

Praktikum 2: Filterung und Vorbereitung zur Segmentierung der Bilddaten

1. Lernziele

- Schreiben und Verwendung eigener Funktionen
- Einbinden und Verwendung von externen Bibliotheken einschließlich des Gebrauchs der Dokumentation
- Anwendung von Rauschunterdrückungsverfahren zur Vorverarbeitung der Bilddaten
- Anwendung von Segmentierungsverfahren

2. Aufgaben

Vervollständigen Sie Ihre bisherige Implementierung um die folgenden Auswertungen:

- Schreiben Sie eine Funktion zur Auswahl des Filteralgorithmus!
- Schreiben Sie eine Funktion zum Zeichnen eines Histogramms!
- Schreiben Sie eine Funktion zur Durchführen der Schwellwertsegmentierung! (Diese wird im dritten Praktikum vervollständigt.)

Nutzen Sie zur Bearbeitung des zweiten und dritten Praktikums den im Ordner bereitgestellten medizinischen Datensatz von MR-Angiographien. Dies sind Datensätze, die durch die Differenzbildung zwischen einer normalen Magnetresonanz-Tomographie (MRT) und der MRT nach Kontrastmittelgabe erzeugt wurden. Durch die Differenzbildung sind die Bilder stark verrauscht und enthalten hauptsächlich Informationen über die Blutgefäße.

2.1. Funktion zur Auswahl des Filteralgorithmus

Eingabeparameter: Eingabebild (Referenz auf ein Bild von Typ `cv::Mat`), Filterergebnis (Referenz auf ein weiteres Bild von Typ `cv::Mat`), Integervariable zur Wahl des Filteralgorithmus, Integervariable zur Wahl der Filtergröße

Abhängig von der übergebenen Wahl soll entweder ein Gauß-Filter, ein Median-Filter oder ein Bilateral Filter auf das Bild angewandt werden! Für den Bilateralfilter können Sie sich die beiden Standardabweichungen frei wählen (z.B. 50 und 1.5)! Für alle anderen Filter nutzen Sie die übergebene Filtergröße zur Steuerung der Stärke der Filterung! Lassen Sie sich die unterschiedlichen Filterergebnisse auf dem Bildschirm ausgeben!

2.2. Funktion zum Zeichnen des Histogramms

Eingabeparameter: Eingabebild (Referenz auf ein Bild von Typ *cv::Mat*), Histogramm (Referenz auf ein Objekt von Typ *cv::Mat*), Bild mit dem Histogramm (Referenz auf ein Bild von Typ *cv::Mat*), Breite des Histogramms, Höhe des Histogrammbildes

In dieser Funktion soll das Histogramm berechnet werden. Sie können dies entweder selbst mit einem Bilddurchlauf bestimmen oder die OpenCV-Funktion *cv::calcHist()* benutzen. Anschließend glätten Sie das Histogramm durch Anwendung eines größeren Mittelwertfilters (7x1 oder größer) und normalisieren es in den Bereich zwischen 0 und 1 mit der Funktion *cv::normalize()*. Zum Schluss zeichnen Sie die Werte mit der Funktion *cv::line()* in ein Bild. Sie können sich dazu an dem OpenCV-Tutorium orientieren.

2.3. Funktion zum Segmentieren

Eingabeparameter: Eingabebild (Referenz auf ein Bild von Typ *cv::Mat*)

Zunächst bestimmen Sie durch Ermittlung des maximalen Grauwerts die Größe des Histogramms (Breite des Histogramms). Danach rufen Sie ihre Funktion zum Zeichnen des Histogramms auf (Nutzen Sie den Wert von 400 für die Höhe des Histogrammbildes). Anschließend bestimmen Sie die maximale Krümmung im Histogramm mit dem Algorithmus von Tsai.

$$\psi_t = \frac{1}{R} \sum_{i=1}^R \frac{h(t+i) - h(t-i)}{2i} \quad (1)$$

mit $t=R, R+1, \dots, L-R$. Für R können Sie 25 wählen. Speichern Sie sich die berechneten Werte in einem *Mat*-Objekt. Aus den einzelnen berechneten Werten ψ können Sie anschließend die Krümmung κ nach der folgenden Formel mit einem weiteren Schleifendurchlauf ermitteln:

$$\kappa_t = \frac{1}{R} \sum_{i=1}^R |\psi_{t+i} - \psi_{t-i}| \quad (2)$$

Schreiben Sie diese Werte gleichfalls in ein *Mat*-Objekt. Zum Ende zeichnen Sie die berechneten Krümmungswerte farbig in Ihr Histogrammbild mit der Funktion *cv::line()* ein. Für die Datei Nr. 20 schreiben Sie die Ergebnisse der drei Histogramme mit eingemalten Krümmungswerten in eine png-Datei.

2.4. Erweiterung der main-Funktion

Erweitern Sie Ihre main-Funktion durch Einbau ihrer selbstgeschriebenen Funktionen in die Schleife über alle Bilder, so dass sich jeweils folgender Ablauf ergibt:

- Einlesen des Bildes
- 3x Aufruf der Funktion zum Filtern für die drei unterschiedlichen Filteralgorithmen (immer ausgehend vom eingelesenen Bild)

- 3x Aufruf der Funktion zum Segmentieren, d.h. für jedes der drei Filterergebnisse wird eine eigene Segmentierung durchgeführt.

3. Testat

Voraussetzung ist ein fehlerfreies, korrekt formatiertes Programm. Sie müssen in der Lage sein, Ihr Programm im Detail zu erklären und ggf. auf Anweisung hin zu modifizieren. Laden Sie Ihren Code und die Bildausgaben der drei Histogrammbilder für die Datei Nr. 20 in den moodle-Kurs hoch.