

Übungen zur Vorlesung
Grundlagen der Bilderzeugung und Bildanalyse (Mustererkennung)
WS 05/06

Aufgabenblatt 1 (12 Punkte)

Vorlesungsstoff: bis ME-I, Kap. 2a

Abgabe am Mittwoch 9.11.05 vor der Vorlesung

Bitte Name und Matrikelnummer auf den Lösungen angeben.

Aufgabe 1.1: Äquivalenzrelation, Invarianz (4 Punkte)

Wir betrachten 2 Relationen (mit Relationssymbol \sim) auf der Menge der Punkte in der komplexen Ebene $\mathbf{z} = x + \mathbf{j}y \in \mathbb{C}$.

$$(1) \quad \mathbf{z} \sim \mathbf{w} \iff \exists a \in \mathbb{R} : \mathbf{w} = \mathbf{z} + a \cdot (1 + \mathbf{j})$$
$$(2) \quad \mathbf{z} \sim \mathbf{w} \iff |\mathbf{z}| = |\mathbf{w}|$$

1. Zeigen Sie, dass es sich um Äquivalenzrelationen handelt.
2. Finden Sie in beiden Fällen eine Funktion $I(\mathbf{z}) : \mathbb{C} \mapsto \mathbb{R}$, die invariant auf den Äquivalenzklassen ist, d.h. es soll gelten $\mathbf{z} \sim \mathbf{w} \Rightarrow I(\mathbf{z}) = I(\mathbf{w})$. Weiter sollen die Merkmale $I(\mathbf{z})$ die Klassen separieren, d.h. sie sollen pro Klasse unterschiedliche Werte annehmen, formal $I(\mathbf{z}) \neq I(\mathbf{w}) \Rightarrow \neg(\mathbf{z} \sim \mathbf{w})$.
3. Beschreiben Sie für alle Äquivalenzrelationen die geometrische Gestalt der Äquivalenzklassen anschaulich.

Aufgabe 1.2: Spiegelungsinvarianten (4 Punkte)

Wir betrachten folgende Äquivalenzrelation auf Punkten in der 2D-Ebene

$$\mathbf{x} \sim \mathbf{y} \iff (\mathbf{x} = \mathbf{y}) \vee (\mathbf{x} = \mathbf{I}\mathbf{y})$$

wobei die Spiegelungsmatrix \mathbf{I} gegeben ist durch $\mathbf{I} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. Geben Sie ein invariantes Merkmal $I(\mathbf{x}) : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ an, welches die resultierenden Äquivalenzklassen separiert und erläutern Sie es geometrisch!

Aufgabe 1.3: Programmieraufgabe: Transformation von Vektoren und Matrizen (4 Punkte)

In dieser Aufgabe sollen 2 Funktionen in Scilab implementiert werden.

1. Permutiert man den Vektor

$$x = [1 \ 2 \ 3 \ 4 \ 5]$$

zyklisch um zwei Stellen nach rechts, so erhält man den Vektor y :

$$y = [3 \ 4 \ 5 \ 1 \ 2]$$

Es bezeichne P_n die Matrix, die durch Multiplikation mit einem Vektor im \mathbb{R}^n eine zyklische Translation dieses Vektors nach links durchführt, d.h. $\mathbf{P}_n^T \cdot (x_1, x_2, \dots, x_n)^T = (x_2, \dots, x_n, x_1)^T$.

Schreiben Sie eine Funktion `cyclmat`, die für beliebige ganzzahlige Dimension $n \geq 2$ eine entsprechende Matrix \mathbf{P}_n erzeugt. Einziges Eingabeargument ist die Dimension n , das einzige Ausgabeargument soll die Matrix \mathbf{P}_n sein. Multiplizieren Sie \mathbf{P}_n zwei Mal mit x und Sie sollten y erhalten.

2. Mit Hilfe einer Matrix können Punkte im zwei dimensionalem Raum dargestellt werden. Die Elemente der ersten Zeile sind die x -Koordinaten, die der zweiten Zeile die y -Koordinaten der Punkte. So beschreibt die folgende Matrix den Buchstaben M :

$$M = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 2 & 3 & 4 & 5 & 5 & 5 & 5 & 5 \\ 1 & 2 & 3 & 4 & 5 & 4 & 3 & 4 & 5 & 4 & 3 & 2 & 1 \end{bmatrix}$$

Man kann sich die Punkte mit Hilfe der Scilab Funktionen `plot2D` und `xpoly` anzeigen lassen:

```
plot2d(0,0,axesflag=5,rect=[-10,-10,10,10]);  
xpoly(M(1,:),M(2:2,:),"lines",0);
```

Verwenden Sie den Buchstaben M oder erzeugen Sie sich einen eigenen Buchstaben. Dieser Buchstabe soll nun mit Hilfe einer Scilab Funktion `translate`, die die Translation aller in der Matrix übergebenen Punkte um x in x -Richtung und y in y -Richtung vornimmt, verschoben werden. Zeichnen Sie anschließend den um $x = 2$ und $y = 3$ translierten Buchstaben.

Hinweise:

- Einige nützliche Funktionen: `eye` zur Erzeugung einer Einheitsmatrix, Transponieren geschieht durch Anhängen eines Hochkomma an die entsprechende Matrix, die s -te Potenz einer Matrix P ist in Scilab durch P^s darstellbar.
- Um eine Erklärung der verwendeten Funktionen zu bekommen, schaut man am besten in der Hilfe von Scilab.

Bitte Autor in den Files angeben (Kommentarzeilen werden durch `//` eingeleitet) und die Implementationen in ausgedruckter Form abgeben.

Anhang: Erste Schritte mit Scilab

Scilab ist ein freies Software Paket mit dem sehr schnell Matrixoperationen durchgeführt werden können. (Die kommerzielle Version heißt Matlab.) Scilab ist für alle herkömmlichen Betriebssysteme unter <http://www.scilab.org> verfügbar. Unter dem Stichpunkt Dokumentation findet sich eine *Introduction to Scilab*, die einen guten Einstieg bietet.

Scilab besitzt eine Kommandozeile in der Befehle direkt eingegeben werden können. Man kann schon vordefinierte Funktionen benutzen oder selber Funktionen definieren. Um eine Scilab Funktion zu definieren wird eine Textdatei mit der Endung `.sci` erzeugt. So könnte eine Funktion aussehen, die zwei Zahlen als Eingabe nimmt und die Summe dieser beiden Zahlen als Ausgabe erzeugt:

```
function res = add(a,b)
// function res = add(a,b)
// implementation of the addition of two numbers.
// input:      a           first number
//            b           second number
// output:    res          sum of two numbers
// Max Mustermann 2.11.2005

res = a + b;

endfunction;
```

Bevor die Funktion in der Kommandozeile benutzt werden kann, muss sie ausgeführt werden (File → Exec).

Ein Vektor wird durch folgenden Eintrag in der Kommandozeile erzeugt:

```
x = [ 1 2 3 4 5 ]
```

Verwendet man kein Semikolon nach der eckigen Klammer, so wird der Vektor noch Mal ausgegeben.

Mit einer runden Klammer kann auf die Elemente von x zugegriffen werden. Mit einem Doppelpunkt können mehrere Elemente ausgegeben werden.

```
x(4) =
      ! 4. !

x(1:3) =
      ! 1. 2. 3. !
```

Eine Matrix wird mit folgendem Scilab Code erzeugt:

```
A = [ 1 2 3 4 ; 5 6 7 8 ]
```

Das dritte Element in der zweiten Zeile bekommt man durch:

```
A(2,3)
```

Die gesamte zweite Zeile wird mit einem Doppelpunkt ausgewählt:

```
A(2,:) 
```