

## 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 7 Ballspiele I (10 Punkte)

Die Programmieraufgaben der Übungsblätter 2-6 haben als Gesamtziel Übungsblätter haben als Gesamtziel, die Flugbahn eines geworfenen Balles vorherzusagen. Das heisst, nach drei bis vier Bildern der Bildsequenz (Abb. 1) zeigt das Programm ein Kreuz an der Stelle an, wo der Ball den Boden treffen wird. Mit weiteren eintreffenden Bildern wird die Vorhersage verbessert. Die so berechnete Position könnte z.B. an einen Roboter kommandiert werden, der so den Ball mit einem Netz auffängt oder mit einem Schläger zurückschlägt.

In dieser Aufgabe soll der fliegende Ball im Bild erkannt werden (Programmstart: `ballgame -circles images/ball.??? .jpg`). Ihr benötigt dazu einen funktionierenden SobelFilter (`sobelFastOpenMP`), wenn Ihr nur eine langsame Implementierung habt, ruft die von `sobel` auf. Habt Ihr gar keine, kopiert die Musterlösung hinein.

Implementiert dann nach den Erklärungen der Vorlesung eine Kreis Houghtransformation (`houghCircle.h/.cc`) mit Houghraum  $(x_c, y_c)$ . Sucht hinreichend grosse (Schwellwert) und in einer Umgebung maximale Werte im Houghraum und führt für diese Mittelpunkte eine 1D Houghtransformation ( $r$ ) zur Suche des Radius durch. Implementiert die Funktionen so effizient wie sinnvoll (kritische Teile sind in der Vorlage kommentiert) und verwendet insbesondere Tabellen zur Beschleunigung des Eintragens in den Houghraum. Zum Testen könnt Ihr auch ein Testbild (`ballgame2 -circles images/test.circles.jpg`) mit zwei sehr ausgeprägten Kreisen verwenden.

Ganz eindeutig ist so eine Kreiserkennung nicht, weil in dem unruhigen Hintergrund gerne



Abbildung 1: Aufeinanderfolgende Bilder (Abstand 1/25s) eines geworfenen Balles in einer natürlichen Szene. Dank geht an das Congress Center Bremen für die Drehgenehmigung.

“Phantomkreise” erscheinen. Um dies zu verhindern, könnt Ihr ausnutzen, dass der Ball sich im Bild bewegt. Die Routine `HoughCircle::hough` erhält dazu das vorherige Sobelbild zum Vergleich. Verwendet diese Information um die Zahl von Nicht-Ball-Pixeln die in den Houghraum eingetragen werden zu reduzieren.

Ziel der Aufgabe ist, die oben genannten Methoden selbst zu implementieren, d.h. Ihr dürft nicht die in OpenCV vorhandenen Funktionen verwenden. Wenn Ihr nicht das mitgelieferte Makefile verwendet müsst Ihr den Compilerschalter `MAIN_NR=2` setzen. Mit diesem Mechanismus kann das Hauptprogramm für alle aufeinander aufbauenden Übungszettel verwendet werden.

## Aufgabe 8 Tag des Modellbaus (4 Punkte)

Anlässlich des Tages des Modellbaus möchte der lokale Modellbaclub eine Demonstration mit einem autonom fliegenden Modell Zeppelin zeigen. Im Verlauf des Demos soll der Zeppelin autonom abheben, eine Runde über den Modellflugplatz fliegen und autonom wieder landen. Der Zeppelin schwebt in der Luft und kann durch schwenkbare Propeller längs (vor/zurück), quer (links/rechts) und vertikal (hoch/runter) schweben, sowie sich in der horizontalen Drehen (gieren). Der Modellflugplatz ist eine grosse Wiese auf der sich auch die Besucher befinden. Für die Demonstration kann die Wiese in vernünftigen Umfang mit Markierungen versehen werden.

Erläutert ein Konzept (ca. 1 Seite), wie man das Problem mit Bildverarbeitung lösen kann. Geht auf Art eventueller Markierung, Position der Kamera und welche Merkmale wie im Bild erkannt werden ein. Bedenkt, dass Kameras einen begrenzten Öffnungswinkel (maximal 90-110°) haben und je nach Flughöhe nicht immer alles zu sehen ist. Erläutert besonders, wie das System hinreichend zuverlässig gemacht werden kann.

## Aufgabe 9 Spargel, Bildverarbeitung und soziale Realität (1 Bonuspunkt)

Der Spargelbetrieb, der für die Anwendung “Spargel in Handelsklassen sortieren” in der Übung Modell gestanden hat verfährt zum Sortieren des Spargels wie folgt: Der Hauptteil des Spargels wird von Saisonarbeiterinnen nach Augenmass in Klasse I, II und III sortiert. Die Spargelstangen, bei denen die Arbeiterinnen sich unsicher sind gehen an eine Maschine, die mit Bildverarbeitung sortiert. Könnt Ihr euch diese Vorgehensweise erklären?