

Bildanalyse und Bildverstehen, SoSe 2002 Übungsblatt 4

Bearbeitung durch je 2 Personen gemeinsam erlaubt + erwünscht (bitte nur 1 mal pro Gruppe abgeben).

Abgabe der Lösungen am 26. 06. 2002, bis 15:30 Uhr in der Vorlesung (schriftlich) oder bis zum selben Termin per e-mail an Herrn Zhao, dzhao@informatik.tu-cottbus.de. Lösungen der Programmieraufgabe 4 (Quellcode) und der Aufgabe 5 (Bilddateien) bitte nach Möglichkeit per e-mail an Herrn Zhao.

Verbindliches zu den e-mails: Nur je eine e-mail pro Gruppe (spätere Korrektur-e-mails werden nicht mehr akzeptiert). Subject: **BB-Ubungsblatt 4**. Erste Zeile der e-mail: Namen der beiden AutorInnen und Matrikelnummern. Zweite Zeile: Angabe, ob zusätzlich ein schriftliches Lösungsblatt (mit Lösungen von Theorie-Aufgaben) abgegeben wurde / wird. Quellcode-Dateien und Bilddateien bitte als Attachments anfügen, ggf. archiviert.

Aufgabe 1

Bestimmen Sie zu den folgenden beiden Binärbild-Objekten die folgenden Merkmale: Fläche (in Pixeln), Umfang (exakte Länge, Pixel-Seitenlänge = 1), Schwerpunkt, Formfaktor, Exzentrizität, *aspect ratio* der *Feret box*, Füllungsgrad der *Feret box*, Signatur (Abstand zum gegenüberliegenden Randpunkt für jeden Randpunkt, als Diagramm). (12 P.)



Aufgabe 2

In beliebiger Orientierung liegende und einander nicht überlappende gleichseitige Dreiecke und Quadrate mit gleicher Fläche sollen anhand ihres Formfaktors unterschieden werden. Welche Trennschwelle sollte für die Klassifikation gewählt werden? (3 P.)

Aufgabe 3

Für die folgenden 1-dimensionalen "Texturen" (Grauwertmuster) mit Grauwerten aus {0; 1; 2; 3} sollen die folgenden Merkmale bestimmt werden: Mittelwert, Standardabweichung, Schiefe (der Grauwertverteilung; vgl. Übung 1), Cooccurrence-Matrix (bzgl. direkter Nachbarschaft), Lauflängenmatrix, *short run emphasis*, *long run emphasis*. (9 P.)

(a)

0	0	1	1	2	2	3	3
---	---	---	---	---	---	---	---

(b)

0	2	1	3	0	2	1	3
---	---	---	---	---	---	---	---

(c)

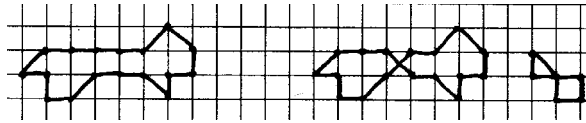
0	0	0	3	0	0	0	3
---	---	---	---	---	---	---	---

Aufgabe 4

(a) Der Kettencode einer Kontur kann als Oktalzahl interpretiert werden, der ein bestimmter Dezimalwert entspricht. Implementieren Sie ein Programm, das für eine einzulesende nicht-negative Dezimalzahl (aus dem **long int**-Bereich, d.h. 64 bit-Zahlen) entscheidet, ob eine geschlossene Kontur vorliegt oder nicht. (6 P.)

(weiter auf der nächsten Seite)

- (b) Das Programm aus (a) soll dahingehend erweitert werden, dass auch auf Überkreuzungsfreiheit geprüft wird (das folgende linke Beispiel ist überkreuzungsfrei, die beiden anderen nicht). (3 P.)



Aufgabe 5

- (a) Laden Sie mit AdOculus das Bild **Pyramide.iv**. Wenden Sie sukzessive die Bildoperationen "Gradient 5*5->x/y" (2 Ausgabebilder), "Kartesisch/Polar Int->Byte" (2 Ein- und 2 Ausgabebilder) und "Verdünnung" (2 Ein-, 2 Ausgabebilder) an. (Parameter für Kart./Polar: Schwellenwert 10, Verdünnung: 30°). Auf das erste Ergebnisbild der Konturverdünnung wenden Sie die Operation "Konturpunkt-Verkettung" an, auf deren Ergebnis die Operation "Approximation" (durch Geradenstücke; Parameter: max. Fehler = 3). Das Ergebnis soll als **tif**-Datei mit IrfanView abgespeichert werden. (2 P.)
- (b) Wenden Sie dieselbe Funktionskette auf **Mzhsrsrc.iv** an und speichern Sie ebenfalls das Endergebnis. (1 P.)
- (c) Von den Bildern **Mzhsrsrc.iv** und **Seidesrsrc.iv** sollen von AdOculus die Co-occurrence-Matrizen für die unmittelbare Nachbarschaft in x -Richtung ermittelt werden (Größe: 128). Zur besseren Sichtbarkeit der Ergebnisse invertieren Sie bitte beide Ergebnisse mit IrfanView ("Negative") vor dem Abspeichern als **tif**-Dateien. (2 P.)