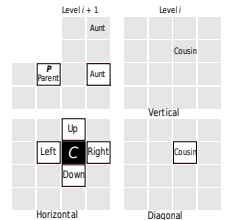


## 4. Übung: Lineare Nachbarschaftsfilter

Michael Grunwald, Pascal Laube, Theresa Kocher

### Aufgabe 1

- Aufgabe 1.1 Implementieren Sie eine Funktion, welche es erlaubt ein Bild (im sinnvollen Rahmen) mit frei wählbaren Filtermasken zu falten.
  - Prototyp: `[ out_image ] = filter(in_image, filter, off);`  
`out_image` Ergebnisbild nach Faltung von `in_image` mit `filter`  
`in_image` Eingangsbild (int)  
`filter` Filtermatrix (float)  
`off` Offset (int)
  - 8-Bit Graustufenbilder als Eingangs- und Ausgangsdaten.
  - Filtermatrix der Größe (N×N) mit  $N = (2K + 1)$ ,  $K = 1, 2, \dots$
  - Ankerpunkt (Hot Spot) ist die Mitte der Filtermatrix.
  - Beispielaufruf: `filter(image, [1 1 1; 1 3 1; 1 1 1], 0);`
- Lesen Sie Kapitel 6 (Filter) in "Digitale Bildverarbeitung".



### Aufgabe 2

- Erweitern Sie die Funktion `filter` um folgende Randbedingungen:
  - min  
Setzt Bildpunkte auf den minimalen Wert (0)
  - max  
Setzt Bildpunkte auf den maximalen Wert (255)
  - mirror  
Spiegelt den Bildinhalt an den Bildkanten
  - continue  
Setzt das Bild außerhalb mit dem gleichen Pixelwert, wie das entsprechende am nächsten liegende Randpixel, fort.
- Prototyp: `[ out_image ] = filter(in_image, filter, off, edge);`  
`edge`: Parameter zur Auswahl der Randbehandlung ('min') – String
- Untersuchen Sie die Randbehandlungen auf ihr Verhalten bei Benutzung verschiedener Filter.

## 4. Übung: Lineare Nachbarschaftsfilter

Michael Grunwald, Pascal Laube, Theresa Kocher

---

### Aufgabe 3

Beantworten Sie folgende Fragen (siehe Buch ab S. 99).

- Aufgabe 3.1 Nennen Sie die Arten und Eigenschaften von linearen Filtern.
- Aufgabe 3.2 Was ist der Unterschied zwischen linearen und nichtlinearen Filtern?

### Bemerkung

- Die Aufgaben werden elektronisch (tkocher@htwg-konstanz.de) vor der nächsten Übungsstunde abgegeben.