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

Aufgabenblatt 2 (12 Punkte)

Vorlesungsstoff: bis ME-I, Kap. 2c

Abgabe am 16.11.05 vor der Übung

Bitte Name und Matrikelnummer auf den Lösungen angeben.

Aufgabe 2.1: Affine Transformation, Rotationsmatrizen (4 Punkte)

Die Punkte $\mathbf{x} = (1,0)^T$, $\mathbf{y} = (0,1)^T$ und $\mathbf{z} = (1,-1)^T$ werden durch eine affine Transformation auf die Punkte $\mathbf{x}' = (11,8)^T$, $\mathbf{y}' = (4,12)^T$ und $\mathbf{z}' = (9,2)^T$ abgebildet und legen diese eindeutig fest.

a) Man berechne die zugehörige affine Transformation $\mathbf{x} \mapsto \mathbf{Ax} + \mathbf{t}$.

b) Stelle die Matrix $\mathbf{A}$ in der Form $\mathbf{A} = \mathbf{R}^T\mathbf{DR}$ dar, wobei $\mathbf{R}^T\mathbf{R} = \mathbf{I}$ und $\mathbf{D}$ eine reelle Diagonalmatrix ist. Was muss für $\mathbf{A}$ gelten, damit dies möglich ist?

Aufgabe 2.2: Rotation, Kongruenzen, Invarianten (4 Punkte)

Eine Polygon $\mathbf{x}$ in der euklidischen Ebene ist durch eine geordnete Punktmenge $\mathbf{x} = (\mathbf{p}_1, ..., \mathbf{p}_n)$ mit $\mathbf{p}_i \in I\!R^2$ repräsentiert. Im folgenden soll das Polygon invariant gegenüber bestimmten geometrischen Transformation repräsentiert werden, dabei soll das ursprüngliche Polygon bis auf die verlorenen Transformationsparameter aus den invarianten Merkmalen $T(\mathbf{x})$ zurückgewonnen werden können, d.h. die Merkmale sollen vollständig sein. Man gebe vollständige, invariante Merkmale bezüglich

a) Translation: $\mathbf{p}_i \mapsto \mathbf{p}_i + \mathbf{t}$,

b) Kongruenzen: $\mathbf{p}_i \mapsto \mathbf{Rp}_i + \mathbf{t}$ mit $\mathbf{R}^T\mathbf{R} = \mathbf{I}$ und $\det(\mathbf{R}) = 1$,

und begründe kurz, warum sie vollständig sind.

Die Merkmale müssen nicht invariant gegenüber der parametrischen Beschreibung sein, d.h. es muss nicht $T(\mathbf{p}_1, ..., \mathbf{p}_n) = T(\mathbf{p}_2, ..., \mathbf{p}_n, \mathbf{p}_1)$ gelten.

Aufgabe 2.3: Programmieraufgabe: Rotation (4 Punkte)

Schreiben Sie eine Funktion, die ein gegebenes Bild $\mathbf{A}$ um einen Winkel ϕ rotiert.

1. Auf der Scilab-Homepage `www.scilab.org` unter Contributions ist ein Scilab-Paket `IMAGE` zu finden. Instruktionen zur Verwendung des Pakets sind hierin enthalten. Auf den Webseiten zur Vorlesung finden Sie ein Demo-Bild `M.pgm` der Größe 30×30. Laden Sie dieses in eine Scilab-Matrix $\mathbf{A}$ mit dem Befehl `[A,h]=readppm("pathToImage/M.pgm")`.
2. Das rotierte Bild $\mathbf{A}'$ ist gegeben durch $\mathbf{A}'(x) := \mathbf{A}(R^T x)$, wobei R eine zweidimensionale Rotationsmatrix ist, welche den Vektor x um den Winkel ϕ dreht. Natürlich fällt der Punkt $R^T x$ in der Regel nicht aus eine ganzzahlige Pixelkoordinate, benutzen sie `round` um sich für den nächstgelegenen Pixel zu entschieden. Der Mittelpunkt des Bildes soll das Rotationszentrum sein. Wählen sie die Größe des Ergebnisbildes $\mathbf{A}'$ so, daß keine Pixel des Ursprungsbildes über den Rand hinaus rotiert werden.
3. Stellen sie beispielhaft Ergebnisse für $\phi = \pi/4, \pi/3, \pi/2$ dar und drucken Sie diese aus.

Hinweise:

- Nachdem $\mathbf{A}$ geladen wurde, kann diese in eine Fließkomma-Matrix verwandelt werden, z.B. via `D = double(A)`, damit normale Rechenoperationen möglich sind.
- Darstellen eines Graubildes geschieht z.B. via `imageview(graycolormap(256), D)`. Eventuell kann es notwendig sein, im Grafikfenster unter dem Punkt *File* ein *Redraw* zu fordern.