotation, scale invariant moments Φ1 ... Φ7 [Hu]

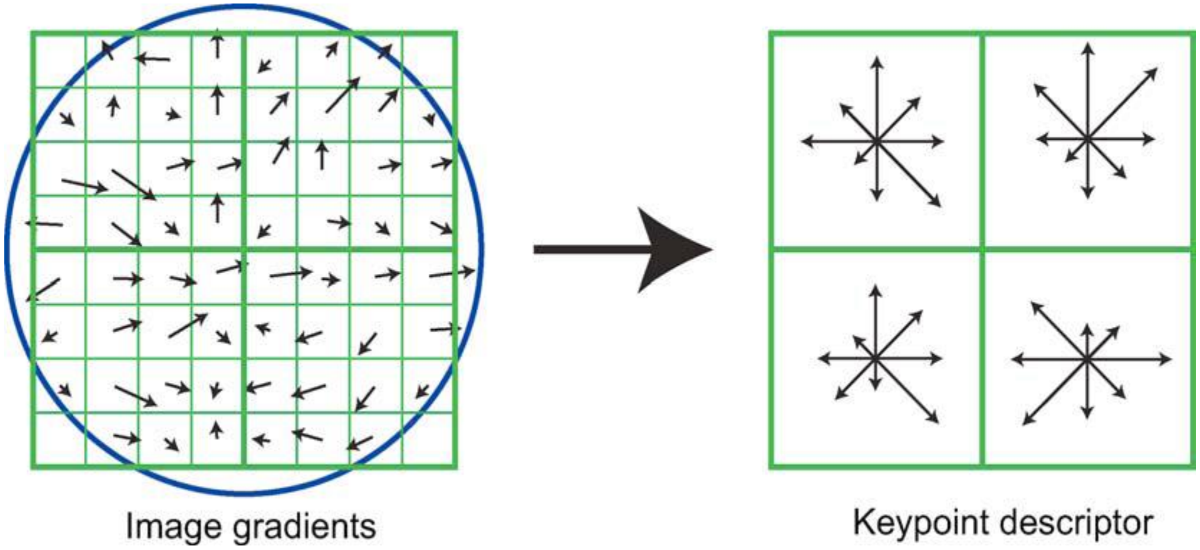
$$\Phi_1 = \eta_{20} + \eta_{02}$$

$$\Phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

  - Geometric/photometric, color invariants [vanGool et al.]
- Filters
  - "local jets" [Koenderink+VanDoorn]
  - Gabor banks, steerable filters, discrete cosine transform DCT

# Descriptors (4)

- SIFT descriptors [Lowe]
  - Scale invariant feature transform
  - Calculated for local patch *P*:   8x8 or 16x16 pixels
  - Subdivision into 4x4 sample regions
  - Weighted histogram of 8 gradient directions:  0º, 45º, …
  - SIFT vector dimension: 128 for a 16x16 patch

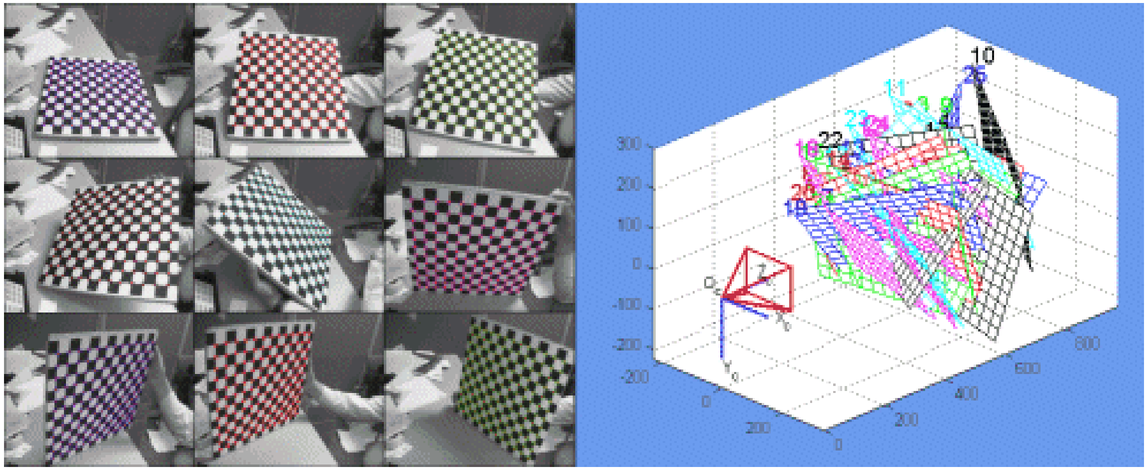



[Lowe]

# Wo braucht man Punktdetektoren und -deskriptoren?

Immer wenn es nötig ist, *Punktkorrespondenzen* herzustellen!

- Kamerakalibrierung
- 3D Tiefe aus Stereokorrespondenz
- …
- Immer, wenn aus Bildern *gemessen* werden soll!




# Messen optischer Größen, Messen aus Bildern Übersicht

## Optische Strahlung, Sensorik

## Geometrie, Photogrammetrie

- Kamerakalibrierung
- Stereo

## Menschliche Wahrnehmung ✓

- Neurophysiologie
- Kognitive Psychologie

## Digitale Bildanalyse

- Digitales Rasterbild, Kenngrößen
- Bildverarbeitungsoperationen, Segmentation
- Salient point detection + description