

Monatlicher Bericht – Studienarbeit "I believe I can fly – Version 2.0"

November 2018 – Henri Kohlberg, TINF16B1

Aktueller Status:

Nach längerer Einarbeitung in die bereits bestehenden Versionen von "I believe I can fly" konnten beide Versionen erfolgreich ausgeführt und nachvollzogen werden.

Version 1: Funktioniert recht zuverlässig, benötigt jedoch Xbox Kinect zur Erkennung von Personen und läuft nur auf Windows

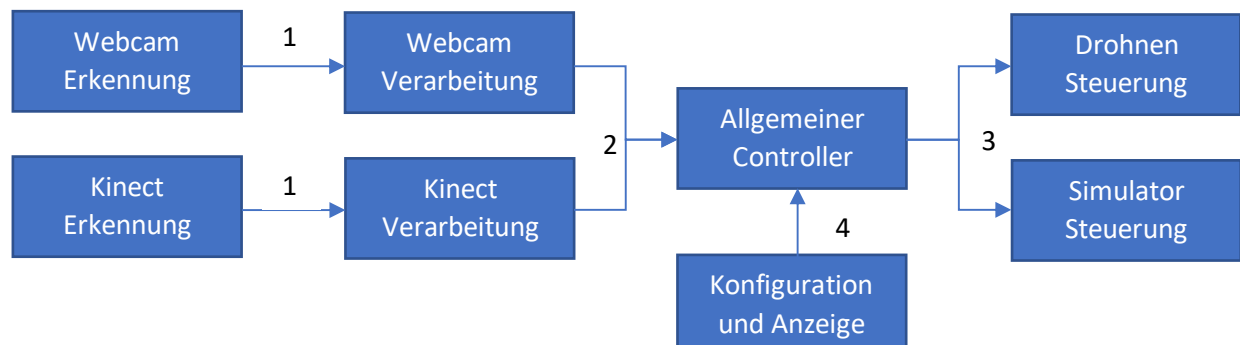
Version 1.5: Verwendung von ROS und Kinect-Tracking auf Ubuntu 14, jedoch reagiert der verwendete Simulator recht träge und hängt grafisch einige Jahre hinterher.

Start der Arbeit an Version 2, die eine normale 2D-Webcam zur Gesten-Erkennung verwenden soll. Dafür wurden verschiedene Lösungen zur Personen-Erkennung verglichen und getestet. Die nun verwendete Bibliothek heißt [OpenPose](#). Da ich kaum Kenntnisse in C/C++ habe, werde ich für die Implementierung Python verwenden. Dies ist kein Problem, da ROS sowohl C++ als auch Python nativ unterstützt, selbiges gilt für OpenPose. OpenPose habe ich daher in einen ROS Node verpackt, welcher die erkannten Körperteile einer Person in dem Channel `"/person_detection"` für andere Nodes zu Verfügung stellt. Die Erkennung von Personen benötigt zwischen 80% und 90% der GPU Leistung, mehr dazu unter "Zu lösende Problemen".

Auch AirSim, der geplante Simulator, hat mittlerweile eine offizielle ROS Integration erhalten, bisher sieht das nach einer vielversprechenden Lösung aus.

Planung:

Da ich mit ROS arbeite, habe ich mir vorerst ein Konzept für die ROS Nodes überlegt:



1) Rohdaten der Sensoren

2) Einheitliches Eingabeformat für Steuerungs-Controller

3) Steuerungsbefehle für entsprechenden Ausgang

4) Steuerung des Controllers zur Konfiguration eines Nutzers (Auswahl von Eingang und Ausgang, ...)

Diese Aufteilung kann sich durchaus noch ändern, es war lediglich ein erster Gedanke von mir, in welche Richtung sich das Projekt entwickeln könnte. Fokus liegt vorerst auf der Steuerung durch die Webcam und die Verarbeitung der Rohdaten => Gestenerkennung

Zu lösende Probleme:

- Die GPU wird bereits zum Großteil (80-90% Auslastung) durch die Personen-Erkennung beschäftigt, deshalb sind nur noch wenige Ressourcen zum Rendern der Simulation übrig, welche auch relativ aufwendig für die GPU ist.
- Die Gesten werden nicht 1:1 von der Kinect-Steuerung übernommen werden können, da die Positionen der Körperteile lediglich in 2D-Koordinaten und nicht in 3D-Koordinaten vorliegen, wie sie die alte Steuerung benötigt. Deshalb muss ein neuer Algorithmus geschrieben werden, welcher die Gesten aus 2D-Koordinaten erkennen kann.

Zeitliche Planung:

Aktuell habe ich noch gar nicht an der eigentlichen Arbeit geschrieben, ich habe mich vorerst nur mit der Implementierung beschäftigt. Dies wird sich wahrscheinlich auch nicht mehr in den verbleibenden Wochen des 5. Semesters ändern. Des Weiteren werde ich, um Aussichten auf ein erfolgreiches Projekt-Ende zu haben, auch in den Praxisphasen/Freizeit daran weiter arbeiten.