

2019

Praktikum Bildverarbeitung

Aufgabenblatt 1

Kernfunktionalitäten in der Bildverarbeitung

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_1.py
- **Die Frist für die Abnahme und das Hochladen der Files wird im Praktikum bekanntgegeben.**

Einleitung

Zur Veranschaulichung von Funktionalitäten in der Bildverarbeitung soll die Bibliothek OpenCV verwendet werden. Eine Beschreibung von OpenCV findet sich in [1]. Ein Tutorial gibt es in [2]. In der Bildverarbeitung wird häufig Ausschnitte eines vorhandenen Bildes benötigt. Diese Ausschnitte müssen dabei in einem bestimmten Format und Größe vorliegen. Innerhalb von Python/OpenCV verwendet man dafür n-dimensionale Arrays aus der Bibliothek numpy (numpy.ndarray). Unter Verwendung von Formate wie z.B. jpeg, png oder gif soll ein Bild mit cv2.imshow() aus der Python/OpenCV Bibliothek angezeigt werden.

Eine weitere wichtige Library in der Bildverarbeitung ist die Bibliothek imutils [3]. Unter anderem stellt die Bibliothek imutils Funktionen bereit, Bilder zu rotieren. Das Problem bei der Rotation von rechteckigen Bildern ist, dass die Ränder auf nicht ausgefüllt werden und deswegen als schwarze Flächen dargestellt werden. Eine Möglichkeit dem entgegenzuwirken ist das Bild mit dem OpenCV Kommando flip [4] zu drehen und ein neues Bild aus dem Originalbild zu erzeugen. Und zwar werden acht gedrehte Bilder an allen vier Kanten und vier Ecken des Originalbilds eingefügt. Es wird dann nur ein Ausschnitt des vergrößerten Bildes verdreht.

Aufgabe

Es soll eine Anwendung programmiert werden, die ein Bild, gespeichert auf der Festplatte, auslesen und dann mit Hilfe von Tastenkommandos weiterverarbeitet. Nach dem Aufruf soll ein neues Bild erzeugt werden, das an allen vier Ecken und Kanten gedrehte Bilder enthält. Dann soll die Anwendung ein Fadenkreuz und ein Rechteck auf das neue Bild zeichnen. Der Benutzer soll mit Hilfe von Tastenkommandos die Position des Fadenkreuzes und des Rechtecks verändern und die Größe des Rechtecks einstellen können. Ein weiteres Kommando speichert den Inhalt des Rechtecks zunächst in ein Python-Image Objekt, um danach auf eine bestimmte Größe normiert zu werden. Danach wird das Objekt auf die Festplatte gespeichert. Die User Stories sind in der folgenden Tabelle aufgelistet:

Als	will ich	damit
Studierender	selbstständig alle erforderlichen Libraries und Funktionalitäten studieren	er seine Expertise mit Bildverarbeitung erweitern kann
Anwendung	ein Bild an den Ecken und Kanten erweitern	der Benutzer bei der Rotation keine Schwarze Flächen sieht
Anwendung	ein erweitertes Bild darstellen	der Benutzer ein Ausschnitt herausschneiden kann
Anwendung	ein Rechteck und ein Fadenkreuz darstellen	der Benutzer die Position des Rechtecks und des Fadenkreuzes verändern kann
Benutzer	die Pfeiltasten bedienen	die Position des Rechtecks und des Fadenkreuzes verändert wird
Benutzer	die Tasten „+“ und „-“ bedienen	die Größe des Rechtecks verändert wird
Benutzer	die Tasten „r“ und „l“ bedienen	das Bild rotiert wird. Rechteck und Fadenkreuz wird nicht rotiert.
Benutzer	die Taste „s“ bedienen	der Ausschnitt im Rechteck auf eine Größe 124x124 normiert wird und im png-Format auf die Festplatte gespeichert wird.
Benutzer	die Taste „q“ bedienen	sich die Anwendung schließt

Tabelle: User Stories

Links

- [1]: <https://docs.opencv.org/3.0-beta/index.html>
- [2]: https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_tutorials.html
- [3]: <https://pypi.org/project/imutils/>
- [4]: https://docs.opencv.org/3.4/d2/de8/group__core__array.html