

Einsendeaufgabe 2 im Modul:

M07 Bildverarbeitung

Semester: WS 24/25

Name: Stefan Berger

Korrektoren: Frau Prof. Dr. Astrid Haibel,
Herr Prof. Dr. Andreas Tewes

Inhaltsverzeichnis

Überblick.....	3
Quelltext	3
Klasse HistogramMatchingPlugin_.....	3
void main(String[] args)	3
void loadReferenceImage()	3
int[] getReferenceCumulativeHistogram()	4
boolean matchHistograms()	4
double[] histogramProbabilityDistribution(int[], boolean)	4
Klasse HistogramMatchingGUI	4
Konstruktor	4
void update()	5
void actionPerformed(ActionEvent).....	5
Klasse HistogramPanel	5
Konstruktoren.....	5
void paintComponent(Graphics)	5
int[] normalize(int[] histogram)	5
Kompilierskript und plugins.config	5
Tests.....	6

Überblick

Das Entwicklungsprojekt zur Lösung der ESA 2 ist als ZIP-Datei Bestandteil des Uploads und liegt außerdem im GitHub-Repository <https://github.com/berger-devel/M07-ESA2/settings> vor. Aufbauend auf den zwei zur Verfügung gestellten Java-Dateien und zwei Beispielbildern kommen insgesamt vier Java-Dateien, ein Kompilierskript und sieben Bilddateien im PNG-Format zum Einsatz. Das ImageJ-Plugin kann sowohl in ImageJ als auch im Debugger der IDE (IntelliJ Idea) ausgeführt werden.

Im Verzeichnis „sberger“ befinden sich die Quelltextdateien, die weiter unten beschrieben werden. Das Verzeichnis „compiled“ enthält den Bytecode und die JAR-Datei, die als ImageJ-Plugin verwendet wird.

Die Bilddateien im JPEG-, PNG- und TIF-Format sind vorgegebene Beispielbilder und eigene Anfertigungen. Einzige Ausnahme ist die Datei „brain.jpeg“, die unter der modifizierten (<https://radiopaedia.org/terms>) Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported licence (<https://creativecommons.org/licenses/by-nc-sa/3.0/>) verwendet wird. Die Bildquelle ist <https://radiopaedia.org/images/466634>. Die Datei „brain.png“ wurde daraus konvertiert.

Die Datei il.jar ist dem ZIP-Archiv entnommen, das auf <https://imagej.net/ij/download.html> (Platform Independent) heruntergeladen wurde.

Das Verzeichnis „.git“ wurde durch das VCS generiert. Die Verzeichnisse „.idea“ und „out“ und die Datei „ESA2.iml“ wurden durch die IDE generiert. Die generierten Verzeichnisse und Dateien werden hier nicht weiter betrachtet.

Die Datei BV-ESA2.docx und BV-ESA2.pdf enthalten jeweils dieses Dokument.

Um das Projekt im Debugger zu starten, muss nur die Datei „il.jar“ als Bibliothek hinzugefügt werden. In IntelliJ kann das in den Module Settings bewirkt werden. Alle Eingabebilder müssen als 8-Bit-Graustufenbilder vorliegen und gegebenenfalls vorher konvertiert werden.

Quelltext

Klasse HistogramMatchingPlugin_

```
void main(String[] args)
```

Für die Ausführung im Debugger ohne Installation in ImageJ öffnet die Main-Methode eine Bilddatei aus einem hartkodierten Dateipfad und initialisiert das Plugin.

```
void loadReferenceImage()
```

Diese Methode öffnet den Dateiauswahldialog für das Referenzbild. Sie lädt die gewählte Datei und überprüft das Dateiformat. Stimmt die Bittiefe des Bildes mit der

Vorgabe überein, wird das entsprechende Flag im Model gesetzt, das Bild in einem Fenster dargestellt und der Plugin-Dialog aktualisiert.

`int[] getReferenceCumulativeHistogram()`

Aus dem geladenen Referenzbild erstellt diese Methode das kumulative Histogramm und gibt es zurück.

`boolean matchHistograms()`

Das Matching der Histogramme findet in dieser Methode statt. Es werden die Wahrscheinlichkeitsverteilungen des Histogramms des Ursprungsbildes und des kumulativen Histogramms des Referenzbildes erstellt. Anschließend wird zunächst aus den Grauwerten des Referenzbildes eine Lookup-Table für die Grauwerte des Ursprungsbildes erstellt. Aus dieser Lookup-Table werden schließlich die Grauwerte für das Ursprungsbild ausgelesen und neu zugewiesen.

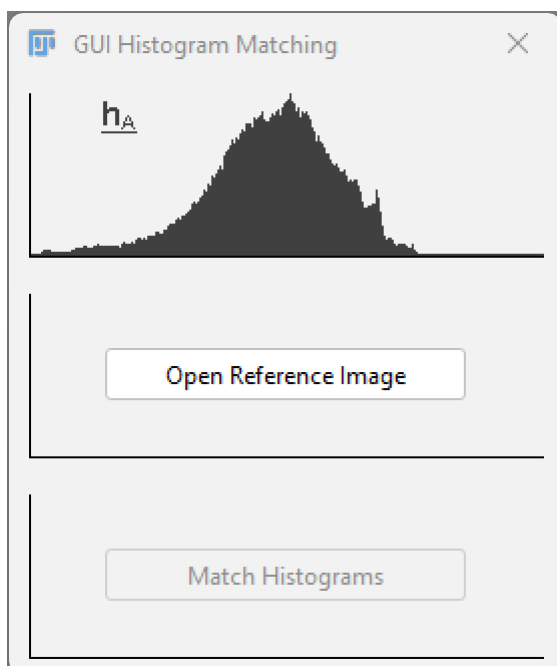
`double[] histogramProbabilityDistribution(int[], boolean)`

Diese Methode erstellt die Wahrscheinlichkeitsverteilung des Histogramms im ersten Parameter. Der zweite Parameter gibt an, ob das Histogramm kumulativ ist oder nicht.

Klasse HistogramMatchingGUI

Konstruktor

Der Konstruktor initialisiert das Dialogfenster des Plugins. Der Dialog erhält drei Histogramm-Panels. Im ersten Panel wird immer das Histogramm des Ursprungsbildes dargestellt. Das zweite Panel enthält initial einen Button, der den Dateiauswahldialog für das Referenzbild öffnet. Das dritte Panel enthält initial den Button zum Starten des Histogramm-Matchings, der vor Auswahl eines Referenzbildes noch deaktiviert ist.



`void update()`

Wenn ein Referenzbild ausgewählt wurde, wird diese Methode aufgerufen. Erfüllt das Referenzbild die Voraussetzungen, wird der Button im zweiten Panel des Dialogs durch das Histogramm des Referenzbildes ersetzt, und der Button zum Starten des Histogramm-Matchings wird aktiviert.

`void actionPerformed(ActionEvent)`

Diese Methode ist der Click-Handler für die Buttons. Sie bewirkt bei Klick auf den Button „Open Reference Image“ das Öffnen des Referenzbildes und zeigt eine Fehlermeldung, wenn das gewählte Bild nicht geeignet ist. Bei Klick auf „Match Histograms“ wird das Histogramm-Matching gestartet und der Button durch das resultierende Histogramm ersetzt.

Klasse HistogramPanel

Konstruktoren

Wie oben beschrieben, enthält ein Panel entweder einen Button oder ein Histogramm. Die Klasse HistogramPanel besitzt daher für jede dieser zwei Möglichkeiten einen Konstruktor, dem entsprechend entweder ein Button oder ein Histogramm übergeben wird. Für ein Histogramm wird zusätzlich eine Beschriftung (h_A , H_R oder $h_{A'}$) festgelegt.

`void paintComponent(Graphics)`

Diese Methode ist für das Zeichnen der Histogramme zuständig. Es werden die Achsen des Histogramms und gegebenenfalls das Histogramm selbst mit seiner Beschriftung in das Fenster gezeichnet.

`int[] normalize(int[])`

Damit auch Histogramme von Bildern mit geringer Gesamtintensität gut erkennbar sind, werden diese Histogramme zur besseren Darstellung an die Höhe des Histogramm-Panels angepasst (skaliert). Diese Methode gibt das angepasste Histogramm zurück.

Kompilerskript und plugins.config

Die Datei compile.bat enthält einige Windows-Kommandozeilen, um den Java-Quelltext zu kompilieren, den Bytecode zusammen mit einem Manifest und der Datei plugins.config in einer JAR-Datei zu verpacken und die JAR-Datei in das Plugins-Verzeichnis von ImageJ zu kopieren.

Die Datei plugins.config enthält Angaben zur Hauptklasse, zum Namen und zum Menüeintrag des Plugins.

Tests

Die folgende Tabelle zeigt die Ursprungsbilder, die jeweiligen Referenzbilder für die Histogrammanpassung und die Ergebnisbilder, die in den Tests verwendet wurden beziehungsweise entstanden sind.

Ursprungsbild	Referenzbild	Ergebnis
		
		
		
		

