

## Teil 1

Prof. Dr. Elke Hergenröther, Version 14.04.2022

### Motivation

Im Praktikum sollen die in den Vorlesungseinheiten zur analytischen Computer Vision vorgestellten Techniken angewendet und vertieft werden. Um dies praxisnah zu tun, wurde das Szenario der „Smarten Aufspannkontrolle“ in einer Fräsmaschine ausgewählt.

### Hinweise zur Durchführung des Praktikums

Die Praktikumsaufgaben werden während der Praktikumstermine in der Gruppen diskutiert. Jeder von Ihnen sollte Lösungen für alle Aufgaben vorbereitet haben und beantworten können. Bis zum **Termin #1** bitte ich Sie *OpenCV* und eine Entwicklungsumgebung zu installieren, in der Sie mit *C++* oder *Python* programmieren können.

### A. Einarbeiten in OpenCV mit C++ oder Python und erste Ansätze zur Objektvermessung

Computer Vision Anwendungen werden häufig zur Qualitätssicherung beziehungsweise zur Unterstützung von Mitarbeitern bei der Bedienung von Maschinen genutzt. In diesem Zusammenhang ist auch die folgende Aufgabe zu sehen. Geplant ist den Vorgang des Einlegens von Rohlingen in einer Fräsmaschine mit einem Computer Vision System zu begleiten. Nachdem der Mitarbeiter den Aluminium- oder Stahlklotz in die Fräsmaschine gelegt hat, soll das Computer Vision System die Kantenlängen des Rohlings ausmessen. Ist der Klotz zu groß, kann es zur Kollision mit der Spindel, in der das Fräswerkzeug eingespannt ist, kommen. Die Reparatur einer solchen Spindel kann mehrere tausend Euros kosten. Ist der Klotz zu klein, kann die gewünschte Form nicht ausgefräst werden. Ein solche Smarte Aufspannkontrolle wäre also ein wertvolles Hilfsmittel. Ihre Aufgabe ist es eine Computer Vision Anwendung zu schreiben, die einen Teil der Smarten Aufspannkontrolle realisiert und automatisch einen Rohling ausmisst.

Da Sie nicht alle einen Aluminium- oder Stahlrohling zur Hand haben, sollen für das Praktikum Werkzeuge genutzt werden, die man in allen Haushalten findet: Einen Inbus-, einen Maul- und einen Ringschlüssel, alternativ können Sie auch ein Messer, eine Gabel oder ein Löffel nutzen. Wichtig ist nur, dass diese zu messende Stücke aus Metall sind, denn das macht ihre Aufgabe anwendungsnah.

Die Länge und Breite des ausgewählten Objekts im Bild zu vermessen, wird schwieriger, wenn die Materialien spiegelnd sind und die Kameraaufnahmen nicht optimal sind. In den meisten Fräsmaschinen sind die Lichtverhältnisse nicht optimal. Um die Wichtigkeit des Aufnahmeprozesses für die anschließende maschinelle Auswertung des Bildes praktisch nachvollziehen zu können, sollten Sie zunächst selbst Aufnahmen der Werkstücke machen. Achten Sie dabei auf den Hintergrund. Am besten ist ein einfarbiger Hintergrund der einen hohen Kontrast zum Werkstück hat. In den aufgenommenen Bildern soll dann mit Hilfe von OpenCV automatisch die Länge und Breite des auszumessenden Objekts in Pixeln ermittelt und anschließend in Millimeter umgerechnet werden. Diskutieren Sie zunächst in Ihrem Team, wie Sie die Aufgabe umsetzen wollen und erarbeiten Sie gemeinsam ein Konzept. Als Inspiration kann Ihnen dieses Video dienen:

<https://www.youtube.com/watch?v=KEYzUP7-kkU> (OpenCV in Python). Es gibt bestimmt noch viele andere passende Videos. Am Ende des Teils B finden Sie eine Auflistung von Operationen in OpenCV, die hilfreich für Sie sein könnten.

#### **Die Aufgabe A zusammengefasst:**

- Bis zum **Termin #1** sollten Sie OpenCV installiert haben und Ihre Grundkenntnisse in Bildverarbeitung aufgefrischt bzw. sich die nötigen Kenntnisse angeeignet haben. Dabei kann Ihnen der Foliensatz „Wiederholung\_Bildverarbeitung.pdf“ der in Moodle hinterlegt ist, helfen.
- Machen Sie Aufnahmen Ihrer ausgewählten Werkstücke.
- Arbeiten Sie sich in OpenCV ein und entwickeln Sie im Zweierteam ein Konzept zur Lösung der Aufgabe. Diskutieren Sie, welche Schritte durchgeführt werden müssen und ob es Alternativen zu Ihren ersten Ideen gibt. Bedenken Sie auch, dass die in Pixeln gemessenen Kantenlängen in Millimeter umgerechnet werden müssen. Nach Abschluss der Aufgabe muss vom Team eine kurze wissenschaftliche Ausarbeitung (ca. 3-4 Seiten) angefertigt werden. Bereiten Sie sich auf das Schreiben der Ausarbeitung vor, indem Sie Ihr geplantes Vorgehen bereits vor der Umsetzung dokumentieren.

## **B. Abschluss der Implementierungsarbeiten zur Objektvermessung mit OpenCV**

**Bis zum Termin #2** sollten Sie Ihre Implementierung abgeschlossen haben. Ihr Programm soll in verschiedenen, von Ihnen gemachten Aufnahmen die Länge und Breite der Objekte automatisch ermitteln und die gemessenen Kantenlängen in Millimeter ausgeben. Gerne können Sie ein Hilfsobjekt (Kalibrierungsobjekt), dessen Länge und Breite Sie kennen, mit aufnehmen. Auf diese Weise kann eine Umrechnung in Millimeter einfach ermöglicht werden. Dokumentieren Sie den Messfehler, den Ihr Computer Vision System beim automatischen Ausmessen macht.

**Zwei Stunden vor Ende des Termins #2** sollen Sie Ihre Lösung in einem **kurzen Vortrag gefolgt von einer Demonstration Ihres Programms** präsentieren. Der Vortrag soll von einer Folien-Präsentation begleitet werden und ca. 5 Minuten dauern. Nach der Demo ist Zeit für eine Diskussion, in der Sie sich mit Ihren Kommilitoninnen und Kommilitonen austauschen können.

#### **Anfertigen einer wissenschaftlichen Ausarbeitung bis zum **Sonntag, 29.05.2022****

Eine sehr kurze Wiederholung zur Anfertigung einer wissenschaftlichen Arbeit: Kriterium hierfür ist ein systematisches, überlegtes Vorgehen. Das sind im Einzelnen die folgenden Schritte:

- die Präsentation und Analyse der Schwierigkeiten der Aufgabenstellung
- Eine Recherche, welche Lösungsmöglichkeiten es bereits gibt
- die Entwicklung eines Durchführungsplans (also eines Konzeptes)
- die anschließende Realisierung des Konzeptes
- letztendlich die Bewertung, wie gut sich die Entwicklung zur Lösung der Aufgabe eignet

Zur Aufbesserung der Note können sie zwei Zusatzaufgaben lösen:

- Gibt es eine Möglichkeit die Größe eines Objekts ohne bekanntes Vergleichsobjekt auszumessen? Wenn ja, schreiben Sie das Konzept auf.
- Welche Idee haben Sie, um die Auflösung der Aufnahmen zu erhöhen, ohne eine Kamera mit der entsprechend hohen Auflösung zu kaufen?

### **Die nachfolgenden Operationen in OpenCV könnten hilfreich für Sie sein:**

Achten Sie bitte darauf, dass die Versionsnummer richtig eingestellt ist. Umstellen können Sie dies ganz oben auf der jeweiligen Seite.

3.1. Binarisierung bzw. Schwellwertbildung:

[https://docs.opencv.org/3.4/d7/d4d/tutorial\\_py\\_thresholding.html](https://docs.opencv.org/3.4/d7/d4d/tutorial_py_thresholding.html)

3.2. Ebenfalls unter <https://docs.opencv.org/master/> finden Sie Weichzeichner, die den Farb- bzw. Grauwertverlauf glätten: [https://docs.opencv.org/master/d4/d13/tutorial\\_py\\_filtering.html](https://docs.opencv.org/master/d4/d13/tutorial_py_filtering.html)

Alternativ können Sie auch mit morphologische Operatoren das Bild bearbeiten:

[https://docs.opencv.org/master/d9/d61/tutorial\\_py\\_morphological\\_ops.html](https://docs.opencv.org/master/d9/d61/tutorial_py_morphological_ops.html)

3.3. Kantendetektoren zur Hervorhebung der Konturen:

Laplace Operator: [https://docs.opencv.org/master/d5/db5/tutorial\\_laplace\\_operator.html](https://docs.opencv.org/master/d5/db5/tutorial_laplace_operator.html)

Sobel Operator: [https://docs.opencv.org/master/d5/db5/tutorial\\_laplace\\_operator.html](https://docs.opencv.org/master/d5/db5/tutorial_laplace_operator.html)

Canny Edge Detection: [https://docs.opencv.org/3.1.0/da/d22/tutorial\\_py\\_canny.html](https://docs.opencv.org/3.1.0/da/d22/tutorial_py_canny.html)

Hough Line: [https://docs.opencv.org/master/d9/db0/tutorial\\_hough\\_lines.html](https://docs.opencv.org/master/d9/db0/tutorial_hough_lines.html)

Hough Circle: [https://docs.opencv.org/master/d4/d70/tutorial\\_hough\\_circle.html](https://docs.opencv.org/master/d4/d70/tutorial_hough_circle.html)

3.4. Segmentierung und sonstige Funktionen, die Ihnen nützlich sein könnten finden Sie hier:

[https://docs.opencv.org/master/d2/d96/tutorial\\_py\\_table\\_of\\_contents\\_imgproc.html](https://docs.opencv.org/master/d2/d96/tutorial_py_table_of_contents_imgproc.html)