



WERKZEUGE MUSTERERKENNUNG & MASCHINELLES LERNEN

Aufgabenblatt 4

(Ausgabe am Fr 10.5.2019 — Abgabe bis So 19.5.2019)

Aufgabe 1

10 P

Es geht darum, mit der Momentenmethode den Schwerpunkt und den Neigungswinkel von acht abgebildeten Raumfahrzeugen zu bestimmen. Die benötigten Grauwertbilder finden Sie in der Datei `moment.rda`.

- (a) Schreiben Sie eine Funktion `moment(x, plot=TRUE)`, die zum Grauwertbild `x` mit Hilfe der zentralen Momente (ME-Skript IV.4, Blatt 6) die Koordinaten des Objektschwerpunkts und den Neigungswinkel des Objekts (in Grad) berechnet; Rückgabe als benannter Vektor.
TIPP: Vektorisieren Sie beispielsweise mit Hilfe von `col()` und `row()`!
- (b) Bei Aufruf mit `plot=TRUE` soll das Bild `x` gezeichnet werden und der berechnete Schwerpunkt mit einem farbigen Fadenkreuz (z.B. `?abline`) markiert werden; beachten Sie den Unterschied zwischen Bildmatrix- und Grafikkoordinaten.
- (c) Schreiben Sie den berechneten Neigungswinkel (in Grad; Nord/West/Ost entspricht $0^\circ/90^\circ/-90^\circ$) rechts neben das Bild (Funktion `mtext()`). Denken Sie unbedingt wieder an `atan2()`! Fügen Sie einen weiteren `abline()`-Aufruf hinzu, der eine Gerade durch den Schwerpunkt mit dem berechneten Neigungswinkel einzeichnet.
- (d) Erstellen Sie nun je Beispielsbild eine (2×2) -Leinwand mit je einem `moment()`-Aufruf für das Original `x`, für sein Negativ `1-x` sowie für die Binärversionen (Funktion `binarise()` zur Binarisierung mit 2-means-Schwelle finden Sie in `moment.rda`) von Original und Negativ.
- (e) Für welche der vier Verarbeitungsvarianten in (d) werden Schwerpunkt und Neigungswinkel der Objekte verlässlich berechnet? Welche Bildeigenschaft führt zum Versagen des Verfahrens bei den Raketen `modes` und `trend`? Welche Vorfilterung wäre geeignet, damit `moment()` auch hier den korrekten Neigungswinkel entdeckt?

Abzugeben sind die Programmdatei `moment.R` sowie Ihre drei schriftlichen Antworten zu (e).

Aufgabe 2

10 P

Das Grauwerthistogramm eines Rasterbildes soll durch das Verfahren der kanonischen Gleichverteilung (ME-Skript IV.3) egalisiert werden.

- (a) Laden Sie die 'R'-Objekte-Datei `equalize.rda` (Webseite zur Übung), welche die folgenden Grauwertbildobjekte enthält:

```
algae couple JKKennedy mri1 soil turbinate GUESS
```

- (b) Schreiben Sie eine 'R'-Funktion `equalize(x)`, die eine Bildmatrix `x` auf ihr Grauwerthistogramm hin analysiert und das egalisierte Bild (Grauwertbereich $[0, 1]$) als Ergebnis abliefert.

TIPP: Verwenden Sie eine der Rangordnungsfunktionen `sort()`, `rank()` oder `order()` für diesen Zweck.

- (c) Schreiben Sie nun eine 'R'-Funktion `plot.equalize(x, main="", K=64)`, die eine Bildmatrix `x` und ihre egalisierte Variante zusammen mit den beiden absoluten Grauwerthistogrammen und den beiden kumulativen Grauwertverteilungen in der Auflösung `K` (Anzahl Grauwertzellen) auf einer (3×2) -Leinwand visualisiert. Verwenden Sie zur Berechnung der Grauwertstatistiken die 'R'-Funktion `hist()`.

TIPP: Vermeiden Sie, doppelten Code für `x` und `equalize(x)` zu schreiben!

- (d) Testen Sie Ihre Implementierung mit den obigen Bildern und geben Sie für jedes Bild einen Kurzkomentar (nur eine Zeile) zum Erfolg oder zum Misserfolg der Grauwertegalisierung.

- (e) Wer singt was in welcher Stadt auf dem Rätselbild `GUESS`?

- (f) Fügen Sie jetzt Ihrer Funktion `equalize` einen zweiten Aufrufparameter `RGB` vom Typ `logical` hinzu sowie eine Verzweigung, die automatisch Farbbilder erkennt. Für `RGB=TRUE` werde das Farbbild `x` separat je RGB-Kanal egalisiert.

TIPP: Lassen Sie `equalize` sich selbst (für die drei Farbkanäle) aufrufen!

- (g) Für undefiniertes `RGB` (siehe `?missing`) werde die egalisierte Grauwertbildversion abgeliefert. Für `RGB=FALSE` werden die Pixel `isoton` transformiert (R/G/B-Proportionen unverändert), so dass die Gesamthelligkeit zur Gleichverteilung wird.

TIPP: Zum Einstieg Graubildversion `g` von `x` und `equalize(g)` erzeugen.

- (h) Zeichnen Sie nun das Farbbild `potus` erst im Original und dann mit den drei implementierten Egalisierungsvarianten.

Abzuliefern ist der 'R'-Programmcode `equalize.R` zur sukzessiven Erzeugung der (3×2) -Grafiken und der (2×2) -Farbseite sowie die Textantwort zu den Aufgabenteilen (d,e).

Hinweise zum Übungsablauf

- ➡ Die wöchentliche WMM-Vorlesung findet am Mittwoch um 12:15 Uhr statt.
Das Aufgabenblatt gibt es immer am Freitag (PDF im Netz).
Der späteste Abgabetermin ist Sonntag 23:59 Uhr.
- ➡ Die Übungsaufgaben dürfen natürlich (und sollten sogar) in Gruppenarbeit (2 Mitglieder) gelöst werden.
- ➡ Schriftliche Lösungen („Textantworten“) sind als PDF beizufügen oder direkt im e-Mail-Textkörper unterzubringen.
- ➡ Alle anderen Lösungen (Programmieraufgaben, Daten und Grafiken) sind als elektronischer Anhang der Lösungs-e-Mail abzuliefern.
- ➡ Programmcode (Dateien *.R) muss auch wirklich in 'R' ausführbar sein.
(Kommando `Rscript <name>.R` auf einem der Rechner des FRZ-Pools)
- ➡ Ganz wichtig:
Schriftliche Antworten werden von mir gedruckt, gelesen, kommentiert und korrigiert.
Deshalb diese Textteile bitte **niemals** im abgegebenen Programmcode verstecken!
- ➡ Je Gruppe und je Aufgabenblatt ist **genau eine** e-Mail zu senden:
 - Vermerk »WMM/*n*« und Gruppenname im **subject**-Feld
(*n* ∈ ℕ ist die laufende Nummer des Übungsblattes)
 - die Namen der beteiligten Gruppenmitglieder im Textrumpf
 - Tabellen, Bilder, Programmcode, Sensordaten als Attachments
(elektronische Anlagen)
 - etwaige schriftliche Antworten im Textrumpf der Post oder als Attachment
(Text/PDF)
- ➡ Einige Aufgabentexte verweisen Sie zum Nachschlagen von Details auf das Folien-skript zur Vorlesung Mustererkennung; Sie finden es unter der URL
<http://www.minet.uni-jena.de/fakultaet/schukat/ME/Scriptum/>.
Die Angabe *ME-Skript II.6* bedeutet: Kapitel II, Abschnitt 6