

Übungszettel Nr. 1 zur Vorlesung Bildverarbeitung im Sommersemester 05 an der BA Mannheim

Ausgabe am: 13.04.2005

Abgabe bis: 20.04.2005

Binärbildverarbeitung und morphologische Operationen

Ziel dieses Übungszettels ist, dass sie sich mit der Binärbildverarbeitung sowie morphologischen Operationen innerhalb der Binärbildverarbeitung auseinandersetzen. Die beiden benötigten Eingabebilder heißen 'Objekte1.bmp' und 'Objekte2.bmp'. Sie wurden bereits mit ORASIS^{3D} installiert.

Ein Hinweis vorweg: In ORASIS^{3D} sind Binärbilder folgendermaßen definiert: Ein Pixelwert == 0 wird als FALSE interpretiert und alle Pixelwerte != 0 als TRUE.

Aufgabe 1

Benutzen sie für diese Aufgabe bitte das Bild Objekte1.bmp. Entwickeln sie eine Dilatations-Funktion und eine Erosions-Funktion bei denen strukturierendes Element die 'Elementarraute' auf das jeweilige Eingabebild angewendet werden soll.

Achtung: Sie haben bei dieser Funktion 3 Objekte in der TList (1 Eingabeobjekt, 1 Ausgabeobjekt und das vorgeschriebene Fehlerobjekt). Daher muss die Kontrolle, ob die richtige Anzahl von Objekten übertragen wurde, möglicherweise angepasst werden:

```
if (t1->Count != 3){
    ((TZI_Error*)t1->Last())->errorcode = 1001;
    return false;
}
```

Die Operatoren dieser Übung arbeiten auf Binärbildern und sollen auch nur darauf angewandt werden können. Dazu nehmen sie bitte folgende Anpassungen vor:

```
// setzen des Ausgabebildtyps ...
BYTE* outImageData = (BYTE*)outImage->setType(TYPE_BINARY);

// setzen des Typs zum Lesen des Eingabebildes ...
BYTE* inImageData = inImage->getDataAsBinary();

//Anpassung der Überprüfung des korrekten Eingabebildtyps
if (inImage->getType() != TYPE_BINARY){
    ((TZI_Error*)t1->Last())->errorcode = 1030; //gemaess error.ini
    return false;
}
```

Wenn sie die beiden Funktionen erzeugt haben, erstellen sie in ORASIS^{3D} ein Projekt, bei dem zunächst das Bild 'Objekte1.bmp' mit der Konvertierungsfunktion in ein Binärbild umwandelt. Wenden sie zunächst die Erosion auf das Bild an und auf das Ausgabebild dann die Dilatation. Beschreiben sie, was die Anwendung der Funktionen bewirkt? Können sie sich vorstellen was sich verändert, wenn statt einer 'Elementarraute' die '8er-Nachbarschaft' verwendet wird? Begründen sie ihre Antwort.

Aufgabe 2

Entwickeln sie eine Funktion, die die Grassfire-Methode zur Vereinzelung und Markierung von Objekten in Binärbildern realisiert. Die Markierung der vereinzelt Objekte soll durch unterschiedliche Grauwerte symbolisiert werden.

Wie der Algorithmus funktioniert wurde in der Vorlesung erklärt.

Ein Tipp bezüglich der Nummerierung:

Es läuft ein Zähler während der Berechnung mit, der bei jedem neuen Objekt inkrementiert wird. Im Ausgabebild werden die Objekte nicht in einem Binärbild sondern in einem Grauwertbild eingetragen.

```
// Das Ausgabebild auf Typ 8BITGRAY setzen
    BYTE* outImageData = (BYTE*)outImage->setType(TYPE_8BITGRAY);
```

Das erste Objekt wird mit dem Wert '10' gekennzeichnet, das zweite mit dem Wert '20' und so weiter. Der Hintergrund muss vorher auf 0 gesetzt worden sein.

Einige Hinweise:

Speicher für Zwischenbilder kann wie folgt reserviert werden:

```
    BYTE* ImgTmp = new BYTE[width * height];
```

Man kann dann auf die einzelnen Zellen wie folgt zugreifen:

```
    ImgTmp[y * width + x]= ...
```

Der Speicher ist am Ende natürlich wieder freizugeben:

```
    delete[] ImgTmp;
```

Wenden sie diese Funktion auf das Bild 'Objekte2.bmp' an. Argumentieren sie in welchen Bereichen der Bildverarbeitung dieser Operator angewendet werden kann. Denken sie dabei nicht nur an verschiedene Anwendungsszenarien sondern auch an das in der Vorlesung vorgestellte Mustererkennungs-Paradigma (bitte die einzelnen Stufen des Mustererkennungs-Paradigmas benennen, kurz erläutern und die Grassfire-Methode entsprechend einordnen).