

# Computergrafik 2: Objekt-/Bilderkennung

Prof. Dr. Michael Rohs, Dipl.-Inform. Sven Kratz

[michael.rohs@ifi.lmu.de](mailto:michael.rohs@ifi.lmu.de)

MHCI Lab, LMU München

Folien teilweise von Andreas Butz, sowie von Klaus D. Tönnies  
(Grundlagen der Bildverarbeitung. Pearson Studium, 2005)

# Themen heute

- Objekt-/Bilderkennung
- Lokale Merkmale
- Skalenraum
- Scale Invariant Feature Transform (SIFT)

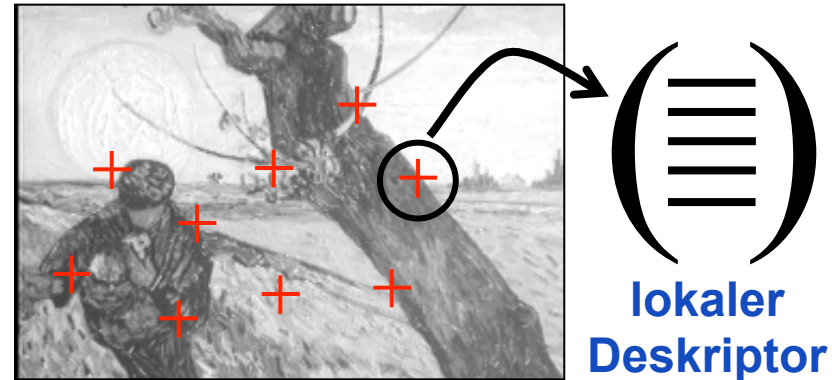
# **OBJEKT-/BILDERKENNUNG MIT LOKALEN MERKMALEN**

# Objekt-/Bildererkennung

- Identifikation von Objekten, Szenen, Teilbildern und Bestimmung von Parametern (Position, Größe, Orientierung, etc.)
- Bildregistrierung: Transformation berechnen, um zwei Bilder der selben Szene in Übereinstimmung bringen
  - unbekannte Perspektivenänderung der Kamera
- Anwendungen
  - Visuelle Suche: Finden ähnlicher Bilder zu einem Anfragebild
  - Automatisierung und Qualitätskontrolle in der Industrie
  - Mobile Roboter, Spielzeuge, Benutzungsschnittstellen
  - Lokalisierung
  - Panoramabilder
  - Augmented Reality

# Objekt-/Bilderkennung durch lokale Merkmale

- Charakteristische Orte im Bild (“interest points”)
  - charakteristisch, unverwechselbar, hoher Informationsgehalt
  - stabil lokalisierbar
  - robust gegenüber Veränderung der Perspektive, Helligkeit, etc.

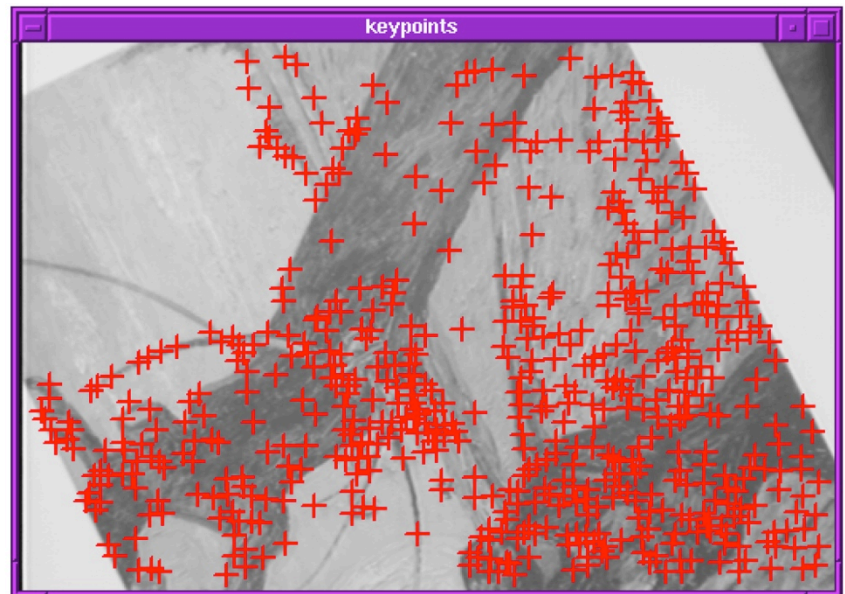
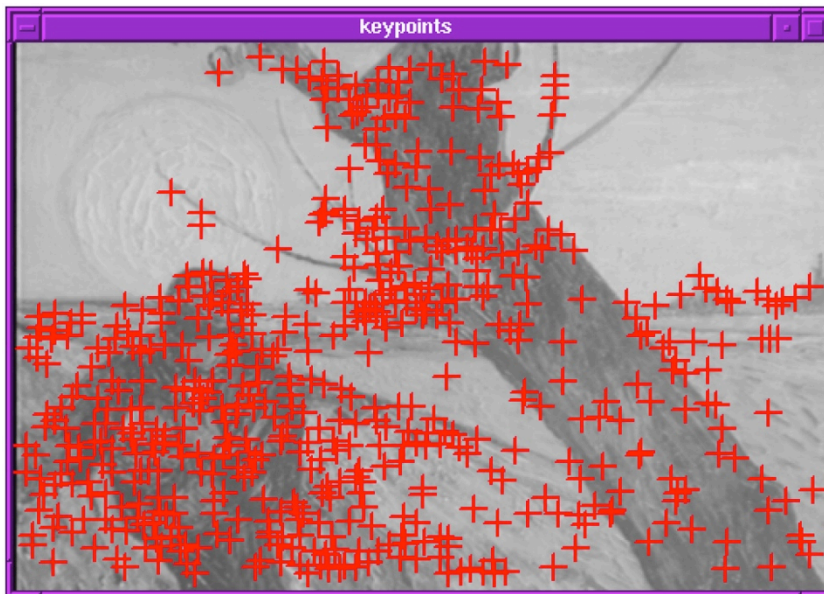


- lokale Deskriptoren, Merkmalsvektoren
  - beschreiben / repräsentieren charakteristische Orte im Bild
  - robust / invariant gegenüber Veränderung der Perspektive, Helligkeit, Translation / Rotation / Skalierung, etc.
  - effizient berechenbar
- Vorteil: teilweise Verdeckung unproblematisch

**Bildquelle:** Schmid, Mohr: Local Grayvalue Invariants for Image Retrieval. PAMI, 19(5):530-534, 1997.

# Objekt-/Bilderkennung durch lokale Merkmale

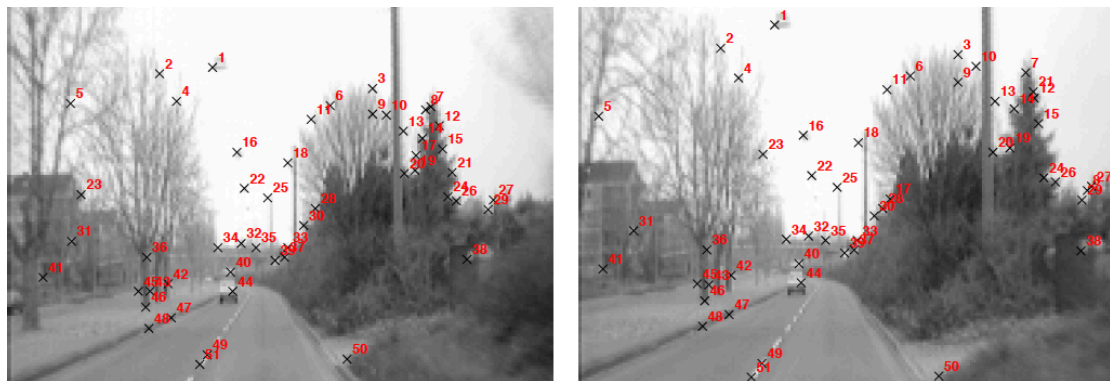
- Finden von ähnlichen Bildern
  1. Charakteristische Orte extrahieren (z.B. Harris Corner-Detektor)
  2. lokale Deskriptoren berechnen
  3. korrespondierende Deskriptoren in anderen Bildern finden
  4. Bild mit den meisten Treffern auswählen → Korrektheit?



**Bildquelle:** Schmid, Mohr: Local Grayvalue Invariants for Image Retrieval. PAMI, 19(5):530-534, 1997.

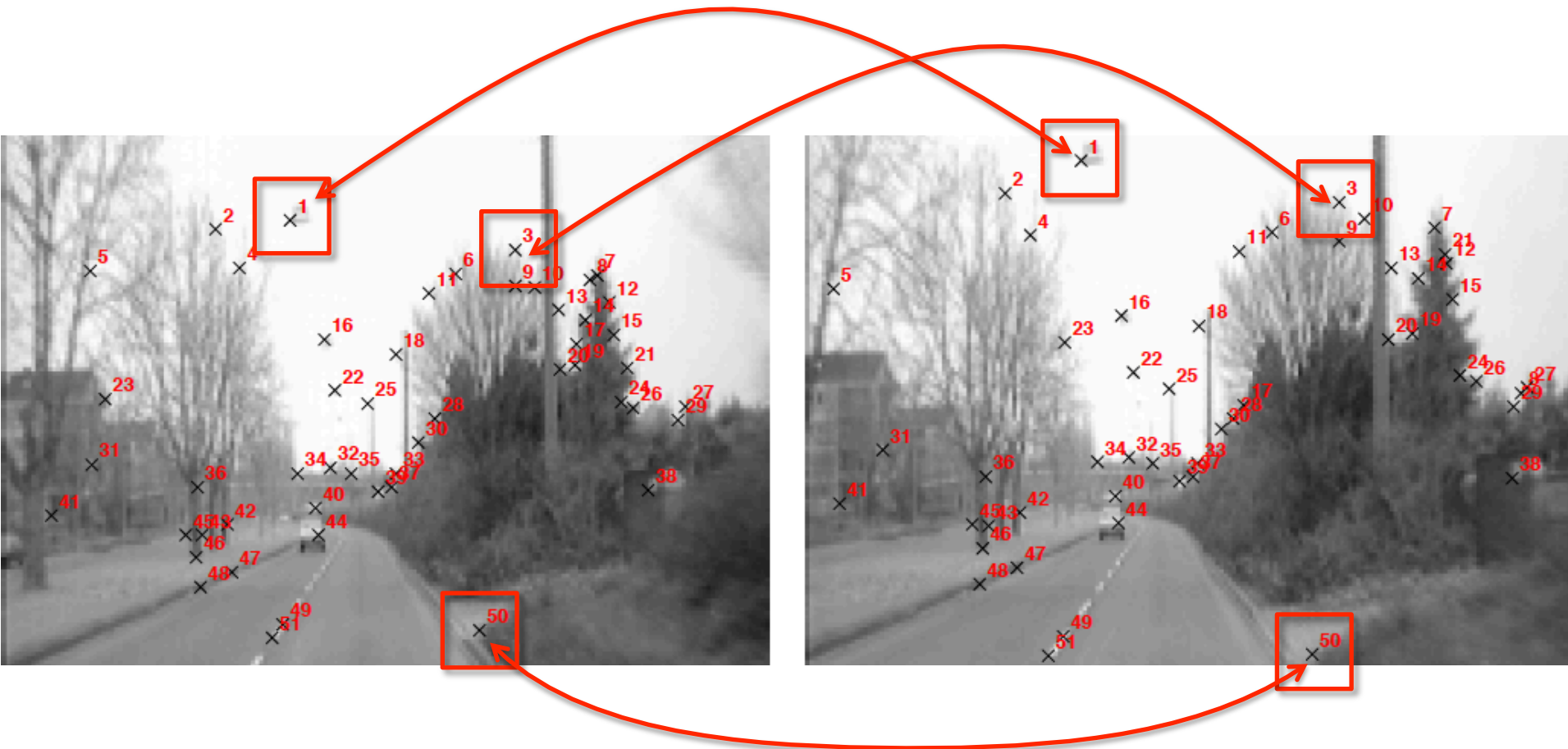
# Zhang et al. 1994: Matching Corners

- Bildregistrierung mit Hilfe lokaler Merkmale
  - Bildregistrierung: Transformation berechnen, um zwei Bilder der selben Szene in Übereinstimmung bringen
  - unbekannte Perspektivenänderung der Kamera
- Finden von korrespondierenden Punkten in den Bildern
  - Harris Corner-Detektor
  - Template-Matching durch Korrelation an den Ecken



**Bildquelle:** Zhang, Deriche, Faugeras, Luong: A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry. Technical Report, INRIA, 1994.

# Zhang et al. 1994: Matching Corners



- Template-Matching: Skalierung? Rotation?

**Bildquelle:** Zhang, Deriche, Faugeras, Luong: A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry. Technical Report, INRIA, 1994.



# Schmid & Mohr 1997: Rotationsinvarianz

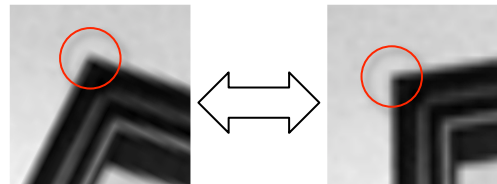
- Finden von korrespondierenden Punkten in den Bildern
  - Harris Corner-Detektor
  - Rotationsinvariante Merkmale (Kombinationen von Gaußableitungen, z.B. quadrierte Gradientenlänge, Laplace-Operator)
- Rotationsinvarianz, eingeschränkte Skalierungsinvarianz
- Voting zum Finden des ähnlichsten Bildes
  - Bilder in Datenbank nummeriert 1..k
  - Merkmal  $j$  für Bild  $i$  in Datenbank gespeichert als  $(m_{ij}, i)$
  - Für alle Merkmale  $m_{qx}$  im Anfragebild:  
Falls Distanz  $d(m_{qx}, m_{ij}) < t$ , dann erhält Bild  $i$  eine Stimme
  - Auswahl des Bildes mit den meisten Stimmen
  - (Korrektheit? Effizienz?)

Schmid, Mohr: Local Grayvalue Invariants for Image Retrieval. PAMI, 19(5):530-534, 1997.

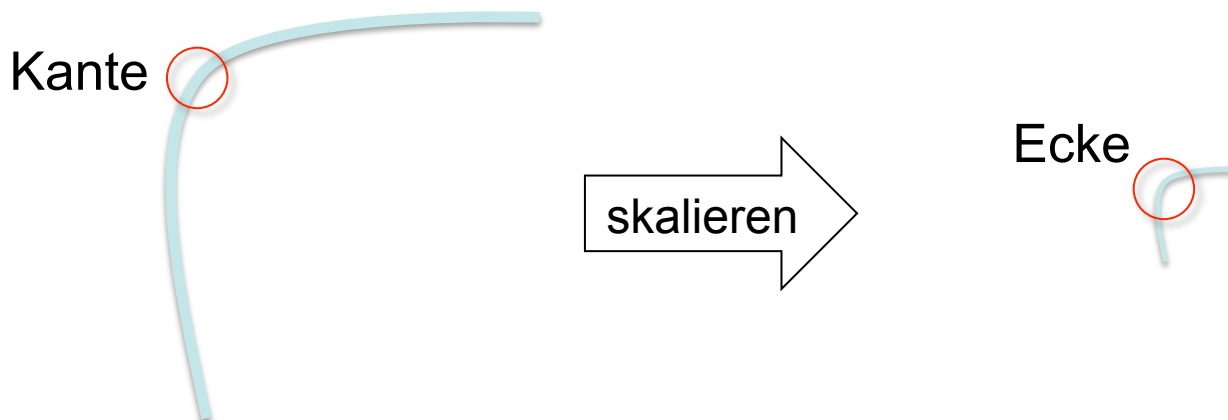
# SKALENRAUM

# Robustheit des Harris Corner Detektors

- Invariant gegenüber Helligkeitsänderungen
  - bis zu einem gewissen Grad
- Invariant gegenüber Translation und Rotation

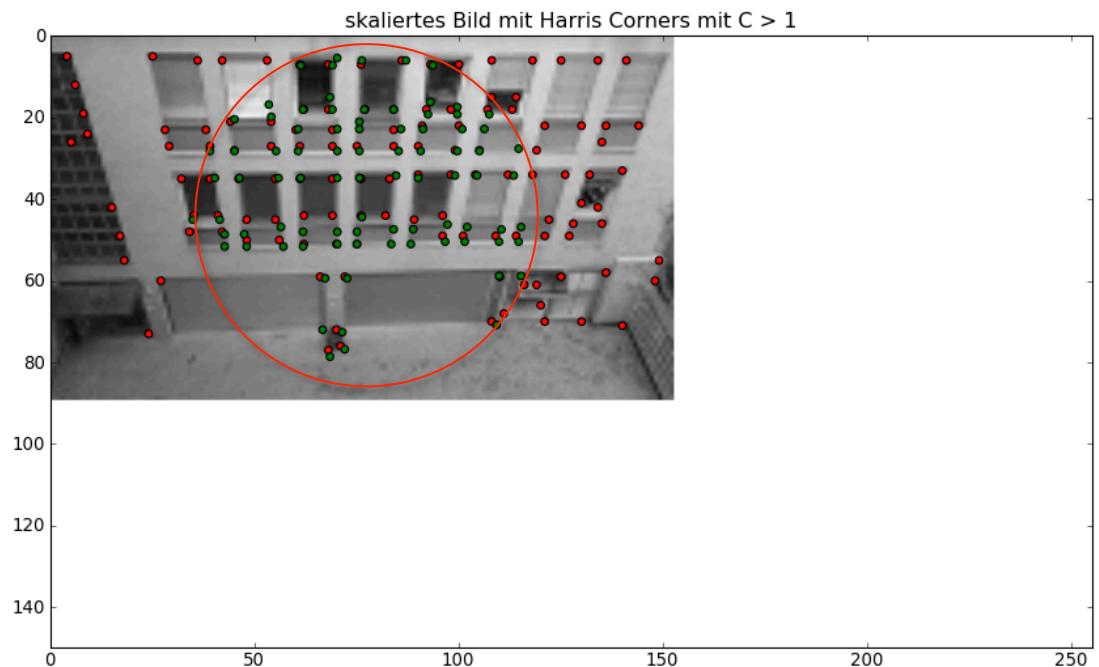
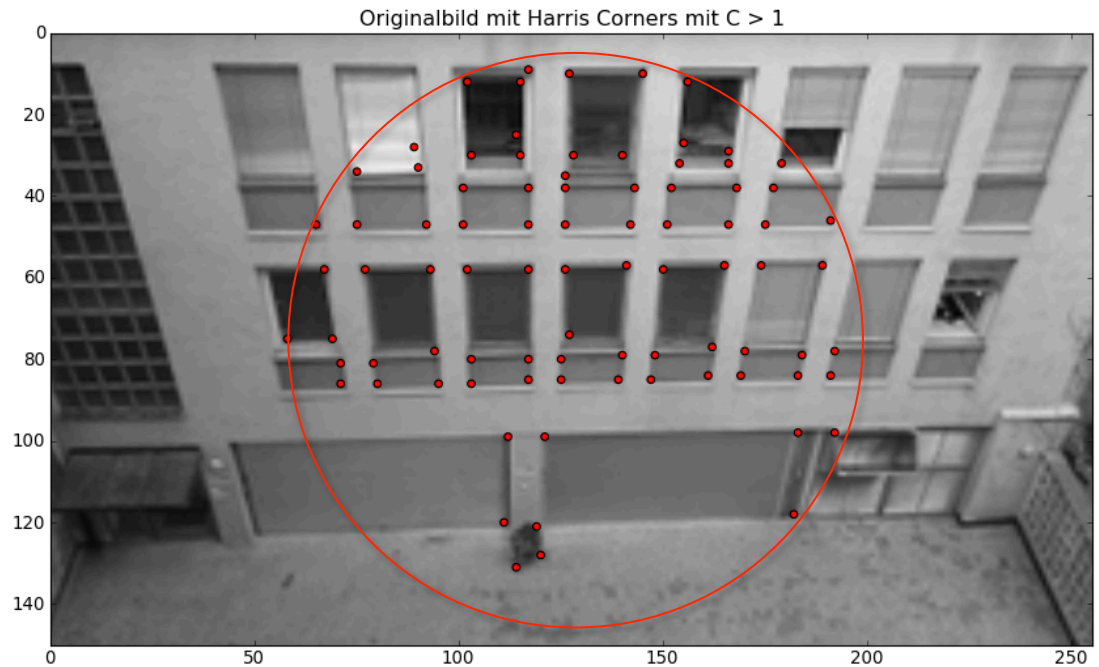
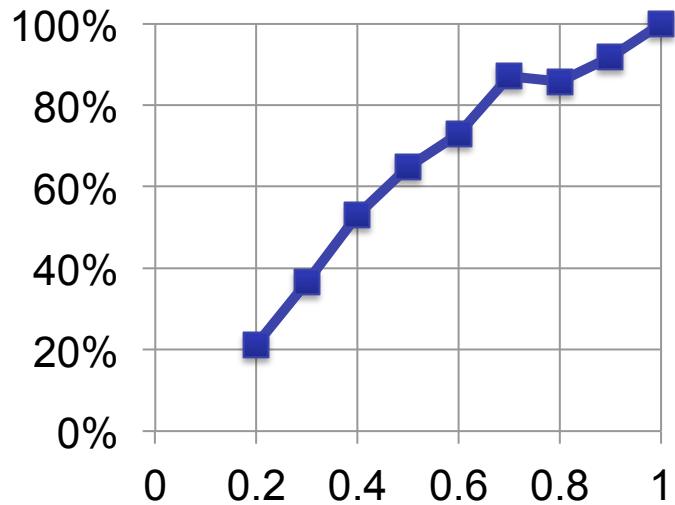


- Nicht invariant gegenüber Skalierung



# Repeatability

- Starke Abhängigkeit des Harris-Detektors von der Skalierung:



# Skalierungsinvarianz

- Bestimmung der charakteristischen Größe eines „interest points“
- Kombiniert Harris-Corner-Detection mit Laplace-Operator
- Harris-Laplace-Methode
  - Berechnung des Skalenraums für das Bild
  - Berechnung von Harris-Corners für jede Skalierung → interest points
  - Berechnung der zweiten Ableitung (Laplace-Operator) an den interest points
  - Auswahl von Harris-Corners, die lokales Maximum entlang der Skalierungs-Achse aufweisen

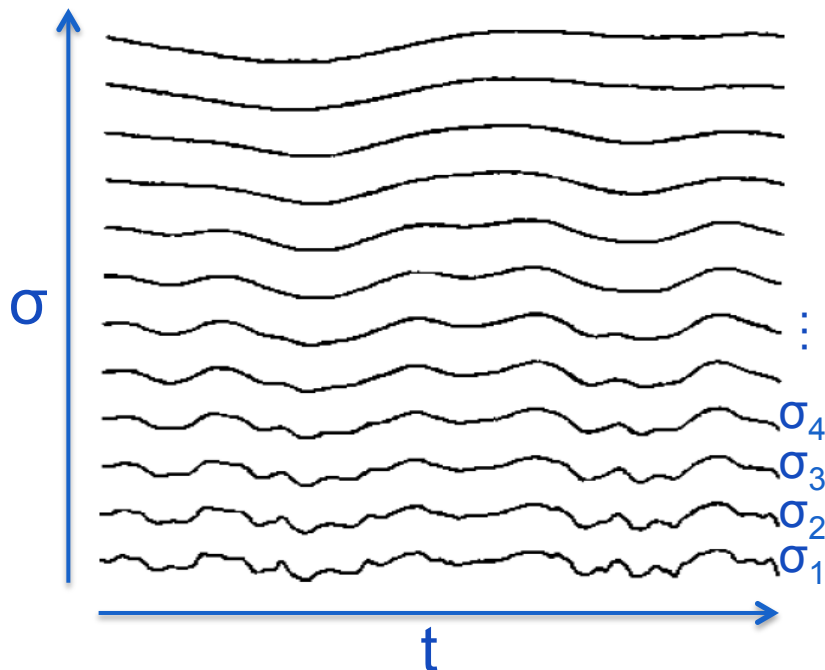


Bild: K.Tönnies, Grundlagen der Bildverarbeitung

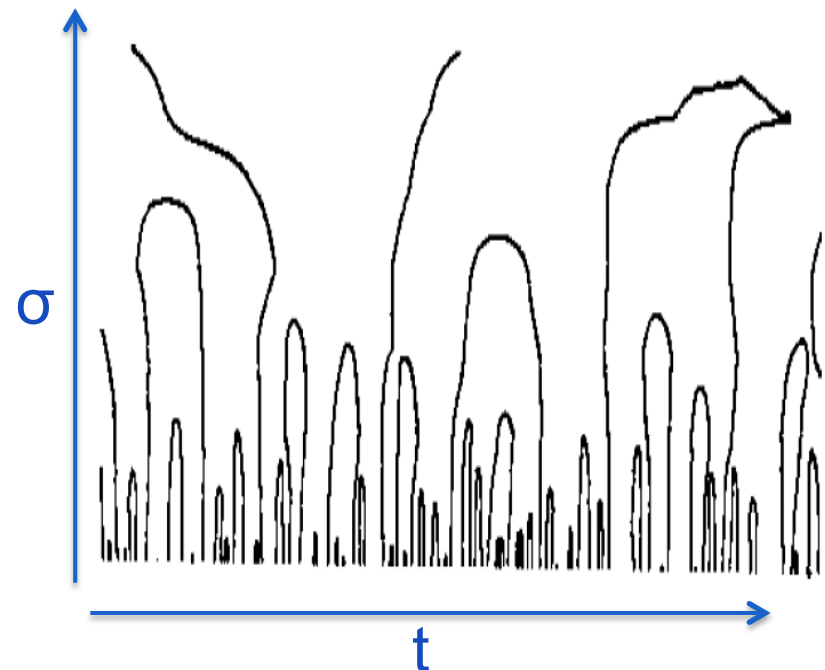
# Multiskalen-Repräsentation eines Signals

- Glätten eines 1D-Signals mit Gaußfiltern
- Größere Strukturen auf größeren Skalierungsstufen

Signal  $f$  (Multiskalen-Repr.):



zweite Ableitung  $f_{xx} = 0$ :

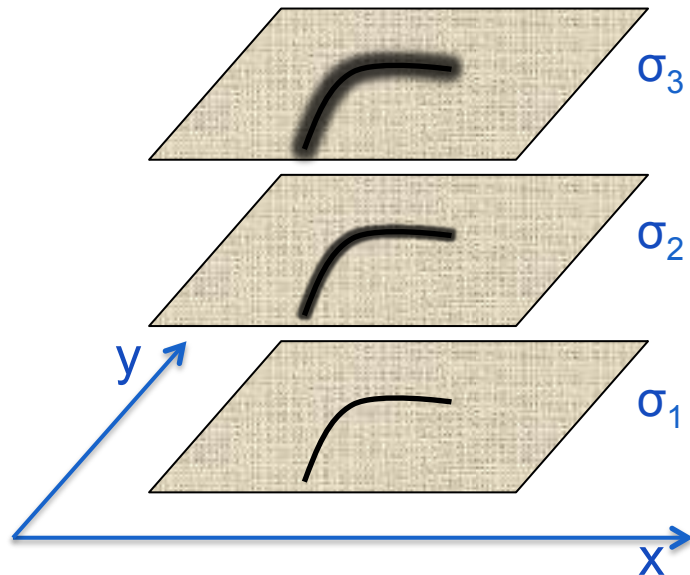


Andrew P. Witkin: Scale-Space Filtering. IJCAI, pp. 1019-1022, 1983.

# Multiskalen-Repräsentation eines Bildes

- Glätten eines 2D-Signals mit Gaußfiltern
- Größere Strukturen auf größeren Skalierungsstufen

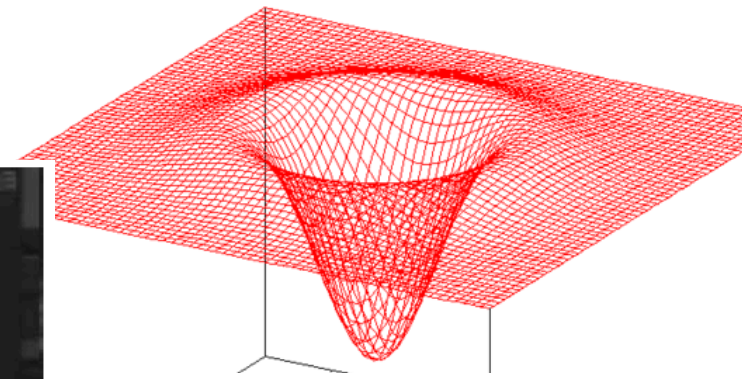
Bild (Multiskalen-Repr.):



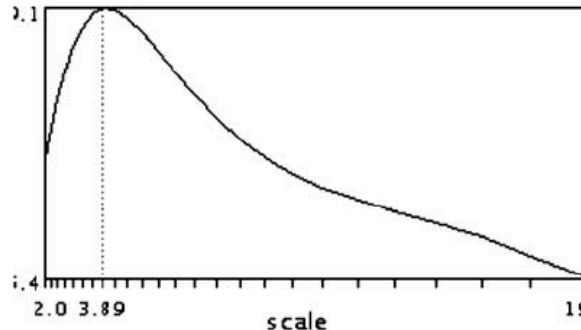
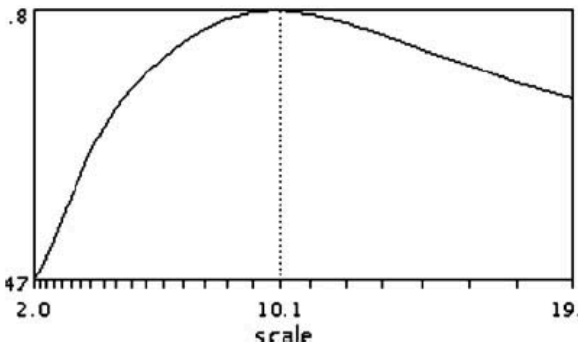
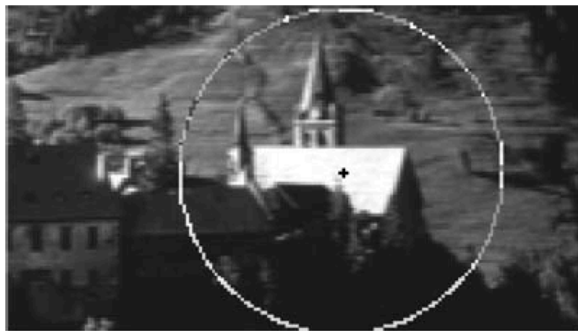
# Charakteristische Skalierungsstufe

- Charakteristische Skalierung eines Merkmals ist lokales Extremum des LoG-Operators

LoG-Operator:



“blob detector”



charakteristische Skalierung:

$\sigma = 10.1$  (links)

$\sigma = 3.90$  (rechts)

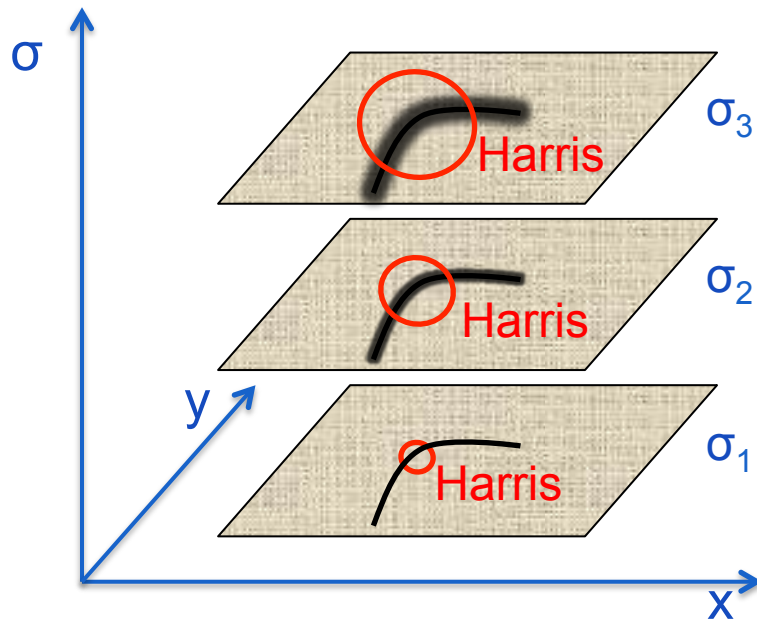
Bild: Mikolajczyk, Schmid: Indexing Based on Scale Invariant Interest Points. ICCV 2001, pp. 525-531



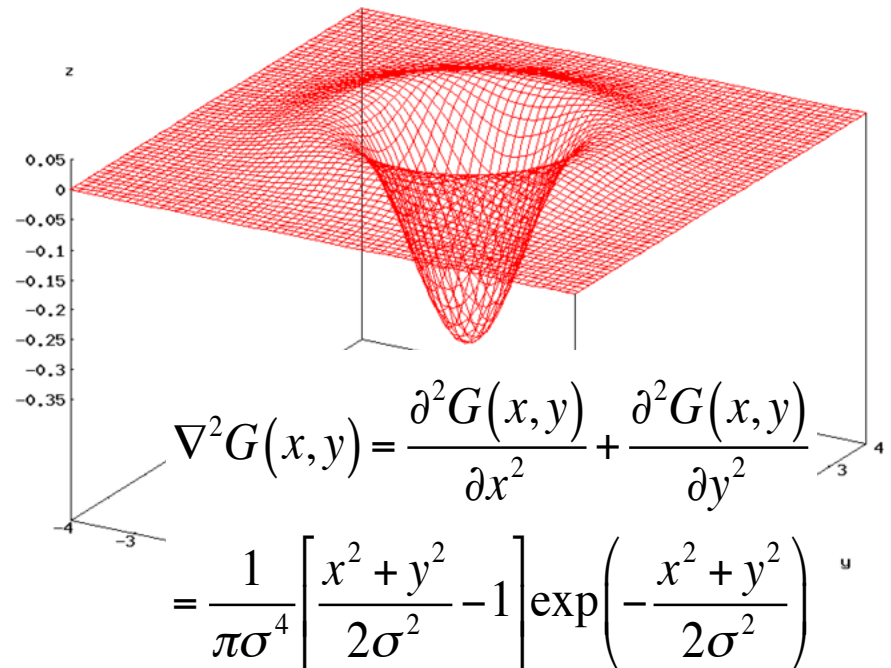
# Harris-Laplace-Methode (Mikolajczyk, Schmid, 2001)

- Glätten eines 2D-Signals mit Gaußfiltern
- Größere Strukturen auf größeren Skalierungsstufen
  - Laplace der Gaußfunktion (LoG)

Bild (Multiskalen-Repr.):

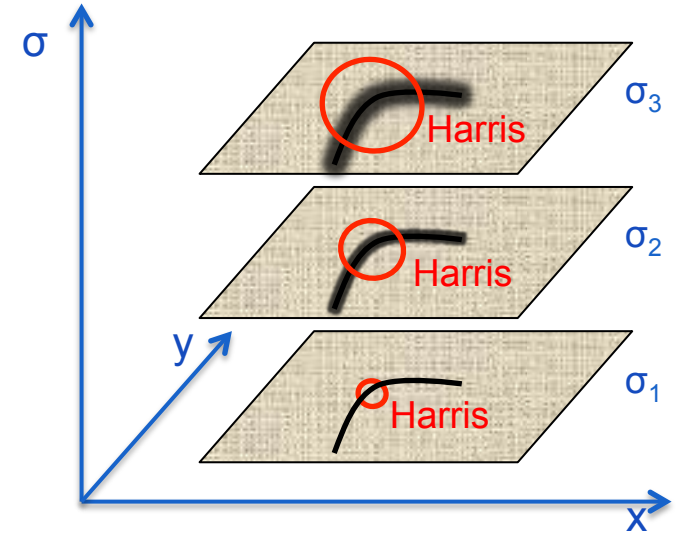


LoG-Operator:



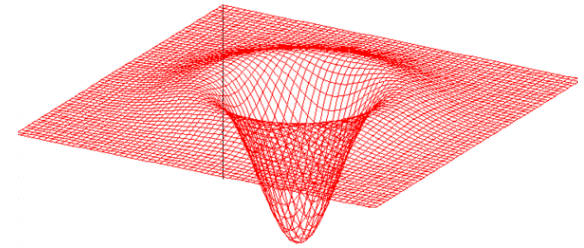
# Harris-Laplace-Methode (Mikolajczyk, Schmid, 2001)

- Berechnung des Skalenraums
- Berechnung von Harris-Corners für jede Skalierung → interest points  
$$C(x,y,\sigma_i) > t_h \wedge C(x,y,\sigma_i) > C(x_w,y_w,\sigma_i)$$
mit  $(x_w,y_w)$  8-Nachbarn
- Berechnung der zweiten Ableitung (LoG-Operator) an den interest points
- Auswahl von Harris-Corners, die lokales Maximum des LoG entlang der Skalierungs-Achse aufweisen  
$$L(x,y,\sigma_i) > t_l \wedge L(x,y,\sigma_i) > L(x,y,\sigma_{i-1}) \wedge L(x,y,\sigma_i) > L(x,y,\sigma_{i+1})$$
- Normalisierung bzgl. Skalierung notwendig
- Präzisere Variante durch Iteration über Ort und Skalierung



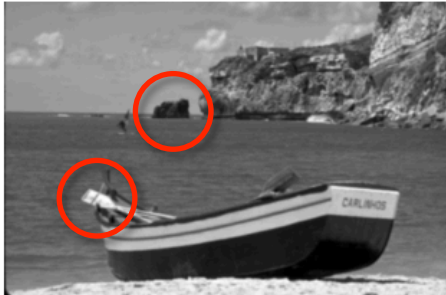
# LoG-Operator

LoG-Operator:



$$\sigma^2 (f_{xx}(x, y, \sigma) + f_{yy}(x, y, \sigma))$$

$f(x, y, \sigma), \sigma = 1.0$



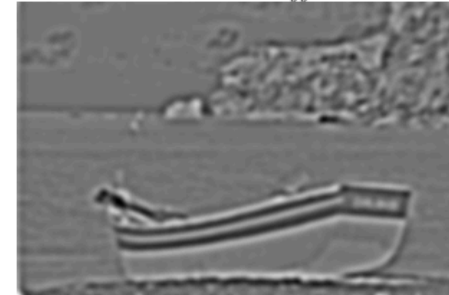
$\sigma^2 (f_{xx}(x, y, \sigma) + f_{yy}(x, y, \sigma))$



$f_\sigma(x, y), \sigma = 4.0$



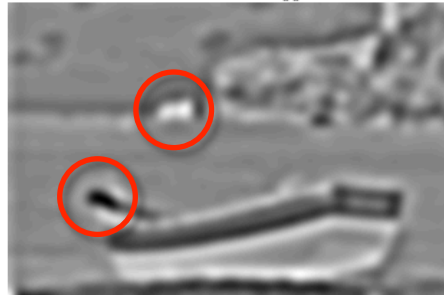
$\sigma^2 (f_{xx}(x, y, \sigma) + f_{yy}(x, y, \sigma))$



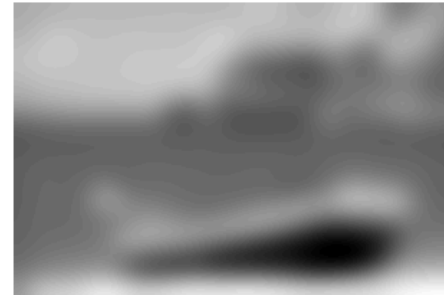
$f_\sigma(x, y), \sigma = 8.0$



$\sigma^2 (f_{xx}(x, y, \sigma) + f_{yy}(x, y, \sigma))$



$f_\sigma(x, y), \sigma = 16.0$



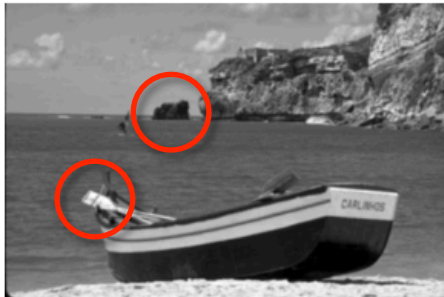
$\sigma^2 (f_{xx}(x, y, \sigma) + f_{yy}(x, y, \sigma))$



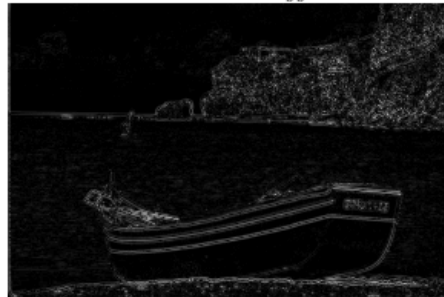
# Skalenraum: Betrag des LoG-Operators

$$\left| \sigma^2 \left( f_{xx}(x, y, \sigma) + f_{yy}(x, y, \sigma) \right) \right|$$

$f(x, y, \sigma), \sigma = 1.0$



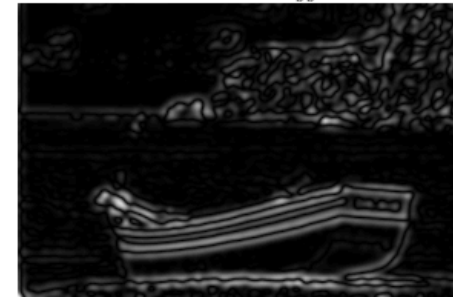
$|\sigma^2 (f_{xx}(x, y, \sigma) + f_{yy}(x, y, \sigma))|$



$f_{\sigma}(x, y), \sigma = 4.0$



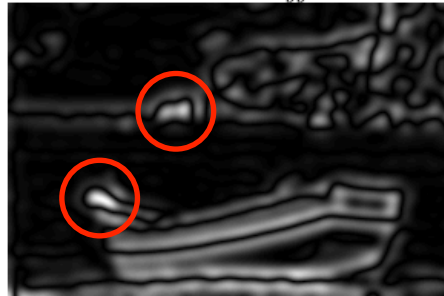
$|\sigma^2 (f_{xx}(x, y, \sigma) + f_{yy}(x, y, \sigma))|$



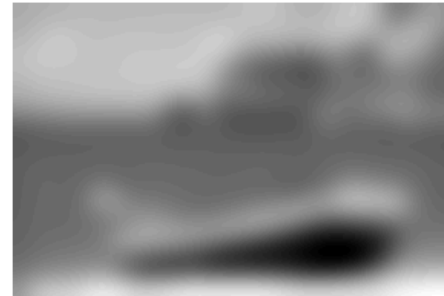
$f_{\sigma}(x, y), \sigma = 8.0$



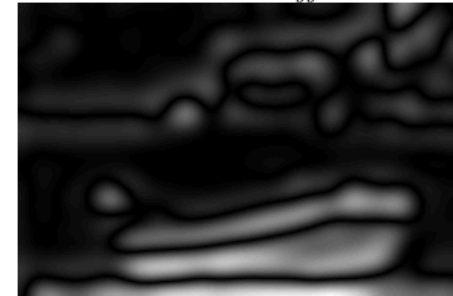
$|\sigma^2 (f_{xx}(x, y, \sigma) + f_{yy}(x, y, \sigma))|$



$f_{\sigma}(x, y), \sigma = 16.0$



$|\sigma^2 (f_{xx}(x, y, \sigma) + f_{yy}(x, y, \sigma))|$



# Skalenraum: Betrag des DoG-Operators

- DoG ist effiziente Näherung für LoG (→ Folien zu Kanten)

$$\left| \left( f(x, y, \sigma_{n-1}) - f(x, y, \sigma_n) \right) \right| = \left| \left( f(x, y) * G(\sigma_{n-1}) - f(x, y) * G(\sigma_n) \right) \right|$$

$f(x, y, \sigma), \sigma = 1.0$



$\left| (f(x, y) * G(\sigma_{n-1}) - f(x, y) * G(\sigma_n)) \right|$



$f_\sigma(x, y), \sigma = 4.0$



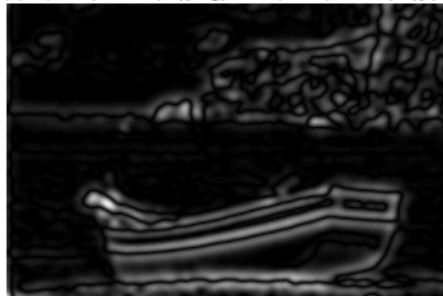
$\left| (f(x, y) * G(\sigma_{n-1}) - f(x, y) * G(\sigma_n)) \right|$



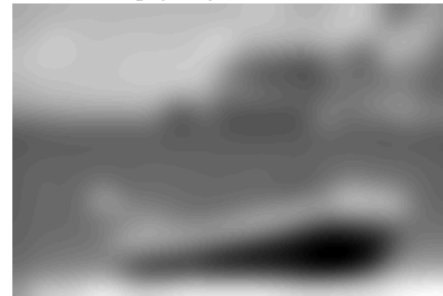
$f_\sigma(x, y), \sigma = 8.0$



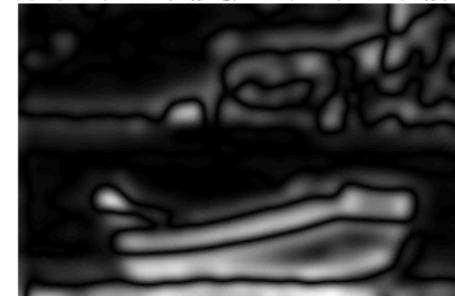
$\left| (f(x, y) * G(\sigma_{n-1}) - f(x, y) * G(\sigma_n)) \right|$



$f_\sigma(x, y), \sigma = 16.0$

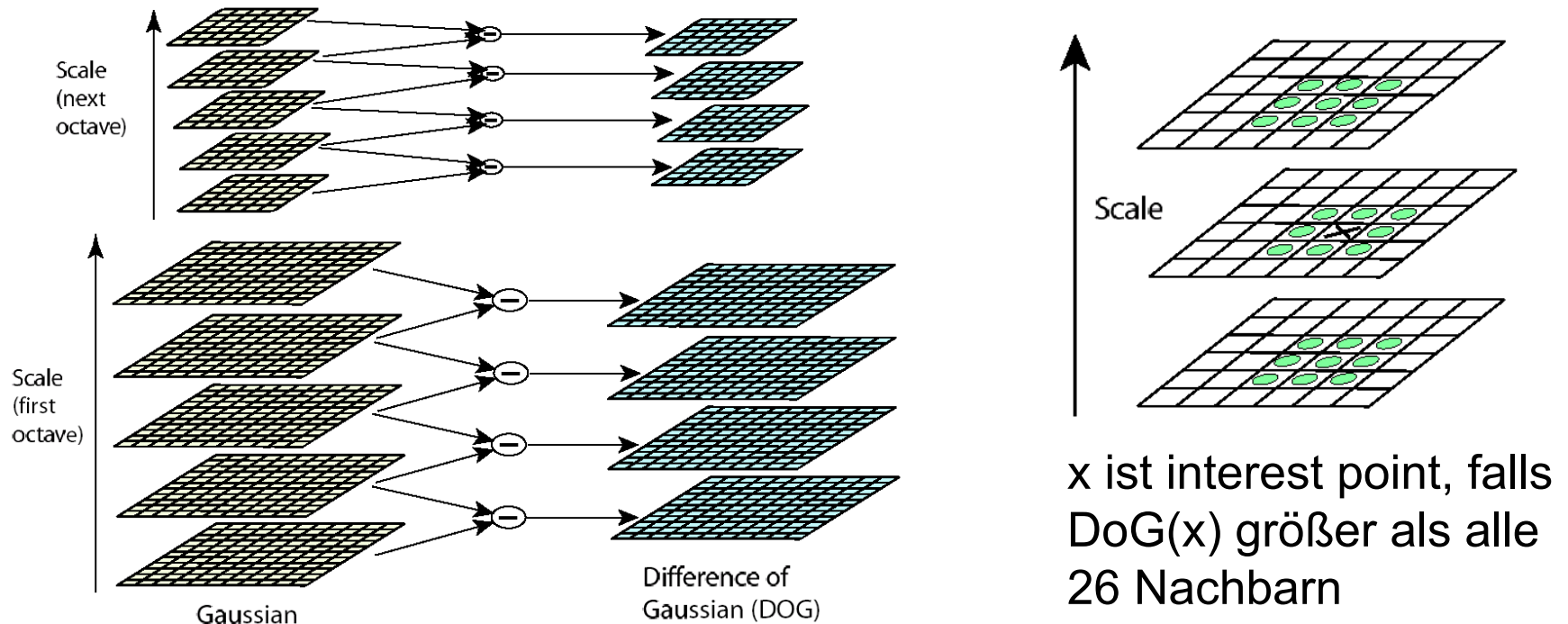


$\left| (f(x, y) * G(\sigma_{n-1}) - f(x, y) * G(\sigma_n)) \right|$



# David Lowe: SIFT Interest Points

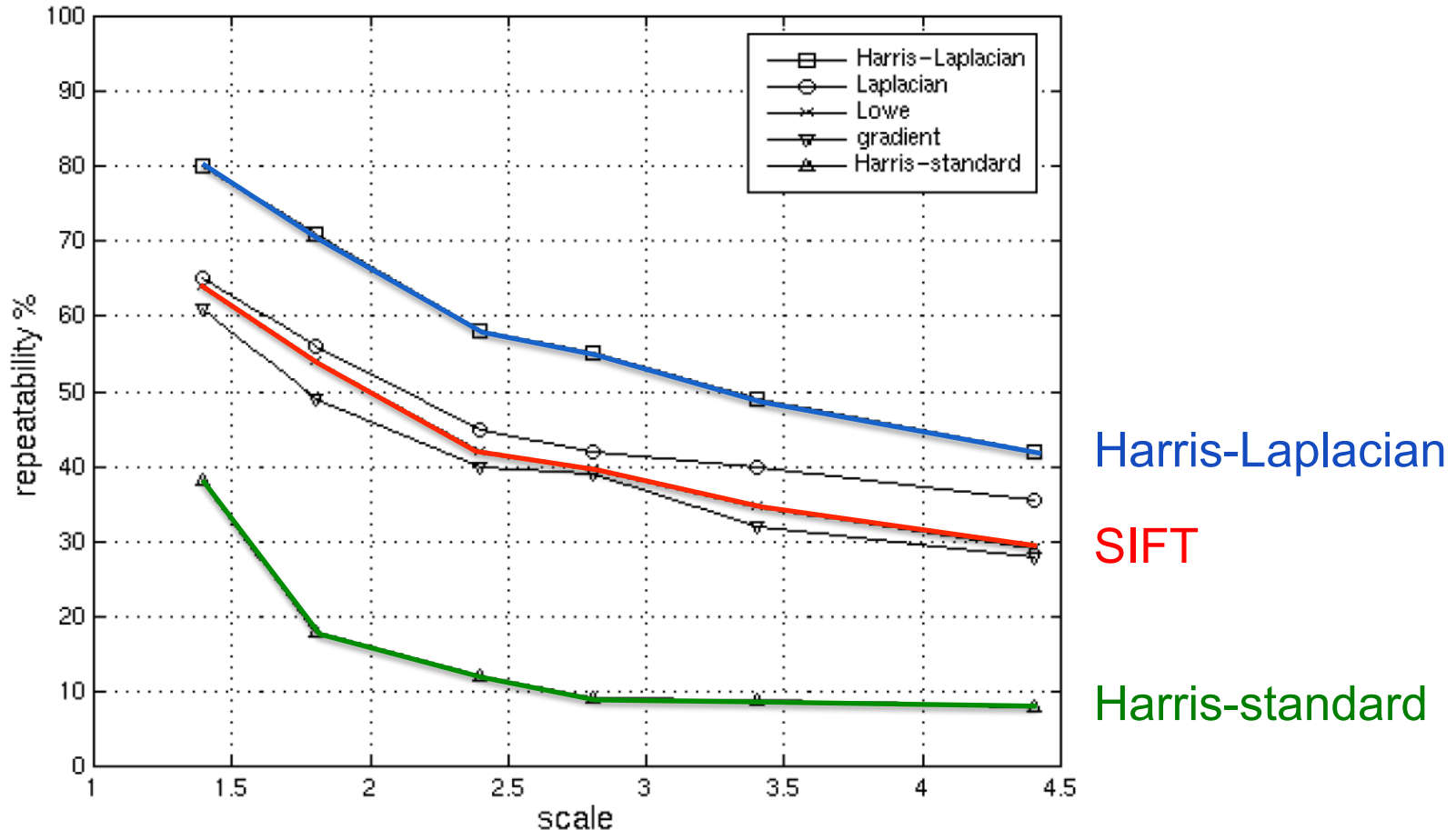
- SIFT = Scale Invariant Feature Transform
- DoG-Operator für Lokalisierung in Bild und Skalierung  
Interest points sind Maxima und Minima in 3D-Nachbarschaft



Quelle Abb.: David G. Lowe: Object Recognition from Local Scale-Invariant Features. Proc. of ICCV 1999.

# Robustheit gegenüber Skalierung

- Harris-Laplacian hat höhere repeatability als SIFT



Mikolajczyk, Schmid: Indexing Based on Scale Invariant Interest Points. ICCV 2001, pp. 525-531

# FEATURE DESCRIPTOREN (MERKMALSVEKTOREN)



# David Lowe: Skalierungsinvariante lokale Merkmale (SIFT)

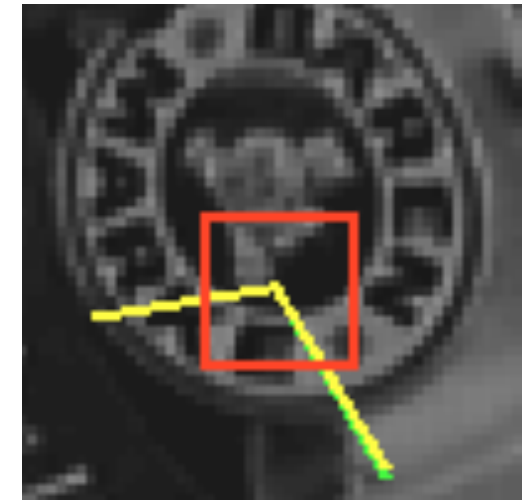
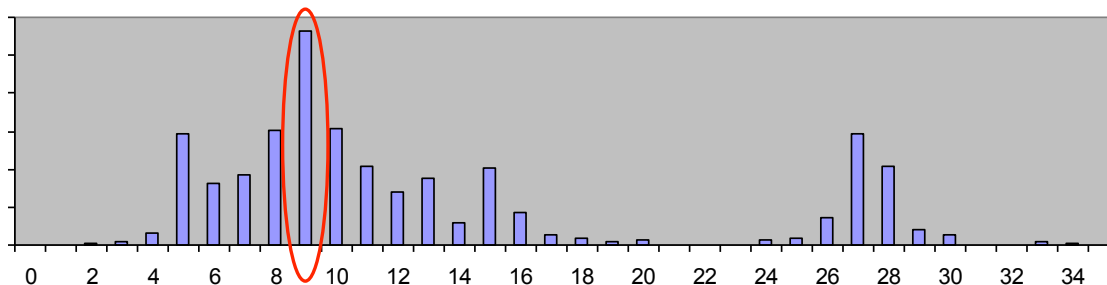
- Lokales Koordinatensystem um interest point
  - invariant gegenüber Translation, Rotation, Skalierung



David G. Lowe: Object Recognition from Local Scale-Invariant Features. Proc. of ICCV 1999.

# David Lowe: Skalierungsinvariante lokale Merkmale (SIFT)

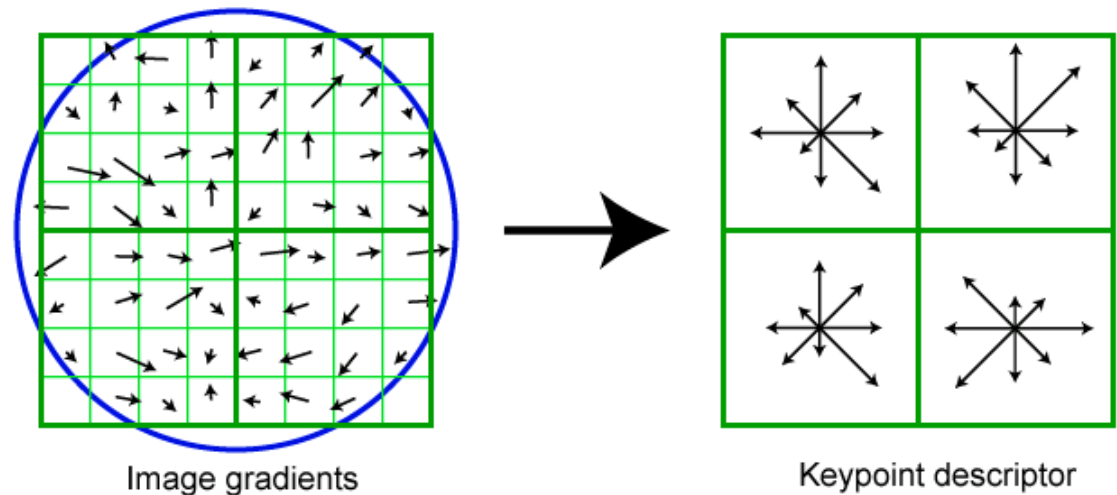
- Skalierungsinvarianz
  - Bestimme charakteristische Skalierung für jedes Merkmal
- Rotationsinvarianz
  - Ausrichtung des lokalen patches entsprechend der dominanten Orientierung der Gradienten
  - Histogramm der Gradientenorientierungen im lokalen patch
    - Auswahl der Maxima im Histogramm
    - falls mehrere Maxima: ein Merkmalsvektor für jedes Maximum
    - falls zu viele Maxima: Unterdrücken des Punktes



David G. Lowe: Object Recognition from Local Scale-Invariant Features. Proc. of ICCV 1999.

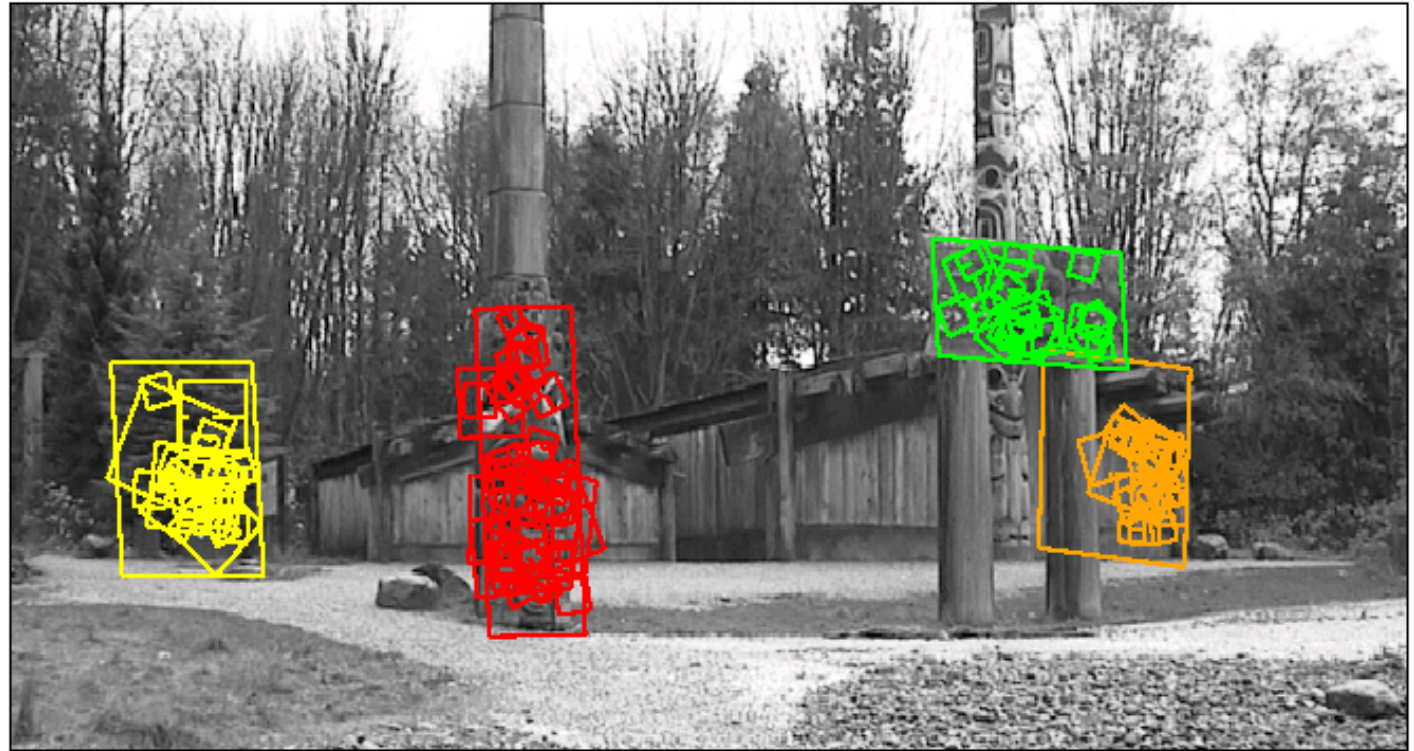
# SIFT Merkmalsvektor

- patch: 16x16 „pixel“ im lokalen Koordinatensystem des den interest points
    - Koordinatensystem lokalisiert in x, y, Skalierung, Orientierung
  - 4x4 Orientierungshistogramme mit je 8 Orientierungen
    - Gewichtet mit Gradientenlänge und Distanz zum Zentrum
- 128 Dimensionen



Quelle Abb.: David G. Lowe: Object Recognition from Local Scale-Invariant Features. Proc. of ICCV 1999.

# Bilderkennung mit SIFT



Quelle Abb.: David G. Lowe: Object Recognition from Local Scale-Invariant Features. Proc. of ICCV 1999.

# SIFT for Mobile Devices [Wagner et al.]

- Variant of SIFT with 36 component feature vector
  - Less computation (original feature vector: 128 components)
- Efficient computation of interest points
  - Variant of FAST corner detector
- Process template at multiple scales



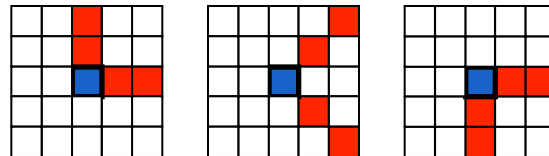
Daniel Wagner, Gerhard Reitmayr, Alessandro Mulloni, Tom Drummond, Dieter Schmalstieg: [Pose Tracking from Natural Features on Mobile Phones](#). Proc. ISMAR 2008.

# Interest Points

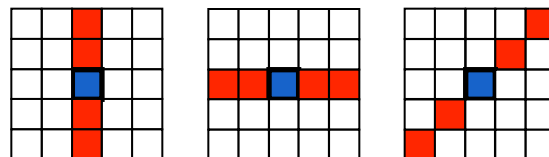
- Choose interest points
  - Can be precisely located in images
  - Robust and repeatable under changing lighting, perspectives, sizes, rotations
  - Surrounding patch descriptive for the image

- Corners are well localized & robust

- Corners:



- No corners:



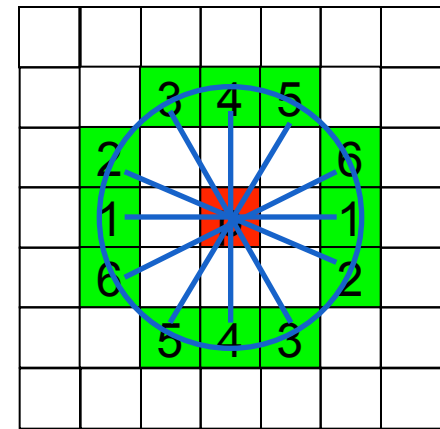
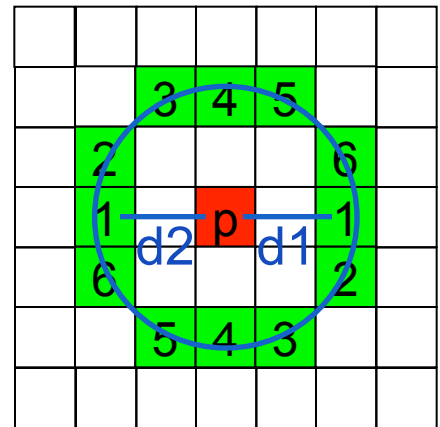
- Corner detection

- Idea: “It’s a corner if it’s not part of a straight line”
- Can be implemented efficiently

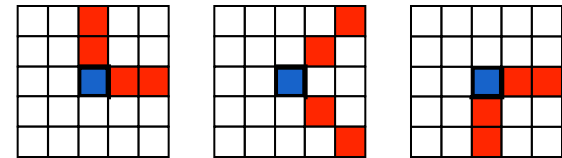


# Interest Points

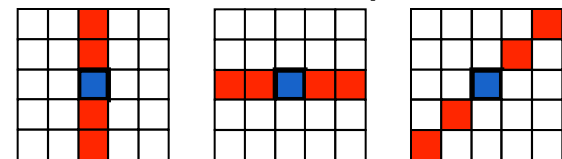
- Corner response of pixel  $p = (x,y)$ 
  - //  $I(p)$  = intensity (grayscale value) of pixel  $p$
  - $d_{min} = \infty$
  - for all opposite points  $(p_1, p_2)$  on circle
    - $d_1 = \text{abs}(I(p) - I(p_1))$
    - $d_2 = \text{abs}(I(p) - I(p_2))$
    - $d = \max(d_1, d_2)$
    - $d_{min} = \min(d_{min}, d)$
  - cornerResponse =  $d_{min}$
- Non-maximum suppression
- Threshold to generate ~150 corners



Corner examples:

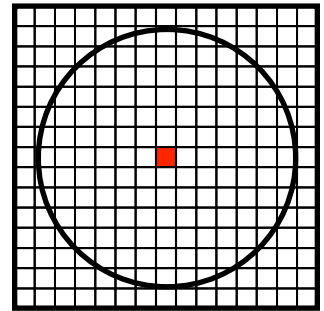


Not corner examples:

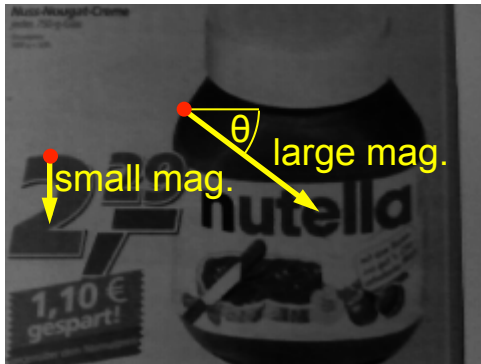


# Mobile SIFT: Feature Descriptors

- Inspect 15x15 pixel patch around corner point
- Compute gradient magnitudes and orientations
  - Convolution with a derivative of Gaussian kernel



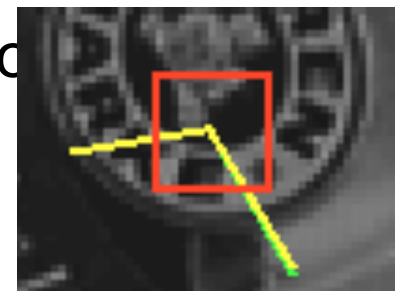
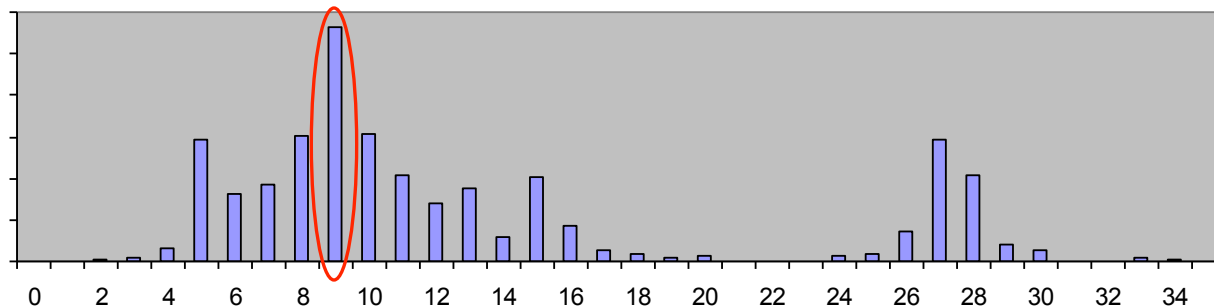
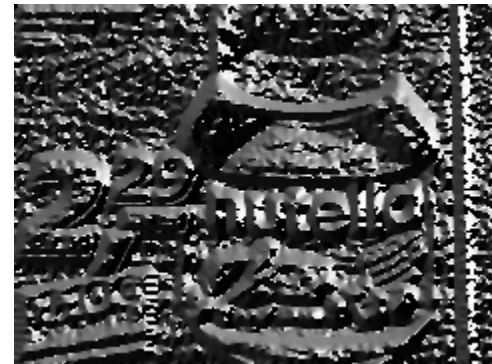
Original image:



Gradient magnitudes:



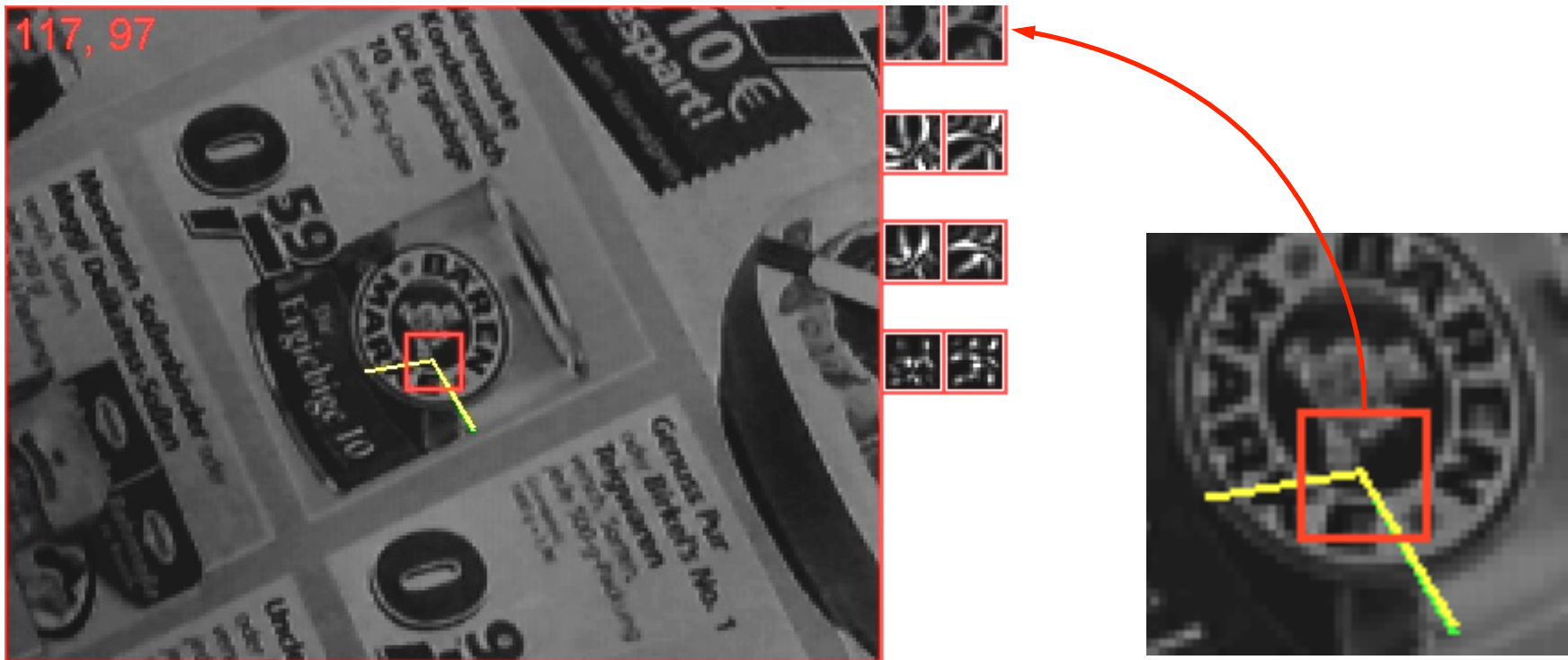
Gradient orientations:





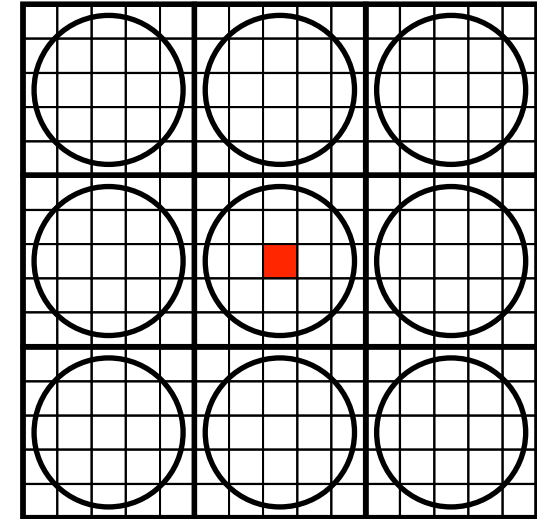
# Mobile SIFT: Feature Descriptors

- Compute dominant orientations (within 80% of max)
  - Weighted by distance patch center
  - Discard feature if too many orientations
- Rotate patch to each dominant orientation

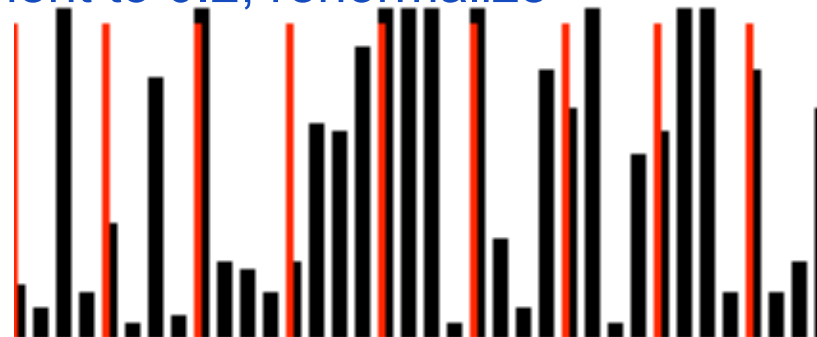


# Mobile SIFT: Feature Descriptors

- Patch rotated to dominant orientation
- Create 3x3x4-component SIFT vector
  - 9 sub-regions (3x3 patches)
  - 4 orientations
- For 3x3 sub-patches with 5x5 pixels each
  - Compute gradient magnitudes and orientations
  - Compute orientation histogram with 4 buckets (each covering 90°)
  - Normalize feature vector (length 1)
  - Limit longest component to 0.2, renormalize



Example feature vector  
with 36 components:



# Mobile SIFT: Scale Space for Template

- Compute SIFT features for multiple scales
- Allows recognizing features over wider range of scales

Scale 0



Scale 1



Scale 2



Scale 3



Scale 4



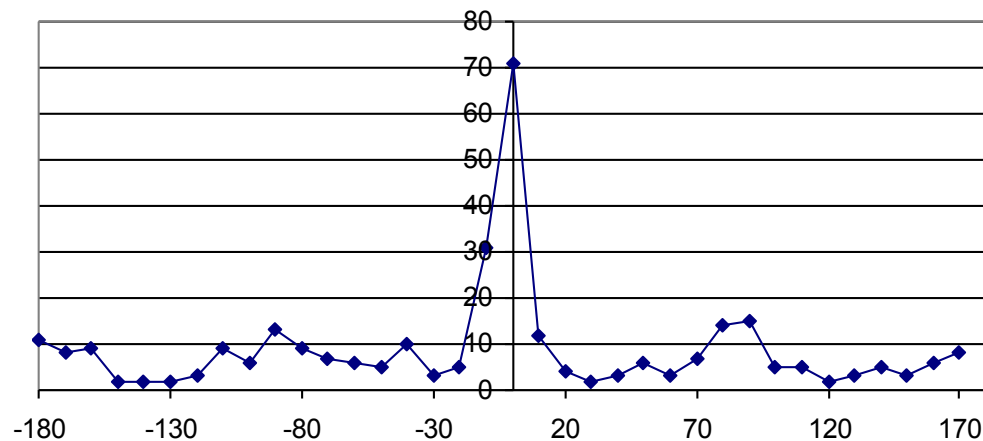
Scale space pyramid  
 $\text{width}(i+1) = \text{width}(i) / \text{sqrt}(2)$

# Mobile SIFT: Feature Matching

- SIFT features are robust to perspective distortion
  - Each individual feature relatively weak (about 30-40% correct matches)
- Find best feature matches
  - Given query descriptor, find closest descriptor in template
  - Distance measure: sum of squared distances (ssd)
  - Match: pairs of SIFT features in camera image and template (minimum ssd)
  - Search: very time consuming for linear search, use approximate nearest neighbor KD tree
    - [Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. International Conference on Computer Vision Theory and Applications (VISAPP), Lisbon, Portugal, Feb. 2009.]
    - [Silpa-Anan, Hartley: Optimised KD-trees for fast image descriptor matching. CVPR 2008.]

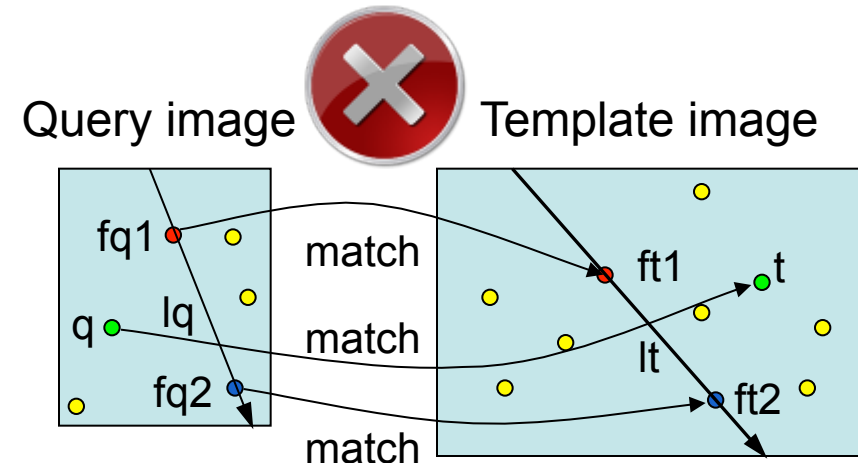
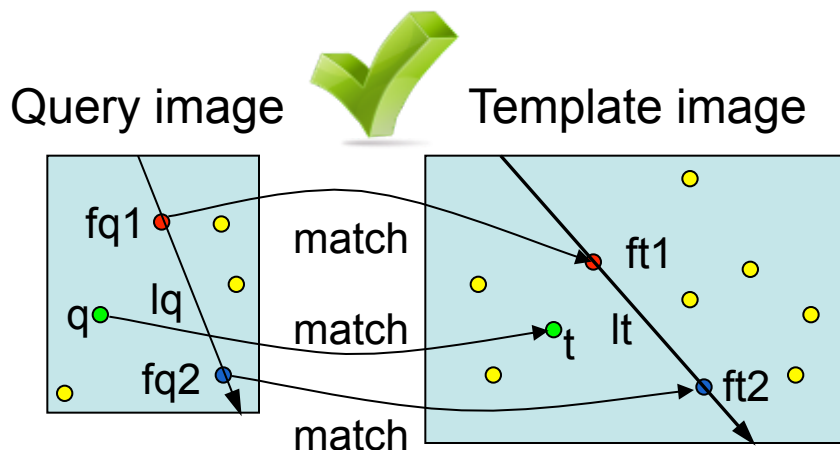
# Mobile SIFT: Feature Matching

- Found feature matches may be wrong
  - Might not be identical, even though descriptor very close
- Removing outliers from feature matches
- Orientation difference test
  - For all matches compute orientation differences
  - Remove all matches  $\geq \pm 30^\circ$  from histogram peak
  - Works best for planar scenes



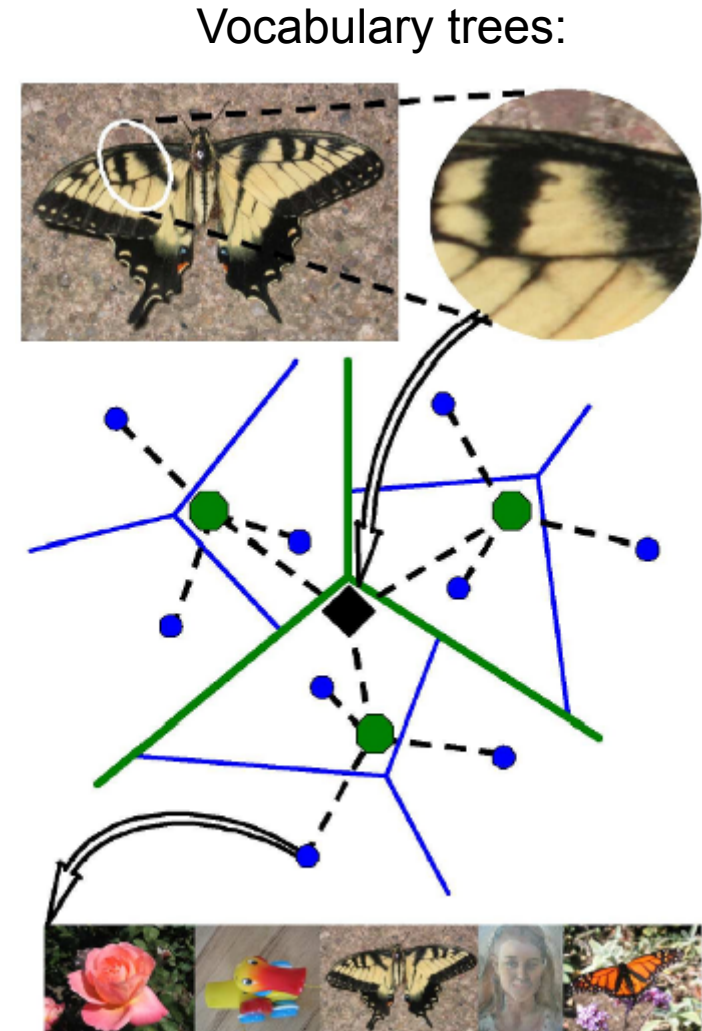
# Mobile SIFT: Feature Matching

- Line test
  - Select two “good” (small  $ssd$ ) matches:  $(fq1, ft1)$ ,  $(fq2, ft2)$
  - Compute line  $lq$  through  $(fq1, fq2)$  in query image
  - Compute line  $lt$  through  $(ft1, ft2)$  in template image
  - For other matches  $(q,t)$ : position of  $q$  to  $lq$  should be same as  $t$  to  $lt$
  - Measure number of inliers for  $(lq, lt)$
  - If inlier rate  $>70\%$  then assume line is correct and remove outlier matches



# Matching Images in large Databases

- Scalability
  - Find matching image in large database
- Vocabulary trees
  - Term Frequency Inverse Document Frequency (TF-IDF)
  - [Nister, Stewenius: Scalable Recognition with a Vocabulary Tree. CVPR 2006.]



**THE END**

