
Übung zur Vorlesung

Rechnersehen 1

WS 2018/2019

Übungsblatt 5: Bildsegmentierung

Ausgabe: 9.1.2019

Abgabe: 23.1.2019

Aufgabe 1 Hough-Transformation für Geraden

(6 Punkte)

Zur Detektion parametrisch beschreibbarer Objekte in Kantenbildern kann die *Hough-Transformation* verwendet werden. Als *Hough-Akkumulator* wird dazu im Folgenden ein diskretisierter Parameterraum bezeichnet.

1. Bei der *Geradendetektion* wird hierfür sinnvoller Weise die *Polardarstellung* gewählt. Dabei wird eine Gerade durch ihren Normalenwinkel φ und dem Abstand r zum Ursprung beschrieben, wie in Abbildung 1 angedeutet. Für jeden Punkt $\mathbf{p} = (x, y)$ im Kantenbild wird dann für alle Winkel φ der Parameter

$$r = x \cos \varphi + y \sin \varphi \quad (1)$$

einer Geraden durch diesen Kantenpunkt bestimmt. Die Einträge an den entsprechenden Stellen (φ, r) im Hough-Akkumulator werden inkrementiert. Lokale Maxima beschreiben letztendlich diejenigen Parameterkombinationen, die Geraden im Bild darstellen, auf denen viele Kantenpunkte liegen.

2. Implementieren Sie eine Python-Funktion, welche in einem gegebenen Kantenbild die ersten $n = 50$ Geraden detektiert und stellen Sie diese dar! Stellen Sie sicher, dass die einzelnen Parameter leicht verändert werden können! Visualisieren Sie ebenfalls den Hough-Akkumulator und zeigen Sie die Zusammenhänge zum Detektionsergebnis! Welchen Einfluss hat die Wahl der Größen für die Diskretisierungsschritte auf das Ergebnis?
3. Das vorgestellte Verfahren weist einige Schwächen auf, welche durch einfache Erweiterungen abgeschwächt werden können. So können die konkreten Kantenstärken beim Inkrementieren der Akkumulatorzellen berücksichtigt werden. Darüber hinaus würde eine Glättung des Akkumulators zu eindeutigen, unverrauschten Maxima führen.

Erweitern Sie Ihre Routine derart, dass sie robust gegenüber Ausreißern und Fehldetektionen wird!

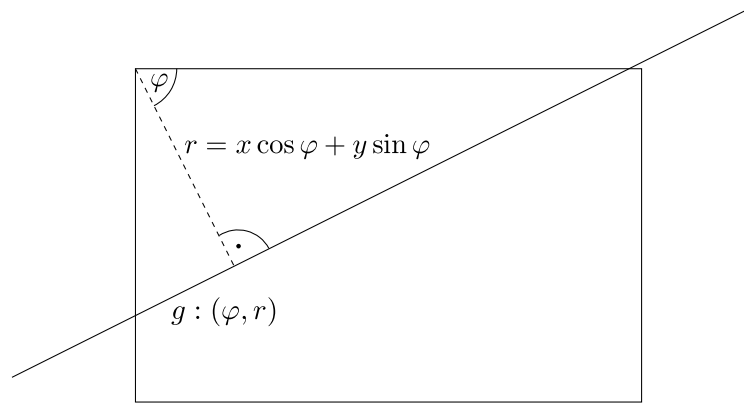


Abbildung 1: Parametrisierung einer Geraden über ihren Normalenwinkel φ und ihren Abstand r zum Ursprung.

Aufgabe 2 Region Growing

(4 Punkte)

Mit *Region Growing* wird eine Klasse von Verfahren bezeichnet, die ein Bild derart in zusammenhängende Regionen zerteilen, sodass jeder Bildpunkt zu genau einer Region gehört. Gestartet wird dabei mit sogenannten *Seed-Punkten*, von denen aus für alle benachbarten Punkte überprüft wird, ob sie zum Startpunkt passen. Dies wird iterativ so lange wiederholt, wie die Region noch wächst.

Als Homogenitätskriterium können komplexe Texturanalysen durchgeführt oder im einfachsten Fall Ähnlichkeiten von Grauwerten überprüft werden. Als formales Homogenitätskriterium h für Grauwerte v wäre beispielsweise

$$h(v) = \begin{cases} 1, & |v - v_{\text{seed}}| < \theta \\ 0, & \text{sonst} \end{cases} \quad (2)$$

denkbar. Dabei bezeichnet θ einen gegebenen Schwellwert und v_{seed} den Grauwert des Startpunktes. Um Startpunkte zu finden, kann man z. B. in der linken oberen Ecke des Bildes ein erstes Region Growing beginnen und dann sukzessive alle noch keiner Region zugewiesenen Punkte als neue Startpunkte verwenden. Prinzipiell sind aber natürlich auch ganz andere Initialisierungen möglich.

Implementieren Sie eine Python-Funktion, die mittels Region Growing ein gegebenes Grauwertbild in zusammenhängende Regionen teilt! Erzeugen Sie dabei ein Ausgabebild, auf dem die Regionen entweder farblich kodiert sind oder die Regionen jeweils als homogene Gebiete mit dem mittleren Grauwert der Region dargestellt werden!

Viel Spaß und Erfolg!