HOCHSCHULE
HANNOVER
UNIVERSITY OF
APPLIED SCIENCES
AND ARTS
–
Fakultät IV
Wirtschaft und
Informatik

Übungen zur Vorlesung Computergrafik 2

Wintersemester 2022/2023

Prof. Dr. Ingo Ginkel

Aufgabe 3: Kantenfilter

Erweitern Sie die Funktionalität ihrer Filter so, dass auch Kantendetektionsfilter realisiert werden können. Konkret soll es also möglich sein, dass die Tabelle im GUI für ihre bisherigen Filter auch die entsprechenden Einträge für Kantendetektionsfilter enthalten kann.

a) Passen Sie dazu die Gewichtung der Filter-Matrix Inhalte entsprechend an, falls es sich um Ableitungsfilter handelt.

b) Speichern Sie das Ergebins so ab, dass durch Umskalierung der "normale" Wertebereich $[0, 255]$ genutzt werden kann. Bei der Verwendung der Ableitungen soll diese Rechnung dann wieder rückgängig gemacht werden.

Aufgabe 4: Kantendetektion und Schärfung

a) Implementieren Sie einen Canny Edge Detektor.
Alle Steuerungs-Parameter sollen aus dem GUI übernommen werden. Bei Änderung einzelner Parameter sollen nur diejenigen Teil-Berechnungen neu gemacht werden für die dies notwendig ist.

b) Implementieren Sie einen USM-Filter.
Auch hier sollen alle Steuerungs-Parameter aus dem GUI übernommen werden.

Hinweise:

Die Implementierung erfolgt im Verzeichnis 'Sheet3' in der Datei `edgefilter.cpp`. Folgende Methoden sind für die jeweilige Funktionalität vorgegeben und sollen mit Inhalt gefüllt werden:

- `QImage* doEdgeFilter(QImage * image, int* derivative_filter, int smoothing_filter, int desired_image);`
- `QImage* doLaplaceFilter(QImage * image, int** laplace_filter);`
- `QImage* doCanny(QImage * img, double sigma, int tHi, int tLo);`
- `QImage* doUSM(QImage * image, double sharpening_value, double sigma, int tc);`

Die Bedeutung und Wertebereiche der Übergabeparameter ist im Code oberhalb des Methodenrumpfes dokumentiert.