

Testaufgaben zur Vorlesung

Computergrafik 2

Teil 2/3



Bemerkungen:

- Bei jeder Aufgabe ist eine Kategorie angegeben, die Aufschluss darüber gibt welches Grundprinzip die Aufgabenstellung verfolgt:
 - **Reproduktion (RP):** Wissensfragen und Aufgaben die gelerntes Wissen abfragen. Die Lösung zur Aufgabe steht mehr oder weniger wörtlich in den Folien oder wurde im Screencast erläutert.
 - **Reorganisation (RO):** Aufgaben in denen Beispiele und Inhalte aus der Vorlesung aufgegriffen und leicht modifiziert werden.
 - Z.B. Beispielaufgabe rechnen oder Algorithmus auf Beispiel anwenden mit anderen Werten oder leicht veränderten Rahmenbedingungen
 - **Verständnis und Zusammenhang (VZ):** Verständnis überprüfen und Wissensbereiche miteinander verknüpfen. Inhaltliche Antworten nicht nur hinschreiben, sondern auch Begründungen geben warum dies korrekt ist / funktioniert.
 - **Transfer (TR):** Aufgaben die den Kontext so stark verändern, dass eine Lösungsstrategie erst aus den vorhandenen Wissen abgeleitet und konstruiert werden muss. Fragen deren Antworten nicht in den Unterlagen zu finden sind, sondern aus dem eigenen Verständnis heraus schlussgefolgert werden müssen. Schwierige Aufgaben, die selten vorkommen und dazu dienen festzustellen ob jemand eine 1 als Note verdient hat.
- Die Teilaufgaben sind potentielle Klausuraufgaben, bzw. waren das auch teilweise so oder so ähnlich schon in vergangenen Jahren.
- **Es gibt keine Bonuspunkte für die Bearbeitung der Testaufgaben!** Die Punkte bei den Aufgaben dienen nur zur Orientierung um einschätzen zu können, wie hoch die Gewichtung bezogen auf die Gesamtpunktzahl einer Klausur ist (Die Punkte sind angegeben in Bezug auf eine Klausur mit 60 Punkten gesamt)
- Es sind hier mehr Test-Aufgaben angegeben als in der Klausur zu einem Thema zu finden sein werden (siehe Punkte Gewichtung)
- **Abgabe dieser Testaufgaben:** Bis Montag 30.11. 23:59 Uhr (Moodle-Upload), falls Sie eine Korrektur wünschen (freiwillig).

Teilthema 1: Kantenerkennung:

- a) Geben Sie die strukturelle Funktionsweise des Canny-Algorithmus gemäß der Vorlesung als Pseudocode an. (RP, 5P)
- Sie können dazu die Routinen `getOrientationSector(...)`, `isLocalMax(...)`, `traceAndThreshold(..)` sowie die benötigten Filter für die Vorverarbeitung als bekannt voraussetzen.
 - Es ist also die Struktur sowie der korrekte Aufruf der Sub-Routinen gefragt: (Bedingungen wann aufgerufen wird, Parameterliste und die Bedeutung der Parameter)
- b) Erläutern Sie wie bei der Canny-Edge-Detection die Kanten-lokalisierung erfolgt (es ist konkret die Funktionsweise der Routine `isLocalMax(...)` gemäß der Vorlesung gefragt). Beachten Sie dabei folgende Teilaspekte (RP)
- Geben sie die Liste der Parameter beim Aufruf sowie den Rückgabewert an und erläutern Sie deren Bedeutung. (3P)
 - Geben Sie die Funktion der Routine als Pseudocode an und erläutern Sie die prinzipielle Funktionsweise. (3P)
Hinweis: Sie können die Routine `getOrientationSector(...)` bzw. deren Ergebnis als gegeben annehmen.
- c) Erläutern Sie die Routine `getOrientationSector(...)` bei der Canny-Edge Detection gemäß der Vorlesung. Beachten Sie dabei folgende Teilaspekte (RP)
- Geben sie die Liste der Parameter beim Aufruf sowie den Rückgabewert an und erläutern Sie deren Bedeutung. (2P)
 - Warum wird diese Methode benötigt, d.h. welchen Nutzen hat Sie? (1P)
 - Geben Sie die grundlegende Arbeitsweise als „Kochrezept“ oder Pseudocode an. (2P)
 - Erläutern Sie warum die in der Vorlesung gezeigte Umsetzung effizient ist. (1P)
- d) Erläutern Sie anhand einer Skizze, wie mit Hilfe einer geschätzten zweiten Ableitung eine Kantendetektion realisiert werden kann. In welchen Fällen ist diese Vorgehensweise besser geeignet als ein Sobel- oder Prewitt-Operator? (RP, 3P)

-
- e) Erläutern Sie kurz das Grundprinzip von Unsharp Masking (USM). (RP, 3P)
Welchen Nachteil hat das Verfahren?
- f) Erläutern Sie anhand einer ausführlichen Berechnung (also incl. Formeln) wieso die Laplace-Schärfung ein Spezialfall der USM Schärfung darstellt. (VZ, 4P)
- g) Erläutern Sie wie das Ergebnis der Canny-Edge-Detection auf einen 7 Pixel breiten diagonal verlaufenden schwarzen Balken auf grauem Hintergrund hat. (TR, 2P)

Teilthema 2: Hough Transformation:

a) Gegeben: Grundidee naiver brute-force Algorithmus Hough:

- Führe eine Vor-Klassifikation mittels Canny-Edge durch um n Kantenpunkte $\mathbf{p}_1, \dots, \mathbf{p}_n$ als „Geradenpunkte-Kandidaten“ zu finden.
- Lege eine Gerade durch je zwei Punkte $\mathbf{p}_i = (x_i, y_i)$ und $\mathbf{p}_j = (x_j, y_j)$
- aus den Bedingungen $y_i = k_{ij} \cdot x_i + d_{ij}$ und $y_j = k_{ij} \cdot x_j + d_{ij}$ lassen sich k_{ij}, d_{ij} berechnen.
- Prüfe für jedes k_{ij}, d_{ij} durch Einsetzen in die Gleichung, ob auch der Punkt $\mathbf{p}_l = (x_l, y_l)$ mit $1 \leq l \leq n$ und $l \neq i, l \neq j$ auf dieser Geraden liegt.
- Mit einem Schwellwert wird festgestellt ab welcher Anzahl eine Gerade detektiert wird.

angenommen bei einem HD-Bild sind 1/20 der Punkte (also ca. $n = 100000$) als Kantenkandidaten klassifiziert.

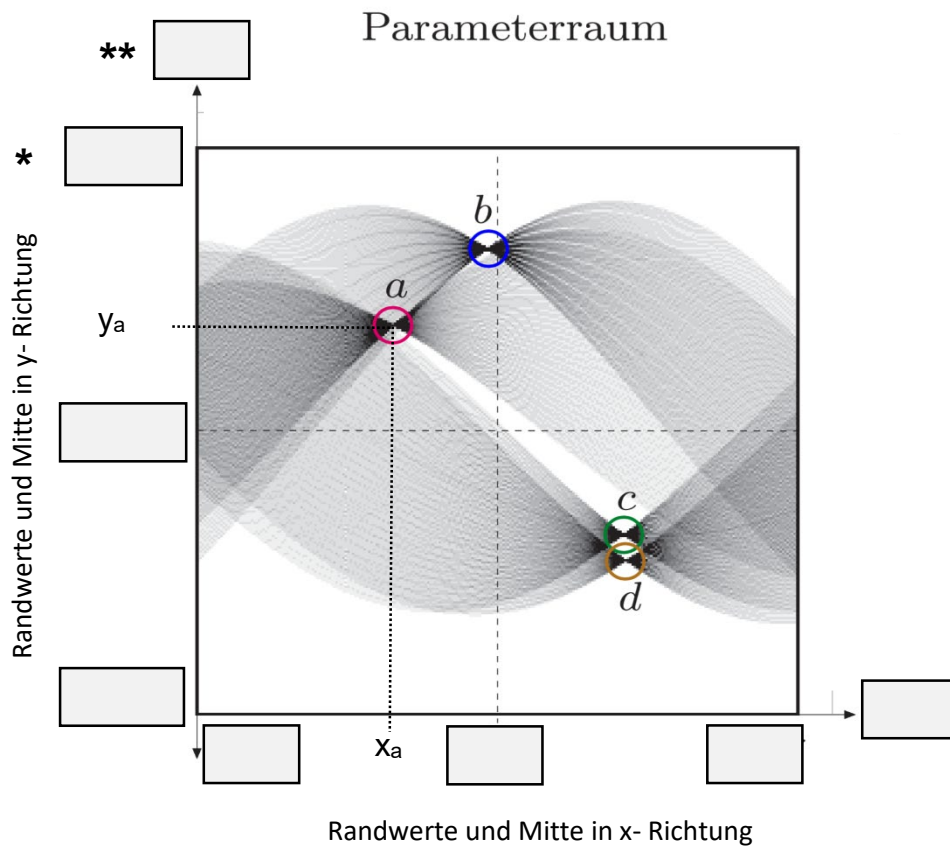
- Berechnen Sie die Laufzeit für den brute-force Algorithmus in Sekunden (3 Ghz Takt und 1 Takt je Operation) (RO, 3P)

Hinweis: Die Rechnung ist genau wie in der VL gezeigt, nur mit anderen Zahlen

b) Gegeben ist folgende Skizze zum Akkumulator-Array bei der Hough Transformation für Geraden in Hesse'scher Normalform (HNF).

- Geben Sie an welche Werte auf der x bzw. y-Achse abgetragen werden, (1P)
- Geben Sie an welche Größen die jeweiligen Maximal-, Minimal- bzw. Mittel-Werte auf der x bzw. y Achse haben. (2P)

Vervollständigen Sie dazu die Beschriftung Skizze an den grauen Rechtecken



- Welche Bedeutung hat die auf der y-Achse abgetragene Größe (**) bei der Hough Transformation? (1P)
- Geben Sie an wie die Größe der Wertes an Position (*) berechnet wird. (1P)

c) Im Bildraum sei die Mitte des Bildes mit x_r bezeichnet.

- Skizzieren sie den ungefähren Verlauf der im Parameterraum an den Punkten a und c detektierten Geraden (vgl. Bild zu Aufgabenteil b), Zuordnung beschriften!). (1P)
- Geben Sie an, wo die Werte x_a und y_a , in der Skizze unten abgetragen werden (1P)

