Institut für Echtzeitsysteme
und Softwaretechnik
Prof. Dr. Derk Rembold

Hochschule
Albstadt-Sigmaringen
Albstadt-Sigmaringen University

2022

Praktikum Bildverarbeitung

Aufgabenblatt 3

Bit-Operationen mit Bildern

Anforderungen:

- Die Aufgabe wird in Python programmiert.
- Die Aufgabe wird von jedem Teilnehmer einzeln erstellt!
- Der Teilnehmer kommt zur Abnahme auf den Dozenten zu. Die Abnahme erfolgt für jeden Teilnehmer 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 Teilnehmer mindestens ein Bild mit einem beliebigen Gesicht mit einer Transparenzschicht erzeugt. Diese Transparenzschicht nennt man Alphakanal. Der Alphakanal hat Informationen über Pixel, wie sie auf dem Bildschirm darstellt werden sollen. So wurde die Aufgabe gestellt, dass Pixel innerhalb einer Kontur sichtbar sind, und die Pixel außerhalb der Kontur transparent sind. Außerhalb der Kontur können somit die Pixel eines 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 Original-Bilder mit Hilfe der Alphakanälen in Hintergrundbilder nahtlos eingefügt werden.

Aufgabe

Eine Applikation soll ein zwei Gesichtsbilder einladen; das erste Bild mit Gesicht aus Aufgabe 2 mit Alphakanal und ein zweites Bild mit Gesicht ist beliebig. Beim zweiten Bild kann auf ein Alphakanal verzichtet werden. Bei beiden Bildern soll die Applikation die Augenpositionen der Bilder mit Hilfe der *Haar*-Bibliothek [2] ermitteln. Dabei soll jeweils der Winkel der Geraden zwischen

den Augen ermittelt werden, siehe auch Aufgabe 1. Zusätzlich ist der Abstand der Augen zu ermitteln. Das erste Bild soll mit Hilfe der Winkel rotiert und mit Hilfe des Abstands mit der OpenCV *resize*-Methode [3] skaliert werden. Es soll dabei der Augenabstand beider Bilder gleich sein. Die Anwendung überlagert nun beide Bilder unter Verwendung des Alphakanals des ersten Bilds. Im Transparenzbereich außerhalb des Gesichts des ersten Bildes sollen die Pixel des zweiten Bildes erscheinen.

Die User Stories sind in der folgenden Tabelle aufgelistet:

Als	**will ich**	**damit**
Teilnehmer	selbstständig alle erforderlichen Libraries und Funktionalitäten studieren,	ich die Anwendung der Aufgabe programmieren kann.
Teilnehmer	mit der programmierten Anwendung zwei Bilder mit unterschiedlichen Gesichter laden,	beide Bilder überlagert werden können.
Anwendung	den Augenabstand bei beiden Bildern mit der *Haar*-Bibliothek ermitteln,	das erste Bild skaliert werden kann, sodass die Augen beim Überlagern übereinander liegen.
Anwendung	die Winkel bei beiden Bildern die zwischen den Geraden durch die Augen und der x-Achse gehen, ermitteln,	das erste Bild rotiert werden kann, sodass die Augen beim Überlagern übereinander liegen.
Anwendung	beide Bilder überlagern,	unter Verwendung des Alphakanals keine abrupten Ränder entstehen.
Teilnehmer	eine Funktion programmieren, die zwei Bilder als Parameter hat,	ein überlagertes Bild unter Berücksichtigung der User Stories zurückliefert.

Tabelle: User Stories

Links

[1]: https://docs.opencv.org/master/d0/d86/tutorial_py_image_arithmetics.html
[2]: https://github.com/opencv/opencv/tree/master/data/haarcascades
[3]: https://docs.opencv.org/3.4/da/d54/group__imgproc__transform.html