

## Computergrafik 1

### Abgabetermin:

Die Lösung zu diesem Übungsblatt ist bis zum Freitag den **17. Juli 2009, 12:00 Uhr s.t.** (d.h. in **2 Wochen**) per Email abzugeben.

### Inhalt:

In diesem Blatt erweitern Sie Ihr bereits erstelltes Fenster um weitere Filterfunktionen. Dieses Mal beschäftigen Sie sich mit *Region Growing* und *Segmentierung*. Die grundlegenden Konzepte haben Sie in der Vorlesung bereits kennen gelernt, hier sollen Sie Ihr Wissen praktisch anwenden und vertiefen. In diesem Blatt können maximal 40 Punkte erreicht werden. Zum Bestehen des Übungsblatts müssen also **MINDESTENS** 10 Punkte in jeder Aufgabe erreicht werden.

Geben Sie insgesamt (d.h. ein Projekt für alle Aufgaben zusammen) alle benötigten Header-, Source-, und Projektdateien (von *Qt Creator* erzeugt) mit ab, d.h. \*.h, \*.cpp und \*.pro Dateien. Abgaben die nicht kompilieren werden mit 0 Punkten bewertet. Fassen Sie alle Aufgaben zu einer zip-Datei zusammen und senden Sie diese an: `cg1_ss09@medien.ifi.lmu.de`.

### Aufgabe 30 Region Growing

(20 Punkte)

In dieser Aufgabe werden Sie das *Region Growing* schrittweise implementieren. Dabei werden Sie Ihr Frame dieses Mal um einen Filter erweitern. Das hier betrachtete Verfahren ist dem Region-Labeling ähnlich, aber 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  $(j,k)$  und  $\mu_R$  der mittlere Grauwert in  $R$  ohne  $f(j,k)$  ist.

*Region Growing* ist ein sogenannter 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.

- a) Implementieren Sie einen Filter namens *RegionGrow*, der den Algorithmus später implementiert. Implementieren Sie außerdem ein weiteres Menü namens *Other Filters* und einen Menüpunkt *Region Growing* (**2 Punkte**).
- b) Implementieren Sie einen Dialog, der dem Benutzer erlaubt, das Homogenitätskriterium (siehe Formeln 1, und 2) und den Threshold anzugeben (**3 Punkte**).
- c) Nach der Bestätigung des Dialogs soll der Benutzer dann die Möglichkeit haben, auf dem bereits zuvor aktiven Bild (d.h. MDI-Fenster) den Startpunkt (d.h. *seed*) interaktiv mit der Maus zu wählen (**3 Punkte**).
- d) Implementieren Sie nun den Filter, der den *Region Growing* Algorithmus auf dem aktuellen Bild unter Berücksichtigung der Eingaben durchführt (**12 Punkte**).

### Aufgabe 31 Segmentierung, Region-Labeling

(20 Punkte)

In der Vorlesung haben Sie gelernt, was Pixel-Nachbarschaft, Pixel-Pfade und Zusammenhangskomponenten sind. In dieser Aufgabe sollen Sie mit einem einfachen Nachbarschafts-Algorithmus Zusammenhangskomponenten in einem Binärbild finden. Verwenden Sie hierbei die 8er-Nachbarschaft. Der zu verwendende Algorithmus heißt 'Grassfire Labeling'<sup>1</sup>. Dieser Algorithmus arbeitet auf einem Ausgangs- und einem Zielbild. Der Algorithmus scannt ein Binärbild zeilenweise. Bei jedem Pixel das sich von Null unterscheidet, wird dann eine rekursive Labeling-Prozedur aufgerufen. Diese Prozedur fängt damit an, das Startpixel im Ausgangsbild auf 0 zu setzen und im Zielbild wird dem Pixel der Wert des aktuellen Labels zugewiesen. Danach werden alle 8er-Nachbarn rekursiv betrachtet. Am Ende der Prozedur sind alle Pixel der Zusammenhangskomponente im Ausgangsbild 0 (also schwarz, bzw. *verbrannt*). Im Zielbild haben Sie den Wert des aktuellen Labels. Am Schluß wird der Wert des Labels inkrementiert und das Scannen geht weiter bis das nächste Pixel ungleich 0 gefunden wird.

- a) Implementieren Sie einen Filter namens *RegionLabeling*, der den Algorithmus später implementiert. Fügen Sie außerdem einen weiteren Menüpunkt *Region Labeling* und fügen Sie diesen zu dem bereits erstellten Menü *Other Filters* hinzu (**2 Punkte**).
- b) Implementieren Sie nun den Filter und verwenden Sie dabei den oben erwähnten *Grassfire*-Algorithmus (**12 Punkte**).
- c) Färben Sie nun die gefundenen Zusammenhangskomponenten je nach Label in einer unterschiedlichen Farbe ein (**6 Punkte**).

---

<sup>1</sup>*Pitas, I.: Digital Image Processing Algorithms, Prentice Hall (1993)*