MMI 2: Mobile Human-Computer Interaction

Camera-Based Mobile Interaction

Prof. Dr. Michael Rohs

michael.rohs@ifi.lmu.de

Mobile Interaction Lab, LMU München

Lectures

#	Date	Topic
1	19.10.2011	Introduction to Mobile Interaction, Mobile Device Platforms
2	26.10.2011	History of Mobile Interaction, Mobile Device Platforms
3	2.11.2011	Mobile Input and Output Technologies
4	9.11.2011	Mobile Input and Output Technologies, Mobile Device Platforms
5	16.11.2011	Mobile Communication
6	23.11.2011	Location and Context
7	30.11.2011	Mobile Interaction Design Process
8	7.12.2011	Mobile Prototyping
9	14.12.2011	Evaluation of Mobile Applications
10	21.12.2011	Visualization and Interaction Techniques for Small Displays
11	11.1.2012	Mobile Devices and Interactive Surfaces
12	18.1.2012	Camera-Based Mobile Interaction
13	25.1.2012	Sensor-Based Mobile Interaction
14	1.2.2012	Application Areas
15	**8.2.2012**	**Exam**

Aktuelles

- Klausur am 8.2.2012
  - Anmeldung
- Fragen zur Klausur
  - jeweils zu Beginn der Vorlesungen

Review

- What is “collapse-to-zoom”?
- What is the advantage of “wedge” over “halo”?
- What is “pseudo transparency”?
- What is Buxton’s “three-state model of input”?

Preview

- Physical hyperlinking (“mobile tagging”)
- Visual codes for camera phones
- Image recognition
- Optical movement detection
- Target acquisition with camera phones

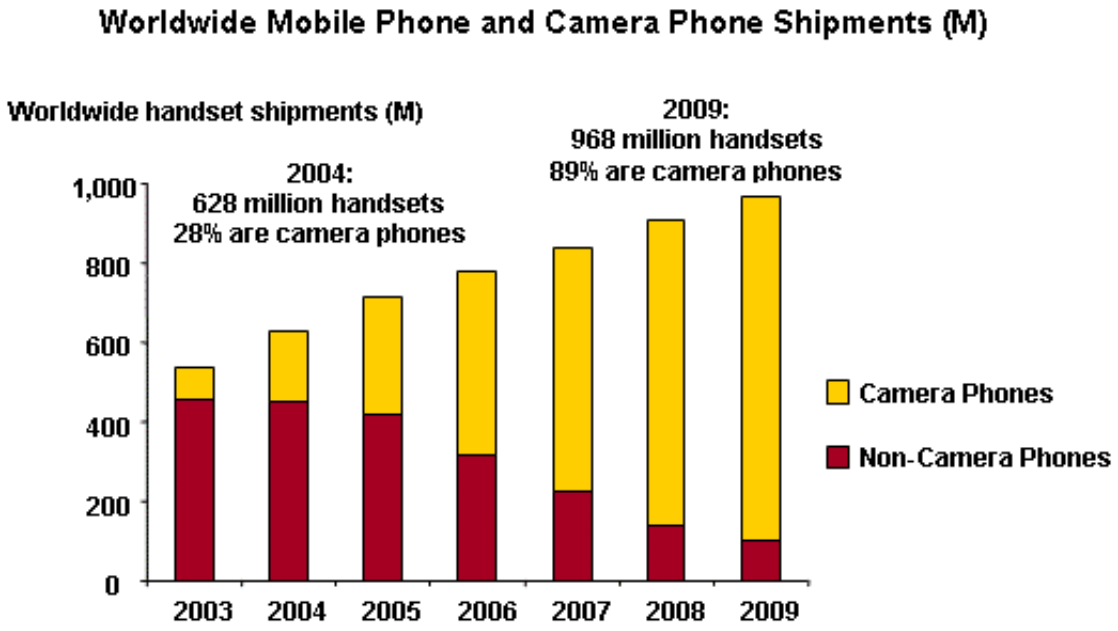
CAMERA-BASED MOBILE INTERACTION

Integrating Cameras in Mobile Devices

- Taking snapshot of surroundings
  - Camera phones
- Additional data input channel
  - 1D / 2D barcodes
- Bridging different media types
  - Paper and electronic media, mobile devices and electronic displays
- Linking mobile devices
  - Authentication between devices via the visual channel
- Overlaying information onto the real world
  - Augmented reality
- Creating input devices
  - Optical movement detection
- Server-based image recognition
  - Server analyzes uploaded image

The Ubiquitous Camera Phone

- 2000: 1st camera phone (Sharp J-SH04)
  - 110k pixel CMOS sensor
- 2009: 89% of mobile phones shipped with integrated camera



First camera phone (2000)
Sharp J-SH04
110k pixel CMOS sensor

Source: Jeff Hayes, InfoTrends CAP Ventures, http://www.capv.com/home/Multiclient/MobileImaging.html

Camera Phones for Physical Interaction

- Linking the physical to the virtual world
  - The environment as part of the interface
  - Integration with the user's activities
- Camera phones as "bridging" devices
  - Always available imaging device
  - Continuous wireless connectivity
  - Processing power enables on-device image processing
  - Display and audio capabilities
- Handheld camera vs. fixed camera
  - Traditionally predominantly fixed cameras

Categories of Camera-based Interaction

Type of image data	Site of image processing	Type of image processing	On-device processing requirements	Application
Single image	none	Image capturing	low	MMS, human-human, documentation
Single image	server	Advanced image recognition on server	low	“Tourist guide” applications
Single image	mobile device	Simple image analysis	low / medium	Marker recognition
Video stream	none	Showing live video stream	medium	“Reality browsers”
Video stream	mobile device	Simple real-time image analysis	medium	Continuous marker recognition
Video stream	mobile device	Simple real-time optical flow analysis	medium	Optical movement recognition
Video stream	mobile device	Markerless tracking algorithms	high	Augmented reality

Issues of Camera-based Interfaces

- Digital cameras: Sources of rich sensor data
  - Interpretable by humans and machines
  - Can be processed in many ways
- Issues of perceptual interfaces (computer vision, gesture recognition, speech recognition)
  - Potential for recognition errors
  - Impact on user experience depends on application
- Issues of camera-based interfaces
  - Recognition errors
  - Delay for processing (responsiveness)
  - Dependence on lighting conditions
  - Needs a lot of computational resources

IMAGE CAPTURING WITHOUT IMAGE RECOGNITION

Snapshots for Documentation

- Taking snapshots for documenting the real world
  - Usage model: take snapshot, upload to server, send to others, discuss
- Example: architect on construction site
- No image processing required

First camera phone (2000)
Sharp J-SH04
110k pixel CMOS sensor

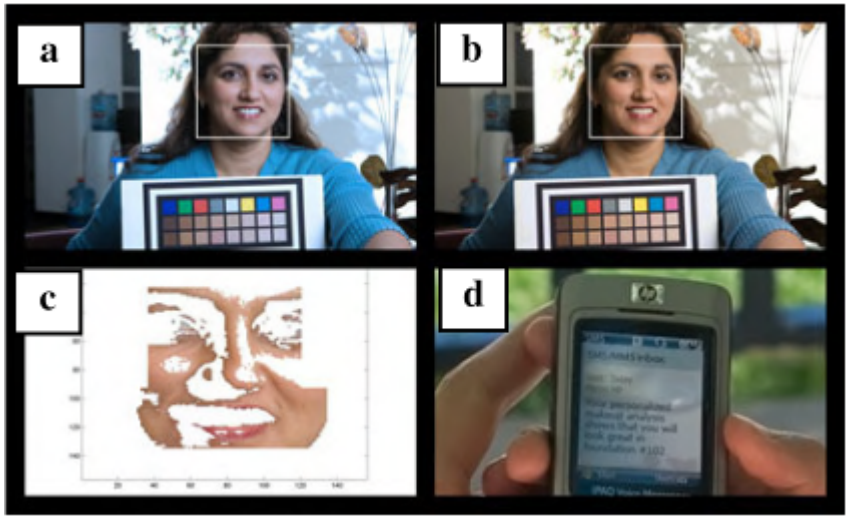
PhotoMap: Georeferenced Snapshots of Specialized "You are Here" Maps

- Problem: Standard maps often don't show special areas such as parks, hiking trails, campuses
- Solution: Camera device with GPS positioning
  - Take image of paper map
  - Scroll to "You are here"
  - Phone associates map position and GPS position
  - Current position on photo updated by GPS

Cheverst, Schöning, Krüger, Rohs: Photomap: Snap, Grab and Walk away with a "You are Here" Map. MIRW 2008.

ColorMatch (Jain et al., MobileHCI 2008)

- Mobile cosmetic advisory system
- Help to select colors matching to skin color
- Match colors of clothes



- Social aspects
  - Take color card image in public vs. private
  - Trust in advisory system
  - Asking friends about system's recommendation
- No installation of software necessary

Source: Jain et al.: Color Match: An Imaging Based Mobile Cosmetics Advisory Service. MobileHCI 2008.

“Reality Browsers”

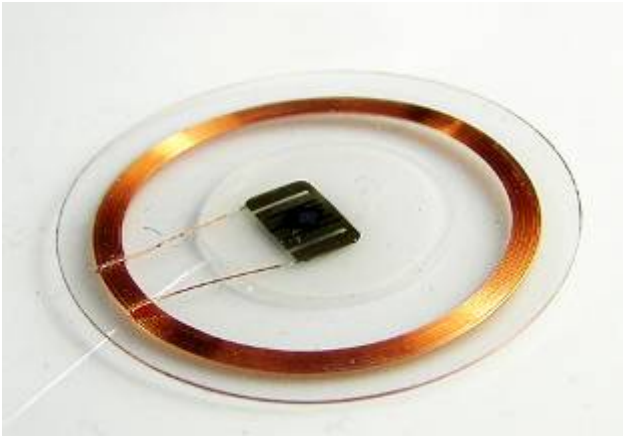
- Show augmented video stream
- No image processing, other sensors
  - GPS
  - Accelerometer
  - Magnetometer
- Examples
  - Layar
  - Wikitude
- Limitations
  - No real registration with the real world view

MOBILE TAGGING

Object Tagging
1D, 2D Barcodes and RFID/NFC

- 2D barcodes

- RFID / NFC readers



Hyperlinks in Cinema Magazines

- Physical hyperlinks to specific URLs associated with individual regions
- Linking printed documents with online services
- Example services
  - Ordering cinema tickets
  - Movie trailers
  - Soundtracks
  - Information on actors
  - Rating movies (suspense, fun, action, difficulty)

QR Codes: Mass Market in Japan

- Used in newspapers, business cards, coupon flyers, etc.
- 72% of Japanese users have phone with QR code reader
- 80% of customers with enabled phones used the feature

(2007)

NWA QR code campaign billboard at Shinjuku station, October 2005

Europe: Newspapers start to use QR Codes in 2007

- Link to Web pages, movie ratings, movie trailers, online news

WELT KOMPAKT

FREITAG, 9. NOVEMBER 2007 + Redaktionsschluss 23:08 Uhr + B / NR. 218 / 70 CENT

Umfrage: Merkel ist beliebter als der Papst
Politik, Seite 4

Bier soll 40 Prozent teurer werden
Wirtschaft, Seite 12

Streik lässt die Bänder still stehen

Zeitung Handy Internet

WELT KOMPAKT ist ab heute Ihr Link ins mobile Internet

Ein magisches Quadrat macht's möglich – Mit dem Handy draufhalten, schon geht's los

NACHRICHTEN

SCHMIERGELDSKANDAL
Neue Dimension

UEFA-POKAL
Niederlagen und Unentschieden

SHERWOOD FOREST
Robin Hoods Revier in Gefahr

SCHLUSSKURSE
Dax stärker, Dow schwächer

ZUGREIFEN!

WELT ONLINE
GESTERN GEKLICKT
Die Favoriten auf www.welt.de

SMS-Service

Wie gesund ist Tee wirklich? Klinische Beweise für die Wohltaten gibt es nur wenige Leben, Seite 25

Ihr Name als Programm

Dirty Pretty Things zeichnen auf dem zweiten Album ein trauriges Sittenge

Von Daniel-C. Schmidt

Dass sich Großbritannien in Sachen Kultur oftmals im Auge des Sturms befindet, ist zuweilen beneidenswert. Bis sich aus dem Epizentrum des guten Geschmacks die jüngsten Moden aufs deutsche Festland verirren, vergeht manch Monat.

Aber! Nicht gleich über jedes Stöckchen springen. Denn dass auf der Insel nicht alles allererste Sahne ist, davon singen Dirty Pretty Things diverse, mehrstrophige Lieder auf ihrem zweiten Album „Romance At Short Notice".

Dirty Pretty Things? Ist etwas kompliziert, aber schnell erzählt: Es war DPT-Sänger Carl Barât, der zusammen mit Peter Doherty The Libertines gegründet hatte. Als die sich 2004 auflösten und Doherty schon parallel mit seinen Babyshambles ein neues Projekt aufgestellt hatte, gönnte sich Barât etwas mehr Zeit, bevor er Dirty Pretty Things ins Leben rief. 2006 kam das Debüt „Waterloo To Anywhere" heraus.

Beide, Doherty und Barât, laufen seit der Libertines-Trennung ihrer Form hinterher. Dankbar für den historischen Vergleich machten die Medien aus den beiden talentierten Knaben eine neue Konstellation à la Strummer/Jones oder Morrissey/Marr.

In eingeschworener Verbundenheit und zugleich konkurrierend, schaukelten sich Doherty/Barât zu musikalischen Höchstleistungen auf – die sie in ihren neuen Bands ohne den anderen nur selten wiederholen konnten.

Während Doherty dann auch immer mitleidiger wird, haben sich Barâts DPT für ihr zweites Album ein interessantes Sujet ausgesucht. Sie verfrachten die altmodische Heimatliebe der Libertines ins Vereinigte Königreich der Jetztzeit: Irgendwas ist faul im Staate England. Die erste Single vom neuen Album heißt bezeichnend „Tired of England".

„Ich selbst werde nicht müde, ich liebe die Welt. Natürlich gibt es Dinge, die mir auf den Sack gehen. In England geht es beispielsweise momentan drunter und drüber", erzählt Carl Barât WELT KOMPAKT im Interview.

DPT-Gitarrist Anthony Rossomando sieht die Lage ähnlich: „Das Lied nimmt auf, was in unserer Umgebung geschieht – echt am A
abgeht – d
die Messe
sind vollk
Aber ‚Tir
auch Opti
Das and
schwingt
Romantik
die treibe
Bösewicht
geben."
DPT er
Indie-Roc
verlassen
Sinn für s
lodien. U
leisten, ist
beobachtu
ben: Es
Tors, sich
lieren, um
men.

Dirty Pretty Things mit „Tired of England". Gelangweilt klingen sie aber nicht:

Dirty Pretty Things At Short Notice" (Uni

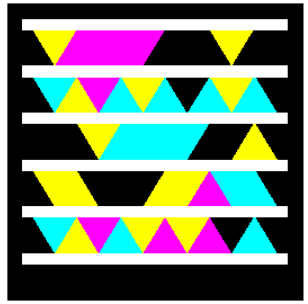
Berlin U Bahn

Google “Favorite Place” Window Decals

- “Favorite Places”
  - Based on search rankings in Google and Google Maps, Google sent display window decals to over 100,000 businesses
- QR code to access reviews and coupons

http://gizmodo.com/5420737/in-the-future-we-all-will-be-google+approved
http://www.google.com/help/maps/favoriteplaces/business/barcode.html

Microsoft Tag

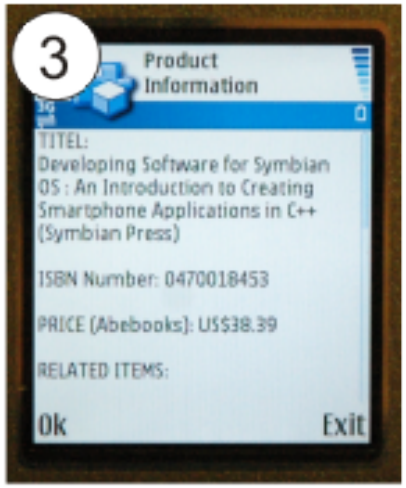


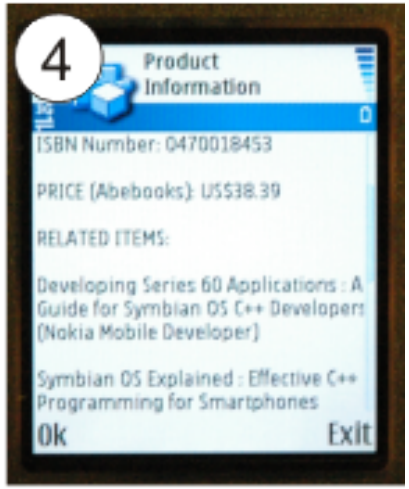
- “High Capacity Color Barcode” (HCCB)
  - 4 colors (2 bits per triangle)
  - 5 x 10 triangles = 100 bits
- Tag types
  - URL, free text, vCard, dialer
- http://tag.microsoft.com

4 color barcode
storing 1 byte (8bits)
Uses 4 symbols

1D Barcode Recognition by Camera

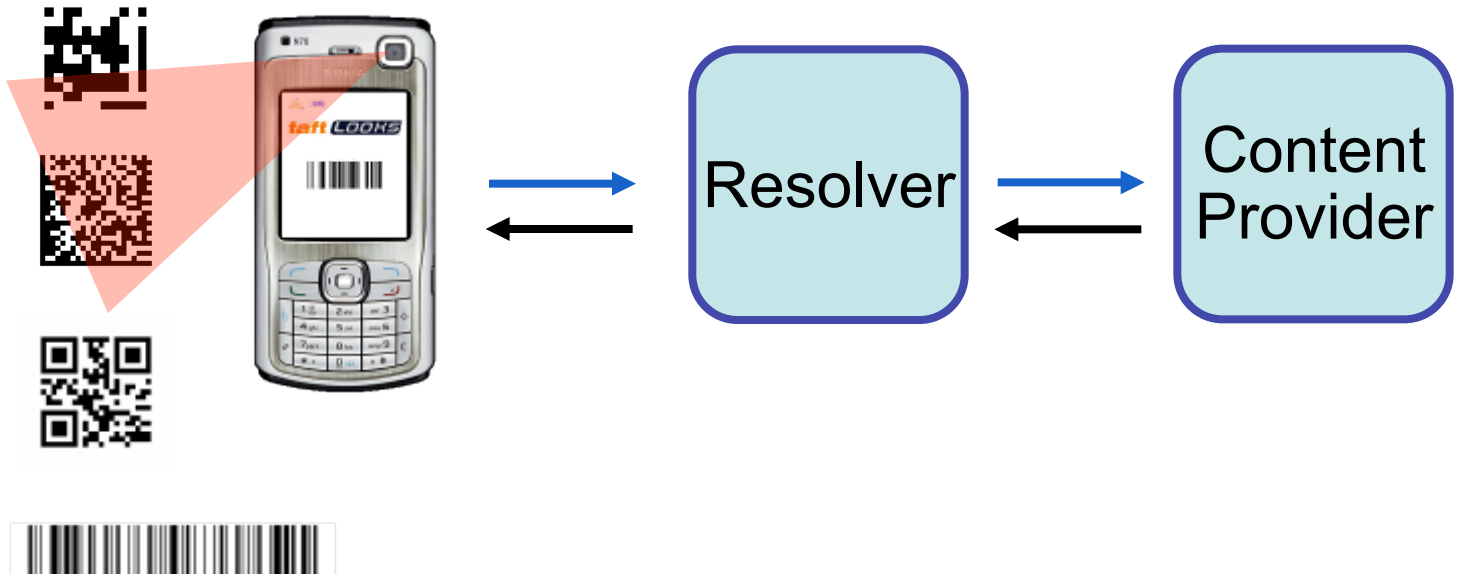
- 1D barcodes on every retail item
- Camera resolution now sufficient to resolve lines
- Free toolkit (GPL): BaToo
  - people.inf.ethz.ch/adelmanr/batoo





Resolving Identifiers

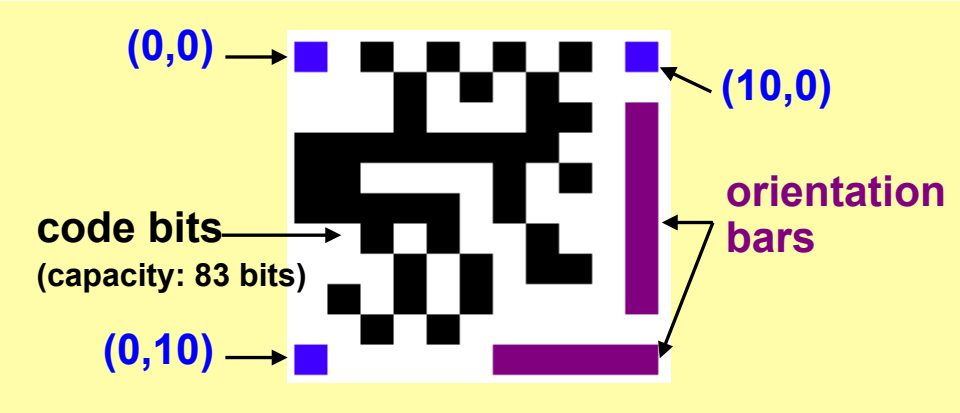
- Markers contain content or link to content
  - Direct (no resolver): store URL, phone number, text
  - Indirect (resolver): store identifier that resolver maps to content



VISUAL CODES FOR CAMERA PHONES

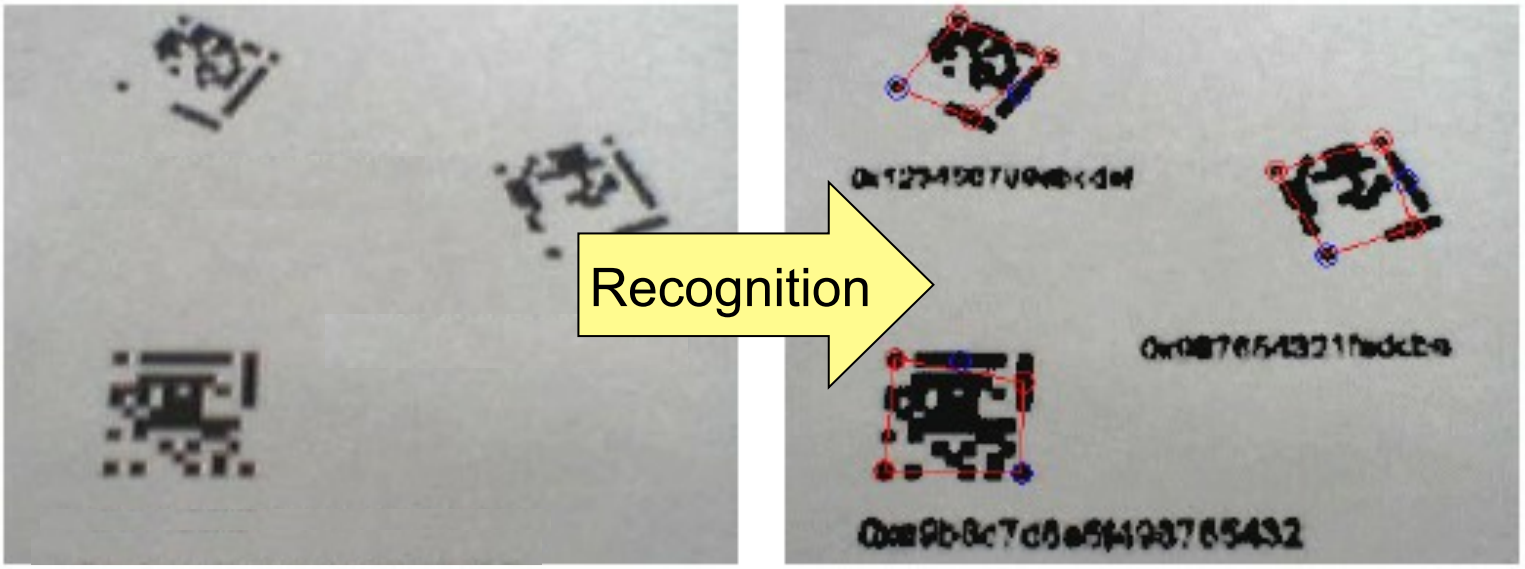
Visual Codes for Camera Phones

- For low-resolution phone cameras
  - Requires coarsely grained code
- Lightweight recognition algorithm
  - Real-time recognition in video stream
- 76 bits of data
  - Sufficient for IP address + port, Bluetooth address, UUIDs



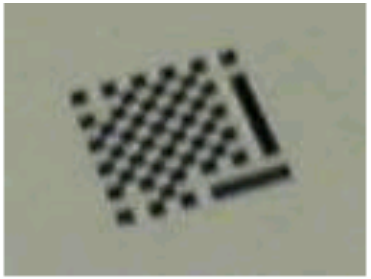
Low Quality Camera Images

- Low resolution
- Blurred edges
- Low contrast
- Uneven illumination
- Radial lens distortion (“barrel distortion”)



Grayscaling and Adaptive Thresholding

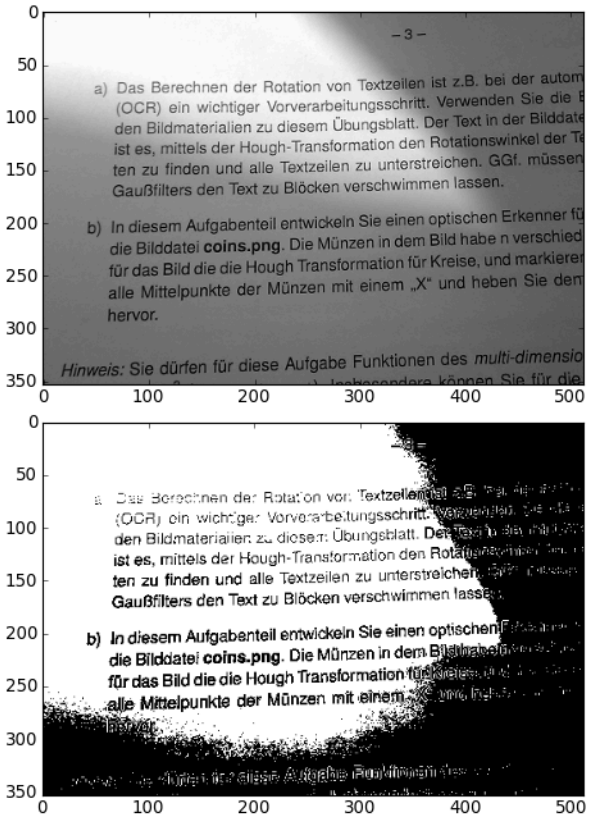
- Grayscaling: grey = (red + green) / 2
  - more efficient than
    Y = 0.2126 × red + 0.7152 × green + 0.0722 × blue
  - good approximation
    Y = (218 * red + 732 * green + 74 * blue) >> 10



- Adaptive thresholding algorithm by Wellner
  - Global threshold problematic
    due to uneven illumination
  - Traverse scan lines top to bottom
  - Adaptive threshold (moving average)
  - Modified to avoid floating point operations

Adaptives Thresholding

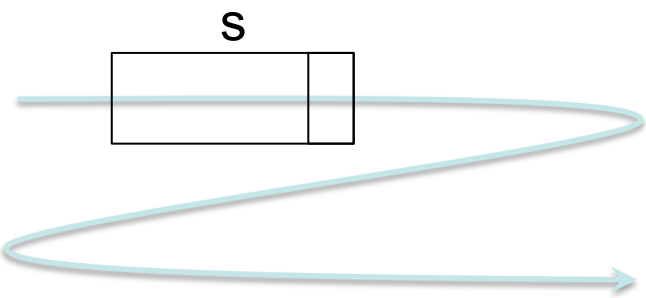
- Globale Schwellenwerte problematisch
  - bei Shading nicht oft mehr global (für das gesamte Bild) definierbar
- Variable Schwellenwerte
  - Schwellenwert wird an jedem Punkt im Bild neu berechnet
  - typische Strategie: zeilenweises Durchlaufen im zick-zack, gleitender Durchschnitt der letzten n Grauwerte



Wellner's Adaptive Thresholding

- Zeilenweises Durchlaufen im Zick-zack
- (*s*-facher) Durchschnittsgrauwert der letzten *s* Pixel:

$$g_s(n) = g_s(n-1) \cdot \left(1 - \frac{1}{s}\right) + p_n, s = 30..80$$



- Initialisierung: $g_s(0) = s\frac{c}{2}$
- Schwellenwert:

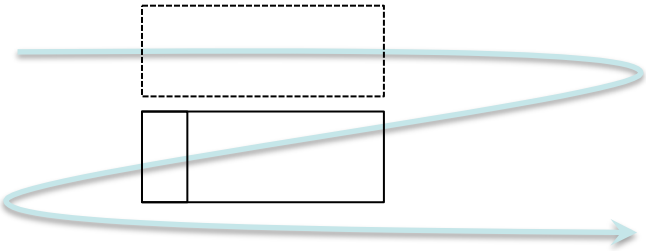
$$T(n) = \begin{cases} 0 & falls\ p_n < \frac{g_s(n)}{s} \cdot \frac{100-t}{100} \\ 1 & sonst \end{cases}, t = 15$$

Pierre D. Wellner. Adaptive thresholding for the DigitalDesk. Technical Report EPC-93-110, Rank Xerox Research Centre, Cambridge, UK, 1993.

Wellner's Adaptive Thresholding

- Verbesserung: Berücksichtigung des durchschnittlichen (s-fachen) Grauwerts über dem aktuellen Pixel

$$h_s(n) = \frac{1}{2}\left(g_s(n) + g_s(n-w)\right)$$



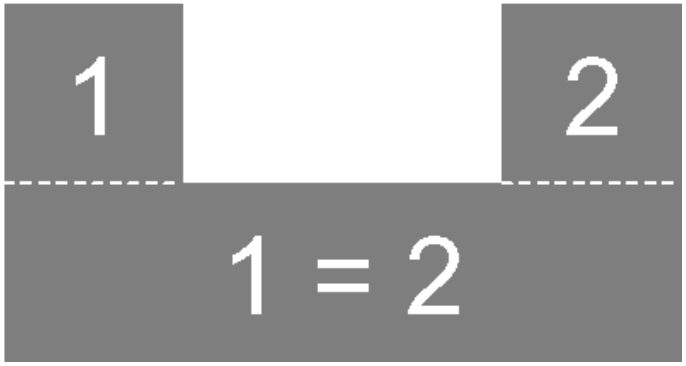
$$T(n) = \begin{cases} 0 & falls\ p_n < \dfrac{h_s(n)}{s} \cdot \dfrac{100-t}{100} \\ 1 & sonst \end{cases}, t = 15$$

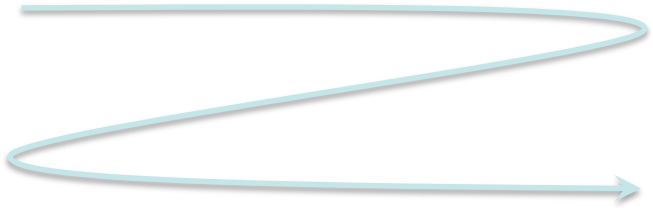
Pierre D. Wellner. Adaptive thresholding for the DigitalDesk. Technical Report EPC-93-110, Rank Xerox Research Centre, Cambridge, UK, 1993.

Labeling Phase 1: Vorläufige Label

- Zusammenhängenden Bereichen Label zuordnen
- Zeilenweises Durchlaufen im Zick-zack
- Bei Vordergrund-Pixel p: betrachte Nachbarpixel darüber und links/rechts davon
  - falls beide Nachbarn Vordergrund, dann Label(p) = Label(darüber)
    - falls Label(darüber) ≠ Label(links/rechts), dann markiere beide Label als äquivalent (für späteres Mergen der Äquivalenzketten)
  - falls einer der Nachbarn Vordergrund, dann setze Label(p) auf dessen Label
  - falls kein Nachbar Vordergrund, dann vergebe neues Label für p
    - füge in eigene Äquivalenzkette ein

Labeling Phase 2: Auflösen der Äquivalenzen

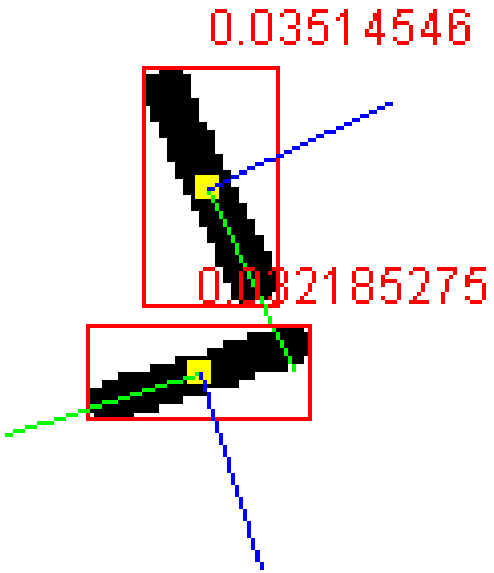




- Label-Äquivalenzen werden möglicherweise erst später erkannt (→ Durchlauf zeilenweise)
- Labeling 2. Phase: Auflösen der Äquivalenzen
  - Druch Äquivalenzkette laufen und jedem Pixel in der gleichen Äquivalenzkette das gleiche finale Label geben

Calculate Region Statistics: Size, Shapes and Orientations

- Compute second-order moments
  - For symmetric regions: axes of symmetry
- Gives information about orientation and “ellipsity” of regions
  - ratio = 0 for lines, 1 for circles
- Example:

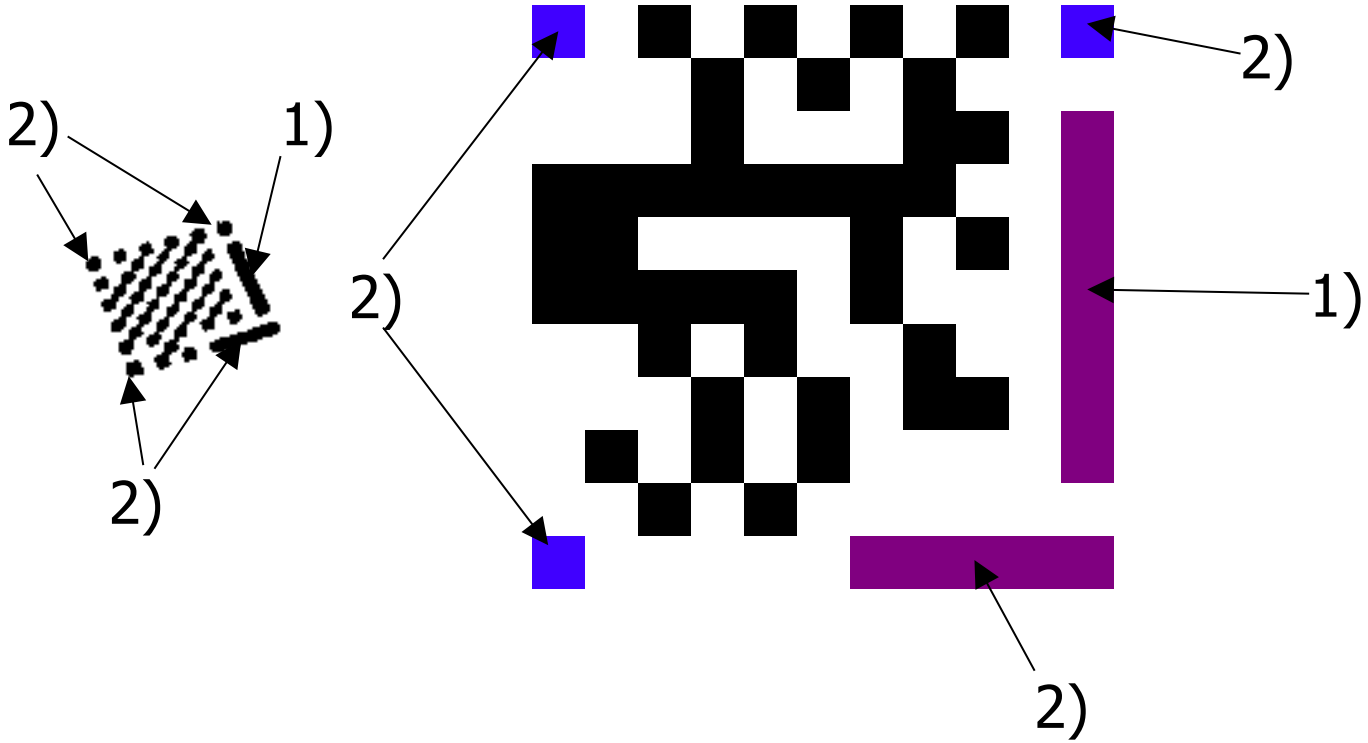


Locating Codes in the Camera Image

1) Search region that looks like a bar

Locating Codes in the Camera Image

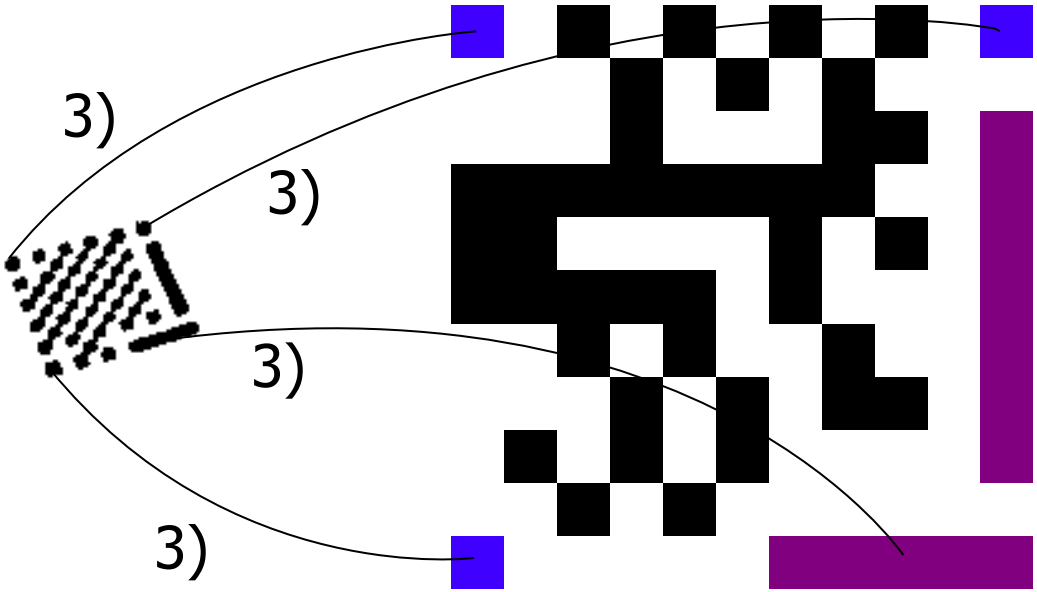
2) Check whether other features are present



Reading the Encoded Bits

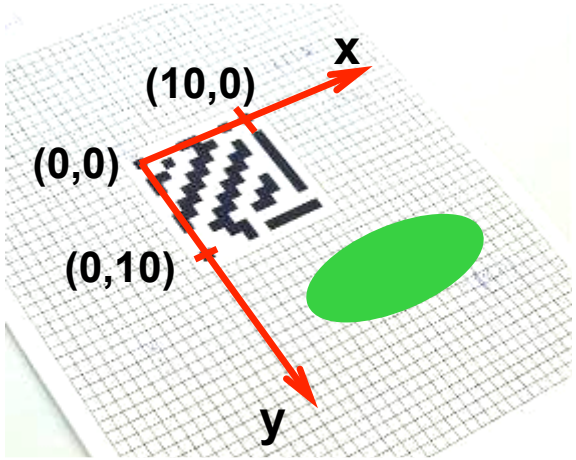
3) Distortion correction with projective mapping (homography)

4) Error detection with (83,76,3) linear code



Visual Code Parameters

- Rotation, tilting, and distance
- Code coordinate system
- No camera calibration required
- Enables intuitive manipulation



code coordinate system

distance

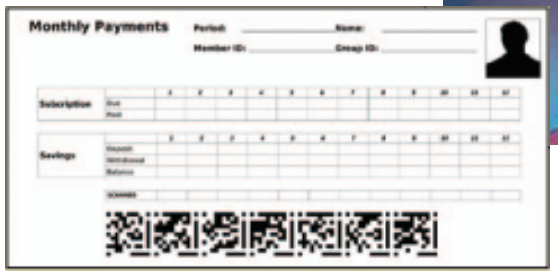
rotation

tilting

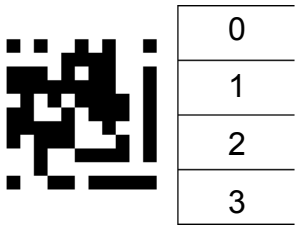
Michael Rohs: Real-World Interaction with Camera Phones. Proc. of UCS 2004.

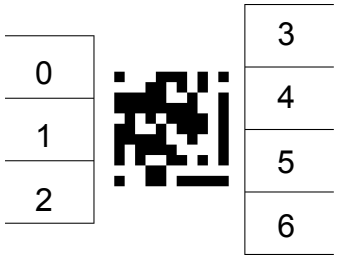
Academic Projects using Visual Codes

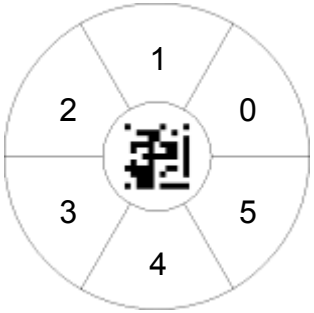
- Mitchell, Race, and Clarke: CANVIS: Context-Aware Network Visualisation using Smartphones. MobileHCI 2005
  - Real time monitoring and visualization of a campus network
- Parikh: Using Mobile Phones for Secure, Distributed Document Processing in the Developing World. IEEE Pervasive Computing, 4(2):74
  - Embed processing commands into paper forms
  - Mobile phones more common than PCs

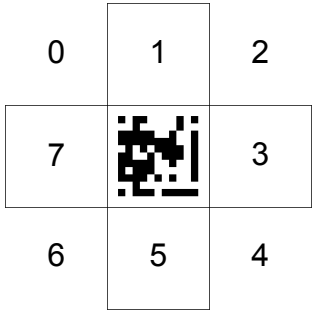


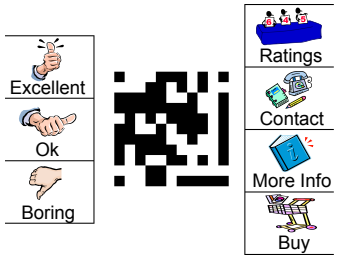
Visual Code Menus

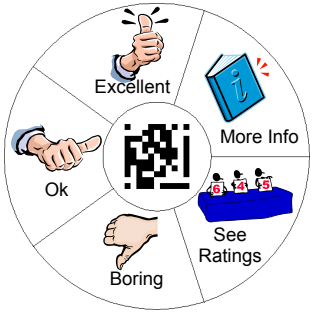


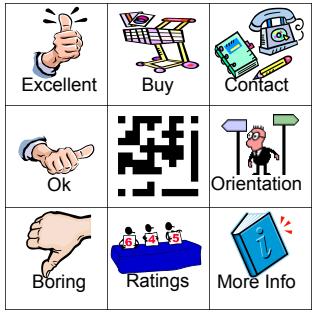


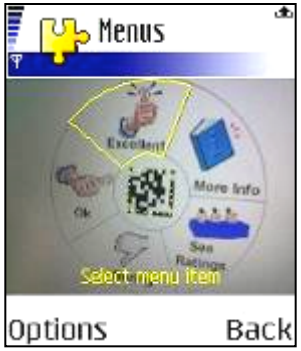






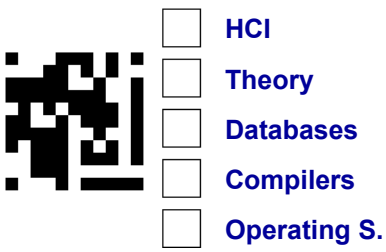




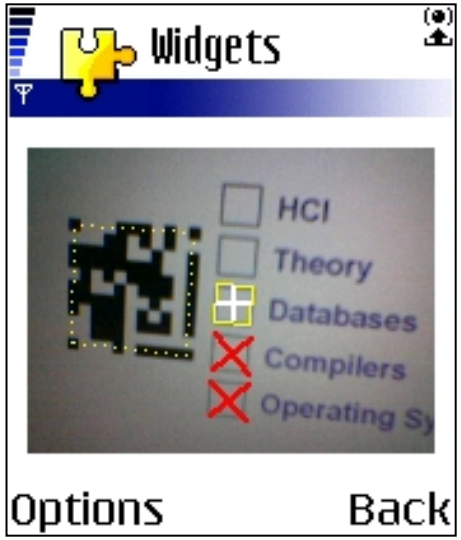


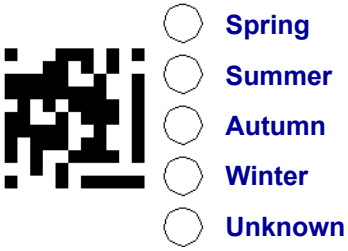
Check Boxes and Radio Buttons

- Selection from printed media
  - Feedback forms, machine configuration, ratings
- Graphical overlay of widget state



(a) check boxes





(b) radio buttons

IMAGE RECOGNITION

Server-based Image Recognition

Source: Rahul Swaminathan, T-Labs

Server-based Image Recognition

- Landmark recognition under varying illumination and pose
  - Time of day, weather conditions
- Creating landmark database
  - Keeping database up-to-date
- Location to restrict search space
  - GPS, GSM cell id
- Applications
  1. Advertisements
  2. Museum guide
  3. Tourist guide

Source: Rahul Swaminathan, T-Labs

Visual Search

- Camera phones recognize the world around us
- Example: Google Goggles for Android
  - Visual search queries for the Web
  - Recognizes a wide range of artifacts
  - Text translation
- Privacy?

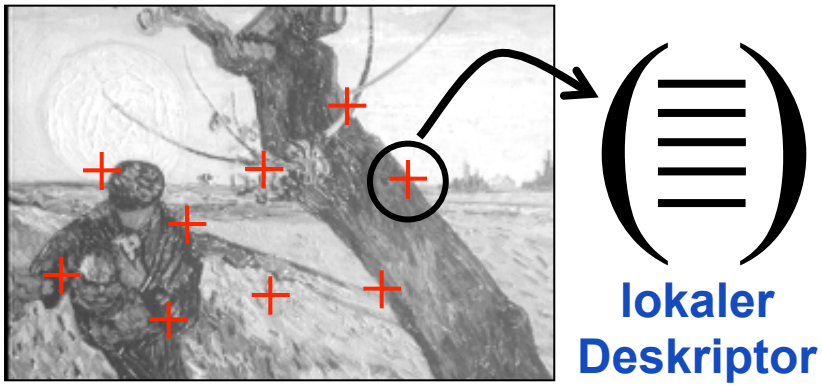
IMAGE RECOGNITION WITH LOCAL FEATURES

Objekt und Bilderkennung

- Identifikation von Objekten, Szenen, Teilbildern
- 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
  - 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.



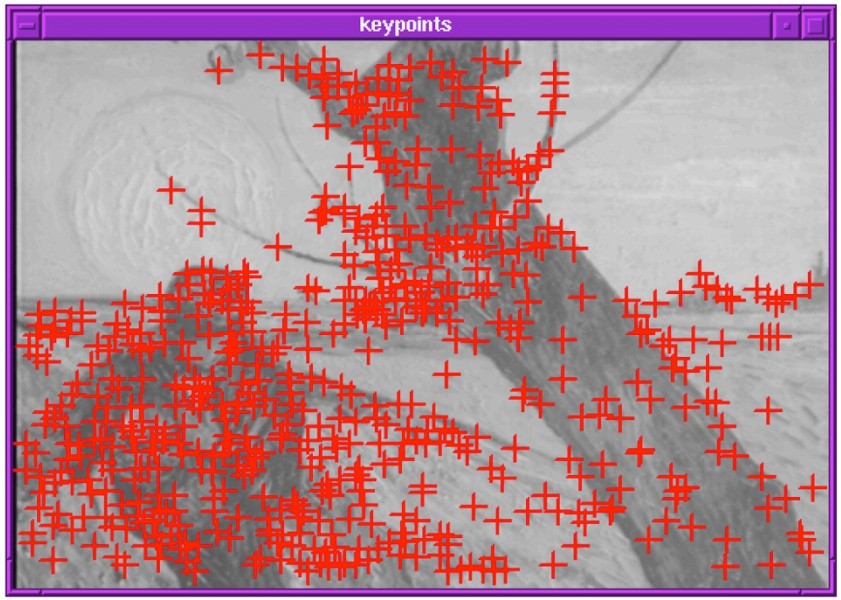
- lokale Deskriptoren, Merkmalsvektoren
  - beschreiben / repräsentieren charakteristische Orte im Bild
  - robust / invariant gegenüber Veränderung der Perspektive, Helligkeit, Translation / Rotation / Skalierung, etc.
  - effizient berechenbar
- Vorteil: teilweise Verdeckung unproblematisch

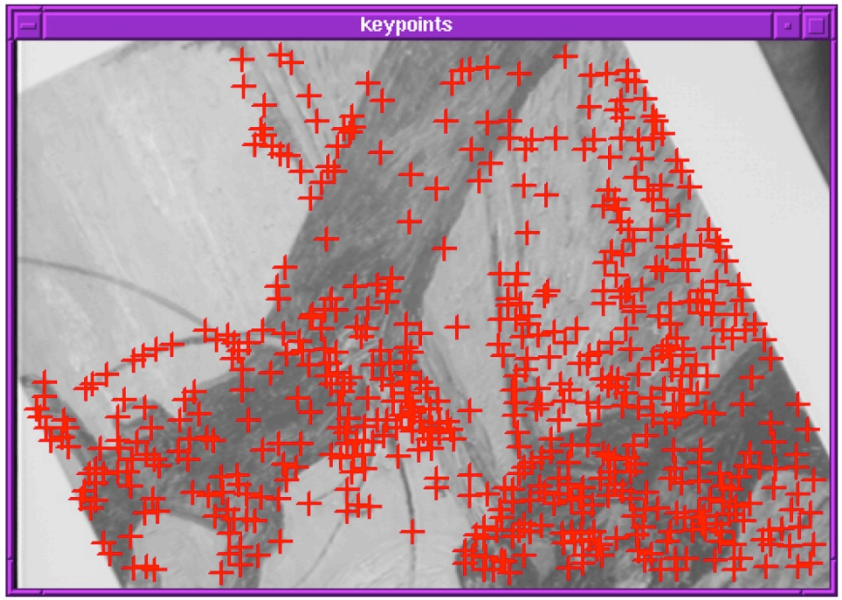
Bildquelle: Schmid, Mohr: Local Grayvalue Invariants for Image Retrieval. PAMI, 19(5):530-534, 1997.

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.

Stitching Panoramic Images

- PhotoSynth, MSR
  - Tools to create and view panoramic images
  - http://photosynth.net
  - Capture multiple images from single location

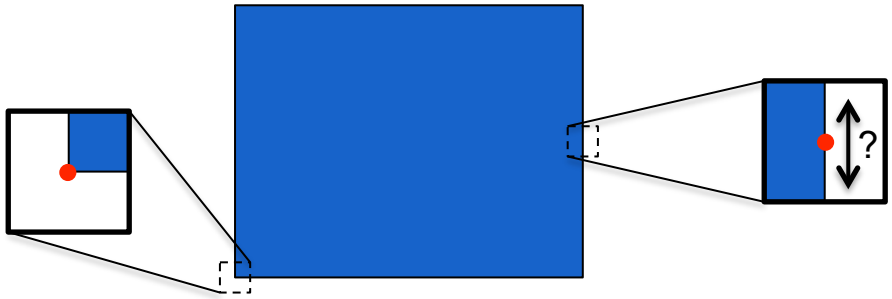
- Alternative: Capture images with camera phone walking around object
  - http://www.technologyreview.com/computing/37021/?a=f

Stitching Panoramic Images

http://www.youtube.com/watch?v=tXCobp1ViS0

Finden lokaler Merkmale

- Viele Bilderkennungsalgorithmen benötigen Merkmale, die eine stabile Position in (x,y) haben
- Kanten sind nur in einer Richtung lokalisiert
  → Ecken in zwei



- Gewünschte Eigenschaften von Merkmalen
  - Genaue Lokalisierbarkeit
  - Invarianz gegenüber Perspektiv- und Helligkeitsänderung
  - Robust gegenüber Rauschen

Slide and illustration adapted from Bernd Girod, Digital Image Processing

Gradienten: Erste Ableitung von Bildern

- Gegeben: 1D-Graustufenbild $f(x)$
- Erste Ableitung: $f_x = f(x+1) - f(x)$
  - Differential durch Differenz approximiert

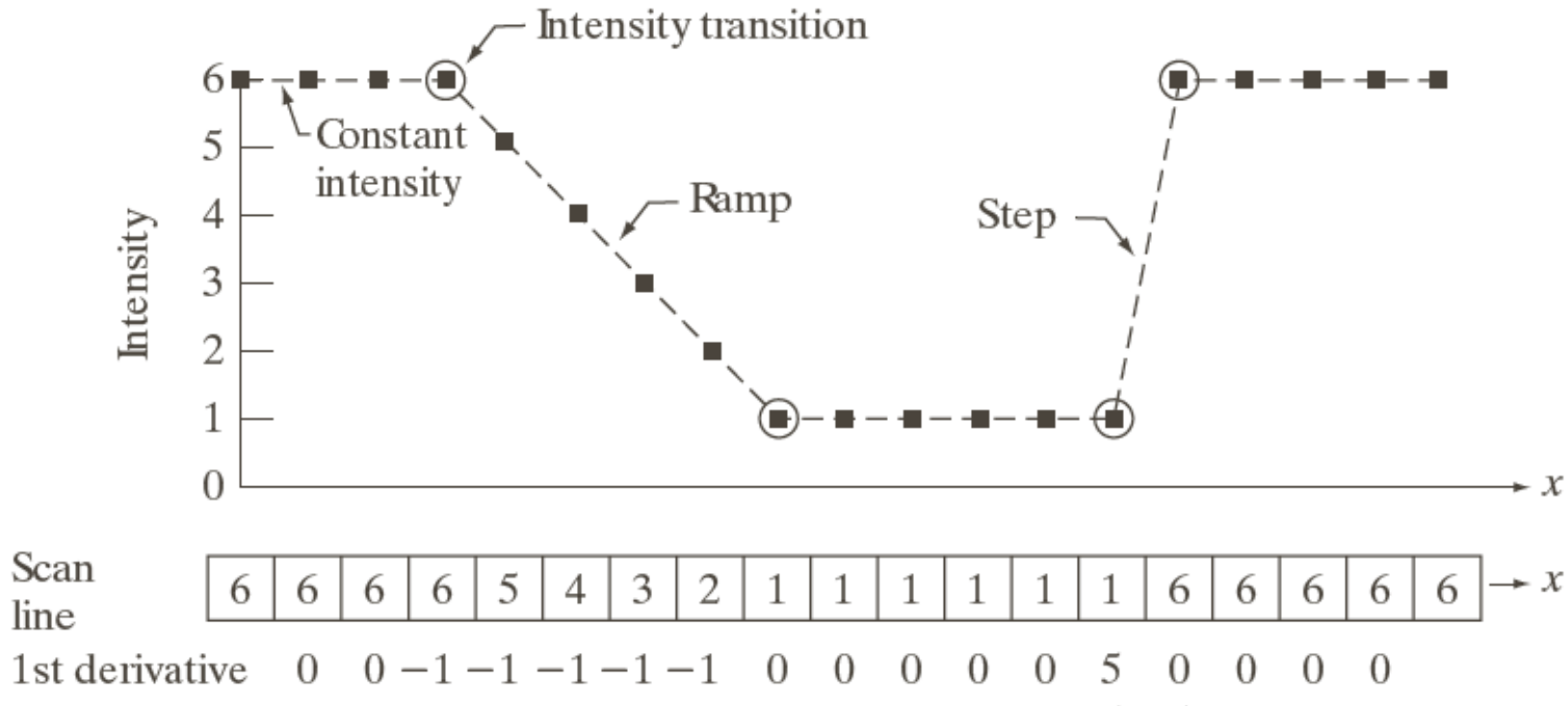
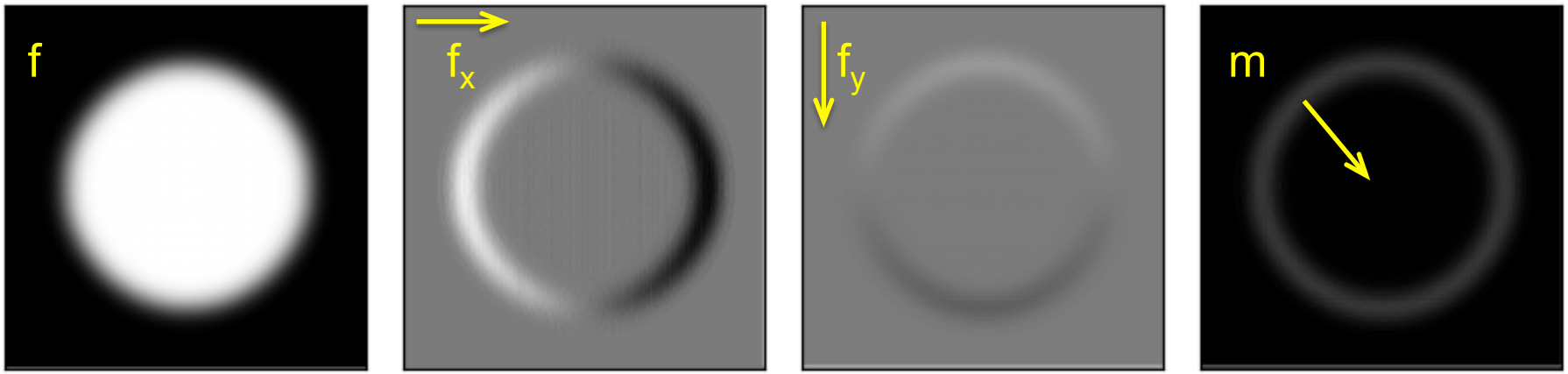


Abbildung: © R. C. Gonzalez & R. E. Woods, Digital Image Processing

Kanten im 2D-Raum: Gradienten

- Bild mit Grauwert f(x,y) an Position (x,y)
- Gradient: Vektor ($f_x(x,y)$, $f_y(x,y)$) der partiellen Ableitungen in (x,y)-Ebene mit
  - Richtung: Richtung der größten Steigung
    $\theta(x,y) = \text{atan}(f_y(x,y) / f_x(x,y))$
  - Länge: Stärke der größten Steigung
    $m(x,y) = \text{sqrt}(f_x(x,y)^2 + f_y(x,y)^2)$

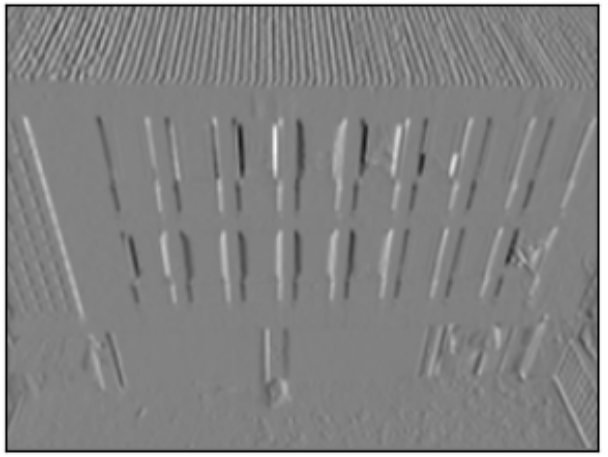


Gradienten finden: Konvolution mit dem Sobel-Operator

$*$

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

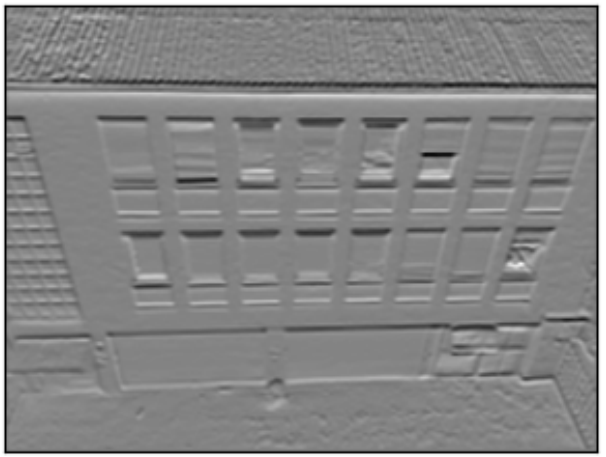
$=$



$*$

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

$=$



Filterung im Ortsraum

- Lineare Filterung
- m×n Filtermaske
- Lokale Umgebung
- Vorgegebene Operation auf Pixeln in lokaler Umgebung
- Skalarprodukt
  f(x-1,y-1)*w(-1,-1) +
  ... + f(x,y)*w(0,0) +
  ... + f(x+1,y+1)* w(1,1)

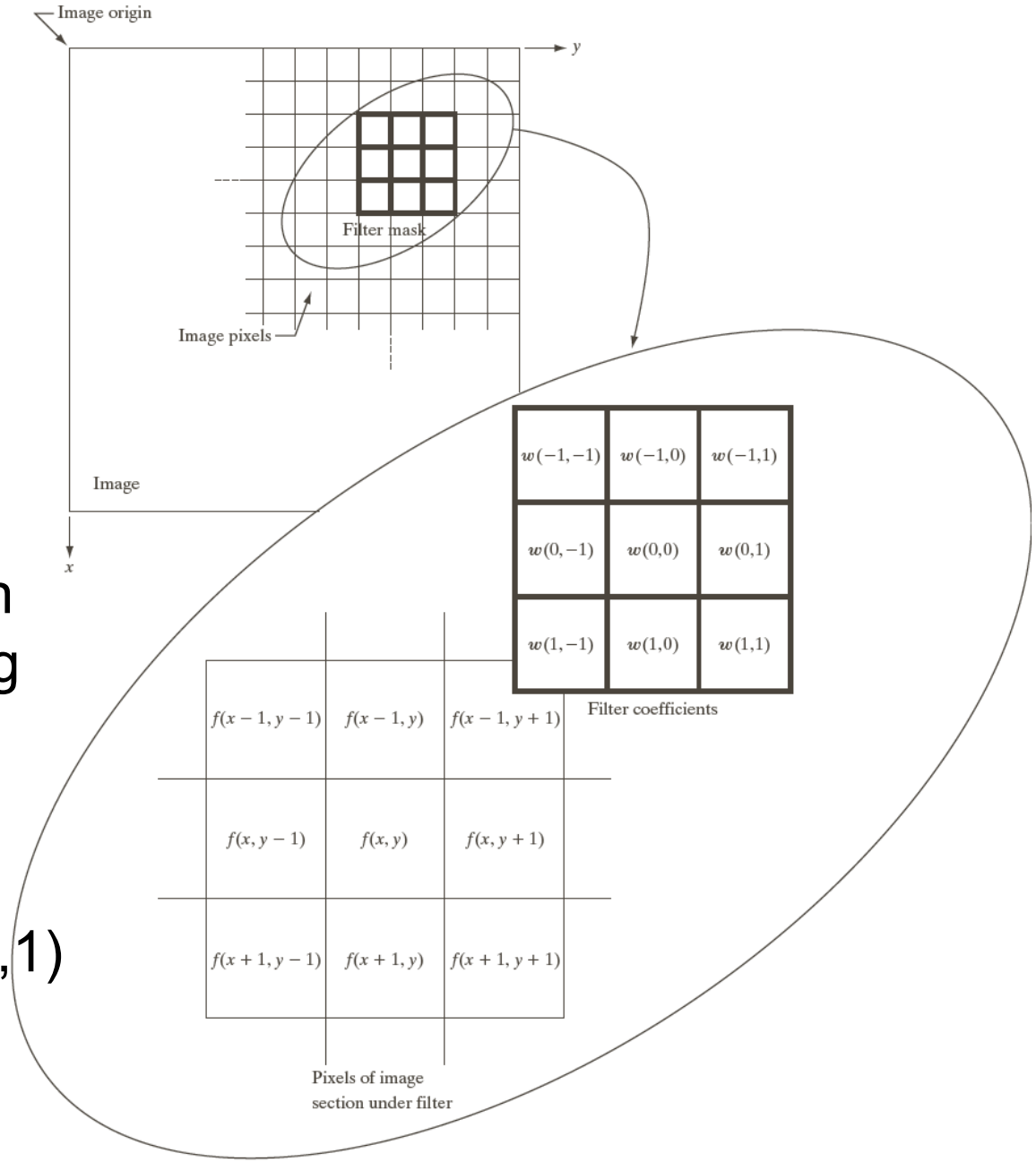
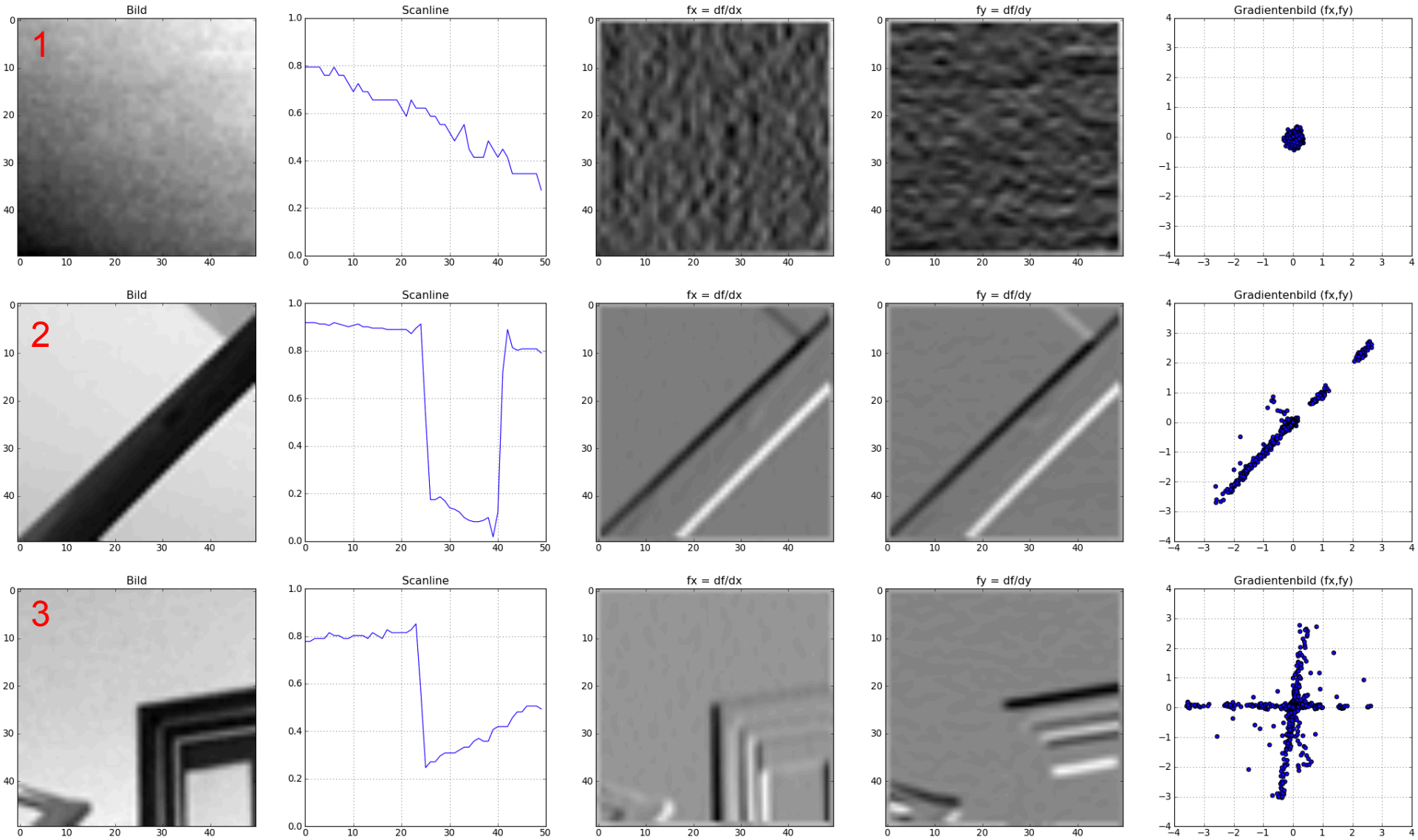


Abbildung: © R. C. Gonzalez & R. E. Woods, Digital Image Processing

Verteilung der Gradienten in Bild

Verteilung der Gradienten in Bild



Verschiebung

- Effekt einer Verschiebung um (kleine) Δx, Δy
  - Flache Region: keine Änderung im Erscheinungsbild
  - Kante: keine Änderung bei Verschiebung entlang der Kante
  - Ecke: große Änderung in jeder Richtung
- Intensitätsänderung bei Verschiebung um Δx, Δy

$$s(\Delta x, \Delta y) = \sum_{(x,y) \in window} \left[f(x,y) - f(x+\Delta x, y+\Delta y) \right]^2$$

- Lineare Approximation für (kleine) Δx, Δy

$$f(x+\Delta x, y+\Delta y) \approx f(x,y) + f_x(x,y)\Delta x + f_y(x,y)\Delta y$$

  - f_x, f_y sind Gradienten in x- bzw. y-Richtung

Verschiebung

- Lineare Approximation für (kleine) Δx, Δy

$$s(\Delta x, \Delta y)$$

$$\approx \begin{pmatrix} \Delta x & \Delta y \end{pmatrix} \begin{pmatrix} \sum_{(x,y)\in window} f_x^2 & \sum_{(x,y)\in window} f_x f_y \\ \sum_{(x,y)\in window} f_x f_y & \sum_{(x,y)\in window} f_y^2 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

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

- M: Kovarianzmatrix, „autocorrelation matrix“

Animation: http://www.aiaccess.net/English/Glossaries/GlosMod/e_gm_covariance_matrix.htm#Animation_covariance%20matrix

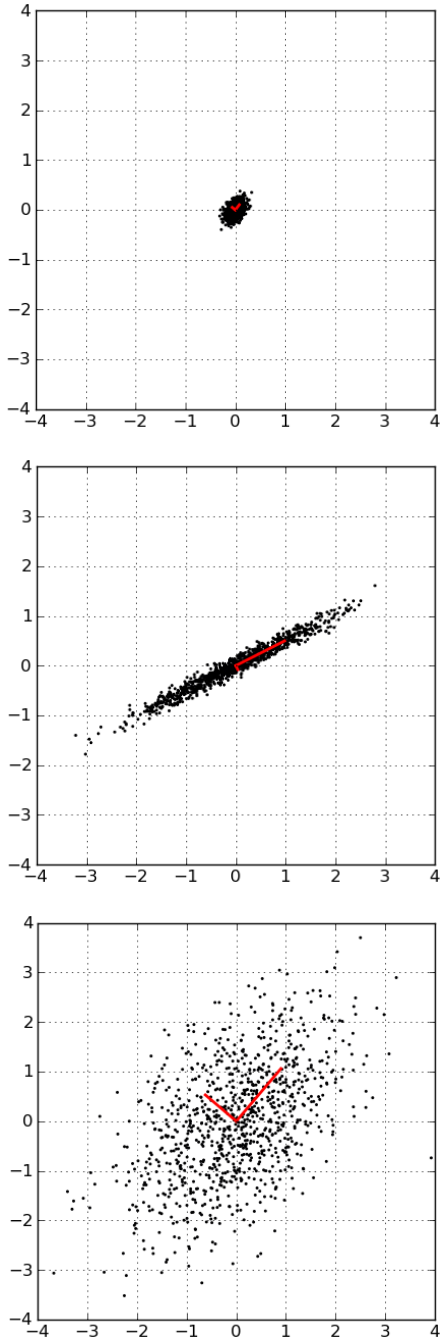
Eigenwerte der Kovarianzmatrix

- Eigenvektoren v_1, v_2; Eigenwerte λ_1, λ_2
  - Richtung: Hauptachsen der Verteilung
  - Länge: Varianz entlang der Hauptachsen
- Keine dominante Orientierung
  - $\lambda_1 = 0.017$
  - $\lambda_2 = 0.006$

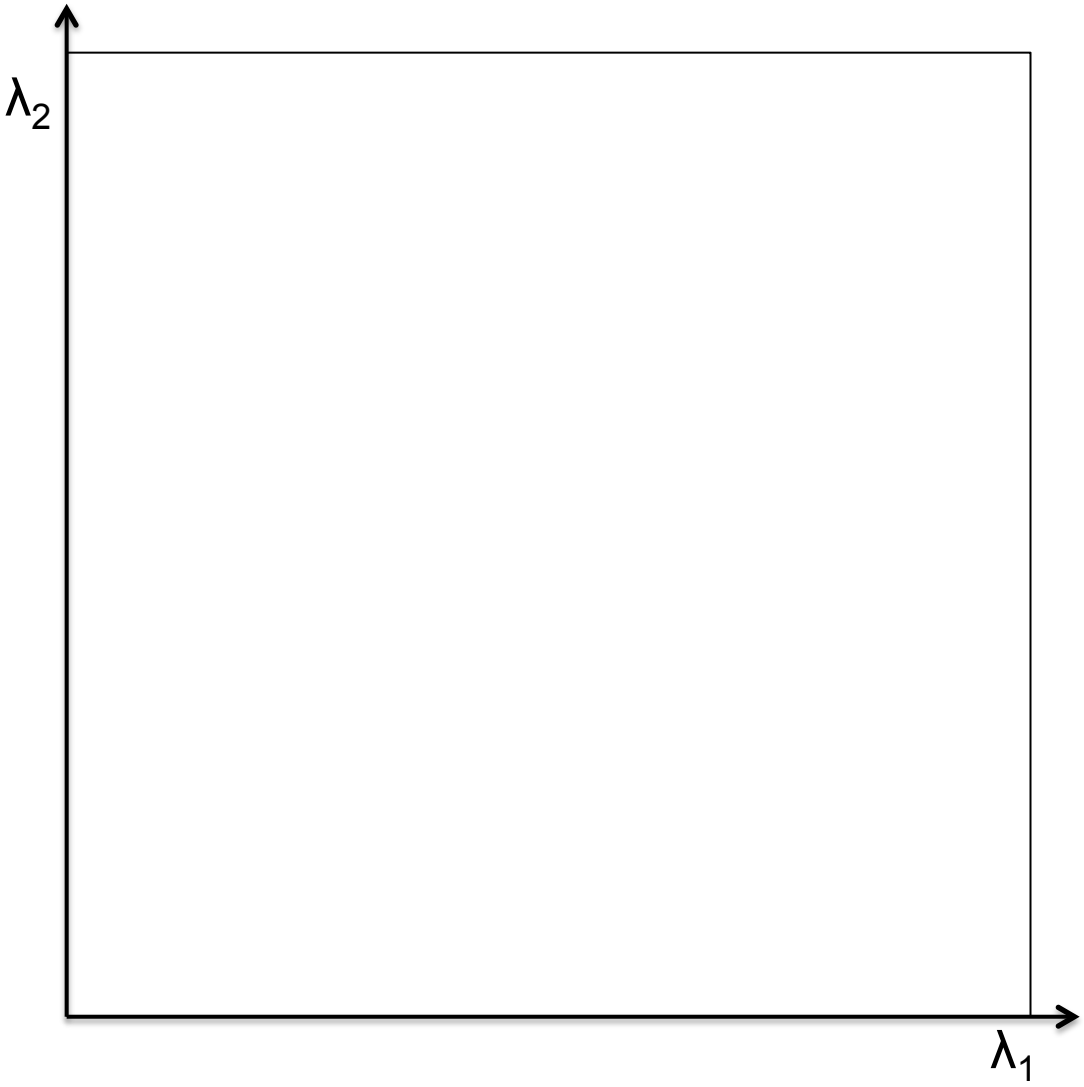
  → λ_1 klein, λ_2 klein
- Nur eine dominante Orientierung
  - $\lambda_1 = 1.313$
  - $\lambda_2 = 0.008$

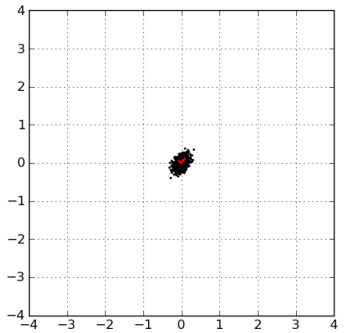
  → λ_1 groß, λ_2 klein
- Mehrere dominante Orientierungen
  - $\lambda_1 = 1.936$
  - $\lambda_2 = 0.669$

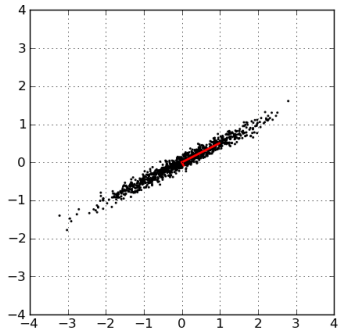
  → λ_1 groß, λ_2 groß

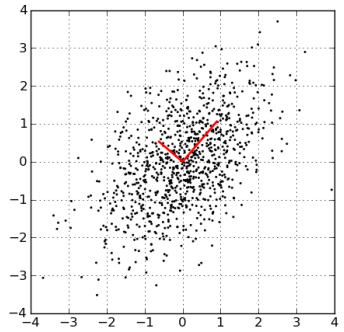


Klassifikation über Eigenvektoren

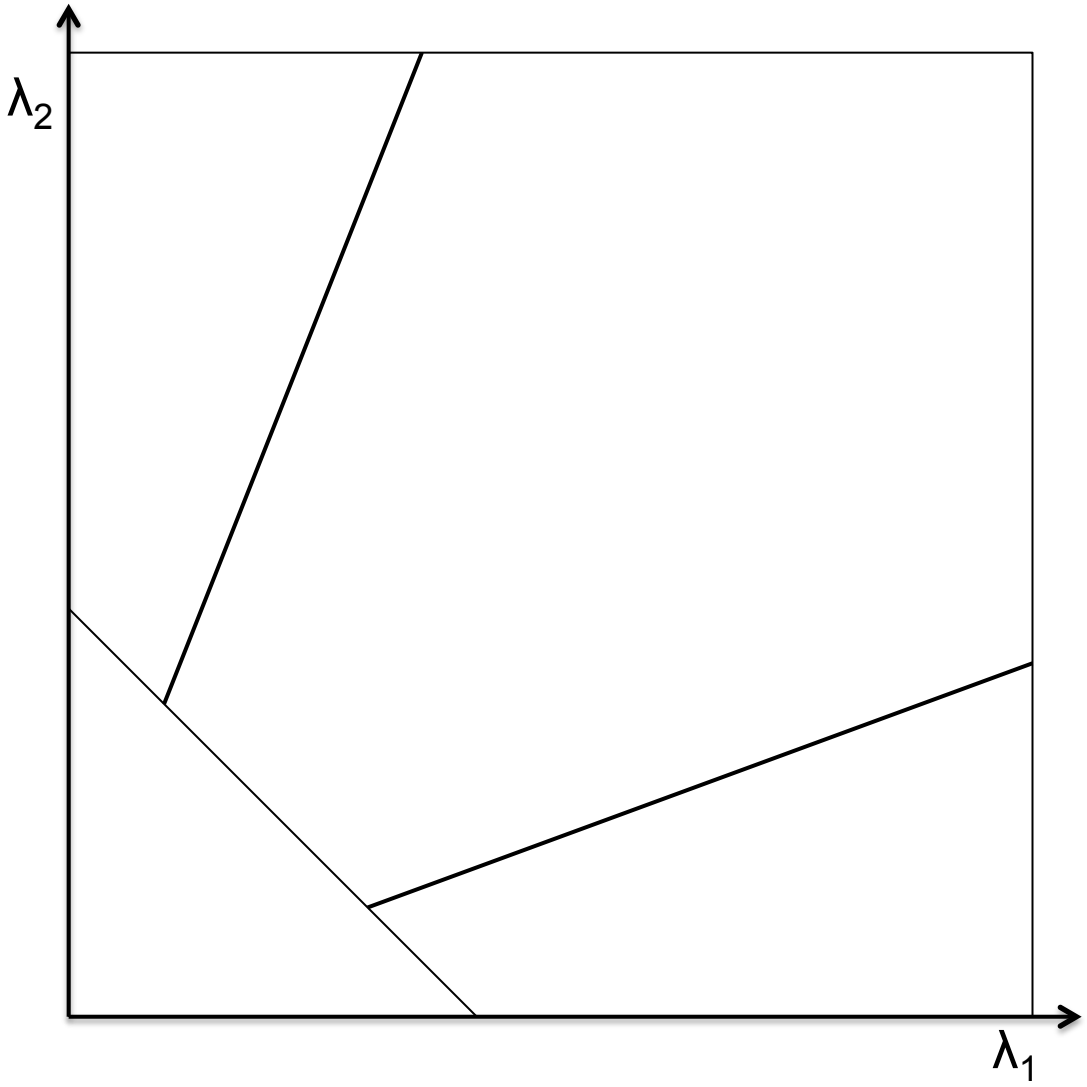


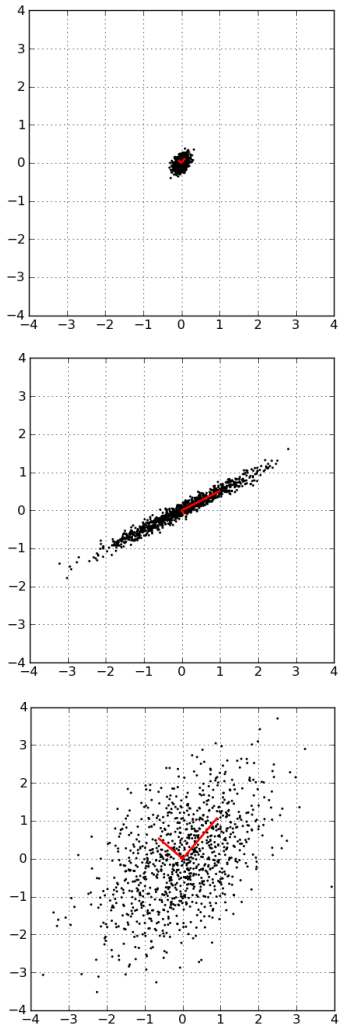




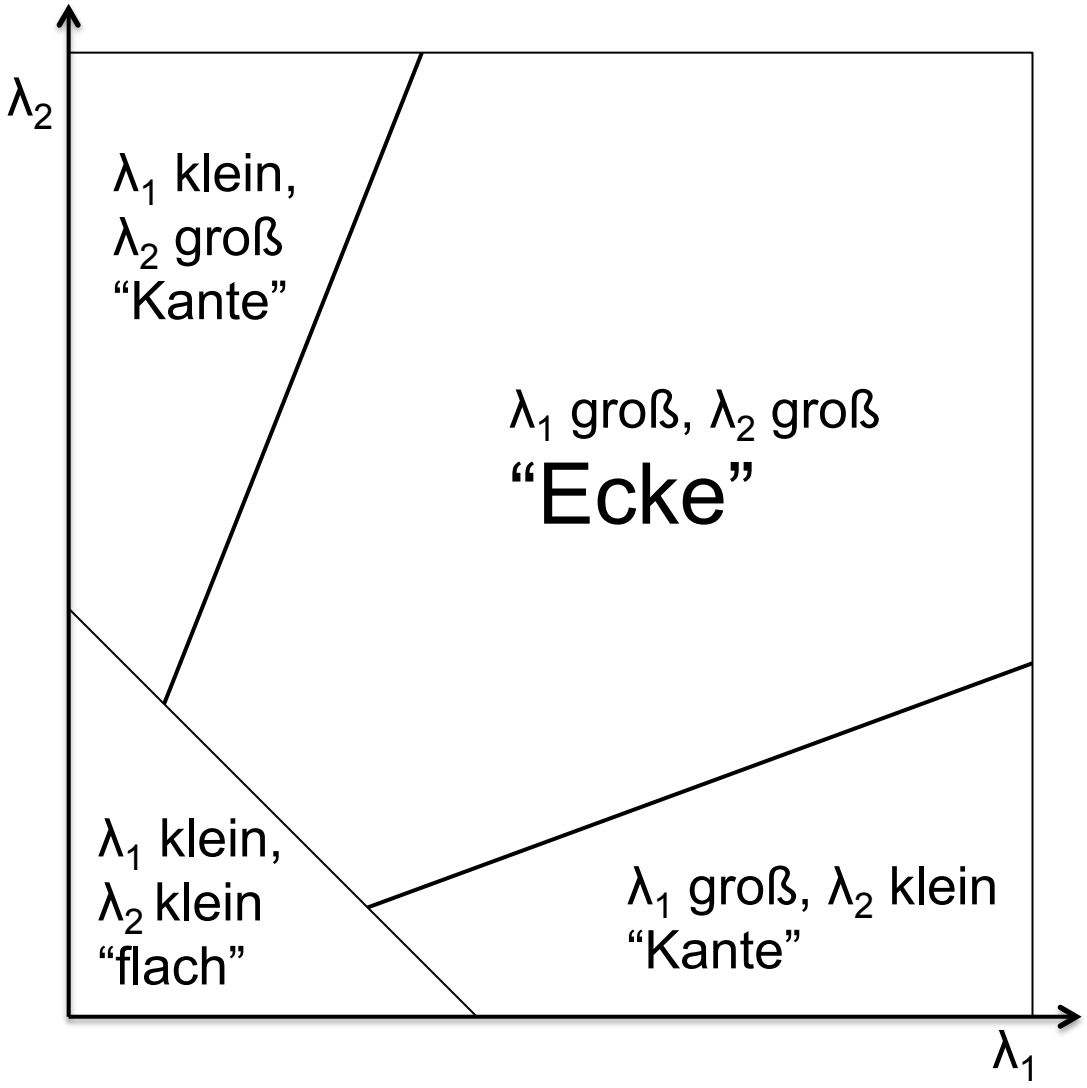


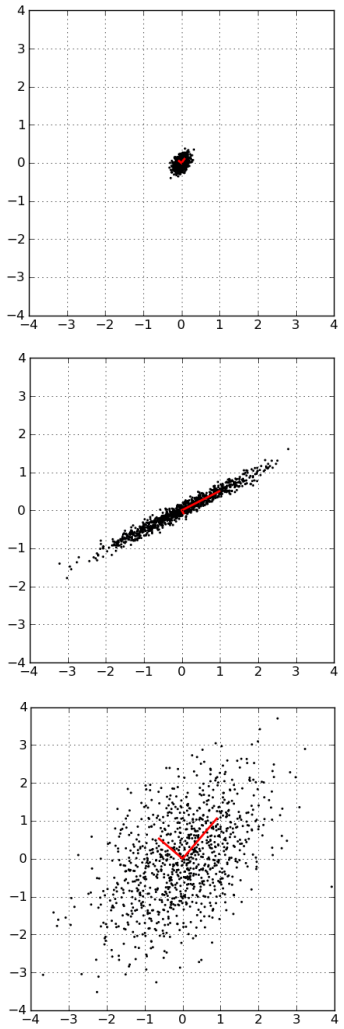
Klassifikation über Eigenvektoren





Klassifikation über Eigenvektoren





Harris Corner Detektor (Harris, Stephens, 1988)

- gewichtete Kovarianzmatrix, „autocorrelation matrix“

$$M = \sum_{(x,y)\in window} w(x,y) * \begin{pmatrix} f_x^2(x,y) & f_x(x,y)f_y(x,y) \\ f_x(x,y)f_y(x,y) & f_y^2(x,y) \end{pmatrix}$$

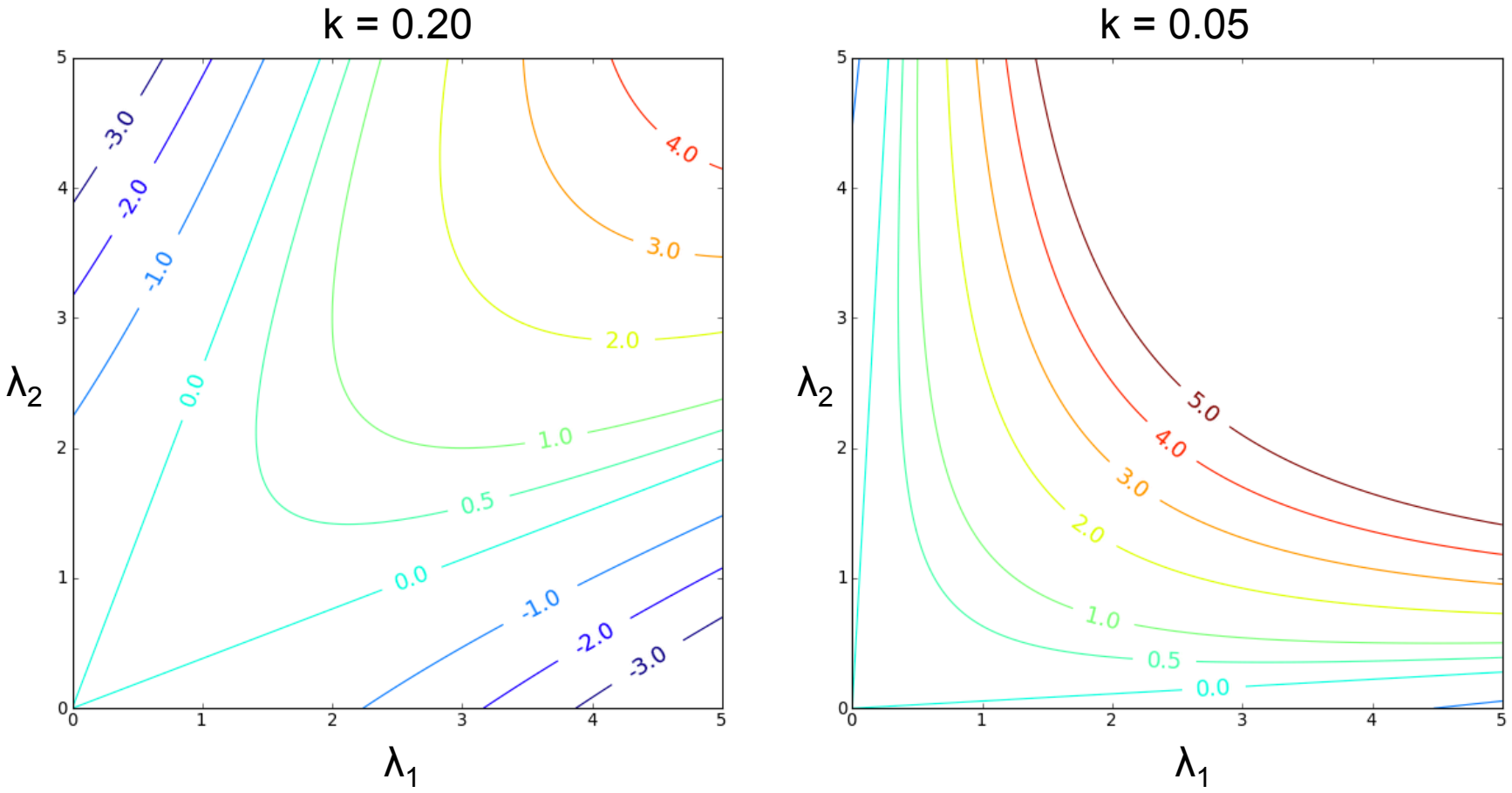
w(x,y): Gewichtungsfunktion, z.B. Gauß-Funktion

- Maß für die “Stärke” der Ecke

$$C(x,y) = \det(M) - k(trace(M))^2 = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2$$

k = 0.04..0.06

Harris Corner Detektor: Parameter “k”



- $C = \det(M) - k\ \mathrm{trace}(M)^2 = \lambda_1 \lambda_2 - k\ (\lambda_1 + \lambda_2)$

Harris Corner Detection – Algorithmus

- Berechne die Ableitungen f_x und f_y in x- und y-Richtung
  - z.B. per Konvolution mit dem Sobel-Operator
- Berechne die elementweisen Produkte der Ableitungen

  $f_x^2 = f_x f_x, \quad f_y^2 = f_y f_y, \quad f_{xy} = f_x f_y$
- Berechne $f_{xxSum} = w * f_x^2, \quad f_{yySum} = w * f_y^2, \quad f_{xySum} = w * f_{xy}$, wobei * der Konvolutionsoperator ist und (z.B.)

$$w = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

- Definiere für jedes Pixel (x,y) die Matrix M(x,y)

$$M(x,y) = \begin{pmatrix} f_{xxSum}(x,y) & f_{xySum}(x,y) \\ f_{xySum}(x,y) & f_{yySum}(x,y) \end{pmatrix}$$

Harris Corner Detection – Algorithmus

- Berechne für jedes Pixel das Maß der Eckenstärke

$$C(x,y) = \det(M) - k(trace(M))^2 = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2$$

  k = 0.04..0.06

- Wende eine untere Schranke und non-maximum Unterdrückung an
  - z.B. im Radius 2 um den jeweils betrachteten Punkt

Beispiele

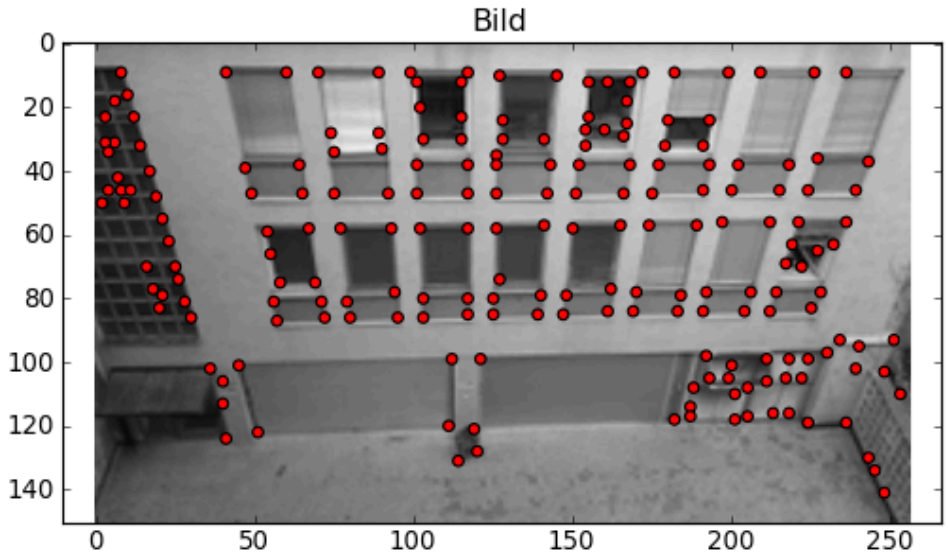


Bild
0
20
40
60
80
100
120
140
0
50
100
150
200
250

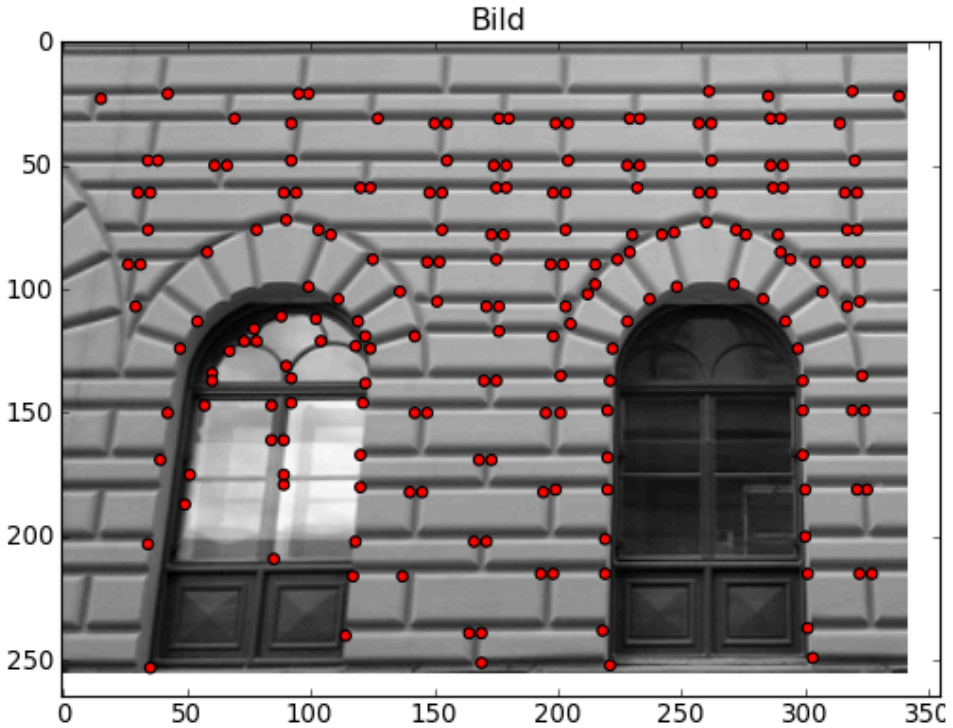
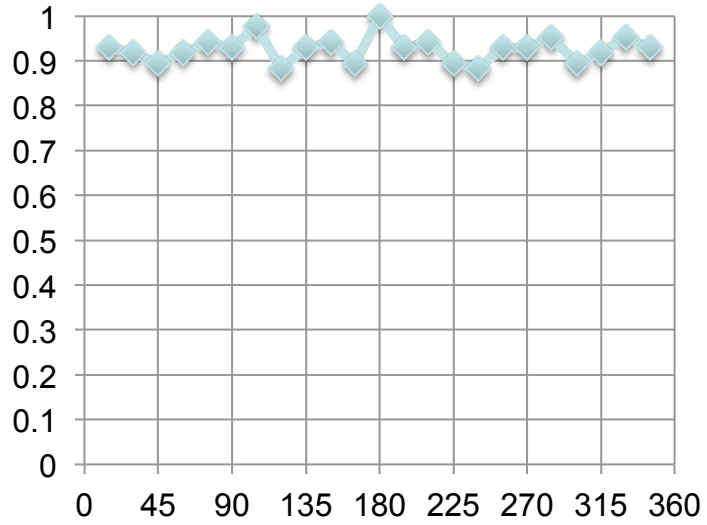
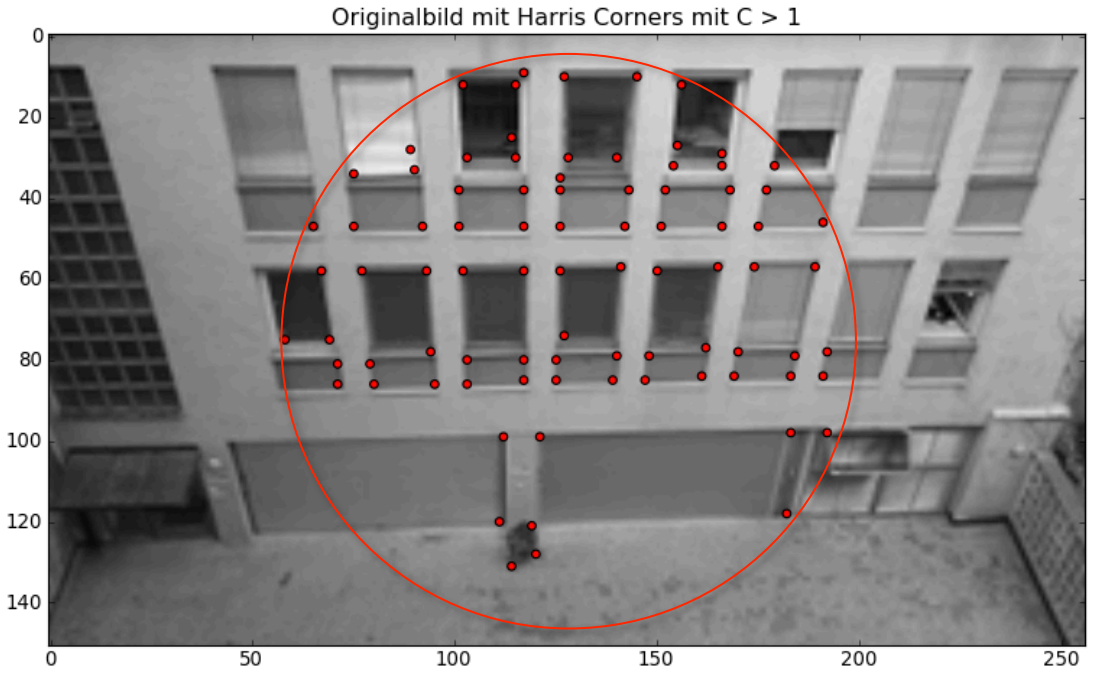


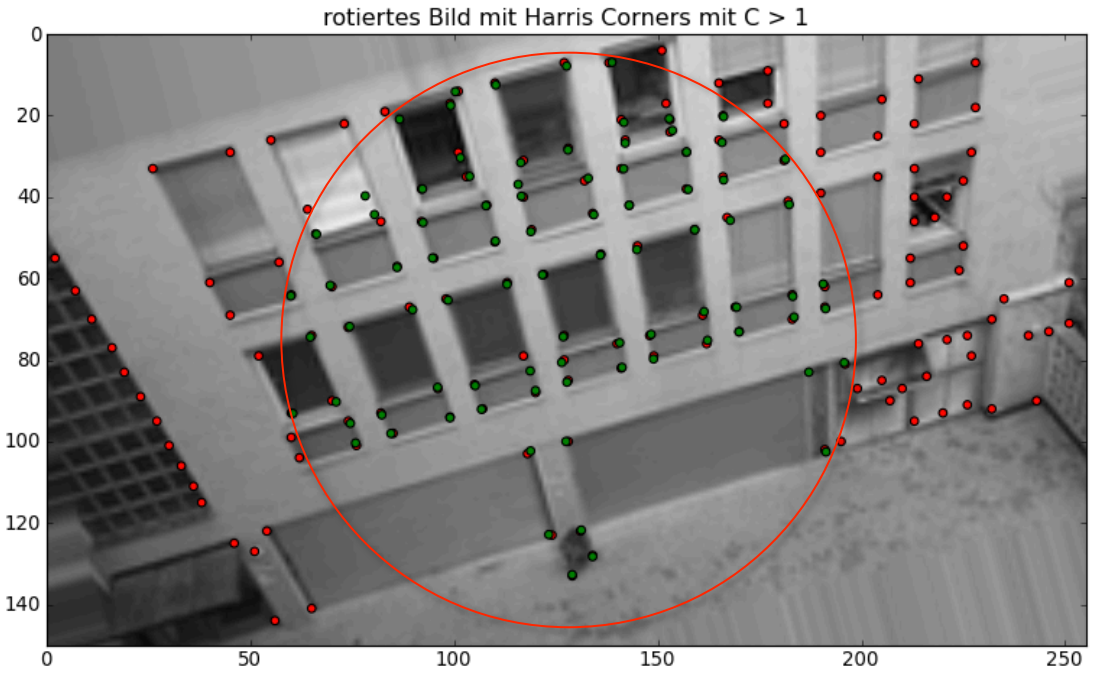
Bild
0
50
100
150
200
250
0
50
100
150
200
250
300
350

Repeatability

- Robustheit gegenüber Rotation, Skalierung, Änderung der Perspektive
- hier Rotation:

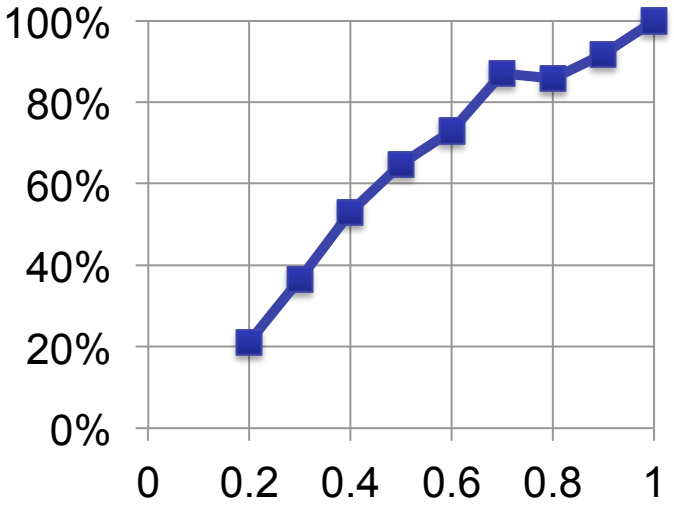


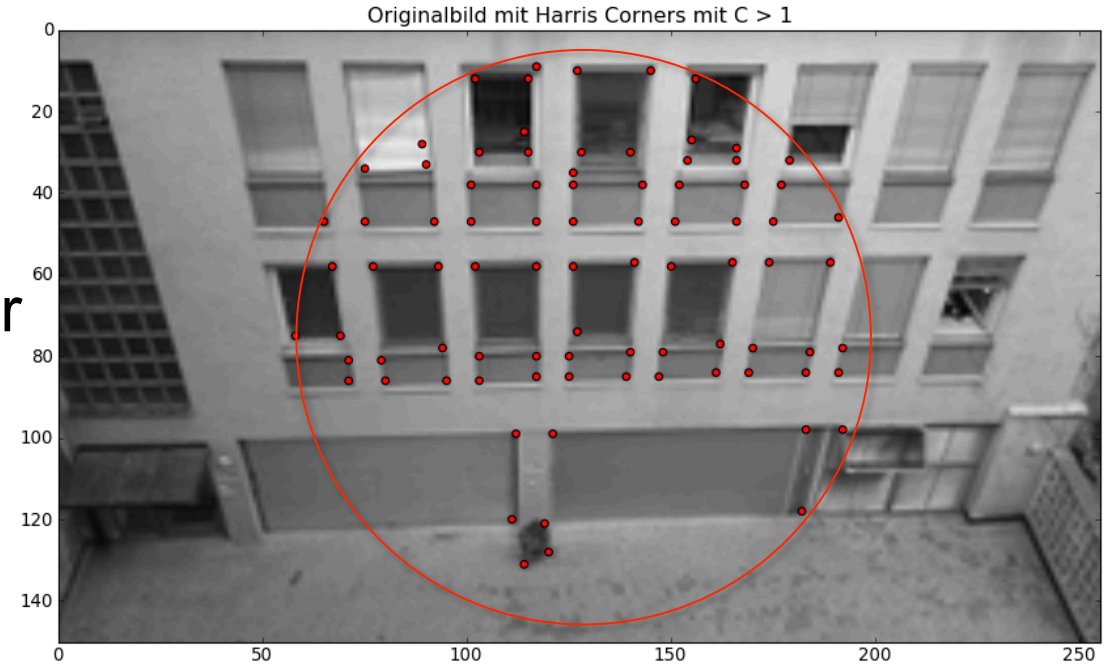


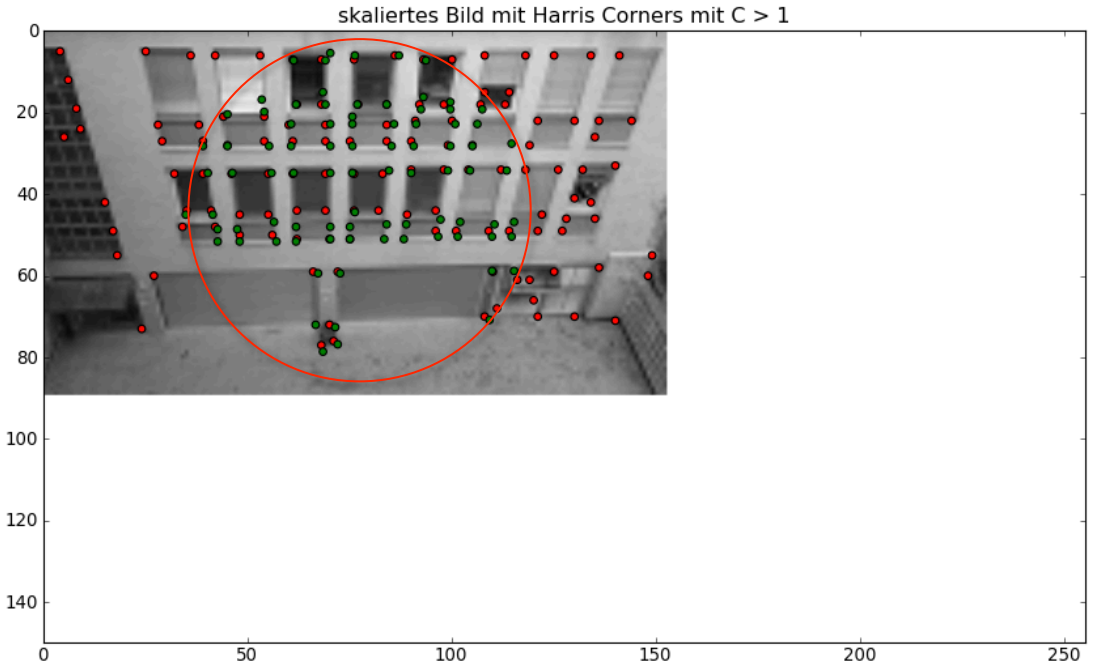


Repeatability

- Robustheit gegenüber Rotation, Skalierung, Änderung der Perspektive
- hier Skalierung:

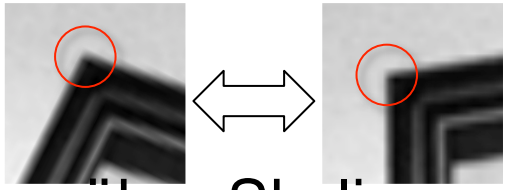




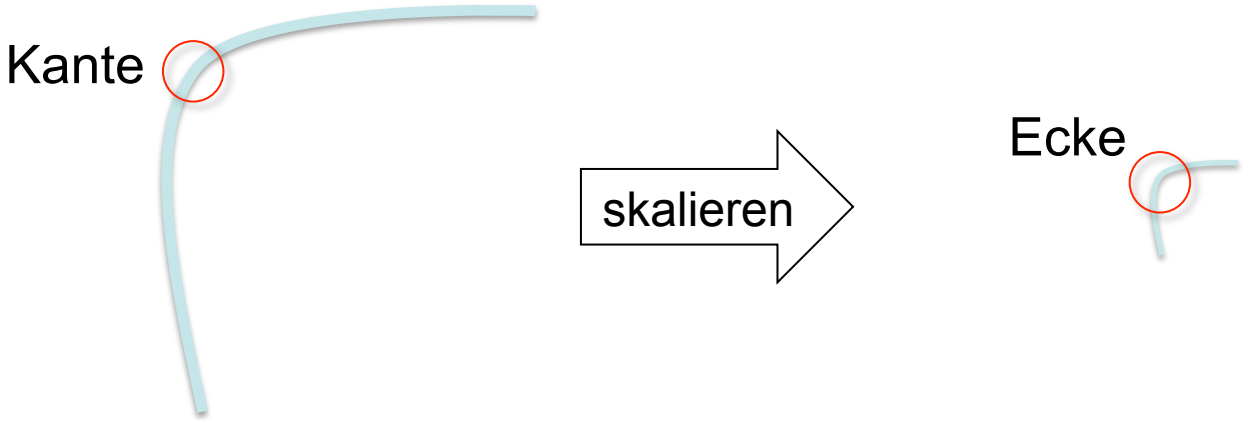


Robustheit des Harris Corner Detektors

- Invariant gegenüber Helligkeitsänderungen
- Invariant gegenüber Translation und Rotation



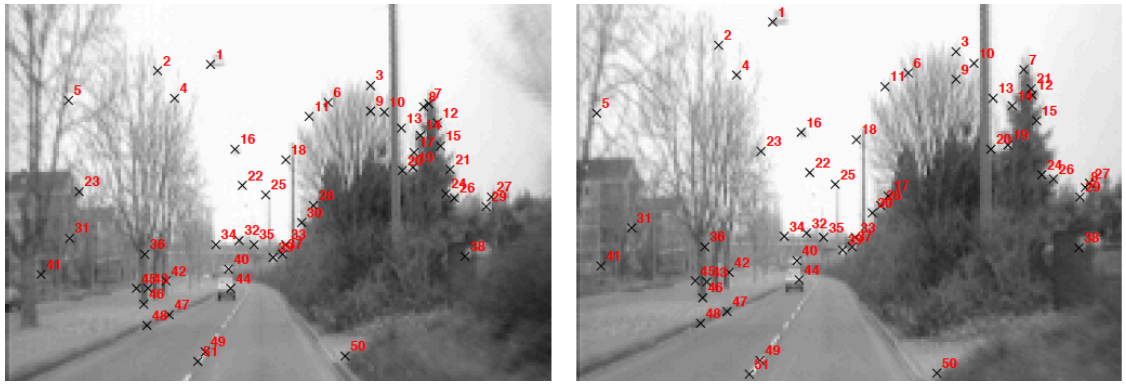
- Nicht invariant gegenüber Skalierung



Slide and illustration adapted from Bern Girod, Digital Image Processing

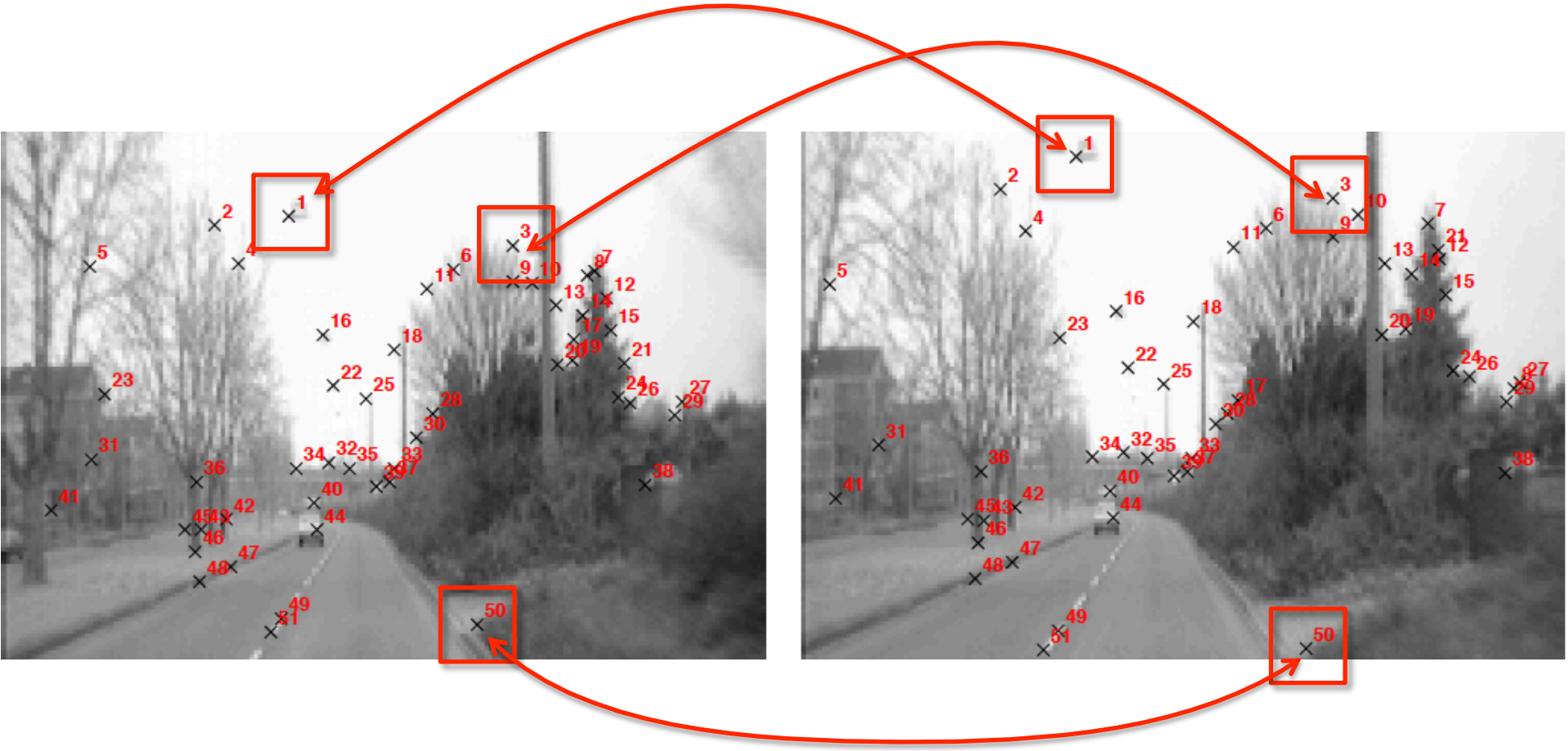
Zhang et al. 1994: Matching Corners

- 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?

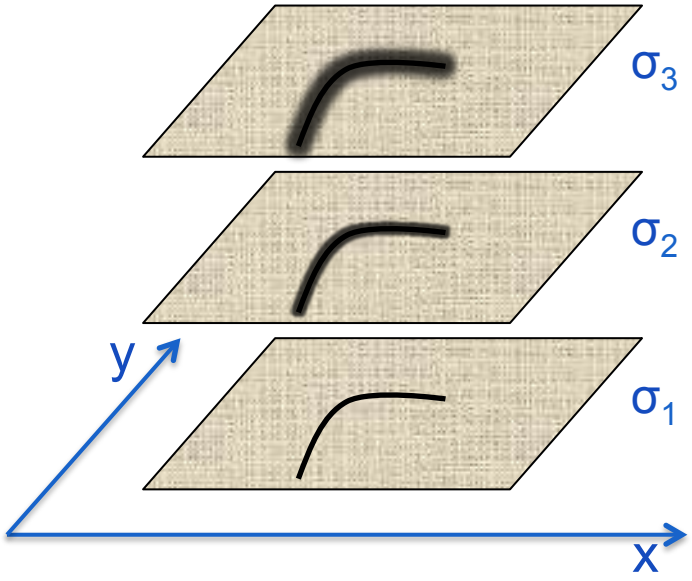
Bildquelle: Zhang, Deriche, Faugeras, Luong: A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry. Technical Report, INRIA, 1994.

SKALENRAUM

Multiskalen-Repräsentation eines Bildes

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

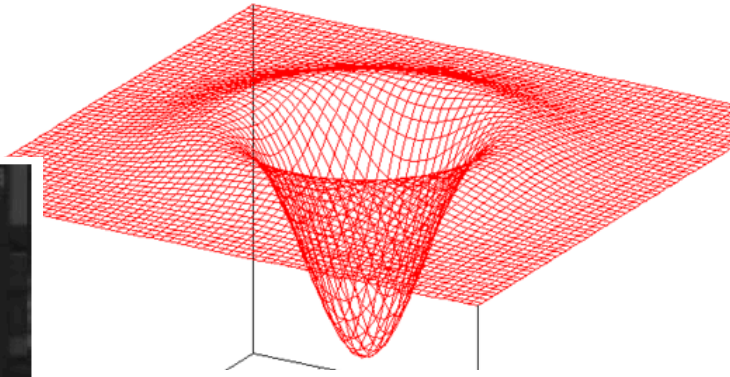
Bild (Multiskalen-Repr.):



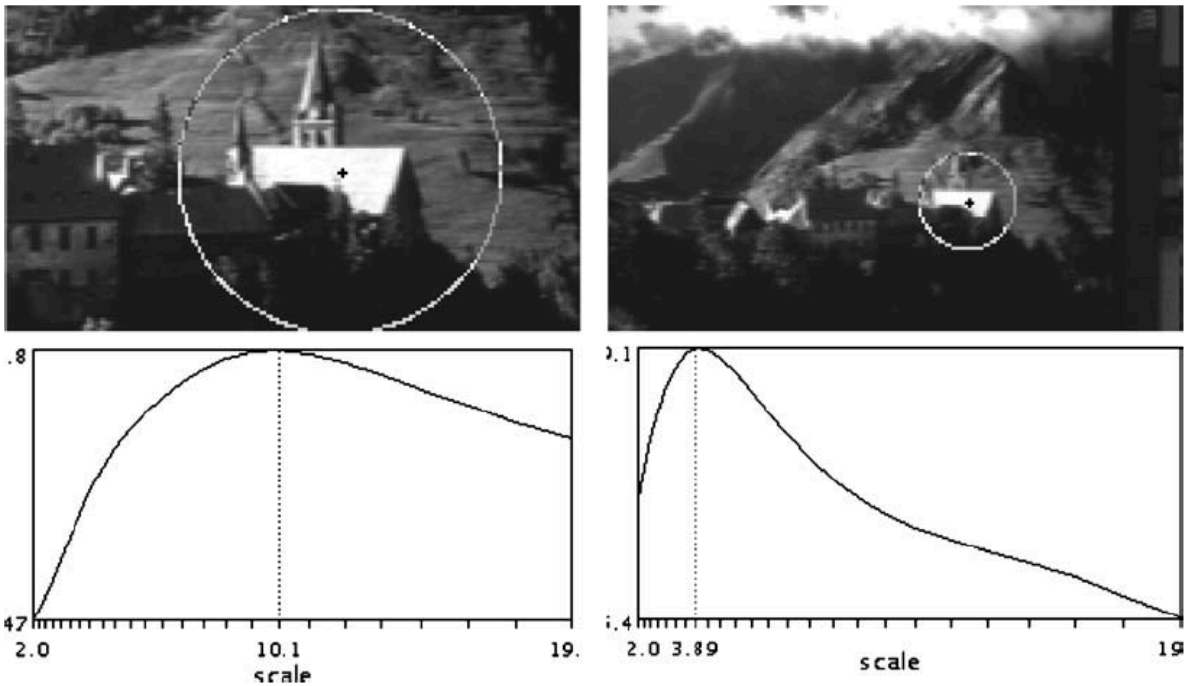
Charakteristische Skalierungsstufe

- Charakteristische Skalierung eines Merkmals ist lokales Extremum des Laplacian of Gaussian (LoG) Operators

LoG-Operator:



"blob detector"



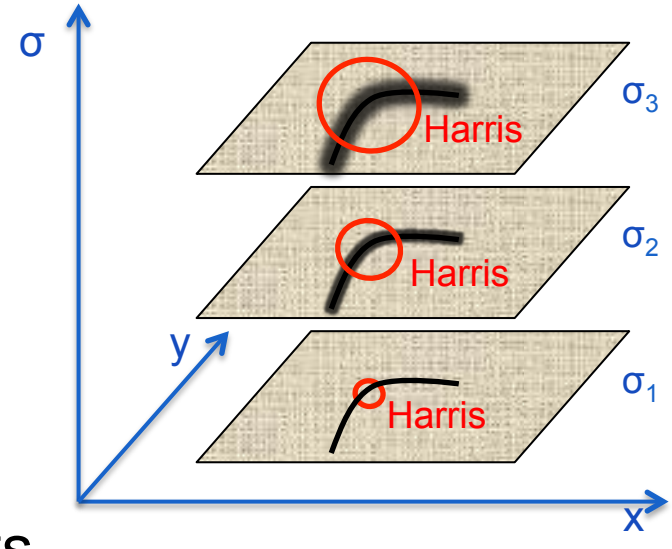
charakteristische Skalierung:
σ = 10.1 (links)
σ = 3.90 (rechts)

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

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



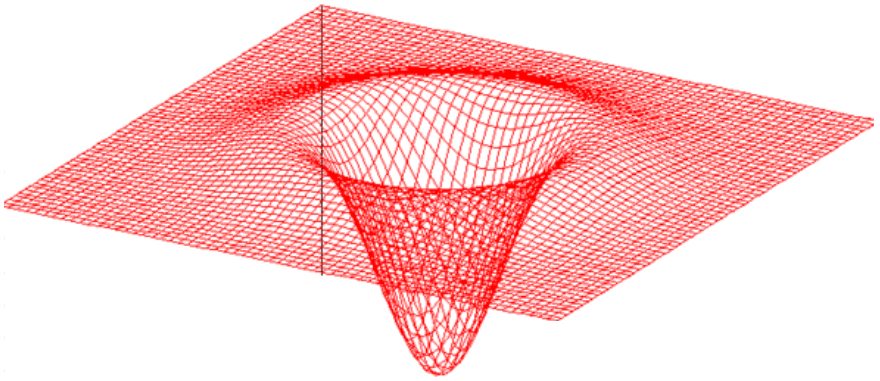
- Berechnung der zweiten Ableitung (LoG-Operator) an den interest points
- Auswahl von Harris-Corners, die lokales Maximum des LoG entlang der Skalierungs-Achse aufweisen

  $L(x,y,\sigma_i) > t_l \wedge L(x,y,\sigma_i) > L(x,y,\sigma_{i-1}) \wedge L(x,y,\sigma_i) > L(x,y,\sigma_{i+1})$
- Normalisierung bzgl. Skalierung notwendig
- Präzisere Variante durch Iteration über Ort und Skalierung

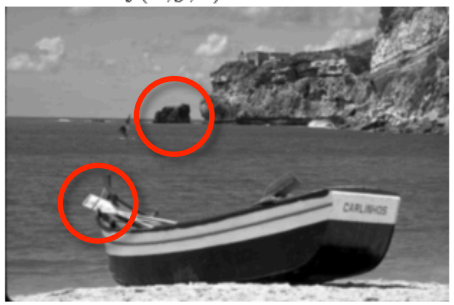
LoG-Operator

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

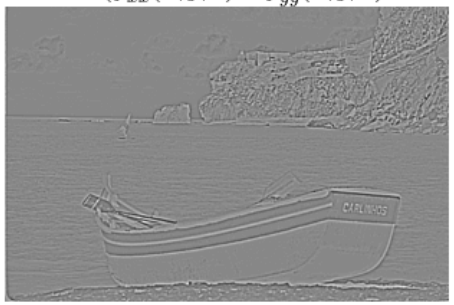
LoG-Operator:



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

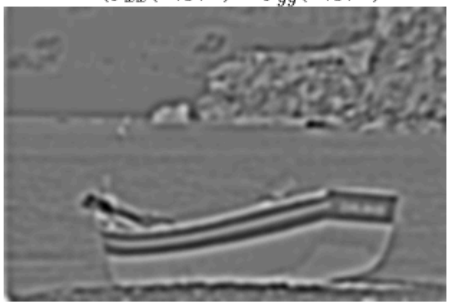


$\sigma^2\ (f_{xx}(x,y,\sigma) + f_{yy}(x,y,\sigma)$



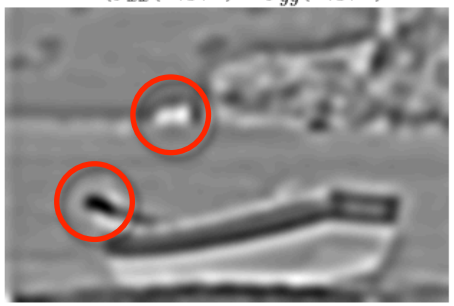
$f_\sigma\,(x,y),\ \sigma = 4.0$

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

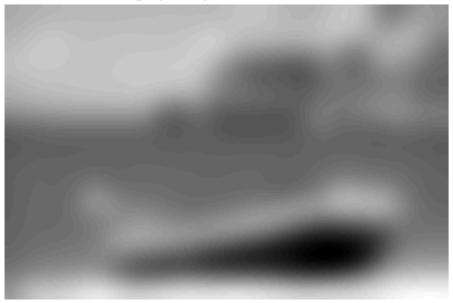


$f_\sigma\,(x,y),\ \sigma = 8.0$

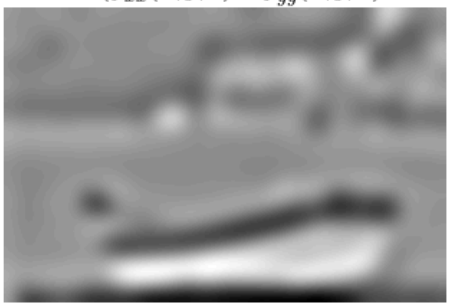
$\sigma^2\ (f_{xx}(x,y,\sigma) + f_{yy}(x,y,\sigma)$



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



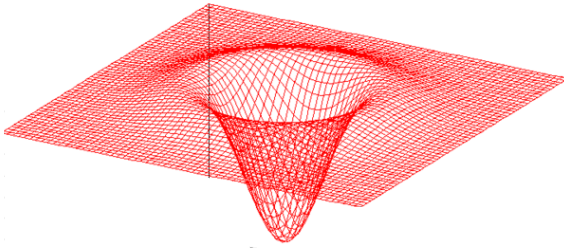
$\sigma^2\ (f_{xx}(x,y,\sigma) + f_{yy}(x,y,\sigma)$



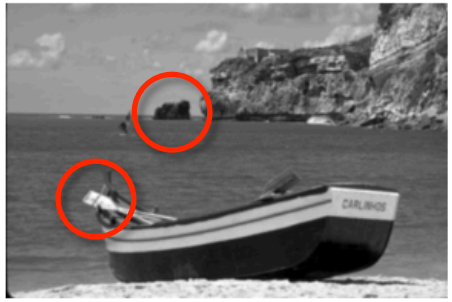
Skalenraum: Betrag des LoG-Operators

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

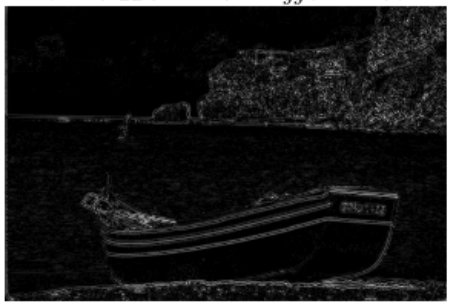
LoG-Operator:



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

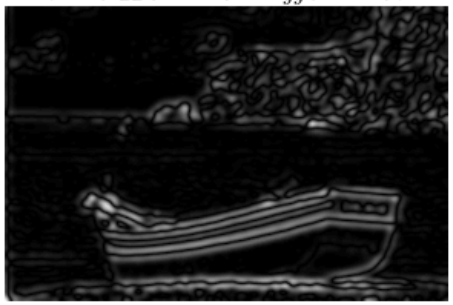


$|\sigma^2\ (f_{xx}(x,y,\sigma)+f_{yy}(x,y,\sigma)|$

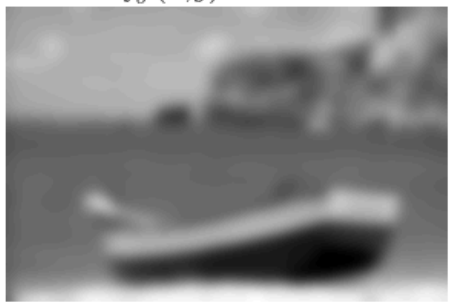


$f_\sigma\,(x,y),\ \sigma=4.0$

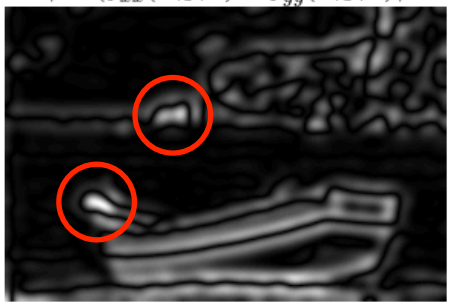
$|\sigma^2\ (f_{xx}(x,y,\sigma)+f_{yy}(x,y,\sigma)|$



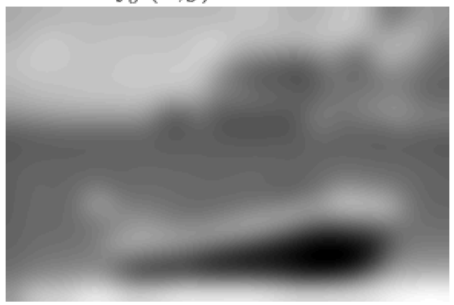
$f_\sigma\,(x,y),\ \sigma=8.0$



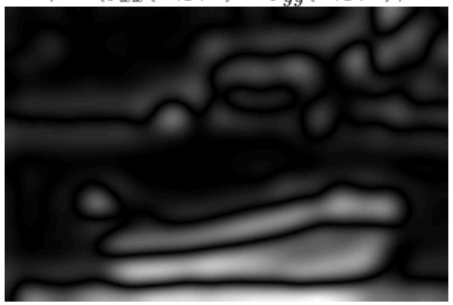
$|\sigma^2\ (f_{xx}(x,y,\sigma)+f_{yy}(x,y,\sigma)|$



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

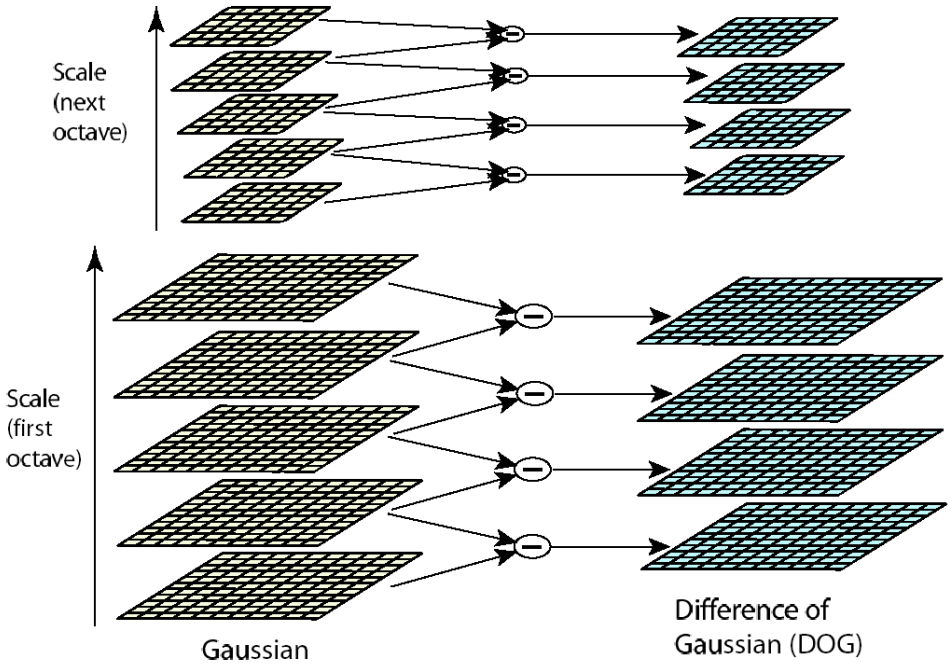


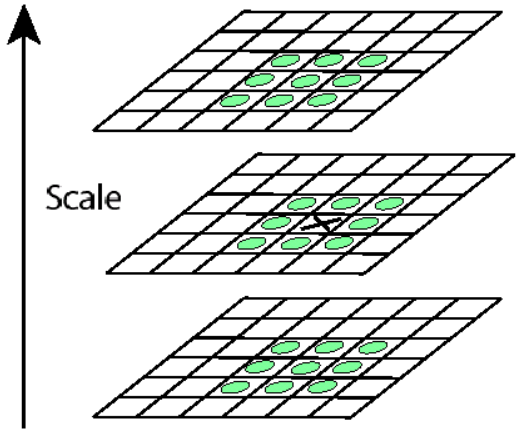
$|\sigma^2\ (f_{xx}(x,y,\sigma)+f_{yy}(x,y,\sigma)|$



David Lowe: SIFT Interest Points

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



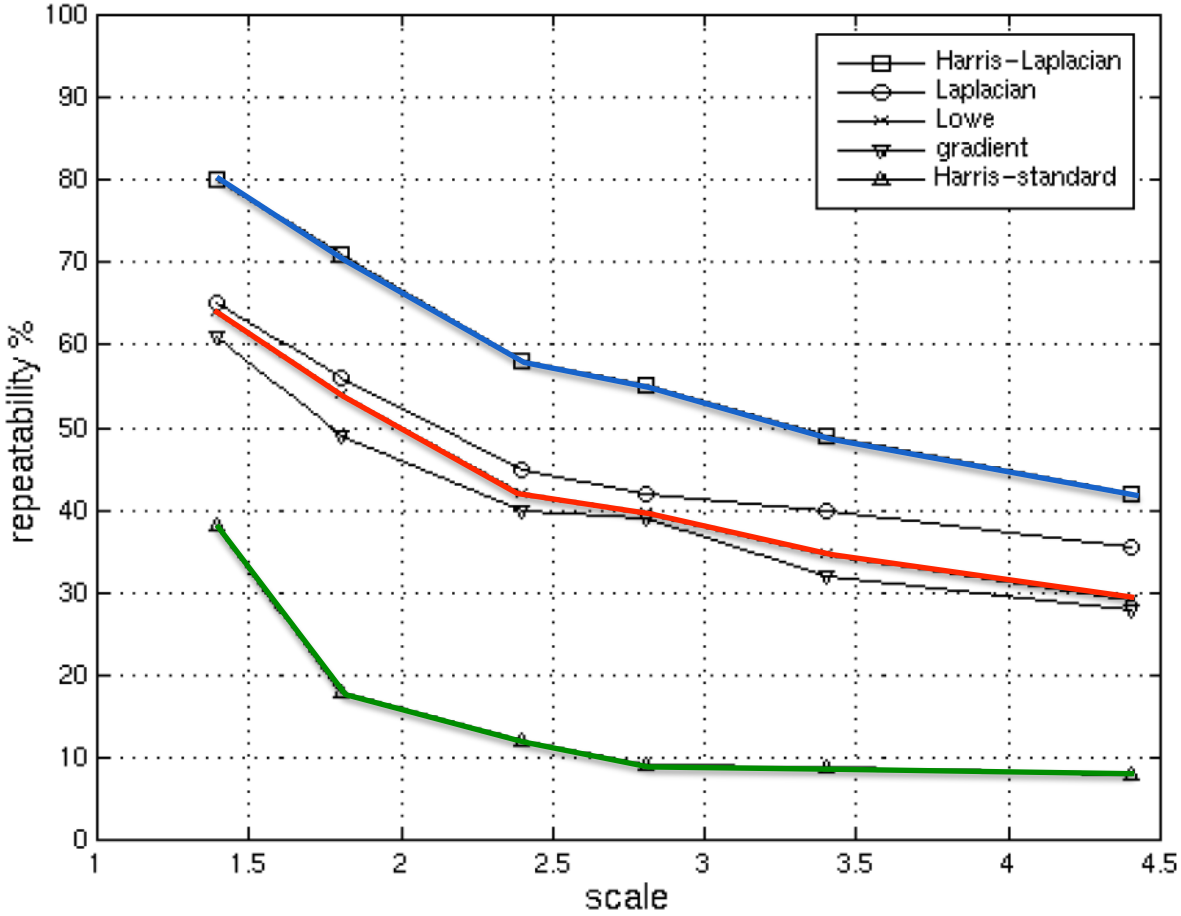


x ist interest point, falls DoG(x) größer/kleiner als alle 26 Nachbarn

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



Harris-Laplacian

SIFT

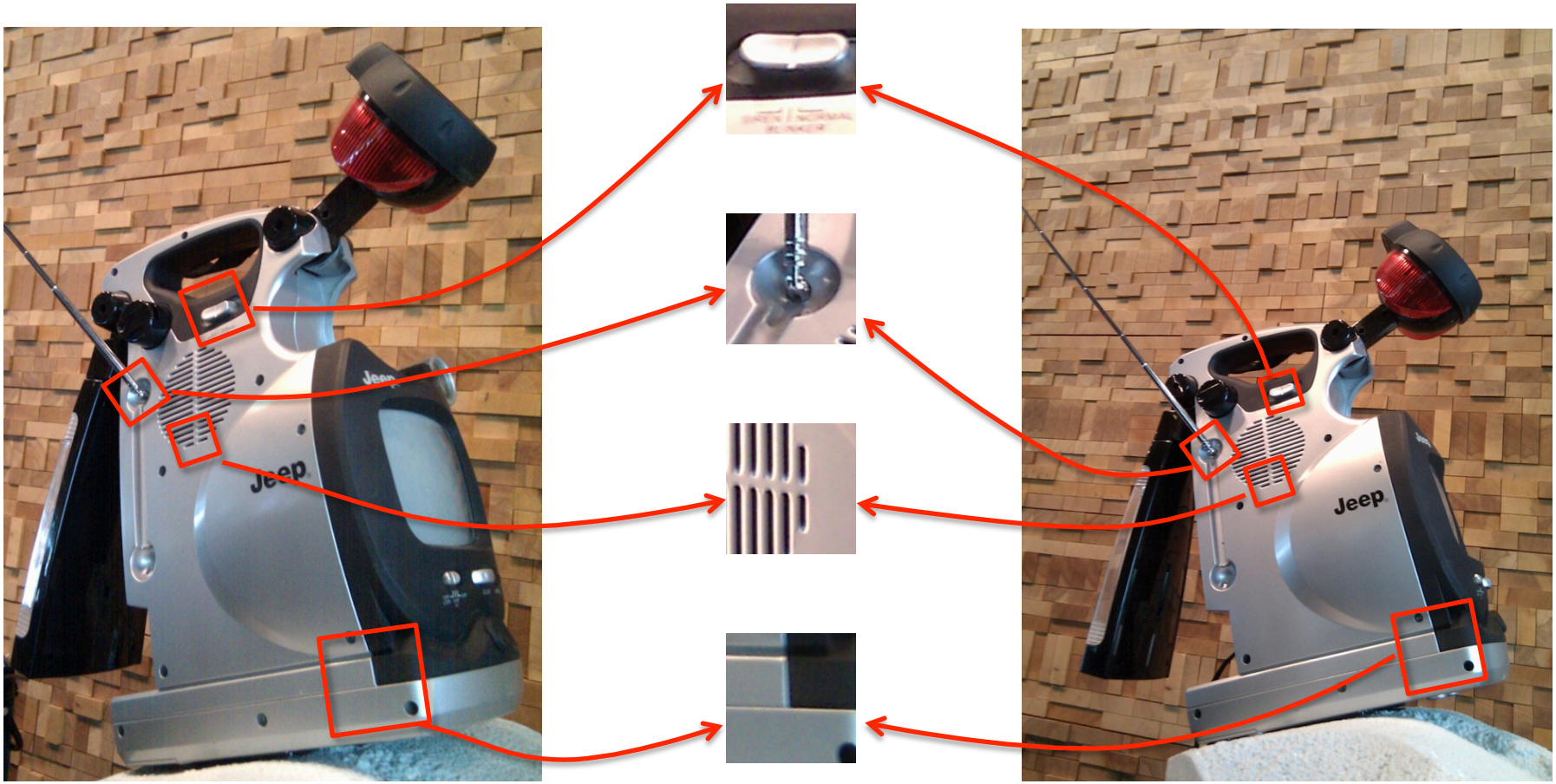
Harris-standard

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

FEATURE DESCRITOREN (MERKMALSVEKTOREN)

David Lowe: Skalierungsinvariante lokale Merkmale (SIFT)

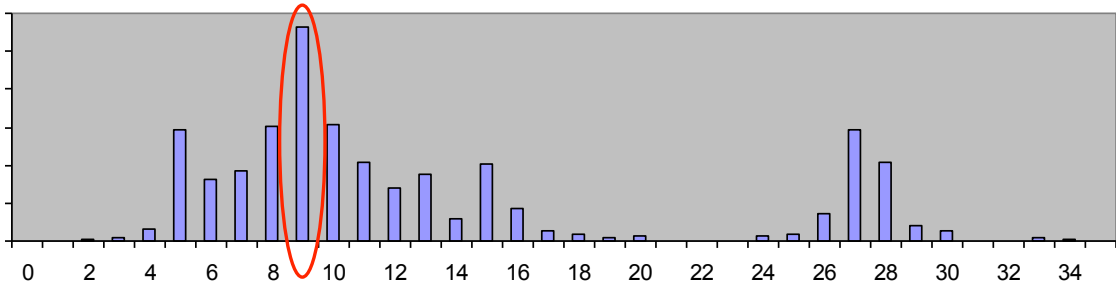
- Lokales Koordinatensystem um interest point
  - invariant gegenüber Translation, Rotation, Skalierung

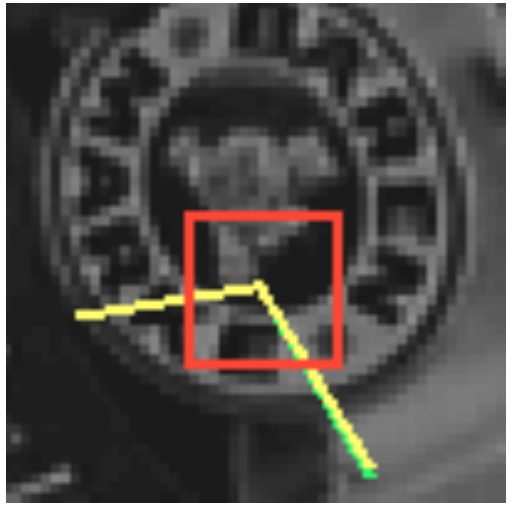


David G. Lowe: Object Recognition from Local Scale-Invariant Features. Proc. of ICCV 1999.

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



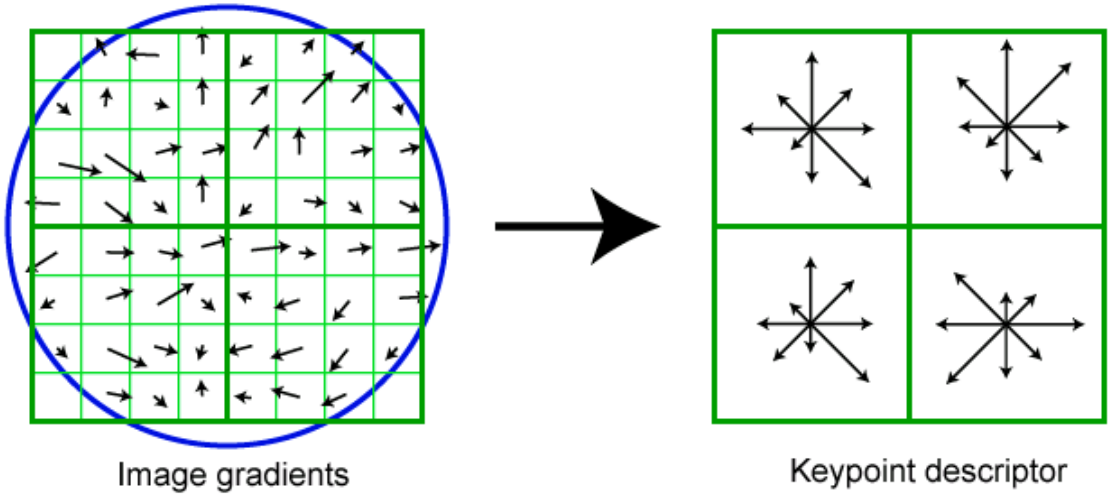


David G. Lowe: Object Recognition from Local Scale-Invariant Features. Proc. of ICCV 1999.

SIFT Merkmalsvektor

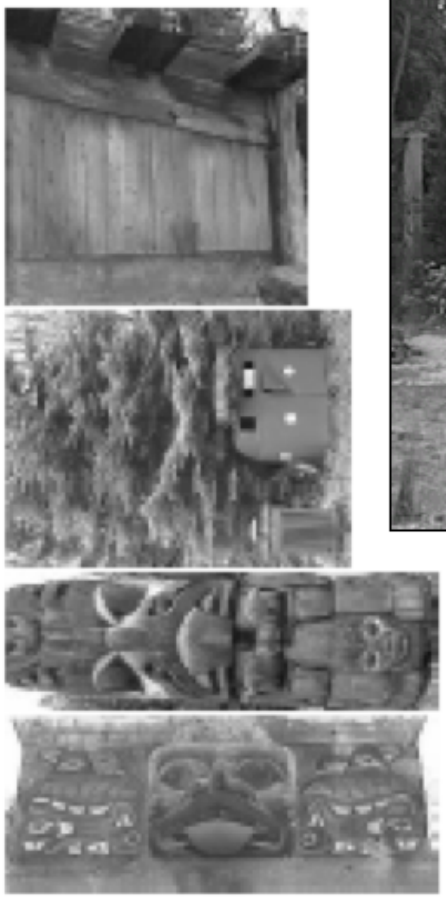
- patch: 16x16 „pixel“ im lokalen Koordinatensystem des interest points
  - Koordinatensystem lokalisiert in x, y, Skalierung, Orientierung
- 4x4 Orientierungshistogramme mit je 8 Orientierungen
  - Gewichtet mit Gradientenlänge und Distanz zum Zentrum

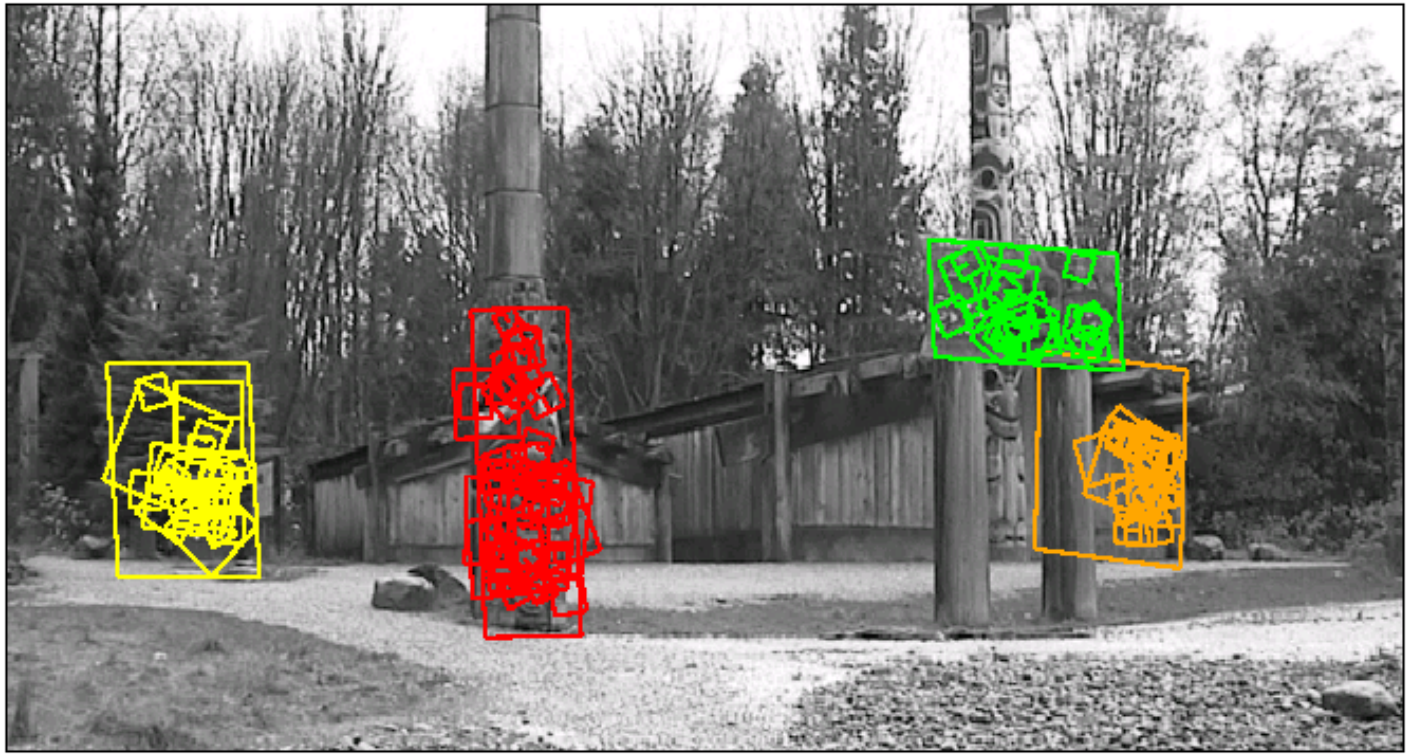
→ 128 Dimensionen



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

Bilderkennung mit SIFT





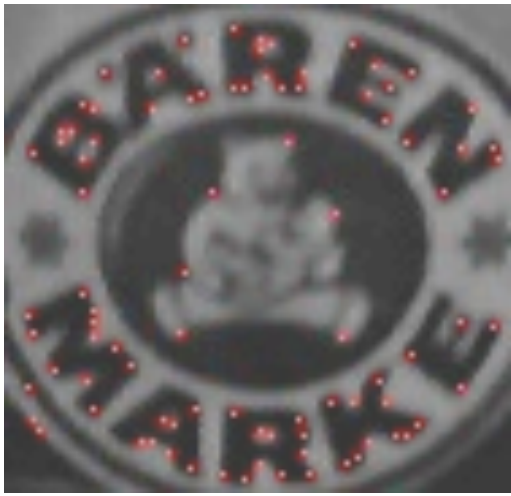
Quelle Abb.: David G. Lowe: Object Recognition from Local Scale-Invariant Features. Proc. of ICCV 1999.

SIFT for Mobile Devices [Wagner at al.]

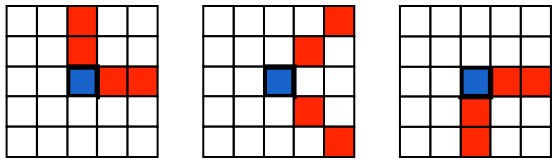
- Variant of SIFT with 36 component feature vector
  - Less computation (original feature vector: 128 components)
- Efficient computation of interest points
  - Variant of FAST corner detector
- Process template at multiple scales

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

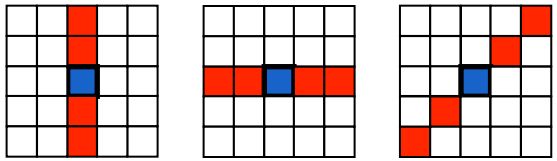
Interest Points



- 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



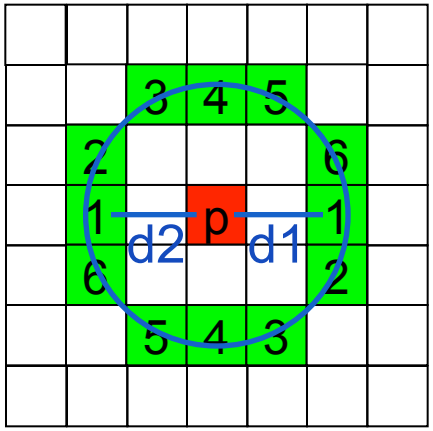
  - Idea: “It’s a corner if it’s not part of a straight line”
  - Can be implemented efficiently

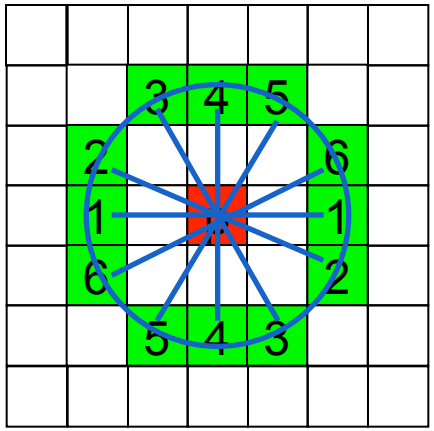
Interest Points

- Corner response of pixel p = (x,y)

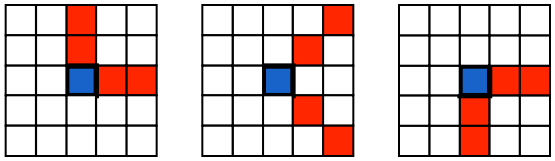
```
// f(p) = intensity (grayscale value) of pixel p
dmin = ∞
for all opposite points (p1,p2) on circle
    d1 = abs(f(p) - f(p1))
    d2 = abs(f(p) - f(p2))
    d = max(d1, d2)
    dmin = min(dmin, d)
cornerResponse = dmin
```

- Non-maximum suppression
- Threshold to generate ~150 corners

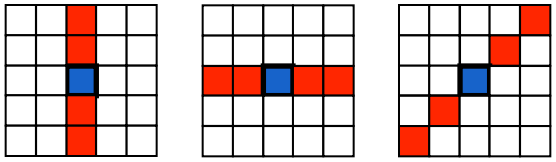




Corner examples:



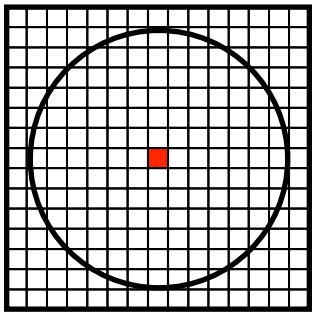
Not corner examples:



FAST Corner Detector

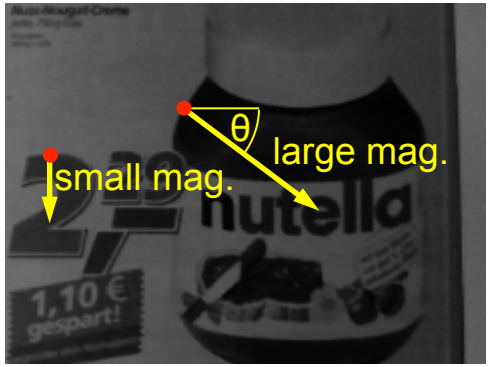
- auf voriger Folie: vereinfachte Version des FAST Corner Detectors
  - FAST: Features from Accelerated Segment Test
- E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In Proc. of European Conference on Computer Vision, May 2006.
- Algorithmus: für jedes Pixel p
  - konstruiere Bresenham-Kreis mit Radius r = 3
  - früher Abbruch möglich, falls d_{min} früh unter Schwellenwert

Mobile SIFT: Feature Descriptors



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

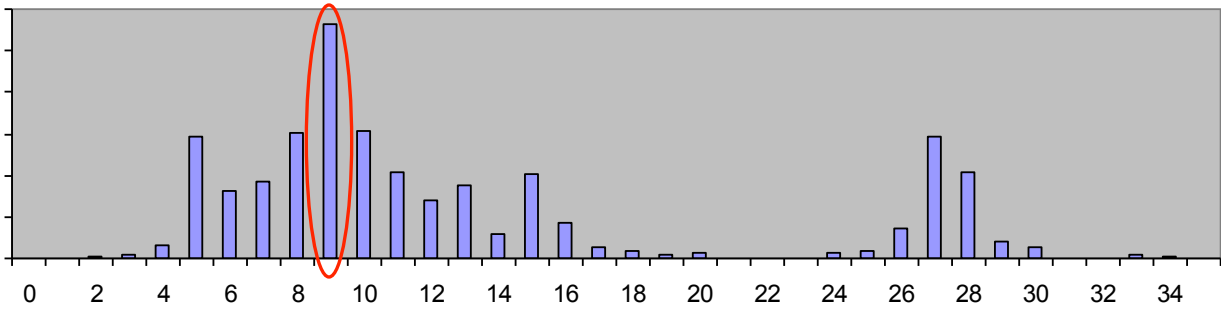
Original image:

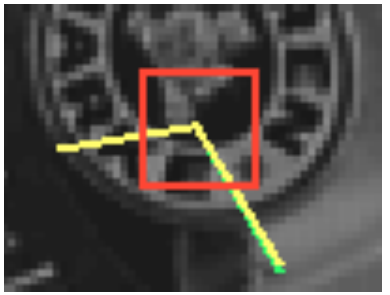


Gradient magnitudes:

Gradient orientations:

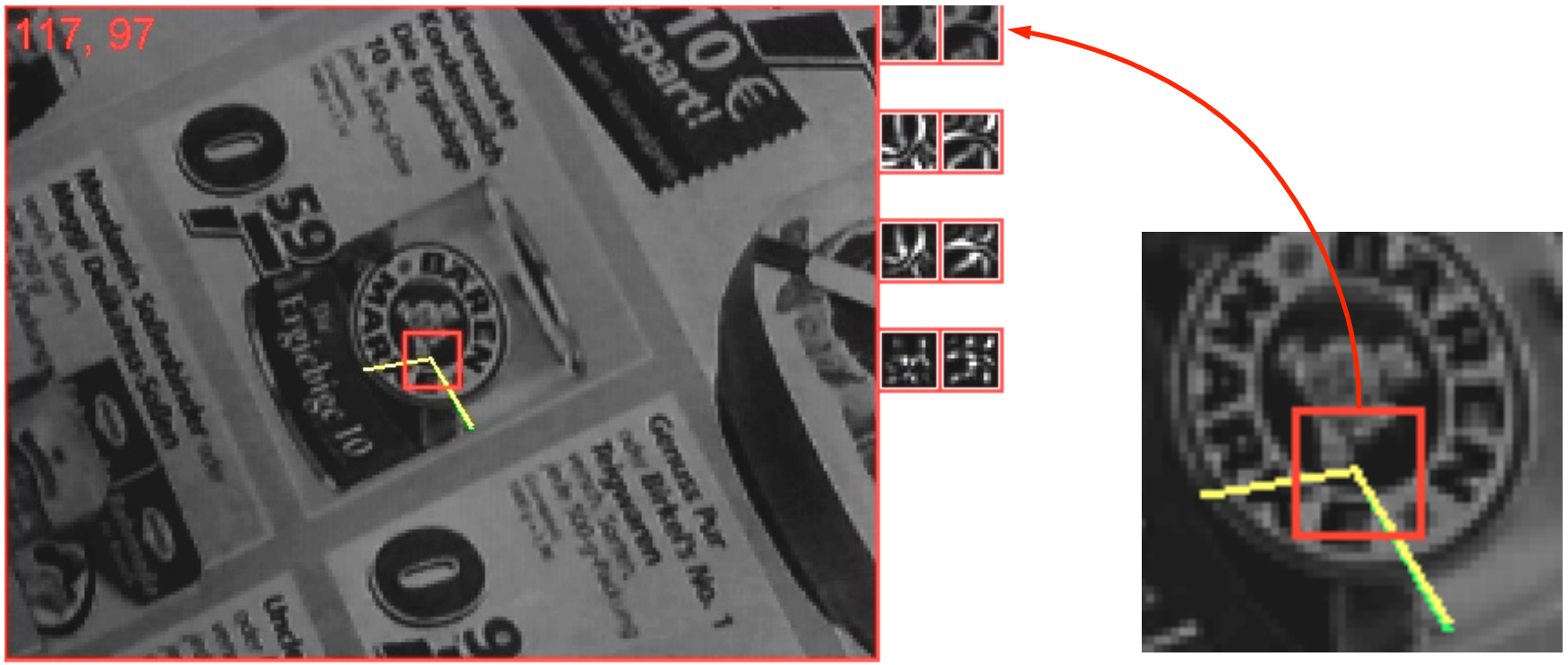
- Orientation histogram with 36 buckets (each covering 10°)



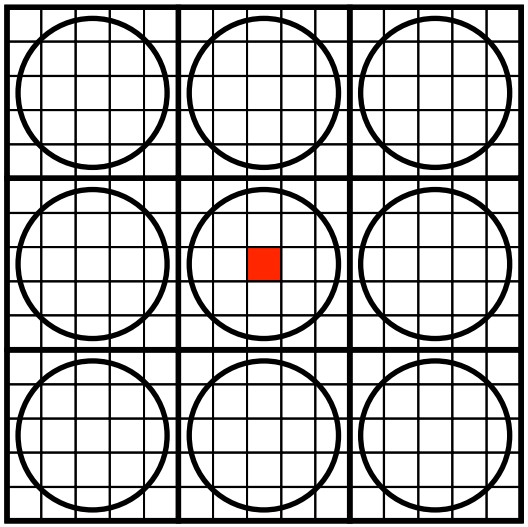


Mobile SIFT: Feature Descriptors

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

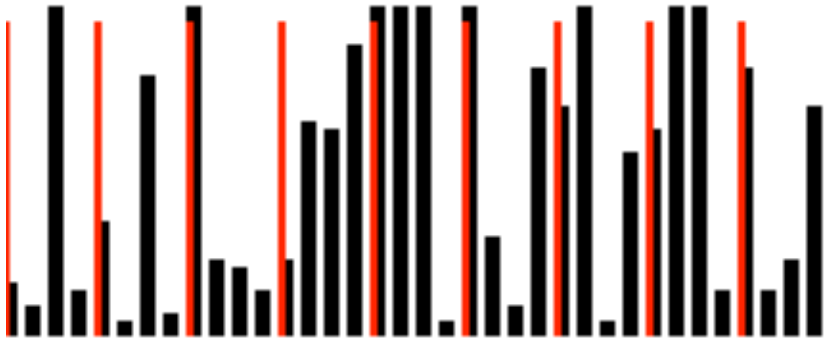


Mobile SIFT: Feature Descriptors



- 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

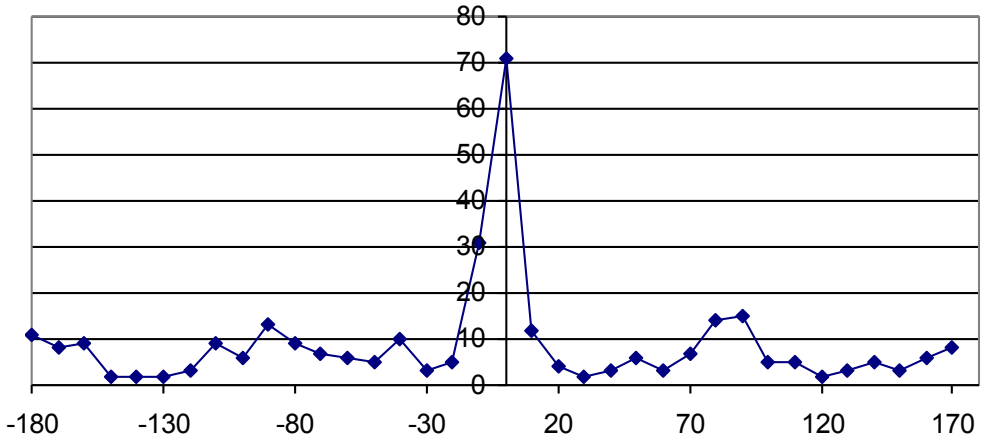
Scale space pyramid
width(i+1) = width(i) / sqrt(2)

Mobile SIFT: Feature Matching

- SIFT features are robust to perspective distortion
  - Each individual feature relatively weak (about <30% correct matches)
- Find best feature matches
  - Given query descriptor, find closest descriptor in template
  - Distance measure: sum of squared differences (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.]

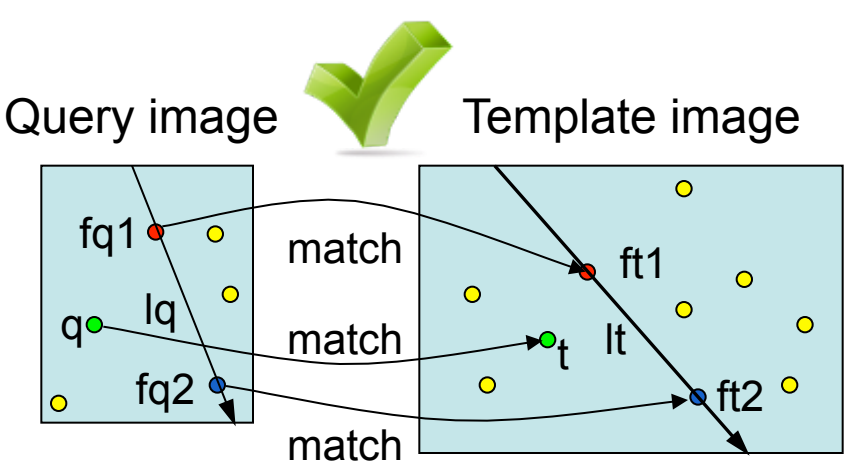
Mobile SIFT: Feature Matching

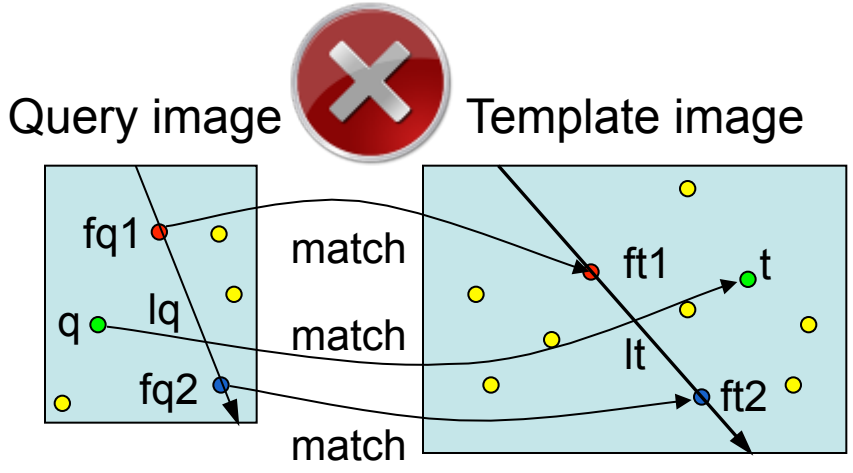
- Found feature matches may be wrong
  - Might not be identical, even though descriptor very close
- Removing outliers from feature matches
- Orientation difference test
  - For all matches compute orientation differences
  - Remove all matches ≥ ±30° from histogram peak
  - Works best for planar scenes



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



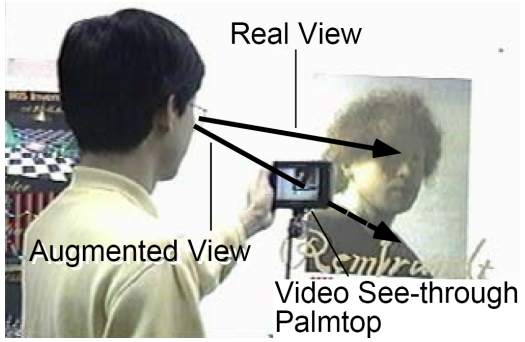


Markerless Tracking for Mobile Devices

- Daniel Wagner et al.: Pose Tracking from Natural Features on Mobile Phones. ISMAR 2008.
  - Real-time tracking of natural features on planar targets
  - FAST corner detector
  - Not fully scale invariant

Mobile Augmented Reality

- Video see-through augmentation with camera-equipped handheld devices
  - Handheld device as alternative to HMDs
- Align superimposed graphics with real-world view
  - Registration problem



Source: Rekimoto: Magnifying Glass Approach to Augmented Reality Systems, 1995

Source: Wagner et al., ISMAR 2008

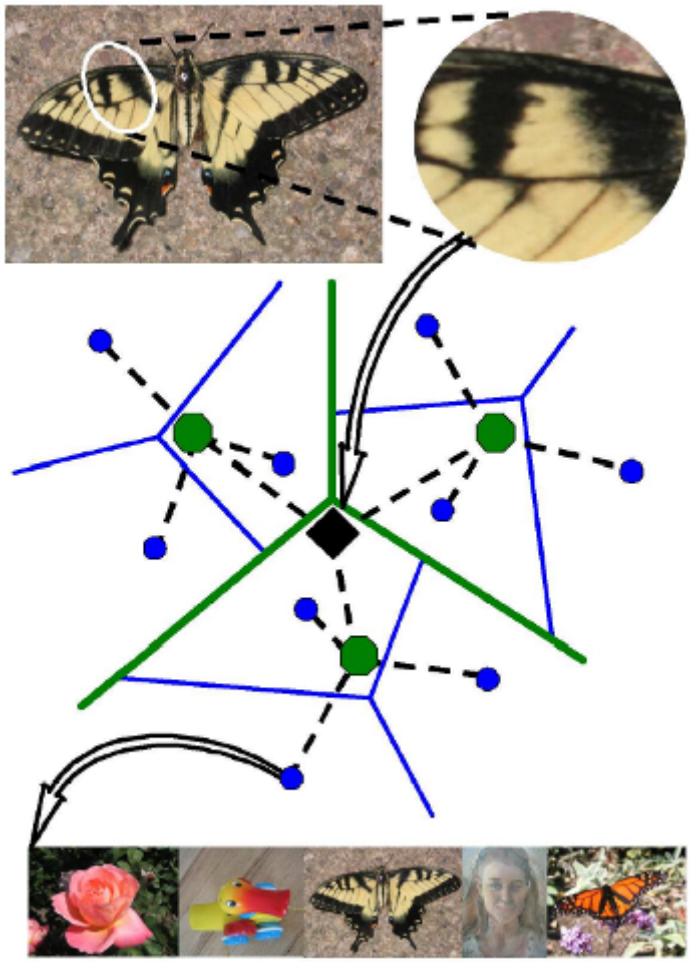
Source: Wagner: Handheld AR Displays, VR 2006

Source: Föckler: PhoneGuide, 2005

Matching Images in large Databases

- Scalability
  - Find matching image in large database
- Methods from document retrieval
  - Compute clusters of descriptors (“visual words”)
  - Weight descriptor by frequency in image and inverse frequency across images
  - Term Frequency Inverse Document Frequency (TF-IDF)
  - [Nister, Stewenius: Scalable Recognition with a Vocabulary Tree. CVPR 2006.]

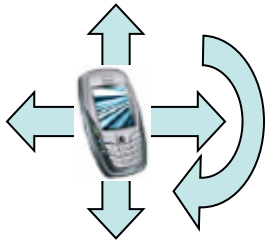
Vocabulary trees:

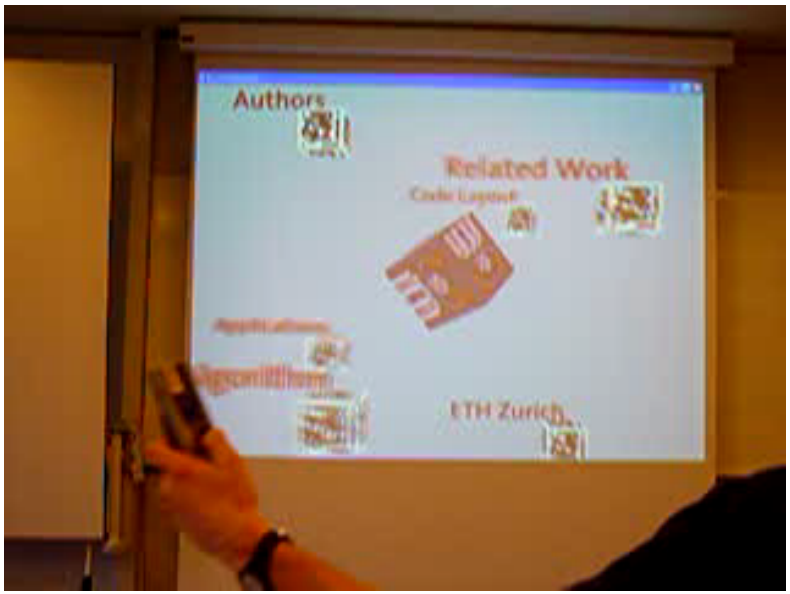


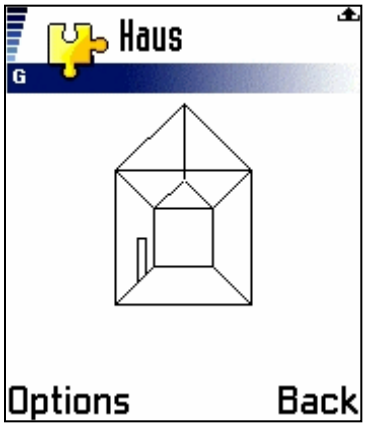
OPTICAL MOVEMENT DETECTION

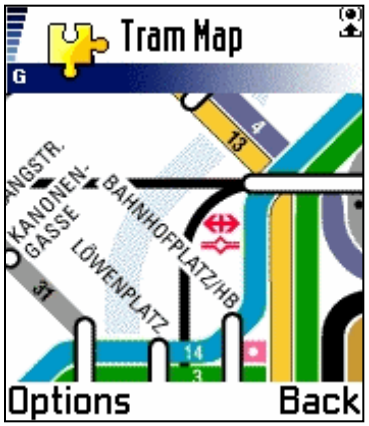
Optical Movement Detection (“Sweep”)

- Visual detection of device movement relative to the large display
- Continuous scrolling of screen contents
- Direct control of external displays
- 3 degrees of freedom (DOF)
  - x
  - y
  - θ



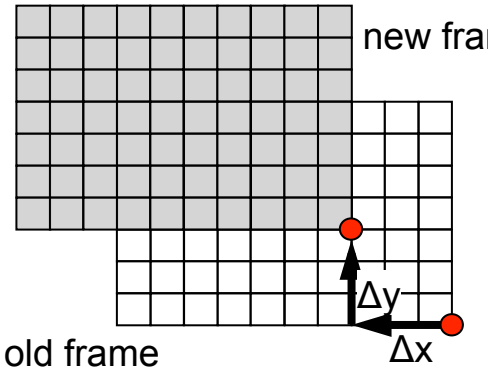


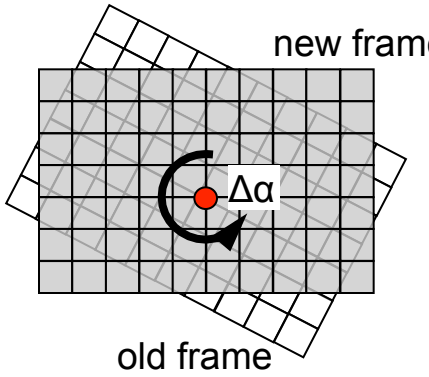


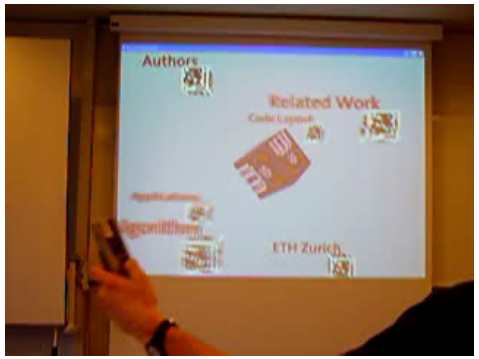


Rafael Ballagas, Michael Rohs, Jennifer G. Sheridan, Jan Borchers: Sweep and Point & Shoot: Phonecam-Based Interactions for Large Public Displays. Extended abstracts of CHI 2005.

Optical Movement Detection ("Sweep")







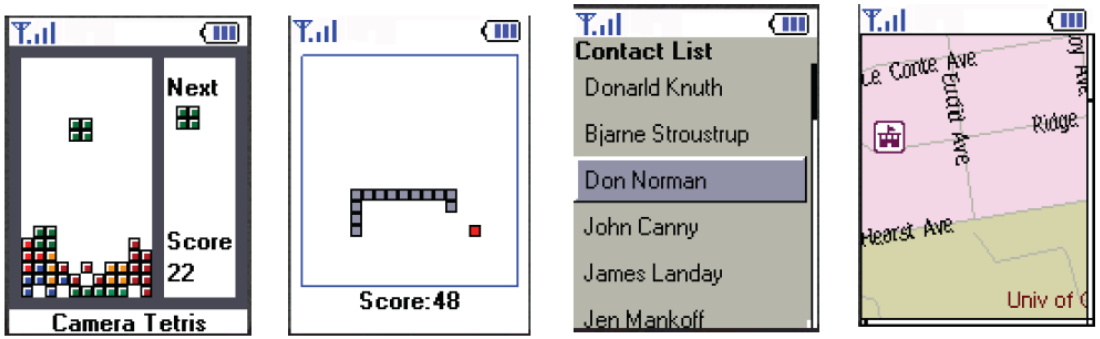
- Subdivide 176x144 pixel image in 8x8 pixels blocks
- Compute cross-correlation between adjacent frames
  - Frames are 33 ms apart (at 30 fps)
  - Sample spacing: 4 pixels
- Try a range of linear (Δx, Δy) offsets

$$r_t(dx,dy) = \frac{\sum_{y=0}^{h-1}\sum_{x=0}^{w-1} b_1(x,y)b_2(x+dx,y+dy)}{(w-|dx|)(h-|dy|)}$$

$$(\Delta x,\Delta y) = \underset{dx,dy\in\{-4,\ldots,4\}}{\operatorname{argmax}} r_t(dx,dy)$$

TinyMotion: Camera Phone Based Motion Sensing (Jingtao Wang, Berkeley)

- Camera-based sensing of device motion
  - Detects horizontal, vertical, rotational, tilt
  - Controls scrolling, zooming, menu selection, cursor movement, gesture/handwriting input
- References
  - Wang, Zhai, Canny: Camera Phone Based Motion Sensing: Interaction Techniques, Applications and Performance Study, UIST 2006.
- http://tinymotion.org



Source: Wang et al. TinyMotion: Camera Phone Based Interaction Methods. CHI 2006.

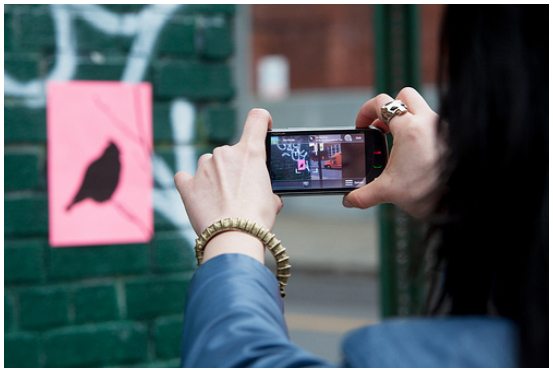
TARGET ACQUISITION WITH CAMERA PHONES

Mobile Augmented Reality Pointing

Layar

Wikitude

WikEye



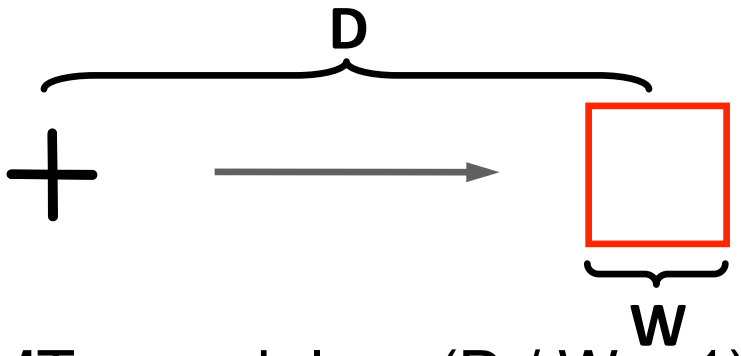
Nokia Point & Find

Google Goggles

Image courtesy of VentureBeat

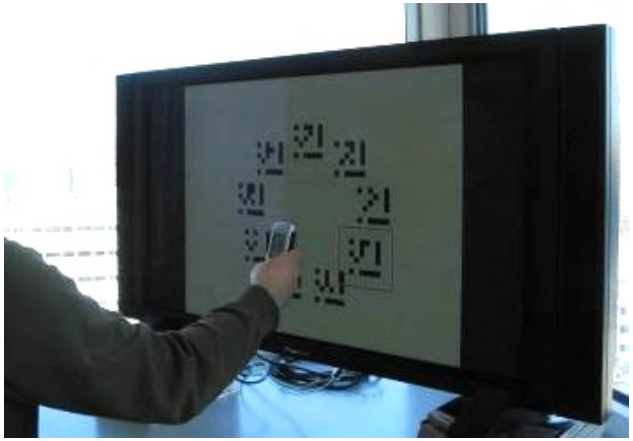
Modeling Mobile AR Pointing with Fitts' Law?

- Goal-directed movement onto target



- $MT = a + b \log_2 (D / W + 1)$

- Lab study (Rohs, Oulasvirta, 2008): Fitts' law does not accurately predict movement time for see-through AR pointing

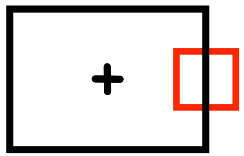


Analysis of Mobile AR Pointing Task

- Task: Move cursor onto target

- Phase 1: Target directly visible
  Task 1: Move lens over target

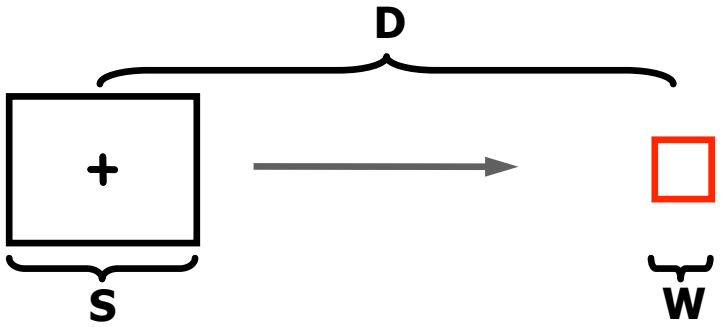
- Phase 2: Target behind display
  Task 2: Move crosshair over target



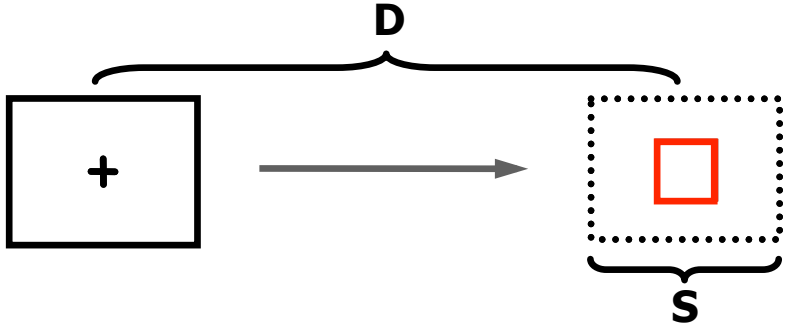
(Rohs, Oulasvirta, Target Acquisition with Camera Phones when used as Magic Lenses. CHI 2008)

Analysis of Mobile AR Pointing Task

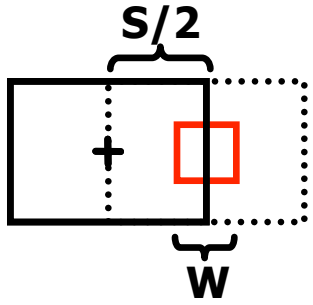
- Task: Move cursor onto target



- Phase 1: Target directly visible
  $MT_p = a_p + b_p \log_2(D / S + 1)$



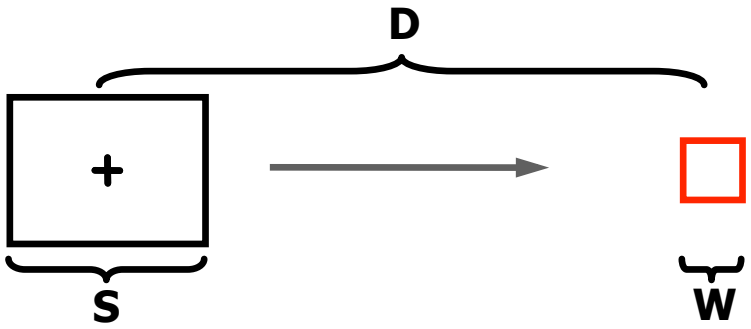
- Phase 2: Target behind display
  $MT_v = a_v + b_v \log_2(S/2 / W + 1)$



(Rohs, Oulasvirta, Target Acquisition with Camera Phones when used as Magic Lenses. CHI 2008)

Model for Mobile AR Pointing

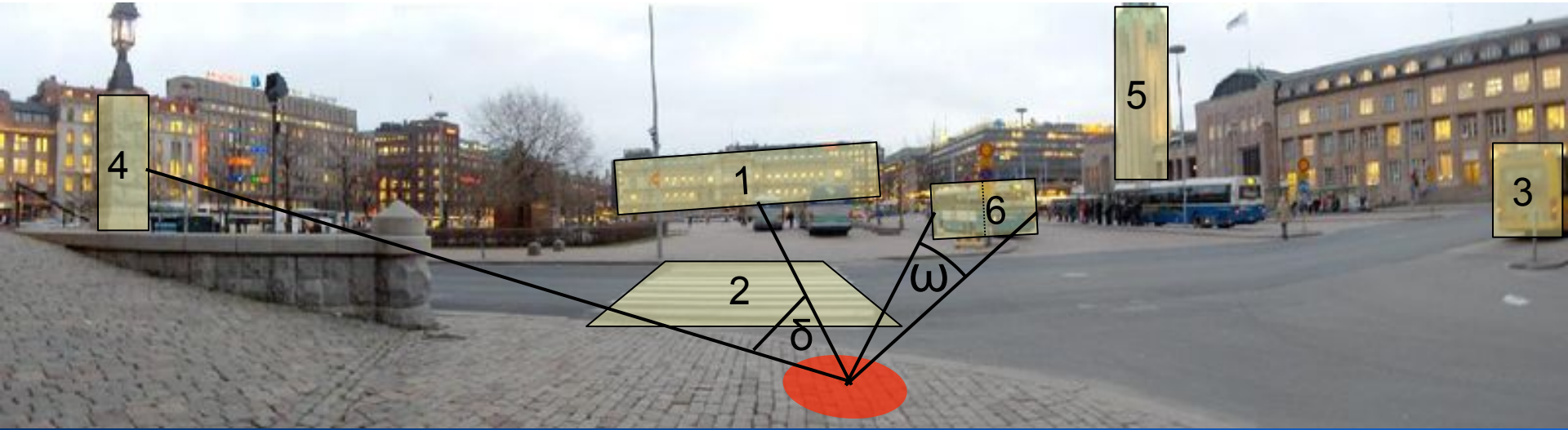
- Two-component Fitts' law model
- $MT = a + b \log_2(D / S + 1) + c \log_2(S/2 / W + 1)$



(Rohs, Oulasvirta, Target Acquisition with Camera Phones when used as Magic Lenses. CHI 2008)

Mobile AR Pointing in the Real World

- 3D targets, varying shape, size, z-distance, visual context
- Angular measure of target distance δ and size ω



Experimental Results

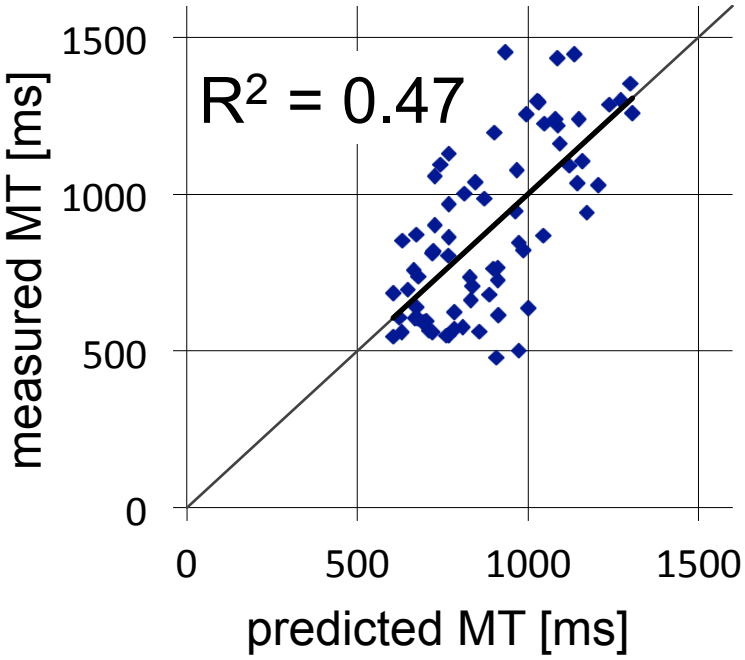
MT_{avg} = 885 ms, errors = 2%

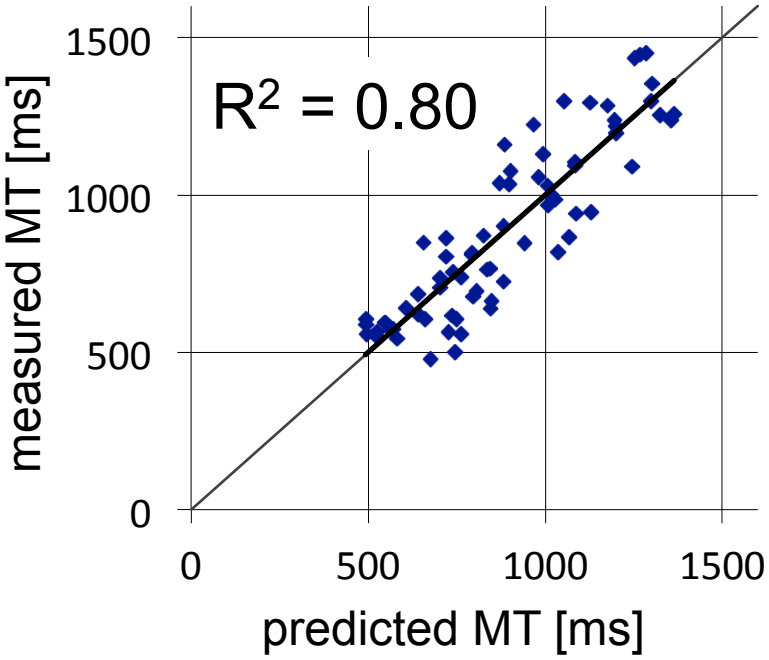
Standard Fitts' law:

$t = 447 + 220 \log_2(D/W+1)$ [ms]

Two-component model:

$t = 185 + 727 \log_2(D/S+1) + 62 \log_2(S/2/W+1)$ [ms]





The End