

$$R(u,v) = \sum_{i=-\delta_i/2}^{+\delta_i/2} \sum_{j=-\delta_j/2}^{+\delta_j/2} H(i,j)I(i+u,j+v)$$

Beispielrechnung Faltung

$$\begin{array}{ccccccc} & \vdots & \vdots & \vdots & \vdots & & \\ & 7 & 7 & \textcircled{7} & 0 & 0 & 0 \\ & 7 & \textcircled{7} & 7 & \textcircled{0} & 0 & 0 \\ & 7 & 7 & 7 & 0 & 0 & 0 \\ & \vdots & \vdots & \vdots & & & \end{array} * \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} = \begin{array}{ccccccc} & \vdots & \vdots & \vdots & \vdots & & \\ & 0 & 0 & \textcircled{28} & 28 & 0 & 0 \\ & 0 & \textcircled{0} & 28 & \textcircled{28} & 0 & 0 \\ & 0 & 0 & 28 & 28 & 0 & 0 \\ & \vdots & \vdots & \vdots & & & \end{array}$$

$$\begin{array}{ccccccc} & \cdots & \cdots & \cdots & \cdots & \cdots & \\ & 0 & 0 & 0 & 0 & 0 & \\ & 0 & 0 & 0 & 0 & 0 & \\ & 7 & 7 & 7 & 7 & 7 & \\ & 7 & 7 & 7 & 7 & 7 & \\ & \cdots & \cdots & \cdots & \cdots & \cdots & \end{array} * \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} = \begin{array}{ccccccc} & \cdots & \cdots & \cdots & \cdots & \cdots & \\ & 0 & 0 & 0 & 0 & 0 & \\ & 0 & 0 & 0 & 0 & 0 & \\ & 0 & 0 & 0 & 0 & 0 & \\ & 0 & 0 & 0 & 0 & 0 & \\ & \cdots & \cdots & \cdots & \cdots & \cdots & \end{array}$$



Faltung von Bildern (Bsp.)

Gaußkernel
mit Standardabweichung σ

$$G_{\sigma}(i,j) = \frac{1}{2\pi\sigma^2} e^{-\left(i^2 + j^2\right)/2\sigma^2}$$

(Im Kernel: hell = hoher Wert; dunkel = niedriger Wert)



Center Surround Filter



Gauß-Filter

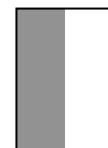
Kantenerkennung: Sobelfilter

Sobelfilter

Filter zur Erkennung von Grauwertkanten (horizontal, vertikal), basierend auf Faltung mit einer Kombination/Auswahl der folgenden Kernels (hier 3x3):

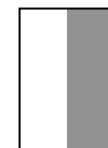
H_{SobelV}

$$\begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}$$



H_{SobelV}

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$



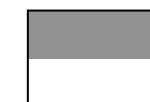
H_{SobelH}

$$\begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$



H_{SobelH}

$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$





Sobel gefiltertes Bild

Sobelfilter betonen Kanten;
absolute Grauwerte sind egal

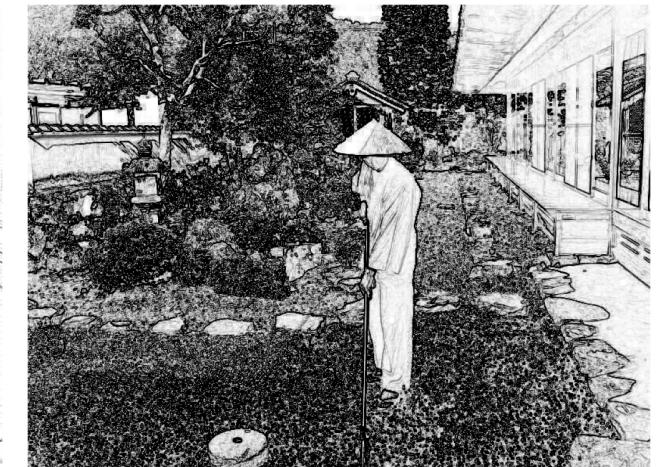
mit H_{SobelV}



mit H_{SobelH}



Addition aller 4 H/V



Laplace-Filter

Beobachtung

Sobelfilter erkennen unangenehm viele Kanten!
(hochfrequente Strukturen, Bildrauschen)

Ideen

1. Glätte das Bild (z.B. mit Gaußfilter) vorm Kantenerkennen!
2. Benutze 1 Filter, der glättet **und** Kanten erkennt!

2D-Laplace-Operator

Für (2x) diff'bare Funktionen:
Summe der beiden 2. Ableitungen

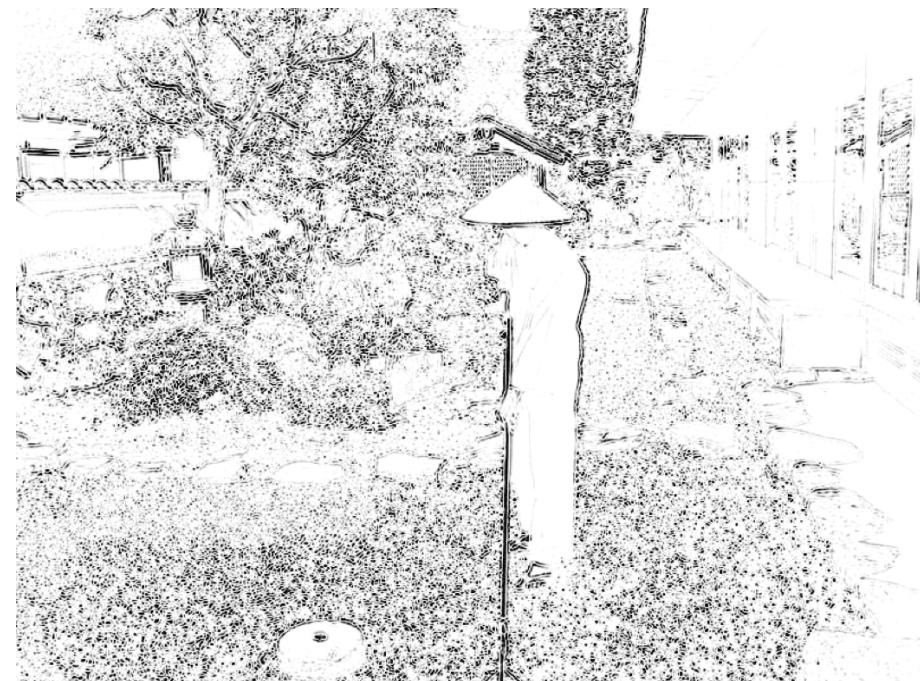
$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Laplace-Kernel

approximiert L-Operator
im Diskreten

$$H_{\text{Laplace}} = \pm \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Laplacegefiltertes Bild



$$\begin{array}{ccccccccc} \vdots & \vdots & \vdots & & & & \vdots & \vdots & \vdots \\ 0 & 0 & 7 & 7 & 7 & 0 & 0 & 0 & -7 \\ 0 & 0 & 7 & 7 & 7 & 0 & 0 & 0 & -7 \\ 0 & 0 & 7 & 7 & 7 & 0 & 0 & 0 & -7 \\ 0 & 0 & 7 & 7 & 7 & 0 & 0 & 0 & -7 \\ \vdots & \vdots & \vdots & & & & \vdots & \vdots & \vdots \end{array} * \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} = \begin{array}{ccccccccc} 0 & -7 & 7 & 0 & 7 & -7 & 0 \\ 0 & -7 & 7 & 0 & 7 & -7 & 0 \\ 0 & -7 & 7 & 0 & 7 & -7 & 0 \\ 0 & -7 & 7 & 0 & 7 & -7 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \end{array}$$

Kantenerkennung

Bekannte Filter lassen zu viele Kanten übrig. Nachbehandlung:

Nicht-Maximum-Unterdrückung

Reduziere schwache, aber mehrere Pixel breite Kanten

- scharfe Kantenlinien, 1 Pixel breit

Schwellwert und Hysterese

Akzeptiere nur Kantenpixel über Schwellwert t_h .

Zusätzlich akzeptiere alle Kantenpixel über t_n (wobei $t_n \ll t_h$) wenn diese mit Kantenpixeln über t_h verbunden sind.

- durchgängige Kantenzüge trotz schwankender Stärke

Canny-Algorithmus zur Kantenerkennung

Canny-Kantenerkennung (Beispiel)



Canny mit hohem Schwellwert t_h

Eckenerkennung

- Wenn Kanten interessant in Bildern, dann erst recht Ecken!
(Markante Punkte zum Matching → Korrespondenzproblem)
- Ecken entstehen aus aufeinander treffenden Kanten – können dann Kantenerkenner auch Ecken erkennen?

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 7 & 7 & 7 & 0 & 0 \\ 7 & 7 & 7 & 0 & 0 \\ 7 & 7 & 7 & 0 & 0 \end{matrix} \quad \dots \quad \begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & 7 & 0 \\ 0 & 0 & 21 & 21 & 0 \\ 0 & 0 & 28 & 28 & 0 \\ 0 & 0 & 28 & 28 & 0 \end{matrix} \quad \dots$$
$$* \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} = \quad \dots$$

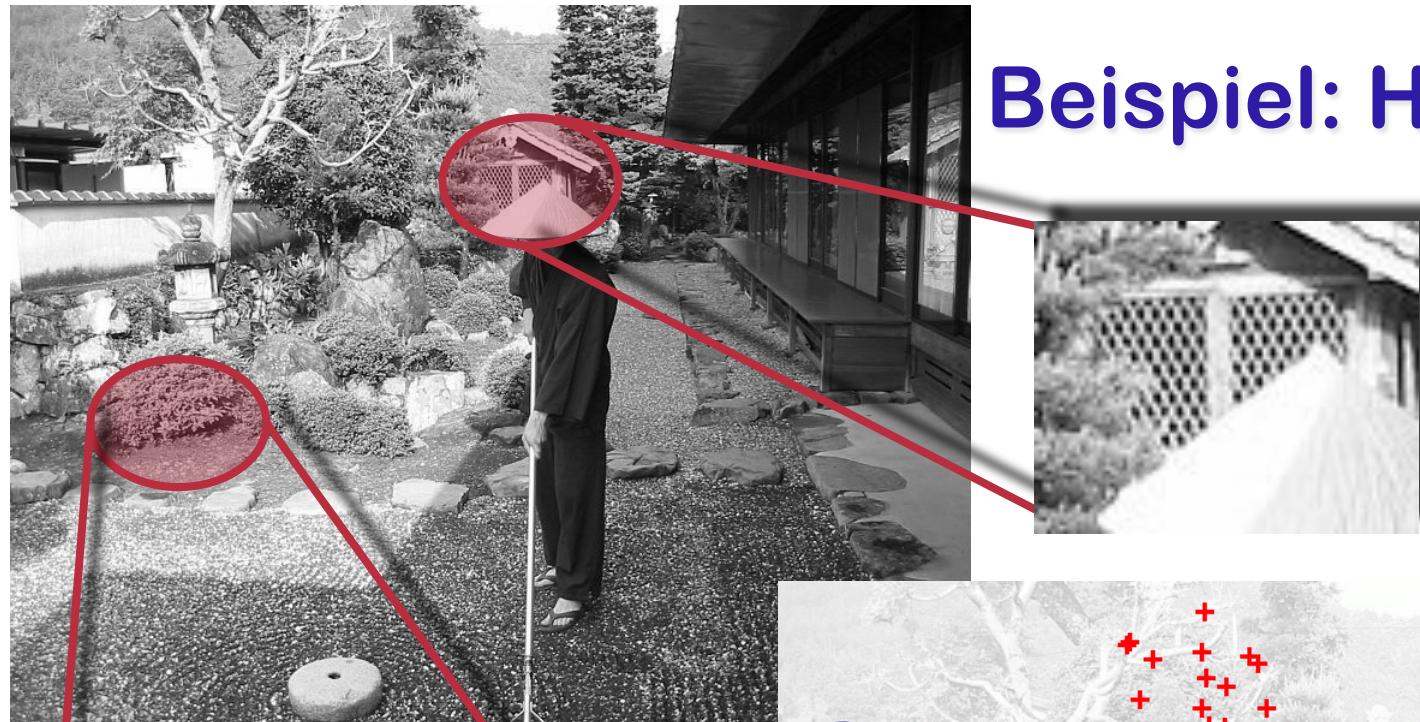
- Kantenerkenner „verwaschen“ Ecken!
- Kantenerkenner nutzen den klaren Gradienten an der Kante;
an einer Ecke gibt es den gerade **nicht!**

Der Harris-Eckenerkenner

Idee für einen **Eckenerkenner** (Harris&Stephens):

- Eine Ecke hat starke Gradienten in **zwei** Richtungen!
 - x bzw. y -**Gradient** I_x bzw. I_y an Bildpunkt $\mathbf{p}=(u,v)$ ist die Summe der Grauwertdifferenzen von \mathbf{p} und Nachbarn aus über n Pixel in x bzw. y -Richtung (für kleines n)
 - Betrachte **Gradientenmatrix** an \mathbf{p} nach Glätten (z.B. mit Gaußkernel) zur Rauschunterdrückung
- $$C(\mathbf{p}) = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$
- Hat $C(\mathbf{p})$ 2 deutlich positive Eigenvektoren (also: um \mathbf{p} gibt es 2 unterschiedliche, klare Gradienten), dann ist \mathbf{p} ein Eckpunkt
 - Hat $C(\mathbf{p})$ 1 deutlich positiven Eigenvektor und der andere ist ≈ 0 , so liegt \mathbf{p} auf einer Kante

Beispiel: Harris-Ecken



Eigenschaften des Harris-Erkenners

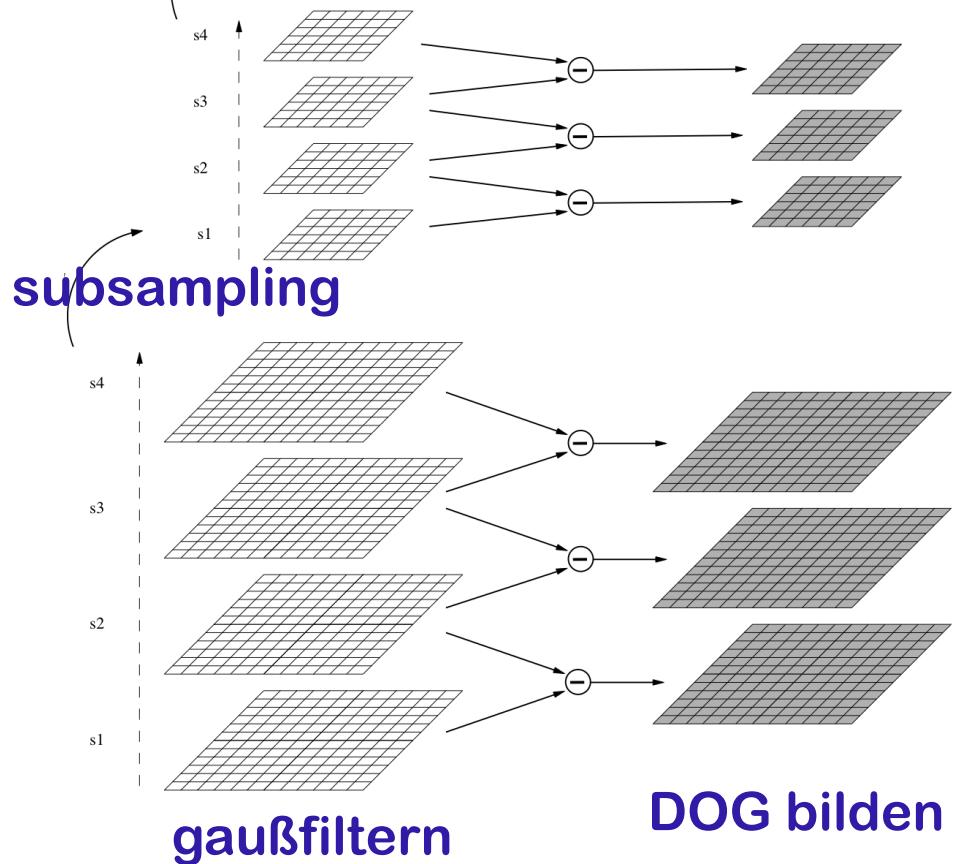
- rotationsinvariant
(sofern Glättung mit rotationssymmetrischem Kernel, z.B. Gauß)
- nicht großeninvariant
- ursprünglich entwickelt für Objektverfolgung in Bildsequenzen; auch verwendet für Stereo-Korrespondenz
- erkennt nicht nur Ecken, sondern allgemein „markante Stellen“ (*points of interest*) bzgl. Gradienten
(das können auch Linienenden oder kleine „Flecken“ sein)
- effizientere Implementierung braucht nicht konkrete Eigenwerte von $C(\mathbf{p})$, sondern Verhältnis der Eigenwerte
(über Berechnung von Determinante und Spur der Matrix $C(\mathbf{p})$)

SIFT-Merkmale: Die Idee

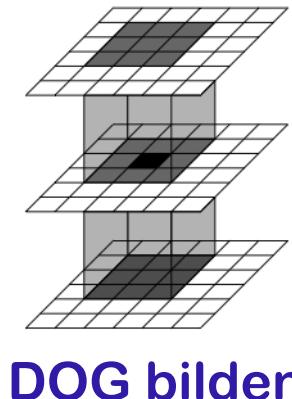
D. Lowe: *Distinctive Image Features from Scale-Invariant Keypoints*. J. Computer Vision, 2004

1. Entdecke **lokale Extrempixel** in einer Pyramide von Gaußsch verrauschten Versionen des Originalbilds
2. Aus diesen Extrema selektiere stabile **Schlüsselpunkte**
3. An allen Schlüsselpunkten berechne **lokale Intensitätsgradienten** (Stärke, Orientierung)
4. Relativ zu Gradienten erzeuge **normierte Deskriptoren** der Schlüsselpunkt-Umgebungen
 - **SIFT-Merkmal** = \langle Schlüsselpunkt, Gradient, Deskriptor \rangle
 - In normalen Bildern ~500-1000 SIFT-Merkmale
 - Rauschen und Normierung machen robust gegen Variation
 - SIFT erfolgreich einsetzbar für Matchen von Bildern i.A.!
Stereo-Matching, Objekterkennung, Tracking, Roboterlokalisierung, ...

SIFT#1 (Verrauschen+Extrema): Extrema in DOGs



- Iteriere Verrauschung
(→ „Gaußpyramide“)
 - Reduziere dabei Auflösung durch *subsampling*
 - Berechne Differenzbilder (*difference of Gaussians*, **DOGs**) zwischen Nachbarn gleicher Auflösung
- **Größeninvarianz** (in Grenzen)



DOG bilden

Extrema sind solche Pixel, die unter ihren 26 (= $1 \times 8 + 2 \times 9$) Nachbarpixeln der DOG-Pyramide maximal oder minimal sind

SIFT#2 (Schlüsselpunkte): Verwirf „unsichere“ Extrema

Nur solche Extrema aus #1 werden zu Schlüsselpunkten, die

- sich hinreichend stark (Schwellwert der Extrema) von ihren Nachbar-DOG-Pixeln unterscheiden und
- die im Bild nicht auf Grauwertkanten liegen

Details gehen über diese Vorlesung hinaus;
s. Lowe, 2004

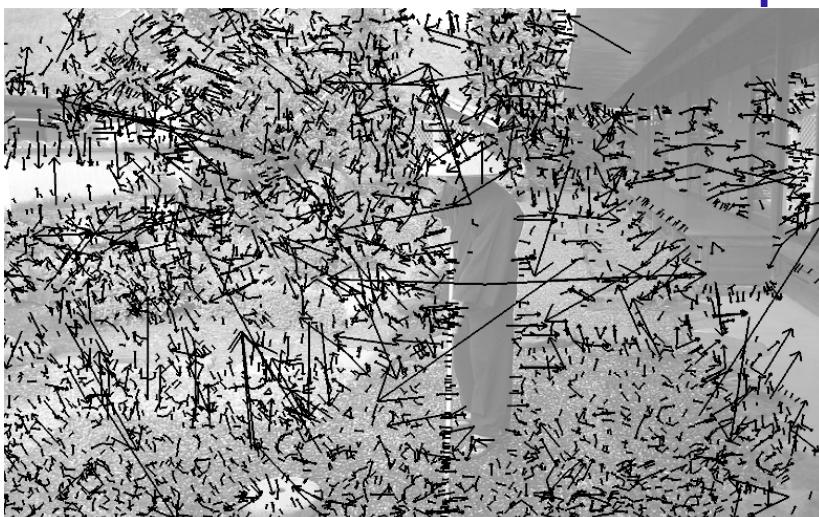
SIFT#3 (Intensit.-Gradienten): Schlüsselorientierung

In dem gefalteten/verrauschten Bild $R=I^*G_\sigma$, berechne für Schlüsselpunkt (x,y) den **Gradienten**

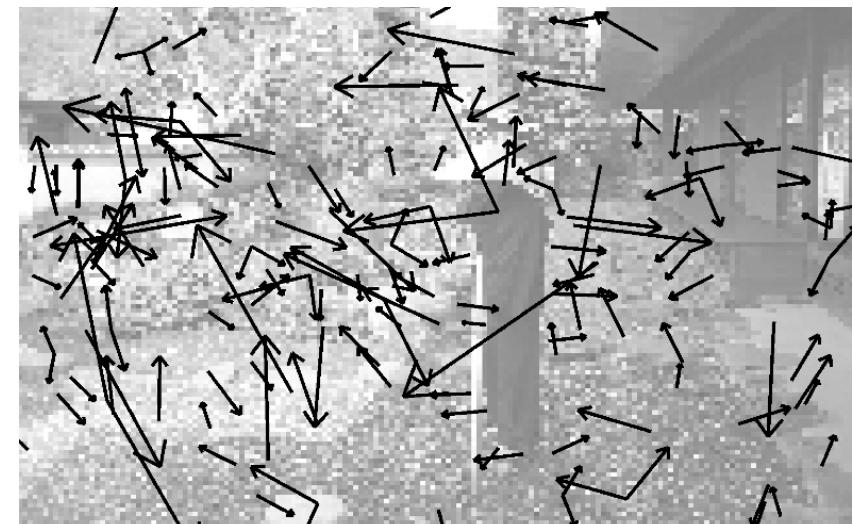
mit Magnitude $\Gamma(x,y) = \sqrt{(R(x+1,y) - R(x-1,y))^2 + (R(x,y+1) - R(x,y-1))^2}$

und Orientierung $\theta(x,y) = \text{atan} 2\left(\frac{R(x,y+1) - R(x,y-1)}{R(x+1,y) - R(x-1,y)}\right)$

Schlüsselpunkte mit Gradienten



Originalgröße 800x600: 4805 Schl.-P.

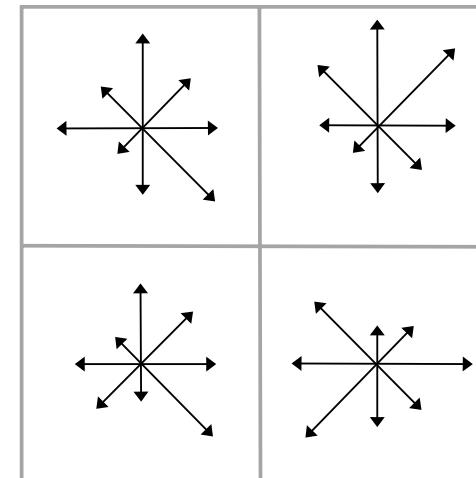
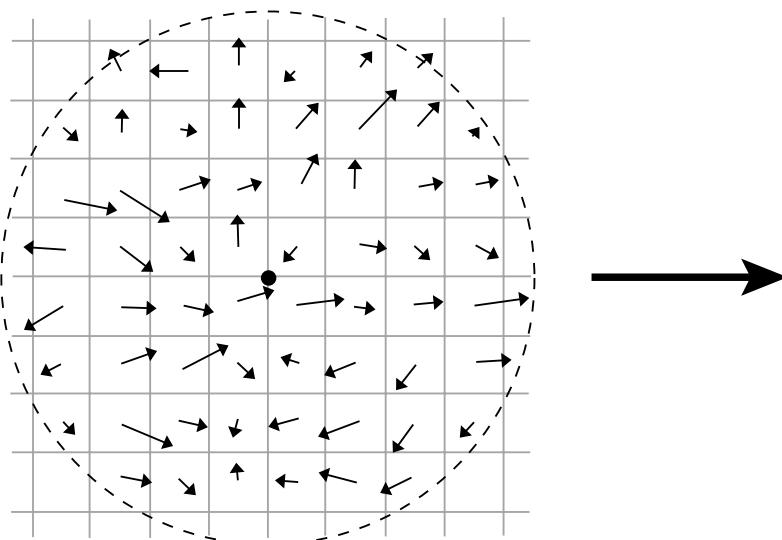


162x121: 174 Schlüsselpunkte

SIFT#4 (Deskriptoren): Lokale Gradienten

Schlüsselpunkt-Orientierung definiert lokales Bezugssystem!

In diesem Koordinatensystem (\rightarrow **Rotationsinvarianz**) berechne Gradienten der Umgebungspixel des Schlüsselpunkts; fasse diese gaußgewichtet quadrantenweise zu Histogrammen mit 8 vergröberten Gradientenrichtungen zusammen



SIFT-Deskriptor
ist $8 \times 2 \times 2$ -Vektor
dieser Werte!

In diesem Schema.
Auch $8 \times 4 \times 4$ möglich/
gebräuchlich