

## 5. Die Hough-Transformation

(sprich: "haff")

P.V.C. Hough, Patentschrift von 1962

Ziel: Erkennen von Geraden

– allgemeiner: von einfachen parametrischen Kurven (Kreise, Ellipsen...)

Anwendung:

Detektion kollinearere Punkte

Konturbeschreibung

⇒ auch zur *Bildsegmentierung* gerechnet

Idee:

- Transformation in einen Raum, wo Geraden (bzw. andere Kurven) als Punkte dargestellt werden ⇒ kollineare Geradenstücke fallen alle auf einen Punkt
- Zählung in einer Akkumulator-Matrix
- Maxima in dieser Matrix entsprechen den Geraden im Originalbild

Geradengleichung:

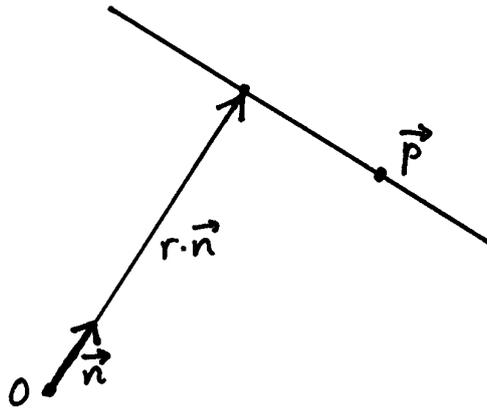
$$y = mx + b$$

- Parameter  $m, b$  (Steigung,  $y$ -Achsen-Abschnitt) wenig geeignet, da  $m$  auch  $\infty$  werden kann

bessere Geradendarstellung: *Hessesche Normalform* (HNF)

Herleitung:

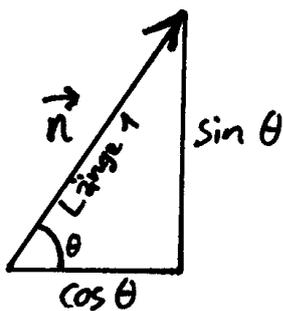
Gerade kann charakterisiert werden durch Normaleneinheitsvektor  $\vec{n}$  (steht senkrecht zur Geraden und zeigt weg von 0) und ihren Abstand  $r$  vom Nullpunkt



- ⇒  $r \cdot \vec{n}$  liegt auf der Geraden
- ⇒ für jeden beliebigen Punkt  $\vec{p}$  auf der Geraden gilt:
  - $\vec{p} - r \cdot \vec{n}$  hat die Richtung der Geraden
  - ⇔  $(\vec{p} - r \cdot \vec{n}) \perp \vec{n}$
  - ⇔  $(\vec{p} - r \cdot \vec{n}) \cdot \vec{n} = 0$
  - ⇔  $\vec{p} \cdot \vec{n} - r \cdot 1 = 0$
  - ⇔  $\vec{p} \cdot \vec{n} = r$  (HNF)

(der dicke Punkt bezeichnet das Skalarprodukt).

Statt der Angabe beider Komponenten von  $\vec{n}$  genügt der Winkel  $\theta$  von  $\vec{n}$  zur x-Achse:



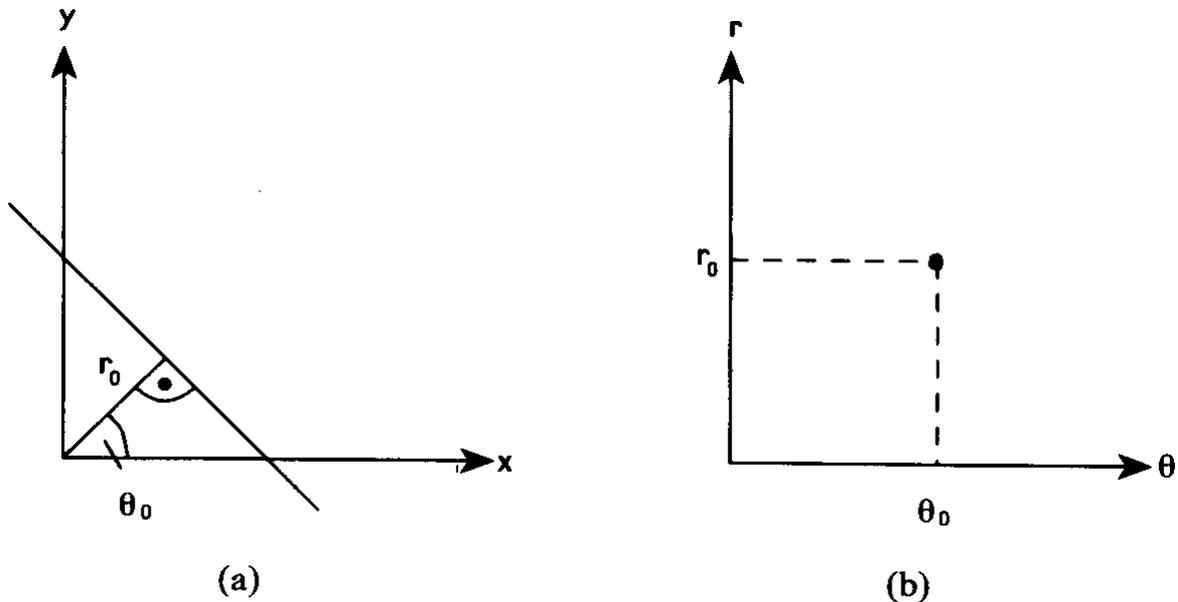
$$\vec{n} = \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix}$$

Somit (mit  $\vec{p} = \begin{pmatrix} x \\ y \end{pmatrix}$ ):

Geradengleichung  $x \cos \theta + y \sin \theta = r$

⇒ Codierung der Geraden durch  $(r, \theta)$ .

Somit kann jede Gerade durch einen Punkt im  $(r, \theta)$ -Raum dargestellt werden:



( $\theta$  im Intervall  $[0; \pi)$ ,  $r$  darf negativ werden.)

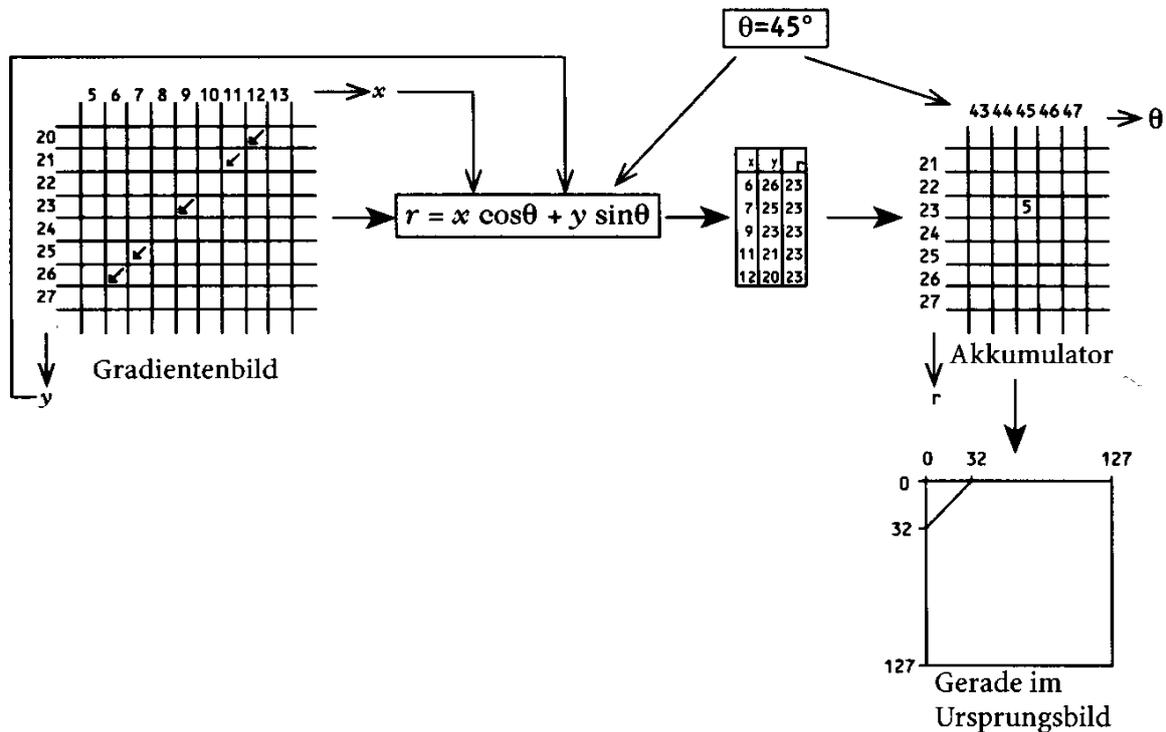
Hough-Transformation: Gerade-zu-Punkt-Transformation

- dient der Aufbereitung der Daten, um sie in einem nachfolgenden Schritt einfacher auswerten zu können

Vorangehende Schritte sind i.allg.:

- Konturextraktion durch Gradientenoperation (z.B. Sobel, Laplace)
- Verdünnung des Kantenbildes (dazu später mehr)
- Richtung orthogonal zum Gradienten stellt Kontur-Richtung dar

Beispiel: 5 Kontur-Vektoren, die auf einer Geraden liegen



Winkelinformation  $\theta$  aus der Richtung des Gradienten  $r$  aus  $x$ ,  $y$  und  $\theta$

⇒ hier für alle 5 Punkte dasselbe  $r$

⇒ Wert 5 in der Akkumulator-Tabelle

(Akkumulator diskretisiert wie ein digitales Bild)

Ergebnis der Hough-Transformation: gefüllter Akkumulator

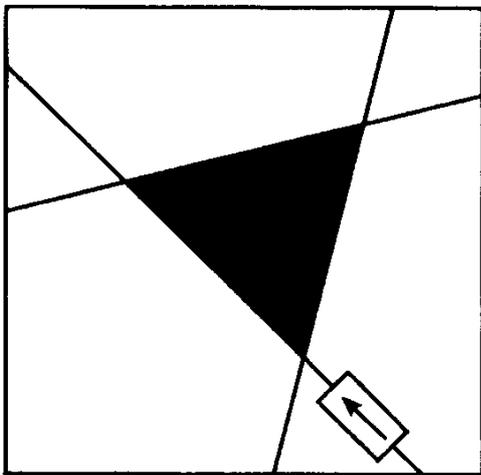
nächste Schritte:

- evtl. "Säubern" des Akkumulators (Beseitigung von Störeinflüssen: etwa Zusammenfassen von Clustern)
- Analyse des Akkumulators (Maxima, Überschreitung von Schwellenwerten)
- Rücktransformation der gefundenen "kritischen Punkte" (Maxima) ins Originalbild: dort *Geraden*

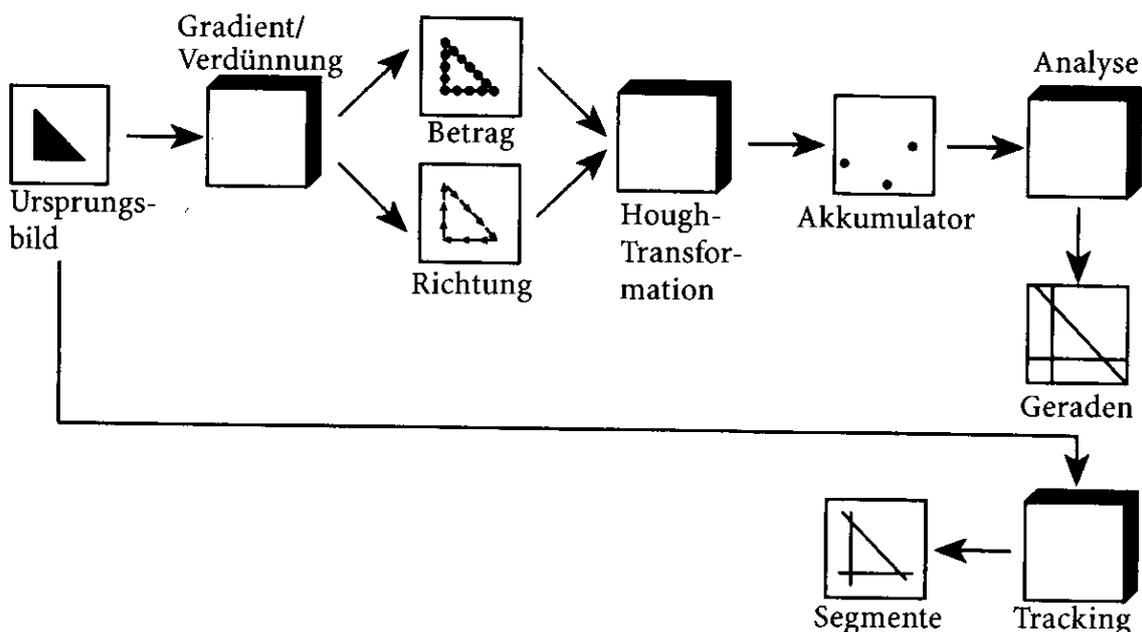
Problem: man will meist Geradenstücke haben  
 (Information über Anfangs- und Endpunkte), keine  
 kompletten Geraden

Verwendung der kompletten Geraden als "Wegweiser"  
 durch das Originalbild:

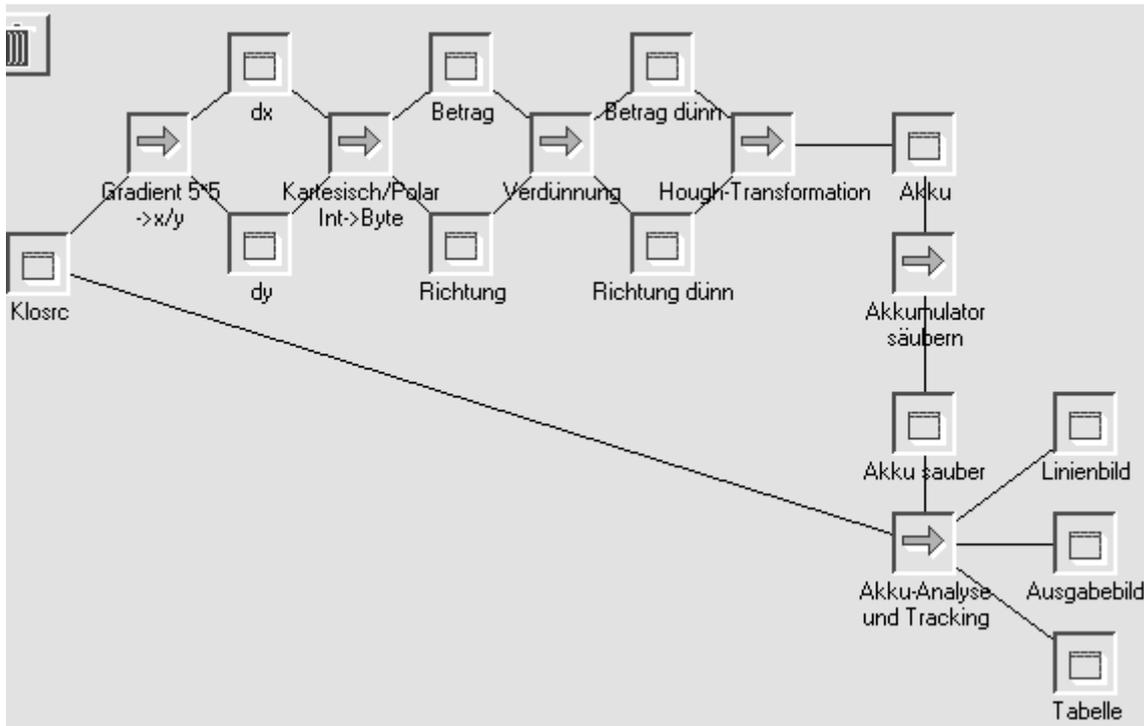
z.B. dort morphologischen Filter entlanglaufen lassen  
 Vergleich der Pixel rechts und links der Geraden  
 wenn vorgegebene Grauwertdifferenz überschritten wird:  
 Filtermaske befindet sich auf einer Objektkontur



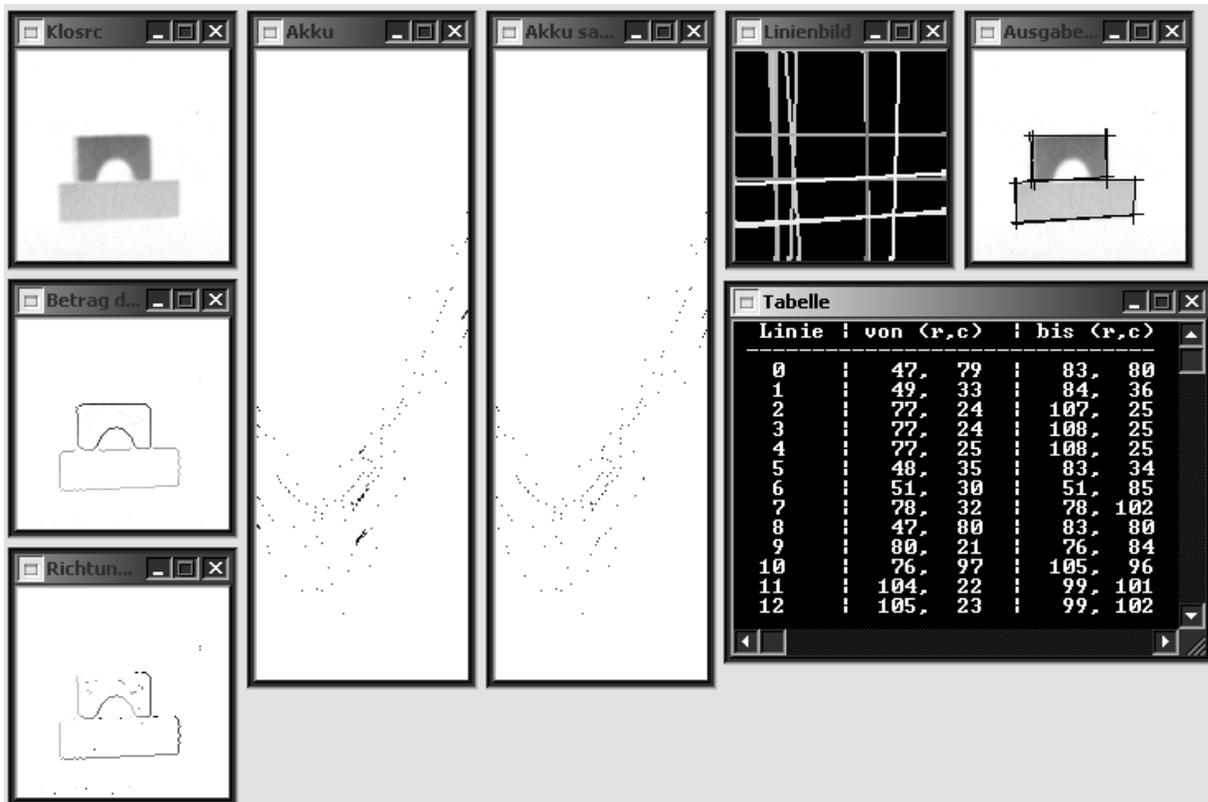
Kompletter Arbeitsgang (aus Bässmann & Kreyss 1998):



## Beispiel mit AdOculos-Software (Setup HOUGH.SET):



## Ergebnisse (invertiert):



Original  
Kantenbild  
Richtungsbild

Akkumulator

Akkumulator  
(geclustert)

Geraden

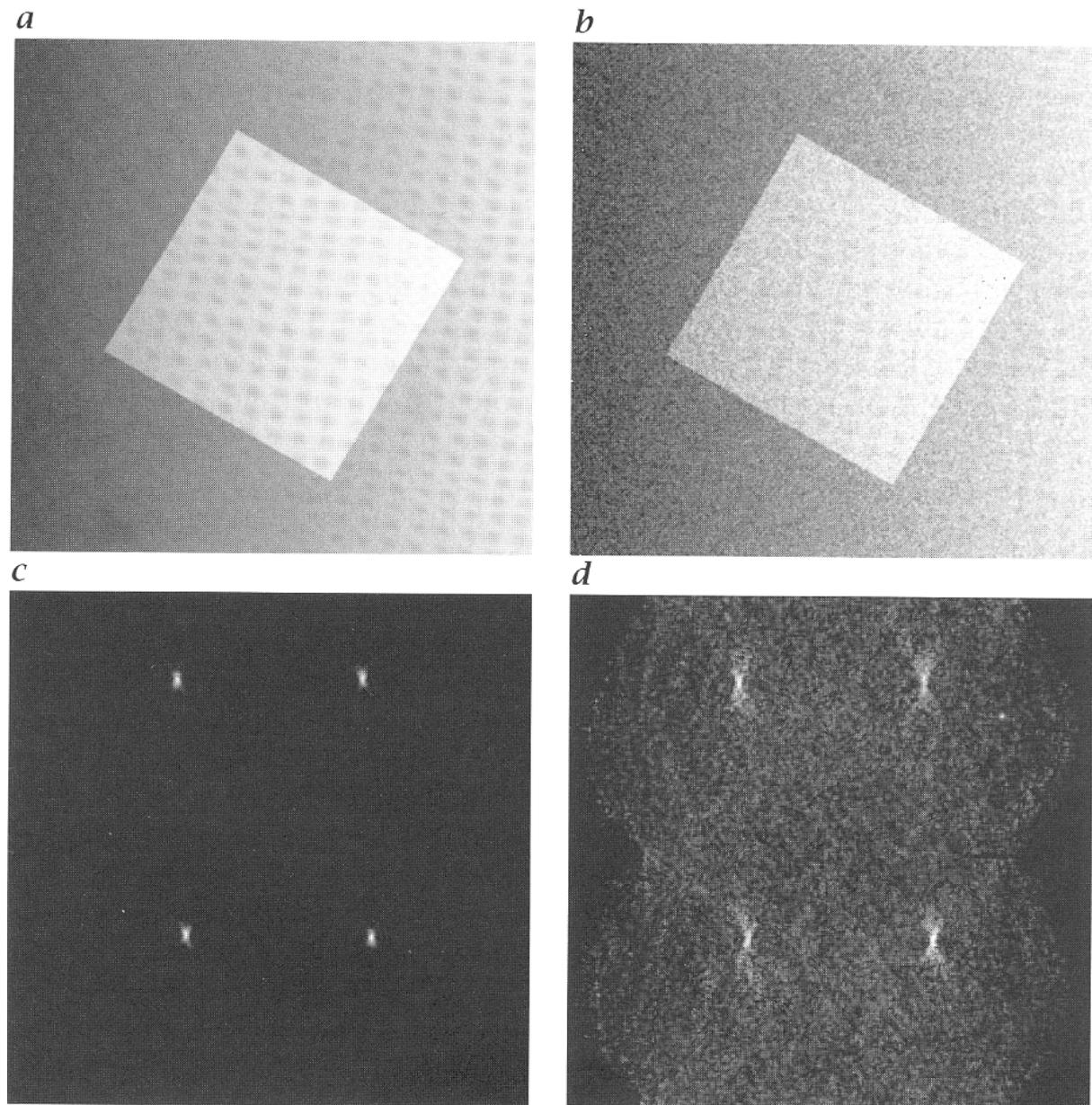
Geradensegmente  
im Originalbild  
tabellarische Darstellung d. Seg.

- Vorteil: Geraden werden auch erkannt, wenn sie verrauscht sind oder Unterbrechungen aufweisen.

anderes Beispiel (aus Jähne 2002):

a, b: ungleichmäßig beleuchtete, verrauschte Quadrate

c, d: entsprechende Hough-Modellräume ( $(r, \theta)$ -Ebene) zu a und b



Beachte: parallele Geraden stehen in derselben Spalte des Akkumulators  $\Rightarrow$  Verfahren auch zum Erkennen von Parallelität verwendbar.

## Probleme:

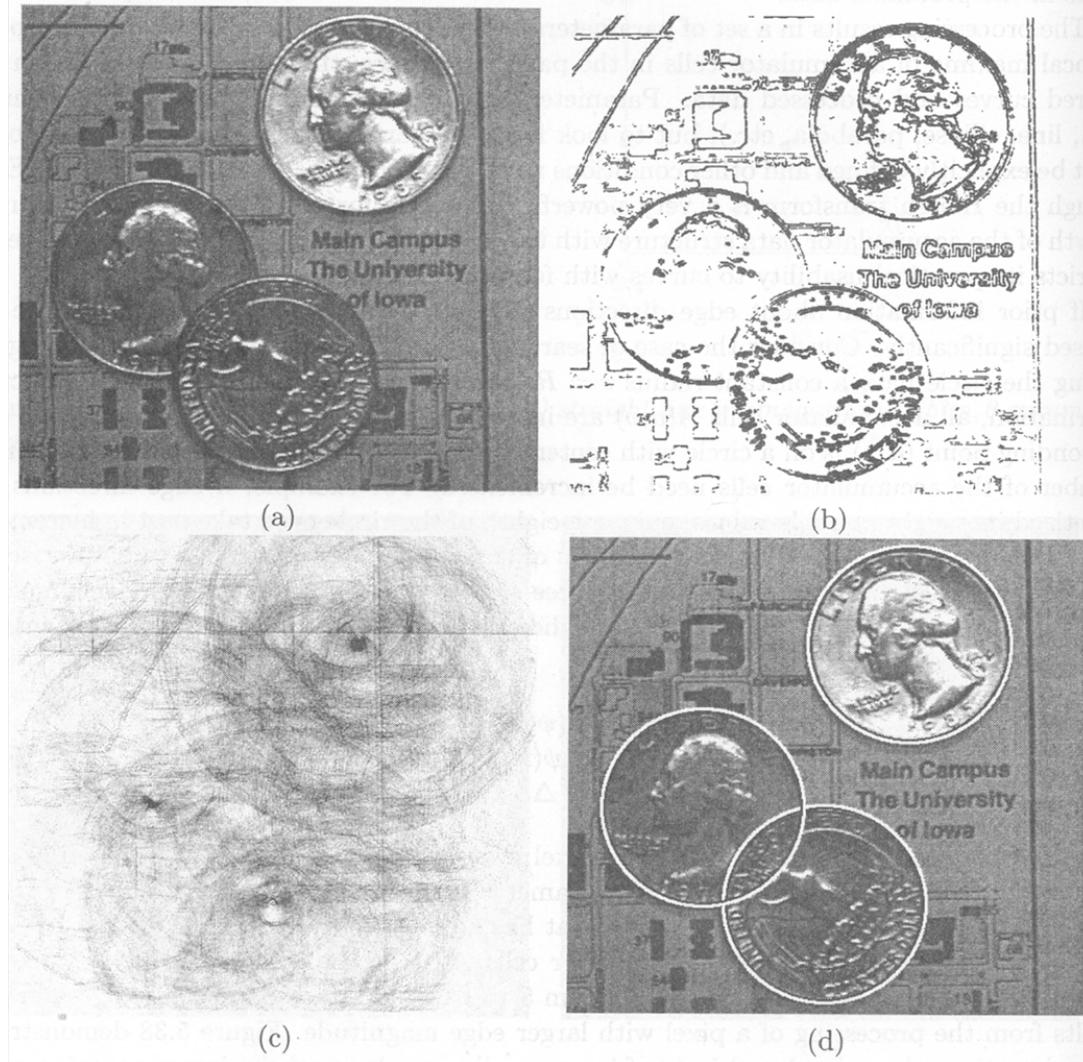
- in dieser Form nur zur Detektion von geraden Konturen geeignet (– aber Verallgemeinerung möglich!)
- Clusterung im Akkumulator kann die "falschen" Geraden entfernen
- großer Speicherbedarf des Akkumulators
- Tracking der Geraden ist rechenaufwändig

verschiedene Verbesserungen des Verfahrens wurden vorgeschlagen

## Verallgemeinerung:

z.B. auf Kreise (aus Sonka et al. 1999):

(a) Original, (b) Kantenbild, (c) Parameterraum, (d) Orig. m. überlagerten Kreisen



verallgemeinerte Hough-Transformation in ähnlicher Weise möglich für Ellipsen, Parabeln...

weitere Verallgemeinerung:

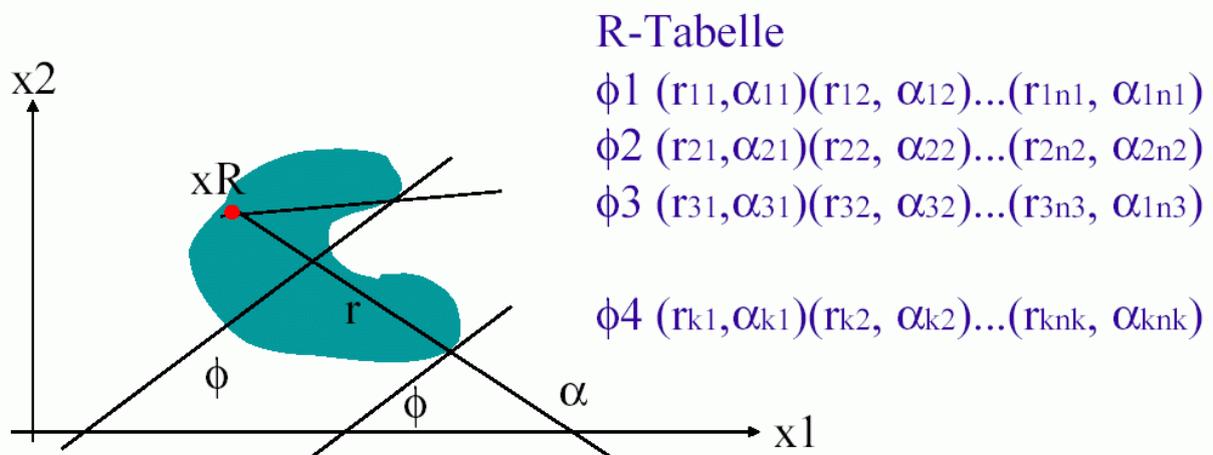
für beliebige (nicht-parametrische), aber fest vorgegebene Form, die im Bild aufgefunden werden soll

⇒

*generalized Hough transform* nach Ballard (s. Sonka et al. 1999):

- in der zu suchenden Form (Kontur) wird ein Referenzpunkt  $R$  festgelegt
- in einem preprocessing-Schritt wird eine R-table (Referenztabelle) angelegt: für jeden von  $R$  ausgehenden Strahl wird der Schnittpunkt mit der Konturlinie der Form bestimmt
- im Schnittpunkt wird die Kontur-Richtung und der Abstand von  $R$  bestimmt und in die Tabelle eingetragen
- die Tabelle wird nach den Kontur-Richtungen (Winkeln) sortiert
- für jeden Kontur-Punkt (mit Richtungsinformation) *im Bild* wird nun anhand der Tabelle ein Kandidaten-Punkt für  $R$  (unter der Hypothese, dass diese Kontur an dieser Stelle durch die gesuchte Form erzeugt wurde) errechnet (evtl. auch mehrere Kandidaten-Punkte)
- die Koordinatenpaare der Kandidaten-Punkte bilden den Hough space (Parameterraum; Akkumulator)
- bei Punkten mit hohem Akkumulator-Eintrag ist tatsächlich die gesuchte Form lokalisiert
- wenn Rotationen und Skalierungen der gesuchten Form zugelassen sind, erhöht sich die Dimension des Hough space von 2 auf 4

zum preprocessing der Konturpunkte (Erstellung der R-table):



*Beachte:*

Nach der Hough-Transformation (einfach oder verallgemeinert) ist die Aufgabe der automatischen Detektion vorgegebener Formen noch nicht gelöst.

- Detektion dunkler Stellen im Akkumulator-Raum zu leisten
- evtl. Zusatzinformation aus dem Bild erforderlich (z.B. Endpunkte von Geradenstücken)

je komplizierter die Form, desto mehr Parameter werden benötigt  $\Rightarrow$  Akkumulator-Raum wird hochdimensional  $\Rightarrow$  Verfahren wird sehr speicher- und rechenintensiv

Einsatz der Hough-Transformation:

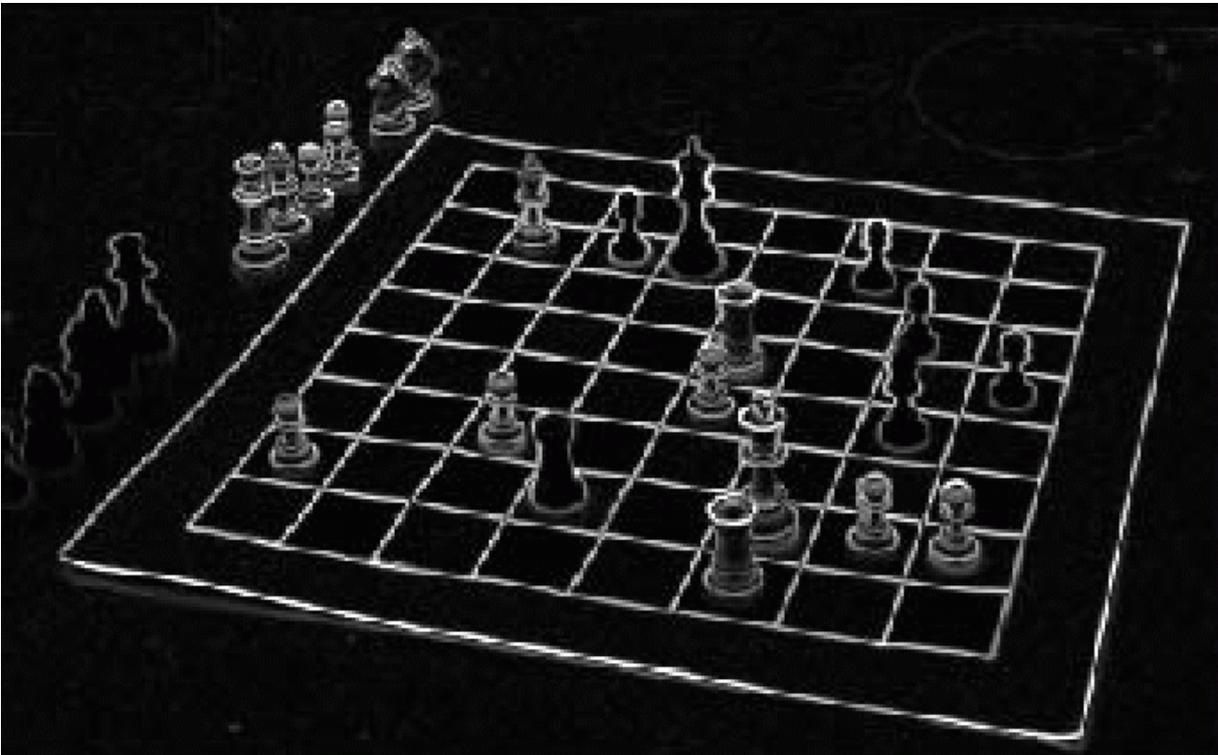
besonders in industriellen Anwendungen

- einfache Formen werden gesucht
- viele Konturen sind Geradenstücke oder Kreisbögen
- Parameter können oft eingeschränkt werden

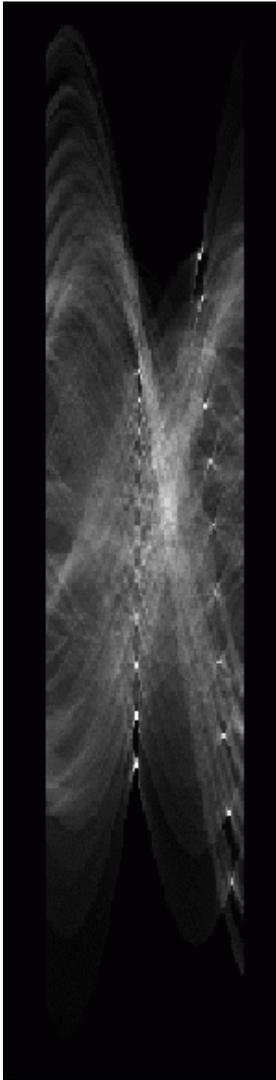
Beispiel Schachbrett (aus Beichel 2002):



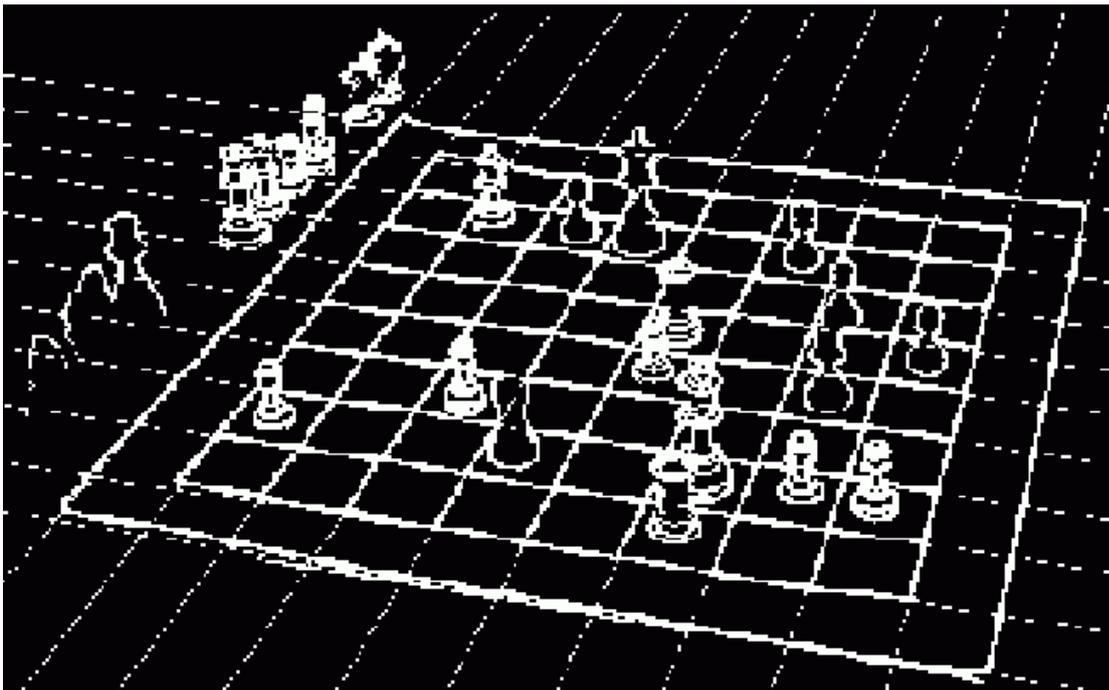
Original-Aufnahme



Kantenbild



Akkumulator (Hough-Space)



Ergebnis der Hough-Transformation (Geraden)

weiterverwendbar für formale Verarbeitung:

