

Filterung im Ortsbereich

Aufgabenstellung:

Es sind zwei Unterprogramme zur Bildfilterung zu schreiben

- a) Medianoperator,
- b) Sobel-Operator.

Die korrekte Funktionsweise der Filter ist zu überprüfen.

Implementierungshinweise:

In die vorbereitete Funktionshülle (RV_02)

```
void RV02::operator() (int argc, char *argv[]) {  
    .....  
}
```

sind die Unterprogramme *Median(...)* und *Sobel(...)* der Aufgabenstellung einzufügen.

```
a) void RV02::Median(const channel8& sPic,  
                    channel8& dPic,  
                    const int MaskSizeX,  
                    const int MaskSizeY)
```

Zweck: Berechnet aus dem Quellbild *sPic* das Median-gefilterte Bild *dPic*. Die Größe der Medianmaske soll für die beiden Koordinatenrichtungen (x und y) unabhängig vorgebar sein. Aus Symmetriegründen sollen nur ungerade Maskengrößen erlaubt sein, z.B. 3x3, 5x17, 51x51. Fehleingaben sollen automatisch korrigiert werden (z.B. 2x16 --> 3x17).

Hinweis zur Arbeitsweise:

Den Medianwert einer Menge von Grauwerten erhält man üblicherweise durch folgende Verarbeitungsschritte:

- a) Sortieren der Maskengrauwerte nach ihrer Größe,
- b) Auswahl des in der Mitte der Sortierung stehenden Grauwertes.

Diese Implementierungsvariante ist jedoch für große Maskengrößen durch die notwendige Sortierung uneffizient.

Besser ist dann folgendes Vorgehen (s. nachfolgendes Bild):

- a) Für die Grauwerte der Maskenpixel wird das Histogramm berechnet,
- b) im Maskenhistogramm wird dann der Medianwert bestimmt.

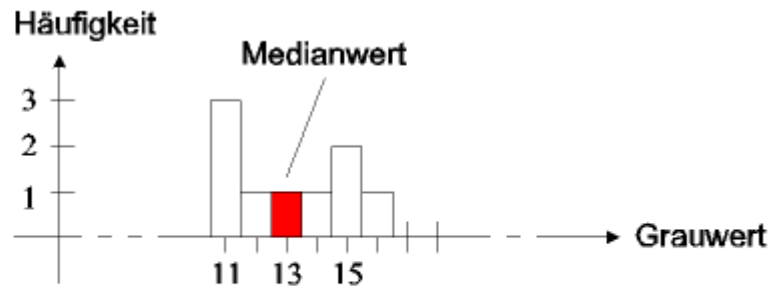
Anm.: Der Median ist derjenige Grauwert, wo das akkumulierte Histogramm gerade den Wert $(\text{Maskengröße}_x * \text{Maskengröße}_y + 1) / 2$ hat.

Das zu entwickelnde Unterprogramm soll nach diesem Verfahren arbeiten.

Filterung im Ortsbereich

	11	13	15
	11	11	15
	12	14	16

Bildausschnitt mit
Medianmaske



Histogramm

Für die Ablaufgeschwindigkeit ist es von Vorteil, wenn bei der Histogrammberechnung der Minimal- und Maximalwert der betrachteten Bildumgebung ermittelt wird. Bei der Histogrammsuche kann man dann die Suche auf den Bereich zwischen Min- und Maxwert beschränken.

```
b) void RV02::Sobel(const channel8& sPic,
                    channel8& GradientPic,
                    channel8& DirectionPic)
```

Zweck: Berechnet aus dem Quellbild `sPic` das Sobel-gefilterte Bild `GradientPic` sowie das Richtungsbild `DirectionPic`.

Hinweis zur Arbeitsweise:

Gradientenwert und -richtung eines Bildpunktes erhält man durch

- a) Berechnung der Richtungsableitungen G_x und G_y mit den Faltungsmasken:

-1	0	1
-2	0	2
-1	0	1

-1	-2	-1
0	0	0
1	2	1

- b) Berechnung des Gradienten mit:

$$G = \sqrt{G_x^2 + G_y^2}$$

Filterung im Ortsbereich

Das Ergebnis ist auf den Zahlenbereich 0..255 zu normieren.

c) Berechnung der Gradientenrichtung mit:

$$\alpha = \arctan\left(\frac{G_y}{G_x}\right)$$

Die Gradientenrichtung soll wie folgt normiert werden:

- 22.5° ... 22.5°	--> 0
22.5° ... 67.5°	--> 1
67.5° ... 112.5°	--> 2
112.5° ... 157.5°	--> 3
157.5° ... 202.5°	--> 4
usw.	

Vorzubereiten:

Arbeitsweise der zu entwickelnden Filter (Median, Sobel) verstehen.

Aufgabenbearbeitung:

Fertigzustellen sind

- Sourcecode der entwickelten Software.
- Ergebnisbilder, welche die Arbeitsweise der Filter dokumentieren.