Echtzeitbildverarbeitung — Übungsblatt 5

www.informatik.uni-bremen.de/agebv/de/VeranstaltungEBV15 Abgabe: 10.07.2015

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.

Aufgabe 13 Harte Kante auf der Grafikkarte (10 Punkte)

Wie in der vergangen Vorlesung angesprochen, eignet sich das „Sobeln“ eines Bildes gut dazu parallelisiert zu werden. Dies sollt ihr in dieser Aufgabe mittels CUDA auch umsetzen. Dazu müsst ihr einen eigenen Kernel schreiben und auch den Aufruf des Kernels, samt Vorbereitungen, erstellen. Zusätzlich zur Implementierung auf der GPU braucht Ihr keine weiteren Optimierungen anwenden.

Das Programm wird wie folgt gestartet: `gpuSobel -edges images/ball.001.jpg`

Das Ergebnis wird nicht auf dem Bildschirm angezeigt, sondern in eine Datei gespeichert.

Bei Problemen bitte rechtzeitig bei Florian melden (flomaass@informatik.uni-bremen.de), auch wenn die Frage noch so trivial lauten mag.

Anmerkungen:

- Der Rahmen ist für die Linux-Rechner auf E0 im MZH (z.B. `x01.informatik.uni-bremen.de`) vorgesehen. Die unkompliziertesteMöglichkeit ist, wenn Ihr dort vor Ort arbeitet.

- Alternativ kann auch mittels SSH von zuhause arbeiten. Dazu kopiert man den jeweils geänderten Sourcecode mit `scp` auf den E0 Rechner, kompiliert ihn dort, führt ihn dort aus und holt sich mit `scp` ggf, das Ergebnisbild zurück. (Achtung: `scp` muss erst eingerichtet werden)

- Eine typische Implementierung benötigt, auf einen der E0-Rechner, circa 1,4 ms. (inklusive allozieren und hin und her kopieren). Hier könnte man noch einiges verbessern, was aber über diesen Übungszettel hinaus geht.

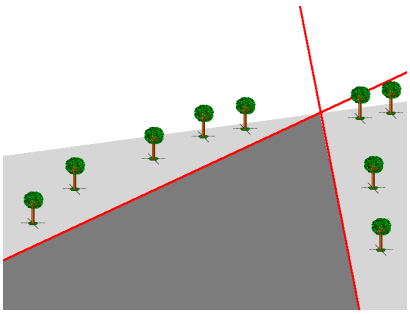

Abbildung 1: Eine kalibrierte Kamera sieht eine Landebahn. Die Bildverarbeitung erkennt die Ränder der Landebahn als Geraden (nicht Strecken!). Welche geometrische Information kann man daraus entnehmen?

Aufgabe 14 Schwarz und Weiss wir stehen an Eurer Seite (4 Punkte)

In der RoboCup small-size league spielen radgetriebene Roboter gegeneinander Fussball. Sowohl die Roboter als auch der Ball werden über eine (oder zwei) Kamera an der Decke verfolgt. Gegenwärtig realisiert jedes Team seine eigene Bildverarbeitung, häufig durch eigene farbige Markierungen auf den Robotern. Bei Turnieren verursacht die Notwendigkeit für jedes Team ein oder zwei eigene Kameras anzubringen und zu kalibrieren erheblichen Aufwand.

Da die Bildverarbeitung nicht im Schwerpunkt der Forschung dieser Liga liegt, gibt es Überlegungen, statt individueller Lösungen einen standisierten Vision Server für alle Teams zur Verfügung zu stellen. In dieser Aufgabe soll eine ungewöhnliche Lösung entworfen werden, die ausserdem die Abhängigkeit der Farberkennung von der Beleuchtung umgeht.

Seien dazu die Roboter mit einem einzelnen weissen Kreis auf schwarzem Grund markiert. Die Kamera erkennt diesen Punkt, sowie den ebenso hellen Ball. Ausserdem erhält der Vision Server Fahrkommandos für die Roboter beider Mannschaften, d.h. ihre momentane Geschwindigkeit in vorwärts / rückwärts Richtung, seitlich und ihre momentane Drehgeschwindigkeit. Diese Information kann natürlich nur einem neutralen Server zur Verfügung gestellt werden.

Entwerft ein Verfahren, wie Roboter und Ball trotz dieser minimalen Bildverarbeitungsinformation verfolgt werden können. Wenn Ihr einen Filter verwendet, erleutert insbesondere Zustandsraum, Dynamikfunktion und Messfunktion. Es reicht, wenn Ihr es inhaltlich aber klar beschreibt, eine Formel ist nicht nötig.

Aufgabe 15 Hinterm Horizont gehts weiter (1 Bonuspunkt)

An einem Flugzeug ist eine intern kalibrierte Kamera angebracht, die die Landebahn beobachtet und als zwei Geraden (nicht Strecken!) erkennt (Abb. 1). Die Geraden liegen in der Bodenebene (X-Y), sind zur Y-Achse parallel und haben bekannten Abstand. Den Horizont selbst erkennt die Kamera nicht, z.B. wegen Bäumen. Welche Information kann man über die Pose von Kamera bzw. Flugzeug daraus gewinnen? Beschreibt genau, die beobachtbaren bzw. nicht beobachtbaren Freiheitsgrade.