Institut für Echtzeitsysteme
und Softwaretechnik
Prof. Dr. Derk Rembold

Hochschule
Albstadt-Sigmaringen
Albstadt-Sigmaringen University

2021

Praktikum Bildverarbeitung

Aufgabenblatt 3

Bit-Operationen mit Bildern und Template-Matching

Anforderungen:

- Die Aufgabe wird in Python programmiert.
- Die Aufgabe wird von jedem Studierenden einzeln erstellt!
- Der Studierende kommt zur Abnahme auf den Dozenten zu. Die Abnahme erfolgt für jeden Studierenden einzeln. Die Kenntnis des Quellcodes wird erwartet.
- Programmcode wird auf Ilias hochgeladen. Die Lokation wird im Praktikum bekanntgegeben. Das File hat folgendes Format:
    - <Name>_<Vorname>_<Matrikelnummer>_Aufgabe_3.py

- **Die Frist für die Abnahme und das Hochladen der Files wird im Praktikum bekanntgegeben.**
- **Die hochgeladenen Files werden nach der Frist nochmals kontrolliert. Erst nach dieser Kontrolle gilt die Aufgabe als vollständig bestanden.**

Einleitung

In Aufgabe 2 hat der Studierende mindestens zehn Bild mit Wally-Köpfen erzeugt, die eine vierte Schicht haben, und zwar den Alpha-Kanal. Der Alpha-Kanal hat Informationen über Pixel, wie sie auf einem Bild darstellt werden sollen. So kann nur ein Pixel innerhalb einer Kontur sichtbar gemacht werden, und die Pixel außerhalb der Kontur sind transparent. Außerhalb der Kontur können z.B. die Pixels Hintergrundbildes dargestellt werden.
Dafür sind Bit-Operationen notwendig, z.B. *bitwise_and*, *bitwise_or* etc [1]. Durch die Anwendung einer Reihe von Bit-Operationen können Bilder mit Alpha-Kanälen nahtlos in Hintergrundbilder eingefügt werden.

Es gibt in der Bildverarbeitung die Möglichkeit, die Ähnlichkeit von zwei Bildern zu ermitteln. Methoden sind die Berechnung des Mean-Square, oder der Cross-Entropie. Bei der Mean-Square-Methode z.B. wird die Pixel Werte von beiden Bilder subtrahiert und quadriert. Dann werden alle Ergebnisse zusammensummiert. Das Ergebnis ist der Gesamtfehler, welcher eine Indikation für die Ähnlichkeit zweier Bilder sind, siehe auch [2].

Aufgabe

Der Studierende soll ein frei verfügbares Wally-Rätsel-Bild, unter Berücksichtigung der Urheberrechte, in eine Applikation einladen. Dann soll der Benutzer mit der Maus in das Bild gehen und beim Mauszeiger soll ein Wally-Kopf-Bild dargestellt werden. Hier ist zu beachten, dass die Information aus dem Alpha-Kanal verwendet werden. Und zwar sollen nur die Pixel innerhalb der Kontur dargestellt werden und außerhalb der Kontur die Pixel des Hintergrundbildes. Dies soll mit Bit-Operationen erreicht werden.
Stets soll dabei die Ähnlichkeit zwischen dem Wally-Kopf und dem dahinterliegenden Ausschnitt des Wally-Rätsel-Bilds berechnet werden. Der berechnete Fehler soll im Bild oben links, z.B. mit der OpenCV-Funktion *putText*, dargestellt werden.

Die User Stories sind in der folgenden Tabelle aufgelistet:

Als	**will ich**	**damit**
Studierender	selbstständig alle erforderlichen Libraries und Funktionalitäten studieren	ich die Anwendung der Aufgabe programmieren kann.
Studierender	mit der programmierten Anwendung ein Wally-Rätsel-Bild laden und darstellen	dem Benutzer beim Mauszeiger ein Wally-Kopf erscheint.
Studierender	mit der rechten Maustaste drücken	der Wally-Kopf ausgetauscht wird. Die Wally-Kopf-Bilder wurden bereits in Aufgabe 2 erzeugt.
Studierender	mit der Maus in das Rätsel-Bild umher bewegen	der Template-Matching Algorithmus (z.B. Mean Square) den Fehler zwischen Wally-Kopf-Bild und Hintergrundbild berechnet.
Studierender	den Fehler oben links am Wally-Rätsel-Bild sehen	der Benutzer die Ähnlichkeit zwischen Mauszeiger-Bild und Hintergrundbild erkennen kann.
Studierender	mit der Maus mit 10 Wally-Köpfen auf dem tatsächlichen Wally vom Rätselbild zeigen	der Studierende einen Eindruck über den berechneten Fehler mit den einzelnen Wally-Kopf-Bilder erhält.

Tabelle: User Stories

Links

[1]: https://docs.opencv.org/master/d0/d86/tutorial_py_image_arithmetics.html
[2]: https://docs.opencv.org/master/d4/dc6/tutorial_py_template_matching.html