Echtzeitbildverarbeitung — Übungsblatt 1

www.informatik.uni-bremen.de/agebv/de/VeranstaltungEBV13 Abgabe: 8.5.2015

Allgemeine Hinweise

Benennt alle `.sample-cc` Dateien in `.cc` um. Die noch fehlenden Teile des Programms sind im Sourcecode mit `TODO` markiert, dahinter steht die jeweilige Punktzahl für den Teil. Implementiert diese Teile. Kopiert die bearbeiteten Teile des Codes in ein `.pdf` zusammen mit Euren Bearbeitungen der anderen Aufgaben. Dieses `.pdf` wird ausgedruckt, korrigiert und zurückgegeben.

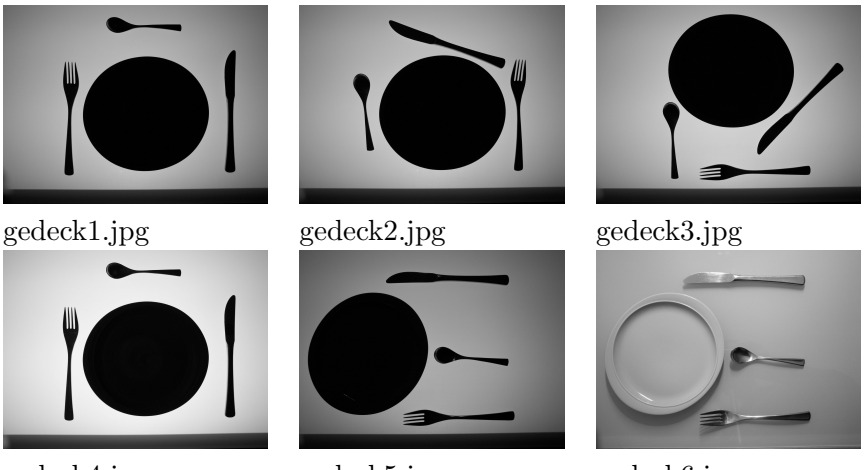
Gebt dann die veränderten Sourcen zusammen mit der `.pdf`-Datei in einer `zip`-Datei bei Stud.IP ab. Die `.zip`-Datei soll den Namen `ebvassignment??-<namen>.zip` haben.

Gebt Namen und Email aller Gruppenmitglieder an. Gebt ausserdem die Zeit (max. aller Gruppenmitglieder) an, die Ihr für den Zettel benötigt habt. Diese Angabe ist nur als Rückmeldung für uns und geht nicht in die Bewertung ein.

WICHTIG: Wir legen sehr viel Wert darauf, dass Routinen wirklich richtig sind und nicht nur funktionieren. Das heißt, eine Routine sollte genau das tun, was in der Spezifikation steht. Insbesondere sollte sie keine undokumentierten Seiteneffekte haben oder Annahmen über das Szenario oder die Werte von Rückgabevariablen machen. Wenn ihr denkt, "einen dreckigen Hack" machen zu müssen, kommentiert ihn, das ist auch okay, d.h. es gibt keinen Punktabzug.

Weitere Hinweise zum Programmieren:

- Benutzt einen Debugger, z.B. Visual Studio oder DDD.
- **Routinen müssen das tun, was in der Dokumentation steht. Wenn Ihr spezifische Annahmen macht müsst Ihr sie dokumentieren.**
- In C++ werden Objekte meist nicht mit `new` angelegt, wenn doch müssen sie mit `delete` freigegeben werden.
- Wenn eine Eigenschaft im Programm gelten muss, schreibt ein `assert` um es zu überprüfen (In vernünftigem Umfang). Das hilft beim Fehler finden.
- Schaut kritisch auf das Ergebnis, manche Fehler sind schwer zu sehen.



gedeck1.jpg gedeck2.jpg gedeck3.jpg

gedeck4.jpg gedeck5.jpg gedeck6.jpg

Abbildung 1: Testdaten für Aufgabe 1. Bilder 1-4 sind im Hintergrund beleuchtet; in Bild 5 ist eine kleine störende Reflexion eines Deckenlichtes zu sehen (funktioniert euer Programm trotzdem?); Bild 6 zeigt eine normale Beleuchtung und erfordert wohl andere Methoden.

Aufgabe 1 Wer hat mein Tellerchen verrückt? (10 Punkte)

Entwickelt ein Programm, das in den angehängten Bildern (Abb. 1) Teller, Messer, Gabel und Löffel erkennt, ihre Position bestimmt und im Bild beschriftet anzeigt. Benutzt das vorgegebene Rahmenprogramm um Bilder einzulesen und das Ergebnis darzustellen. Implementiert nach den Erklärungen der Vorlesung folgende Schritte des "industriellen Ansatzes":

1. Binarisieren des Bildes mit einem manuell eingestellten Schwellwert und gleichzeitiges Umsetzen in Intervalle (RLE). Findet einen Schwellwert, der für alle Bilder passt.
2. Zusammenfassen von Regionen (union-find Algorithmus)
3. Bestimmen von Schwerpunkt und Hauptträgheitsachsen. Eine Routine für Eigenwerte / -vektoren einer 2×2 Matrix ist im Rahmenprogramm enthalten.
4. Klassifikation der Objekte an Hand der Hauptträgheitsmomente bzw. Fläche.

Die Schritte (1)-(3) sind bei den meisten Anwendungen gleich und in der Vorlesung dargestellt. Der letzte Schritt ist "applikationsabhängig", d.h. hier seid Ihr gefragt. Überlegt Euch, wie man die 4 Objekte unterscheiden kann. Ihr könnt ruhig "harte Grenzen" verwenden, erläutert aber welche Vorraussetzungen an die Umgebung, die Aufnahmesituation, das Besteck ihr damit macht.

Nehmt das angehängte Programm `cutlery.sample.cc` als Rahmen für Eure Implementierung und benennt es nach `cutlery.cc` um. Es existieren schon Datenstrukturen und Routinen zum Anzeigen des Ergebnisses. Ziel der Aufgabe ist, die oben genannten Methoden selbst zu implementieren, d.h. Ihr dürft nicht die in OpenCV vorhandenen Funktionen verwenden.

Aufgabe 2 Kekse (4 Punkte)

Eine Backwarenfirma stellt die berühmten Kekse mit den 52 Zähnen her (Abb. 2). Häufig brechen beim Backen Zacken ab oder überlaufender Teig führt zu verunstalteten Keksen und Reklamationen erboster Kunden. Mit Hilfe von Echtzeitbildverarbeitung (viele Kekse) sollen mangelhafte Kekse identifiziert und aussortiert werden. Entwerft einen Lösungansatz und beschreibt die Vorgehensweise (1/2 bis 1 Seite). Geht insbesondere auf Beleuchtung, Position der Kamera, Bildverarbeitungsansatz, benutzte Merkmale und das Kriterium zur Entscheidung auf Grund der Merkmale ein. Schlagt aber auch knapp vor, wie die Einbindung in den Produktionsprozess, nämlich Zuführung der Kekse und Aussortieren der fehlerhafte Kekse erfolgen soll.

Abbildung 2: Die Zähne eines Kekse sollen per Bildverarbeitung kontrolliert werden (Quelle:Wikipedia/Bahlsen).

Aufgabe 3 Teller sind ein weites Feld (2 Bonuspunkte)

In Aufgabe 1 habt Ihr Annahmen über die Umgebung machen müssen um die Objekte zu klassifizieren. Wie könnte man den Klassifizierungsschritt (4) realisieren, wenn Ihr die Testbilder nicht vorher gekannt hättet, d.h. die genaue Gestalt des Geschirrs unbekannt wäre. Eine Idee reicht aber sie sollte auf algorithmischer Ebene beschrieben werden, so dass jemand mit Erfahrung sie implementieren kann. Z.B. "man klassifiziert die Objekte in länglich vs. rund über einen Schwellwert auf das Verhältnis der Hauptträgheitsachsen".