

Übungszettel Nr. 6 zur Vorlesung Bildverarbeitung im Sommersemester 05 an der BA Mannheim

Ausgabe am: 16.06.2005

Abgabe bis: 22.06.2005

Merkmalsextraktion und Klassifikation

Als Abschluss soll nun (fast) der ganze Weg der Bildverarbeitung begangen werden. Das heißt in unserem Fall von der Bildsegmentierung über die Merkmalsextraktion zur Klassifikation. Dazu soll zunächst ein simpler Klassifikator realisiert werden (Aufgabe 1). In Aufgabe 2 sollen dann Objekte aus Bildern segmentiert und die Merkmale zu den Objekten bestimmt werden. Im letzten Schritt sollen die Objekte anhand dieser Merkmale mittels eines euklidischen Abstandsklassifikators (Minimum-Distanz-Klassifikator) klassifiziert werden.

Hinweis: Um die Arbeit etwas zu erleichtern, haben die Bilder zu dieser Übung einen schwarzen Hintergrund (Intensitätswert 0). Alle Pixel, die zu den dargestellten Objekten gehören, haben Grauwerte größer als 0.

Aufgabe 1: Erstellung eines einfachen Klassifikators

Zunächst braucht man einige Testbilder für das Training des Klassifikators. Mit Hilfe dieser Bilder sollen für die einzelnen Objekte folgende Merkmale (mit zugehörigen Normierungsfaktoren) berechnet werden:

| Merkmal | Normierungsfaktor |
|------------|-------------------|
| Fläche | 0.0002 |
| Umfang | 0.003 |
| Formfaktor | 0.63 |
| Helligkeit | 0.006 |
| Kontrast | 0.00016 |

Die Normierungsfaktoren sorgen dafür, dass die Ergebnisse zwischen 0 und 1 liegen.

Für jedes Objekt erhalten Sie drei Testbilder (`Objekt1_1.bmp`, ..., `Objekt3_3.bmp`). Die Ergebnisse für jedes der 5 Merkmale können dann über die drei Beispiele gemittelt und als Prototypen für die anschließende Objektklassifikation genutzt werden. Entwickeln Sie einen Operator, der die drei Beispielbilder als Eingabe erwartet und ein Textobjekt ausgibt, das die gemittelten Werte für die 5 Merkmale enthält. Bitte genau einen Wert pro Zeile, ohne Text etc.

Aufgabe 2: Klassifikation von Objekten anhand ihrer Merkmale

Bei der Objekterkennung muss man zunächst die Objekte aus dem Testbild segmentieren (Grassfire). Anschließend werden für jedes Objekt die oben genannten Merkmale bestimmt und mittels folgender Formel mit den Prototypen aus Aufgabe 1 verglichen:

$$Fehler = \sqrt{\sum (Merkmal_{Prototyp} - Merkmal_{Objekt})^2} \quad (1)$$

Hier ein Beispiel, wie sie die Werte der Prototypen wieder aus den Textobjekten einlesen können:

```
AnsiString text, text1;
text = tx->data;
int pos= text.AnsiPos('\n');
text1 = text.SubString(1, pos-1);
Merkmal_1 = text1.ToDouble();

text= text.SubString(pos+1, text.Length());
pos= text.AnsiPos('\n');
text1= text.SubString(1, pos-1);
Merkmal_2= text1.ToDouble(); ...
```

Das zu untersuchende Objekt wird dann der Objektklasse zugeordnet, bei der der Fehler am geringsten ist.

Implementieren Sie einen Operator, der die drei Textausgabeobjekte aus Aufgabe 1 und das Bild mit den zu klassifizierenden Objekten (`Objektbild.bmp`) als Eingabe erwartet. Pro Objekt sollen die extrahierten Merkmale, das Ergebnis der Fehlerberechnung (pro Klasse) und das Klassifikationsergebnis (Zuordnung zu einer Klasse) als Text ausgegeben werden.

Würde sich für diese Aufgabe das Klassifikationsergebnis ändern, wenn statt des Abstandes zum Prototypen, der Abstand zum nächsten Nachbarn berechnet werden würde? Begründen Sie ihre Antwort.