ALBERT-LUDWIGS-UNIVERSITÄT FREIBURG
INSTITUT FÜR INFORMATIK
Lehrstuhl für Mustererkennung und Bildverarbeitung
Prof. Dr.-Ing. Hans Burkhardt

Georges-Köhler-Allee
Geb. 052, Zi 01-029
D-79110 Freiburg
Tel. 0761 - 203 - 8260

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

Musterlösung 1

Aufgabe 1.1: Äquivalenzrelation, Invarianz

1. Zeigen Sie

$$\begin{array}{lccl} (1) & \mathbf{z} \sim \mathbf{w} & \Longleftrightarrow & \exists a \in \mathbb{R} : \mathbf{w} = \mathbf{z} + a \cdot (1 + \mathbf{j}) \\ (2) & \mathbf{z} \sim \mathbf{w} & \Longleftrightarrow & |\mathbf{z}| = |\mathbf{w}| \end{array}$$

ist eine Äqivalenzrelation.

Eine Relation $\sim$ ist Äquivalenzrelation, gdw. gilt:

$$\begin{array}{rcll} & \mathbf{z} \sim \mathbf{z} & & \text{reflexiv} \\ \mathbf{z} \sim \mathbf{w} & \Longleftrightarrow & \mathbf{w} \sim \mathbf{z} & \text{symmetrisch} \\ \mathbf{z} \sim \mathbf{w} \wedge \mathbf{w} \sim \mathbf{v} & \Rightarrow & \mathbf{z} \sim \mathbf{v} & \text{transitiv} \end{array}$$

(1)

reflexiv z.z. $\exists a \in \mathbb{R} : \mathbf{z} = \mathbf{z} + a \cdot (1 + \mathbf{j})$
Wähle $a = 0$

$$\mathbf{z} = \mathbf{z} + 0 \cdot (1 + \mathbf{j})$$

symmetrisch z.z. $\exists a' \in \mathbb{R} : \mathbf{z} = \mathbf{w} + a' \cdot (1 + \mathbf{j})$
Wähle $a' = -a$

$$\begin{array}{rcl} \mathbf{z} \sim \mathbf{w} & \overset{def}{\Longleftrightarrow} & \mathbf{w} = \mathbf{z} + a \cdot (1 + \mathbf{j}) \\ & \Longleftrightarrow & \mathbf{z} = \mathbf{w} + (-a) \cdot (1 + \mathbf{j}) \\ & \Longleftrightarrow & \mathbf{z} = \mathbf{w} + a' \cdot (1 + \mathbf{j}) \Longleftrightarrow \mathbf{w} \sim \mathbf{z} \end{array}$$

transitiv z.z. $\exists a' \in \mathbb{R} : \mathbf{v} = \mathbf{z} + a' \cdot (1 + \mathbf{j})$
Wähle $a' = a_1 + a_2$

$$\begin{array}{rcl} \mathbf{z} \sim \mathbf{w} \wedge \mathbf{w} \sim \mathbf{v} & \Longleftrightarrow & \mathbf{w} = \mathbf{z} + a_1 \cdot (1 + \mathbf{j}) \wedge \mathbf{v} = \mathbf{w} + a_2 \cdot (1 + \mathbf{j}) \\ & \Rightarrow & \mathbf{v} = \mathbf{z} + (a_1 + a_2) \cdot (1 + \mathbf{j}) \\ & \Longleftrightarrow & \mathbf{v} = \mathbf{z} + a' \cdot (1 + \mathbf{j}) \Longleftrightarrow \mathbf{z} \sim \mathbf{v} \end{array}$$

q.e.d.

(2)

reflexiv z.z. $|\mathbf{z}| = |\mathbf{z}|$
trivial

symmetrisch z.z. $|\mathbf{z}| = |\mathbf{w}| \Longleftrightarrow |\mathbf{w}| = |\mathbf{z}|$
trivial

transitiv z.z. $|\mathbf{z}| = |\mathbf{w}| \wedge |\mathbf{w}| = |\mathbf{v}| \Longrightarrow |\mathbf{z}| = |\mathbf{v}|$
trivial

q.e.d.

2. Finden sie zu Äquivalenzrelationen (1) und (2) jeweils eine Invariante

(1) Eine Projektion der komplexen Zahl auf die Richtung $(1 - \mathbf{i})$ erfüllt die Invariantenbedingung.

$$\begin{aligned} I_1(\mathbf{z}) &= \mathbf{z}(1+\mathbf{i}) + \bar{\mathbf{z}}(1-\mathbf{i}) \\ &= \mathbf{z} + \bar{\mathbf{z}} + \mathbf{i}(\mathbf{z} - \bar{\mathbf{z}}) \\ &= 2 \cdot (Re(\mathbf{z}) - Im(\mathbf{z})) \end{aligned}$$

(2) $I_2(\mathbf{z}) = |\mathbf{z}|$

Bei (2) kann die Invariante direkt aus der Definition der Äquivalenzrelation abgelesen werden, da die Relationen in diesem Fall nur auf Gleichheit basieren.

3. Beschreiben Sie die geometrische Gestalt der Äquivalenzklassen

(1) Jede Äquivalenzklasse bildet in der komplexen Ebene eine Gerade mit Steigung 1. (Abb. 1)

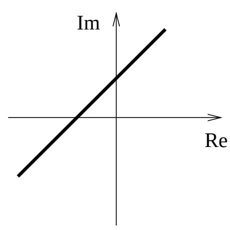

Abbildung 1: Beispiel einer Äquivalenzklasse der Relation (1)

Eine Abbildung die diese Gestalt erzeugt ist:

$$\mathbf{z}_a(t) = (1+\mathbf{i}) \cdot t + (1-\mathbf{i}) \cdot a$$

(2) Jede Äquivalenzklasse bildet einen Kreis um den Ursprung der komplexen Ebene (Abb. 2)
Eine Abbildung, die diese Gestalt erzeugt ist:

$$\mathbf{z}_a(t) = a \cdot e^{\mathbf{j}t}$$

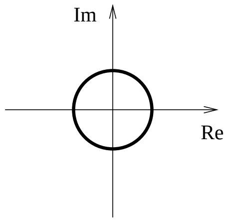

Abbildung 2: Beispiel einer Äquivalenzklasse der Relation (2)

Aufgabe 1.2: Spiegelungsinvarianten

Gesucht ist eine invariantes Merkmal $I(\mathbf{x})$, wobei $\mathbf{x} = (x, y), x, y \in \mathbb{R}$, das die Äquivalenzklassen der gegebenen Spiegelungsrelation separiert. Eine Äquivalenzklasse wird durch den Vektor $\mathbf{x} = (x, y)$ und den gespiegelten Vektor $\mathbf{xs} = (y, x)$ gebildet.

Für die Inavrianten gibt es mindestens zwei denkbare Lösungen:

(1) $I(x, y) = (x + y, |x - y|)$
Dabei lässt sich $x + y$ geometrisch als Projektion auf die Winkelhalbierende (WH) interpretiern; $|x - y|$ ist proportional zum Abstand zur WH.

(2) $I(x, y) = (\max(x, y), \min(x, y))$
Der Punkt wird durch die max und min Operation normalisiert: Falls ein Punkt oberhalb der WH liegt wird er gespiegelt und liegt dann unterhalb der WH.

Aufgabe 1.3: Programmieraufgabe (Transformationen von Vektoren und Matrizen) (2)

Die in der Aufgabenstellung angegebene Wirkung von $\mathbf{P}_n^T \cdot \mathbf{x}$ für $\mathbf{x} \in \mathbb{R}^n$ definiert die Matrix $\mathbf{P}_n$ eindeutig. z.B. für $n = 4$

$$\mathbf{P}_4 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$\mathbf{P}_n$ wird also einfach durch Verschieben der Einheitsmatrix $\mathbf{I}_n$ um einen Eintrag nach unten erzeugt.

Eine mögliche Implementierung ist daher:

```
function p = cyclmat(n)
// function p = cyclmat(n)
//
// implementation of cyclic translation matrix of dimension n.
// Any vector can be translated cyclically by simple
// multiplication with the transpose of this this matrix.
//
// input:    n            dimension of matrix, integer >=1
//
// output:   p            cyclic translation matrix, e.g. n=4
//                        implies p^t * (1 2 3 4)^t = (2 3 4 1)^t

// Bernard Haasdonk 24.10.2002
```

```
[out,in] = argn(0);

if in~=1,
   error('wrong number of input arguments.');
end;

p = eye(n,n);
p = [p($,:) ; p(1:$-1,:)];
endfunction;
```

Hierbei bezeichnet `$` den höchstmöglichen Index der entsprechenden Matrixdimension, : indiziert den gesamten möglichen Indexbereich. Teile der Indexbereiche kann man auch konkret durch `n:m` angeben. `p($,:)` liefert also den Teilbereich der Matrix `p`, der deren unterster Zeile entspricht, `p(1:$-1,:)` bezeichnet den Bereich der ersten bis vorletzten Zeile von `p`. Die Routine erzeugt also zunächst die Einheitsmatrix, und setzt deren letzte Zeile über die restlichen Zeilen.

2. Nun sollte die Transformation der Matrix berechnet werden. Hierzu benötigen wir eine Funktion `translate`:

```
function res = translate (pts,x,y)

// function for moving points
// pts - the points ( given in a matrix:
//                    first row: x-coordinates
//                    second row: y-coordinates
// x   - shift along the x-axis
// y   - shift along the y-axis
// res - the translated points

res = pts;
res(1,:) = pts(1,:) + x;
res(2,:) = pts(2,:) + y;

endfunction;
```

Das verschieben des Buchstabens M und die grafische Darstellung der verschiebung erreicht man durch das Ausführen folgender Befehle:

```
M = [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]

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

//transformation
m=translate(M,2,3);

xpoly(m(1,:),m(2,:),"lines",0);
```

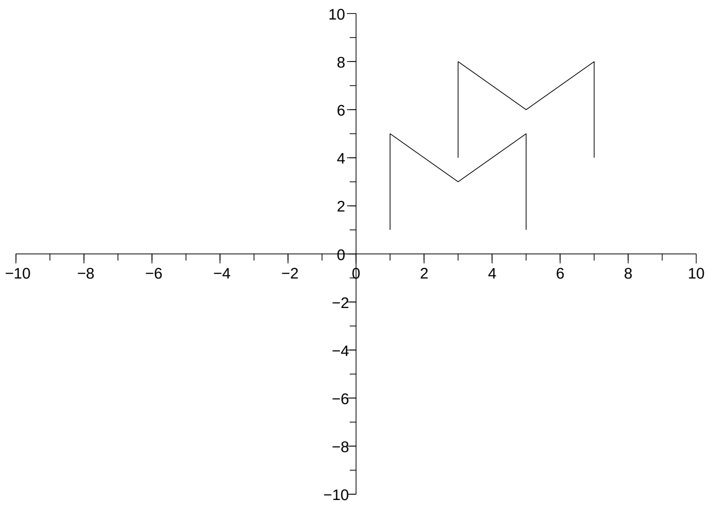

Abbildung 3: Buchstabe M vor und nach der Translation um $x = 2$ und $y = 3$