

Computergrafik 1

Abgabetermin:

Die Lösung zu diesem Übungsblatt ist bis zum Freitag den **3. Juli 2009, 12:00 Uhr s.t.** (d.h. in **3 Wochen**) per Email abzugeben.

Inhalt:

In diesem Blatt erweitern Sie Ihr bereits erstelltes Fenster um weitere Filterfunktionen. Dieses Mal beschäftigen Sie sich mit sogenannten *Konvolutions-* und *Morphologischen Filtern*. Ferner werden Sie die *Hough-Transformation* kennenlernen und einsetzen. Pro Aufgabe können maximal 20 Punkte erreicht werden. Zum Bestehen des Übungsblatts müssen in **JEDER** Aufgabe mindestens zehn Punkte 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.ifl.lmu.de.

Aufgabe 26 Konvolutionsfilter

(20 Punkte)

In dieser Aufgabe erlernen Sie den Umgang mit *Konvolutionsfiltern* am Beispiel von Weich- und Scharfzeichnern. Die Standardimplementierungen der Filter sollen keine Dialoge enthalten.

- Fügen Sie ein neues Untermenü *Convolution* in Ihr bereits erstelltes Menü *Filters* ein. Dieses Menü werden Sie im Folgenden schrittweise erweitern (2 Punkte).
- Erstellen Sie sich zunächst eine Klasse namens *ConvolutionFilter*, die wiederum von *ImageFilter* erbt. Diese Klasse soll folgendermaßen implementiert werden:
 - Ein *ConvolutionFilter* enthält eine quadratische Matrix beliebiger Dimension (**Hinweis:** Üblicherweise ist die Dimension ungerade und im Bereich von 3 bis 25). Diese wird von den erbbenden Klassen bereitgestellt und später den sogenannten *Kernel* für die Funktion `apply` bereitstellen. Dazu ist es sinnvoll auf einen Standardkonstruktor zu verzichten, damit erbende Klassen grundsätzlich einen *Kernel* beim Erzeugen übergeben müssen.
 - Die Funktion `apply` soll diesen *Kernel* nun verwenden. Dazu muss für jedes Pixel des Zielbildes das zugehörige Pixel des Quellbildes mit dem *Kernel* verarbeitet werden. Die Formel für die Verarbeitung des alten Pixels (P' an der Stelle i, j) mit dem *Kernel* K der Dimension n (n ungerade, $n \geq 3$) lautet:

$$P_{i,j} = q \cdot \sum_{x=1}^n \left[\sum_{y=1}^n \left(k_{x,y} \cdot P'_{i+x-(n+1)/2, j+y-(n+1)/2} \right) \right];$$

wobei für den Faktor q gilt: $q = \frac{1}{\sum_{x=1}^n \sum_{y=1}^n k_{x,y}}$;

Beachten Sie, dass diese Berechnung am Rand des Bildes problematisch werden kann, da der *Kernel* eventuell über das Bild hinausragt. Verwenden Sie daher nur die möglichen Werte des *Kernels* (d.h. alle die, die noch **innerhalb** des Bildes liegen). Passen Sie außerdem in einem solchen Fall den Faktor q entsprechend an (**9 Punkte**).

- c) Erstellen sie nun eine neue Klasse namens *Blur*, die wiederum von *ConvolutionFilter* erbt. Der zu übergebende *Kernel* soll folgendermaßen aussehen:

$$\begin{bmatrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 4 & 5 & 4 & 2 \\ 3 & 5 & 6 & 5 & 3 \\ 2 & 4 & 5 & 4 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{bmatrix}$$

Fügen Sie ferner einen neuen Menüpunkt *Blur* in das *Convolution* Menü ein, durch den der Weichzeichner auf dem aktuell aktiven Bild durchgeführt wird (**3 Punkte**).

- d) Erstellen sie nun eine neue Klasse namens *Sharpen*, die wiederum von *ConvolutionFilter* erbt. Der zu übergebende *Kernel* soll folgendermaßen aussehen:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Fügen Sie ferner einen neuen Menüpunkt *Sharpen* in das *Convolution* Menü ein, durch den der Scharfzeichner auf dem aktuell aktiven Bild durchgeführt wird (**3 Punkte**).

- e) Erstellen sie nun eine neue Klasse namens *Edges*, die wiederum von *ConvolutionFilter* erbt. Der zu übergebende *Kernel* soll folgendermaßen aussehen:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Fügen Sie ferner einen neuen Menüpunkt *Edges* in das *Convolution* Menü ein, durch den der Kantenerkennung auf dem aktuell aktiven Bild durchgeführt wird (**3 Punkte**).

Aufgabe 27 Morphologische Filter

(20 Punkte)

In dieser Aufgabe erlernen Sie den Umgang mit *Morphologischen Filtern* am Beispiel von *Dilatation* und *Erosion*. Die Standardimplementierungen der Filter sollen keine Dialoge enthalten.

- a) Fügen Sie ein neues Untermenü *Morphology* in Ihr bereits erstelltes Menü *Filters* ein. Dieses Menü werden Sie im Folgenden schrittweise erweitern (**2 Punkte**).
- b) Erstellen Sie sich nun zwei Klassen namens *QuadraticDilatation* und *CircularDilatation*, die wiederum von *ImageFilter* erben. Die Klassen sollen folgendermaßen implementiert werden (**Hinweis**: Da beide Filter die gleiche Berechnung abhängig vom *Kernel* durchführen, kann die *CircularDilatation* selbstverständlich auch von *QuadraticDilatation* erben):

- Die *Dilatation* enthält eine quadratische Matrix beliebiger Dimension (**Hinweis**: Üblicherweise ist die Dimension ungerade und im Bereich von 3 bis 25). Analog zu den Konvolutionsfiltern spiegelt diese ebenfalls den *Kernel* wieder.

- Die Funktion `apply` soll diesen *Kernel* nun verwenden. Dazu muss für jedes Pixel des Zielbildes das zugehörige Pixel des Quellbildes mit dem *Kernel* verarbeitet werden. Die Formel für die Verarbeitung des alten Pixels (P' an der Stelle i, j) mit dem *Kernel* S der Dimension n (n ungerade, $n \geq 3$) lautet:

$$P_{i,j} = \max\{P'_{i+x,j+y}, x, y \in D_S\};$$

wobei für D_S gilt: $x, y \in D_S$ falls $S_{x,y} = 1$;

Beachten Sie, dass diese Berechnung am Rand des Bildes problematisch werden kann, da der *Kernel* eventuell über das Bild hinausragt. Verwenden Sie daher nur die möglichen Werte des *Kernels* (d.h. alle die, die noch **innerhalb** des Bildes liegen). Erstellen Sie nun die *Kernel* (Strukturelemente) der zwei Operatoren (d.h. *quadratisch* und *kreisförmig*) mit einer Dimension von $n = 9$. Fügen Sie ferner zwei neue Menüpunkte *Quadratic Dilatation* und *Circular Dilatation* in das *Morphology* Menü ein, durch die die jeweiligen Filter auf dem aktuell aktiven Bild durchgeführt werden (**9 Punkte**).

- c) Erstellen Sie sich nun zwei Klassen namens *QuadraticErosion* und *CircularErosion*, die wiederum von *ImageFilter* erben. Die Klassen sollen folgendermaßen implementiert werden (**Hinweis**: Da beide Filter die gleiche Berechnung abhängig vom *Kernel* durchführen, kann die *CircularErosion* selbstverständlich auch von *QuadraticErosion* erben):

- Die *Erosion* enthält eine quadratische Matrix beliebiger Dimension (**Hinweis**: Üblicherweise ist die Dimension ungerade und im Bereich von 3 bis 25). Analog zu der *Dilatation* spiegelt diese ebenfalls den *Kernel* wieder.
- Die Funktion `apply` soll diesen *Kernel* nun verwenden. Dazu muss für jedes Pixel des Zielbildes das zugehörige Pixel des Quellbildes mit dem *Kernel* verarbeitet werden. Die Formel für die Verarbeitung des alten Pixels (P' an der Stelle i, j) mit dem *Kernel* S der Dimension n (n ungerade, $n \geq 3$) lautet:

$$P_{i,j} = \min\{P'_{i+x,j+y}, x, y \in D_S\};$$

wobei für D_S gilt: $x, y \in D_S$ falls $S_{x,y} = 1$;

Beachten Sie, dass diese Berechnung am Rand des Bildes problematisch werden kann, da der *Kernel* eventuell über das Bild hinausragt. Verwenden Sie daher nur die möglichen Werte des *Kernels* (d.h. alle die, die noch **innerhalb** des Bildes liegen). Erstellen Sie nun die *Kernel* (Strukturelemente) der zwei Operatoren (d.h. *quadratisch* und *kreisförmig*) mit einer Dimension von $n = 9$ analog zur *Dilatation*. Fügen Sie ferner zwei neue Menüpunkte *Quadratic Erosion* und *Circular Erosion* in das *Morphology* Menü ein, durch die die jeweiligen Filter auf dem aktuell aktiven Bild durchgeführt werden (**9 Punkte**).

Aufgabe 28 Hough-Transformation

(20 Punkte)

In dieser Aufgabe erlernen Sie den Umgang mit der *Hough-Transformation* für das Auffinden und Beschreiben von Linien in einem Bild. Dieser Filter soll einen Dialog enthalten. **Hinweis**: Arbeiten Sie für die *Hough-Transformation* mit binären Bildern, d.h. Graustufenbilder, die lediglich schwarze und weiße Pixel besitzen.

- a) Erstellen Sie eine Klasse namens *HoughTransform*, die von *ImageFilter* erbt. Fügen Sie ferner ein neues Untermenü namens *Transformation* hinzu, das dann den Menüpunkt *Hough Transform* enthält (**2 Punkte**).
- b) Erstellen Sie einen Dialog, der den Benutzer angeben lässt, **wie viele** Geraden im Bild gefunden werden sollen (**2 Punkte**).

c) Die *Hough-Transformation* soll wie folgt arbeiten:

- Jede Gerade in einem Bild kann durch ein Tupel (d, α) beschrieben werden, wobei d den Abstand der Gerade zum Bildmittelpunkt und α den Winkel der Normalen der Geraden beschreibt. Daraus ergibt sich folgende Formel für eine Gerade:

$$d = x \cdot \cos(\alpha) + y \cdot \sin(\alpha);$$

- Der *Hough-Raum* besitzt einen sogenannten *Akkumulator*, der – je größer der Wert – genau das Tupel (d, α) angibt, das eine Gerade beschreibt. Dieser *Akkumulator* ist ein zwei-dimensionales Feld mit den Dimensionen $x = \text{DiagonalesBildes}$ und $y = 180 \cdot \text{Anzahl pro Grad}$.
 - Zu Beginn der Transformation sind alle Elemente des *Akkumulators* 0. Um diesen *Akkumulator* zu füllen, muss für jedes Pixel, das weiß ist jede mögliche Gerade bestimmt werden, die durch dieses führen könnte. Dazu müssen Sie für jedes nicht-schwarze Pixel den möglichen Abstand d zu jedem möglichen Winkel α ($180 \cdot \text{Anzahl pro Grad}$) berechnen. Die beiden Werte d und α liefern Ihnen dann die entsprechenden Koordinaten im *Akkumulator*. An dieser Stelle erhöhen Sie dann den Wert im *Akkumulator*.
 - Gehen Sie nun durch den *Akkumulator* und finden Sie maximal die vom Benutzer angegebene Anzahl an Linien. Sollte ein Wert unterhalb des Schwellwertes (= 50% des maximalen Wertes im *Akkumulator*) so wird er ignoriert. Es kann daher sein, dass weniger Linien gefunden werden, als angegeben.
 - Achten Sie bei der Erkennung der Linien darauf, dass um einen hohen Wert im *Akkumulator* üblicherweise auch die benachbarten Felder sehr hoch bewertet sind. Diese gehören jedoch zu dieser Linie und sollten nicht als eigene Linie erkannt werden. **Hinweis:** Verwenden Sie ein „Fenster“ mit einer gewissen Größe, in dem jeweils nur ein Maximalwert auftreten darf (**8 Punkte**).
- d) Zeichnen Sie nun die erkannten Linien in das Bild ein. Verwenden Sie hierfür eine Farbe, die nicht im Graubereich liegt (z.B. *Rot*). Achten Sie ferner darauf, dass die Linien über den Rand des Bildes gehen sollen, d.h. es sind keine Geraden mit festem (d.h. erkennbarem) Anfang und Ende. Dazu müssen Sie das Bild jedoch zuvor (falls es in Graustufen vorliegt) noch in ein 24-bit Bild umwandeln. Verwenden Sie zur Berechnung der Linien die oben angegebene Formel. Geben Sie anschließend dieses Bild zurück (**8 Punkte**).