Intelligente Sehsysteme

8 Skaleninvariante Erkennung und Beschreibung von markanten Punkten

Interest Points, Harris Detector, Blob Detection

SIFT: Keypoint Detection & Keypoint Descriptors

Volker Steinhage

Inhalt

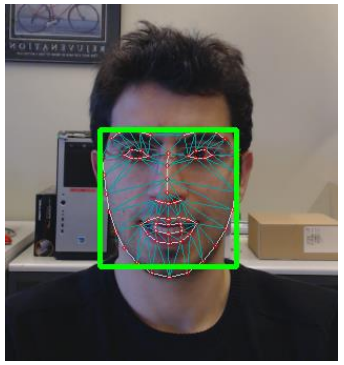
- punktinduzierte Merkmale
  - Markante Punkte (*interest points*) und ihre Erkennung
  - Harris Corner Detector
- regioneninduzierte Merkmale
  - Blob-Erkennung mit LoG
  - charakteristische Skala
- SIFT
  - Gauß-Pyramide und DoG-Pyramide
  - Skaleninvariante Keypoint-Detektion
  - Rotations- und skalierungsinvarianter Keypoint-Deskriptor

Eckenfilter

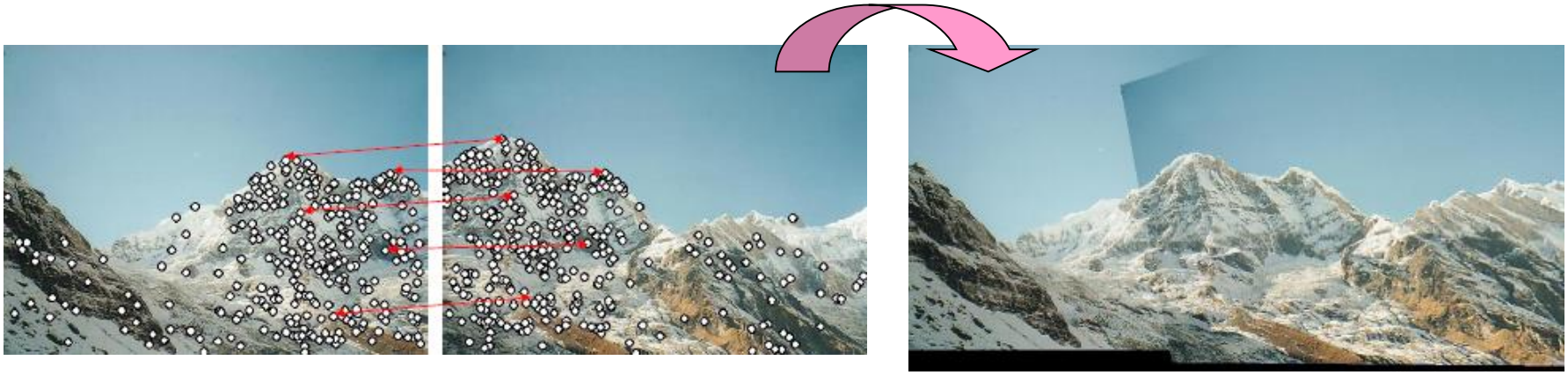
Markante Punkte (engl. *interest points*) haben Anwendungen z. B. bei

- Stereoanalyse
- Tracking
- Bildanschluss (engl. *image alignment*) (s. Abb.)
- Objekterkennung

Die Erkennung von markanten Punkten basiert auf sog. Interest-Operatoren oder auch Eckenoperatoren (engl. *corner detectors*)



Bildquelle: Anil Yüce, Signal Processing Lab 5, École Poytechn. Fed. de Lausanne



Bildquelle: D. Frolova, D. Simakov, Weizmann Inst. of Science, Computer Vision Lab

Ecken und markante Punkte (1)

- Als *Ecke* wird hier verstanden:
  - der Schnitt von zwei Linien
  - ein Punkt, der zwei unterschiedlich orientierte signifikante Gradienten in der lokalen Nachbarschaft zeigt

- Als *markanter Punkt* wird hier ein Bildpunkt verstanden, der eine wohldefinierte Position aufweist und robust identifizierbar ist:
  - Ecken
  - isolierte Punkte mit lokalem Intensitätsmaximum oder -minimum
  - Kreuzungspunkte
  - Linienenden
  - lokale Krümmungsmaxima bei gekrümmten Linien oder Kanten

Liste

Ecken und markante Punkte (2)

- In der Praxis extrahieren die meisten Eckenoperatoren markante Punkte

  → falls also nur Ecken im engeren Sinne extrahiert werden sollen, ist dann eine lokale Untersuchung der gefundenen Punkte nötig

- In der Literatur werden die Begriffe „Ecke“, „markanter Punkt“, „Merkmal“ bzw. "corner", "interest point" und "feature" leider uneinheitlich benutzt

Harris Corner Detector (1)

Der Klassiker:

- Der *Harris-Operator* ist als „Corner Detector“ leicht erklärbar und wird auch als solcher bezeichnet:

  C. Harris, M. Stephens: *A combined corner and edge detector.* Proc. 4th Alvey Vision Conference, pp. 147-151, 1988

- Tatsächlich findet der Harris-Operator aber markante Punkte!

Harris Corner Detector (2)

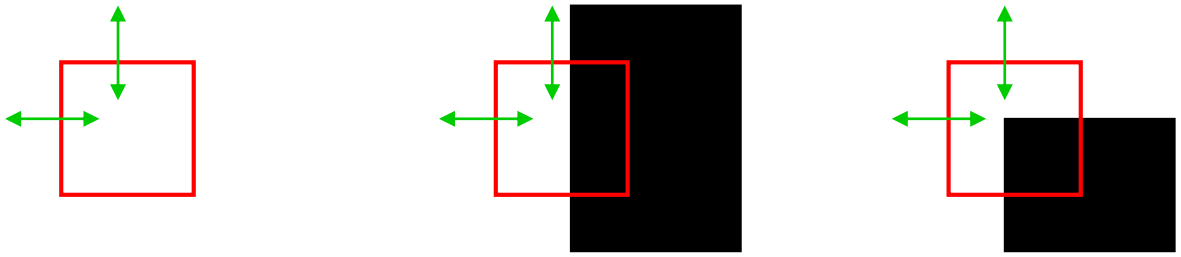
Die Funktionsweise erklärt sich informell durch Bewegen eines Fensters im Bild:

(1) In homogenen Bildbereichen: kaum Änderungen der Intensitätsverteilung im Fenster – egal, in welche Richtung das Fenster verschoben wird

(2) An einer Kante

- geringe Änderungen bei Verschiebung entlang der Kante
- große Änderungen bei Verschiebung senkrecht zur Kante

(3) An einer Ecke: große Änderungen bei Verschiebung in jede Richtung



(1) (2) (3)

Harris Corner Detector (3)

Geg.:

- Grauwertbild **I** = [I(x,y)], Fenster I(u,v) und dessen Verschiebung um (x,y)
- Gewichtsfunktion w(u,v) nach Gauß-Verteilung $w(u,v) = e^{\frac{-(u^2+v^2)}{2\sigma^2}}$
- Änderung der Intensitätsverteilung wird gemessen durch mit w(u,v) gewichtete Summe S von Differenzquadraten über den Intensitätsverteilungen des Fensters I(u,v) und seiner um (x,y) verschobenen Version I(x+u,y+v):

$$S(x,y) = \sum_{u,v} w(u,v)\left[I(x+u,y+v) - I(u,v)\right]^2,$$

Gewichtung

Intensitäten im verschobenen Fenster I(x+u,y+v)

Intensitäten im Fenster I(u,v)

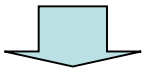
Harris Corner Detector (4)

Für kleine Verschiebungen ist S(x,y) approximierbar über eine Taylor-Näherung:

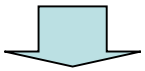
$$I(x+u,y+v) \approx I(u,v) + I_x(u,v)\cdot x + I_y(u,v)\cdot y$$

mit partiellen Ableitungen $I_x = \frac{\partial I}{\partial x}$, $I_y = \frac{\partial I}{\partial y}$:

$$S(x,y) = \sum_{u,v} w(u,v)\left[I(x+u,y+v) - I(u,v)\right]^2$$



$$S(x,y) \approx \sum_{u,v} w(u,v)[I(u,v) + I_x(u,v)x + I_y(u,v)y - I(u,v)]^2$$

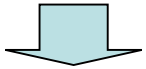


$$S(x,y) \approx \sum_{u,v} w(u,v)[I_x(u,v)x + I_y(u,v)y]^2$$

Harris Corner Detector (5)

Matrix-Darstellung führt zur sog. *Harris-Matrix* ***A****:*

$$S(x,y) \approx \sum_{u,v} w(u,v)[I_x(u,v)x + I_y(u,v)y]^2$$



$$S(x,y) \approx (x,y)\,\mathbf{A}\begin{pmatrix} x \\ y \end{pmatrix} \quad \textit{mit}$$

$$\mathbf{A} = \sum_{u,v} w(u,v)\begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \langle I_x^2 \rangle & \langle I_xI_y \rangle \\ \langle I_xI_y \rangle & \langle I_y^2 \rangle \end{bmatrix}$$

Nur Produkte der Gradienten

Die spitzen Klammern stehen für die mit *w*(*u*,*v*) gewichtete Summe über *u*, *v*

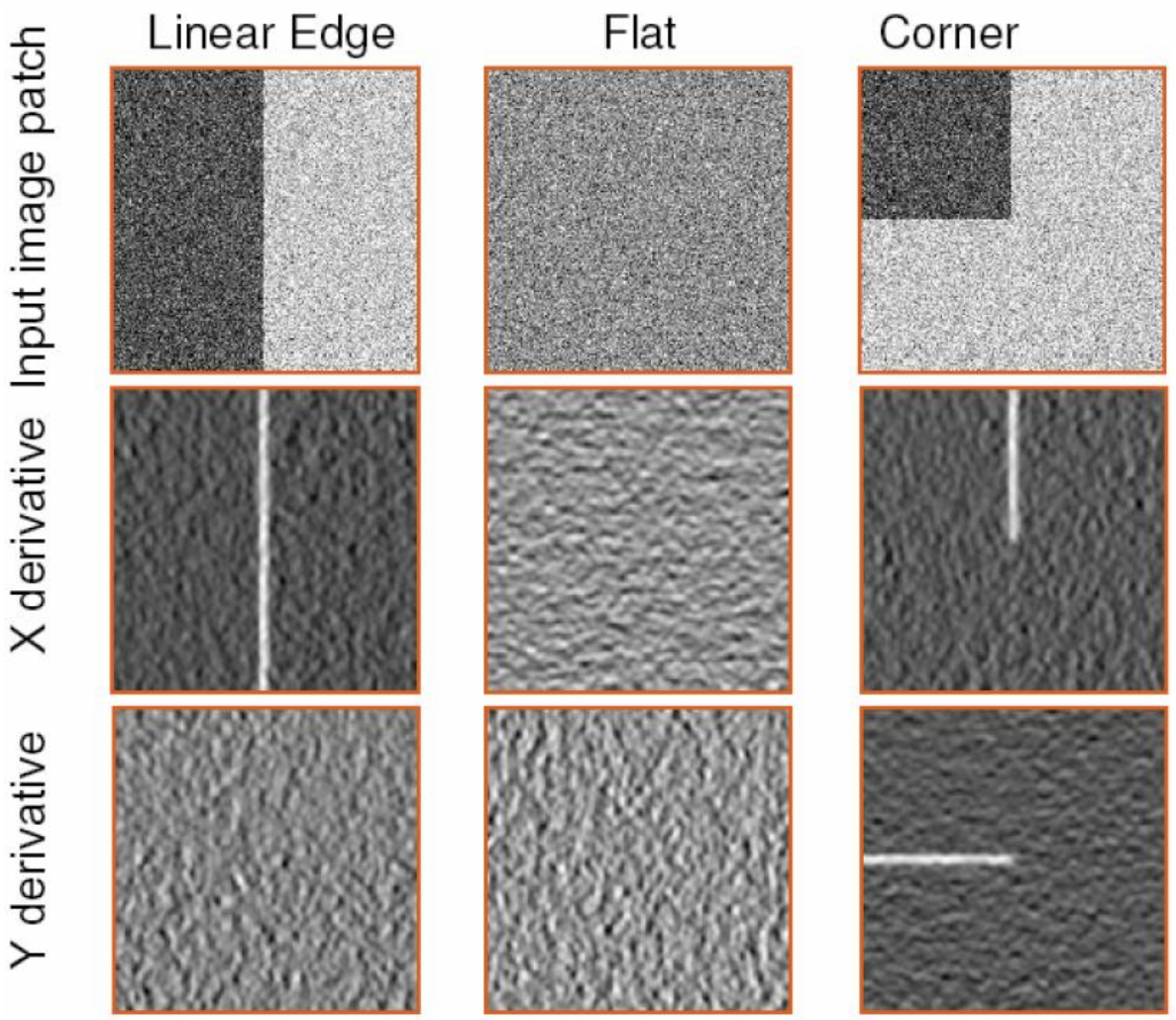
Harris Corner Detector (6)

Die partiellen Ableitungen I_x und I_y werden durch Kantenfilter approximiert:

$$I_x = \frac{\partial I}{\partial x} \approx I * (-1,0,1)$$

$$I_y = \frac{\partial I}{\partial y} \approx I * \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}$$

Harris Corner Detector (7)

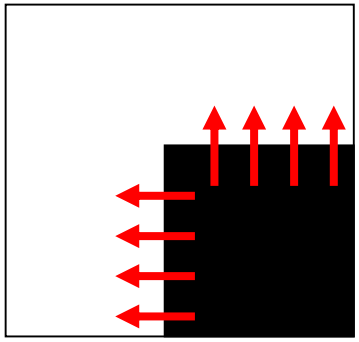


Bildquelle: Robert Collins, Penn State Univ.

Harris-Operator (8)

Die Harris-Matrix $\mathbf{A} = \sum_{u,v} w(u,v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix}$

1) für achsenparallele Ecke:



$$\mathbf{A} = \begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

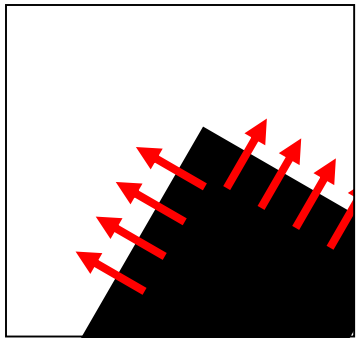
weil dominante Gradienten parallel zur x- oder y-Achse orientiert sind

→ sobald ein λ_i (i $\in$ {1,2} nahe Null ist, liegt keine Ecke vor

→ Ecken liegen vor, wenn λ_1 und λ_2 groß sind

Harris Corner Detector (9)

Die Harris-Matrix $\mathbf{A} = \sum_{u,v} w(u,v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix}$



2) für nicht-achsenparallele Ecke:

A ist symmetrische reellwertige Matrix

- ***A*** ist nach *Spektralsatz* darstellbar als

$$\mathbf{A} = \mathbf{Q} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \mathbf{Q}^T \quad \Rightarrow \quad \mathbf{A}\,\mathbf{q}_i = \lambda_i\,\mathbf{q}_i\,, \quad i = 1,2$$

mit orthogonaler Matrix ***Q*** mit allen Eigenvektoren $\boldsymbol{q}_i$ von ***A***

- Die Eigenwerte von ***A*** geben die Stärken der Intensitätsänderungen entlang der beiden orthogonalen Hauptgradientenrichtungen wieder

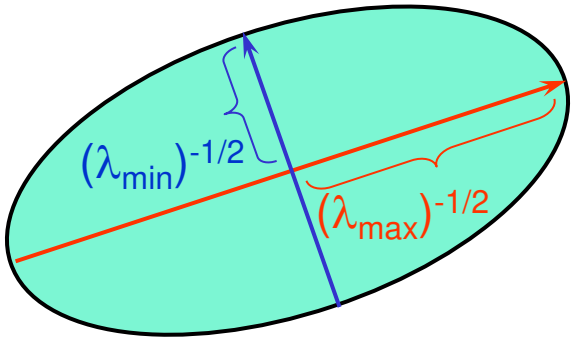
Harris Corner Detector (10)

Geg.: Harris-Matrix $\mathbf{A} = \mathbf{Q} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \mathbf{Q}^T$

mit orthogonaler Matrix ***Q*** mit allen Eigenvektoren $\boldsymbol{q}_i$ von ***A***

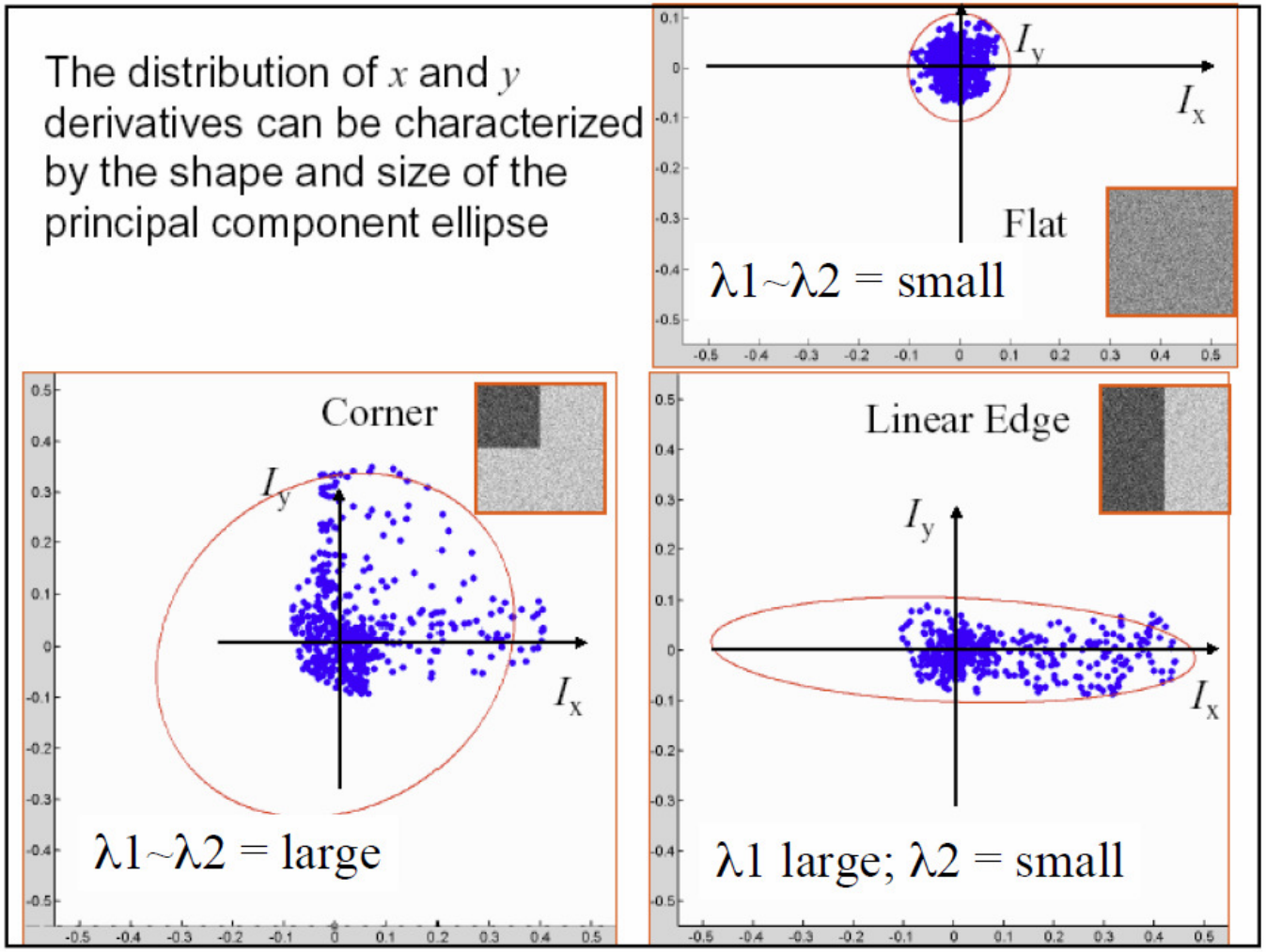
→ Visualisierung als Ellipse, deren Dimension und Orientierung durch die Eigenwerte und Eigenvektoren von ***A*** definiert sind:

Richtung mit schwächerem Gradienten



Richtung mit stärkeren Gradienten

Harris Corner Detector (11)



Bildquelle: Robert Collins, Penn State Univ.

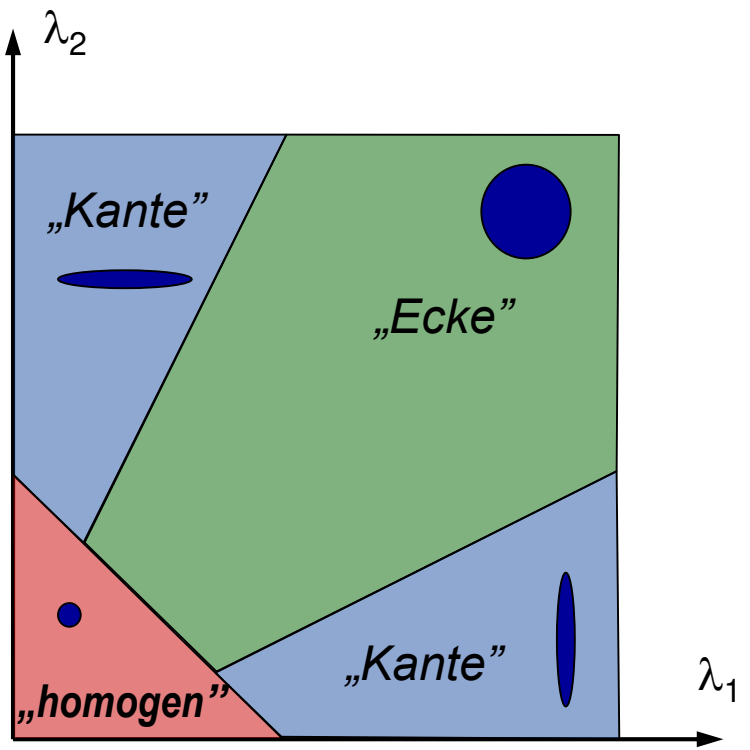
Harris Corner Detector (12)

- Eine Ecke ist „ein Punkt, der zwei unterschiedlich orientierte signifikante Gradienten in der lokalen Nachbarschaft zeigt“ (vgl. Folie 4)

→ bei einer Ecke muss **A** zwei große Eigenwerte λ_1 und λ_2 aufweisen

→ Klassifikation:

- wenn $\lambda_1 \approx 0$ und $\lambda_2 \approx 0$,
  dann ein homogener Bereich vor
- wenn $\lambda_1 \approx 0$ und $\lambda_2 \gg 0$,
  dann liegt eine Kante vor
- if $\lambda_1 \gg 0$ und $\lambda_2 \gg 0$,
  dann liegt eine Ecke vor



Harris Corner Detector (13)

Anstelle der aufwändigen Berechnung der Eigenwerte schlagen Harris und Stephens die Berechnung einer Antwortfunktion (engl. Response Function) *R* vor:

- R basiert auf der Berechnung der Determinante und der Spur von ***A***: *

$$R = det(\mathbf{A}) - \kappa\, trace^2(\mathbf{A}).$$

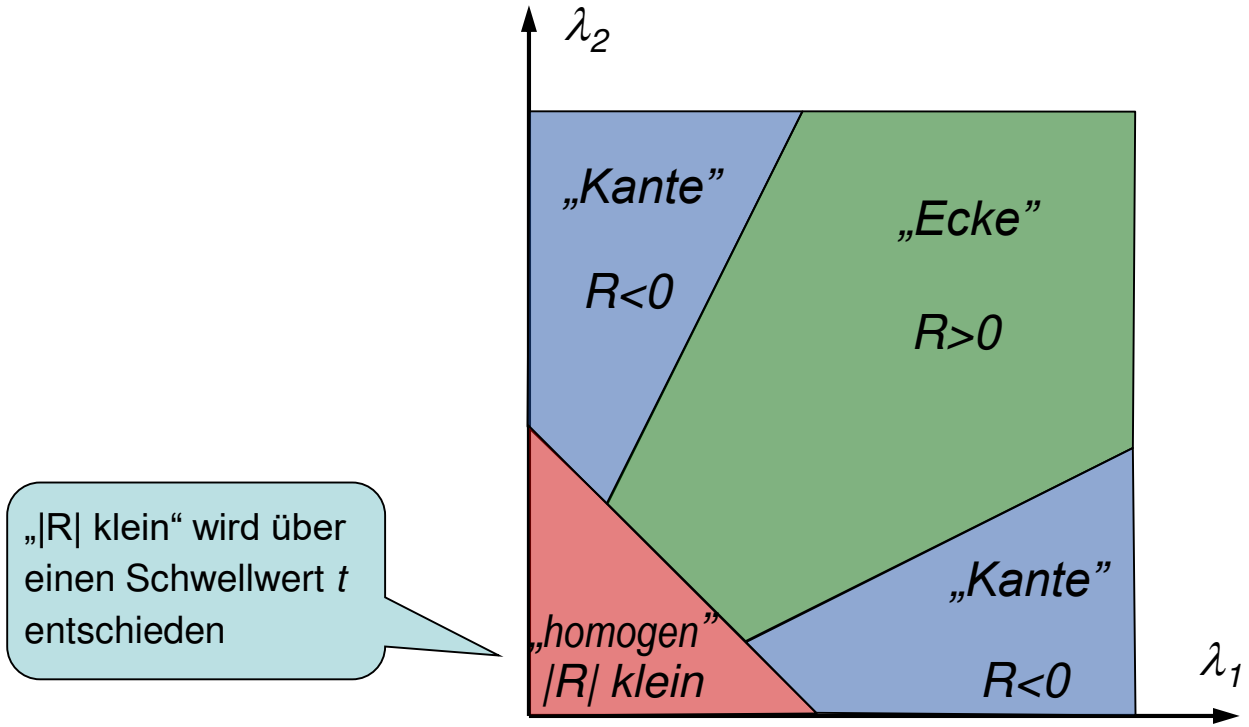
- κ ist heuristischer Parameter mit $0.04 \leq \kappa \leq 0.15$ (nach Harris und Stephens)

---

* *Die Determinante einer 2x2-Matrix* $\mathbf{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ *ist ad-bc.*
Die Spur einer nxn-Matrix ***A*** *ist die Summe der Elemente auf der Hauptdiagonalen.*

Harris Corner Detector (14)

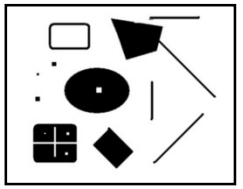
Ergebnisinterpretation über Antwortfunktion $R = \det(\boldsymbol{A}) - \kappa \, \mathrm{trace}^2(\boldsymbol{A})$:



Bildquelle: Weizmann Inst. of Science, Computer Vision Lab

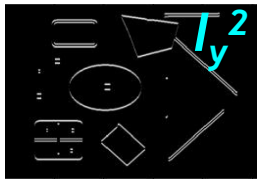
Harris Corner Detector (15)

Zusammenfassung:



1) Ableitung der Harris Matrix

$$\mathbf{A} = \begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix}$$



2) Berechnung der Antwortfunktion

$R = \det(\mathbf{A}) - \kappa \, \mathrm{trace}^2(\mathbf{A})$

$= \langle I_x^2 \rangle \cdot \langle I_y^2 \rangle - \langle I_x \cdot I_y \rangle^2 - \kappa \cdot (\langle I_x^2 \rangle + \langle I_y^2 \rangle)^2$

3) Optional: Non-Maxima-Unterdrückung
(vgl. Canny-Operator)

Bildquelle:
K. Mikolajczyk, Dept of Engineering Science, Univ. of Oxford

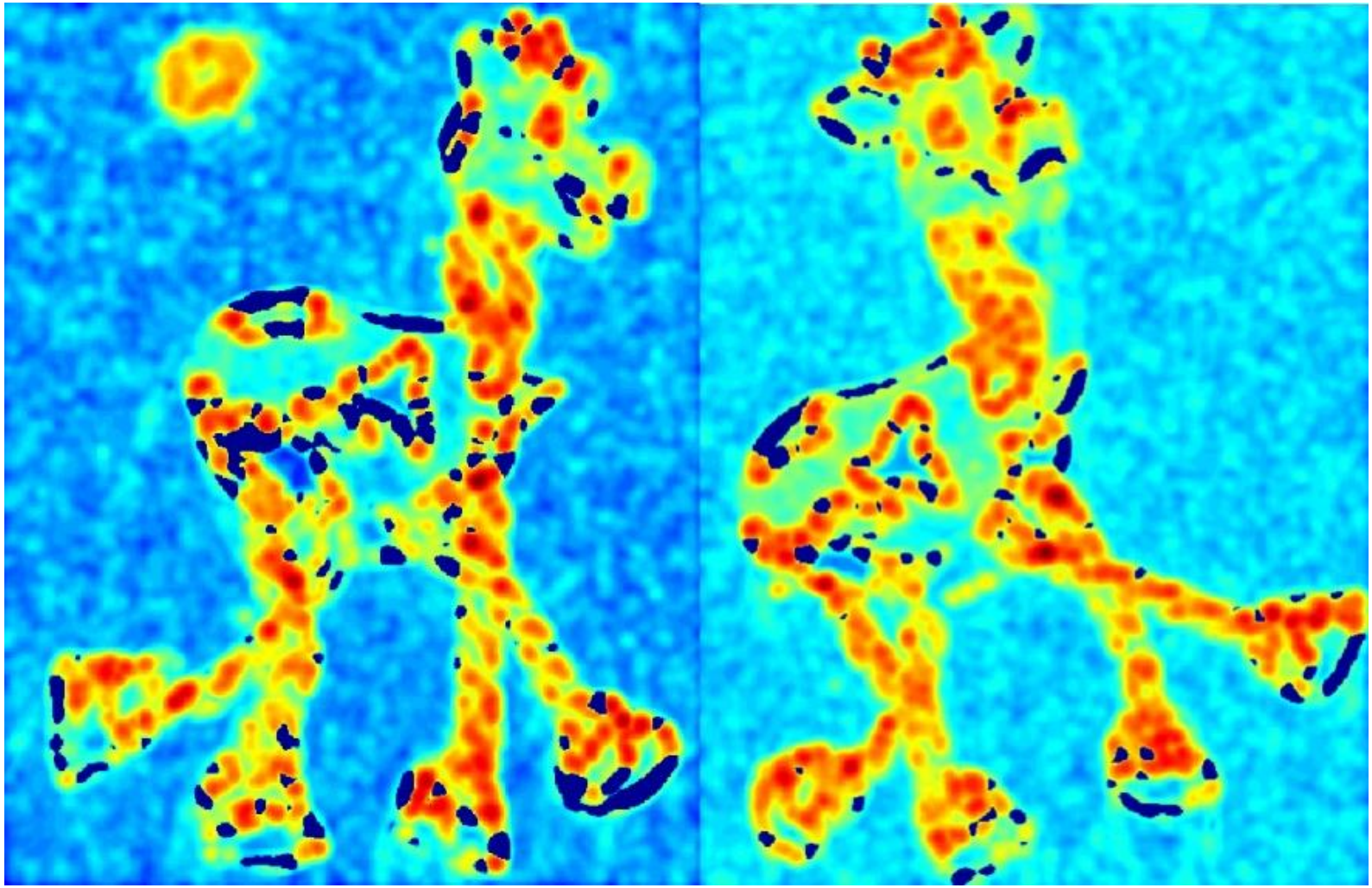
Harris Corner Detector – 1. Beispiel (1)

Eingabebilder:

Bildquelle: D. Frolova, D. Simakov, Weizmann Inst. of Science, Computer Vision Lab

Harris Corner Detector – 1. Beispiel (2)

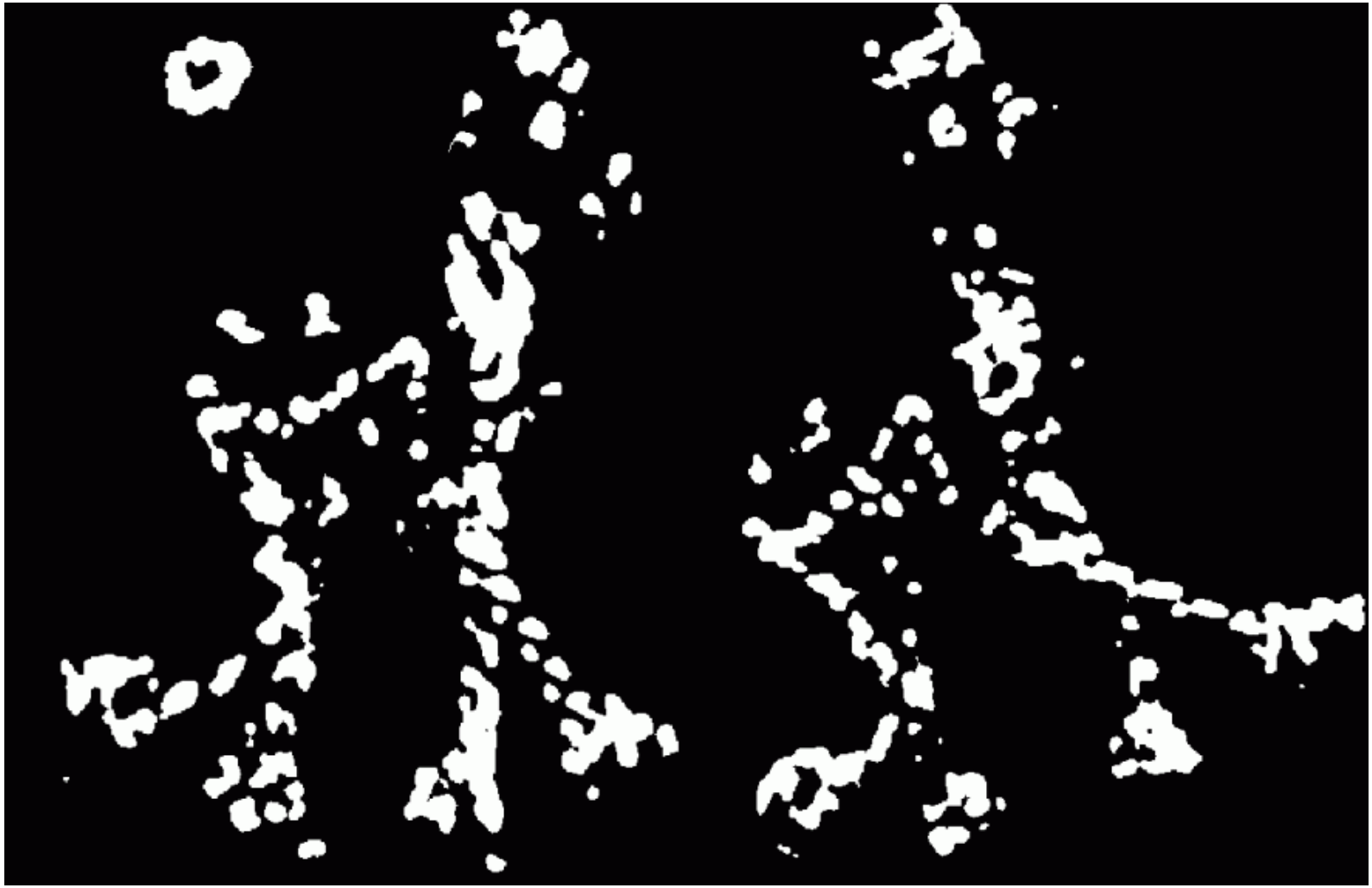
Antwortfunktion:



Bildquelle: D. Frolova, D. Simakov, Weizmann Inst. of Science, Computer Vision Lab

Harris Corner Detector – 1. Beispiel (3)

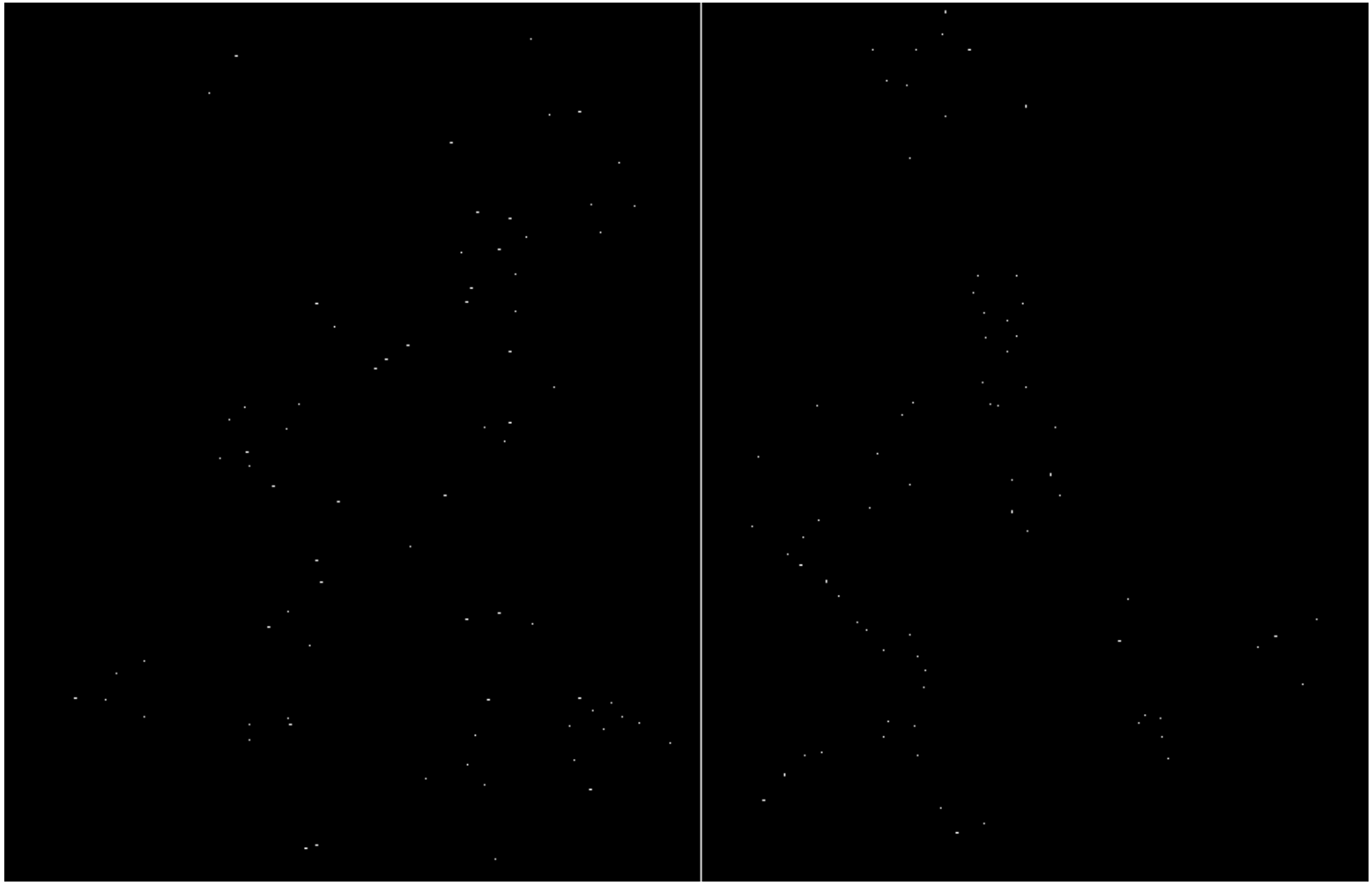
Binarisierung über Schwellwert:



Bildquelle: D. Frolova, D. Simakov, Weizmann Inst. of Science, Computer Vision Lab

Harris Corner Detector – 1. Beispiel (4)

Non-Maxima-Unterdrückung:



Bildquelle: D. Frolova, D. Simakov, Weizmann Inst. of Science, Computer Vision Lab

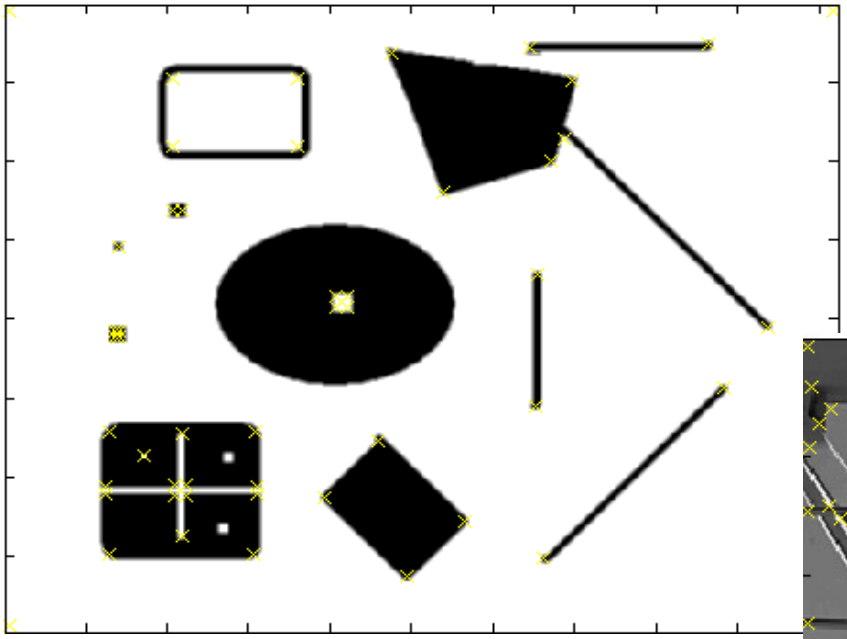
Harris Corner Detector – 1. Beispiel (5)

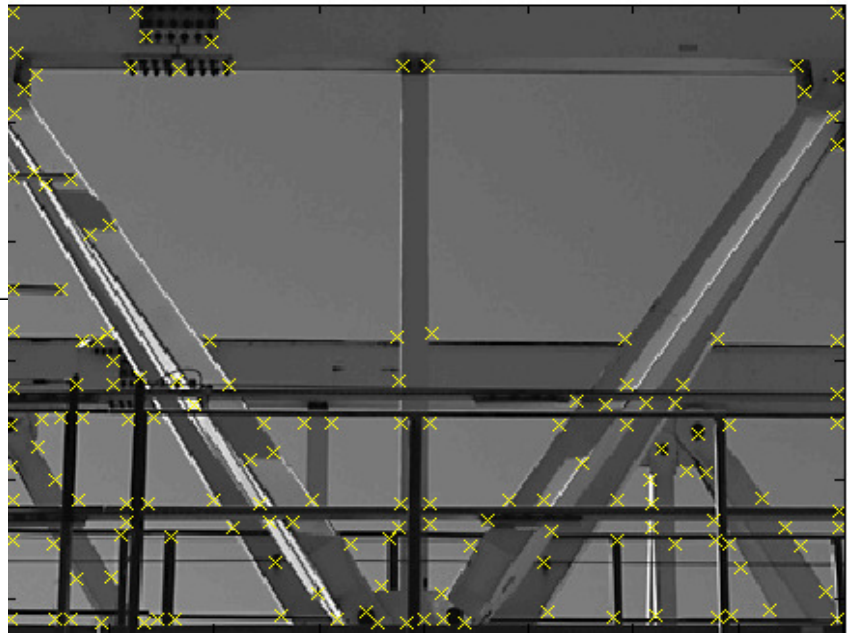
Extrahierte markante Punkte durch *Harris Corner Detector*:

Bildquelle: D. Frolova, D. Simakov, Weizmann Inst. of Science, Computer Vision Lab

Harris Corner Detector – 2. Beispiel

Extrahierte markante Punkte durch *Harris Corner Detector*:





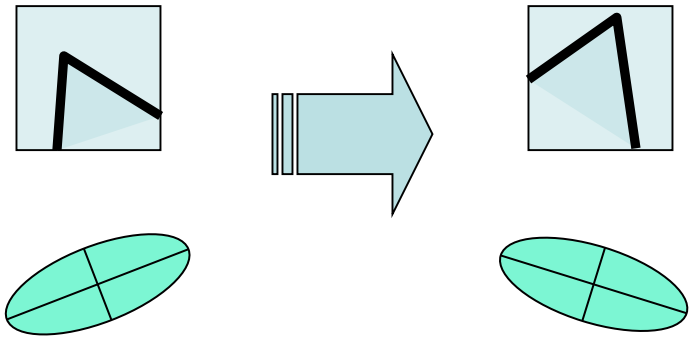
Bildquelle: K. Mikolajczyk, Dept of Engineering Science, Univ. of Oxford

Harris Corner Detector – 3. Beispiel

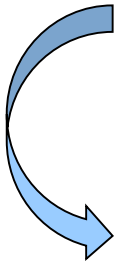
Extrahierte markante Punkte durch *Harris Corner Detector*:

Bildquelle: K. Graumann, University of Texas at Austin

Robustheit des Harris-Detectors (1)



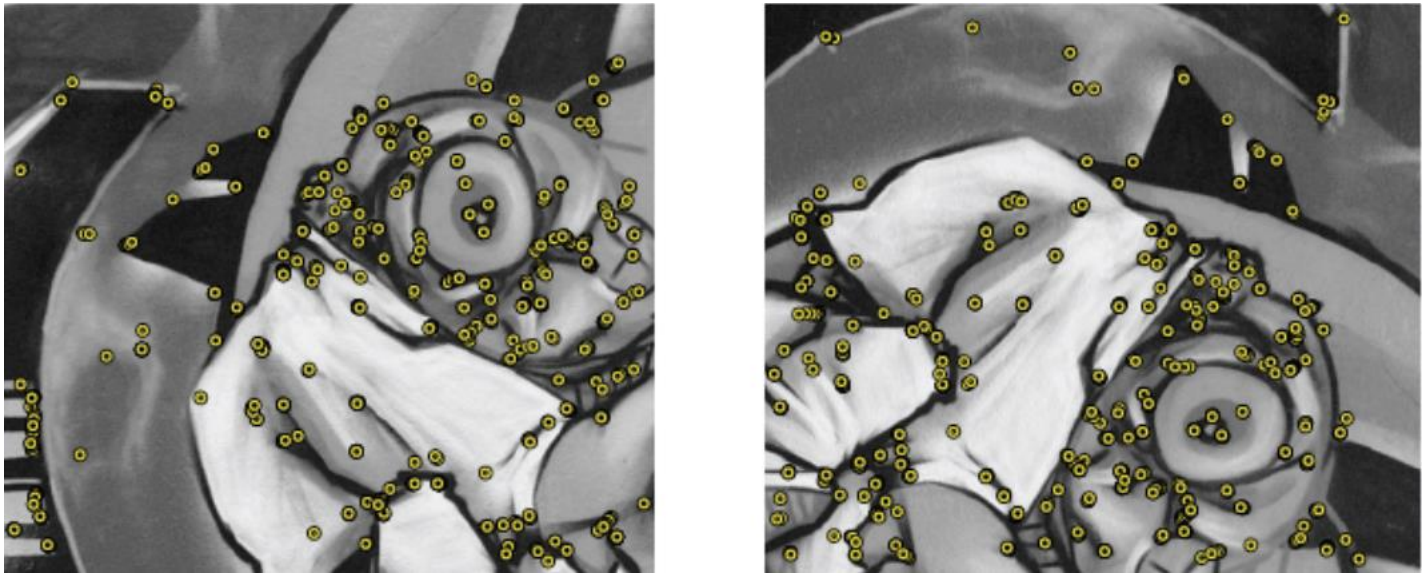
Ellipse rotiert,
aber ihre Form (d.h. die Eigenwerte) bleibt gleich



Antwortfunktion *R* ist invariant bzgl. Rotation in der Bildebene

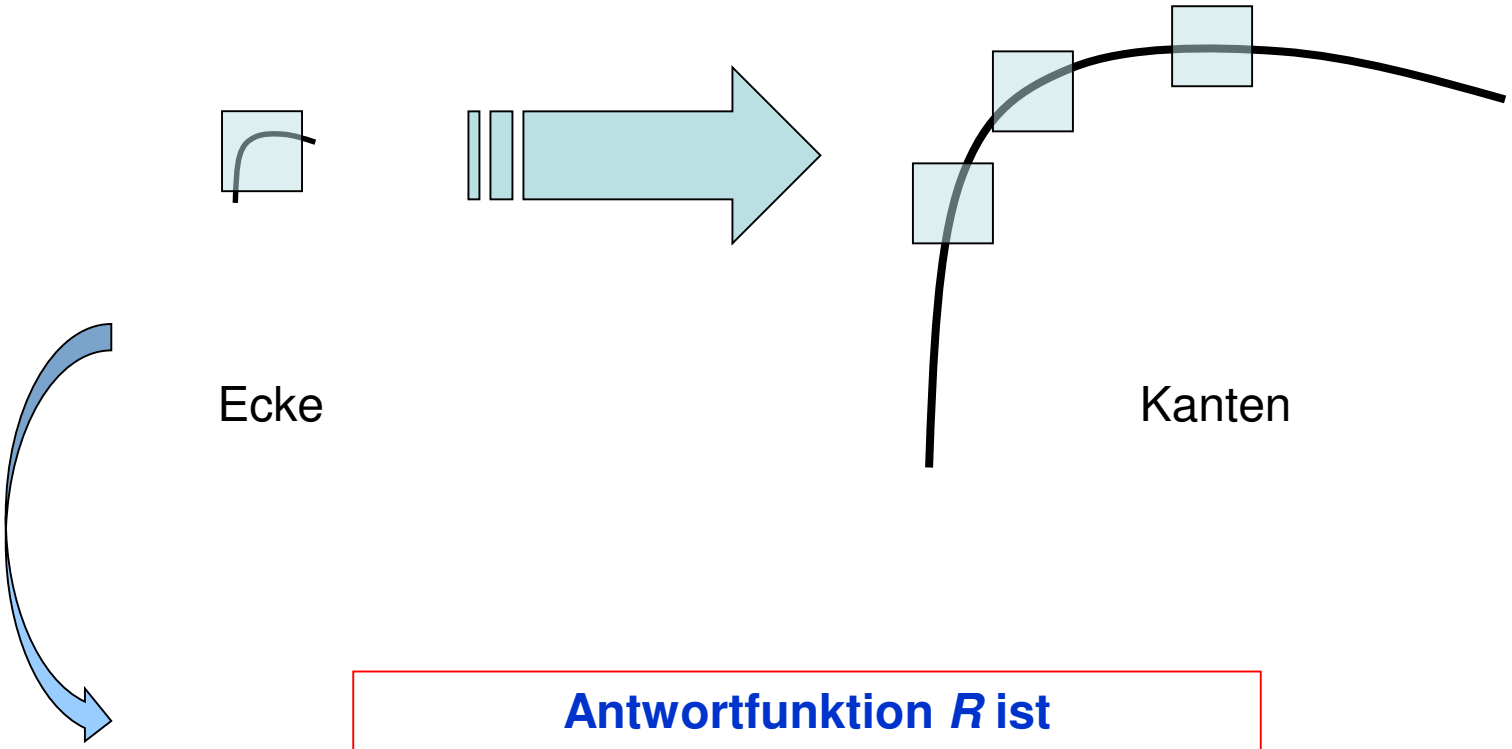
Robustheit des Harris-Detectors (2)

Rotationsinvarianz beim *Harris Corner Detector*:



Bildquelle: K. Mikolajczyk, University of Texas at Austin

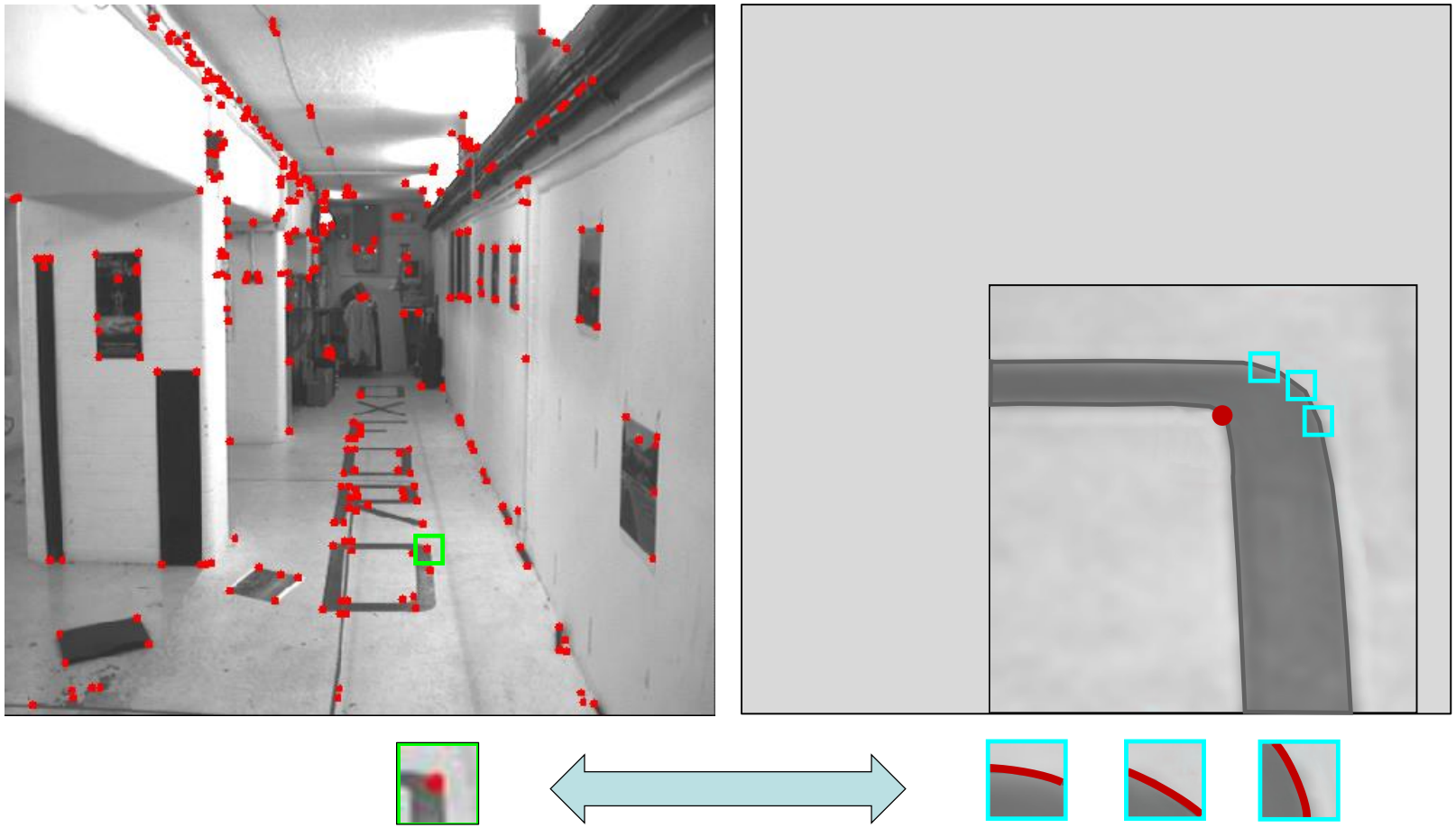
Robustheit des Harris-Detectors (3)



Antwortfunktion *R* ist
nicht invariant bzgl. Skalierung

Robustheit des Harris-Detectors (4)

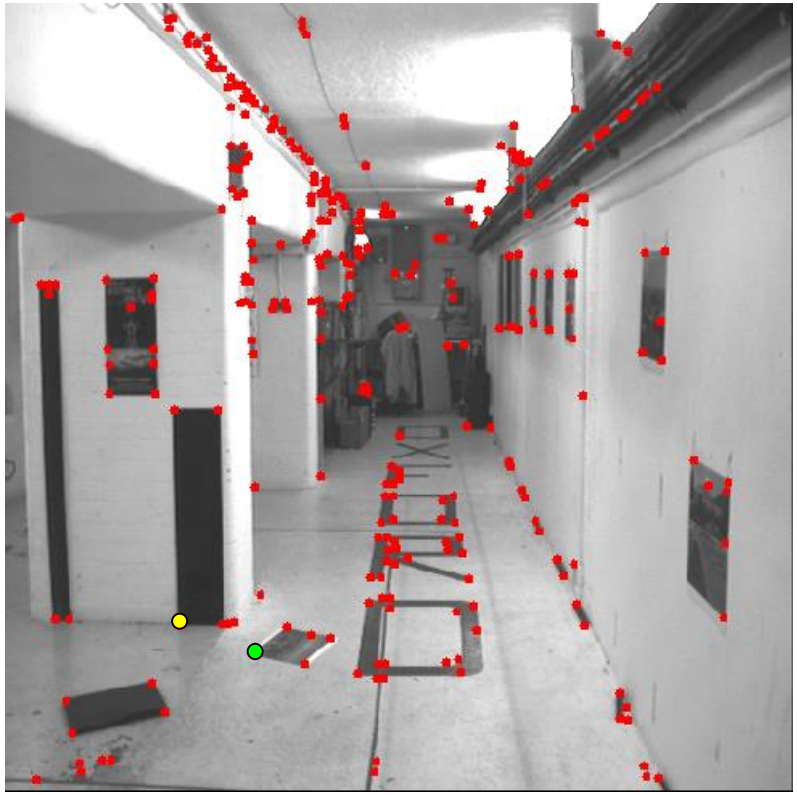
Fehlende Skalierungsinvarianz beim *Harris Corner Detector*:

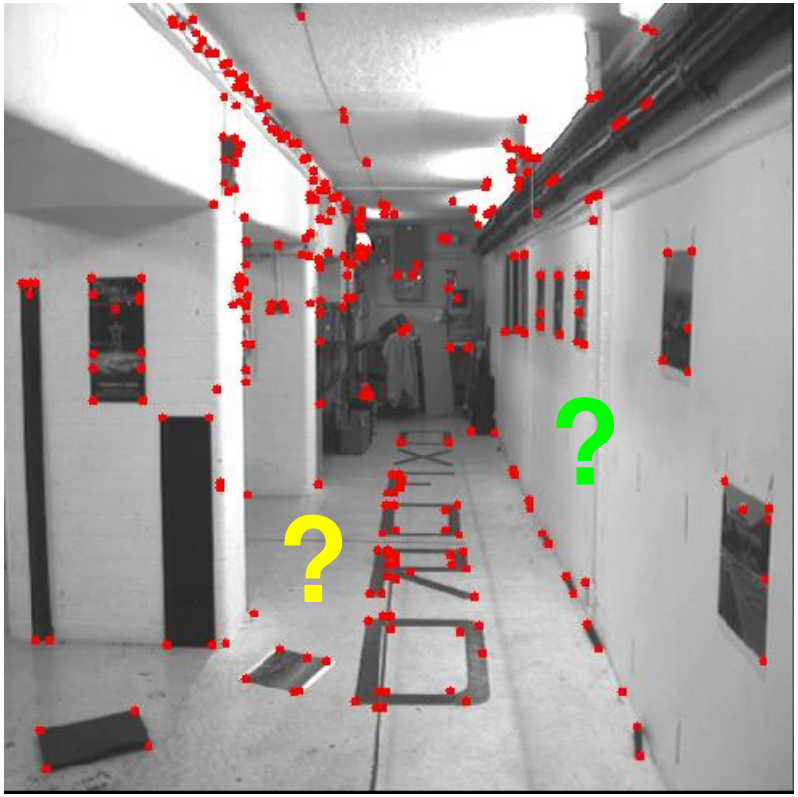


Bildquelle: K. Graumann, University of Texas at Austin

Identifikation von Interest Points (1)

Wie können markante Punkte identifiziert und zugeordnet werden?

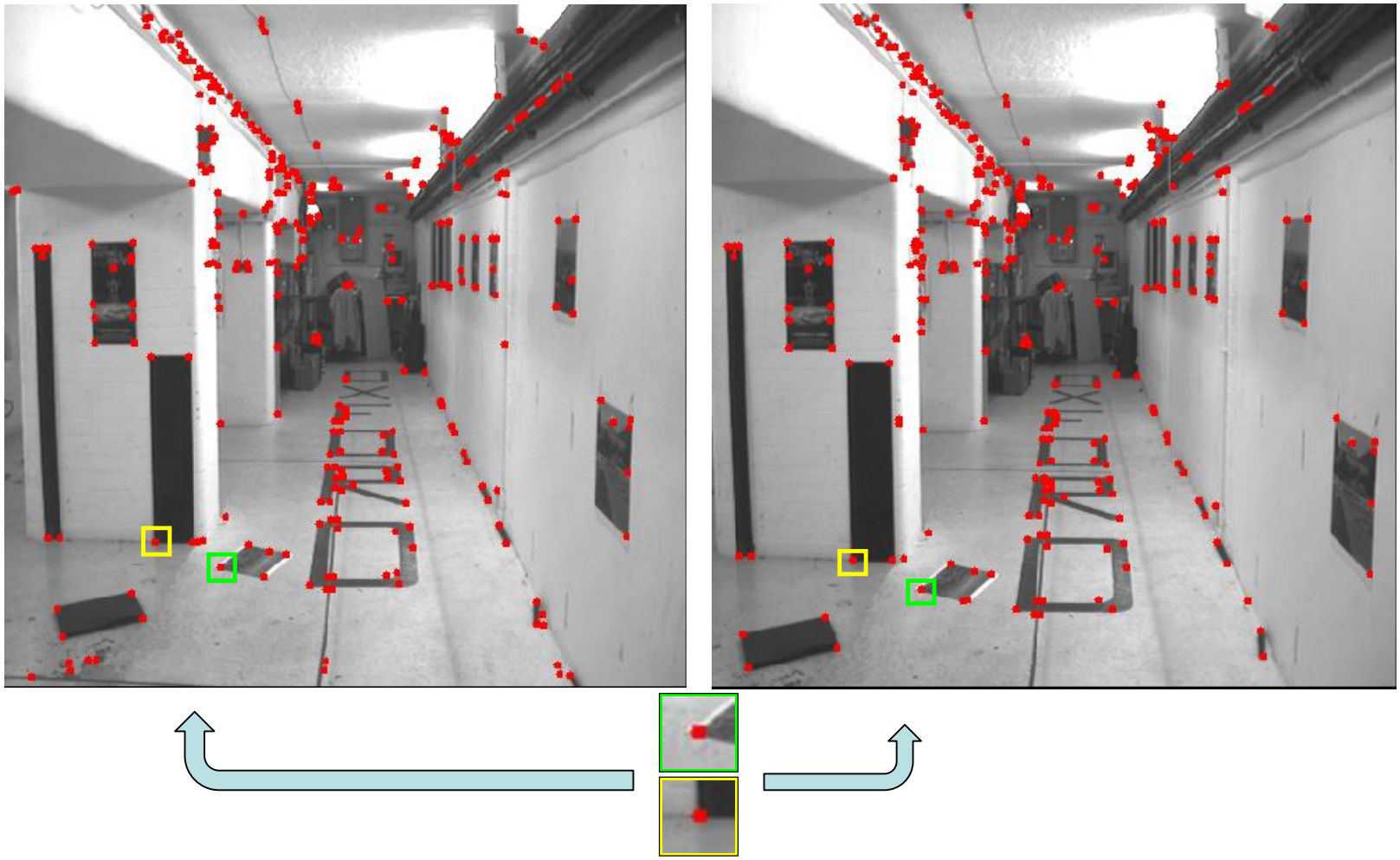




Bildquelle: K. Graumann, University of Texas at Austin

Identifikation von Interest Points (2)

Beschreibung der Region um markante Punkte nötig → Interest Regions

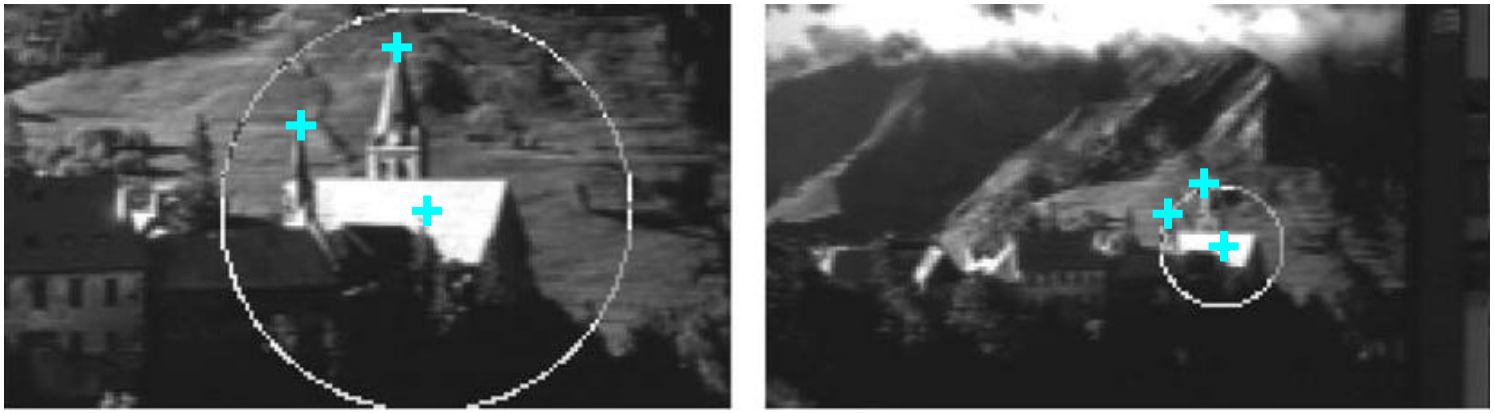


Bildquelle: K. Graumann, University of Texas at Austin

Zur robusten Erkennung und Identifikation von Interest Points

- Zur robuste Identifikation von Interest Points (IP) sind

skaleninvariante Deskriptoren von Regionen von Nutzen



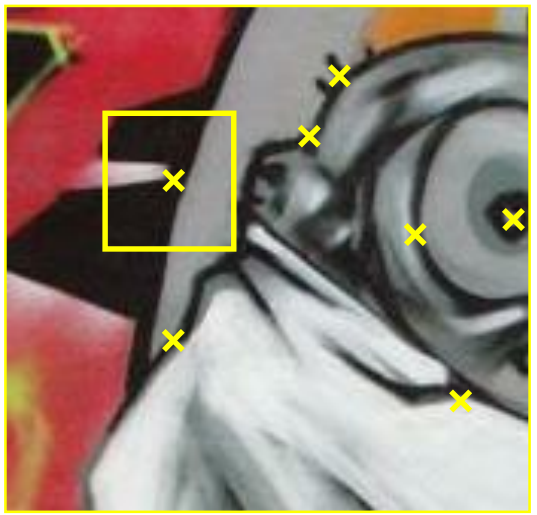
Bildquelle: K. Mikolajczyk, C. Schmid (2004)

- Neue Herausforderung:

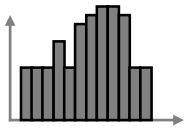
Erkennung von skaleninvarianten *Interest Regions*

Exhaustiver Multiskalen-Vergleich von Regionen (1)

Naiver Ansatz: exhaustive Suche in Multiskalen-Ansatz zum Vergleich von Deskriptoren bei unterschiedl. Regionengröße



f_A



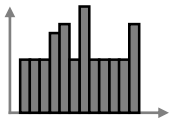
e.g. normalized intensities

Similarity measure

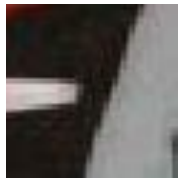
$\neq$

$d(f_A, f_B)$

f_B



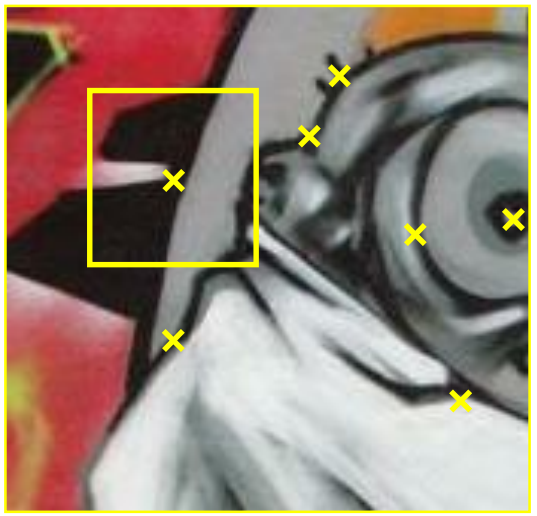
e.g. normalized intensities

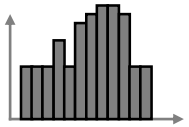


Folie adaptiert von K. Mikolajczyk

Exhaustiver Multiskalen-Vergleich von Regionen (2)

Naiver Ansatz: exhaustive Suche in Multiskalen-Ansatz zum Vergleich von Deskriptoren bei unterschiedl. Regionengröße



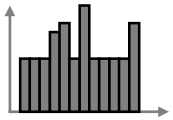


f_A

Similarity measure

$\neq$

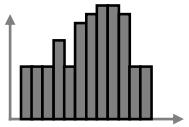
$d(f_A, f_B)$



f_B

Exhaustiver Multiskalen-Vergleich von Regionen (3)

Naiver Ansatz: exhaustive Suche in Multiskalen-Ansatz zum Vergleich von Deskriptoren bei unterschiedl. Regionengröße

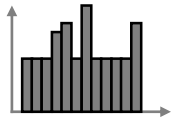


f_A

Similarity measure

$\neq$

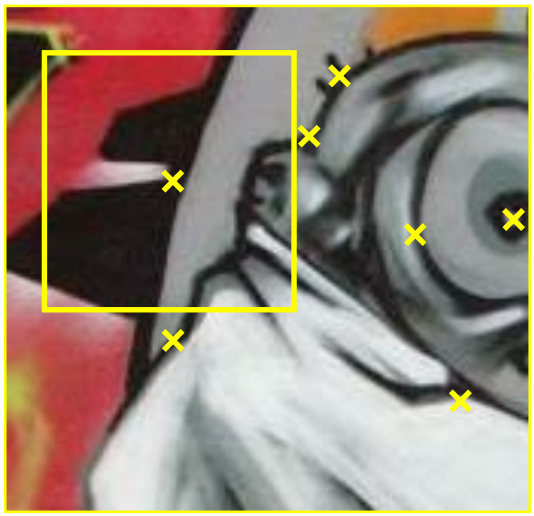
$d(f_A, f_B)$

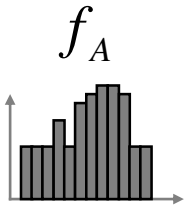


f_B

Exhaustiver Multiskalen-Vergleich von Regionen (4)

Naiver Ansatz: exhaustive Suche in Multiskalen-Ansatz zum Vergleich von Deskriptoren bei unterschiedl. Regionengröße

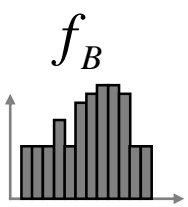




Similarity measure

$=$

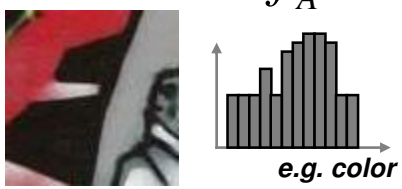
$d(f_A, f_B)$



Exhaustiver Multiskalen-Vergleich von Regionen (5)

Naiver Ansatz der exhaustiven Suche:

- ineffizient
- ungeeignet für Bild-Retrieval in großen Databanken
- ungeeignet für Objekterkennung und Objektverfolgung



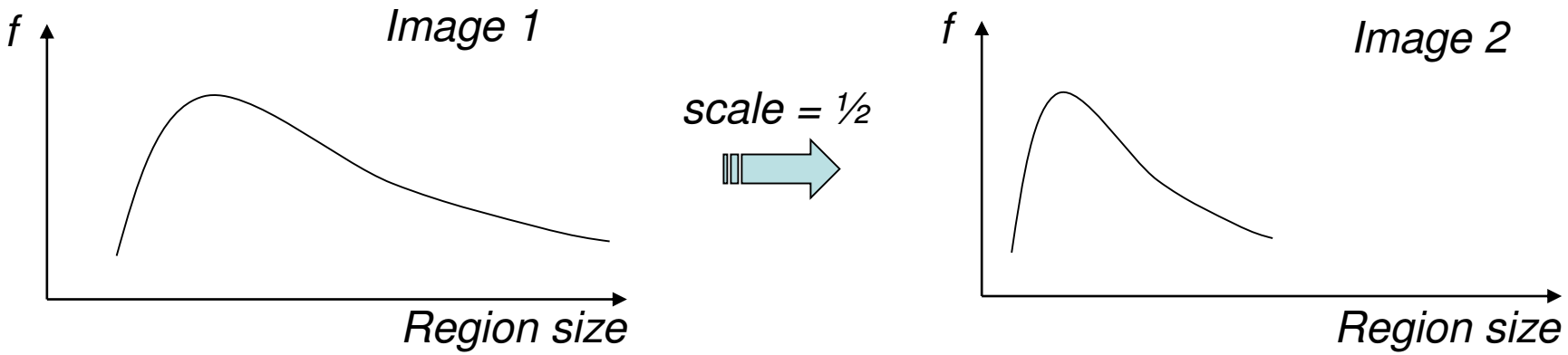
Folie adaptiert von K. Mikolajczyk

Automatische Skalenwahl (1)

Lösung:

- Einsatz einer auf Regionen skaleninvarianten Bewertungsfunktion *f*
- Bspl.: Intensitätsmittelwert

  (ist gleich für *korrespondierende* Regionen – trotz unterschiedlicher Abbildungsgröße)
- Für einen Bildpunkt zeigt sich *f* als Funktion der Regionengröße:



Folie adaptiert von K. Grauman

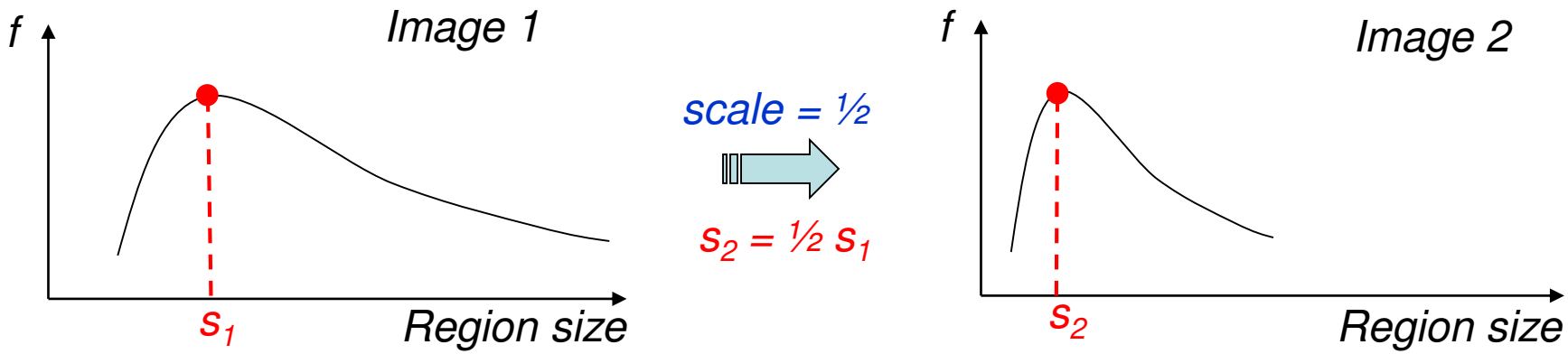
Automatische Skalenwahl (2)

Lösung:

- Auswahl eines lokalen Maximums der Funktion *f*

→ Beobachtung: Regionengröße mit maximalem *f*-Wert ist skaleninvariant

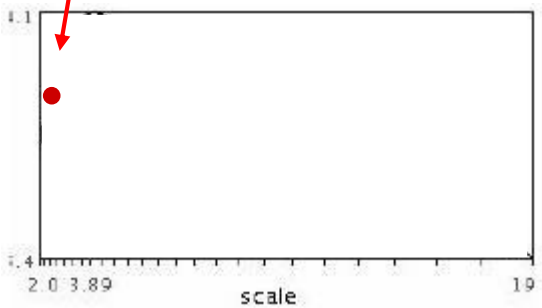
Wichtig: die skaleninvariante charakteristische Regiongröße wird in jedem Bild unabhängig gefunden



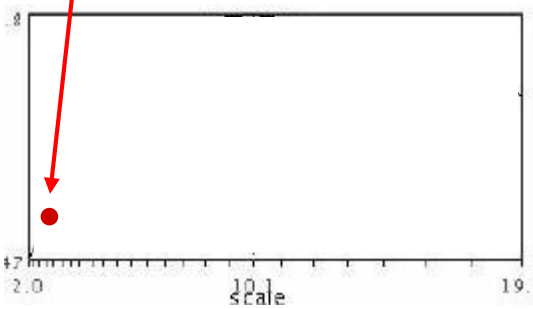
Folie adaptiert von K. Grauman

Automatische Skalenwahl – Bspl. (1)

Beispiel: Funktionswerte mit wachsender Skala



$$f(I_{i_1 \ldots i_m}(x,\sigma))$$

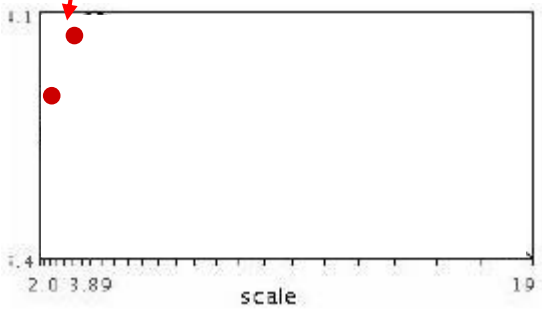


$$f(I_{i_1 \ldots i_m}(x',\sigma))$$

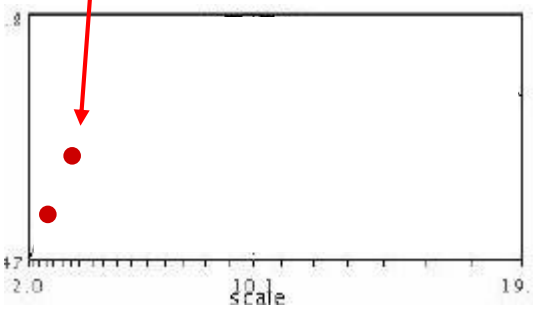
Folie adaptiert von K. Mikolajczyk

Automatische Skalenwahl – Bspl. (2)

Beispiel: Funktionswerte mit wachsender Skala



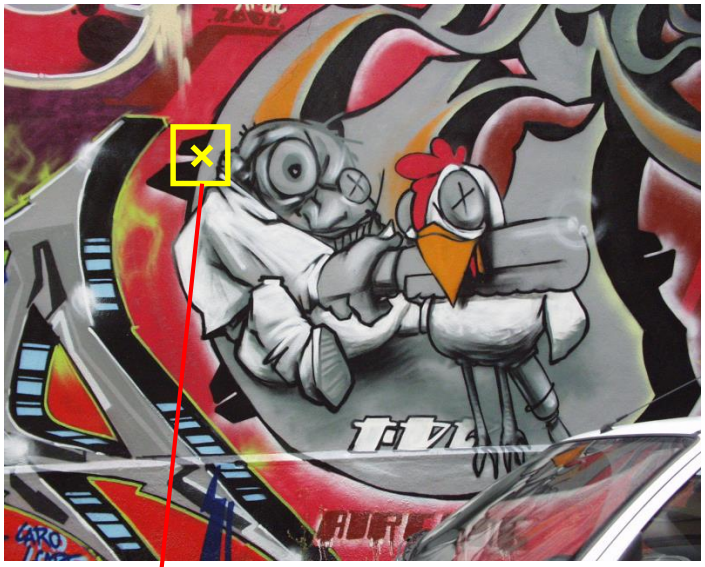
$f(I_{i_1 \ldots i_m}(x,\sigma))$

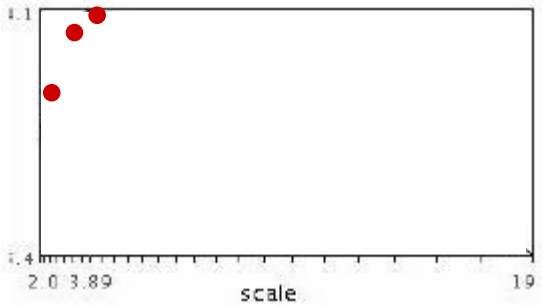


$f(I_{i_1 \ldots i_m}(x',\sigma))$

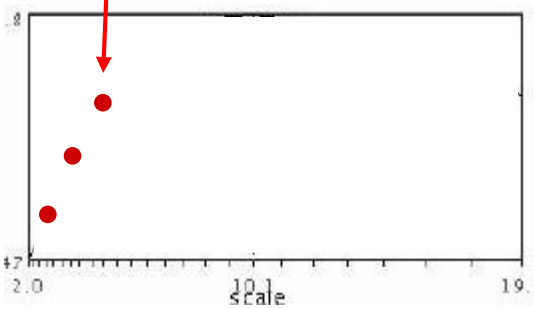
Automatische Skalenwahl – Bspl. (3)

Beispiel: Funktionswerte mit wachsender Skala





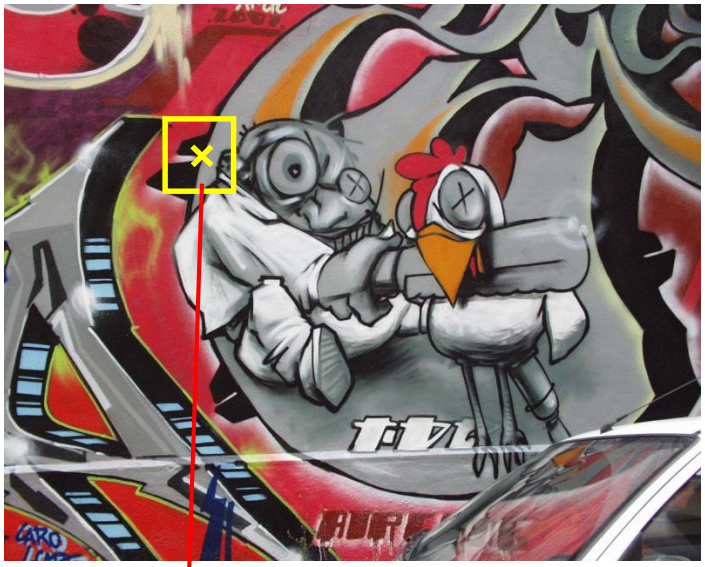
$f(I_{i_1 \ldots i_m}(x,\sigma))$

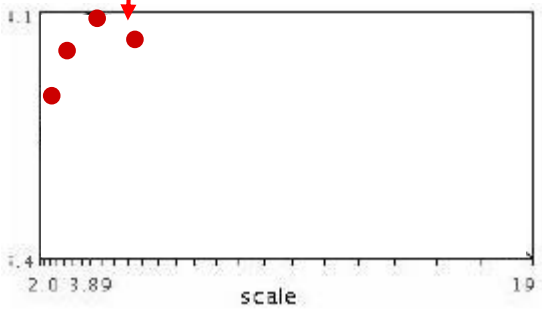


$f(I_{i_1 \ldots i_m}(x',\sigma))$

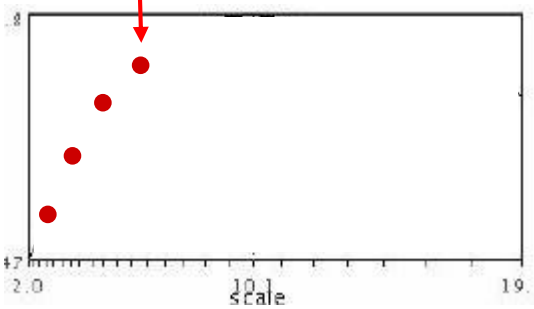
Automatische Skalenwahl – Bspl. (4)

Beispiel: Funktionswerte mit wachsender Skala





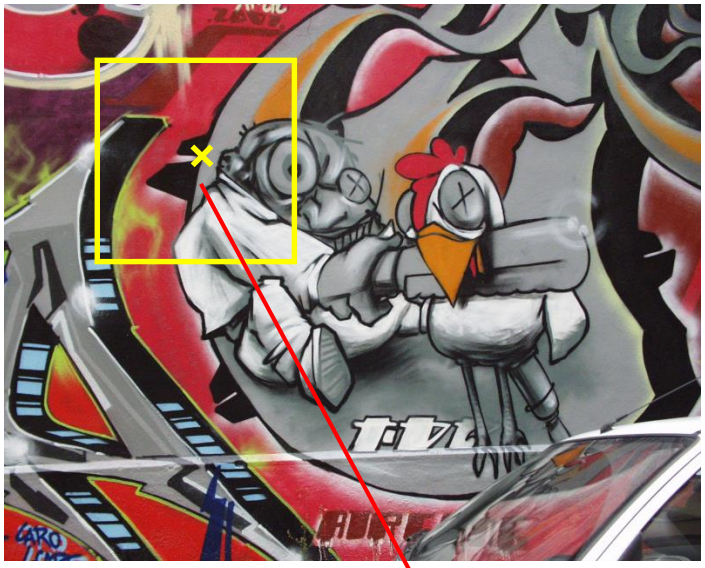
$f(I_{i_1 \ldots i_m}(x,\sigma))$

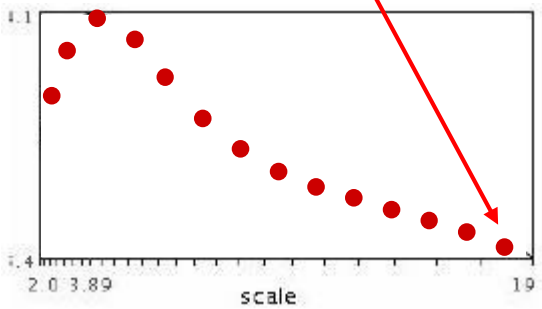


$f(I_{i_1 \ldots i_m}(x',\sigma))$

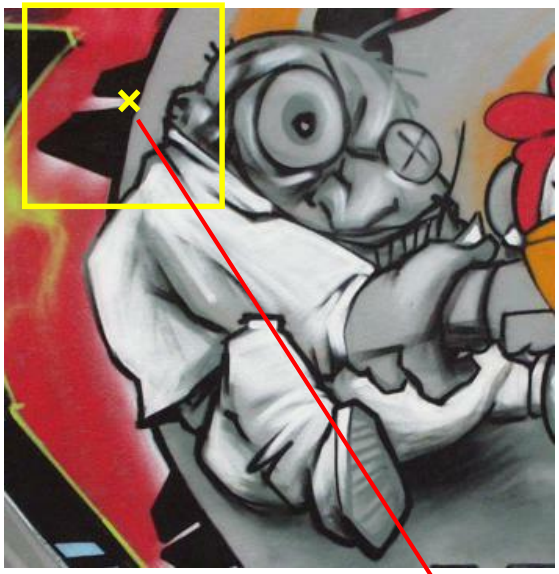
Automatische Skalenwahl – Bspl. (5)

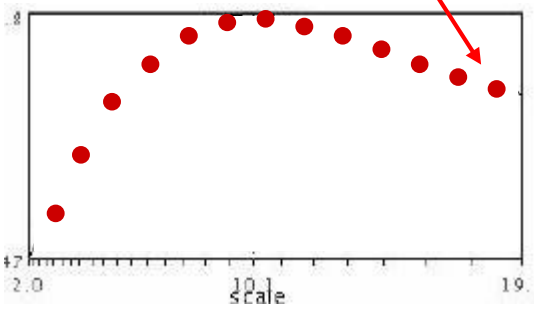
Beispiel: Funktionswerte mit wachsender Skala





$f(I_{i_1 \ldots i_m}(x,\sigma))$

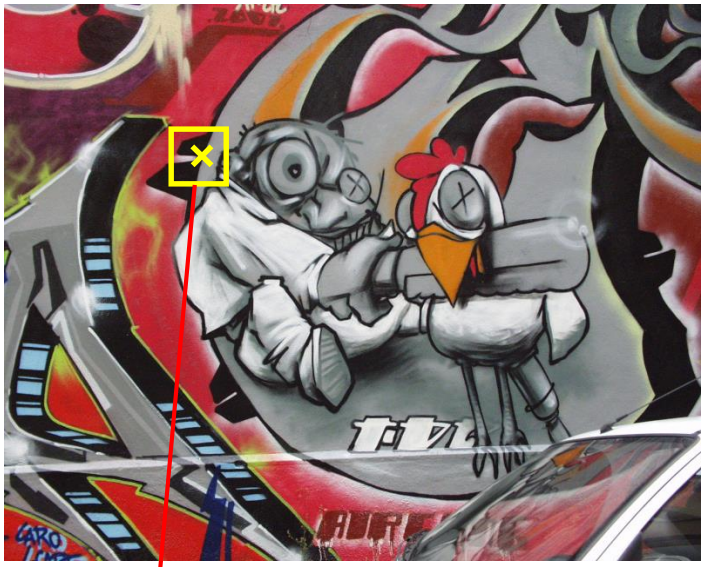


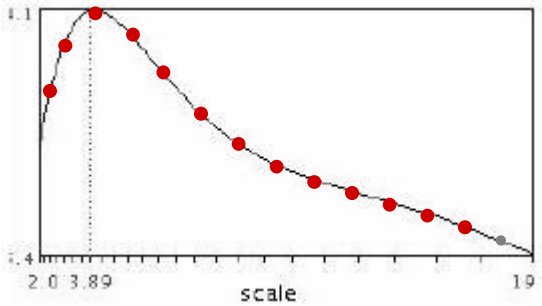


$f(I_{i_1 \ldots i_m}(x',\sigma))$

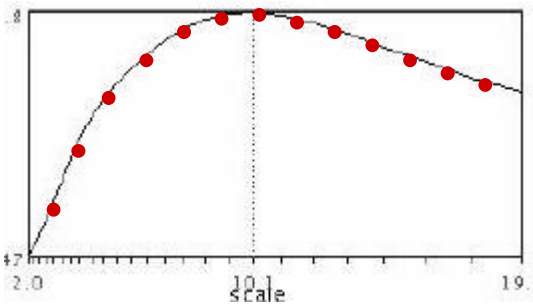
Automatische Skalenwahl – Bspl. (6)

Beispiel: Funktionswerte mit wachsender Skala





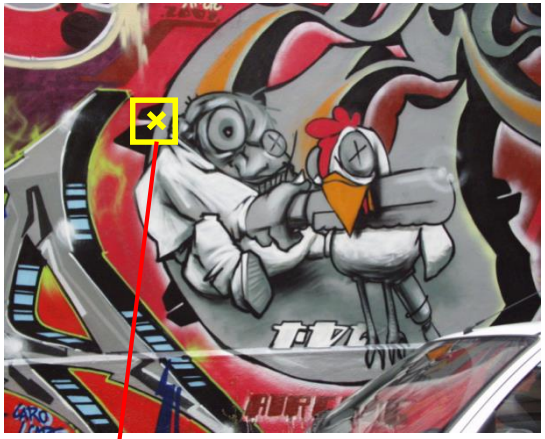
$f(I_{i_1 \ldots i_m}(x,\sigma))$

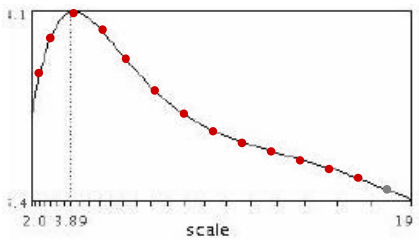


$f(I_{i_1 \ldots i_m}(x',\sigma))$

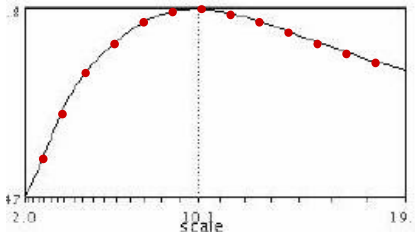
Automatische Skalenwahl – Bspl. (7)

Beispiel: Finale Skalierung auf feste Regionengröße

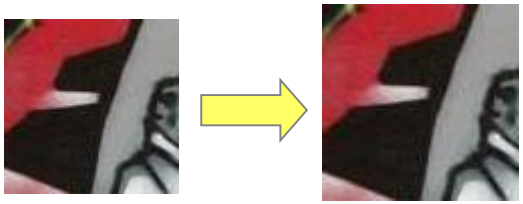


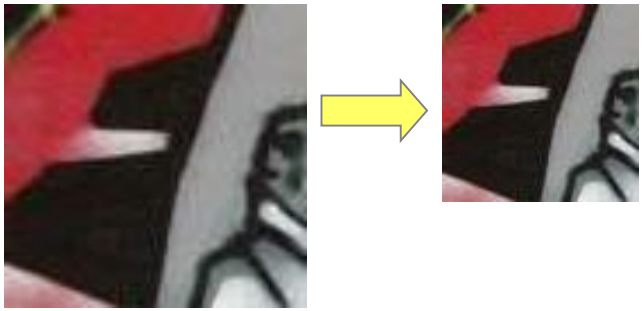


$f(I_{i_1 \ldots i_m}(x,\sigma))$



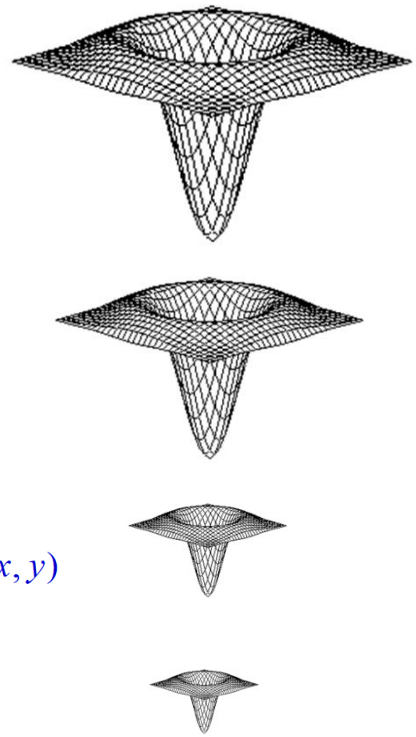
$f(I_{i_1 \ldots i_m}(x',\sigma))$





Wahl der skaleninvarianten Funktion *f*

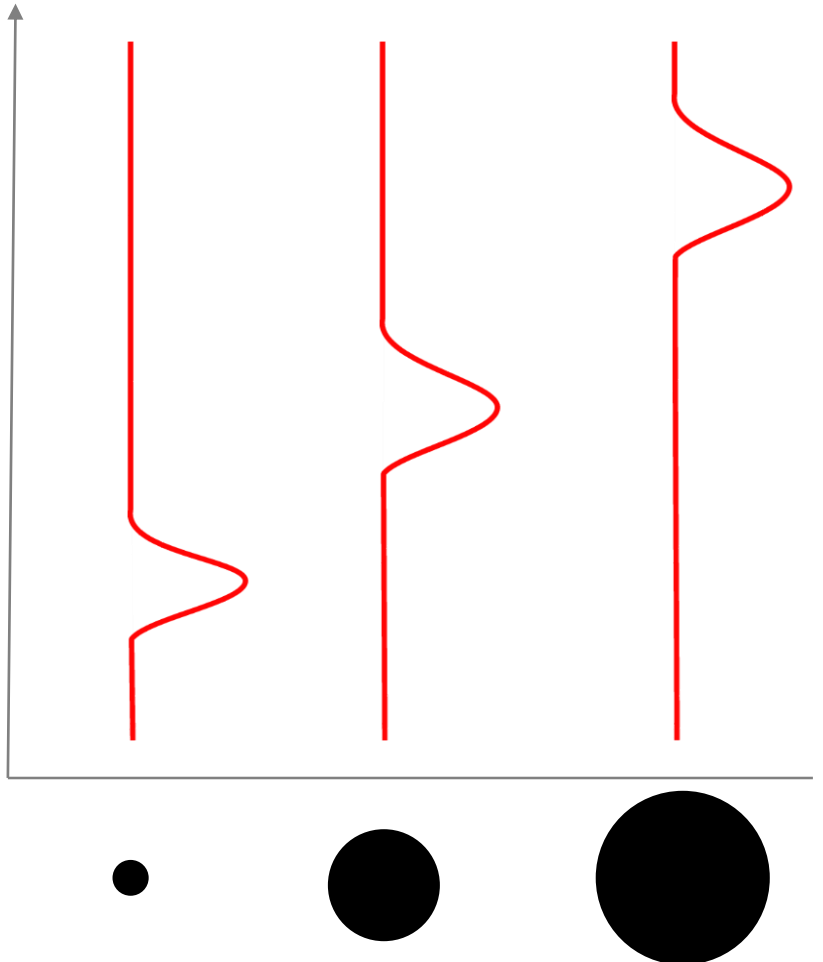
→ Laplacian-of-Gaussian (LoG) (s. Vorl. 3) als „*Blob detector*“



$$LoG(x,y) = \nabla^2 f_{G,\sigma}(x,y)$$

$$= \frac{\partial^2 f_{G,\sigma}}{\partial x^2}(x,y) + \frac{\partial^2 f_{G,\sigma}}{\partial y^2}(x,y)$$

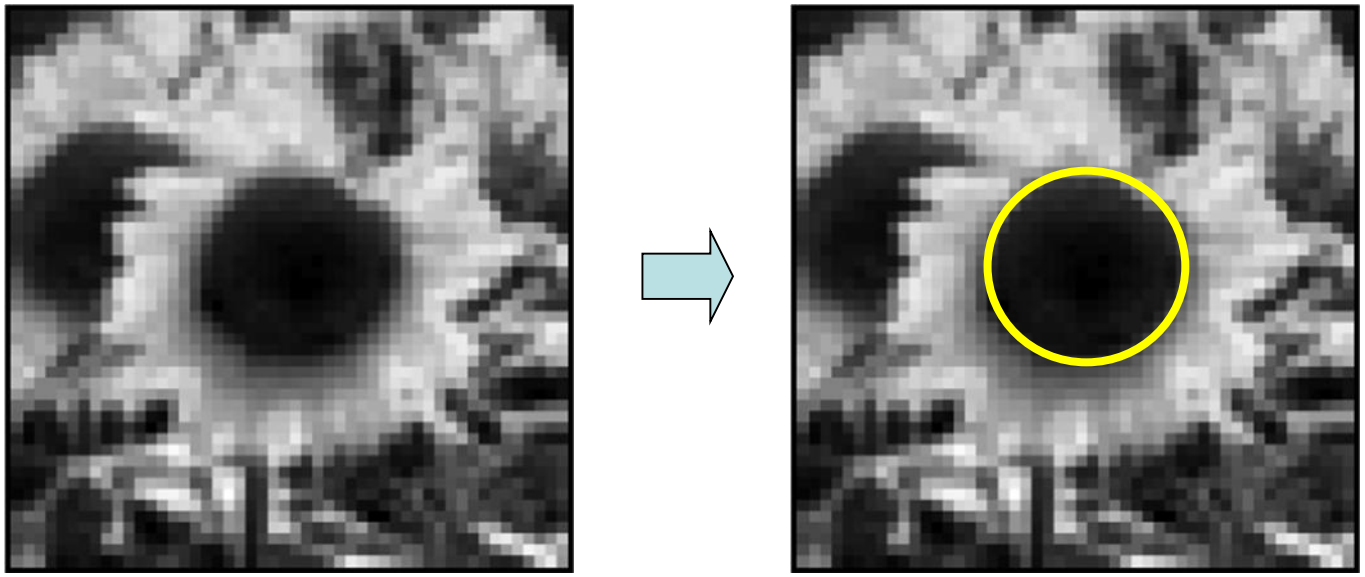
$$= -\frac{1}{\pi\sigma^4}\left(1-\frac{x^2+y^2}{2\sigma^2}\right)e^{-\frac{x^2+y^2}{2\sigma^2}}$$



Blob Detection

- Blobs (dt.: Tropfen, Kleckse) sind i.A. kompakte Bildregionen, die sich bzgl. bestimmter Eigenschaften (Helligkeit, Farbe, etc.) gut von ihrer Bildumgebung unterscheiden

- Blob Detection hat die Erkennung solcher Blobs zum Ziel



Bildquelle: S. Lazebnik, Dept. of Computer Science, Univ.of Illinois at Urbana-Champaign

LoG als Blob Detector (1)

Laplacian-of-Gaussian-Filter (LoG-Filter) ist bekanntester Blob Detector:

1) Konvolution der Bildfunktion $I(x,y)$ mit einer 2D-Gauß-Funktion auf bestimmten Skalen t (hier die Varianz σ^2 der Gauß-Funktion, also $t = \sigma^2$)

$$g(u,v,t) = \frac{1}{2\pi t} e^{-(u^2+v^2)/2t}$$

liefert zunächst Skalenraumrepräsentation $G(x,y,t) = g(x,y,t) * I(x,y)$

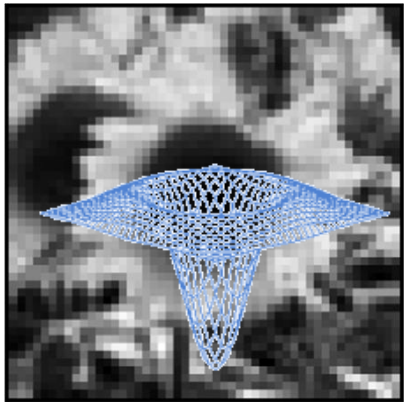
2) dann Anwendung des Laplace-Operators auf $G(x,y,t)$ für jede Skala t

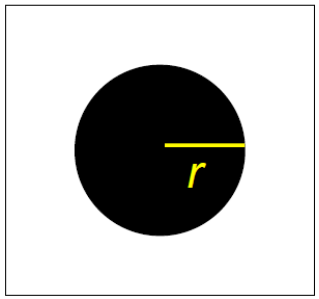
$$\nabla^2 G(x,y,t) = \frac{\partial^2 G}{\partial x^2}(x,y,t) + \frac{\partial^2 G}{\partial y^2}(x,y,t)$$

LoG als Blob Detector (2)

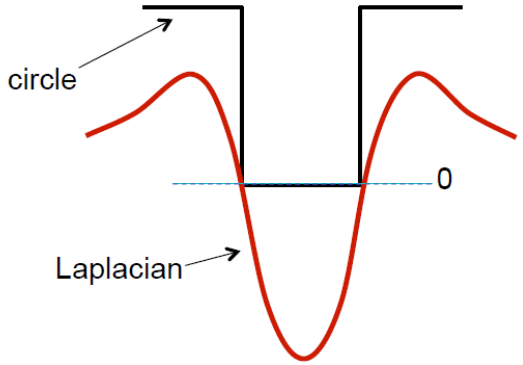
Laplacian-of-Gaussian-Filter (kurz: LoG-Filter) liefert

- starke positive Antworten für dunkle Blobs mit Radius $r = \sqrt{2t} = \sigma\sqrt{2}$
- starke negative Antworten für helle Blobs mit Radius $r = \sqrt{2t} = \sigma\sqrt{2}$





image



Bildquelle: S. Lazebnik, Dept. of Computer Science, Univ.of Illinois at Urbana-Champaign

LoG als Blob Detector (3)

- Ein Multiskalen-Ansatz zur Blob Detection mit automatischer Skalenwahl nutzt ein skalennormiertes LoG-Filter zur Detektion von lokalen Extrema im Skalenraum, also Extrema von $\nabla^2_{\text{norm}} G(x,y,t)$ sowohl bzgl. der Bildposition als auch bzgl. der Bildskala (Lindeberg, 1998)*

$$\nabla^2_{\text{norm}} \mathrm{G}(\mathrm{x},\mathrm{y},\mathrm{t}) = \mathrm{t}\left(\frac{\partial^2 \mathrm{G}}{\partial \mathrm{x}^2}(\mathrm{x},\mathrm{y},\mathrm{t}) + \frac{\partial^2 \mathrm{G}}{\partial \mathrm{y}^2}(\mathrm{x},\mathrm{y},\mathrm{t})\right)$$
$$= \sigma^2\left(\frac{\partial^2 \mathrm{G}}{\partial \mathrm{x}^2}(\mathrm{x},\mathrm{y},\sigma^2) + \frac{\partial^2 \mathrm{G}}{\partial \mathrm{y}^2}(\mathrm{x},\mathrm{y},\sigma^2)\right)$$

- Die Werte von Ableitungen des Gauß-Filters an Kanten sinken mit wachs. σ. Für skaleninvariante Werte ist bei zweifacher Ableitung also mit σ^2 zu multiplizieren.

* T. Lindeberg: Feature detection with automatic scale selection. Inter. Journal Computer Vision 30(2), 77-116, 1998

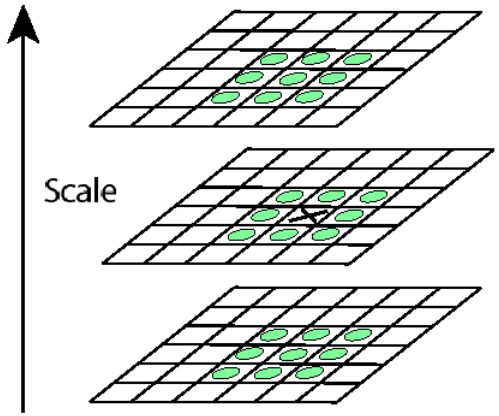
LoG als Scale-space Blob Detector

1. Geg.: diskretes Eingabebild $\mathbf{I} = [I(x,y)]$

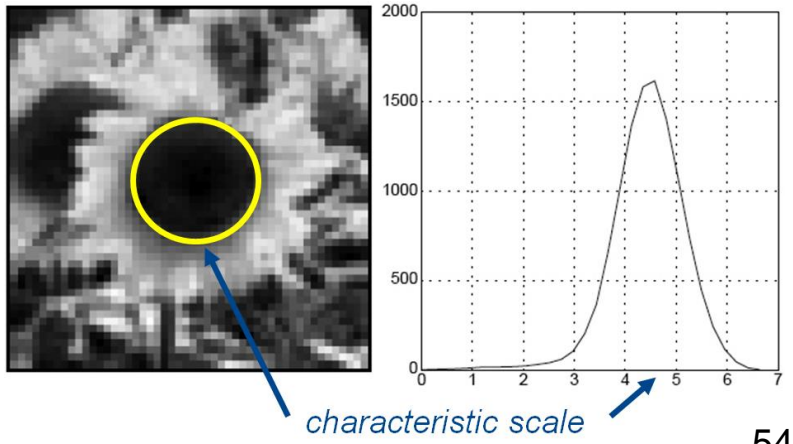
2. Generierung eines Skalenraums $G(x,y,t_k)$, $k = 0, 1, \ldots, r$

3. Erkennung von lokalen Extrema von $\nabla^2_{norm} G(x,y,t)$ im Skalenraum

   → G(x,y,t) ist lokales Maximum/Minimum, wenn ∇^2_{norm} G(x,y,t) größer/kleiner den ∇^2_{norm} -Werten aller 26 Nachbarn ist



4. Automatische Bestimmung der charakteristischen Skala t_c (*characteristic scale*) eines Blobs durch das entspr. lokale Extremum von $\nabla^2_{norm} G(x,y,t)$ in Skala t_c



Bildquellen: D. Lowe, Computer Science Dept., Univ. of British Columbia, S. Lazebnik, Dept. of Computer Science, Univ.of Illinois at Urbana-Champaign

SIFT

Der bekannteste *Multi-scale Blob Detector* ist der *SIFT*-Ansatz von David Love*

- SIFT steht für *Scale-invariant Feature Transform*
- *SIFT* zeigt zwei Beiträge:

  1) Skalierungsinvariante Erkennung von Interest Regions

  2) Skalierungs- und rotationsinvariante *Region Descriptors*

---

* [Lo99] David G. Lowe: Object Recognition from Local Scale-Invariant Features. In Proc. of the ICCV (1999)
[Lo04] David G. Lowe: Distinctive Image Features from Scale-Invariant Keypoints. In IJCV (2004)

SIFT: Workflow

Workflow von *SIFT*:

1) Erkennung von Extrema im Skalenraum
2) Lokalisierung von Keypoints

} Interest Regions

3) Bestimmung einer dominanten Orientation
4) Erzeugung von Keypoint-Deskriptoren

} Region Descriptors

SIFT: Erkennung von Extrema (1)

1) Gauß-Pyramide

- *SIFT* erstellt vom Eingabebild eine Gauß-Pyramide (*Gaussian scale space*) als Funktion $L(x,y,\sigma)$

- $L(x,y,\sigma)$ als Ergebnis der Konvolution mit Eingabebildfunktion $I(x,y)$:

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y) \text{ mit}$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(u^2+v^2)/2\sigma^2}$$

Bemerkg.: die Nomenklatur wird ab jetzt angepasst an [Lo04]:
- Gauß-Funktion $G(x,y,\sigma)$ mit Standartabweichung σ als Parameter
- Gauß-Pyramide $L(x,y,\sigma)$ mit Standartabweichung σ statt Skalierungsparameter t

SIFT: Erkennung von Extrema (2)

2) Effiziente Implementierung

- *SIFT* approximiert das LoG-Filter als Scale-space Blob Detector durch Difference-of-Gaussian-Filter (DoG) $D(x,y,\sigma)$

- $D(x,y,\sigma)$ wird auf $I(x,y)$ angewandt, indem die Differenz zweier benachbarter Ebenen der Gauß-Pyramide errechnet wird:

$$D(x,y,\sigma) = (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y)$$
$$= L(x,y,k\sigma) - L(x,y,\sigma)$$

Bemerkg.: auch hier angepasst an [Lo04] im Ggs. zu Vorl. 9:
Ersetzung der Laplace-Pyramide durch DoG-Pyramide $D(x,y,\sigma)$

SIFT: Erkennung von Extrema (3)

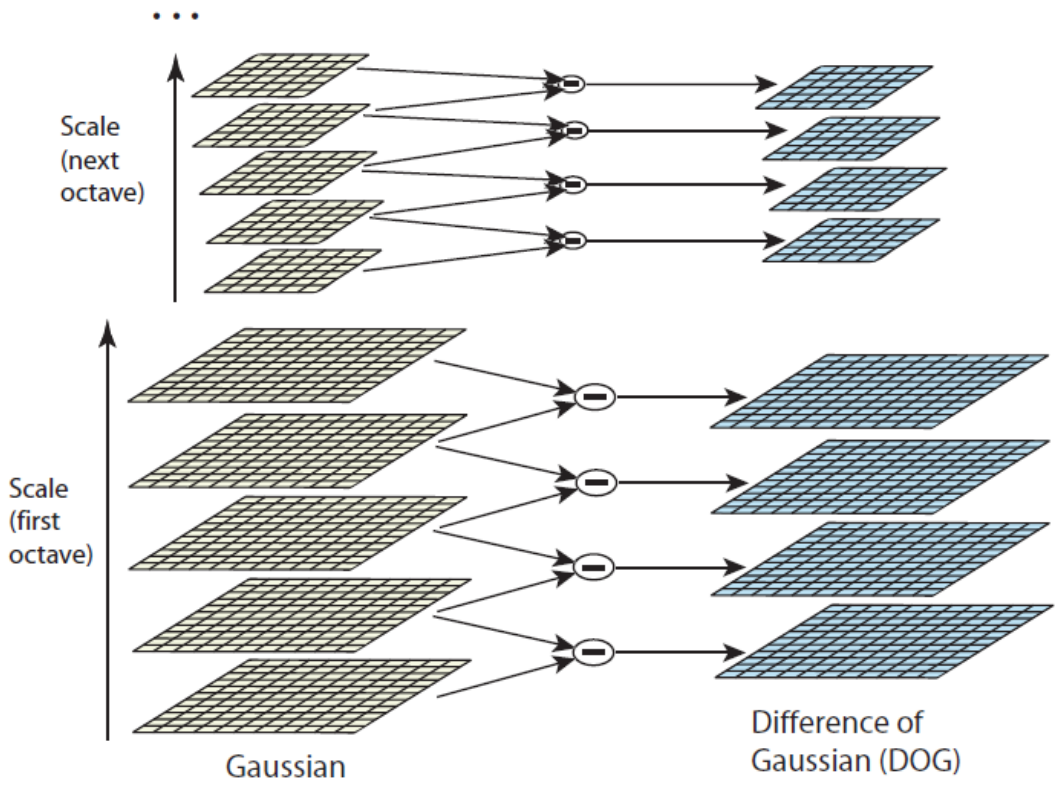
3) Organisation der Skalenräume

- Die Ebenen der Gauß-Pyramide $L(x,y,\sigma)$ sind in Oktaven gruppiert *
- Der Wert des Parameters k, der jeweils zwei aufeinander folgende Ebenen der Gauß-Pyramide separiert, wird so gewählt, dass jede Oktav eine fixe Anzahl von Ebenen aufweist
- Die Ebenen der DoG-Pyramide $D(x,y,\sigma)$ werden aus benachbarten Ebenen einer Oktave der Gauß-Pyramide errechnet und entsprechend gruppiert

* Eine Oktave entspricht der Verdopplung von σ

SIFT: Erkennung von Extrema (4)

4) Gesamtstruktur der Skalenräume



Bildquelle: [Lo04]

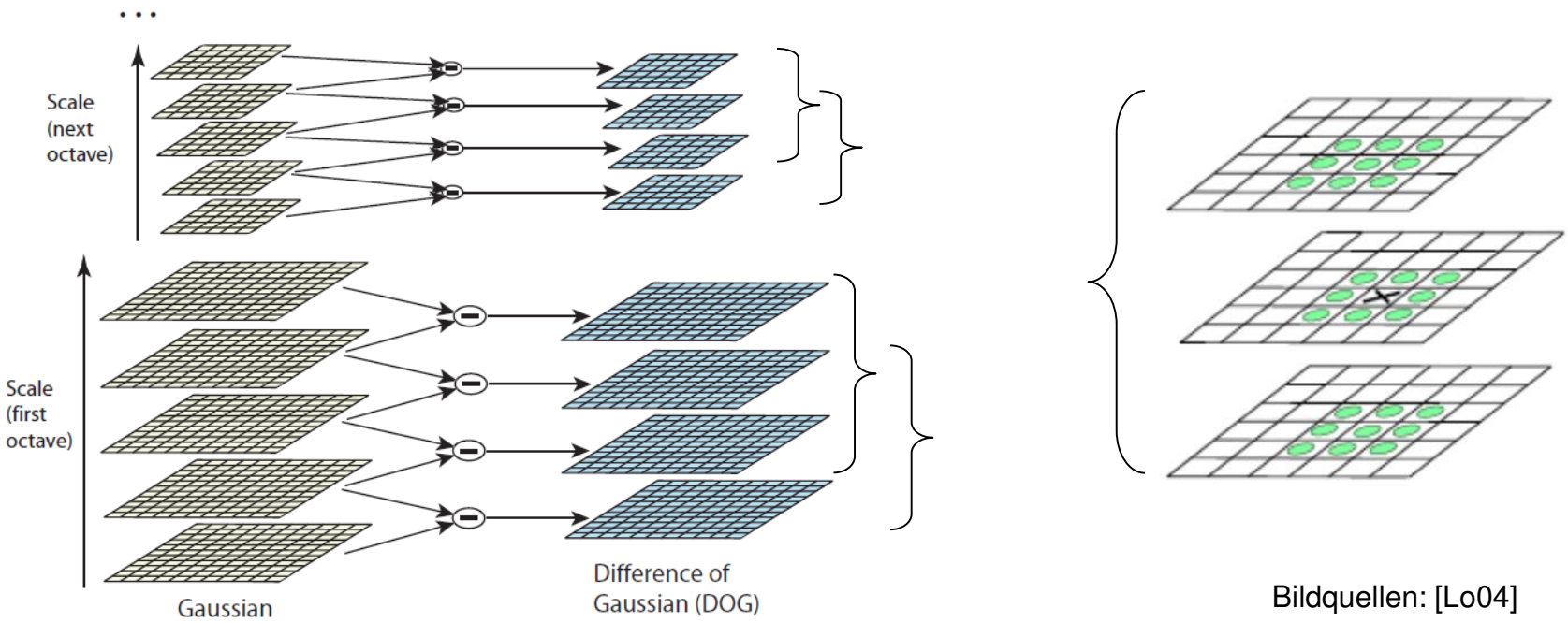
- Links: Gauß-Pyramide $L(x,y,\sigma)$
- Rechts: DoG-Pyramide $F(x,y,\sigma)$
- Nach jeder Octave: Downsampling (Reduce) der Gauß-geglätteten Bildes um Faktor 2

SIFT: Erkennung von Extrema (5)

5) Erkennung der lokalen Extrema von $\nabla^2_{norm} G(x,y,t)$ im Skalenraum

- Folie 54: „$G(x,y,t)$ ist lokales Maximum/Minimum, wenn $\nabla^2_{norm}\, G(x,y,t)$ größer/kleiner den ∇^2_{norm}-Werten aller 26 Nachbarn ist“

  ⇝ Hier: wähle lokale Maxima/Minima in 26er-Nachbarschaften in den Oktaven der DoG-Pyramide



Bildquellen: [Lo04]

SIFT: Erkennung von Extrema (6)

6) Parameter *k*

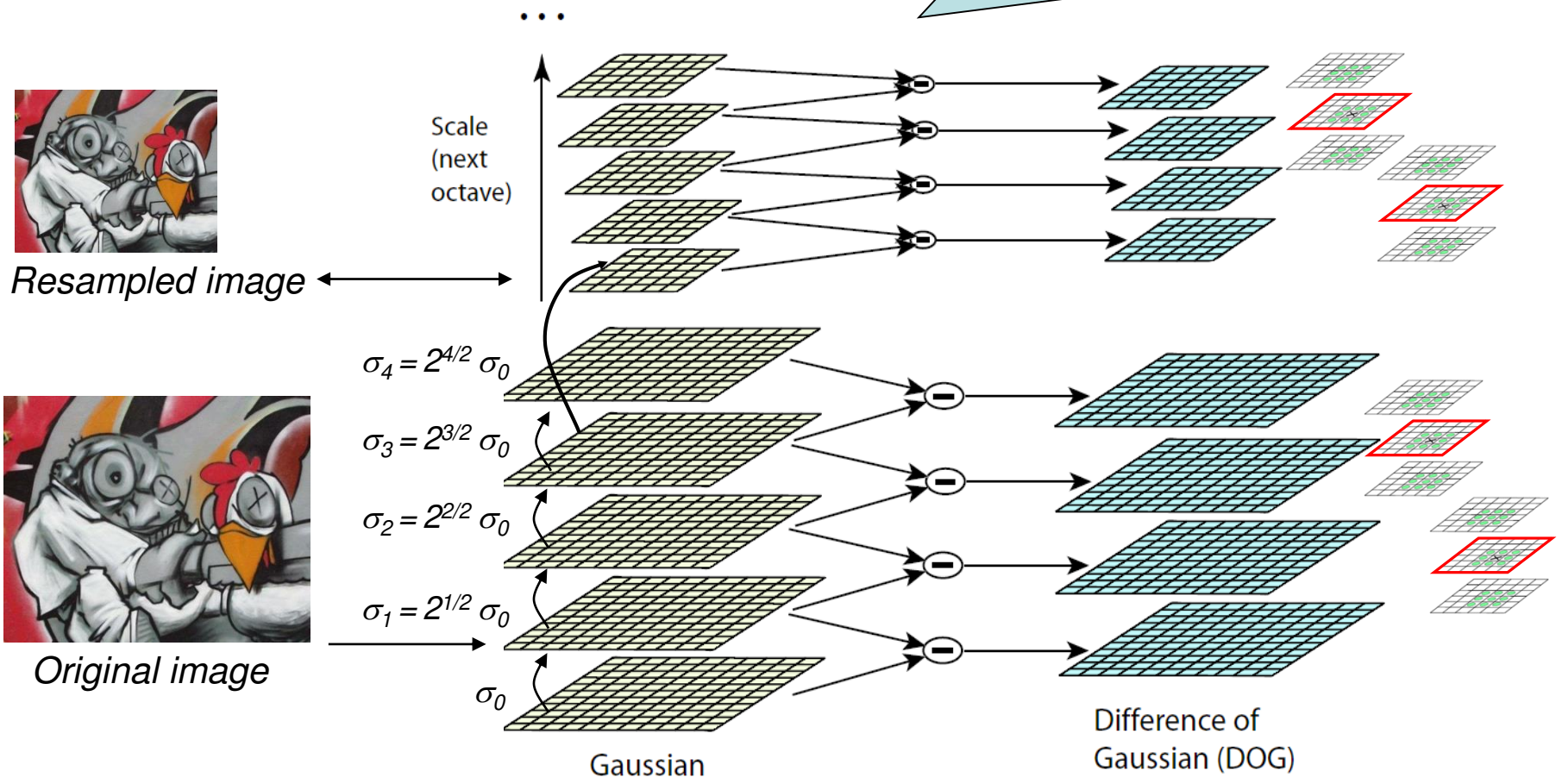
- Jede Oktave soll s Intervalle zu je drei DoG-Ebenen umfassen
  - $k = 2^{1/s}$
  - s + 3 Ebenen in der Gauß-Pyramide $L(x,y,\sigma)$ pro Oktave
- Mit Abschluss jeder Oktave $L(x,y,\sigma)$: Downsampling des Bildes mit doppeltem σ im Vergleich zum Eingangsbild der Oktave um den Faktor 2

SIFT: Erkennung von Extrema (7)

7) Pyramiden für $s = 2 \rightarrow k = 2^{1/2}$

Aus [Lo04]:"*Of course, if we pre-smooth the image before extrema detection, we are effectively discarding the highest spatial frequencies. Therefore, to make full use of the input, the image can be expanded to create more sample points that were present in the original. We double the size of the input image using linear interpolation prior to building the first level of the pyramid.*"

→ Initialer Wert: $\sigma_1 = 2^{1/2}\sigma_0$ und Verdopplung bei $\sigma_3 = 2^{3/2}\sigma_0$



Bildquellen: D. Lowe,

SIFT: Erkennung von Extrema (8)

8) Wahl des Parameters k

- Anzahl s der Skalen pro Oktave?

⇝ Empirische Untersuchung in [Lo04]:

- $s < 3$ → steigende Zahl stabiler Keypoints
- $s = 3$ → maximale Zahl stabiler Keypoints
- $s > 3$ → sinkende Zahl stabiler Keypoints

⇝ Empfehlung: $s = 3 \rightsquigarrow k = 2^{1/s} = 2^{1/3} \approx 1.26$

SIFT: Post-Processing der Extrema (1)

1) Präzise Lokalisierung der Keypoints

- Ableitung der Taylor-Entwicklung der im Keypoint zentrierten Funktion $D(x,y,\sigma)$ gleich Null setzend liefert mit $\boldsymbol{x} = (x,y,\sigma)^T$ den Offset

$$\hat{\mathbf{x}} = -\frac{\partial^2 D^{-1}}{\partial \mathbf{x}^2} \frac{\partial D}{\partial \mathbf{x}}$$

- Bei Offset > 0.5 in eine Richtung wird Keypoint in diese Richtung verschoben und erneuter Offset berechnet

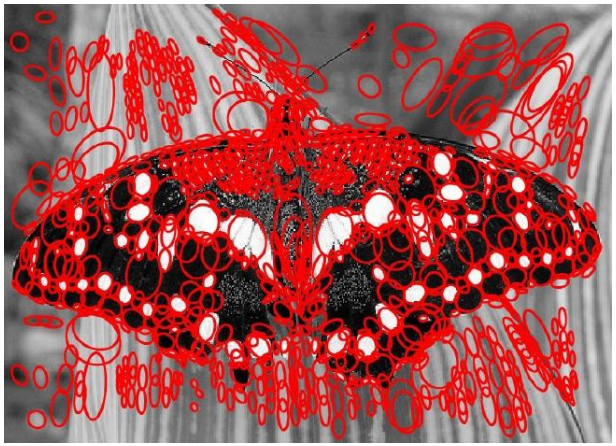
- Finaler Offset wird auf Position des Keypoints addiert.

SIFT: Post-Processing der Extrema (2)

2) Eliminierung von „Kanten-Keypoints“

- *D(x,y,σ)* zeigt auch Extrema an Kanten
- Mit der Harris-Matrix werden solche Keypoints nach folg. Ungleichung mit r = 10 erkannt und verworfen:

$$\frac{trace(H_{x,y})^2}{\det(H_{x,y})} < \frac{(r+1)^2}{r} \quad \text{mit} \quad H_{x,y} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix}$$



Bildquelle: S. Lazebnik, Dept. of Computer Science, Univ.of Illinois at Urbana-Champaign

SIFT: Keypoint Descriptor

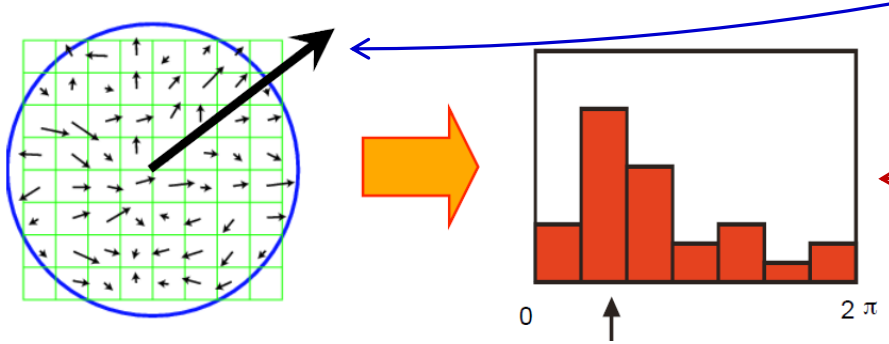
1) Ableitung der dominanten Orientierungen von Keypoints

- für die Rotationsinvarianz der Deskriptoren
- Sei σ die Skala des Keypoints, dann leite die Beträge und Orientierungen Gradienten der Pixel innerhalb des Radius $r_\sigma = 1{,}5 \cdot \sigma$ in $L(x,y,\sigma)$ ab:

$$Betrag := m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

$$Orientg := \Theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1)) / L((x+1,y) - L(x-1,y)))$$

- Eintragung der Beträge in ein Histogramm mit 36 Bins: das Maximum bestimmt die dominante Richtung (s. auch Anhang):

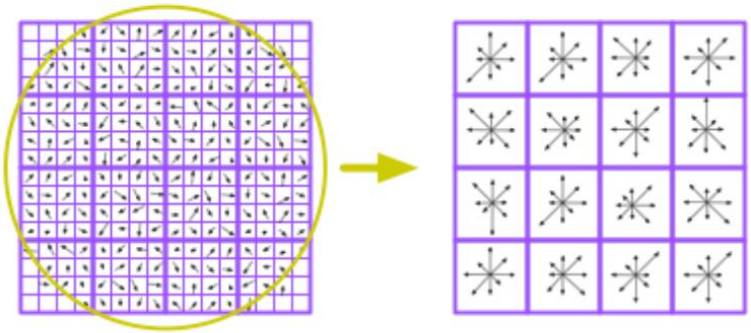


Bildquelle: S. Lazebnik, Dept. of Computer Science, Univ.of Illinois at Urbana-Champaign

SIFT: Keypoint Descriptor

2) Generierung des Keypoint-Deskriptors

- Sei σ die Skala des Keypoints, dann leite die Gradienten der benachbarten Pixel in $L(x,y,\sigma)$ ab und gewichte diese gemäß der Gauß-Funktion mit Standardabweichung σ
- Erfassung der Gradienten in einem 16×16 Sample-Array. Gruppierung der Einträge in $4 \times 4 = 16$ Teilregionen und dort Klassifikation der Orientierungen in 8 Orientierungs-Bins
- Der Keypoint-Deskriptor zeigt $16 \cdot 8 = 128$ Werte

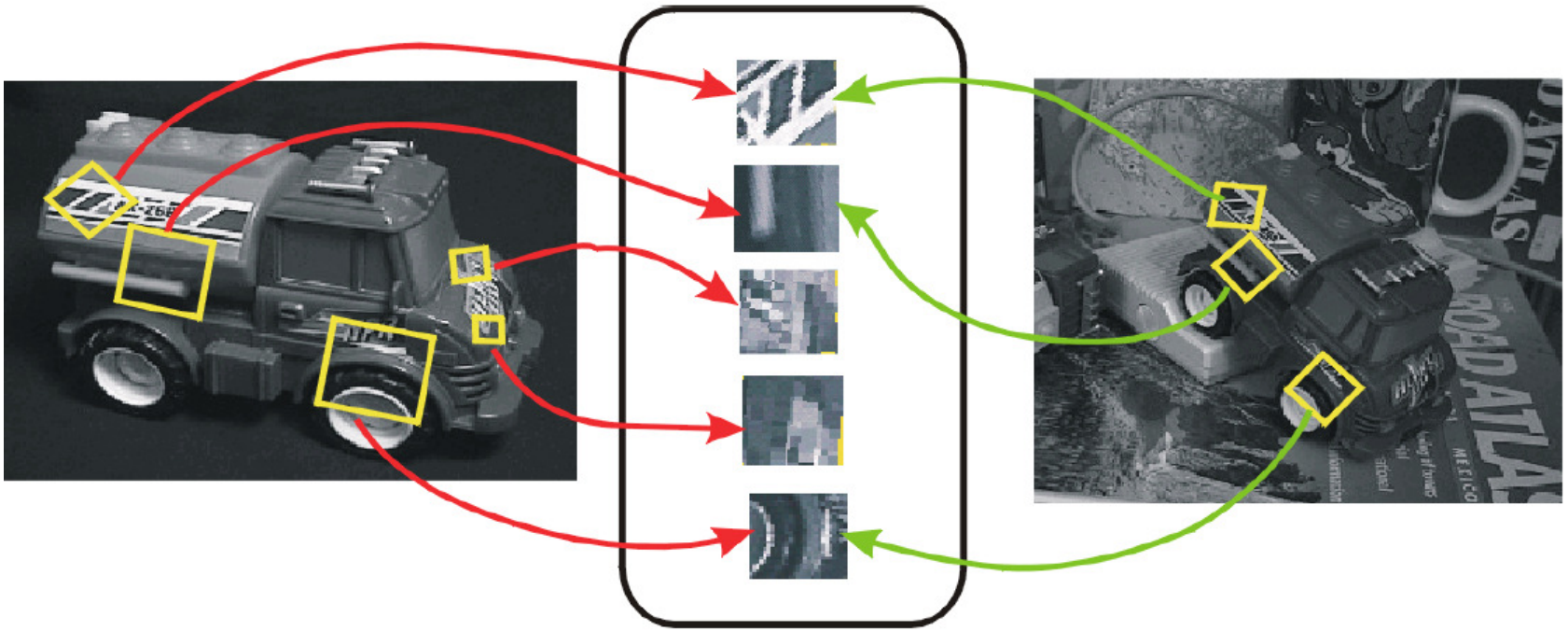


Bildquelle: S. Lazebnik, Dept. of Computer Science, Univ.of Illinois at Urbana-Champaign

SIFT: Eigenschaften (1)

Extraordinarily robust detection and description technique

- Can handle changes in viewpoint
  - Up to about 60 degree out-of-plane rotation
- Can handle significant changes in illumination
  - Sometimes even day vs. night
- Fast and efficient—can run in real time
- Lots of code available



Folienquelle: S. Lazebnik, Dept. of Computer Science, Univ.of Illinois at Urbana-Champaign

SIFT: Eigenschaften (2)

Extraordinarily robust detection and description technique

- Can handle changes in viewpoint
  - Up to about 60 degree out-of-plane rotation
- Can handle significant changes in illumination
  - Sometimes even day vs. night
- Fast and efficient—can run in real time
- Lots of code available

Folienquelle: S. Lazebnik, Dept. of Computer Science, Univ.of Illinois at Urbana-Champaign

Zusammenfassung

- Der Harris-Operator ist ein sog. Eckenoperator, der nach markanten Punkten (Interest Points) sucht.
  - Der Harris-Operator ist invariant bzgl. Translation und Rotation der Objekte in der Bildebene, aber nicht bzgl. des Abbildungsmaßstabes
  - Der Harris-Operator erlaubt per se auch keine Wiedererkennung korrespondierender Punkte

- SIFT steht für *Scale-invariant Feature Transform* und zeigt zwei Beiträge:
  - Skalierungsinvariante Erkennung von Interest Regions
  - Generierung von skalierungs- und rotationsinvarianten Region Descriptors zur Wiedererkennung von Interest Regions

Anhang: Dominante Orientierung von SIFT

- Das Orientierungshistogramm mit 36 Bins überdeckt volle 360°
- Jeder Eintrag in das Histogramm wird mit seinem Gradientenbetrag gewichtet, wobei dieser Betrag wiederum durch eine im Keypoint zentrierte Gauß-Funktion mit $\sigma = 1{,}5 \cdot \sigma_{\text{Skala des Keypoints}}$ gewichtet wird
- Der maximale Eintrag sowie jeder Eintrag, der mind. 80% des maximalen Eintrages aufweist, generiert jeweils einen Keypoint-Deskriptor
- Daher kann es für einen Ort (x,y) und einen Maßstab σ mehrere Keypoints mit unterschiedlicher Orientierung geben
- Lowe stellt fest, dass ca. 15% aller Keypoints mehrere Orientierungsvarianten haben, dies aber signifikant zur Identifikation von Keypoints beiträgt