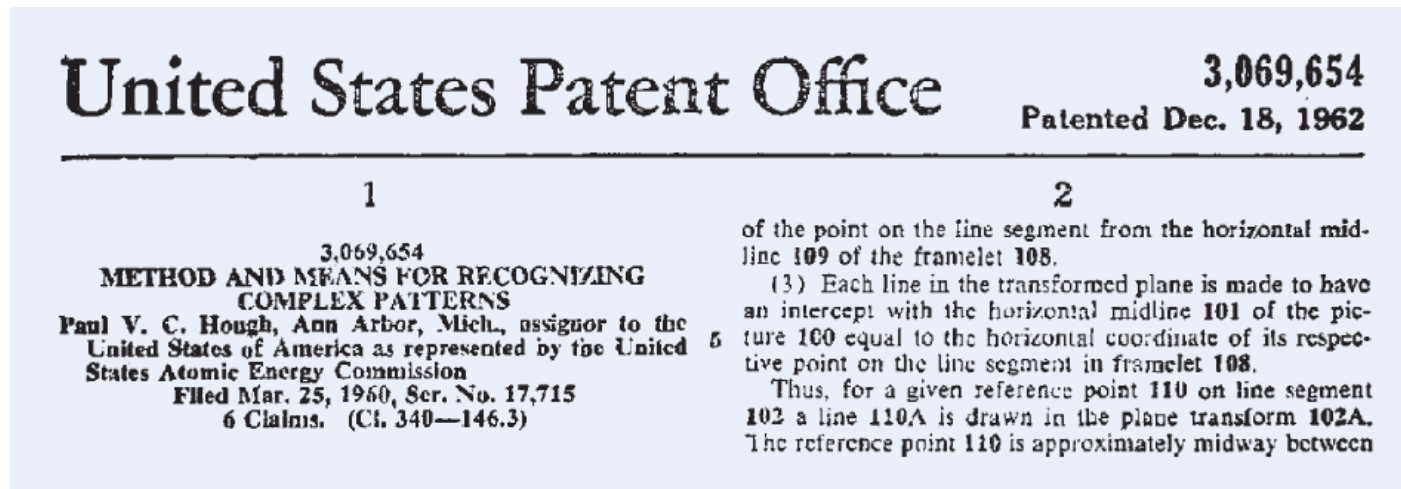


Kapitel 7

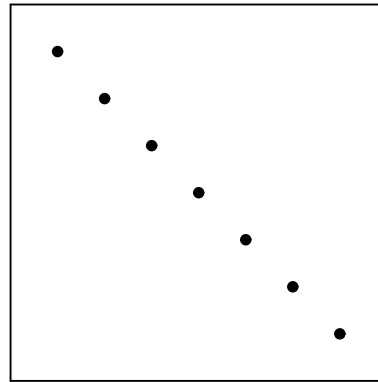
Bildsegmentierung: Hough-Transformation



- Detektion von Geraden
- Detektion von Kreisen
- Detektion beliebiger parametrischer Kurven
- Detektion von Ellipsen
- Probabilistic Hough Transform
- Verallgemeinerte Hough-Transformation

Hough-Transformation (1)

“Globale” Detektion von strukturellen Merkmalen (Geraden, Kreise, Ellipsen, etc.) *ohne Konturverfolgung*; Robustheit gegenüber Störungen, Ungenauigkeit der Kantendetektion und Lücken.



Edge Grouping Problem: The main advantage of the Hough transform technique is that it is tolerant of gaps in feature boundary descriptions and is relatively unaffected by image noise.

Hough-Transformation (2)

Grundprinzip: das binäre Kantenbild $k(r, c)$ in einen anderen Raum (Akkumulator) zu transformieren, wo die Hinweise über das Vorhandensein von Merkmalen gesammelt werden. Auf diese Weise wird aus dem Problem der Merkmalsextraktion das Problem der Maximumfindung im Akkumulator, das i.A. einfacher zu lösen ist.

Outline of Hough Transform:

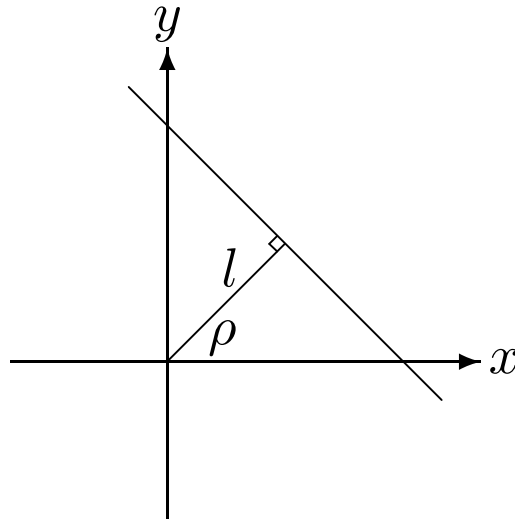
1. Create a grid of parameter values
2. Each edge point votes for a set of parameters, incrementing those values in grid
3. Find maximum or local maxima in grid

Detektion von Geraden (1)

Einführendes Beispiel: Akkumulator mit Punktpaaren

Repräsentation mit der Hesse'schen Normalform:

$$l = x \cos \rho + y \sin \rho; \quad 0 \leq \rho < 2\pi, \quad l \geq 0$$



Es wird ein *diskreter* Akkumulator $H[\rho][l]$ benötigt: sowohl ρ als auch l werden mit endlich vielen Werten:

$$\rho = 0, \Delta\rho, 2\Delta\rho, \dots$$

$$l = 0, \Delta l, 2\Delta l, \dots$$

dargestellt.

Detektion von Geraden (2)

Hough-Transformation für Geraden (basierend auf Kantenpunktpaaren)

Input: Kantenpunkte $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$

Output: Geraden, die durch diese Kantenpunkte verlaufen

setze alle Elemente des Akkumulators $H[\rho][l]$ auf null;

for (alle Paare von Kantenpunkten in $k(r, c)$) {

 berechne die entsprechenden ρ und l ;

$H[\rho][l] ++$; /* Diskretisierung von ρ und l beachten */

}

detektiere Peaks in $H[\rho][l]$;

Es kann davon ausgegangen werden, dass sich im Akkumulator $H[\rho][l]$ Peaks bilden. Jedes Peak entspricht einer Geraden im Bild.

Riesige Anzahl möglicher Kantenpunktpaare \implies großer Aufwand

Detektion von Geraden (3)

Akkumulator mit einzelnen Kantenpunkten:

- Reduzierung des Aufwandes: den Akkumulator $H[\rho][l]$ mit Hinweisen auf mögliche Geraden von einzelnen Kantenpunkten ausgehend aufbauen.
- Leider impliziert ein einziger Kantenpunkt (x_i, y_i) nicht mehr eine eindeutige Gerade, sondern eine Reihe von Geraden (ρ, l) :

$$l = x_i \cos \rho + y_i \sin \rho$$

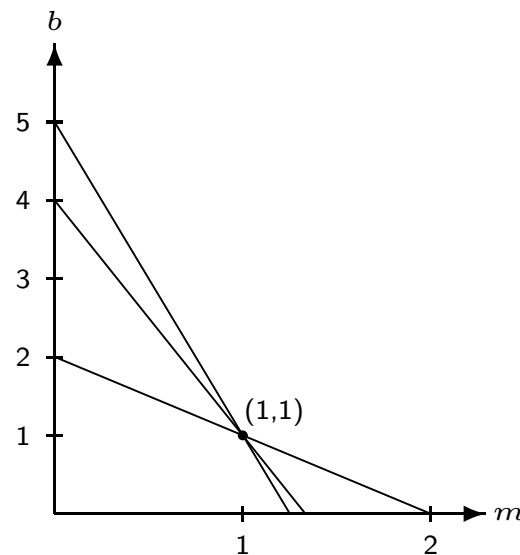
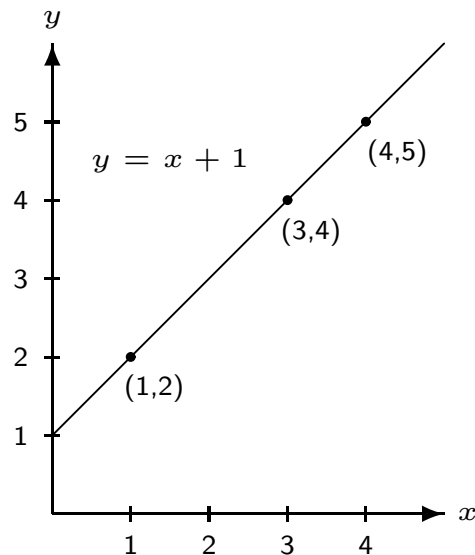
d.h. eine sinusförmige Kurve im ρl -Raum. Jeder Punkt dieser Kurve bedeutet eine Gerade.

- **Lösung:** alle möglichen Parameter (ρ, l) , welche die obige Beziehung erfüllen (Die repräsentierte Gerade passiert somit den betrachteten Kantenpunkt (x_i, y_i)), berücksichtigen und die entsprechenden Zähler im Akkumulator erhöhen.

Detektion von Geraden (4)

Beobachtung:

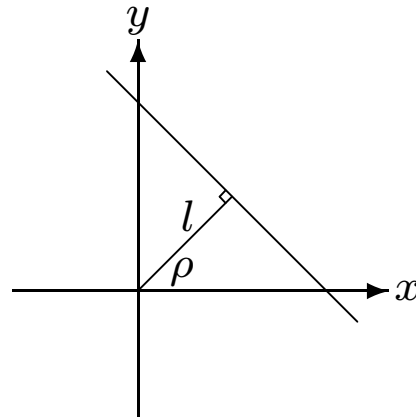
- Eine Gerade $L_{m_0 b_0} : y = m_0 x + b_0$ im Bild kann als ein Punkt (m_0, b_0) im mb -Raum interpretiert werden.
- Ein Kantenpunkt (x_i, y_i) lässt sich durch unendlich viele Geraden mit der Eigenschaft $y_i = m x_i + b$ erzeugen, welche gesamthaft als eine Gerade $b = -x_i m + y_i$ im mb -Raum dargestellt werden können.
- Jeder Punkt einer Geraden $L_{m_0 b_0}$ im Bild führt zu einer Geraden im mb -Raum. Diese Geraden schneiden sich im Punkt (m_0, b_0) .



Die drei Punkte (1,2), (3,4) und (4,5) im Bild links führt jeweils zu einer Geraden im mb -Raum; die drei Geraden dort schneiden sich in (1,1), was auf eine Gerade $y = x + 1$ im Bildraum hindeutet.

Detektion von Geraden (5)

Hesse'sche Normalform: $l = x \cos \rho + y \sin \rho$; $0 \leq \rho < 2\pi$, $l \geq 0$



Ein Parameterpaar (ρ, l) definiert eine Gerade. Bsp. $\rho = 0$, $x = l$ (Gerade parallel zur y -Achse); $\rho = \pi/2$, $y = l$ (Gerade parallel zur x -Achse); parallele Geraden besitzen gleichen ρ -Wert, aber unterschiedliche l -Werte.

- Eine Gerade $L_{\rho_0 l_0} : l_0 = x \cos \rho_0 + y \sin \rho_0$ im Bild kann als ein Punkt (ρ_0, l_0) im ρl -Raum interpretiert werden.
- Ein Kantenpunkt (x_i, y_i) lässt sich durch unendlich viele Geraden erzeugen, welche gesamthaft als eine sinusförmige Kurve

$$l = x_i \cos \rho + y_i \sin \rho$$

im ρl -Raum dargestellt werden können.

- Jeder Punkt einer Geraden $L_{\rho_0 l_0}$ im Bild führt zu einer sinusförmigen Kurve im ρl -Raum. Diese Kurven schneiden sich im Punkt (ρ_0, l_0) .

Detektion von Geraden (6)

Hough-Transformation für Geraden (basierend auf einzelnen Kantenpunkten)

Input: Kantenpunkte $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$

Output: Geraden, die durch diese Kantenpunkte verlaufen

1. Transformiere jeden Kantenpunkt (x_i, y_i) gemäß $l = x_i \cos \rho + y_i \sin \rho$ vom xy -Raum in den ρl -Raum.

for ($\rho = 0; \rho < 2\pi; \rho += \Delta\rho$)

for ($l = 0; l < l_{\max}; l += \Delta l$)

$H[\rho][l] = 0;$

for ($i = 1; i \leq n; i++$)

for ($\rho = 0; \rho < 2\pi; \rho += \Delta\rho$) {

$l = x_i \cos \rho + y_i \sin \rho;$

$H[\rho][l] ++;$ /* Diskretisierung von l beachten */

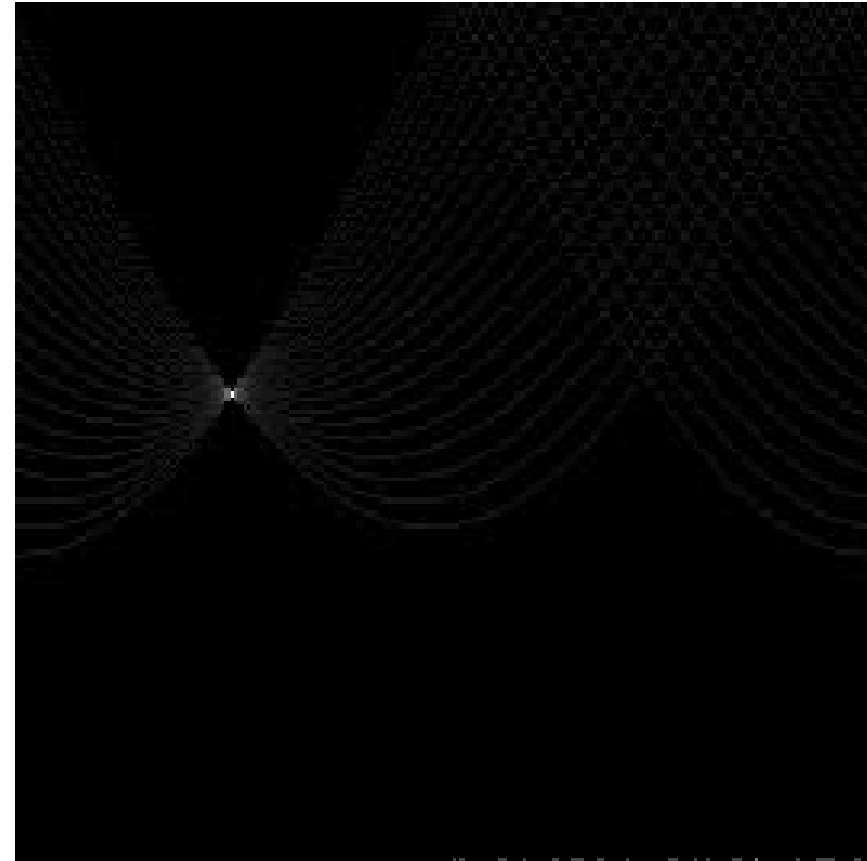
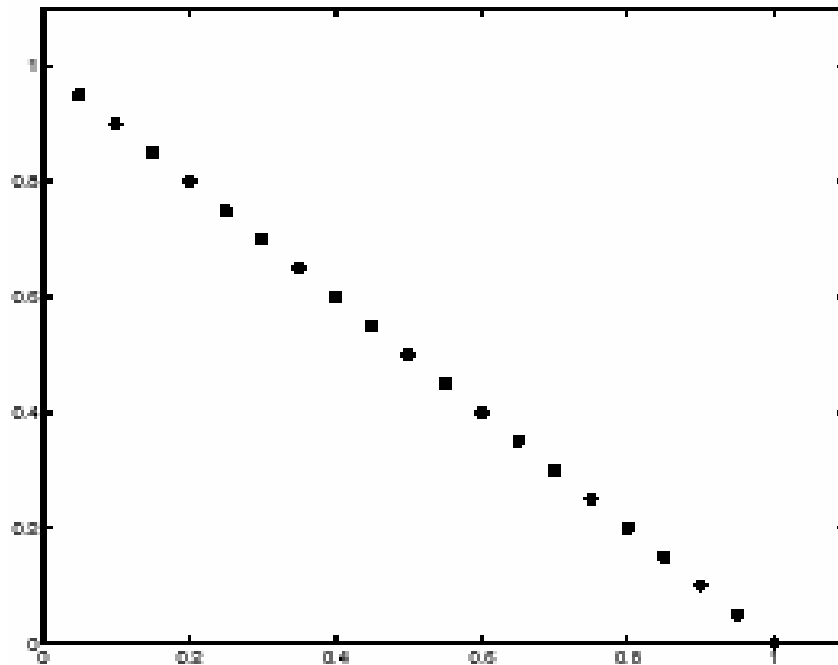
/* Inkrementierung um Kantenstärke $s(x_i, y_i)$ möglich */

}

2. Suche Häufungspunkte im ρl -Raum, d.h. in $H[\rho][l]$. Im einfachsten Fall entspricht ein Häufungspunkt einem lokalen Maximum in $H[\rho][l]$, das einen bestimmten Wert überschreitet. Andere Möglichkeit der Berechnung: Aufsummieren innerhalb eines Fensters, dann Maximum-Suche.
3. Jeder Häufungspunkt (ρ_0, l_0) definiert eine Gerade $l_0 = x \cos \rho_0 + y \sin \rho_0$ im xy -Raum.

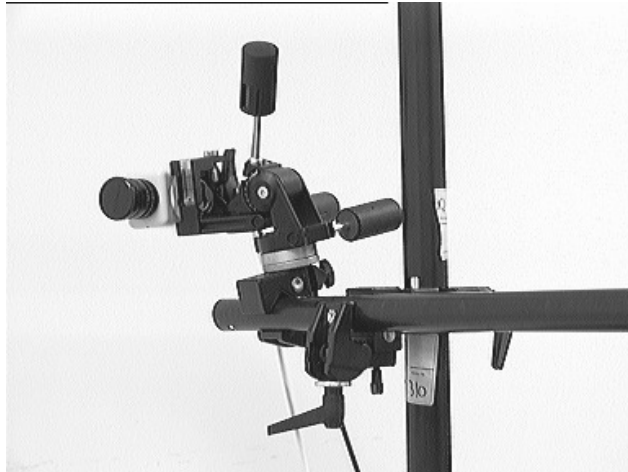
Detektion von Geraden (7)

Beispiel:

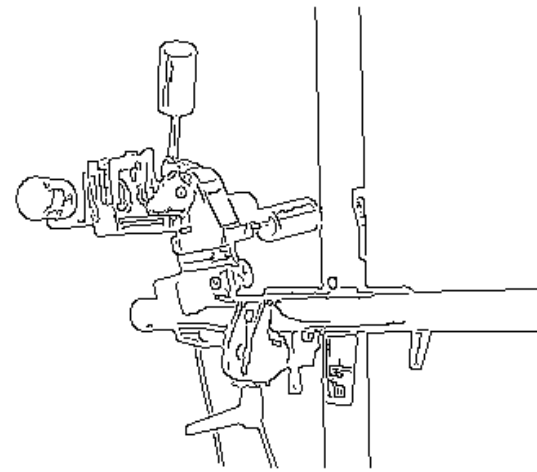


Detektion von Geraden (8)

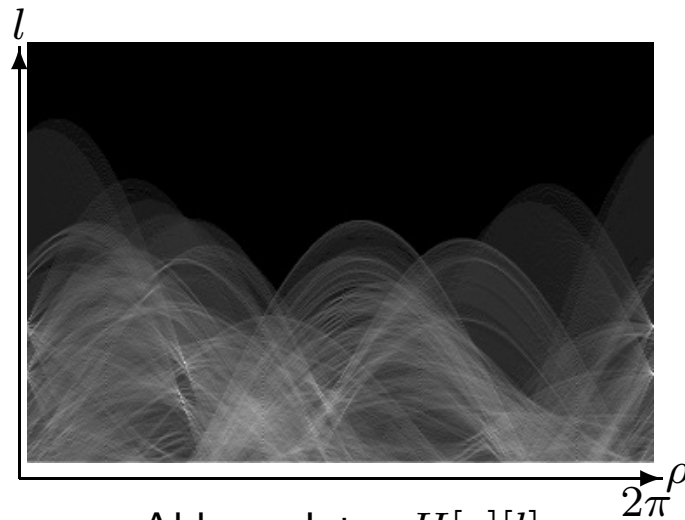
Beispiel:



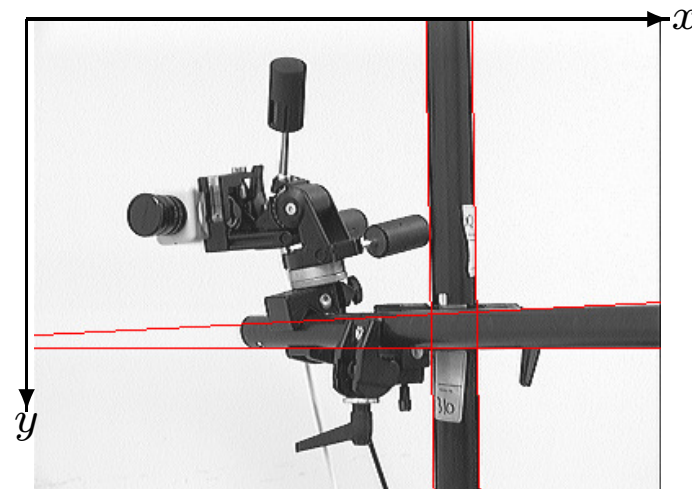
Grauwertbild



Canny ($\sigma = 1.0$, $T_l = 20$, $T_h = 80$)



Akkumulator $H[\rho][l]$



Ergebnis

Detektion von Kreisen (1)

Annahme: Radius r_0 bekannt

- Ein Kreis mit Zentrum (a_0, b_0)

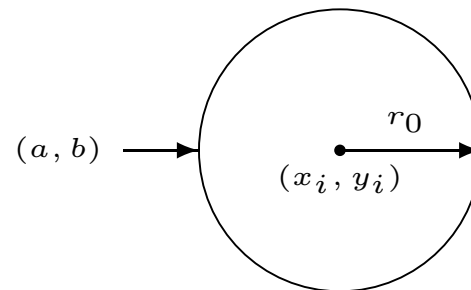
$$C_{a_0 b_0} : \begin{cases} x &= a_0 + r_0 \cos \theta \\ y &= b_0 + r_0 \sin \theta \end{cases} \quad 0 \leq \theta < 2\pi$$

im Bild kann als ein Punkt (a_0, b_0) im ab -Raum interpretiert werden.

- Ein Kantenpunkt (x_i, y_i) lässt sich durch unendlich viele Kreise erzeugen, welche gesamthaft als ein Kreis

$$\begin{cases} a &= x_i - r_0 \cos \theta \\ b &= y_i - r_0 \sin \theta \end{cases} \quad 0 \leq \theta < 2\pi$$

im ab -Raum dargestellt werden können.



- Jeder Punkt eines Kreises $C_{a_0 b_0}$ im Bild führt zu einem Kreis im ab -Raum. Diese Kreise schneiden sich im Punkt (a_0, b_0) .

Detektion von Kreisen (2)

Hough-Transformation für Kreise mit bekanntem Radius

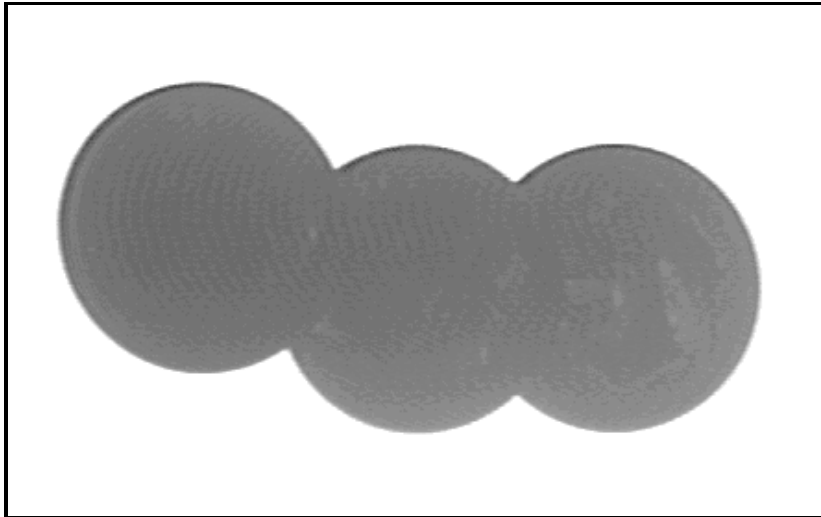
Input: Kantenpunkte $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$

Output: Kreise mit bekanntem Radius r_0 , welche durch diese Kantenpunkte verlaufen

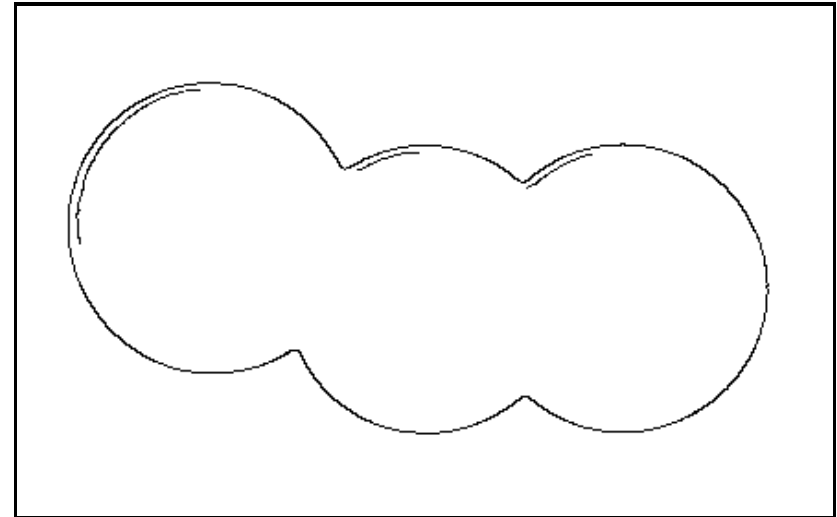
1. Transformiere jeden Kantenpunkt (x_i, y_i) gemäß
$$\begin{cases} a &= x_i - r_0 \cos \theta \\ b &= y_i - r_0 \sin \theta \end{cases} \quad 0 \leq \theta < 2\pi$$
 vom xy -Raum in den ab -Raum. Hierbei Diskretisierung des ab -Raums.
for ($a = a_{\min}; a \leq a_{\max}; a++ = \Delta a$)
 for ($b = b_{\min}; b \leq b_{\max}; b++ = \Delta b$)
 $H[a][b] = 0;$
 for ($i = 1; i \leq n; i++$)
 for ($\theta = 0; \theta < 2\pi; \theta++ = \Delta\theta$) {
 $a = x_i - r_0 \cos \theta; b = y_i - r_0 \sin \theta;$
 $H[a][b]++;$ /* Diskretisierung von a und b beachten */
 /* Inkrementierung um Kantenstärke $s(x_i, y_i)$ möglich */
 }
2. Suche Häufungspunkte im ab -Raum, d.h. in $H[a][b]$. (Strategien analog Geradendetektion)
3. Jeder Häufungspunkt (a_0, b_0) definiert einen Kreis
$$\begin{cases} x &= a_0 + r_0 \cos \theta \\ y &= b_0 + r_0 \sin \theta \end{cases} \quad 0 \leq \theta < 2\pi$$
 im xy -Raum

Detektion von Kreisen (3)

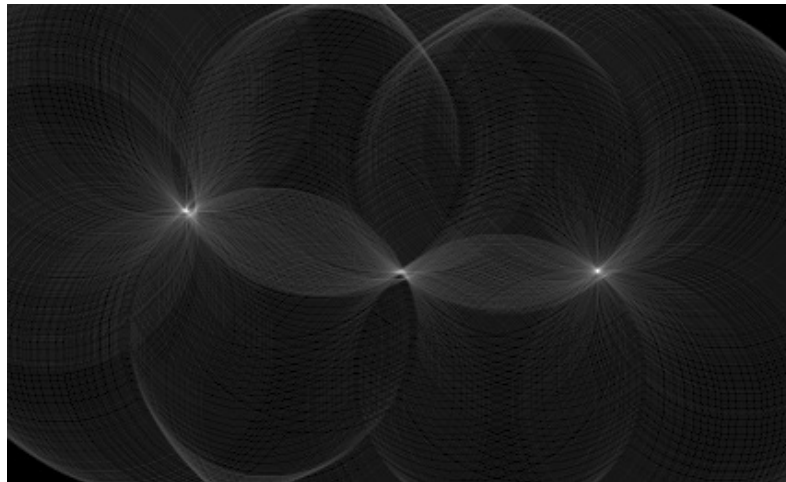
Beispiel:



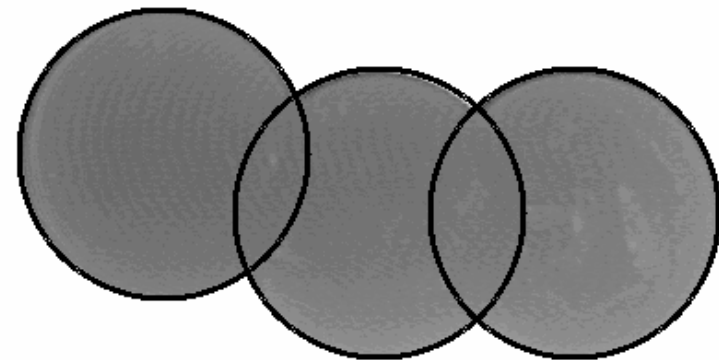
Grauwertbild



Canny ($\sigma = 1.0$, $T_l = 20$, $T_h = 80$)



Akkumulator ($r_0 = 79$)



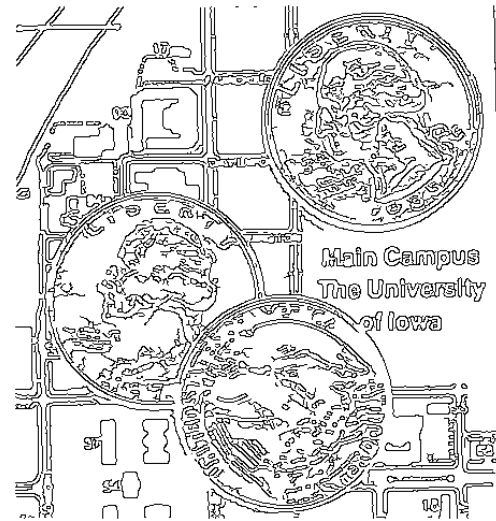
Ergebnis

Detektion von Kreisen (4)

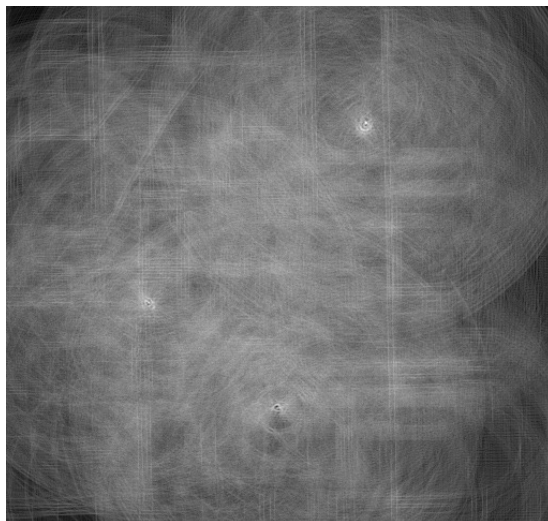
Beispiel:



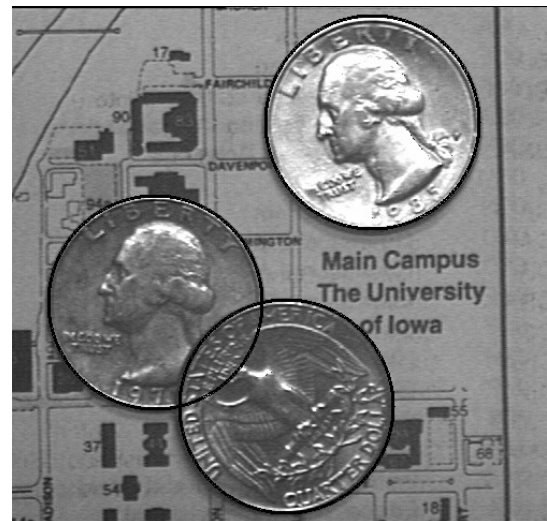
Grauwertbild



Canny ($\sigma = 1.0$, $T_l = 20$, $T_h = 80$)



Akkumulator ($r_0 = 100$)



Ergebnis

Detektion von Kreisen (5)

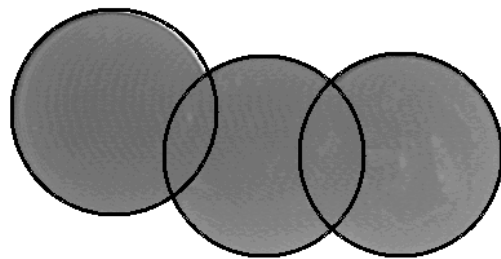
Allgemeiner Fall: Radius r_0 unbekannt

Jeder Punkt (x_i, y_i) eines Kreises mit Zentrum (a_0, b_0) und Radius r_0 im Bild führt zu einem Kegel

$$\begin{cases} a = x_i - r \cos \theta \\ b = y_i - r \sin \theta \end{cases} \quad 0 \leq \theta < 2\pi$$

im abr -Raum. Diese Kegel schneiden sich im Punkt (a_0, b_0, r_0) .

Algorithmus analog; 3-D Akkumulator $H[a][b][r]$; größerer Speicher- und Rechenbedarf als bei bekanntem Radius r_0 .



Radius 78/79/80



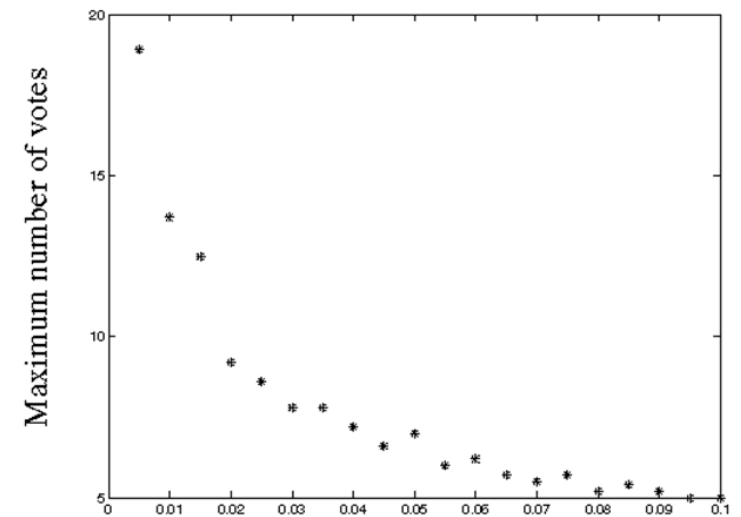
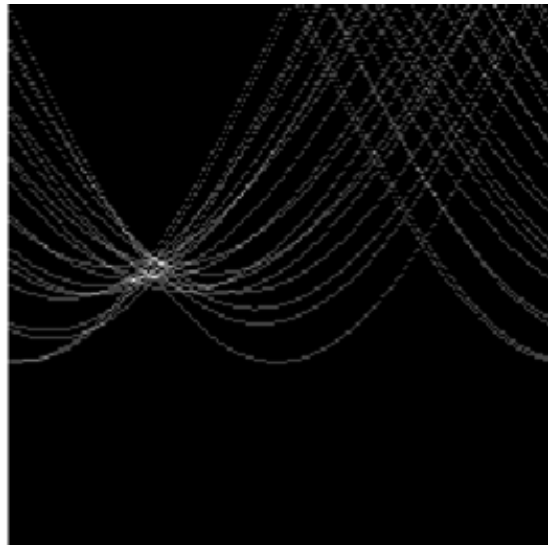
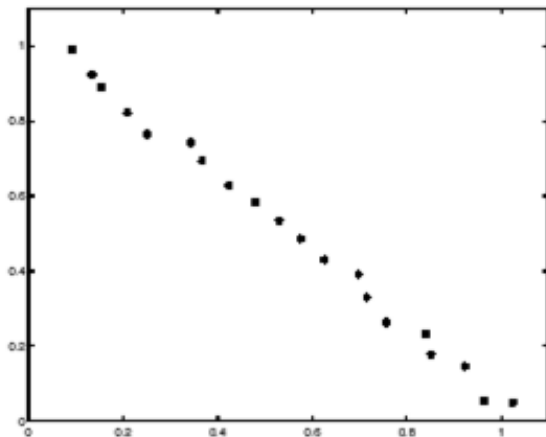
Radius 98/98/101

Effect of noise (1)

Example: Taken from J. Hays

Left: Peak gets fuzzy and hard to locate

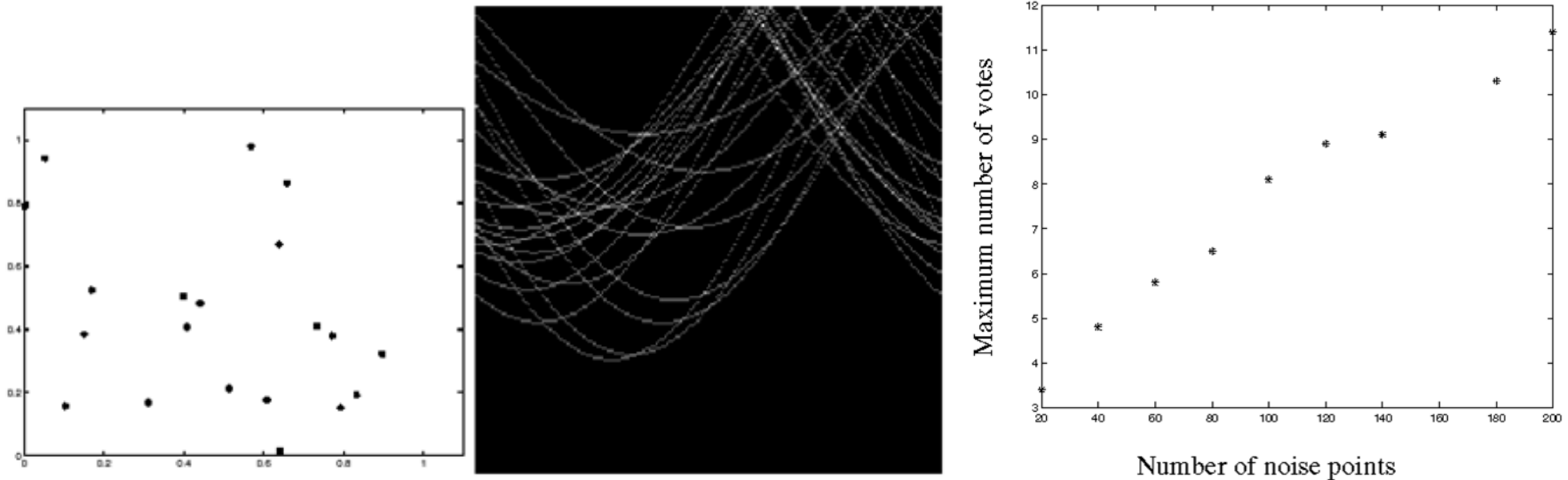
Right: Number of votes for a line of 20 points with increasing noise



Effect of noise (2)

Left: Uniform noise can lead to spurious peaks in the array

Right: As the level of uniform noise increases, the maximum number of votes increases too



Effect of noise (3)

Dealing with noise:

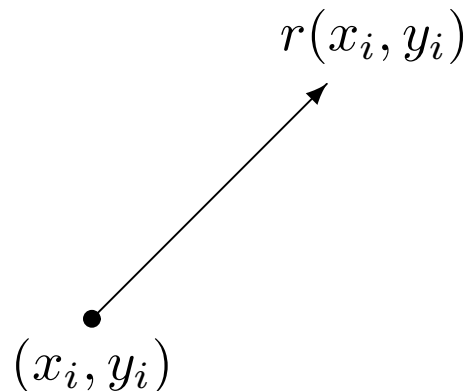
- Choose a good grid / discretization
 - Too coarse: large votes obtained when too many different lines correspond to a single bucket
 - Too fine: miss lines because some points that are not exactly collinear cast votes for different buckets
- Increment neighboring bins (smoothing in accumulator array)
- Try to get rid of irrelevant features: Take only edge points with significant gradient magnitude
- Consider edge orientation information

Verwendung von Richtungsinformation (1)

Beobachtung:

- I.a. ist die Konturrichtung in detektierten Kantenpunkten bekannt; vgl. Sobel-, Canny-Operator, etc.
- Die Richtungsinformation kann bei der Hough-Transformation verwendet werden, um die Freiheitsgrade bei der Besetzung des Akkumulators H zu reduzieren

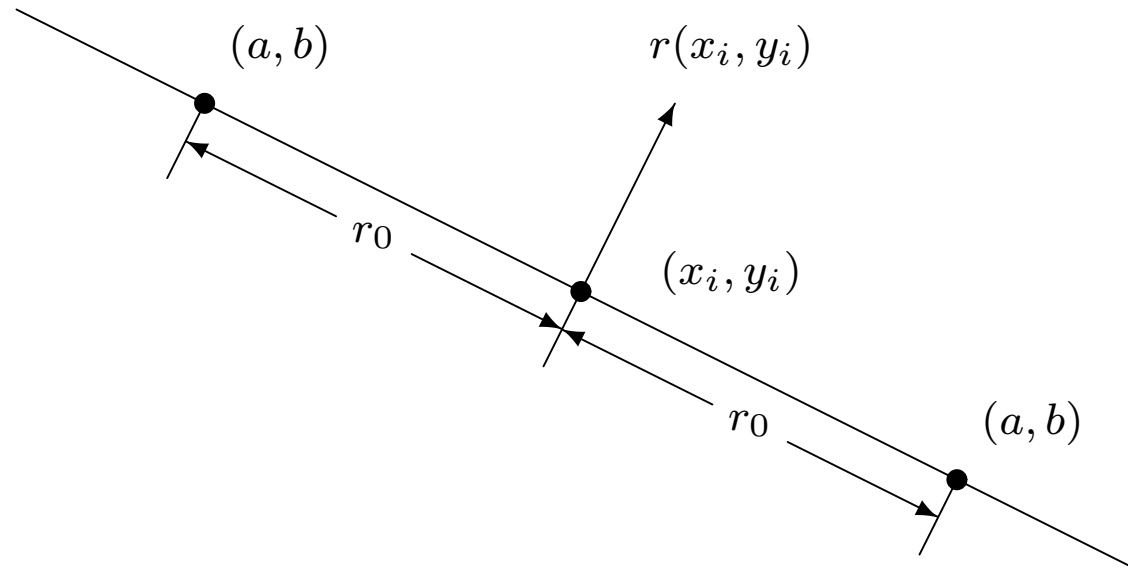
Gerade:



Parameter ρ und l durch Kantenrichtung $r(x_i, y_i)$ eindeutig bestimmt
 \implies nur eine Zelle im Akkumulator $H[\rho][l]$ inkrementieren (wegen Ungenauigkeit von $r(x_i, y_i)$ evtl. einen Bereich $r(x_i, y_i) \pm \gamma$ berücksichtigen)

Verwendung von Richtungsinformation (2)

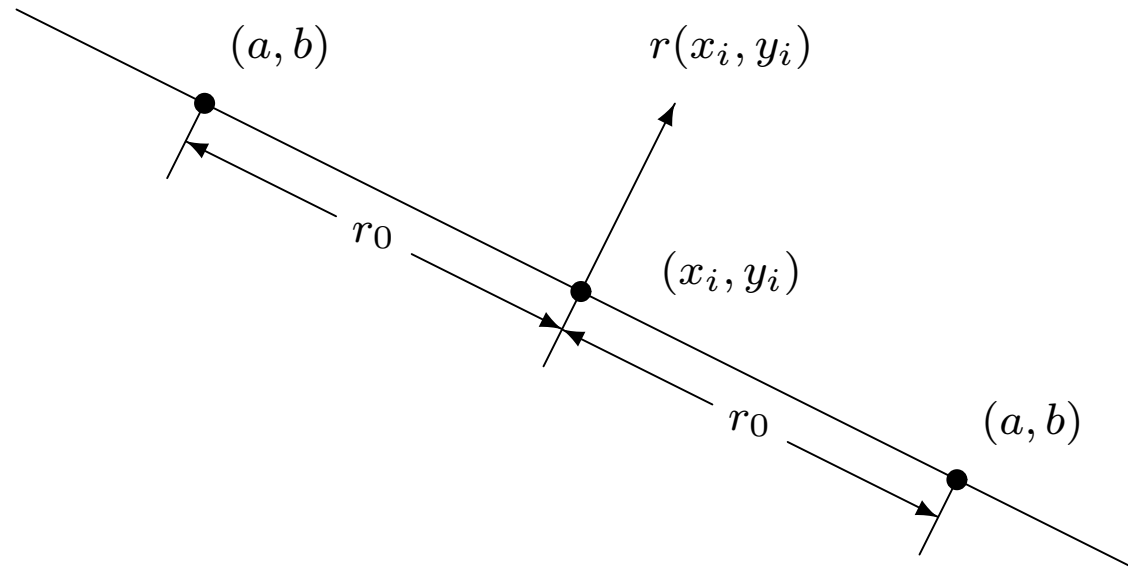
Kreis:



Das Zentrum (a, b) liegt auf einer durch (x_i, y_i) verlaufenden Geraden senkrecht zur Kantenrichtung $r(x_i, y_i)$; bei bekanntem Radius r_0 nur zwei Möglichkeiten.

Verwendung von Richtungsinformation (2)

Kreis:

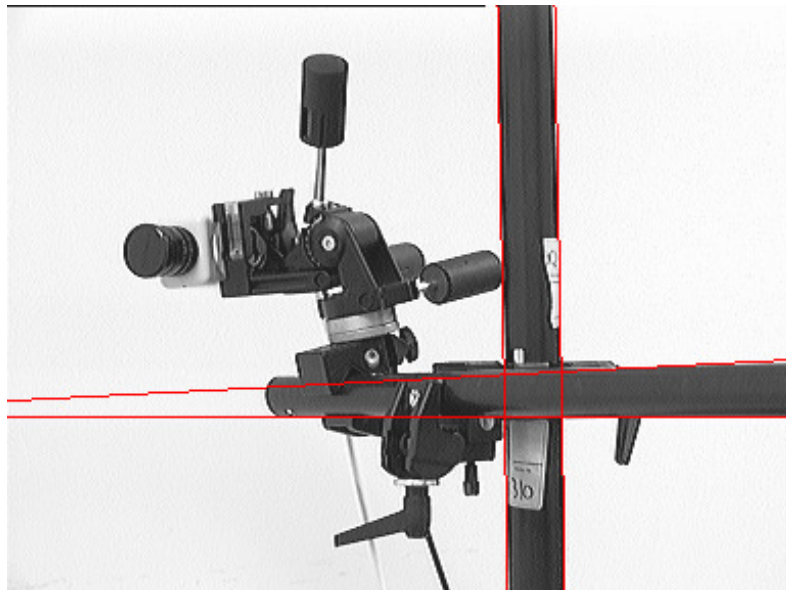
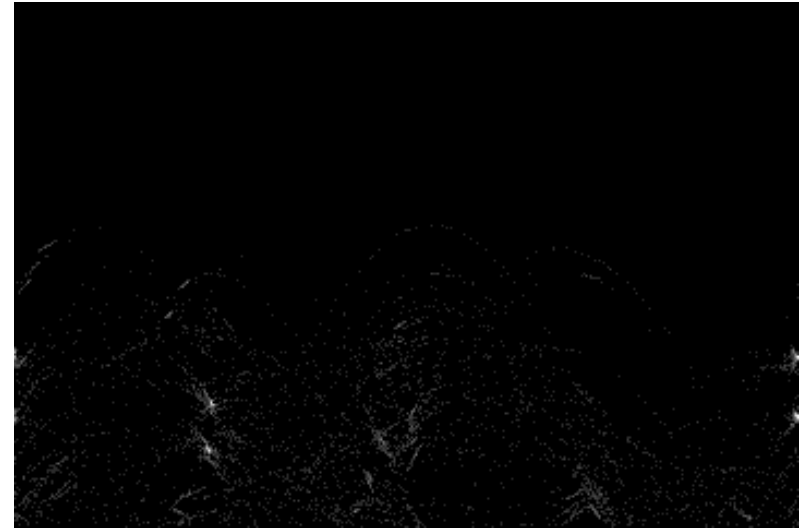
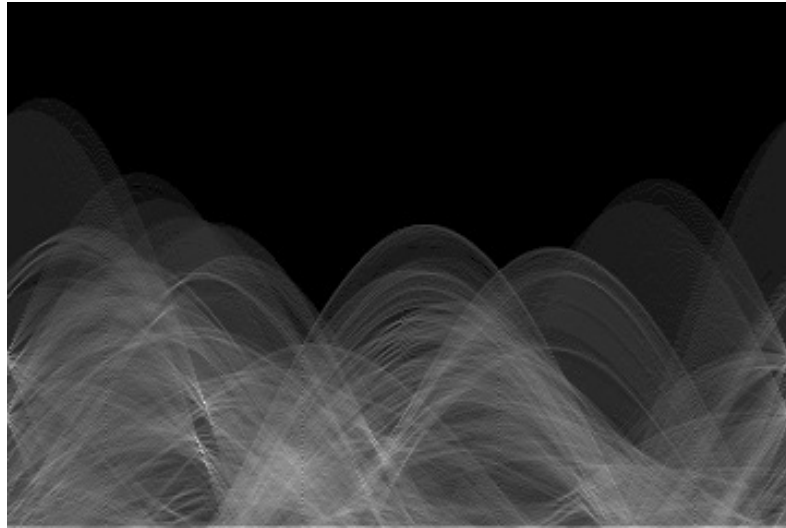


Das Zentrum (a, b) liegt auf einer durch (x_i, y_i) verlaufenden Geraden senkrecht zur Kantenrichtung $r(x_i, y_i)$; bei bekanntem Radius r_0 nur zwei Möglichkeiten.

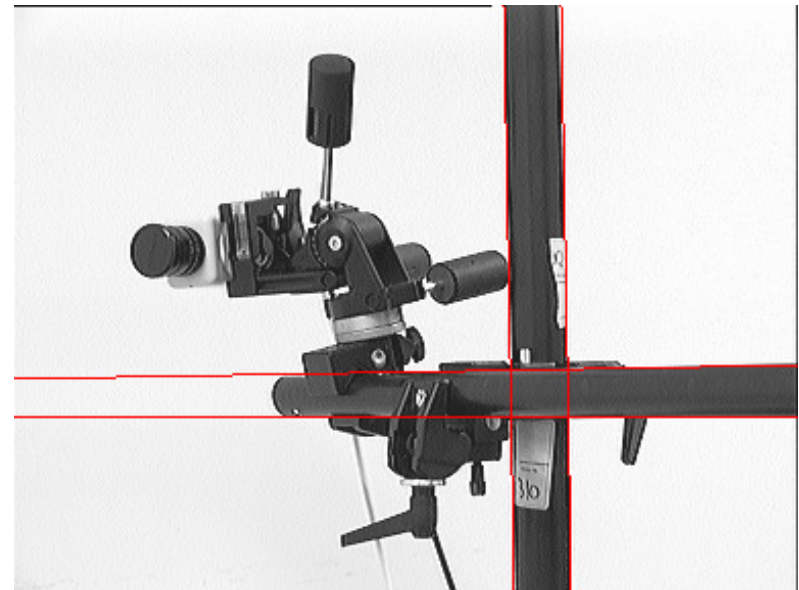
Quiz: Wann gibt es nur eine einzige Möglichkeit?

Verwendung von Richtungsinformation (3)

Beispiel:



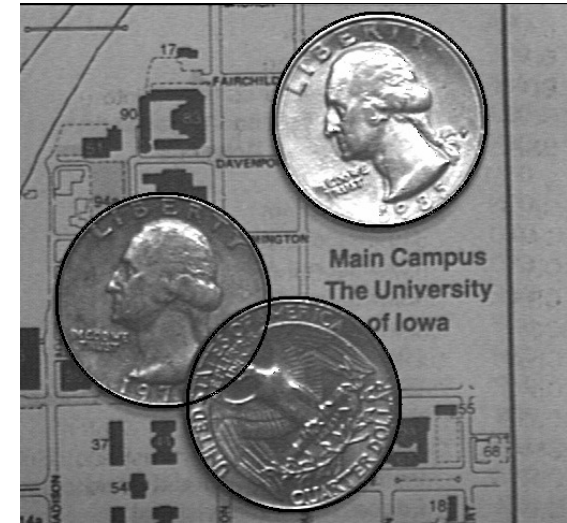
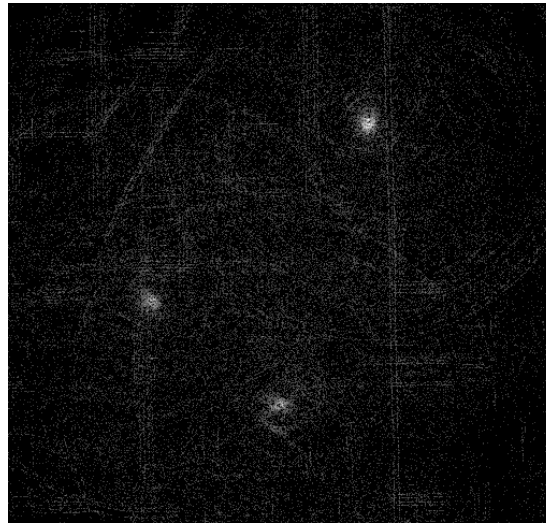
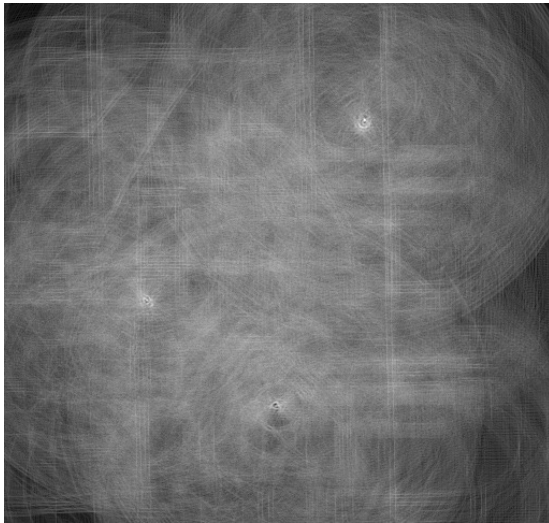
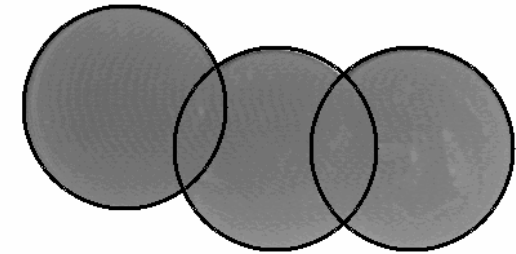
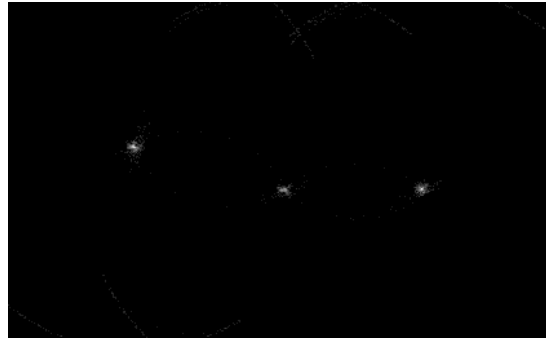
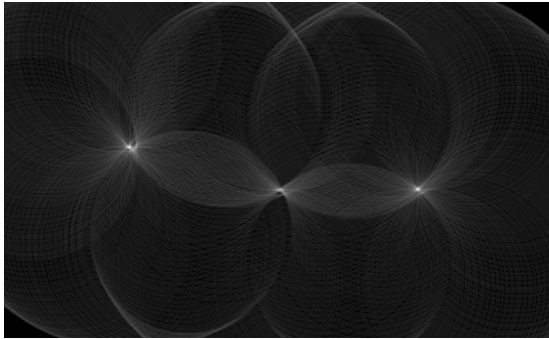
ohne Richtungsinformation



mit Richtungsinformation

Verwendung von Richtungsinformation (4)

Beispiel:



Akkumulator:
ohne Richtungsinformation

Akkumulator:
mit Richtungsinformation

Ergebnis

Verallgemeinerung auf beliebige parametrische Kurven

Ziel: Detektion einer beliebigen Kurve $g(x, y, \vec{a}) = 0$ mit Parametervektor $\vec{a} = (a_1, a_2, \dots, a_m)$

Beispiel: Gerade $x \cos \rho + y \sin \rho - l = 0$, $\vec{a} = (\rho, l)$

Algorithmus:

Diskretisierung: $a_k \in A_k$, $k = 1, 2, \dots, m$

Durchführung für jeden Kantenpunkt (x_i, y_i) :

/* alle \vec{a} mit Eigenschaft $g(x_i, y_i, \vec{a}) = 0$ erzeugen */

for ($a_1 \in A_1$)

for ($a_2 \in A_2$)

 ...

for ($a_{m-1} \in A_{m-1}$) {

 Berechnung von a_m gemäß $g(x_i, y_i, \vec{a}) = 0$;

$H[a_1][a_2] \dots [a_m]++$ für jede a_m -Lösung;

 /* Diskretisierung von a_m beachten */

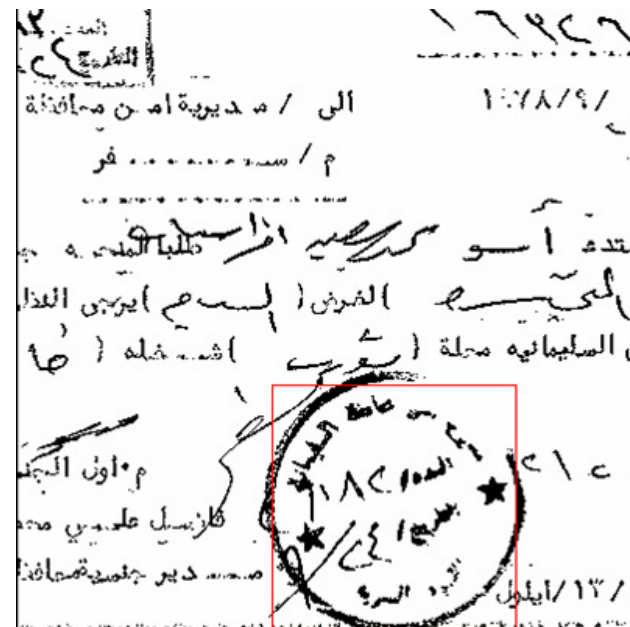
 /* Inkrementierung um Kantenstärke $s(x_i, y_i)$ möglich */

 }

Zeitkomplexität: $O(\# \text{Kantenpunkte} \cdot N^{m-1})$ falls $|A_k| = N$; Reduktion auf $O(\# \text{Kantenpunkte} \cdot N^{m-2})$ bei Verwendung von Richtungsinformation

Bemerkung: hoher Speicher- und Rechenaufwand bei vielen Parametern; Abhilfe schafft die Verwendung von Richtungsinformation

Detektion von Ellipsen (1)



G. Zhu, S. Jaeger, D. Doermann: A robust stamp detection framework on degraded documents. Int. Conf. on Document Recognition and Retrieval (IS&T, SPIE), January 2006

- Majority of stamps in our Arabic document databases are elliptic or circular objects
- Shape distortions due to a variety of factors alter the eccentricity of the object, making even a perfect circle elliptic

Detektion von Ellipsen (2)



Quiz: Warum ist die Detektion von Ellipsen so wichtig?

Detektion von Ellipsen (3)

Parametrisierung:

$$\frac{[(x - x_0) \cos \phi + (y - y_0) \sin \phi]^2}{a^2} + \frac{[(x - x_0) \sin \phi - (y - y_0) \cos \phi]^2}{b^2} = 1$$

Sowohl der Speicherbedarf als auch der Rechenaufwand ist wegen des 5-dimensionalen Problemraums enorm hoch.

Lösung: Aufteilung der fünf Parameter. Z.B. zuerst das Zentrum (x_0, y_0) bestimmen und in einer zweiten Phase die restlichen drei Parameter berechnen.

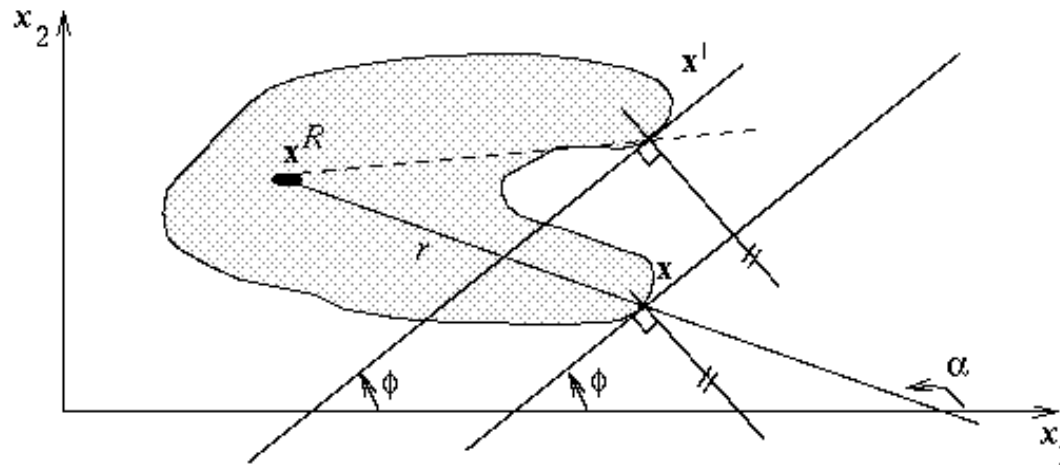
Probabilistic Hough Transform

N. Kiryati, Y. Eldar, A.M. Bruckstein: *A probabilistic Hough transform*. Pattern Recognition, 24(4): 303–316, 1991

- Rather than using all M edge points, only a subset m ($m \ll M$; 15%-25%) is used.
- Intuitively, this works because a random subset of M will fairly represent all features and noise.
- An analysis shows the presence of a thresholding effect for the value of m . Values of m below the threshold gave poor results whilst values above gave very good results. This threshold effect was confirmed experimentally with good results being obtained with as few as 2% of edge points being sampled. However, the value of m must be determined on a per problem basis.
- This strategy is particularly important when dealing with n -tuples (pair, triple, etc.) of edge points.

Verallgemeinerte Hough-Transformation

Ziel: Detektion einer beliebigen Kurve *ohne* mathematische Beschreibung



Die zu detektierende Kurve soll als eine Folge von diskreten Punkten vorliegen. Die Forminformation wird durch eine R -Tabelle abgespeichert:

$$\phi_1 : (\alpha_1, r_1), \dots; \quad \phi_2 : (\alpha_2, r_2), \dots; \quad \dots : \dots; \quad \phi_k : (\alpha_k, r_k), \dots$$

Speziell für den Fall fixer Größe und Orientierung kann leicht ein Akkumulator für einen vordefinierten "Mittelpunkt" x^R aufgebaut werden.

Weitere Erweiterungen sind z.B. Detektion regulärer und nichtregulärer Polygone (ein Polygon als Ganzes zu erkennen). Eine gute Quelle für Hough-Transformationen liefert das Buch von Davies, 2005.

- J. Illingworth and J. Kitler: *A survey of the Hough transform*. Computer Vision, Graphics and Image Processing, 44(1): 87–116, 1988
- Real-Time Imaging, Special Issue on the Hough Transform, 6(2), 2000
- E.R. Davies: *Machine Vision: Theory, Algorithms, Practicalities*, 3rd Edition, Elsevier, 2005
- P.E. Hart: *How the Hough transform was invented*. IEEE Signal Processing Magazine, 26(6): 18–22, 2009