

W.6.24 Medianfilter

Schwierigkeit



Zeitaufwand



Kreativität



Themen

Mit dieser Aufgabe wollen wir Folgendes trainieren:

- Arrays
- Zweidimensionale Arrays

Beschreibung

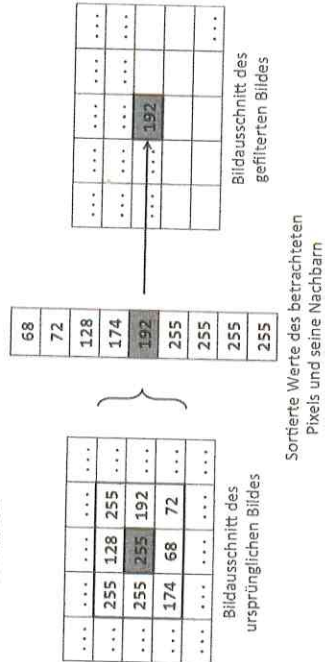
In der Bildverarbeitung werden Medianfilter häufig zur Entfernung von weißen Störpixeln verwendet.

Beim Medianfilter werden die Grauwerte der Pixel in einer definierten Umgebung eines Pixels aufgesammelt und der Größe nach sortiert. Nun wählen wir den mittleren Grauwert (Median) dieser sortierten Liste; dieser ersetzt den Grauwert des aktuellen Pixels.

Ein Bild wird in einem zweidimensionalen int-Array gespeichert, mit Grauwerten von 0 bis 255 für jedes Pixel. Der Grauwert 0 steht für Schwarz, der Grauwert 255 steht für Weiß, die Grauwerte von 1 bis 254 stehen für die heller werdenden Grautöne dazwischen. Ein Medianfilter berechnet aus einem ursprünglichen Bild ein gefiltertes Bild gleicher Größe. Dazu werden für jedes Pixel aus dem ursprünglichen Bild

- sein Grauwert und die acht Grauwerte seiner acht Nachbar-Pixel aufsteigend sortiert in einer Reihe aufgeschrieben
- und der mittlere, also fünfte Wert von neun, als Grauwert des gleichen Pixels im gefilterten Bild genommen.

Das nachfolgende Beispiel zeigt schematisch, wie der Medianfilter aus dem Grauwert 255 eines Pixels im ursprünglichen Bild den Grauwert 192 für das gleiche Pixel im gefilterten Bild berechnet.



Den Grauwert eines Pixels in einem RGB-Bild können wir nach dieser Formel berechnen:

$$\text{Grau} = 0.299 * \text{Rot} + 0.581 * \text{Grün} + 0.114 * \text{Blau}$$

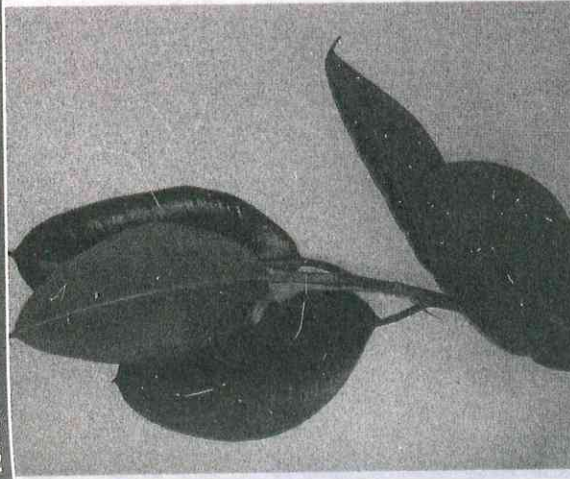
Aufgabenstellung

Schreibe eine Funktion mit dem Namen `medianFilter`, die den beschriebenen Bildfilter implementiert. Nutze dabei die Funktionen von Processing, um ein RGB-Bild einzulesen und auf diesen den Medianfilter anzuwenden.

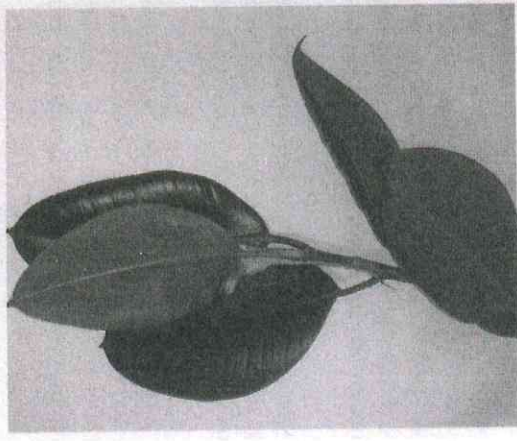
Die Farbwerte (0-255) für Rot, Grün und Blau können mit den Funktionen `red()`, `green()` und `blue()` aus den in Processing verwendeten `Color`-Elementen (RGB-Wert in einem int-Wert kombiniert) ausgelesen werden. Zur Sortierung eines int-Arrays steht die Processing-Funktion `sort()` zur Verfügung.

Testfälle

Vorher



Nachher



Algorithmische Tipps

Wenn du stockst und nicht weiter weißt, dann versuch mal Folgendes:

- Nutze zwei Arrays der Größe 9, in denen du die Pixelwerte des 9x9-Bildbereichs speichern kannst. Ein Array enthält die RGB-Farbwerte, der andere die konvertierten Grauwerte.
- Nach Generierung des Ausgabebildes (`Datentyp PImage`), gehst du jeden Pixelwert einzeln durch. Achte dabei auf die Ränder. Für jeden Pixelwert bildest du jetzt einen 3x3-Bildausschnitt, deren Werte du in das erste Array schreibst. Anschließend berechnest du die Grauwerte aus dem ersten Array und schreibst die Ergebnisse in das zweite Array.
- Nach Sortierung des zweiten Arrays nimmst du den mittleren Wert und schreibst ihn an die selbe Position im Ausgabe-Array. Hierfür müsst du ein neues `Color`-Element erzeugen, da Bilder in Processing immer aus RGB-Werten bestehen.