

# Labor Matlab für die industrielle und medizinische Bildverarbeitung

Prof. Dr.-Ing. Bodo Rosenhahn

Institut für Informationsverarbeitung



# Einleitung

- 19.10. Introduction (1h VL, 3 L), Accountvergabe (Präsenz)
- 26.10. Local operators (Harris, etc.) (1h VL, 3L)
- 02.11. Global Operators (Hough Transform) (1h VL, 3L)
- 09.11. Region Growing / Watershed Segmentation (1h VL, 3L)
- 16.11. Bayes Classifier (1h VL, 3L)
- 23.11. K-Means / Mean shift (1h VL, 3L)
- 30.11. Shape Context (1h VL, 3L)
- 07.12. Morphological Operators (1h VL, 3L)
- 14.12. Disparity estimation (DTW) (1h VL, 3L)
- 21.12. Restarbeiten vor Weihnachten (4L)
- 11.01. Calibration and Triangulation (1h VL, 3L)
- 18.01. PCA (1h VL, 3L)
- 25.01. Tracking (1h VL, 3L)

# Heute: Hough-Transformation

**Aufgabe:** Extrahiere Geraden (Kantensegmente) aus Bildern

**Ziel :** Extraktion markanter Merkmale und damit eine Abstraktion des Bildinhaltes

# Hough-Transformation

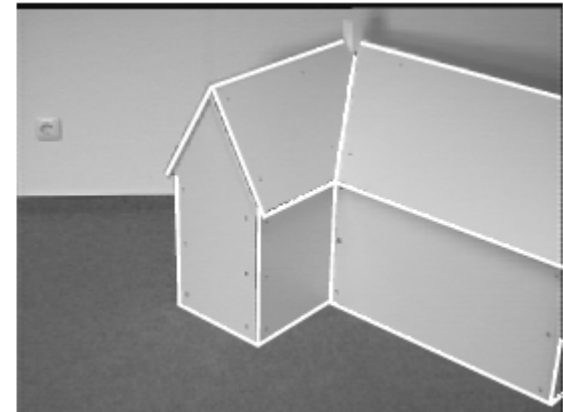
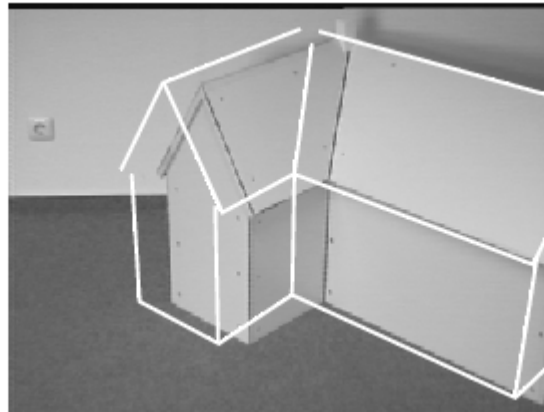
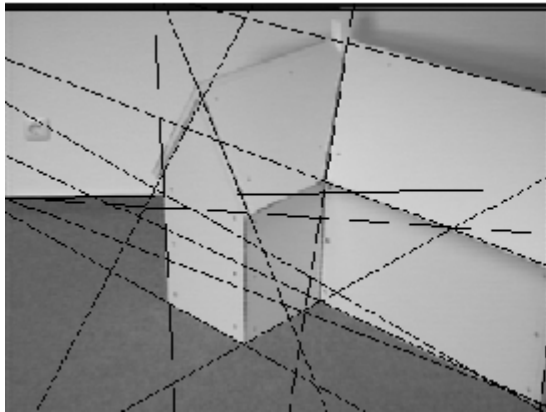
Die **Hough-Transformation** (Sprechweise [hʌf]) ist ein robustes globales Verfahren zur Erkennung von Geraden, Kreisen oder beliebigen anderen parametrisierbaren geometrischen Figuren in einem binären Gradientenbild, also einem schwarz/weiß Bild, nach einer Kantenerkennung. Das Verfahren wurde 1962 von Paul V. C. Hough unter dem Namen „Method and Means for Recognizing Complex Patterns“ patentiert.

<http://de.wikipedia.org/wiki/Hough-Transformation>

# Hough-Transformation

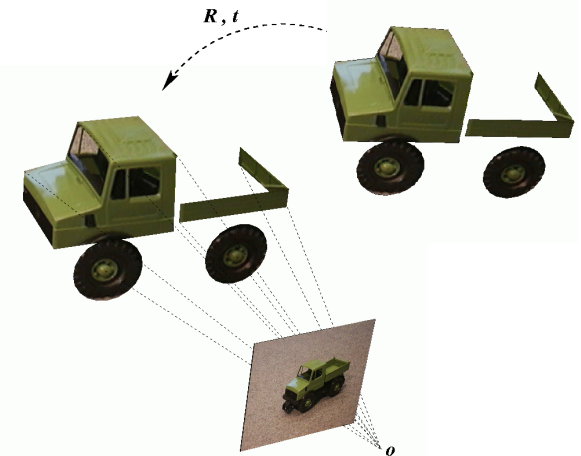
## Beispielanwendung:

Extrahierte Kanten werden an ein 3D CAD-Modell angepasst

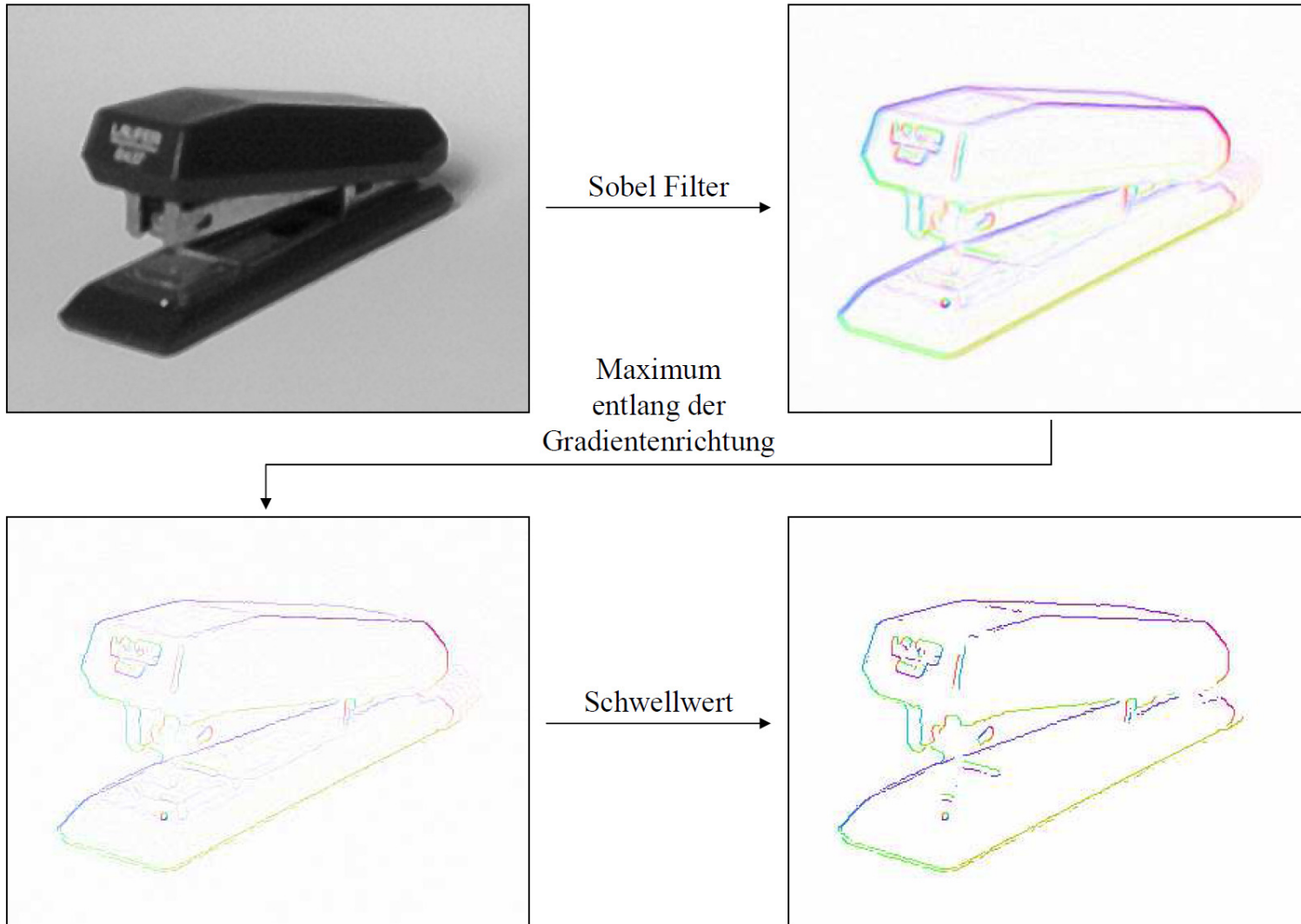


## Poseschätzung:

Vorlesung Rechnergestützte Szenenanalyse

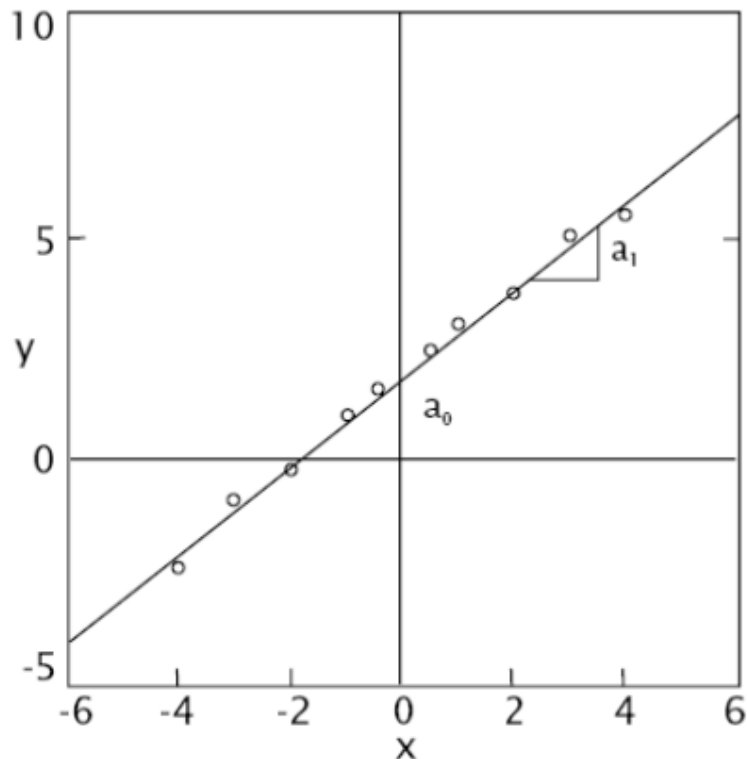


# Vorverarbeitung



# Parametrisierung von Geraden

Klassisch  $y = a_0 + a_1x$



## Problem: Senkrechte Kanten

nếu đường thẳng vuông góc với trục x hoặc y  
thì biến y hoặc x sẽ k đc thể hiện

=> biểu diễn đường thẳng qua số đo góc và khoảng cách

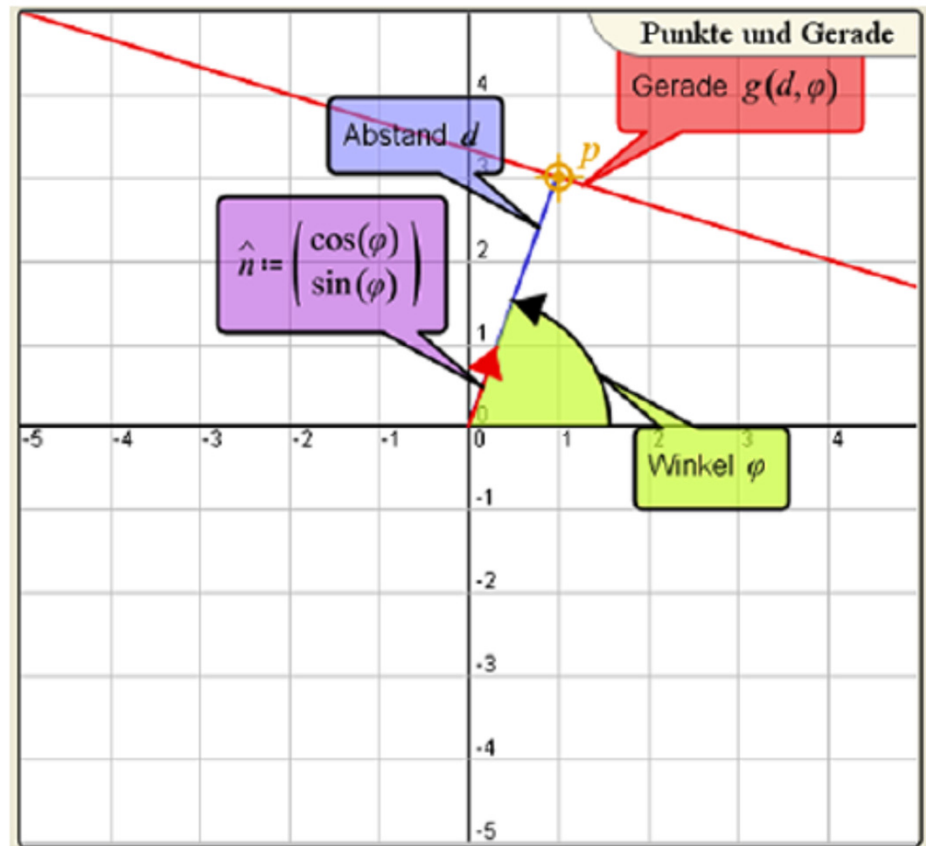
Applet: <http://www.physik.uni-osnabrueck.de/nonlinop/Hough/LineHough.html>

<http://www.saraerhard.de/stuff/Sara-Ausarbeitung.pdf>

# Implizite Geraden

$$x \cos(\phi) + y \sin(\phi) = d$$

$$\Leftrightarrow \underbrace{(\cos(\phi), \sin(\phi))}_{\text{Orientierung}} \begin{pmatrix} x \\ y \end{pmatrix} = \underbrace{d}_{\text{Abstand } d}$$





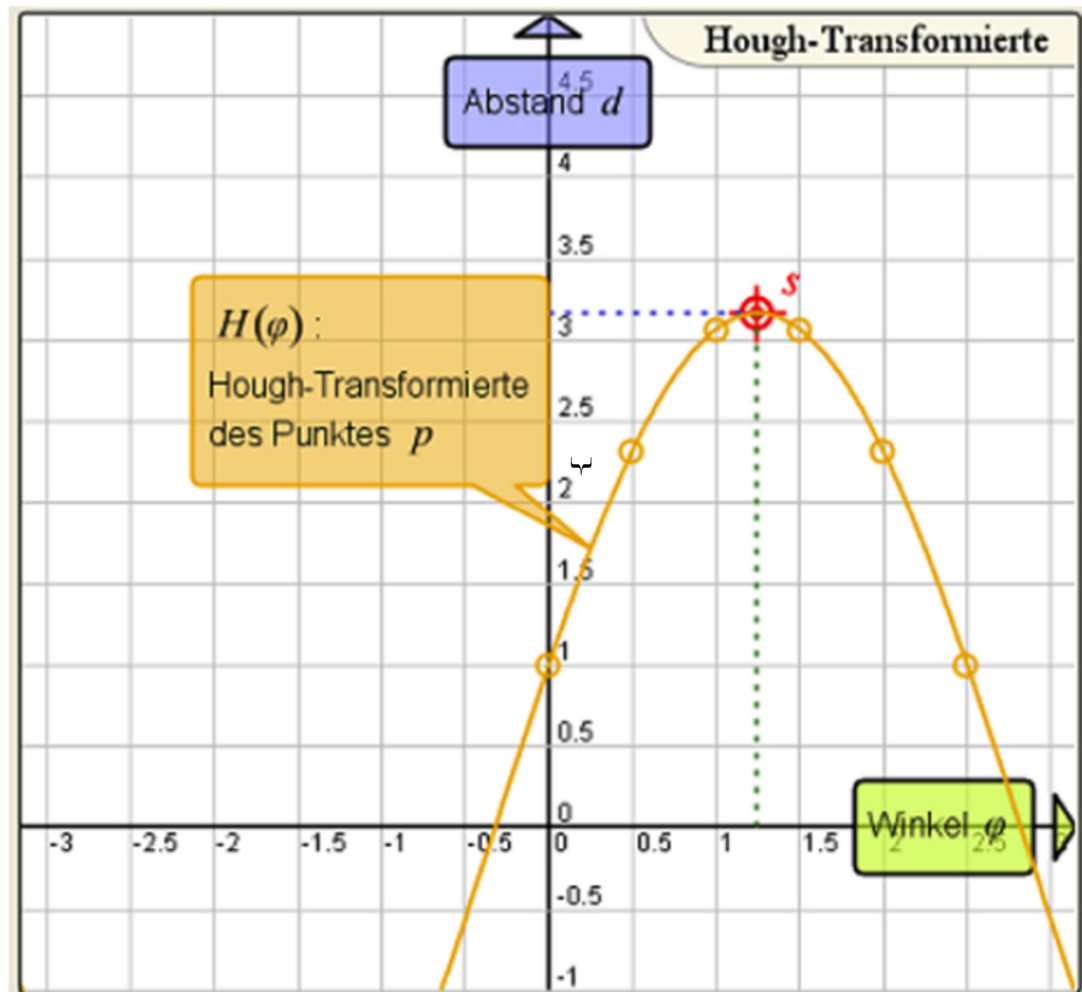
# Hough-Raum

Zu einem gegebenen Kantenpunkt trage alle möglichen Kanten in ein Hough-Array ein, die durch diesen Punkt gehen

Implizite Gerade

$$\underbrace{(\cos(\phi), \sin(\phi))}_{\text{Orientierung}} \begin{pmatrix} x \\ y \end{pmatrix} = \underbrace{d}_{\text{Abstand}}$$

**Idee:** Sample zu jedem detektierten Punkt  $(x, y)$  für verschiedene Winkel  $\phi$  zugehörige  $d$



# Aufbau eines Hough-Arrays

Theoretisch unendlich viele mögliche Linien

- Nur diskrete Werte für  $\phi$  und  $d$  zulassen (Diskretisierung)
- -> Akkumulatorfeld mit Akkumulatorzellen (initialisiere mit 0)
- Jeder detektierte Punkt induziert ein Geradenbündel um diesen Punkt.  
-> erhöhe in dem Array das entsprechende Feld um 1
- Liniendetektion wird transformiert in finden lokaler Maxima im Akkumulator-array (möglichst viel Support von extrahierten Punkten).

**Kann auch auf höherdimensionale Felder und Kurvenbahnen erweitert werden !**

## Algorithmus: Kurvendetektion

Beliebige Kurve dargestellt durch  $f(x,a)=0$

$x$ : Positionsvektor  $a$ : Vektor der Kurvenparameter

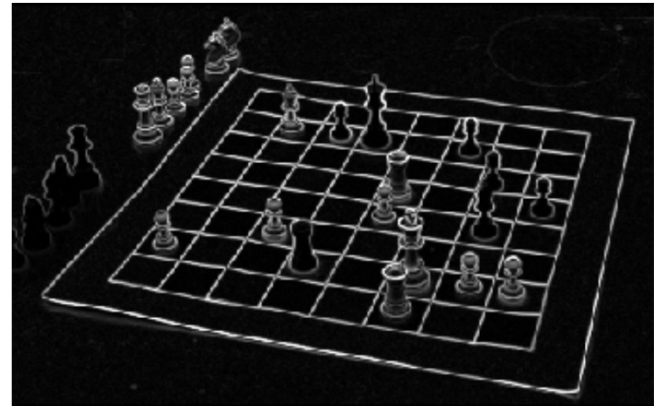
- 1) Quantisierung des Parameterraums innerhalb der Grenzen von  $a$ , Dimension des Parameterraums gegeben durch Dimension  $n$  des Vektors  $a$
- 2) Aufbau des Parameterraums  $A(a)$  entsprechend der Quantisierung, Nullsetzen der Akkumulatorzellen
- 3) Für jedes Kantenpixel  $(x1,x2)$ : erhöhe alle Akkuzellen  $A(a)$  wenn gilt  $f(x,a)=0$ :  $A(a)=A(a)+ dA$  für alle  $a$
- 4) Lokale Maxima im Akkumulator entsprechen Auftreten der Kurve  $f(x,a)$  im Bild

# Beispiel

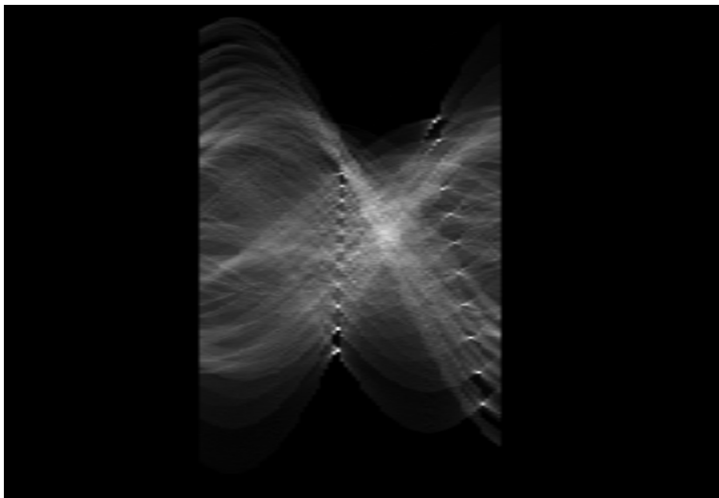
Eingabe



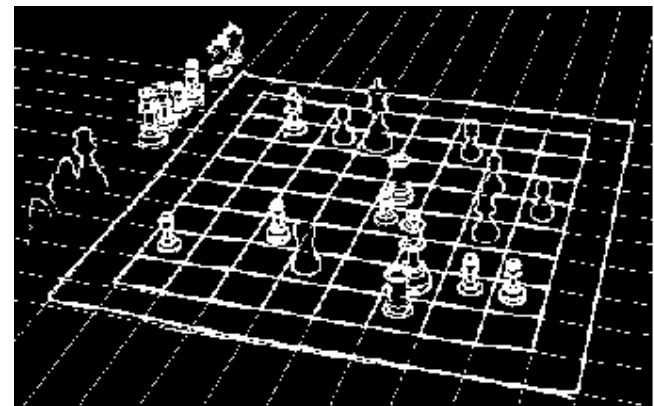
Kantenbild



Hougharray

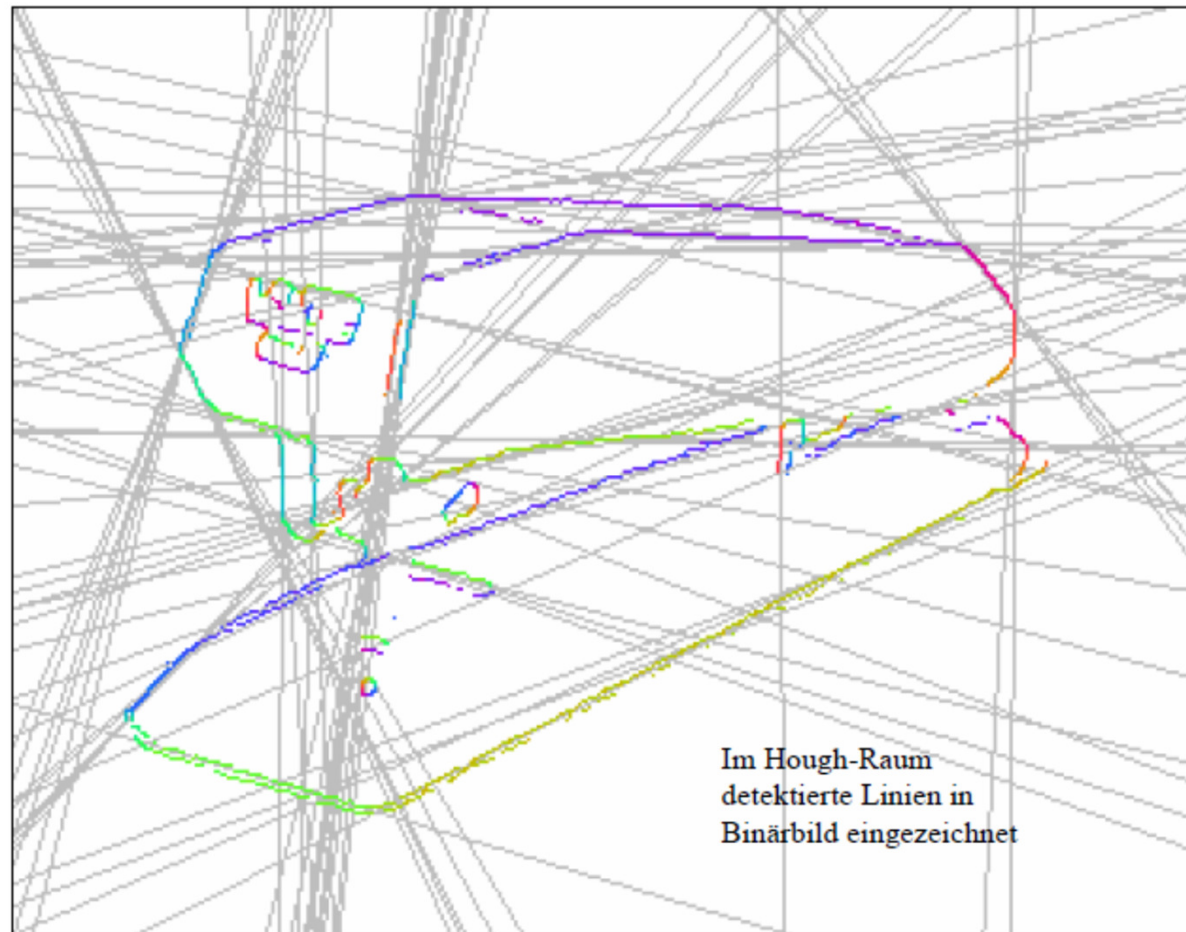


Überlagerte Geraden

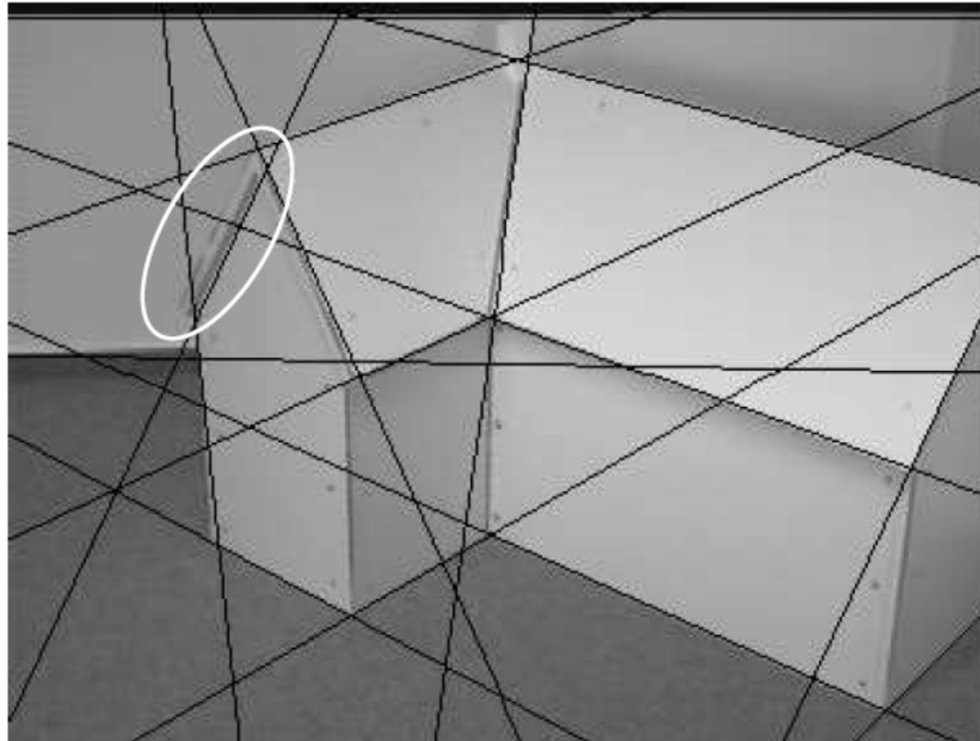


# Problem: Geradenbündelbildung

Bei kurzen Geradensegmenten findet man mehrere Geraden.

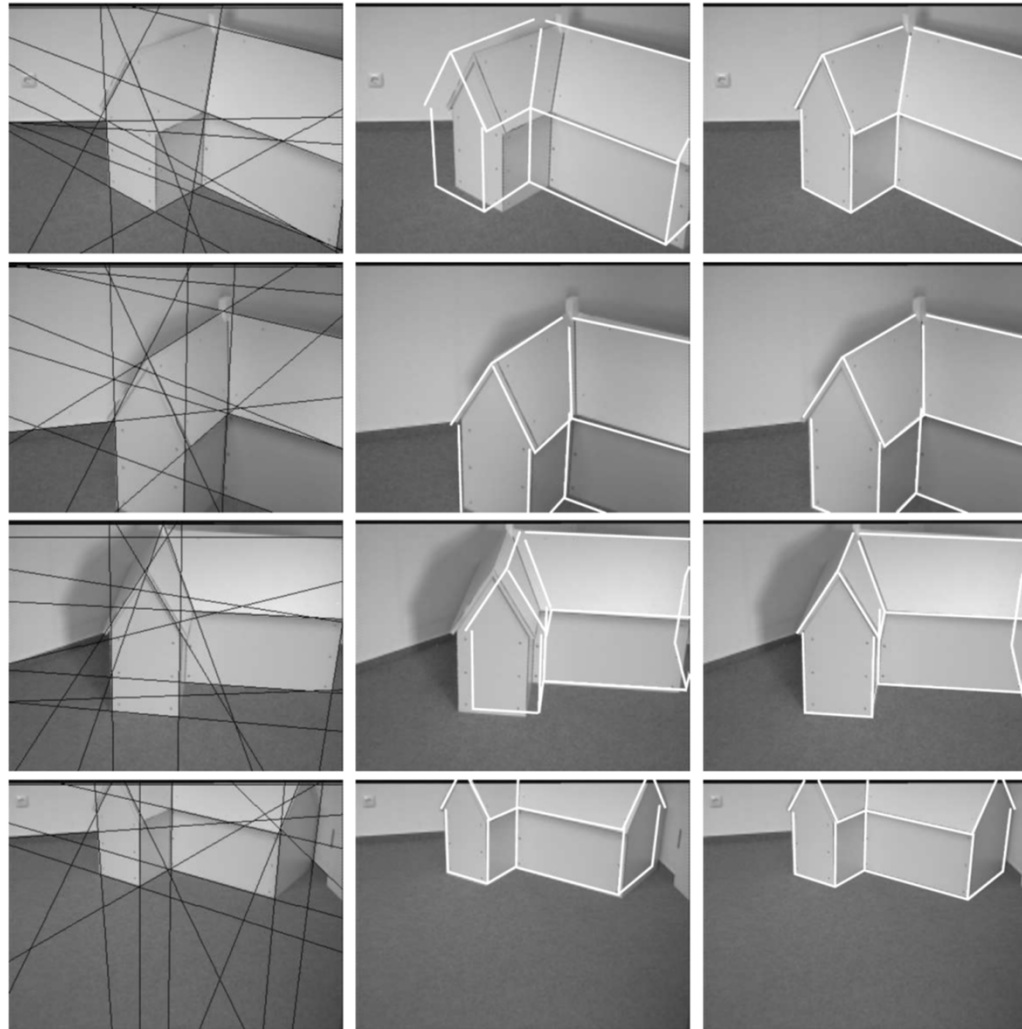


(2) Maximumssuche mit Bereichslöschung:  
Lösche den Bereich um das gefundene Hauptmaxima



**Problem:** Enge parallele Kanten können übersehen werden

# Beispiel



## 2. Beispiel : Kreise

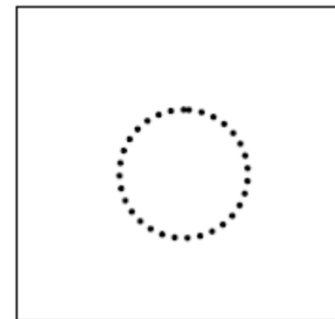
$f(x, a)$  entspricht:  $(x_1 - a)^2 + (x_2 - b)^2 - r^2 = 0$

- 3-dimensionaler Parameterraum
- Für jedes Kantenpixel für alle möglichen Kreismittelpunkte  $(a, b)$  entsprechenden Radius  $r$

**Berechnen und eintragen:**  $A(a, b, r)$

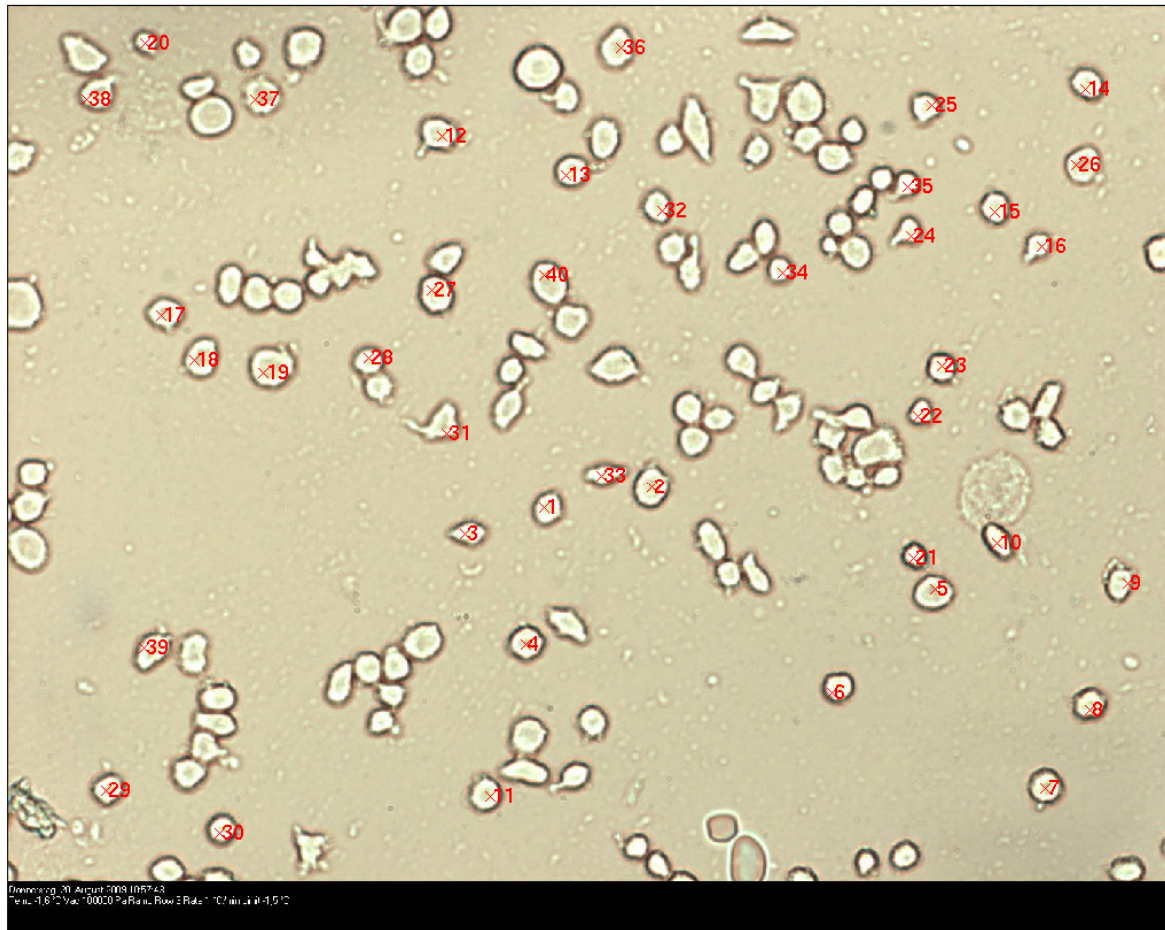
Java-Demo (fixed radius)

<http://www.markschulze.net/java/hough/>



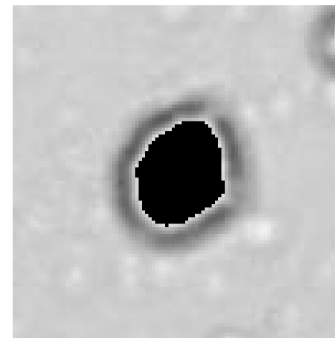
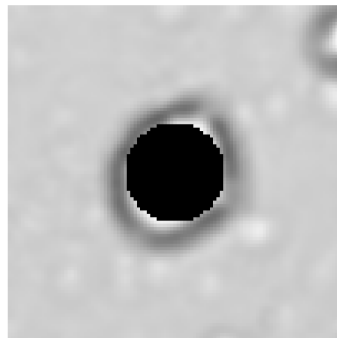
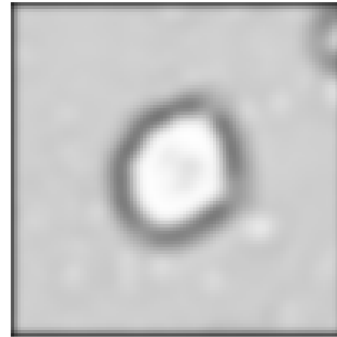
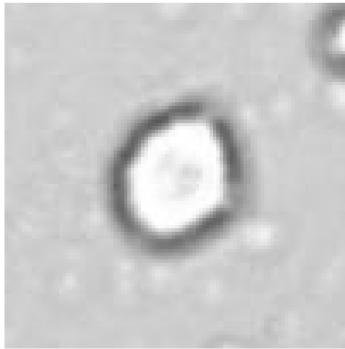


# Beispielanwendung: Zellsegmentierung

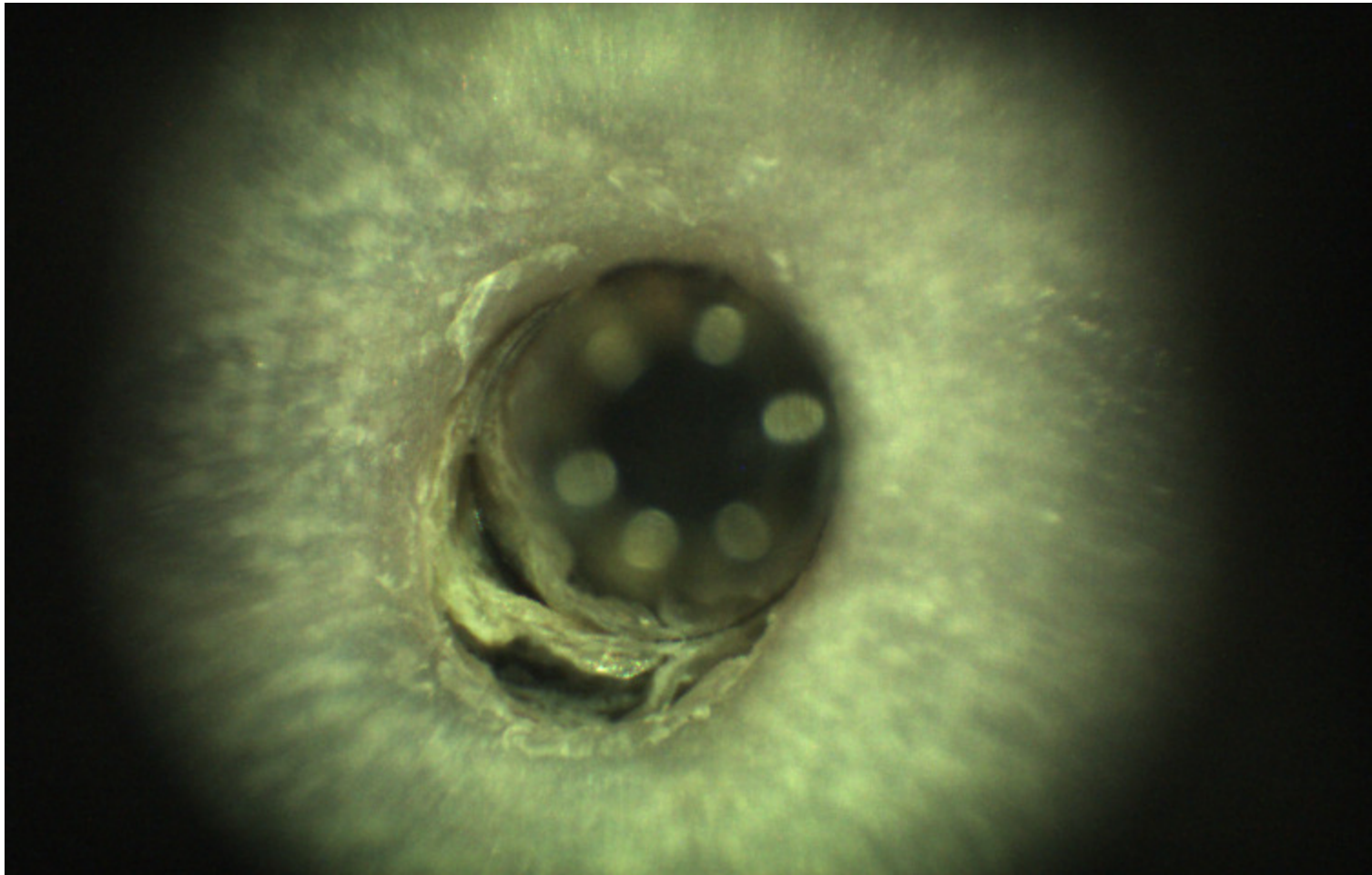


Data: Courtesly provided by Inst. of Multiphase Processes

# Zellsegmentierung

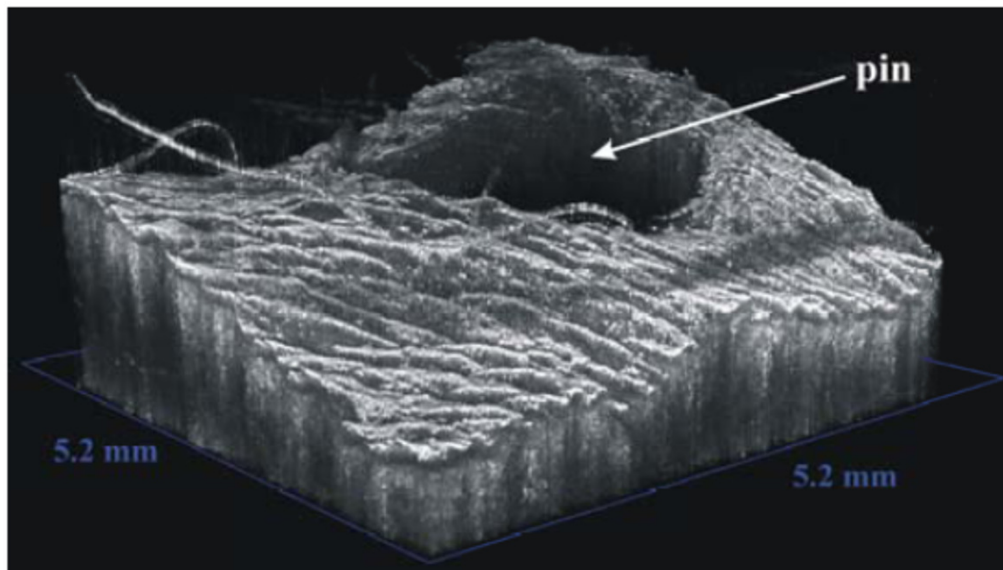
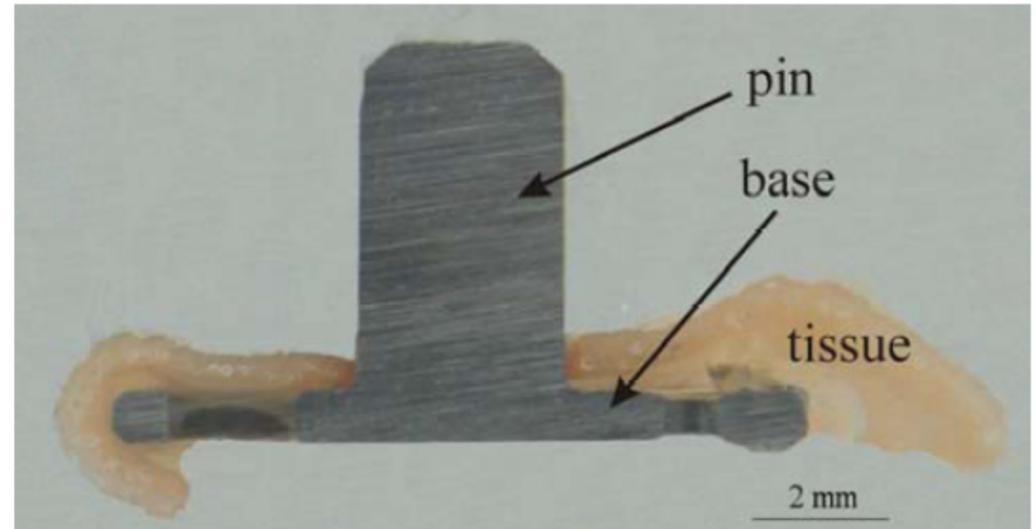


## 2. Beispielanwendung: OCT-Segmentierung



Data : Courtesy provided by LZH-Hannover

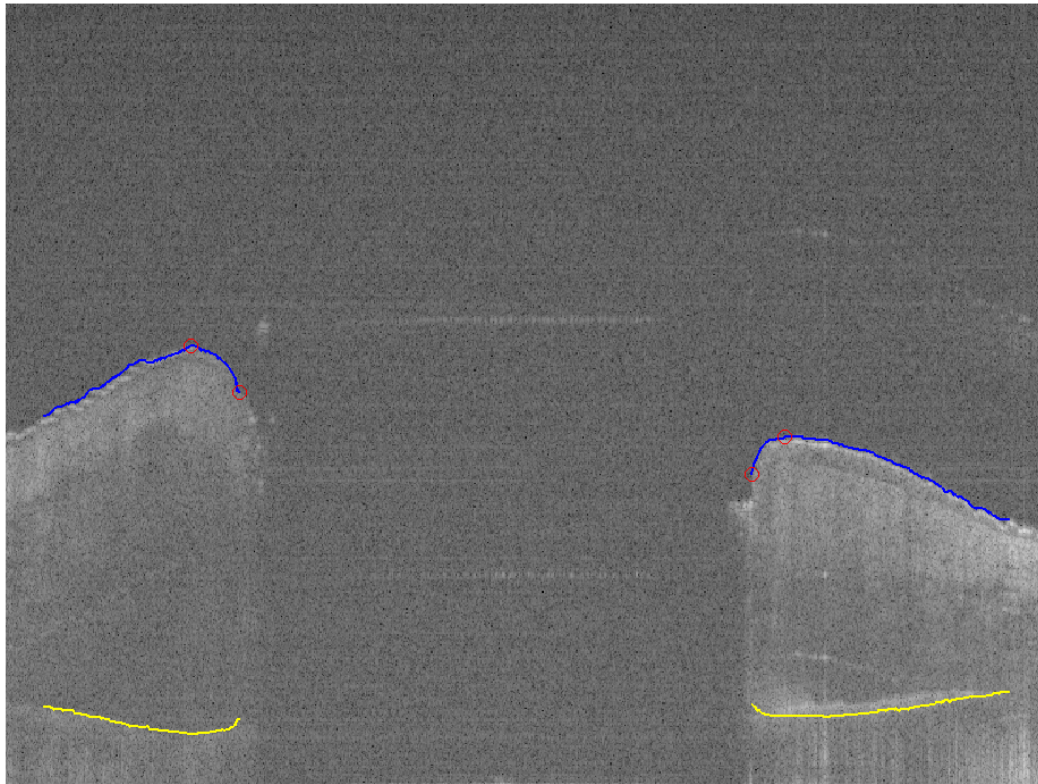
## 2. Beispielanwendung: OCT-Segmentierung



OCT-Scan

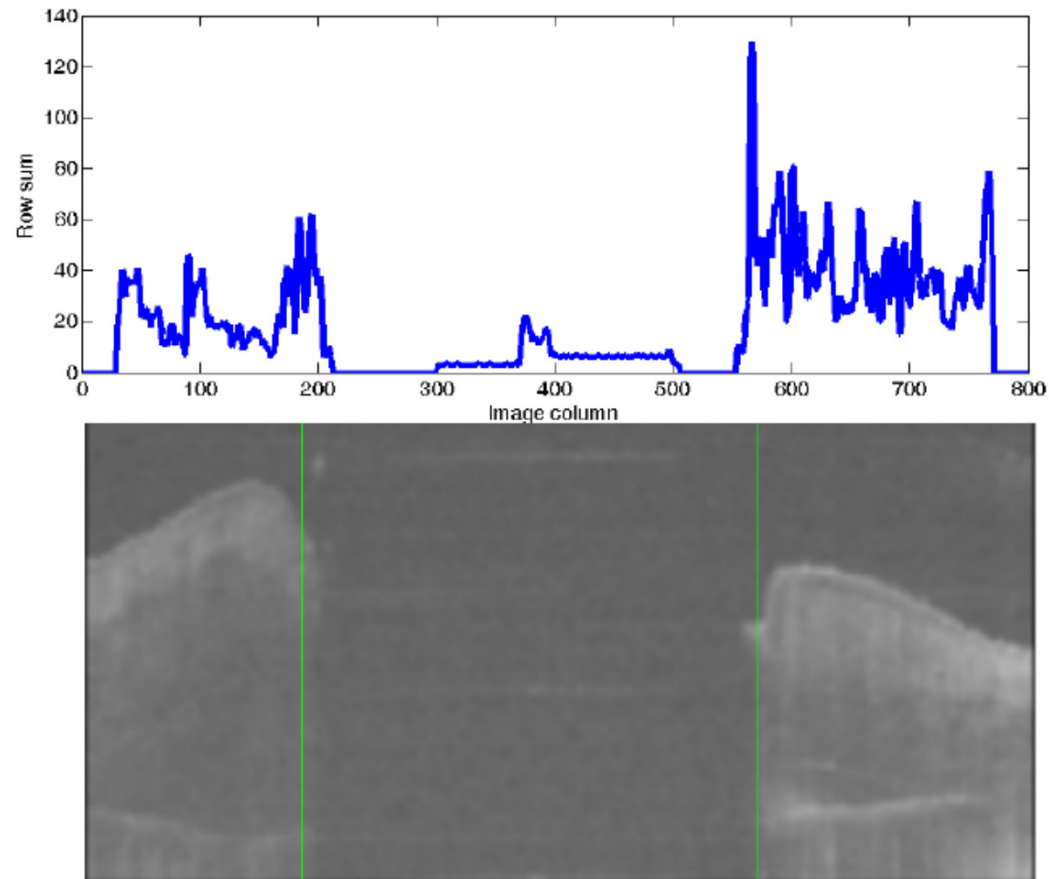


# Haut- / Basisdetektion

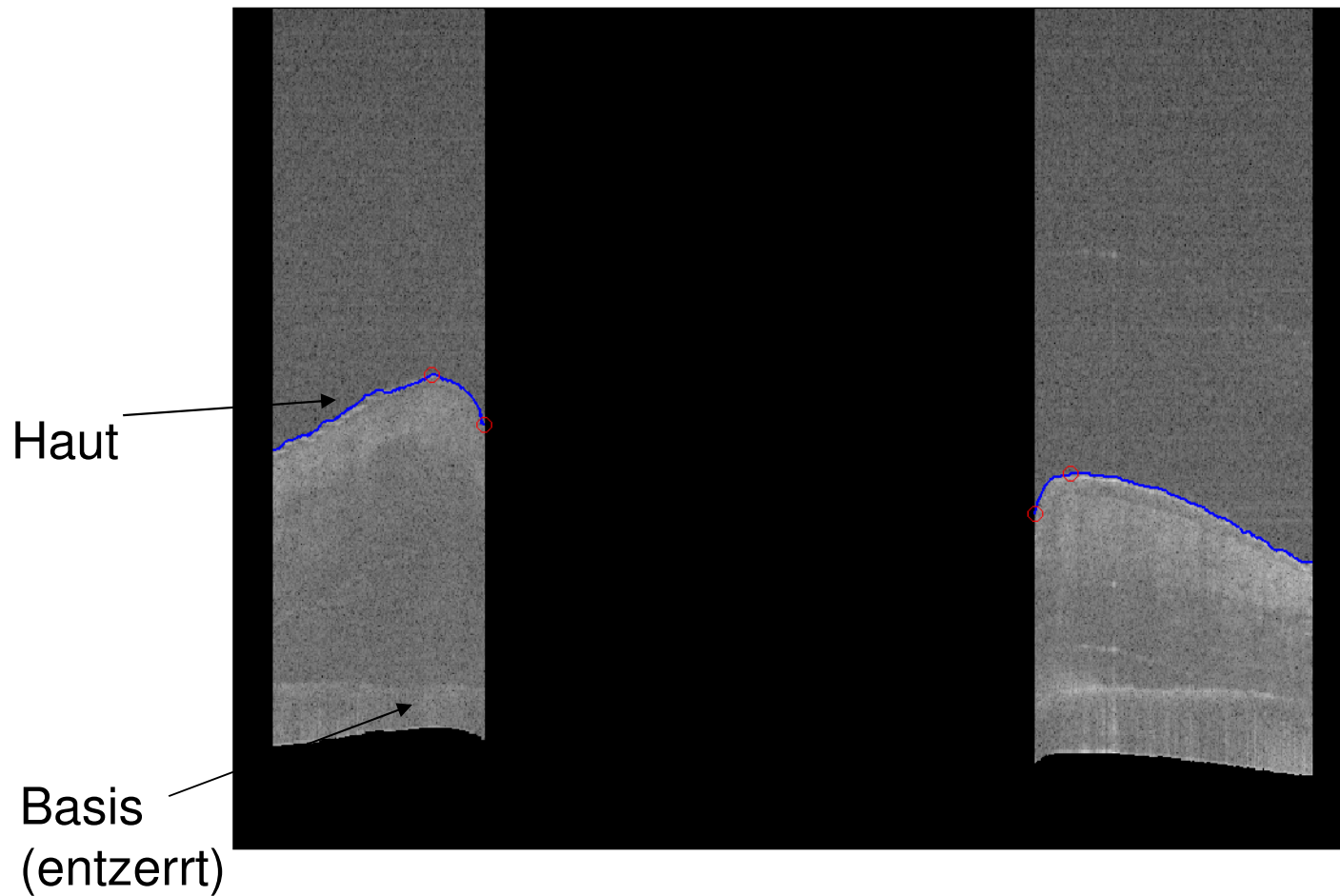


# Beispielanwendung: Pin-Segmentierung

## 1. Aufgabe: Detektiere Pin



# Warping



BZ : 1.32

Orientation : 0.5

Tumidity: 206,  
201 Px.

# Line detection





- HoughTransform Beispiel unter Matlab
  - - Line detection
  - - Circle detection
  - - Y-Hough-Detection