

Übung 2D Grafik

Abgabetermin :

Die Lösung zu diesem Übungsblatt ist bis zum **24./26. Januar 2006** abzugeben.

Inhalt:

In diesem Übungsblatt wird die kontextabhängige Segmentierung sowie die Grundlagen für morphologische Operationen betrachtet.

Aufgabe 1 (P) Region Growing

In der Vorlesung und auf den Blättern 2 und 3 haben sie bereits Segmentierungsverfahren kennen gelernt, die ein Bild komplett in disjunkte Regionen zerlegt (Thresholding, Region-Labeling). In vielen Fällen ist man jedoch gar nicht an allen Bereichen eines Bildes interessiert, sondern lediglich an bestimmten Objekten (z.B. Personen, Körperteile). Insbesondere dann, wenn man die Segmentierung als Grundlage für spätere Bilderkennungsalgorithmen versteht. Beschränkt man sich auf bestimmte Bereiche, kann das mehrere Vorteile haben 1) Die Berechnungszeit sinkt, da nicht das ganze Bild betrachtet werden muß. 2) Durch interaktive Verfahren kann Domänenwissen od. Modellwissen in die Suchverfahren integriert werden was zu besseren Ergebnissen führt.

Das hier betrachtete Verfahren ist dem Region-Labeling ähnlich, ist jedoch nicht auf binär Bilder beschränkt. Um für Graustufenbilder oder Farbbilder zu entscheiden, ob eine Region R zusammenhängend ist oder nicht, muss ein Homogenitätskriterium definiert werden.

$$H(R) = \begin{cases} TRUE & , |f(j,k) - f(m,n)| \leq \Delta, \\ FALSE & , sonst \end{cases} \quad (1)$$

wobei (j,k) und (m,n) die Koordinaten zweier benachbarter Pixel in der Region R sind. Das Homogenitätskriterium 1 sagt also aus, dass die Region R genau dann zusammenhängend ist, wenn zwei beliebige, benachbarte Pixel ein Grauwertdifferenz $\leq \Delta$ aufweisen. Ein ähnliches, weit verbreitetes, Homogenitätskriterium ist folgendes:

$$H(R) = \begin{cases} TRUE & , |f(j,k) - \mu_R| \leq \Delta, \\ FALSE & , sonst \end{cases} \quad (2)$$

wobei $f(j,k)$ der Grauwert an Position (j,k) ist und μ_R der mittlere Grauwert in R ohne $f(j,k)$ ist.

Region Growing ist ein Flood-Fill Algorithmus, der genau ein zusammenhängendes Objekt in einem Bild findet. Als Startpunkt (eng. seed) kann jedes Pixel innerhalb der Zusammenhangskomponente dienen. Dabei wird ein Pixel zu der Region genau dann hinzugefügt wenn gilt: 1) Es gehört noch zu keiner Region. 2) Es ist ein direkter Nachbar eines Pixels aus R (4- oder 8-Nachbarschaft). 3) Die neue Region erfüllt nach Hinzunahme des Pixels immer noch das Homogenitätskriterium.

- Implementieren sie die Klasse *RegionGrower*, die den Algorithmus implementiert. Implementieren sie außerdem die nötigen UI Kontrollelemente, um interaktiv das Startpixel, das Homogenitätskriterium (1,2) und den Threshold auszuwählen.

- b) Implementieren sie die Methode *growToCompletion()* und *getRegionImage()*, die den Algorithmus ausführen, respektive das Ergebniss als *BufferedImage* zurück liefern.
- c) Implementieren sie außerdem die Methoden *grow()*, *getStatusImage()* und *isFinished()*. Verwenden sie diese Methoden um die Iterationsschritte in Echtzeit anzuzeigen.
- d) Laden sie sich das Bild '2DÜbung_Dice.jpg' von der Übungsseite herunter und experimentieren sie mit unterschiedlichen Thresholdwerten.

Aufgabe 2 (P) Morphologische Operationen

Durch die Segmentierung von Bildern, erhält man eine Trennung von Vorder- und Hintergrund. Viele Segmentierungsverfahren leiden unter Störungen in den Ergebnissbildern. Morphologische Operationen werden im wesentlichen dazu verwendet, um die Ergebnisse eines Segmentierungsschrittes nach zu bearbeiten und diese Störungen zu unterdrücken. Dazu werden die formbasierten Eigenschaften (die Morphologie) der erhaltenen Objekte betrachtet.

Dilatation und *Erosion* sind die Basis von allen morphologischen Operationen. Die Erosion verkleinert die Ausdehnung eines Segmentes und Dilatation vergrößert ein Segment. Alle morphologischen Operationen sind Verknüpfungen eines Binärbildes mit einem *Strukturelement*. Das Strukturelement ist, ähnlich dem Kernel der Konvolution, eine Pixelmatrix (meist ungerader Dimension) mit den Werten 1 od. 0. Die Art bzw. Richtung der Ausdehnung und Verkleinerung von Bildobjekten lässt sich durch die Form des Strukturelementes anpassen (d.h. durch die Besetzung der Matrix)

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & \mathbf{1} & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 \\ 1 & \mathbf{1} & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 \\ 1 & \mathbf{0} & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (3)$$

Jedes Strukturelement hat einen Ankerpunkt (fett gedruckt in 3). Dieser muss nicht immer im Mittelpunkt der Matrix sein in der Regel ist dies jedoch der Fall.

Eine morphologische Operation hat die Form $g = f_s \circ s$. Der Operator \circ ist eine der Mengenoperationen Schnitt, Vereinigung oder Differenz. Diese Operationen werden auf der Menge von Segmentpixeln f_s und dem Strukturelement s ausgeführt. Das Resultat wird an die Position des Ankerpunktes geschrieben. Der Wert von g wird berechnet indem das Strukturelement an alle möglichen Positionen verschoben wird und dann die Vereinigung der Resultate gebildet wird.

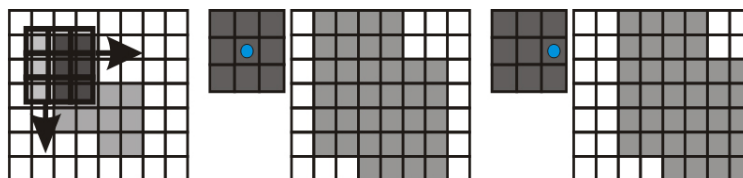


Abbildung 1: Bei der Dilatation wird eine Pixelposition auf 1 gesetzt, falls das Strukturelement mindestens ein Pixel des Segmentes überdeckt, das den Wert 1 hat. Eine Translation des Ankerpunktes führt zu einer entsprechenden Translation des Resultates.

Die Dilatation eines Bildes b (siehe Abb. 1) ist definiert als

$$g(m, n) = \vee_{(m_k, n_k) \in s} b(m + m_k, n + n_k). \quad (4)$$

Dies wird meist durch $g = f \oplus s$ abgekürzt. Das Pixel an (m, n) wird gesetzt wenn eines der durch s überdeckten Pixel von f zum Segment gehört. Die Dilatation führt dazu, dass Löcher in

Vordergrundsegmenten geschlossen, nah beieinander liegende Segmente vereinigt und Segmente größer werden.

Die *Erosion* (siehe Abb. 2) ist genau das Gegenteil der Dilatation. Die Erosion des Bildes b mit

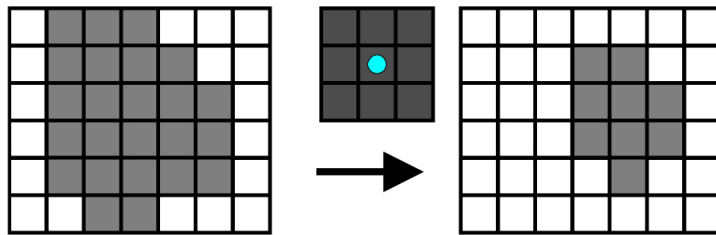


Abbildung 2: Bei der Erosion werden alle Pixel des Bildes auf 0 gesetzt, für die das Strukturelement mindsetens ein Pixel überdeckt, das den Wert 0 hat.

dem Strukturelement s ist definiert als

$$g(m, n) = \bigwedge_{(m_k, n_k) \in s} b(m + m_k, n + n_k). \quad (5)$$

Die Erosion wird durch das Zeichen \ominus abgekürzt. Der Unterschied zur Dilatation besteht also darin, dass statt eines Bool'schen 'oder' ein Bool'sches 'und' ausgeführt wird. Das Ergebnis ist also nur 1, wenn alle durch das Strukturelement überdeckten Pixel von b zu den Vordergrundpixeln gehören. Damit werden Löcher in Segmenten vergrößert und die Ausdehnung des Segments schrumpft.

- Laden sie sich von der Übungsseite das Codegerüst 'MorphologicalOperations.jar'. Dieses enthält die Klassen *StructElement(abstract)*, *BinaryStructElement* und *BinaryMorphologicalOp*.
- In der Klasse *BinaryStructElement*, implementieren sie die Methode *fits(Raster raster, int x, int y)* die ausgibt ob **alle** Pixel unter dem übergebenen Strukturelement '1' sind. Implementieren sie außerdem die Methode *hits(Raster raster, int x, int y)* die ausgibt ob **ein beliebiges** Pixel des Strukturelements mit einem Pixel dessen Wert 1 ist korrespondiert. Beachten sie dabei, dass Pixel ignoriert werden, wenn der korrespondierende Wert des Strukturelementes 0 ist.
- Implementieren sie die Klassen *DilationOp* und *ErosionOp*, die die Dilatation bzw. Erosion implementieren und eine Instanz der *BinaryStructElement* Klasse als Parameter des Konstruktors erwarten. Beide Klassen erweitern die *BinaryMorphologicalOp* Klasse.
- Fügen sie die nötigen UI Kontrollelemente hinzu um die Operationen auf Binärbilder anwenden zu können. Laden sie sich das Bild '2DÜbung_Morph.jpg' von der Übungsseite herunter und experimentieren sie mit den Operationen.