

Bildanalyse und Bildverstehen, SoSe 2006 Übungsblatt 4

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

Abgabe der Lösungen am 29. 06. 2006, bis 7:30 Uhr in der Übung (schriftlich) oder bis zum selben Termin per e-mail an Herrn Mangoldt, tm@informatik.tu-cottbus.de. Lösungen der Programmieraufgabe 4 (Quellcode) und der Aufgabe 5 (Bilddateien) bitte nach Möglichkeit per e-mail an mich.

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 *Ferret box*, Füllungsgrad der *Ferret 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)

