

Programmierung, Algorithmen, Datenstrukturen 1

a.o. Univ.-Prof. Dr. Martin Welk
Dipl.-Ing. Elias Tappeiner
Institut für Biomedizinische Bildanalyse

UNIT TIROL
WS 2021/22
Mech-BSc/ET-BSc

Hausübung 4

Abgabe: 6. Dezember 2021, 23.55 Uhr in Moodle

Es wird empfohlen, die Übungen in Gruppen (max. 3 Studierende) zu bearbeiten und gemeinsam abzugeben.

Hinweise zur Abgabe

- Gruppenarbeiten werden von **einer/m** Studierenden in Moodle eingereicht; die Namen **aller** Beteiligten sind bei der Abgabe zu vermerken! (Empfehlung: Kommentar im Quelltextfile)
- Laden Sie in Moodle die C-Quellcodedatei hoch, bitte keine Codeblocks-Projektdaten o. Ä.

Hinweise zur Bewertung

In die Bewertung der Bearbeitungen fließen ein

- Syntaktische Korrektheit: Das Programm muss fehlerfrei übersetzbar und ausführbar sein.
- Funktionalität: Das Programm muss den vorgegebenen Algorithmus/die vorgegebene Berechnung korrekt ausführen.
- Programmstruktur: Das Programm muss sinnvoll gegliedert (Funktionen!), übersichtlich formatiert (Einrückungen!) und verständlich kommentiert sein.

→

Aufgabe 4.1

Erstellen Sie basierend auf dem Programm `1405-images.c` ein Programm, das zusätzlich zur Invertierung von Bildern folgende Operationen (nach Auswahl durch Konsoleneingabe) ermöglicht:

Binarisierung mit Schwellwert t nach Konsoleneingabe

Hinweis: Dies bedeutet, dass im Ausgabebild alle Pixel, deren ursprünglicher Grauwert $\geq t$ war, auf 255 gesetzt werden und alle anderen Pixel auf 0 gesetzt werden.

Einfacher Gradientenoperator: In jedem Pixel (i, j) mit Ausnahme der ersten und letzten Zeile und Spalte des Bildes wird die x -Ableitung f_x als $\frac{1}{2}(f_{i+1,j} - f_{i-1,j})$ angenähert, die y -Ableitung f_y als $\frac{1}{2}(f_{i,j+1} - f_{i,j-1})$. Im Ausgabebild erhält dieses Pixel dann den Wert $\sqrt{f_x^2 + f_y^2}$. Wenn dieser Wert 255 übersteigt, sollte er durch 255 ersetzt werden. *Hinweise:* 1. Den Pixeln der Randzeilen und -spalten im Ausgabebild darf der Einfachheit halber der Wert 0 zugewiesen werden. – 2. Achten Sie darauf, mit den Ausgabedaten nicht die Eingabedaten zu überschreiben (das heißt, Sie müssen für die Berechnung ein neues Bild anlegen!)

Aufgabe 4.2

Schreiben Sie ein Programm, das eine Textdatei einliest, in der in jeder Zeile ein Wort steht, und diese Wörter anschließend mittels Bubblesort (vgl. Block 3) sortiert. Die Namen der zu sortierenden Textdatei sowie der Ausgabedatei sollen als Usereingaben abgefragt werden.

Hinweise: Die Anzahl der Wörter ergibt sich wieder aus der Textdatei. Sie dürfen der Einfachheit halber annehmen, dass jedes Wort als Zeichenkette in ein `char`-Array der Größe 60 passt; damit können Sie analog zu `1405-images.c` eine zweidimensionale Arraystruktur mit `malloc` konstruieren, die dann an die Bubblesort-Funktion übergeben wird.

In der Bubblesort-Funktion müssen nun anstelle der Vergleiche und Wertzuweisungen für Zahlen (wie in `1307-bb1sort.c`) die entsprechenden String-Funktionen benutzt werden!