
15. Computer Vision

Prof. Dr. Rudolf Kruse

University of Magdeburg

Faculty of Computer Science

Magdeburg, Germany

rudolf.kruse@cs.uni-magdeburg.de



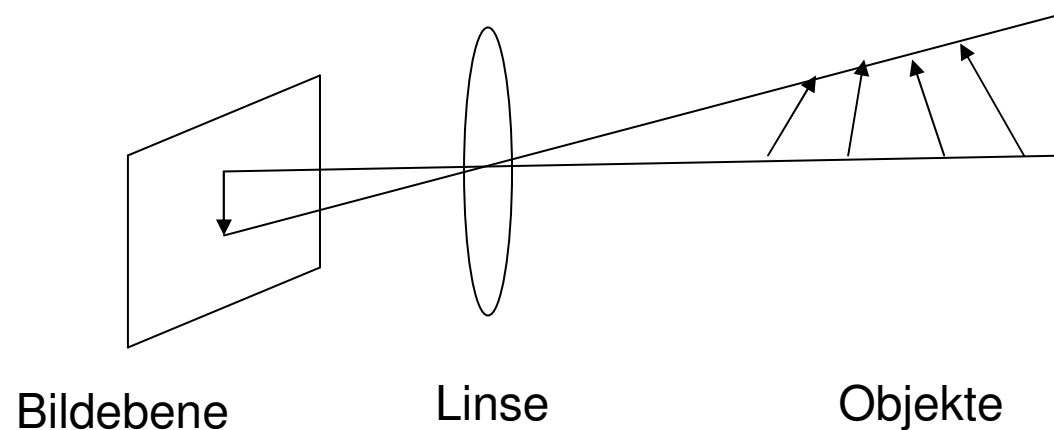
Computer Vision

- Idee: **Dem Rechner beibringen zu sehen.**
- Prinzipielle Vorgehensweise:
 - (i) **Bild(er) der Szene erstellen**
 - (ii) **Szene identifizieren: Merkmalsextraktion**

Bilder der Szene erstellen

■ Probleme:

- Perspektivische Projektion (Kamera), d.h verschiedene Szenen können das gleiche Bild erzeugen,
- das Bild kann verwechselt sein,
- unterschiedliche Beleuchtungen,
- verdeckte Objekte,
- Ambiguitäten,
- ...



Merkmalsextraktion

- Identifikation von Objekten in der Szene, z.B.:
 - Ort, Art der Objekte, Begrenzungen (bei Robotern)
 - Ort, Größe, Form (bei Objektmanipulation)
 - ...



Beispiel: Automobil steuern

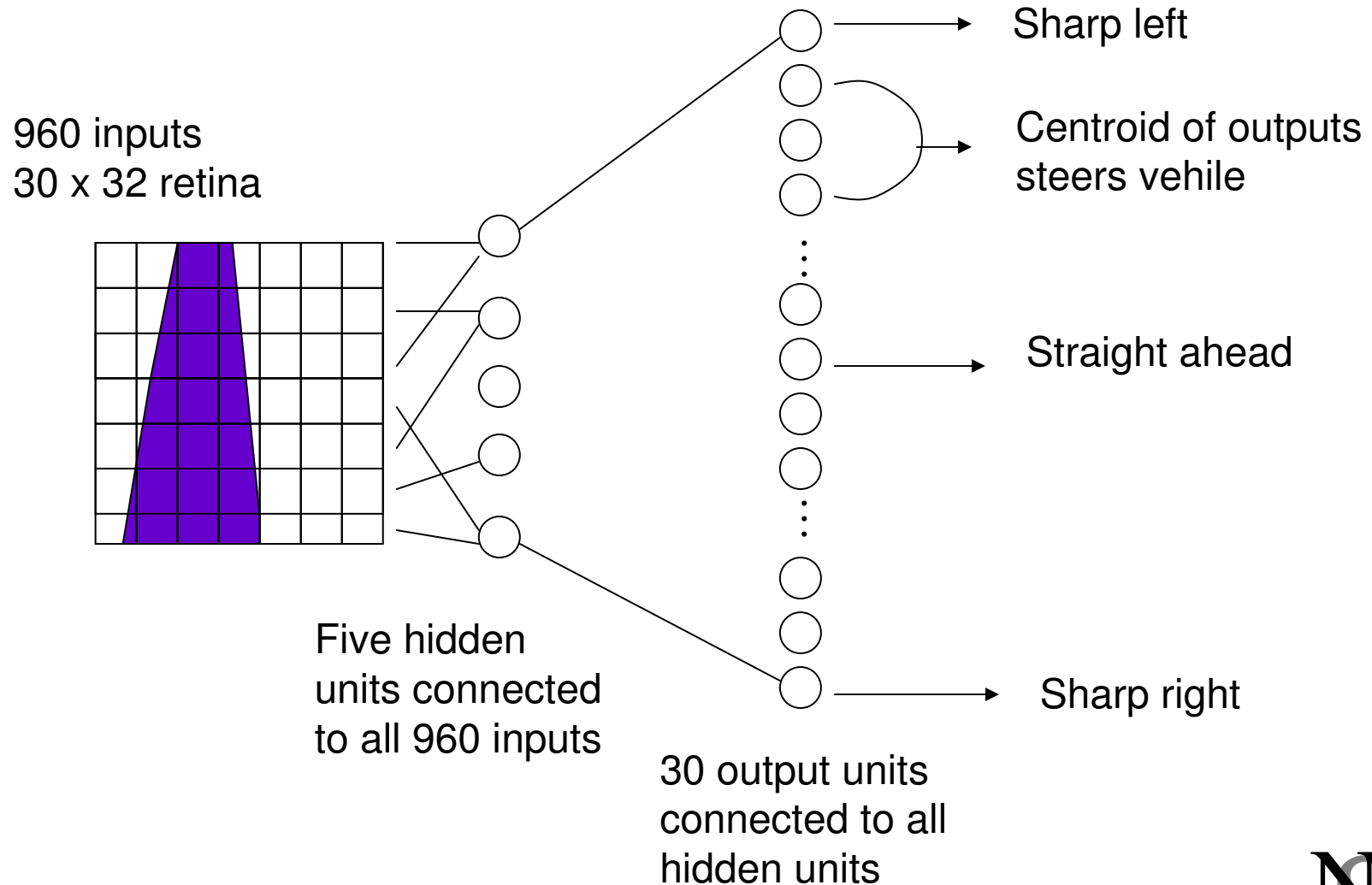
ALVINN Netz

(Autonomous Land Vehicle in a Neural Network)

- **Neurales Netz (Multilayerperzeptron) zur Abbildung von Bildinformationen auf Steueraktion**
- **TV Kamera auf Auto in Richtung nach vorn**
- **960 dimensionale Eingabevektoren (Bilder) werden sukzessive eingegeben**
- **Mensch fährt das Auto, die Lenkwinkel werden zum Trainieren mit Backpropagation genutzt**

Beispiel: Automobil steuern (2)

■ ALVINN Netz



Beispiel: Automobil steuern (3)

- Hauptprobleme des ALVINN Ansatzes:
 - Mensch fährt zu gut, keine Beispiele für Extremsituationen
→ das Bild wird künstlich verändert und zum Trainieren genutzt (problematisch!)
- http://www.ri.cmu.edu/projects/project_160.html

Realistischere Ansätze:

- Auto bereits von München bis Hamburg fahrerlos ohne Unfall gefahren (Komplexerer Ansatz, Dieckmann, TU München, 1990)
- Grand Challenge 2006



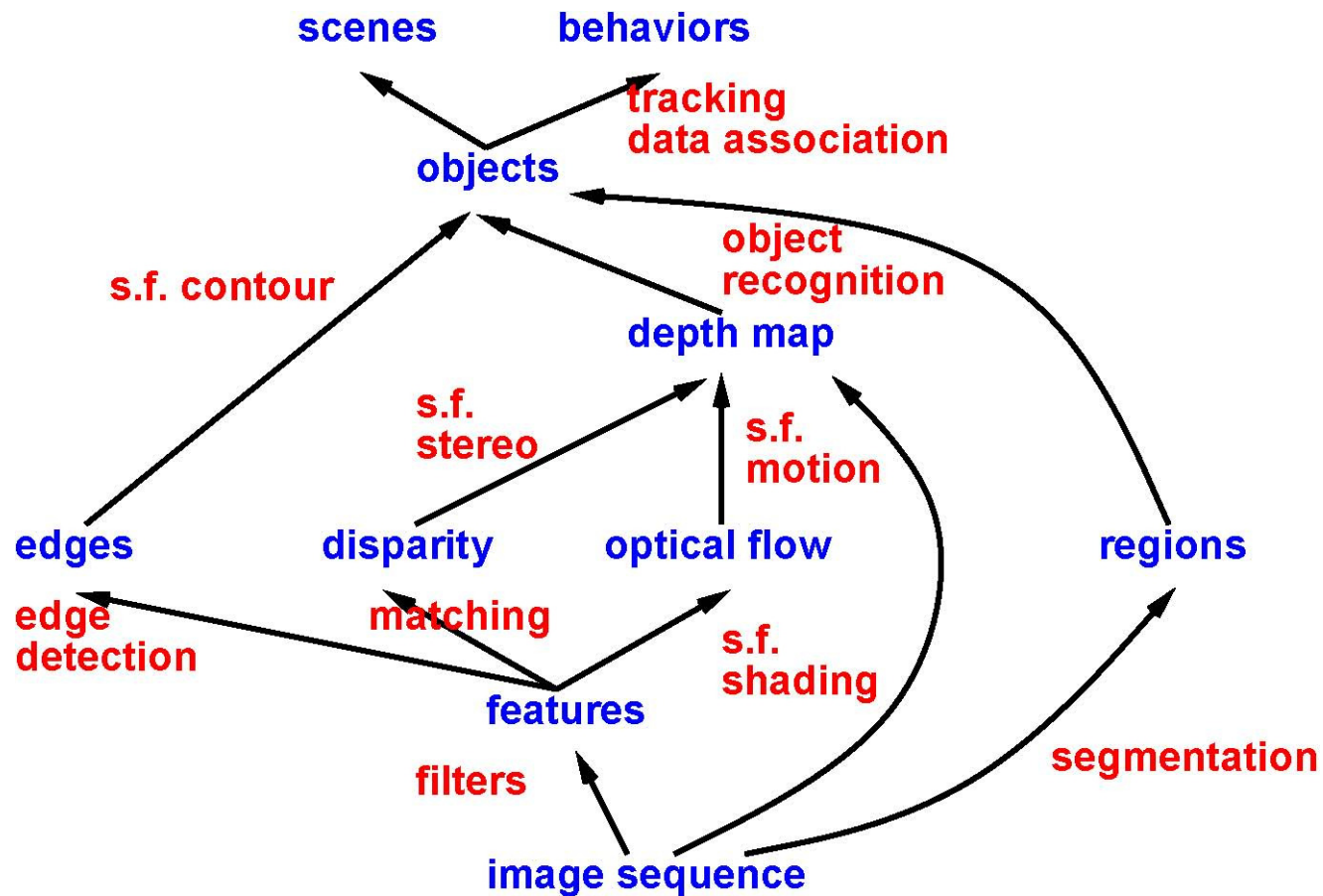
Computer Vision beim „Grand Challenge“

- Ein Team der Universität Stanford hat zusammen mit dem Volkswagen Electronics Research Laboratory (ERL) und seinem auf den Namen "Stanley" getauften VW Touareg die DARPA Grand Challenge 2005 in der Mojave-Wüste gewonnen. Ziel des Wettbewerbs war es, mit autonomen Fahrzeugen einen vorgegeben Parcours von 130 Meilen möglichst schnell zu absolvieren.
- Stanley benötigte insgesamt 6 Stunden, 53 Minuten und 8 Sekunden für die Strecke und war damit 11:42 Minuten schneller als der Zweitplatzierte. Dabei führte die rund 210 Kilometer lange Strecke über holprige Wüstenstraßen, Bergpässe, durch ausgetrocknete Seen und Tunnel.

Die Herausforderungen mussten die autonomen Fahrzeuge nur mit Hilfe von Onboard-Sensoren und dem eingebauten Navigations-Equipment bewerkstelligen - allein, ohne menschliches Eingreifen. Dabei wurden 23 von insgesamt 195 Kandidaten für die Prüfung zugelassen.

- Der Sieger des diesjährigen Wettbewerbs, Stanley, basiert auf einem umgebauten Touareg R5 mit Dieselantrieb, gesteuert mit Hilfe eines "Drive-by-Wire-Systems" von ERL und sechs Rechnern mit Pentium-M-Prozessoren. Zur Orientierung wurde Stanley unter anderem mit **GPS, Laser, Radar und einer Kamera** ausgestattet.

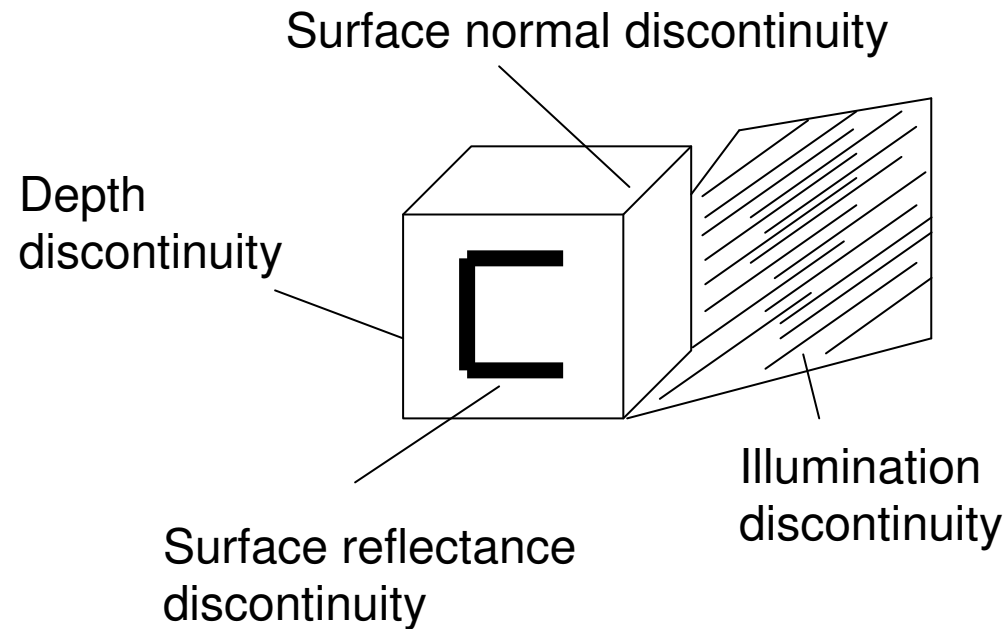
„Subsysteme“ für Vision



Beispiel: Objekte finden

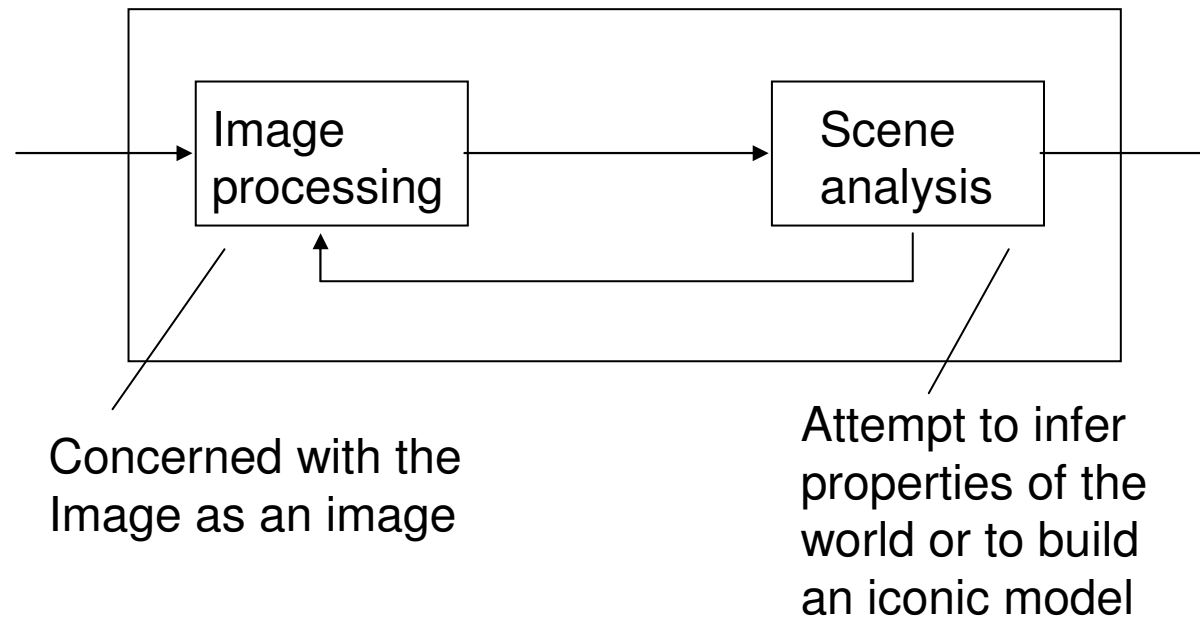
Prinzipieller Ansatz:

- Unstetigkeiten nutzen, um Linien im Bild zu finden und das Bild in Regionen segmentieren zu können



Beispiel: Objekte finden (2)

■ Vorgehensweise:



(vereinfachte Sichtweise, häufig verläuft der Prozess iterativ)

Bildverarbeitungstechniken

- Bildinformationen werden in der Regel in einer $m \times n$ Intensitätsmatrix I gespeichert, wobei n die Anzahl der Zeilen und m die Anzahl der Spalten des diskretisierten Eingabebildes sind:

Image I :

...							
$i_{0,2}$		...					
$i_{0,1}$	$i_{1,1}$						
$i_{0,0}$	$i_{1,0}$	$i_{2,0}$	...				

Intensitätsmatrix

- Die Intensitätsmatrix enthält ganzzahlige Größen, die die Helligkeit der zugehörigen Bildpixel beschreiben.
- Bei Farbbildern wird für jede Grundfarbe eine Intensitätsmatrix benötigt, z.B. beim RGB Farbmodell je eine Matrix für rot, grün und blau.
- Im Folgenden betrachten wir zur Vereinfachung nur Grauwertbilder.

Filteroperationen

- Ein Bild kann durch Filter bearbeiteten werden.

- Hierzu wird eine Filtermatrix $W(i,j)$ definiert, z.B. ein 3x3 Filter:

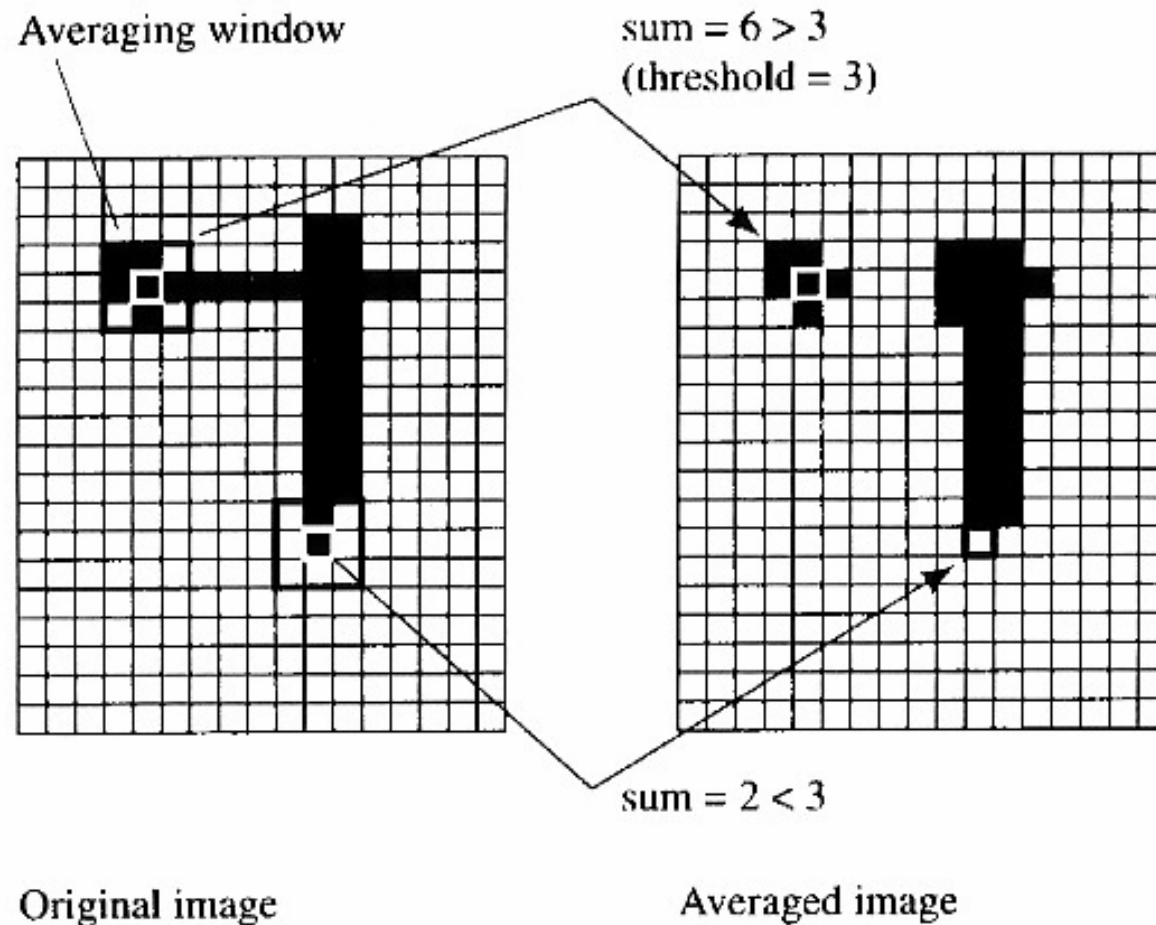
$W_{-1,1}$	$W_{0,1}$	$W_{1,1}$
$W_{-1,0}$	$W_{0,0}$	$W_{1,0}$
$W_{-1,-1}$	$W_{0,-1}$	$W_{1,-1}$

- Filterung: Die Bildintensitätsmatrix $I(x,y)$ wird durch eine diskrete Konvolution (Faltung) mit einem Filteroperator $W(i,j)$ in ein bearbeitetes Bild $I^*(x,y)$ überführt:

wobei $I(x,y)=0$ falls $x \notin \{0,m\} \vee y \notin \{0,n\}$.

$$I^*(x, y) = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} I(u, v) \cdot W(u - x, v - y)$$

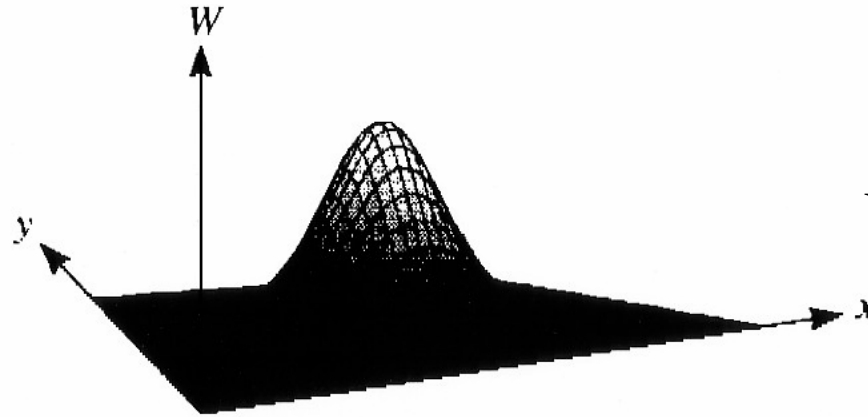
(i) Bildglättung: Schwellwertfilter



Anzahl der schwarzen Pixel in einem 3x3 Fenster werden gezählt
Falls Anzahl der schwarzen Pixel größer 3 → schwarz, sonst weiß

(i) Bildglättung: Gaußscher Filter

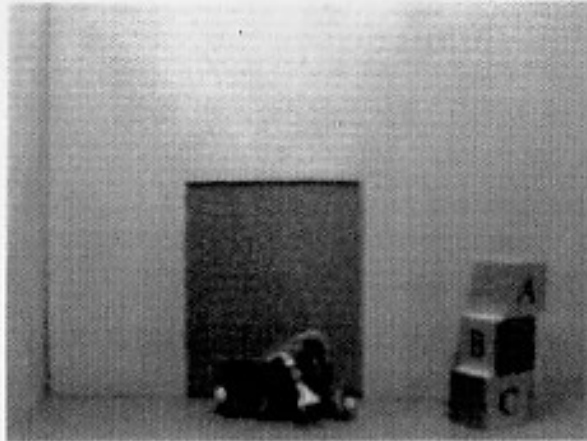
- Ein häufig eingesetzter Glättungsfilter ist der Gaußsche Filter:



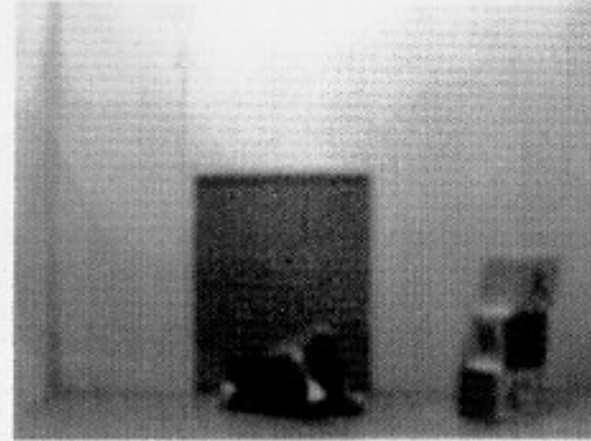
$$W(x, y) = \frac{1}{2\pi\delta^2} e^{-\frac{x^2+y^2}{2\delta^2}}$$

- $W(i,j)$ wird meist am Rand nach einer bestimmten Pixelzahl auf Null gesetzt

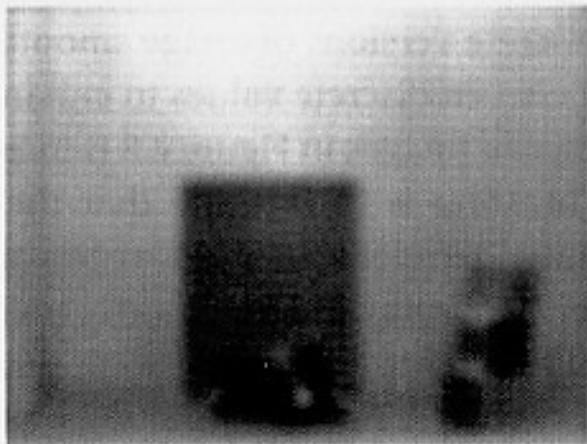
Bildglättung: Beispiele



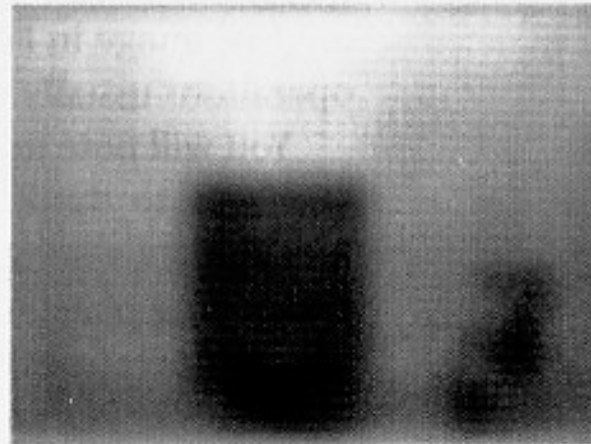
(a) Original image



(b) Width of Gaussian = 2 pixels



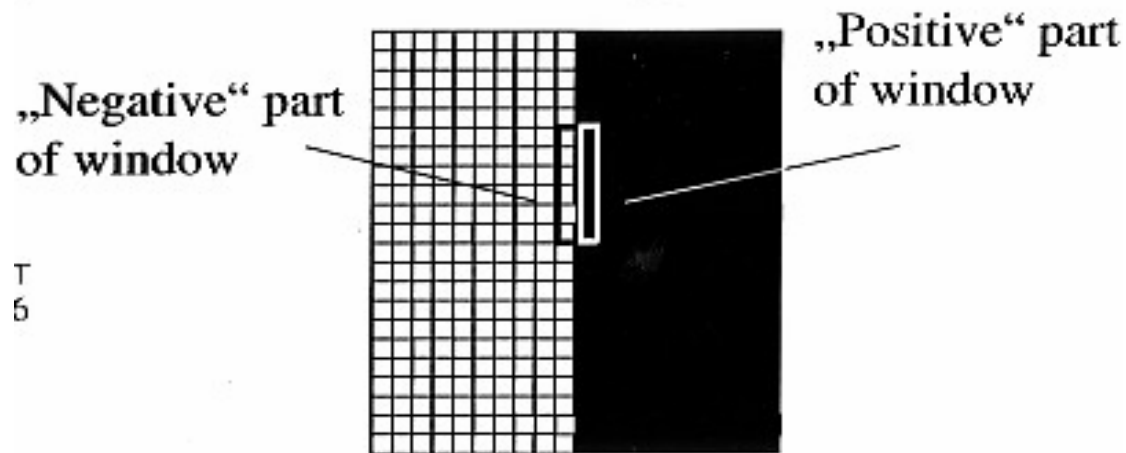
(c) Width of Gaussian = 4 pixels



(d) Width of Gaussian = 8 pixels

(ii) Kantenextraktion

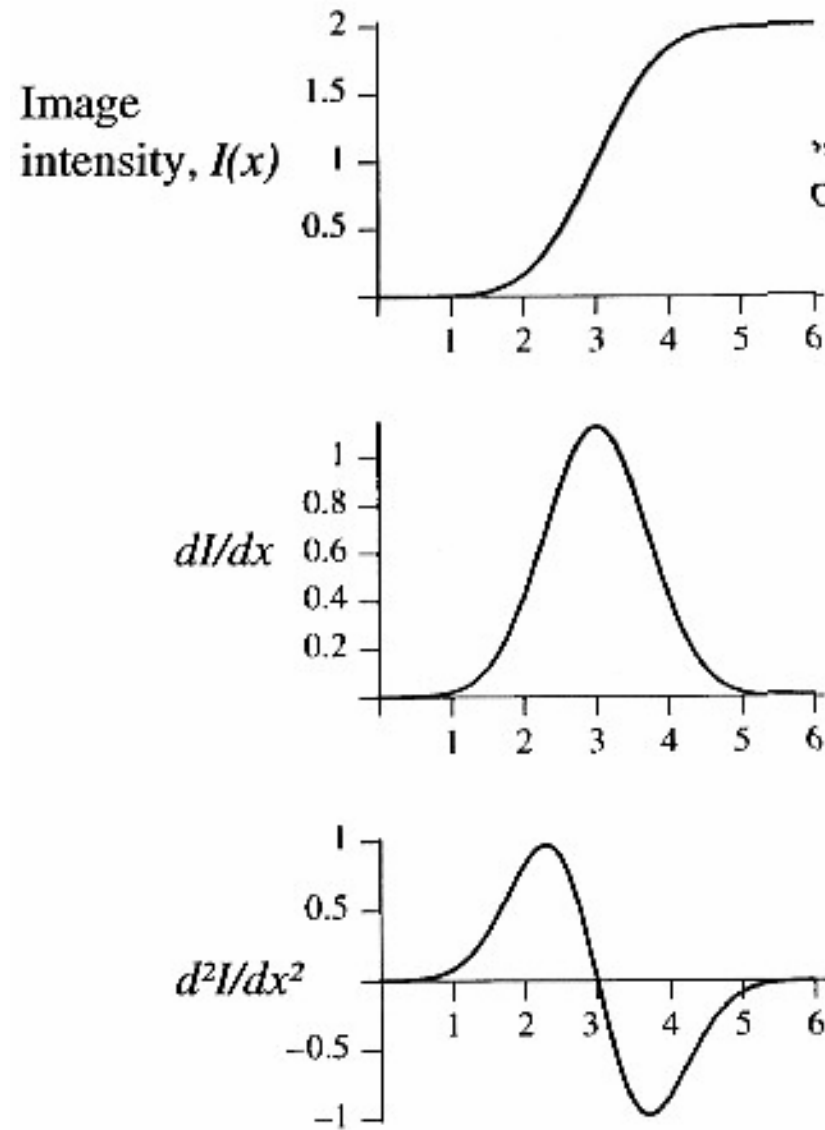
- Zur Kantenerkennung werden Filter benötigt, die Kanten verstärken und andere Bildteile abschwächen oder entfernen.
- Einfacher Filter zum Verstärken vertikaler Kanten:
Die Filtermatrix W besteht aus einem negativen (links) und einem positiven Teil (rechts)



Filter approximiert
Ableitung dl/dx

(ii) Kantenextraktion

- Eine bessere Erkennung (vertikaler) Kanten kann durch die zweite Ableitung erreicht werden
- Vertikale Kanten erkennt man zum Beispiel an Nullstellen der zweiten Ableitung

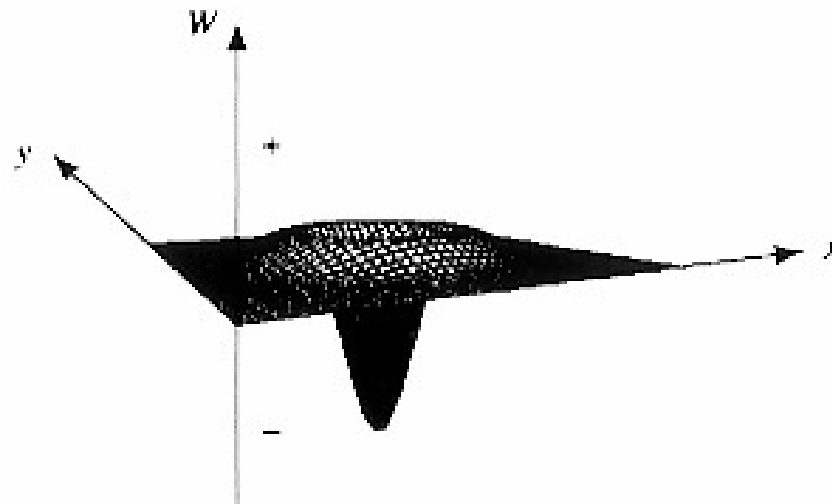


(ii) Kantenextraktion

- Mit Hilfe des Laplace Operators kann man Kanten in beliebigen Richtungen verstärken:

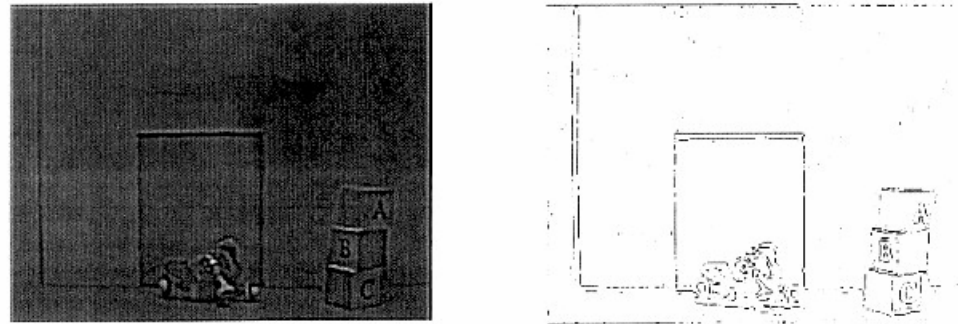
$$\nabla^2 I(x, y) = \frac{\partial^2 I(x, y)}{\partial x^2} + \frac{\partial^2 I(x, y)}{\partial y^2}$$

- Der Sombrero (oder Mexican hat) Filter kombiniert die Gaußsche Glättung mit dem Laplace Filter:



(ii) Kantenextraktion

■ Wirkung der Filteroperation:



■ Anmerkung: Eine Sketch Zeichnung erhält man z.B. durch Sombrero Filterung und anschließende Markierung der Nulldurchgänge (Marr-Hildreth Operator)

(iii) Regionen finden

■ Definition einer Region:

Eine Region ist ein homogener Bildbereich, d.h.

- die Differenz der Intensität der Pixel der Region ist klein bzw.
- eine polynomiale Oberfläche k -ten Grades kann die Intensitätswerte der Region mit kleinen Fehlern approximieren

und keine benachbarten Regionen können zu einer homogenen Region „vereinigt“ werden.

■ Neben den Intensitätswerten können auch Merkmale aus komplexeren Vorverarbeitungs-schritten betrachtet werden (z.B. Texturen wie Gras oder Haut).

Teile-Verschmelze-Methode

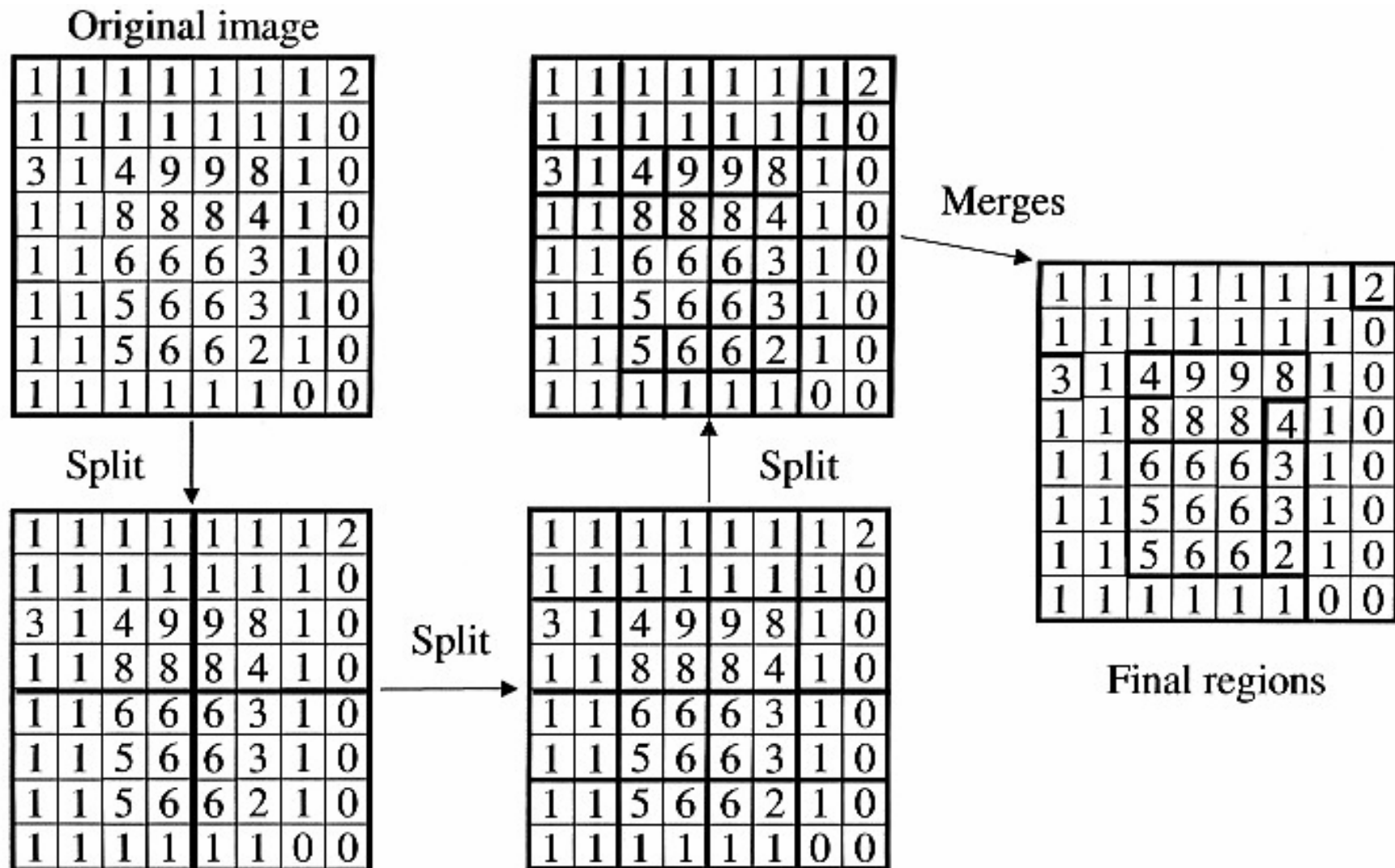
Zur Vereinfachung hier betrachtet:

Bilder in quadratischer $2^l \times 2^l$ Intensitätsmatrix

Regionen Bestimmung:

- **Intensitätsmatrix wird solange immer in zwei gleiche Teile geteilt (horizontal und vertikal) bis die Intensitäten in den Teilregionen sich nur noch um eine Einheit unterscheiden.**
- **Anschließend bei 0 beginnend benachbarte Regionen Verschmelzen, die sich um weniger als eine Einheit unterscheiden.**

Teile-Verschmelze-Methode (2)



Szenen Analyse

Ziel:

- **Informationen über den Inhalt eines Bildes extrahieren.**

Voraussetzungen:

- **Bildinformationen aus Vorverarbeitungsschritten sind vorhanden, z.B. Kanten und Regionen sind im Bild markiert.**
- **In der Regel wird außerdem Vorwissen über den Inhalt des Bildes benötigt, z.B. Art der Objekte im Bild (z.B. nur rechteckige Objekte), Art der Beleuchtung, Kameraposition etc.**

Interpretation von Kantenbildern

- **Annahmen:**
 - Bild enthält nur rechteckige Objekte (aus Ebenen)
 - Nicht mehr als 3 Oberflächen überschneiden sich an einem Punkt
- Dann gibt es nur 3 Möglichkeiten, wie sich zwei Ebenen schneiden können:
 - Verdeckung: Eine Ebene verdeckt eine andere
 - „Schneide: Die Schnittebene stammt von zwei sichtbarer Ebenen mit einer konvexen Kante
 - „Falte“: Die Schnittebene stammt von zwei sichtbarer Ebenen mit einer konkaven Kante

Interpretation von Kantenbildern

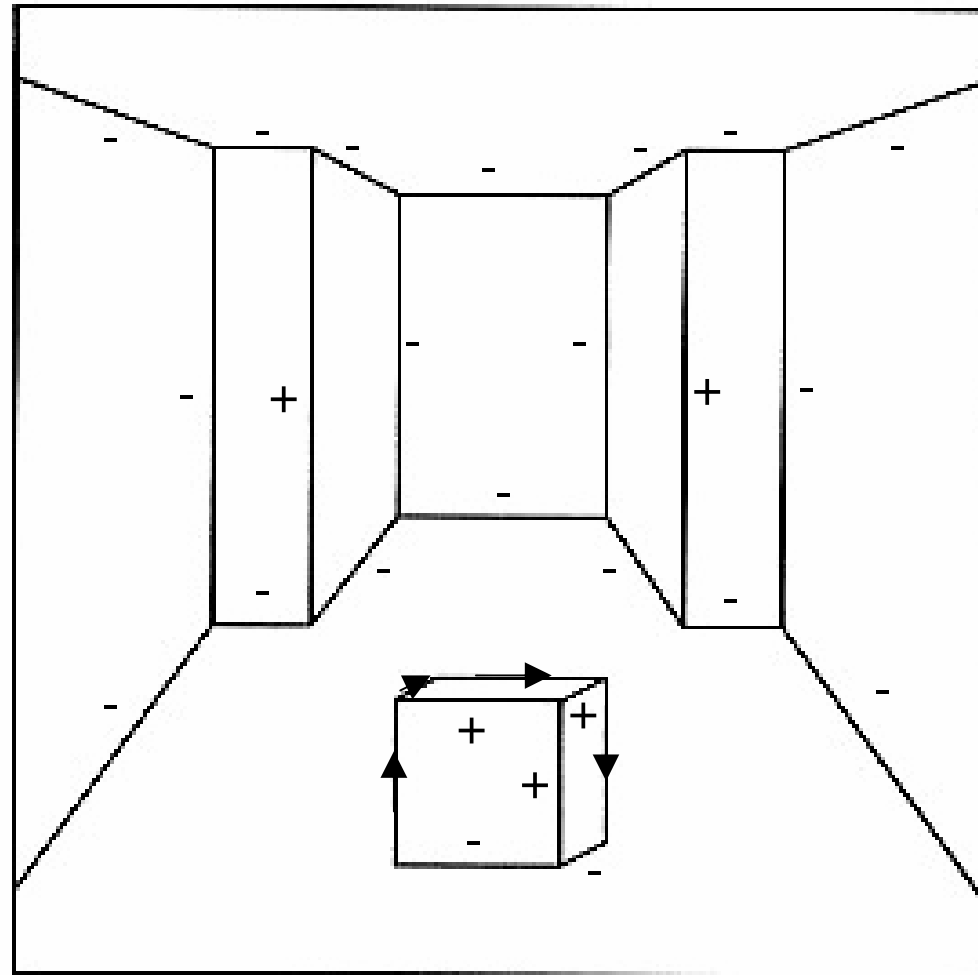
Beispiel

Markierungen:

→ **Verdeckung** (rechts des Pfeils sichtbare Ebene)

+ „Schneide“

- „Falte“

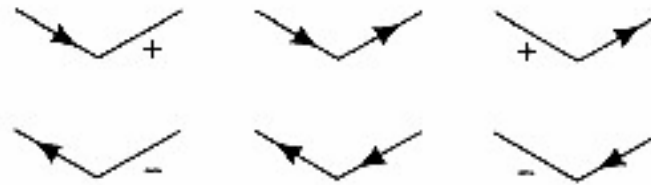


Interpretation von Kantenbildern

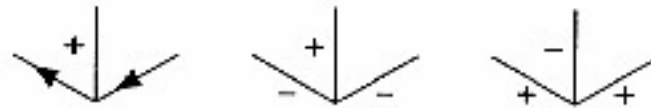
- Aus Bildern, die nur rechteckige Objekte beinhalten, lassen sich unter bestimmten Voraussetzungen die Objekte rekonstruieren:
 - Polygonwelt, wie vorher beschrieben.
 - Aufnahme des Bildes muss so erfolgt sein, dass keine zwei Kanten der Objekte im Bild nur als eine Kante sichtbar sind.
- Idee:
 - Definition aller möglichen Typen von Ecken
 - Suche einer eindeutigen Typisierung der Ecken (bzw. Linienkreuzungen) im Bild. (Dies ist unter den oben genannten Einschränkungen möglich!)

Typen von Ecken (bzw. Kreuzungen)

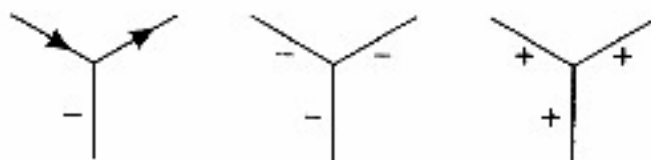
V junctions:



W junctions:



Y junctions:



T junctions:



Interpretation von Kantenbildern

Verfahren:

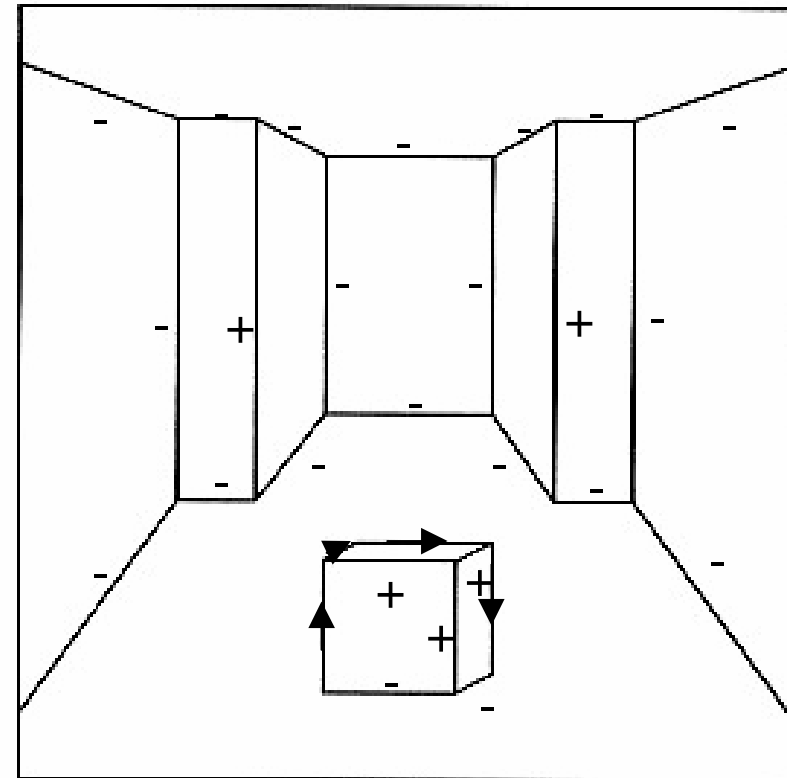
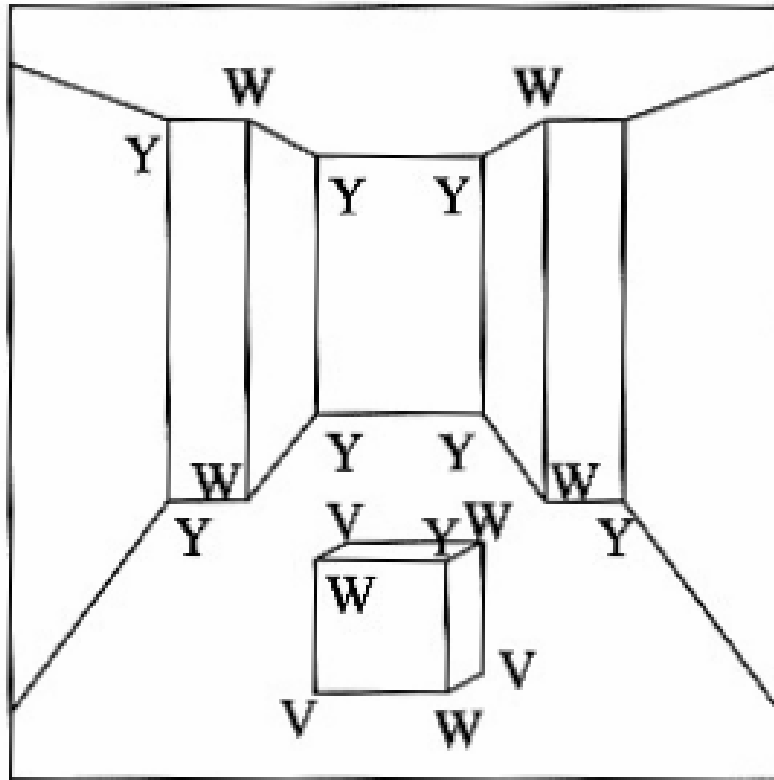
- **Extraktion der Kanten des Bildes**
- **Markieren der Ecken bzw. Kantenkreuzungen:**
 - **Zunächst Markierung der Ecken nach ihrem Grundtypen V, W, Y, T**
 - **Suche nach einer eindeutigen Zuordnung der Untertypen (nach Typ der Ebenenschnitte).**

Bemerkung:

- **Die Bestimmung der Untertypen ist ein Optimierungsproblem mit Nebenbedingungen (*constraint satisfaction problem*). Hierzu gibt es spezielle Lösungsverfahren.**

Interpretation von Kantenbildern

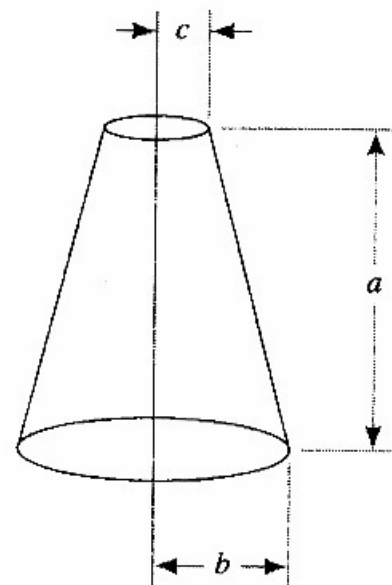
Beispiel:



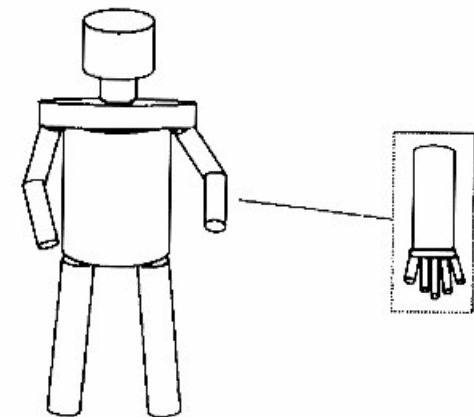
Model Based Vision

- Man nutzt Modelle aller möglichen Objekte einer Szene, um die Objekte zu identifizieren.
- Als Modelle werden z. B. Parallelepipede (verallgemeinerter Quader) und verallgemeinerte Zylinder verwendet.

- Die simulierten Bilder der Modelle werden mit den realen Bildern verglichen.
- Weitere Verbesserung: Tiefeninformation aus Stereo-Bildern (2 Kameras)



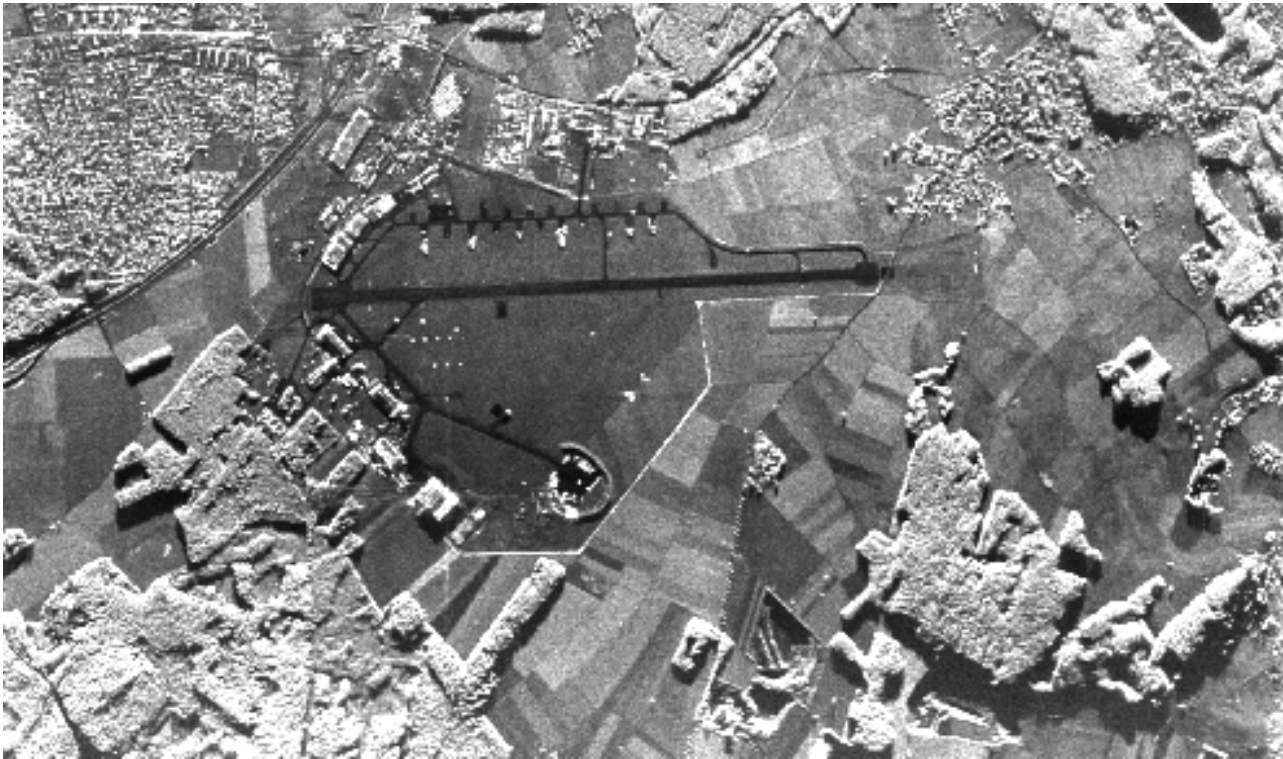
a, b, c + 6 location parameters



Outlook: Information Mining

- **Unter Information Mining versteht man den Prozess, gültige, neue, nützliche und verständliche Muster in heterogene Informationsquellen zu finden.**
- **Informationsquellen sind Datenbanken, Hintergrundwissen von Experten, Texte, Bilder, Geräusche etc.**
- **Eigene Projekte im Bereich Bildverarbeitung:**
 - **Landebahnerkennung (Kooperation mit FHG)**
 - **Erkennen und Klassifizieren von Beulen in Fahrzeugteilen bei der Produktion (BMW)**

Example: Line Filtering



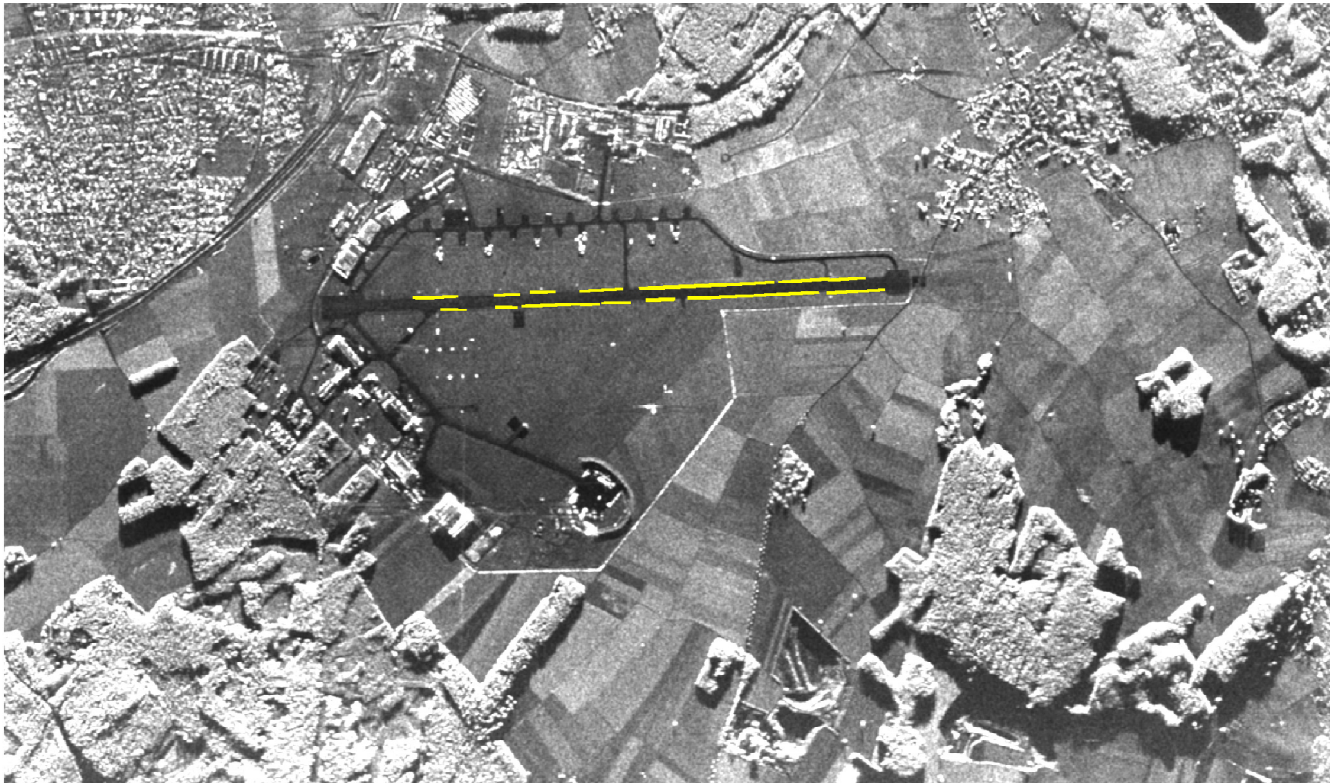
- Extraction of edge segments (Burns' operator)
- Production net:
edges $\rightarrow$ lines $\rightarrow$ long lines $\rightarrow$ parallel lines $\rightarrow$ runways

Example: Line Filtering

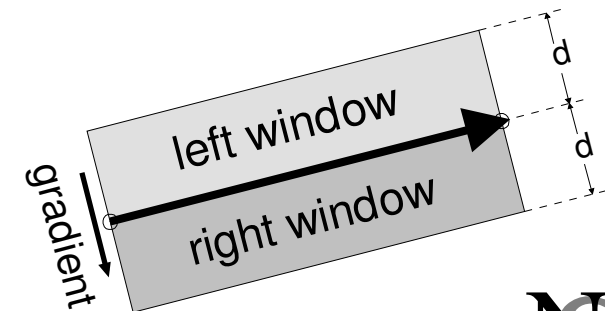


- Problems
 - extremely many lines due to distorted images
 - long execution times of production net

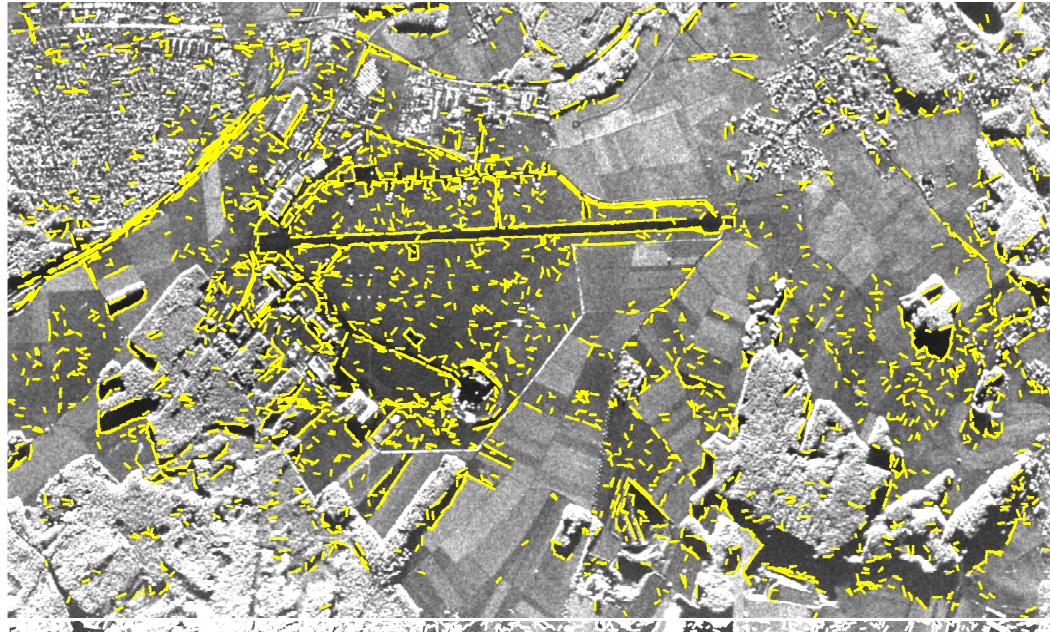
Example: Line Filtering



- Only few lines used for runway assembly
- Approach:
 - Extract textural features of lines
 - Identify and discard superfluous lines



Example: Line Filtering



- Several classifiers:
 - minimum distance, k-nearest neighbor, decision trees, NEFCLASS
- Problems: classes are overlapping and extremely unbalanced
- Result above with modified NEFCLASS:
 - all lines for runway construction found
 - reduction to 8.7% of edge segments

Surface Quality Control: the 2 Approaches

■ Today's Approach

The current surface quality control is done manually → an experienced worker treats the exterior surfaces with a grindstone. The experts classify surface form deviations by means of linguistic descriptions.

Cumbersome – Subjective - Error Prone Time Consuming

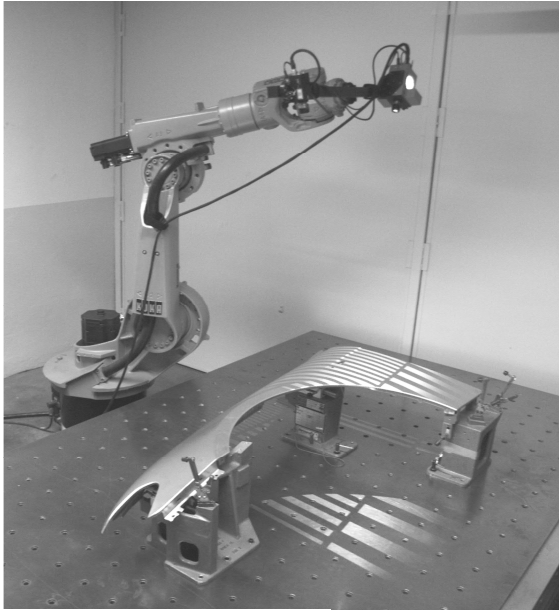


■ The Proposed Approach

Our Approach is based on the digitization of the exterior body panel surface with an optical measuring system.

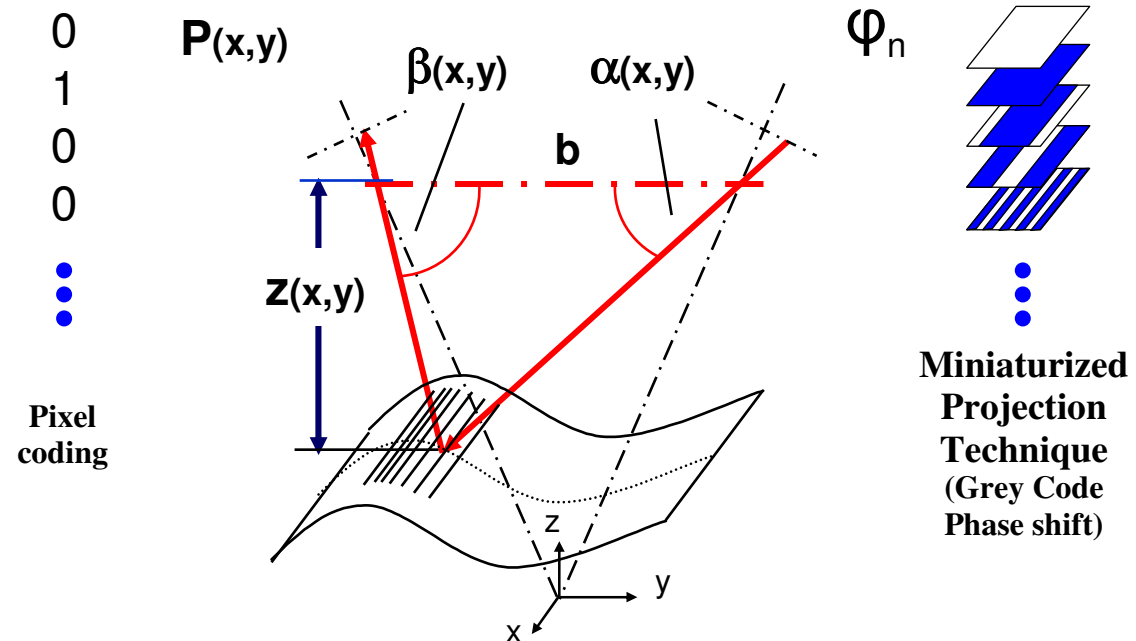
We characterize the form deviation by mathematical properties that are close to the subjective properties that the experts used in their linguistic description.

Topometric 3-D measuring system

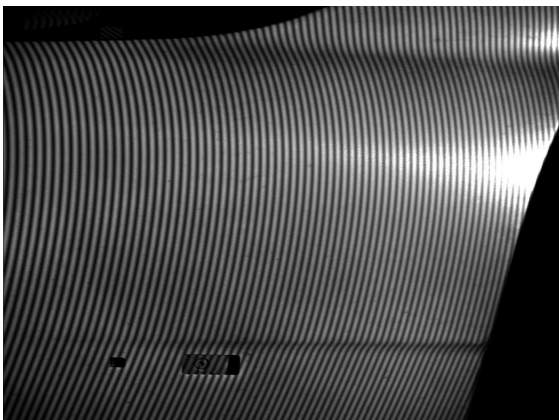


breuckmann 

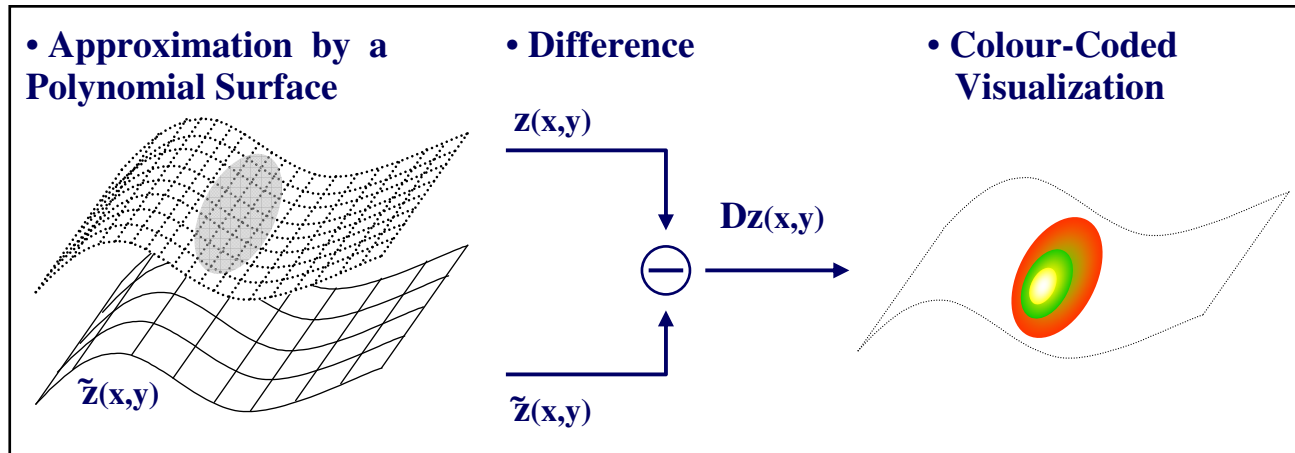
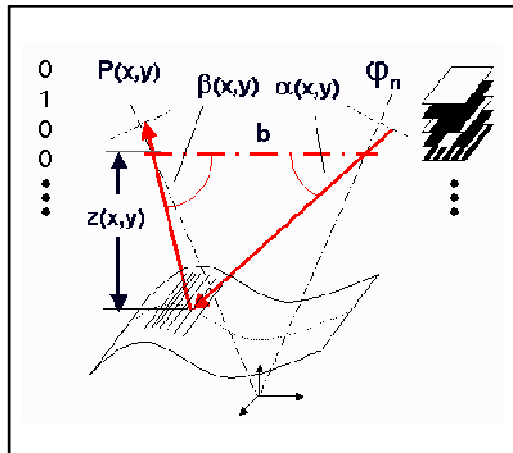
Triangulation and Gratings Projection



- High Point Density
- Fast Data Collection
- Measurement Accuracy
- Contact less and Non-destructive



Data Processing



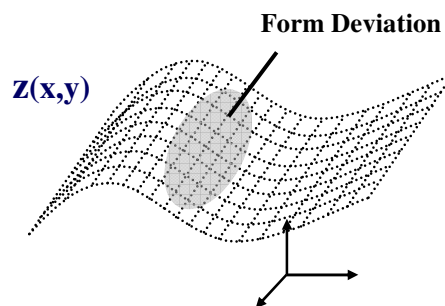
3-D Data Acquisition

Post-Processing

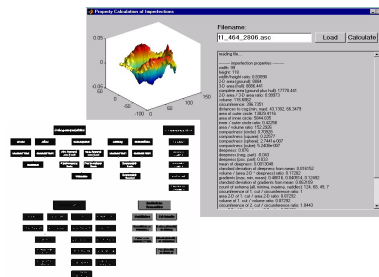
Detection of Form Deviation

Features Analysis

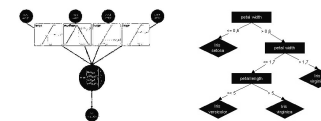
• 3-D-Point Cloud



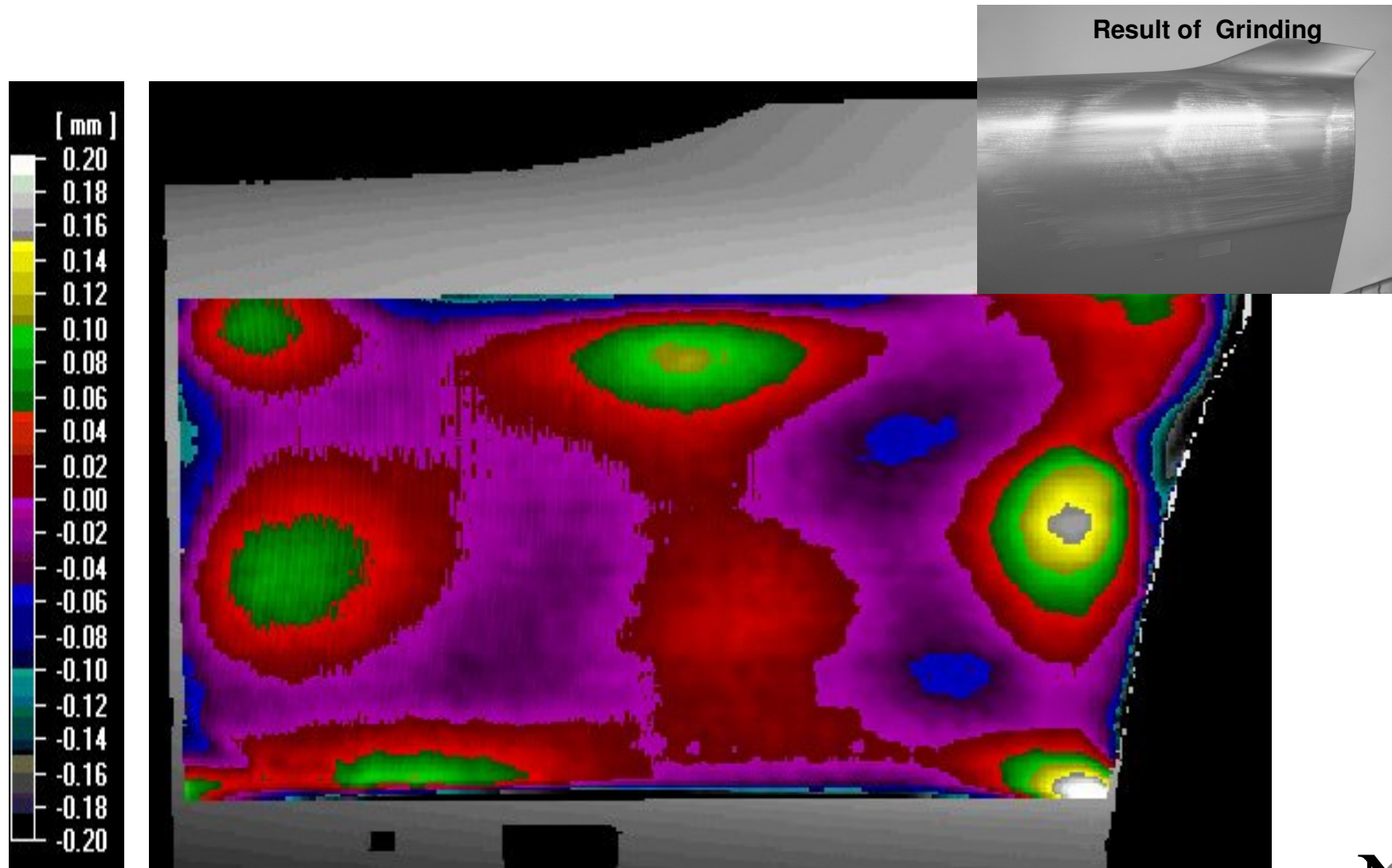
• Feature Calculation



• Classification (Data-Mining)



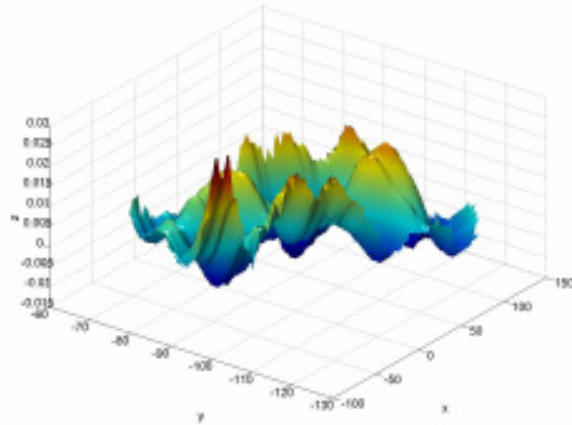
Color Coded Visualization



3D Visualization of Local Surface Defects

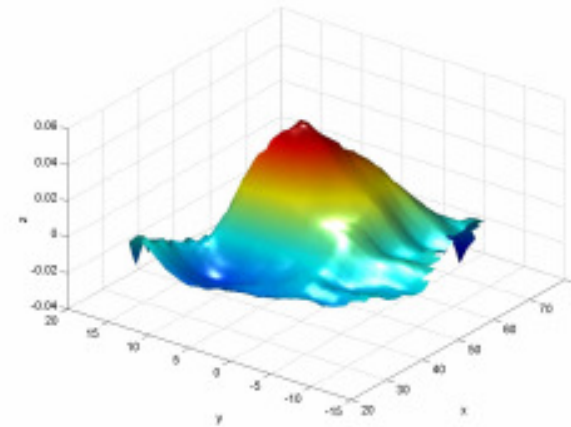
Uneven Surface

(several sink marks in series or adjoined)



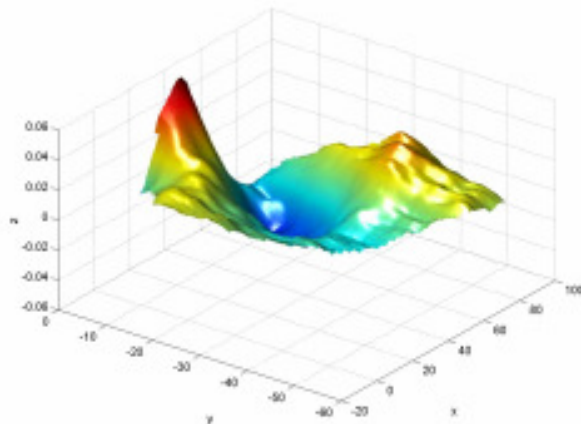
Press Mark

(local smoothing of (micro-)surface)



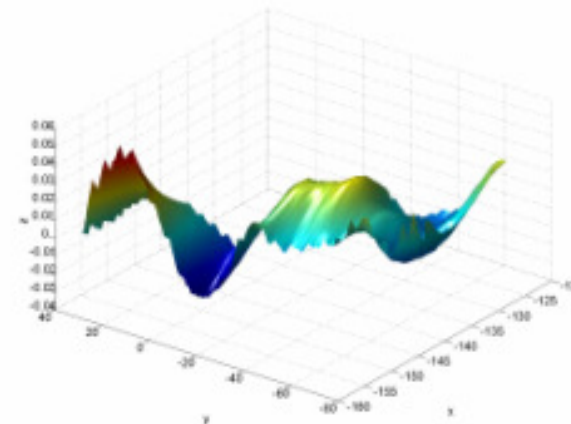
Sink Mark

(slight flat based depression inward)



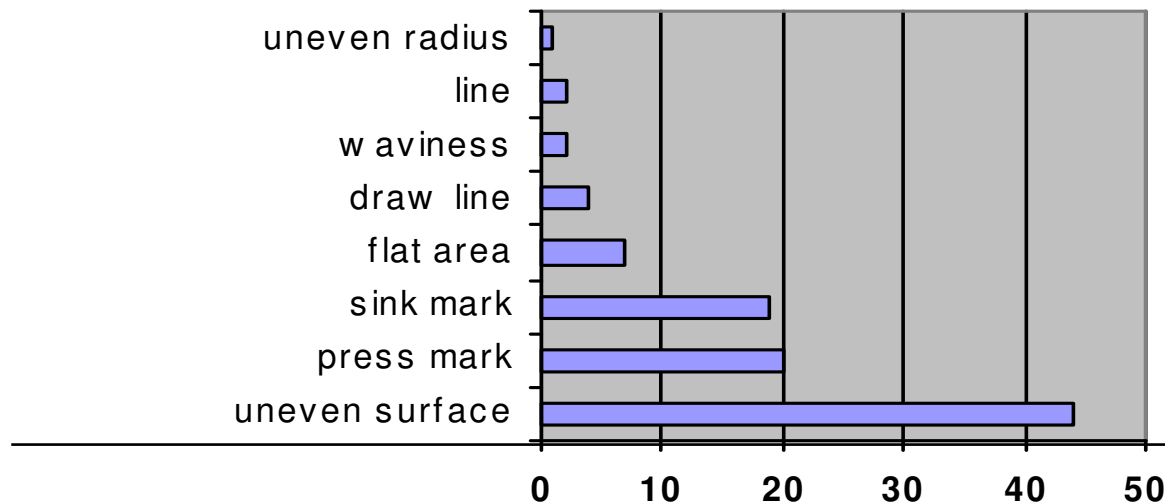
Waviness

(several heavier wrinklins in series)



Data Characteristics

- We analysed 9 master pieces with a total number of 99 defects
- For each defect we calculated 42 features
- The types are rather unbalanced
- We discarded the rare classes
- We discarded some of the extremely correlated features (31 features left)
- We ranked the 31 features by importance
- We use stratified 4-fold cross validation for the experiment.



Application and Results

The Rule Base for NEFCLASS

Rule base

- Rule 1: IF (max_distance_to_cog IS fun 2 AND min_extrema IS fun 1 AND max_extrema IS fun 1) THEN type IS press_mark
- Rule 2: IF (max_distance_to_cog IS fun 2 AND all_extrema IS fun 1 AND max_extrema IS fun 2) THEN type IS sink_mark
- Rule 3: IF (max_distance_to_cog IS fun 3 AND min_extrema IS fun 2 AND max_extrema IS fun 2) THEN type IS uneven_surface
- Rule 4: IF (max_distance_to_cog IS fun 2 AND min_extrema IS fun 2 AND max_extrema IS fun 2) THEN type IS uneven_surface
- Rule 5: IF (max_distance_to_cog IS fun 2 AND all_extrema IS fun 1 AND min_extrema IS fun 2) THEN type IS press_mark
- Rule 6: IF (max_distance_to_cog IS fun 3 AND all_extrema IS fun 2 AND max_extrema IS fun 3) THEN type IS uneven_surface
- Rule 7: IF (max_distance_to_cog IS fun 3 AND min_extrema IS fun 3) THEN type IS uneven_surface

Classification Accuracy

	NBC	DTree	NN	NEFCLASS	DC
Train Set	89.0%	94.7%	90%	81.6%	46.8%
Test Set	75.6%	75.6%	85.5%	79.9%	46.8%

The Rule Base for NEFCLASS

Rule base

- Rule 1: IF (max_distance_to_cog IS fun 2 AND min_extrema IS fun 1 AND max_extrema IS fun 1) THEN type IS press_mark
- Rule 2: IF (max_distance_to_cog IS fun 2 AND all_extrema IS fun 1 AND max_extrema IS fun 2) THEN type IS sink_mark
- Rule 3: IF (max_distance_to_cog IS fun 3 AND min_extrema IS fun 2 AND max_extrema IS fun 2) THEN type IS uneven_surface
- Rule 4: IF (max_distance_to_cog IS fun 2 AND min_extrema IS fun 2 AND max_extrema IS fun 2) THEN type IS uneven_surface
- Rule 5: IF (max_distance_to_cog IS fun 2 AND all_extrema IS fun 1 AND min_extrema IS fun 2) THEN type IS press_mark
- Rule 6: IF (max_distance_to_cog IS fun 3 AND all_extrema IS fun 2 AND max_extrema IS fun 3) THEN type IS uneven_surface
- Rule 7: IF (max_distance_to_cog IS fun 3 AND min_extrema IS fun 3) THEN type IS uneven_surface

Classification Accuracy

	NBC	DTree	NN	NEFCLASS	DC
Train Set	89.0%	94.7%	90%	81.6%	46.8%
Test Set	75.6%	75.6%	85.5%	79.9%	46.8%

Zusammenfassung

- **Computer Vision is hard—noise, ambiguity, complexity**
- **Prior knowledge is essential to constrain the problem**
- **Need to combine multiple cues: motion, contour, shading, texture, stereo**
- **“Library” object representation: shape vs. aspects**
- **Image/object matching: features, lines, regions, etc**