

# Advanced Robot Perception

Fortgeschrittene Konzepte der Wahrnehmung für  
Robotersysteme

Georg von Wichert, Siemens Corporate Technology

# Letzte Woche

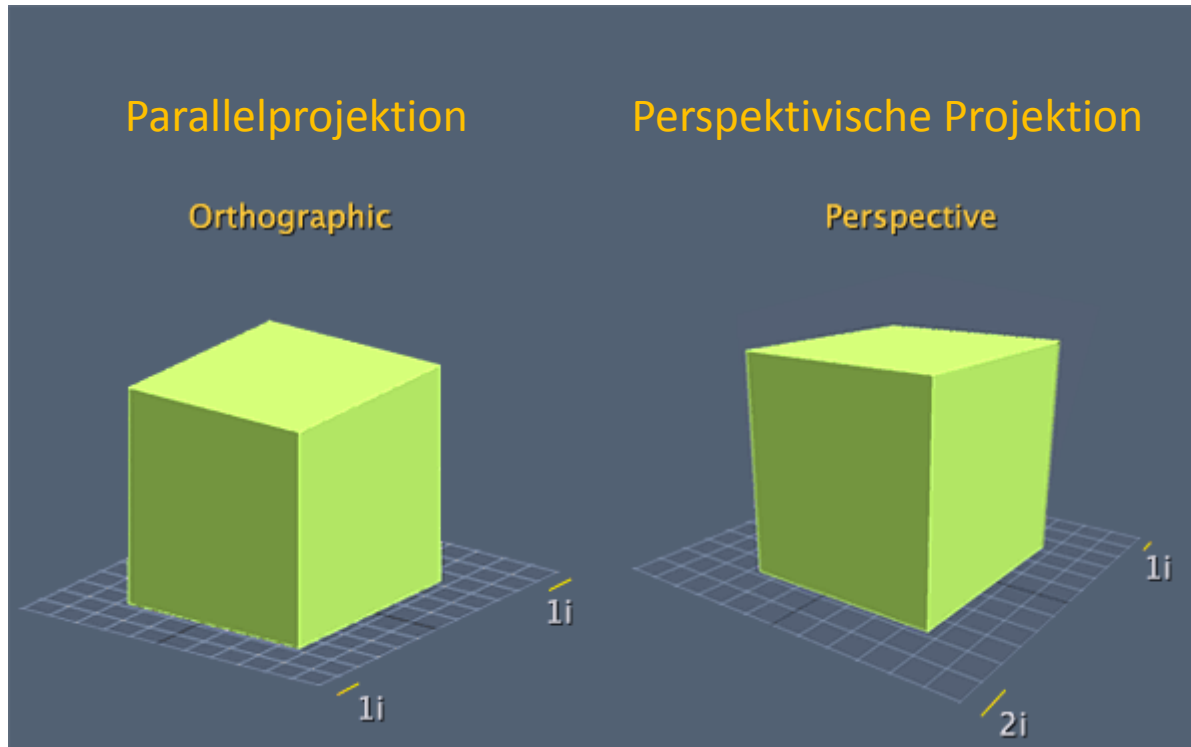
- Ein paar Algorithmenklassiker für den Umgang mit Punktwolken
  - Clusterbildung mit k-Means und Agglomeration
  - Registrierung mit ICP
  - Nachbarschaftsuche mit dem kd-Tree
- Kamerageometrie

# Diese Woche

- Kamerageometrie und Bildentstehung
- Grundlegendes zur Bildverarbeitung
  - Nur einige Basics als Basis für den Rest der Vorlesung
- Grundlegendes zur Mustererkennung
  - Bayes'sche Statistik: Probability & Belief
  - „Gute“ Entscheidungen

# **KAMERAS ALS SENSOREN FÜR DIE ROBOTIK**

# Bildentstehung: Kameramodell

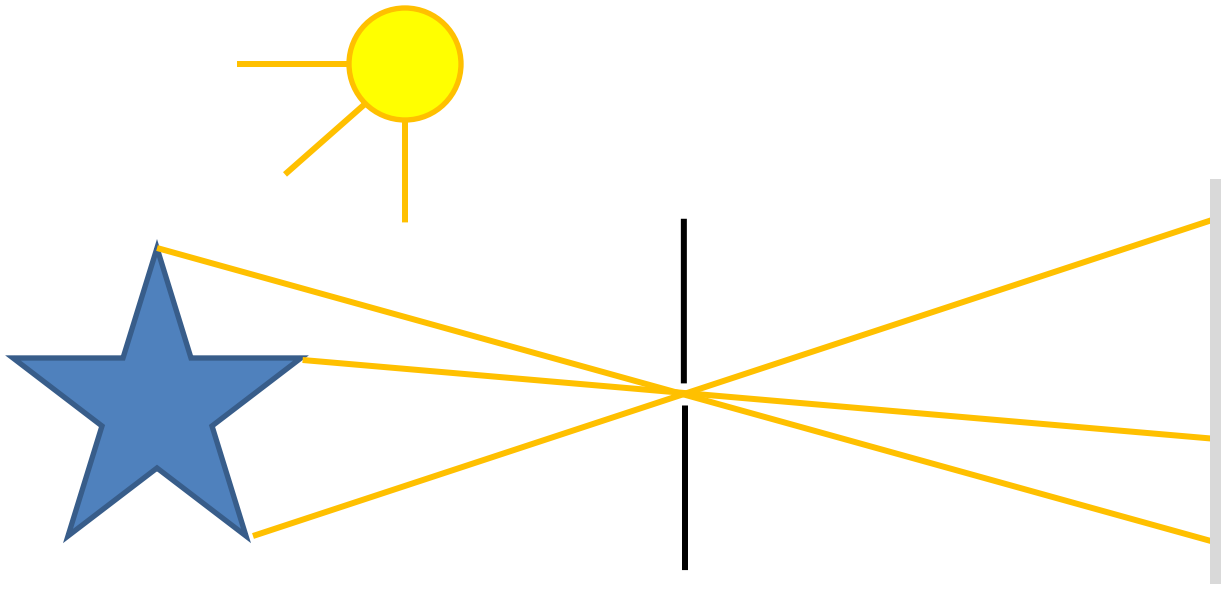


- Abbildungsgröße abstands-unabhängig
- Kein Fluchtpunkt
- Parallele Linien bleiben parallel

- Nahe Objekte erscheinen größer
- Fluchtpunkt
- Parallele Linien schneiden sich im Unendlichen

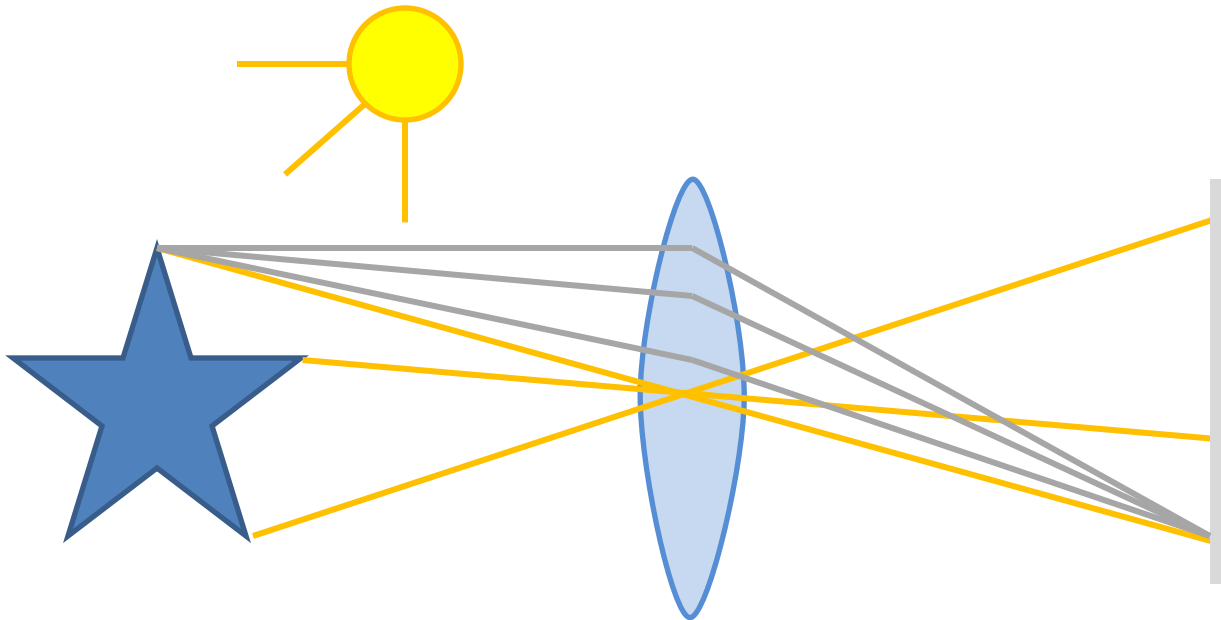
# Lochkamera

- Beleutetes Objekt strahlt Licht ab
- Sensor ist Lichtempfindlich

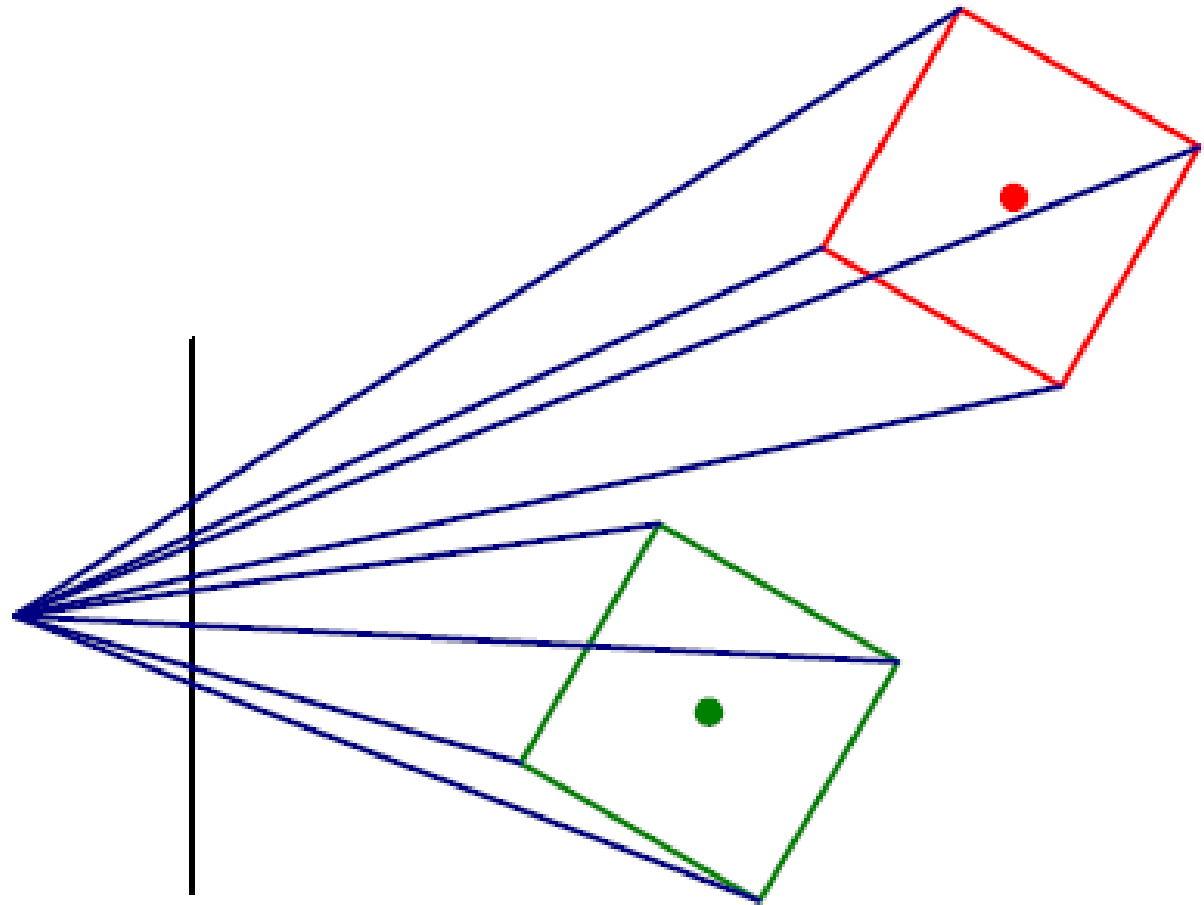


# Kamera mit Objektiv

- Beleuchtetes Objekt strahlt Licht ab
- Sensor ist Lichtempfindlich
- Eine Linse fokussiert das abgestrahlte Licht auf den Sensor

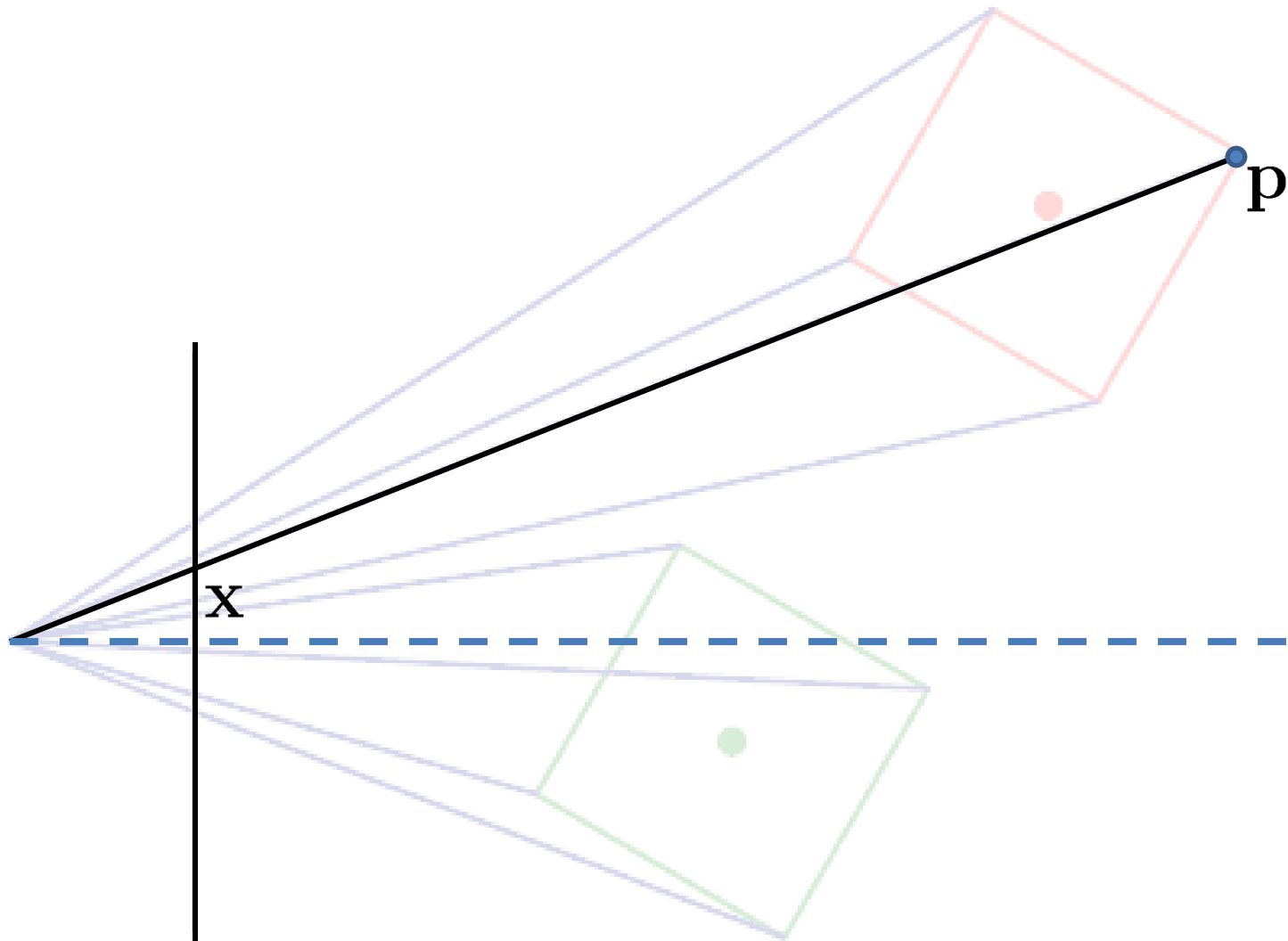


# Perspektivische Projektion (3D auf 2D)

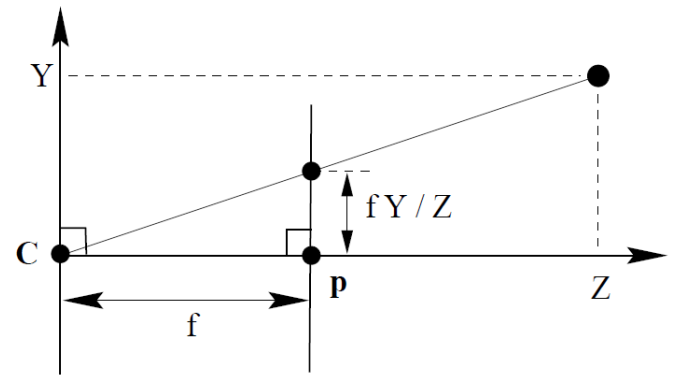
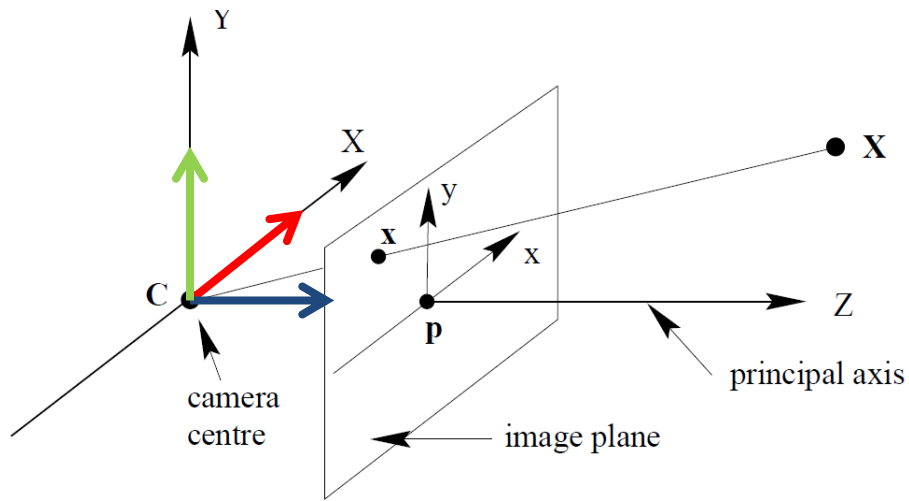




# Perspektivische Projektion (3D auf 2D)



# Kameramodell (Lochkamera)



# Perspektivische Projektion (3D auf 2D)

- 3D-Punkt  $\mathbf{p}$  (in Kamerakoordinaten)
- 2D-Punkt  $\mathbf{x}$  (auf der Bildebene)
- Lochkameramodell

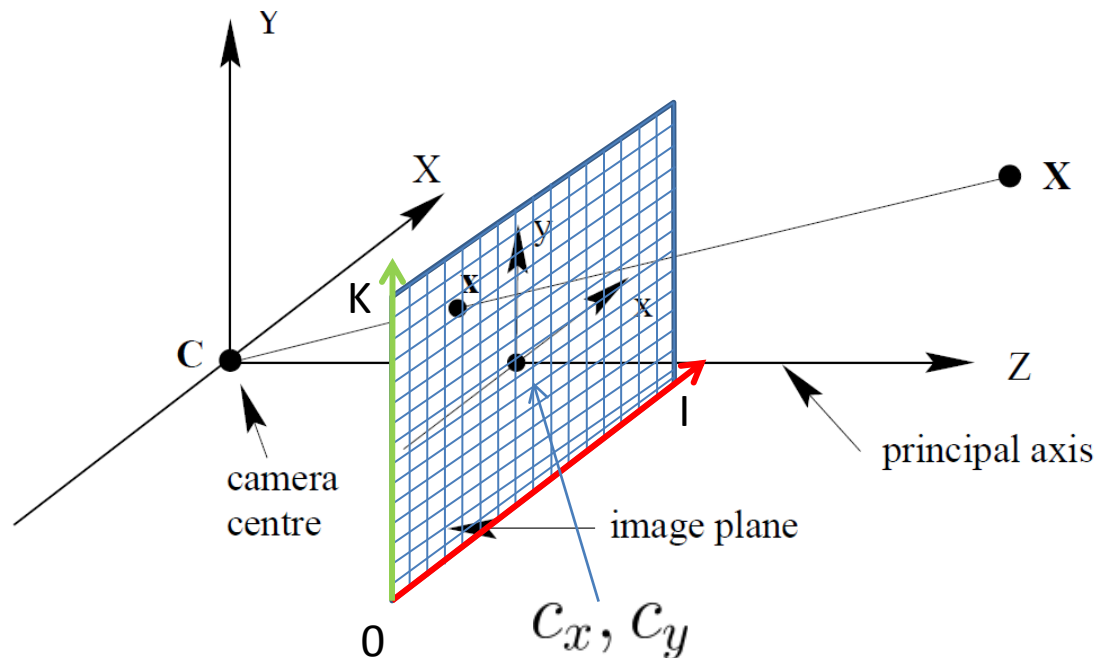
$$\tilde{\mathbf{x}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tilde{\mathbf{p}}$$

- $\tilde{\mathbf{x}}$  in homogenen Koordinaten, muß normalisiert werden

$$\tilde{\mathbf{x}} = \begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{pmatrix} \Rightarrow \mathbf{x} = \begin{pmatrix} \tilde{x}/\tilde{z} \\ \tilde{y}/\tilde{z} \end{pmatrix}$$

# Intrinsische Kameraparameter

- Bis jetzt, 2D Lage des Punktes auf der Bildebene in Meter
- Vom Sensor bekommen wir den Punkt in Pixeln



# Intrinsische Kameraparameter

- Pixelkoordinaten durch Skalierung und Verschiebung

$$\tilde{\mathbf{x}} = \underbrace{\begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}}_{\text{intrinsics } K} \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\text{projection}} \tilde{\mathbf{p}}$$

- Brennweite  $f_x, f_y$
- Bildhauptpunkt  $c_x, c_y$
- Schiefe  $s$

# Extrinsische Kameraparameter

- Für einen Punkt  $\tilde{\mathbf{p}}_w$  gegeben in Weltkoordinaten
- Transformation von Welt in Kamerakoordinaten  
(wird auch als Kamera Extrinsik bezeichnet)

$$\tilde{\mathbf{p}} = \begin{pmatrix} R & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{pmatrix} \tilde{\mathbf{p}}_w$$

- Gesamttransformation inklusive Intrinsik und Projektion

$$\tilde{\mathbf{x}} = \begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} (R \quad \mathbf{t}) \tilde{\mathbf{p}}_w$$

# Reale Kamera mit Objektiv

- Linsenverzerrung durch „nichtideale“ Abbildung (im Vergleich zur Lockkamera)



Rohbild, aufgenommen mit starkem Weitwinkel (sehr kurze Brennweite)



Bild nach der Entzerrung durch den Rechner

# Radiale Verzerrung

- Kreisförmige Verzerrung
  - Hervorgerufen durch nichtideale Linse
  - Vor allem im Randbereich des Bildes
- Wird üblicherweise mit einem Polynom in abhängigkeit des Radius modelliert



$$\hat{x}_c = x_c(1 + \kappa_1 r_c^2 + \kappa_2 r_c^4) \quad \hat{y}_c = y_c(1 + \kappa_1 r_c^2 + \kappa_2 r_c^4)$$

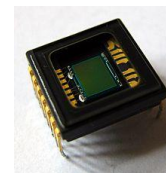
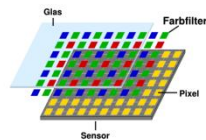


# Bildentstehung

- Objekte in der Szene werden beleuchtet
- Von der Szene reflektiertes Licht wird abgebildet auf ein Helligkeitsmuster auf dem Kamerasensor
- Einflussfaktoren und Störungen
  - Farbe der Beleuchtung
  - Nichtideale Effekte der Linse (Auflösung, Verzerrungen)
  - Auflösung des Sensors (bei Farbe zusätzlich Bayer-Pattern)
  - Signalwandlung mit Rauschen der einzelnen Pixel



Moiré-Effekt

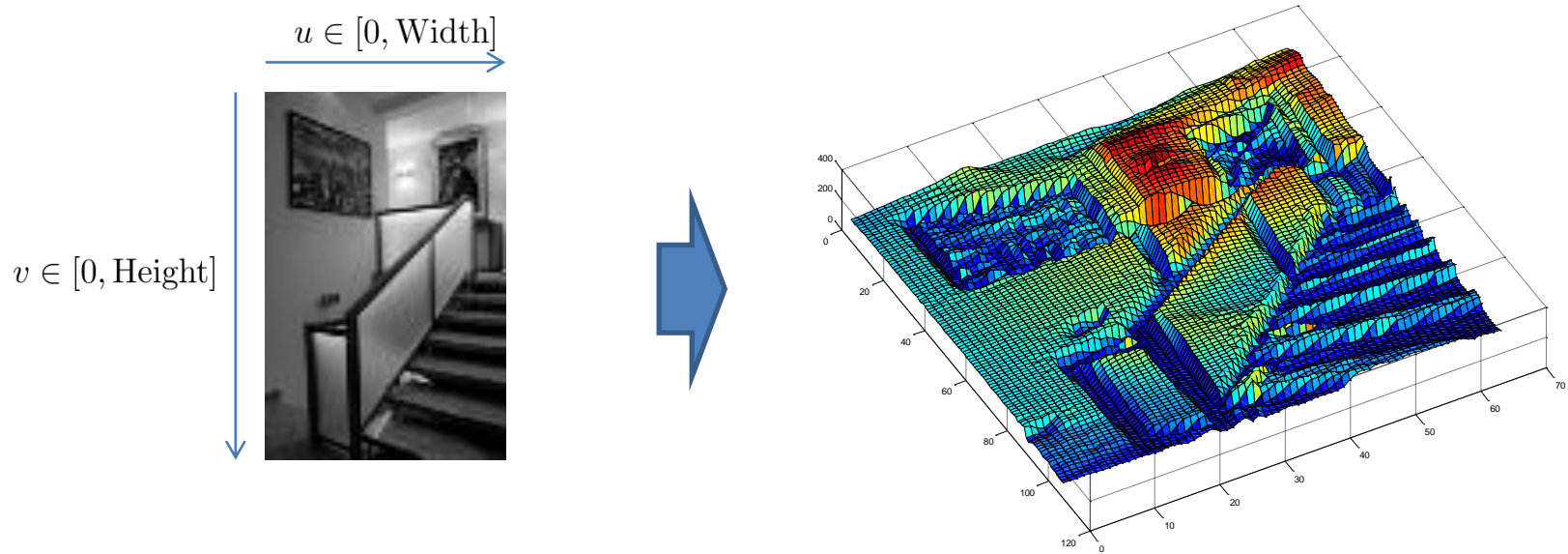


Grundlegendes zur

# **BILDVERARBEITUNG**

# Bild als Funktion des Ortes

- Grauwert des einzelnen Pixels als Funktion  $f(u, v) \in [0, 255]$



- Charakterisierung lokaler Eigenschaften des Bildes über Analyse des Funktionsverlaufs von  $f(u, v)$ 
  - z.B. Kanten als lokale Grauwertänderungen

# Kantenextraktion = Ableitung

- Bildkanten sind Orte großer Steigung der Funktion  $f(u,v)$ 
  - Steigung bzw. Gradient ist richtungsabhängig

$$\frac{\partial f(u,v)}{\partial u}$$



0 = grau

$$\frac{\partial f(u,v)}{\partial v}$$



0 = grau

$$\sqrt{\frac{\partial f^2}{\partial u} + \frac{\partial f^2}{\partial v}}$$



0 = schwarz

# Bildverarbeitung mittels linearer Filter

- Charakterisierung linearer Filter über (vgl. Systemtheorie/Regelungstechnik)
  - Impulsantwort  $h(u,v)$
  - Übertragungsfunktion  $H$
- In der digitalen Signalverarbeitung in der Regel Filter mit endlicher Impulsantwort
  - Filterung durch Faltungsoperation

$$f(u, v) * h(u, v) = \sum_{\tau_1=-\infty}^{\infty} \sum_{\tau_2=-\infty}^{\infty} h(\tau_1, \tau_2) \cdot f(x - \tau_1, y - \tau_2)$$

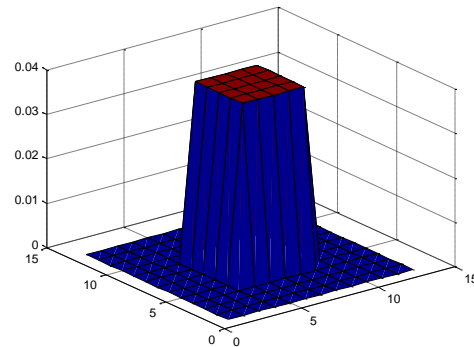
# Filtern von Bildern durch Faltung

- z.B. Glättung durch Faltung mit entsprechendem Kern

$$f(x, y)$$



$$h(x, y)$$



Const-Kern: Größe 13x13,  $\delta=2$

$$f(x, y) * h(x, y)$$



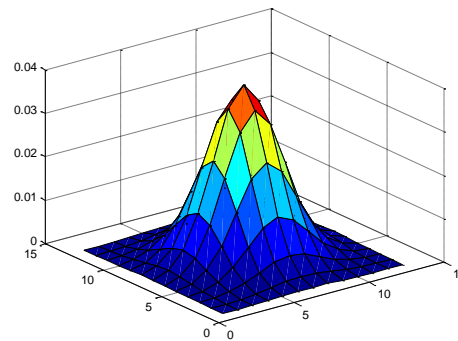
# Filtern von Bildern durch Faltung

- z.B. Glättung durch Faltung mit entsprechendem Kern

$$f(x, y)$$



$$h(x, y)$$



Gauß-Kern: Größe 13x13,  $\delta=2$

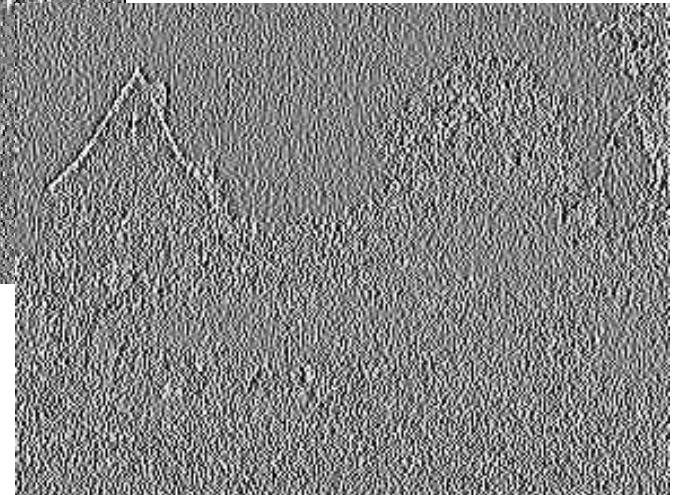
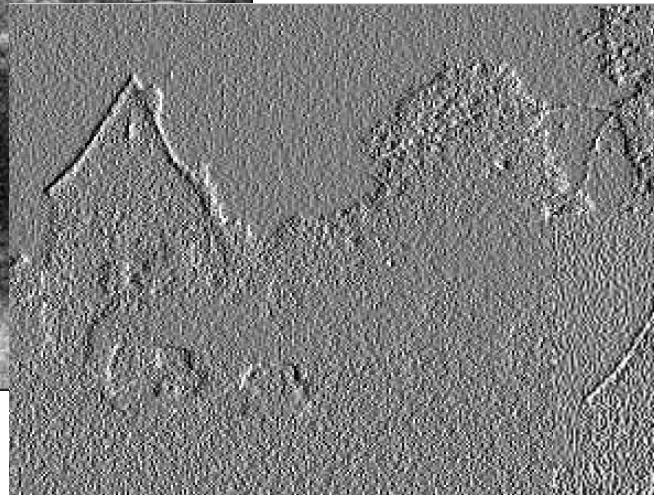
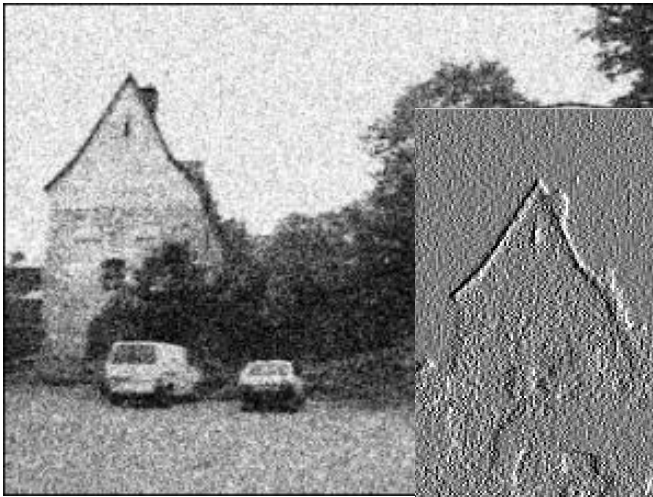
$$f(x, y) * h(x, y)$$





# Gauß'sche Richtungsableitungen

- Differentiation verstärkt Rauschen

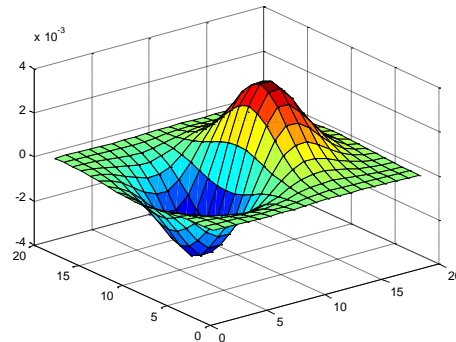
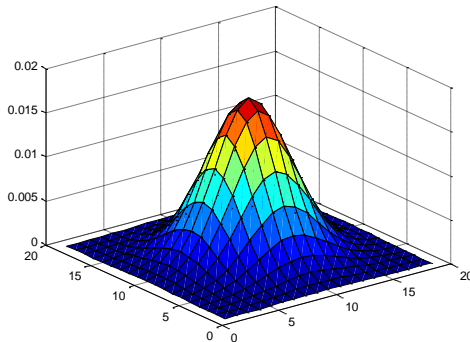




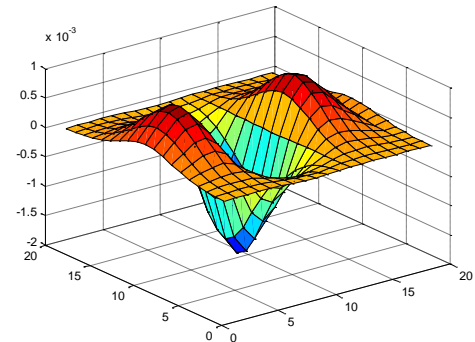
# Gauß'sche Richtungsableitungen

- Erst glätten, dann ableiten
  - Entspricht Faltung mit einem abgeleiteten Glättungskern

$$\frac{\partial}{\partial x}(f(x, y) * h(x, y)) = f(x, y) * \frac{\partial}{\partial x}h(x, y)$$



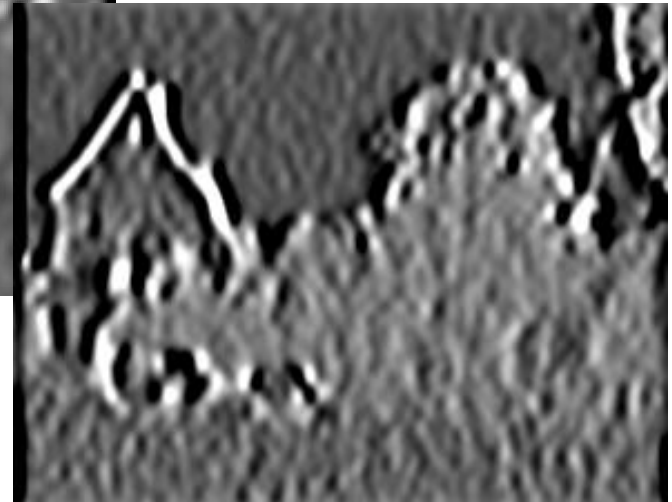
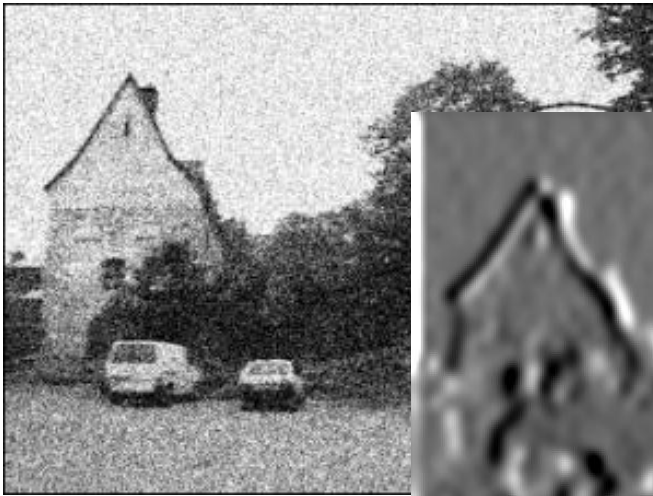
1. Ableitung



2. Ableitung

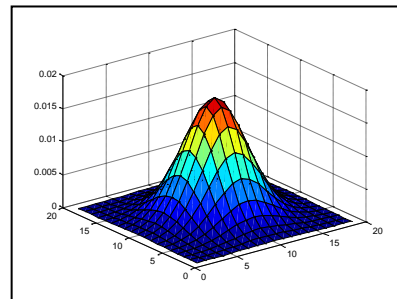
# Gauß'sche Richtungsableitungen

- Glättungsanteil des Filters dämpft Rauschen

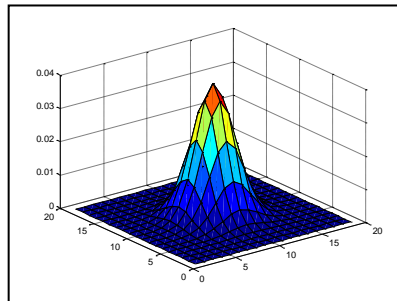


# Detailgrad wählbar

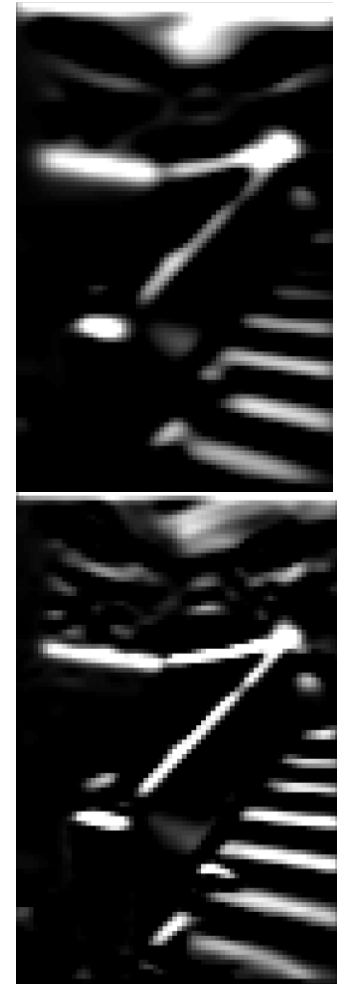
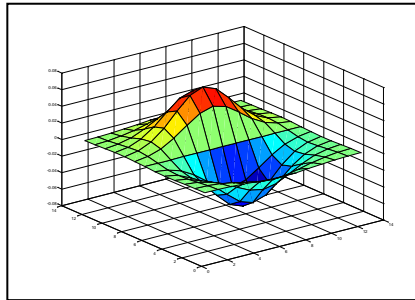
- Varianz der Gauß-Funktion



$$\sigma = 3$$



$$\sigma = 2$$



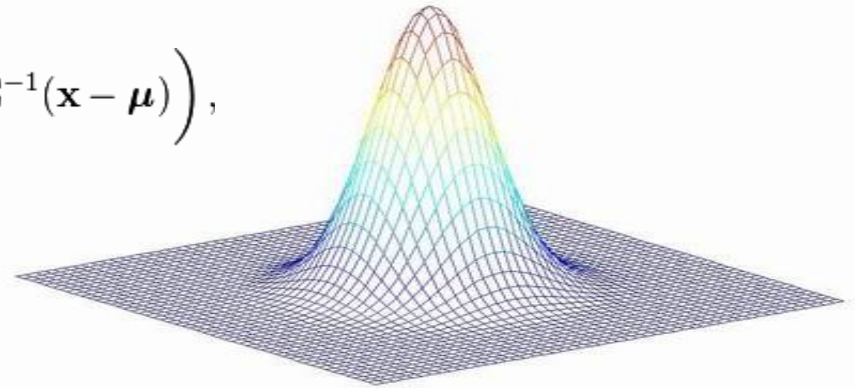
# Merkmalsextraktion

- Lokale Eigenschaften des Grauwertverlaufs können über Gradienten erfasst werden
  - Kantendetektion
  - Eckendetektion (aka interest points)
- In aller Regel wird man das Bild zunächst zur Rauschunterdrückung glätten und dann erst differenzieren
  - Implementierung über Gauß'sche Richtungableitungen
  - Wahl des Detailgrads über Varianz des Gauß-Kerns
- Auch interessant: Skalenraum (Scalespace) nach Lindeberg

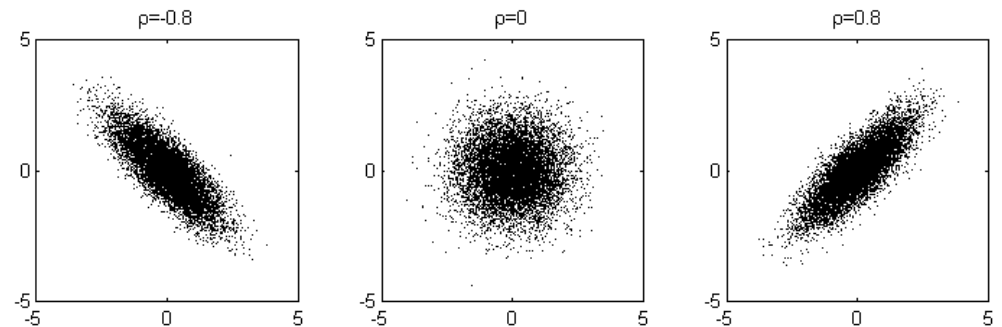


# Mehrdimensionale Gaußfunktion

$$f_{\mathbf{x}}(x_1, \dots, x_k) = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} \exp \left( -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right),$$



$$f(x, y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \exp \left( -\frac{1}{2(1-\rho^2)} \left[ \frac{(x-\mu_x)^2}{\sigma_x^2} + \frac{(y-\mu_y)^2}{\sigma_y^2} - \frac{2\rho(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y} \right] \right),$$



Grundlegendes zur

# **MUSTERERKENNUNG**

# Die Axiome der Wahrscheinlichkeitsrechnung

Notation:  $P(A)$  ist die Wahrscheinlichkeit dafür, dass die Aussage  $A$  gilt.

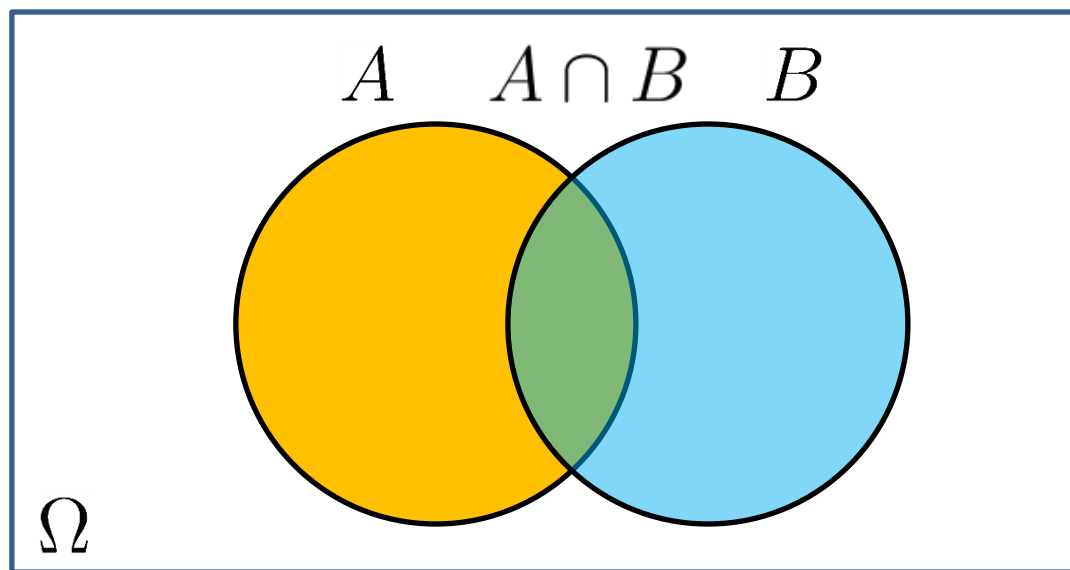
1.  $0 \leq P(A) \leq 1$

2.  $P(\Omega) = 1$        $P(\emptyset) = 0$

3.  $P(A \cup B) = P(A) + P(B) - P(A \cap B)$

## Axiom 3

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$





# Diskrete Zufallsvariablen

- $X$  ist eine **Zufallsvariable**
- $X$  nimmt eine abzählbare Menge von Werten aus  $\{x_1, x_2, \dots, x_n\}$  an
- $P(X = x_i)$  ist die **Wahrscheinlichkeit**, dass die Zufallsvariable  $X$  den Wert  $x_i$  annimmt
- Beispiel:  $P(\text{Room}) = \langle 0.7, 0.2, 0.08, 0.02 \rangle$   
 $\text{Room} \in \{\text{office, corridor, lab, kitchen}\}$

# Kontinuierliche Zufallsvariablen

- $X$  nimmt kontinuierliche Werte an
- $p(X = x)$  bzw.  $p(x)$  ist eine

**Wahrsscheinlichkeitsdichtefunction (PDF)**

$$P(x \in [a, b]) = \int_a^b p(x) dx$$

# Wahrscheinlichkeitsverteilungen integrieren sich zu 1

- Diskreter Fall

$$\sum_x P(x) = 1$$

- Kontinuierlicher Fall

$$\int p(x) dx = 1$$

# Verbundverteilung

- $P(X = x \text{ and } Y = y) = P(x, y)$
- $P(x \mid y)$  ist die bedingte Wahrscheinlichkeit von **x gegeben y**

$$P(x \mid y)P(y) = P(x, y)$$

- Sind  $X$  und  $Y$  **statistisch unabhängig**, dann gilt

$$P(x, y) = P(x)P(y)$$

$$P(x \mid y) = P(x)$$

# Bedingte Unabhängigkeit

- Definition der bedingten Unabhängigkeit

$$P(x, y \mid z) = P(x \mid z)P(y \mid z)$$

- Dies entspricht  $P(x \mid z) = P(x \mid y, z)$

$$P(y \mid z) = P(y \mid x, z)$$

- **Achtung**, dies impliziert nicht  $P(x, y) = P(x)P(y)$

# Randverteilung

- Diskreter Fall

$$P(x) = \sum_y P(x, y)$$

- Kontinuierlicher Fall

$$p(x) = \int p(x, y) dy$$

# Randverteilung

	$x_1$	$x_2$	$x_3$	$x_4$	$p_Y(Y) \downarrow$
$y_1$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{32}$	$\frac{1}{4}$
$y_2$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{32}$	$\frac{1}{32}$	$\frac{1}{4}$
$y_3$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{4}$
$y_4$	$\frac{1}{4}$	0	0	0	$\frac{1}{4}$
$p_X(X) \rightarrow$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{8}$	1

# Satz von der totalen Wahrscheinlichkeit

- Diskreter Fall

$$\begin{aligned} P(x) &= \sum_y P(x, y) \\ &= \sum_y P(x \mid y) P(y) \end{aligned}$$

- Kontinuierlicher Fall

$$\begin{aligned} p(x) &= \int p(x, y) \mathrm{d}y \\ &= \int p(x \mid y) p(y) \mathrm{d}y \end{aligned}$$



# Grundsätzliches zur Mustererkennung

- In dieser Vorlesung: Statistische Verfahren
- Aufgabe: Automatische Klassifikation von Objekten
- Beispiel:
  - Zwei Bäcker stellen Brötchen her, die sich leicht unterscheiden
  - Frage: Können wir an einem der Brötchen erkennen, welcher Bäcker es hergestellt hat?

# Merkmalsextraktion

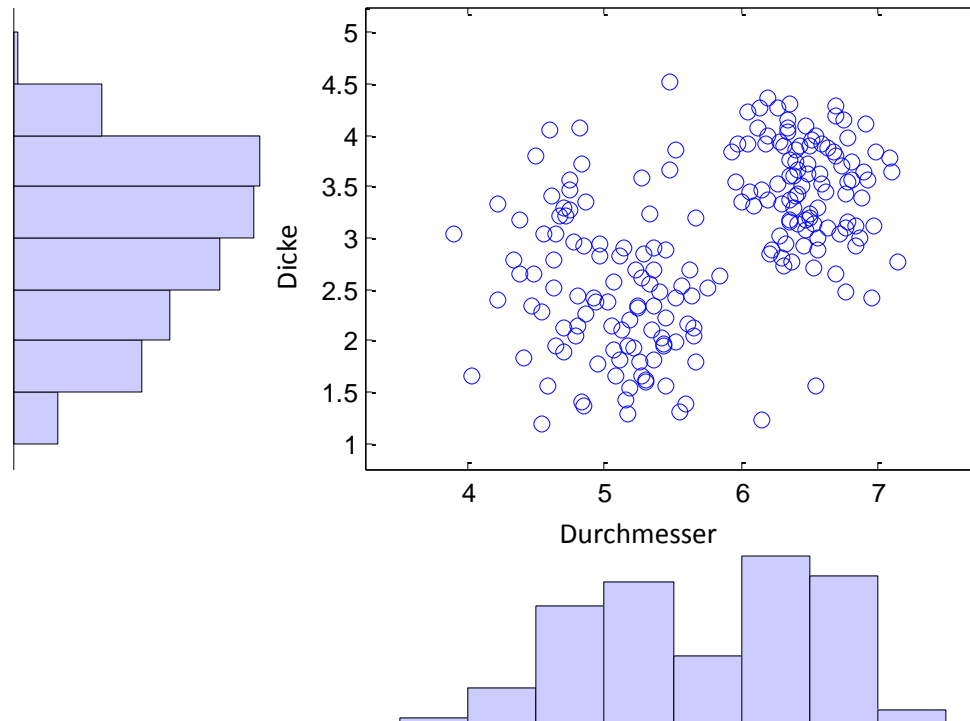
- Merkmale (engl. Features) sind für die Erkennungsaufgabe hilfreiche Zahlenwerte
  - z.B. Dicke und Durchmesser der Brötchen

6,63027181282151	3,53867504726895
6,88686173034205	3,92131040308234
6,88430332022434	3,54949976124462
6,36520879057940	4,02253643322049
6,44322620306702	3,66220939079799
6,57727566517259	4,24876258223907
7,02477397855060	3,38155395317036
6,78444022562949	2,55339370408865
5,98173295133781	3,93233196957361
6,26804193710415	3,30271261332149
6,14414738063608	3,52581774912581

	Dicke	Durchmesser	Varianz Dicke	Varianz Durchmesser
Backer 1	3.5	6.5	0.1	0.3
Bäcker 2	2.5	5	0.2	0.5

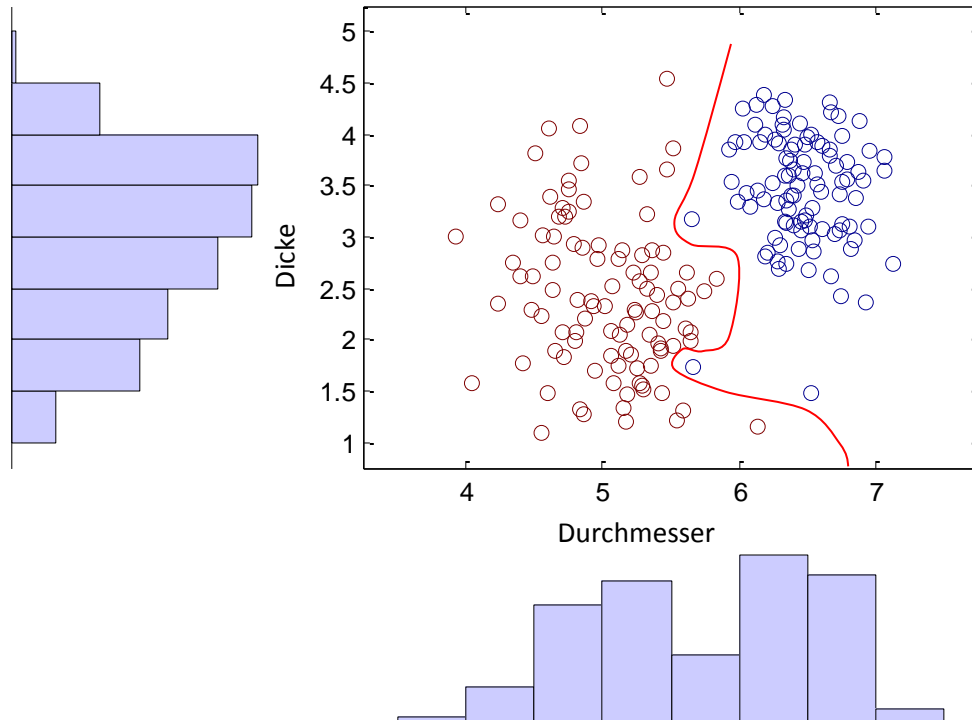
# Mustererkennung anhand der Merkmalsverteilungen

- Im Merkmalsraum wird jedes Objekt zu einen Punkt
- Ähnliche Objekte liegen im Merkmalsraum nahe beieinander
- Sichere Erkennung anhand der einzelnen Merkmale meist unmöglich



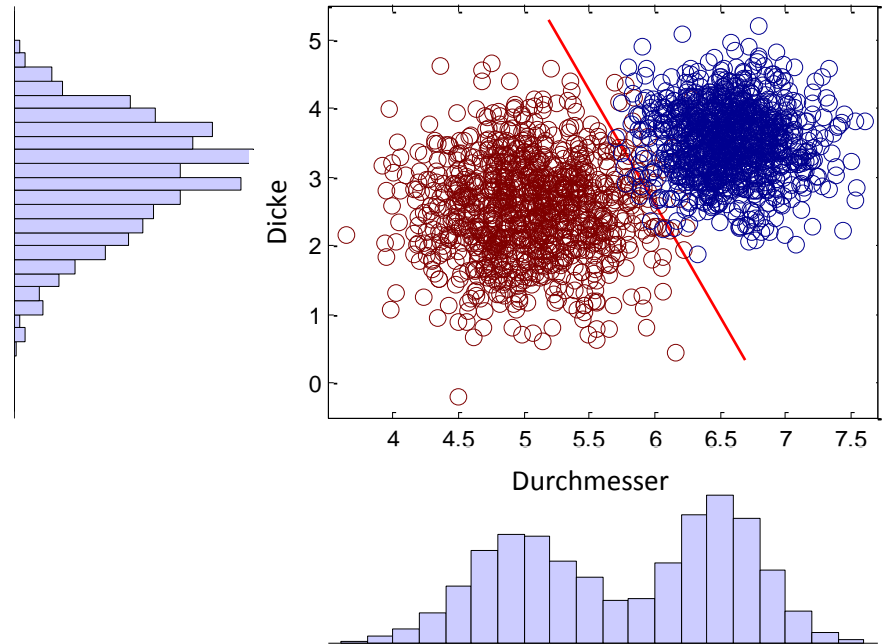
# Zwei grundsätzliche Ansätze

- **Diskriminative** Methoden: Erlernen der Klassengrenzen im Merkmalsraum
- **Generative** Methoden: Auswertung der klassenspezifischen Merkmalsverteilungen



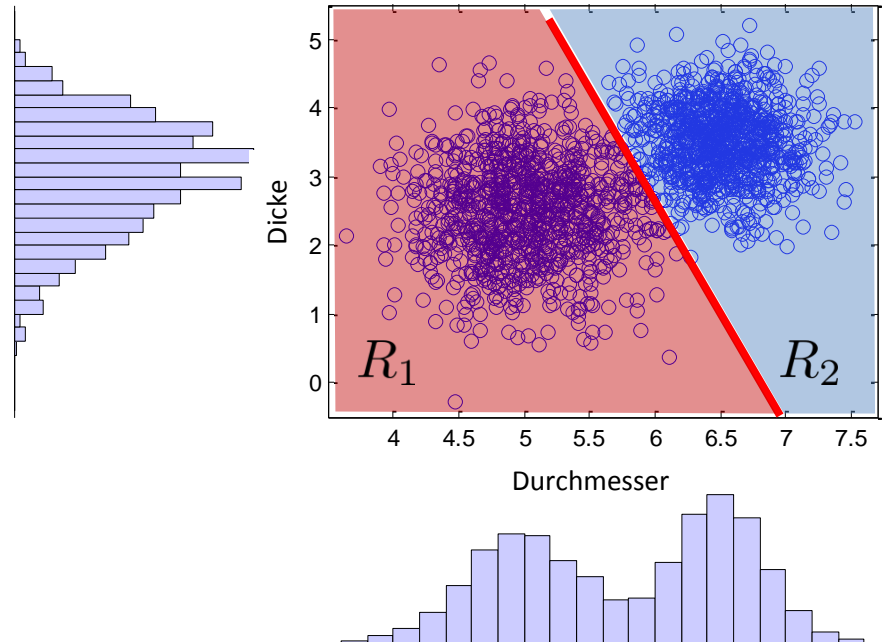
# Fehlerlose Klassifikation meist unmöglich

- 1000 Samples aus derselben Verteilung
  - Man sieht, die Klassengebiete sind überlappend!



# Entscheidungstheorie

- Klassifikation auf Basis der Merkmale
  - Hier im Beispiel: Dicke/Durchmesser
- Was ist die optimale Entscheidungsregel?
- Entscheidungsregionen  $R_n$ : Wir entscheiden uns für Klasse  $C_n$



# Entscheidungstheorie

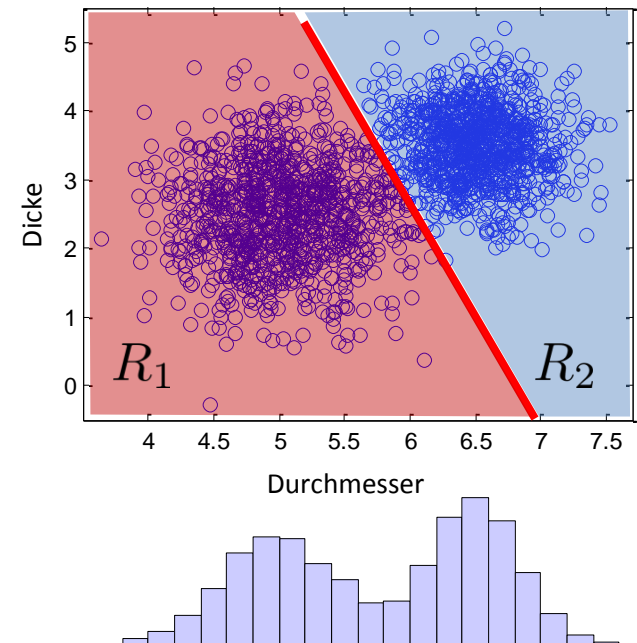
- Was ist die optimale Entscheidungsregel?
  - Minimierung der Wahrscheinlichkeit falscher Klassifikation

$$p(\text{falsch}) = p(\mathbf{x} \in R_1, C_2) + p(\mathbf{x} \in R_2, C_1)$$

$$= \underbrace{\int_{R_1} p(\mathbf{x}, C_2) d\mathbf{x}}_{\text{Minimierung der Wahrscheinlichkeit dass das Objekt aus der anderen Klasse kommt}} + \underbrace{\int_{R_2} p(\mathbf{x}, C_1) d\mathbf{x}}_{\text{Für alle Klassen gleich}}$$

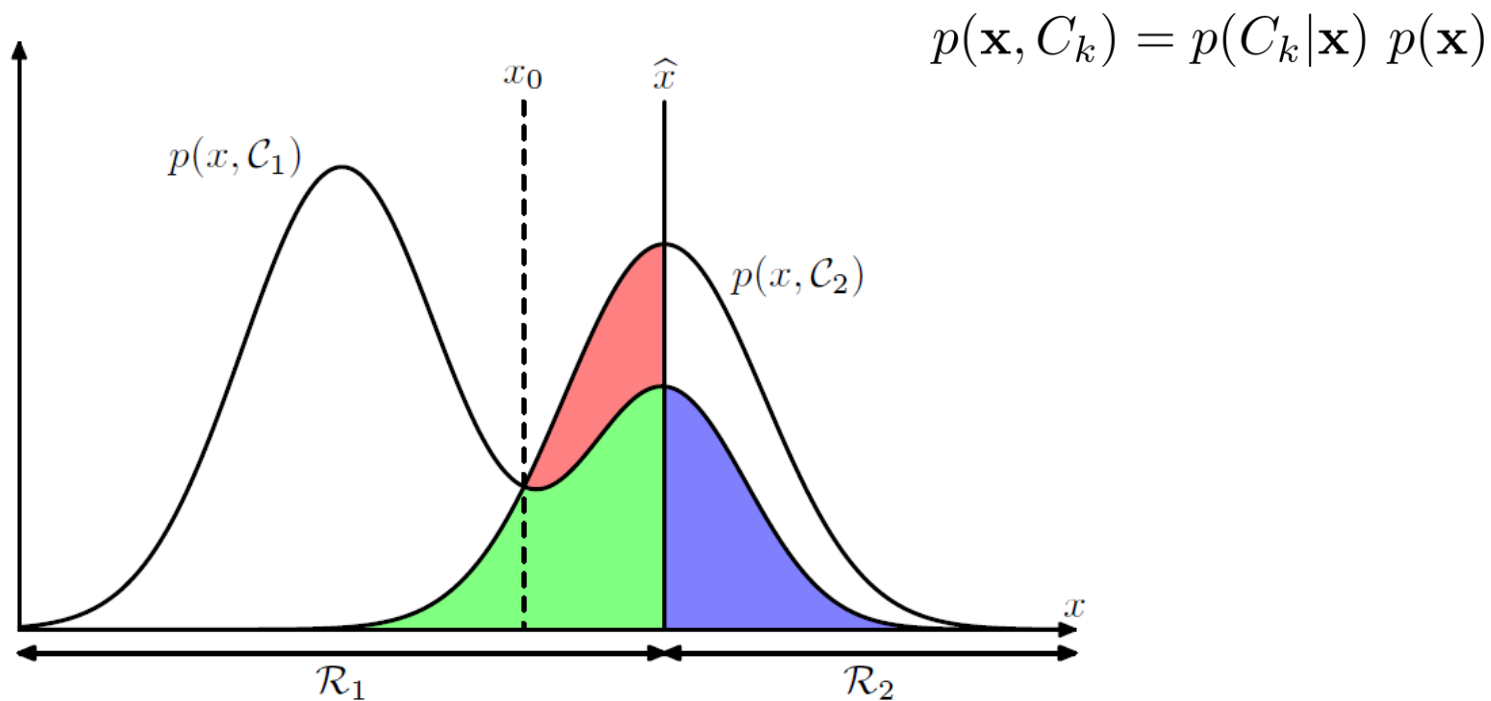
$$p(\mathbf{x}, C_x) = p(C_x|\mathbf{x}) \underbrace{p(\mathbf{x})}_{\text{Optimale Entscheidung: Klasse mit der maximalen aposteriori-Wahrscheinlichkeit}}$$

Optimale Entscheidung: Klasse mit der maximalen aposteriori-Wahrscheinlichkeit



# Entscheidungstheorie


- Wir entscheiden für  $\hat{C} = \operatorname{argmax}_k p(\mathbf{x}, C_k) = \operatorname{argmax}_k p(C_k|\mathbf{x})$





# Risikominimierung

- Risiko = „Kosten für Fehler“ \* Eintrittswahrscheinlichkeit

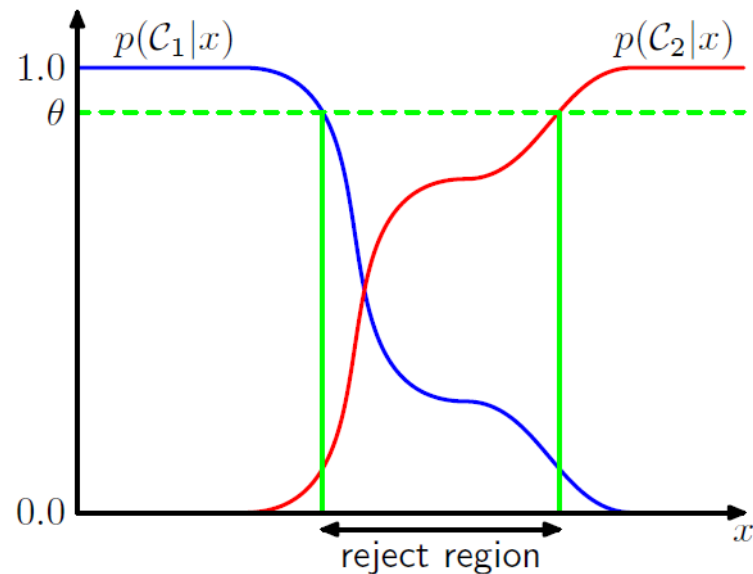
– Kostenmatrix  $L_{i,k}$  

	cancer	normal
cancer	0	1000
normal	1	0

- Durch Entscheidung  $\sum_k L_{kj} p(C_k | \mathbf{x})$  minimieren!
- Kosten „verschieben“ die Entscheidung

# Rückweisung

- Vermeiden von Klassifikationsfehlern in mehrdeutigen Situationen
  - Nur entscheiden, wenn die maximale a posteriori-Wahrscheinlichkeit größer ist, als eine vordefinierte Schwelle  $\theta$



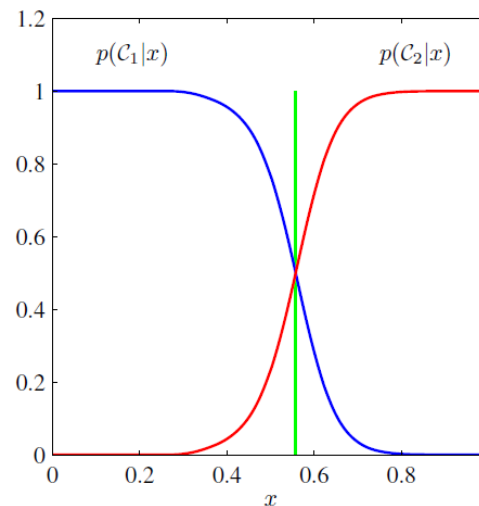
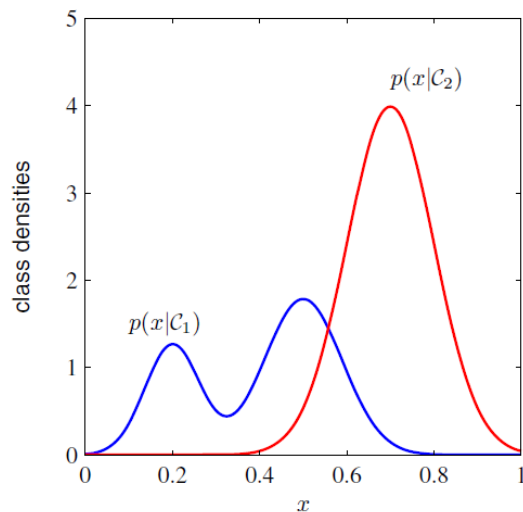
# Inferenz und Entscheidung

- Klassifikation besteht aus zwei Schritten
  - Inferenz: Bestimmen von  $p(C_k|\mathbf{x})$  aus Trainingsdaten
  - Entscheidung:  $\hat{C} = \operatorname{argmax}_k p(C_k|\mathbf{x})$

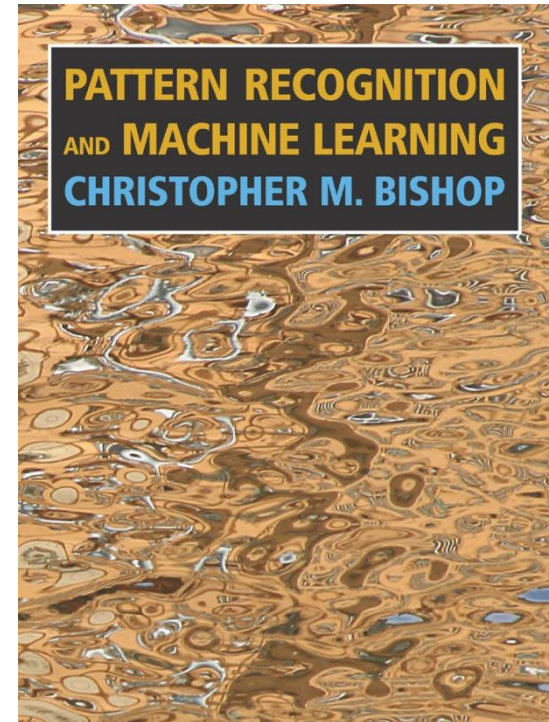
$$p(\mathbf{x}, C_k) = p(C_k|\mathbf{x}) p(\mathbf{x}) \qquad p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k) p(C_k)}{p(\mathbf{x})}$$



Satz von Bayes!!



# Buchempfehlung



Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 2009

<http://research.microsoft.com/en-us/um/people/cmbishop/PRML/index.htm>