



Aufgabe 2: Bildverarbeitung mit Farbbildern

Erweitern Sie - falls nicht sowieso schon geschehen - ihre Funktionalitäten so, dass alle Operationen auch auf Farb-Bilder angewendet werden können, indem diese ins YCbCr-Modell konvertiert werden, die Anpassungen auf dem Helligkeits-Kanal ausgeführt werden und anschließend wieder ein rgb Bild erzeugt wird. Es soll also nicht nur ein Farbbild geladen und in Graustufen angezeigt, sondern jetzt auch ein verändertes Farbbild dargestellt werden. Dies soll sowohl für die bisherigen als auch für die zukünftigen Berechnungen möglich sein.

Aufgabe 3: Lineare Filter

- a) Implementieren Sie einen Linearen Filter mit generischer Größe. Dabei sind folgende Features erwünscht:
 - Die Größe des Filters kann in ihrem GUI eingestellt werden (nur ungerade Größen erlaubt, x,y-Richtung separat einstellbar),
 - Die Koeffizienten werden ganzzahlig eingetragen, Normierung passiert dann in ihrem Programm automatisch.
 - Alle Berechnungen sollen wenn möglich als Integer-Operationen ausgeführt werden,
 - Der Rand des Bildes soll zunächst unbearbeitet bleiben (Größe dieses Randes soll automatisch von der Filtergröße abhängen).
- b) Erweitern sie die Funktionalität von a) so dass auch der Randbereich des Bildes bearbeitet werden kann. Implementieren sie Zero-Padding, konstante und gespiegelte Ränder ohne zusätzlichen Speicherverbrauch (also kein größeres Array anlegen um den erweiterten Rand zu speichern)!
- c) Implementieren Sie einen 2D-Gaussfilter durch Nutzung der Separierbarkeit als Hintereinanderausführung zweier 1D-Gauss-Filter. Der Parameter σ soll wie in der Vorlesung die Breite des Filterkerns bestimmen.

Hinweise:

Die Implementierung erfolgt im Verzeichnis 'Sheet2' in der Datei `filteroperations.cpp`. Folgende Methoden sind für die jeweilige Funktionalität vorgegeben und sollen mit Inhalt gefüllt werden:

- `QImage* filterImage(QImage * image, int**& filter, int filter_width, int filter_height, int border_treatment)`
- `QImage* filterGauss2D(QImage * image, double gauss_sigma, int border_treatment)`

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