

Praktische Übung: Computer Vision für medizinische und industrielle Anwendungen

Versuch 2: Globale Operatoren

Erinnerung - Hough-Transformation

Für die Hough-Transformation eines (Kanten-)Bildes sind die folgenden Schritte notwendig:

- (a) Darstellung der gesuchten Kurve in der Form $f(\mathbf{x}, \mathbf{a}) = 0$ mit Position \mathbf{x} und Parametervektor \mathbf{a}
- (b) Quantisierung des Parameterraums und Initialisierung der Akkumulatormatrix A
- (c) Für jeden Kantenpixel \mathbf{x}_e : Erhöhung der Akkumulatorzellen $A(\mathbf{a})$, wenn $f(\mathbf{x}_e, \mathbf{a}) = 0$
- (d) Bestimmung der (lokalen) Maxima von A

Die lokalen Maxima $\tilde{\mathbf{a}}_i$ zeigen die (wahrscheinlichsten) Kurven $f(\mathbf{x}, \tilde{\mathbf{a}}_i)$ im Bild an.

Einleitung

Starten Sie die Funktion `hough_demo.p`. Hier können Sie an einem einfachen Beispiel die Hough-Transformation zur Detektion von Geraden und Kreisen ausprobieren. Im linken Bild können Sie manuell Kantenpixel definieren, die für die Hough-Transformation verwendet werden. Ein Linksklick setzt einen Punkt, ein Rechtsklick löscht einen Punkt. Die Leertaste löscht das gesamte Bild. Das Programm wird mit Esc abgebrochen. In den weiteren Bildern werden die Akkumulatormatrizen dargestellt. Ein Linksklick hier zeichnet die zugehörige Funktion (Gerade oder Kreis) im Originalbild. Die Tasten 1 und 2 detektieren automatisch die Maxima.

Aufgabe 1 - Hough-Transformation für Geraden

Vervollständigen Sie das Script `houghTransform.m`, das Geraden in einem Bild detektiert und diese in das Bild plottet. Folgen Sie den Anweisungen im Script und setzen Sie die folgenden Teilaufgaben um:

- Erstellung des Kantenbildes
- Hough-Transformation, Bestimmung der Akkumulatormatrix
- Bestimmung der (lokalen) Maxima in der Akkumulatormatrix
- Darstellung der Geraden im Bild

Nutzen Sie MATLAB-eigene Funktionen (`hough`, `houghpeaks`), die nach Geraden in der Form

$$f(x, y; \theta, \rho) = x \cos(\theta) + y \sin(\theta) - \rho = 0$$

suchen. Zur Darstellung der Geraden steht eine Funktion `getEndpoints(I, theta, rho)` zur Verfügung, die die Start- und Endpunkte der implizit gegebenen Geraden im Bild zurückgibt. Sie können die Geraden in MATLAB über den Befehl `plot` darstellen. Achten Sie darauf, dass in Matlab immer x - gegen y -Werte geplottet werden und Bilder in einem ij -Koordinatensystem angezeigt werden.

Das Testbild zeigt ein Bauteil mit unterschiedlich langen Drahtverbindungen. Ziel ist die Detektion der Verbindungen als Geraden im Bild. Wie wirkt sich die Methode zur Kantendetektion auf das Ergebnis aus? Testen Sie die Geradendetektion auch an anderen Bildern.

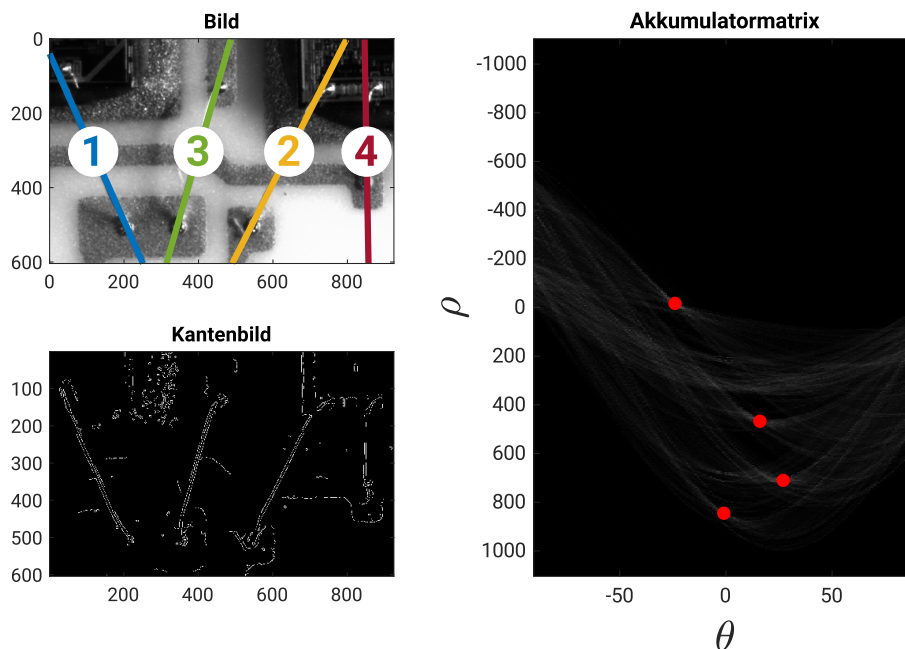


Abbildung 1: Beispielhafte Visualisierung der Ergebnisse aus Aufgabe 1

Aufgabe 2 - Hough-Transformation für Kreise

Implementieren Sie nun eine eigene Funktion zur Detektion von Kreisen in der Form

$$f(x, y; m, n, r) = (x - m)^2 + (y - n)^2 - r^2 = 0.$$

- a) In der Akkumulatormatrix werden für jeden Kantenpixel die Elemente erhöht, die einem Parameterset entsprechen, das den Kantenpixel erzeugt haben könnte. Sollen Kreise detektiert werden, kann jeder Kantenpixel durch einen Kreis mit dem Mittelpunkt im Abstand des Radius r entstanden sein. Die zu erhöhenden Parameter-Sets in der Akkumulatormatrix sind demnach ebenfalls als Kreise mit Radius r darstellbar, wobei die Menge an Punkten nur abhängig von der Lage des Kantenpixel ist. Zur Beschleunigung sollte diese Menge an Punkten als Maske vorab initialisiert werden.

Vervollständigen Sie daher zunächst die Funktion `getAccumulatorUpdate.m`, die diese dreidimensionale Maske in Abhängigkeit des betrachteten Radius bestimmt. Ihre Maske sollte wie folgt aussehen:

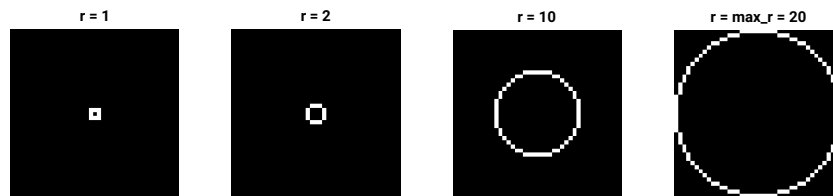


Abbildung 2: ausgewählte 2D-Ansichten bei $r_{max} = 20$

Hier wird für jeden Radius $r = 1, \dots, r_{max}$ eine zweidimensionale Maske (ein Bild mit den Werten 0 und 1) erstellt, in der die Pixel markiert werden, deren (gerundeter) Abstand zum Mittelpunkt r beträgt. Die einzelnen Masken können zu einer dreidimensionalen Maske zusammengefasst werden, die Sie nun in der Akkumulatormatrix an den geeigneten Stellen addieren können.

- b) Vervollständigen Sie die Funktion `houghCircle.m`. Achten Sie auf eine geeignete Aufteilung des Parameterraums, den Sie in der Akkumulatormatrix darstellen wollen und eine geeignete Randbehandlung.

Hinweis: Die einfachste Lösung ist, Kantenpixel, die zu nah am Rand sind, zu ignorieren.

- c) Nutzen Sie Ihre Hough-Transformation, um Zellen in dem Bild `Zellen.jpg` zu markieren. Schreiben Sie dazu ein Testskript, das das Bild einliest, ein Kantenbild erstellt und anschließend die gefundenen Kreise im Bild darstellt, indem Sie die Funktion `plotCircle(m, n, r)` nutzen. Stellen Sie zusätzlich auch ein Histogramm der Radien der gefundenen Kreise dar (`histogram`). Farbbilder müssen zunächst in Graustufenbilder umgewandelt werden (MATLAB: `rgb2gray`).

- d)* Die Hough-Transformation findet oft mehrere ähnliche Kreise. Versuchen Sie das Problem zu beheben, indem Sie einen größeren Bereich um gefundene Maxima unterdrücken.

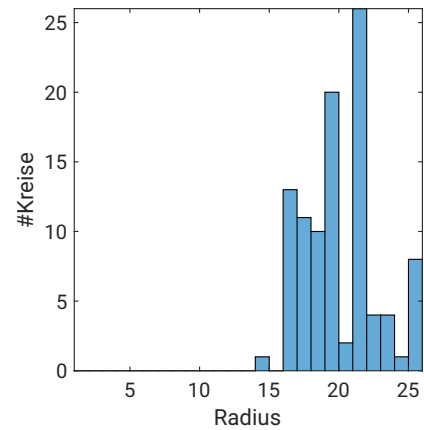
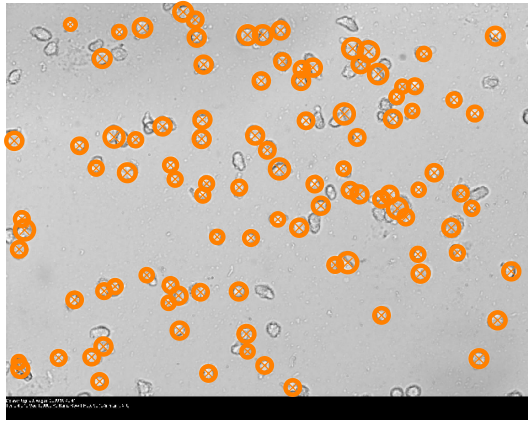


Abbildung 3: Beispielhafte Visualisierung der Ergebnisse aus Aufgabe 2

Kontrollfragen

- Was beschreiben die Parameter θ und ρ in der gewählten Geradendarstellung in Aufgabe 1? Was sind die Vorteile gegenüber anderen Parametrisierungen, wie z.B. über die Steigung und den Schnittpunkt mit der y -Achse?
- Was ist die Akkumulatormatrix?
- Warum werden tendenziell eher große als kleine Kreise im Bild gefunden? Wie kann das Problem gelöst werden?
- Warum wird die Hough-Transformation selten für Objekte mit mehr als 3 Parametern verwendet?