

Bildverarbeitung SoSe 2020

Übung 1: Filter

Prof. Dr. Gefei Zhang

8. April 2020

Die in dieser Übung zu erstellende Applikation demonstriert die Funktionsweise verschiedener Filter. Das Eingangsbild wird zunächst durch ein Rauschen gestört. Anschließend wird versucht, die Störungen durch geeignete Filter zu entfernen.

1 Abgabe und Besprechung

- Diese Übung kann als Einzelabgabe oder in 2er-Gruppen bearbeitet werden. Intensive „Zusammenarbeit“ mit weiteren Personen ist nicht gestattet.
- Abgabe der Lösung im moodle.
- Die Übungsaufgaben sind als komplette Projekte für Eclipse abzugeben
 - Bitte den Projektnamen bzw. den Namen des Eclipse-Projektordners wie folgt wählen: `BV<Nummer>_<Nachname(n)>`. Dabei ist `<Nummer>` die Nummer der Übungsaufgabe. Halten Sie sich bitte genau an diese Konvention. Das erleichtert das Auffinden Ihrer Abgabe in einer Liste mit vielen Abgaben pro Übungsaufgabe. Verwenden Sie einen Unterstrich „-“ und kein Blank oder Minuszeichen.
 - * Beispiel Einzelabgabe: `BV1_Schmidt`
 - * Beispiel Gruppenabgabe: `BV1_Schmidt_Schulz`
 - Erstellen Sie wie folgt eine `zip`-Datei und laden Sie sie hoch.
 - * Ihre `zip`-Datei sollte den Projektordner inklusive der versteckten Dateien `.project` und `.classpath`, den Testbildern und allen Quelldateien (`src/.java`) enthalten.
 - * Außerdem soll die `zip`-Datei jeweils eine Erklärung von den beiden BearbeiterInnen enthalten, dass er/sie die Übung alleine oder nur mit dem/der jeweiligen PartnerIn (bitte Namen angeben) bearbeitet und keine unerlaubte Hilfe in Anspruch genommen hat. Die Erklärung ist von jedem/jeder BearbeiterIn eigenhändig zu

verfassen, mit Datum zu versehen, und zu unterschreiben. Scannen Sie die Erklärungen bitte ein oder photographieren Sie diese, und packen Sie die Scans bzw. die Photos in die **zip**-Datei mit ein. Wenn nicht von jedem/jeder BearbeiterIn diese Erklärung vorliegt, wird die Abgabe abgelehnt.

- Abgabefrist: Montag, den 20.04.2020, um 23:59
- Die Besprechung Ihrer Abgabe findet am Donnerstag, den 23.04.2020 statt.

2 Vorbereitungen

- Laden Sie sich das Archiv **BV1.Vorgabe.zip** herunter und entpacken Sie es in dem Ordner, den Sie als Workspace von Eclipse verwenden.
- Die Vorgaben dieses Kurses funktionieren mit Java 8 bis Java 10 (in diesen Versionen ist JavaFX standardmäßig integriert) und ab Java 11 in speziellen Distributionen, die JavaFX enthalten.
- Benennen Sie den entpackten Ordner **BV1.Vorgabe** in **BV1_<Ihr(e) Nachname(n)>** um. Beispiele: **BV1_Schmidt** oder **BV1_Schmidt_Schulz**
- Importieren Sie das Projekt in Eclipse: File → Import ... → General → Existing Projects into Workspace ... Für das root directory wählen Sie den Ordner des Workspaces.
- Wenn Sie das Programm starten, sehen Sie ein GUI mit einigen Schaltflächen und drei Bildbereichen. Im linken und mittleren Bild ist bereits **lena_klein.png** geladen, das rechte Bild ist leer.
- Verschaffen Sie sich einen groben Überblick darüber, wie diese Applikation programmiert ist. Lokalisieren Sie die mit **TODO** gekennzeichneten Stellen in den Klassen **Main**, **RasterImage** und **FilterAppController**, an denen Sie für die folgende Aufgabenstellung Programmcode erstellen müssen.
- Tragen Sie in der mit **TODO** gekennzeichneten Stelle in der Klasse **Main** Ihren Namen ein.

3 Aufgabenstellung

Erstellen Sie eine JavaFX-Anwendung mit folgenden Eigenschaften: (Wie eine Lösung aussehen könnte, sehen Sie weiter unten)

- Automatisches Öffnen des Bildes **lena_klein.png** beim Programmstart (bereits vorgegeben).
- Manuelles Öffnen anderer Bilder durch einen entsprechenden Button (bereits vorgegeben).

- Automatisches Konvertieren der geöffneten Bilder in Graustufenbilder
 - Implementieren Sie `convertToGray()` in der Klasse `RasterImage`.
- Stören des geladenen Bildes durch Rauschen:
 - Implementieren Sie `addNoise()` in der Klasse `RasterImage`.
 - * Mit dem ersten Slider wird der prozentuale Anteil der zu verändernden Pixel festgelegt. (Bereich: 0% bis 60%)
 - * Für die daraus resultierende Anzahl von Pixel wird jeweils eine zufällige Pixelposition bestimmt. Verwenden Sie dazu `java.util.Random`.
 - Mit einer weiteren Zufallszahl bestimmen Sie, ob der Pixel dunkler oder heller werden soll.
 - Der zweite Slider bestimmt, wieviel Sie der Graustufe des Pixels dazugaddieren bzw. abziehen müssen.
- Filtern des gestörten Bildes und Anzeige des Resultats:
 - Implementieren Sie einen Minimumsfilter, einen Maximumsfilter und einen Medianfilter, die jeweils das in `Filter.java` vorgegebene Interface `Filter` implementieren. Ersetzen Sie in der Methode `processImages()` der Klasse `FilterAppController` die DummyFilter durch Ihre eigenen Filter.
 - * Die Größe des Kernels wird durch einen Slider im Bereich von 1 x 1 bis 9 x 9 eingestellt.
 - * Wählen Sie für die Randbehandlung einen Typen, von dem Sie annehmen, dass er für diese Filter gut funktioniert. Siehe Folie 14 von `01-Bildmanipulationen-mit-Filtern.pdf`.
 - Versuchen Sie bei der Implementierung der drei Filter so wenig wie möglich Quellcode zu duplizieren.

4 Optionale Erweiterung

Ermöglichen Sie die Verarbeitung von Farbbildern.

- Fügen Sie dem GUI eine Checkbox „Grayscale“ hinzu, mit der bestimmt wird, ob das Bild in Graustufen konvertiert wird oder nicht.
- Implementieren Sie sowohl das Rauschen als auch die Filter so, dass sie auf den drei Farbebenen R, G und B unabhängig arbeiten.

5 So könnte Ihre Lösung aussehen

Die folgenden drei Abbildungen zeigen Screenshots für verschiedene Einstellungen einer möglichen Lösung. Die jeweiligen Einstellungen können Sie im unteren Bereich der Screenshots ablesen.



