

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

Dezember 2018 – Henri Kohlberg, TINF16B1

Aktueller Status:

Die von der Webcam aufgenommenen Daten und mit dem neuronalen Netz verarbeiteten Daten können nun als Pose erkannt werden und in Flugbefehle übersetzt werden. Dazu wurden die Daten im JSON-Format gespeichert und könnten somit theoretisch auch von anderen Programmen ohne Probleme verarbeitet werden. Folgende Gesten-Steuerung kommt dabei zum Einsatz:

- Grundstellung, gleich mit Version 1
 - Beide Arme seitlich vom Körper jeweils zu 90° abstrecken
- Flughöhe (Throttle), gleich mit Version 1
 - Mehr Höhe => Arme mit Kreisbewegung seitlich nach oben nehmen
 - Weniger Höhe => Arme mit Kreisbewegung seitlich nach unten nehmen
- Drehung / seitliche Bewegung (Yaw / Roll), gleich mit Version 1
 - Links => Linken Arm nach unten nehmen, rechten Arm nach oben nehmen
 - Rechts => Rechten Arm nach unten nehmen, linken Arm nach oben nehmen
- Vorwärts (Pitch), anders als bei Version 1
 - Vorwärts: Beide Arme auf ca. Schulterhöhe nach vorne nehmen
 - Rückwärts: Ein Rückwärtsflug ist mit dieser Steuerung nicht möglich, jedoch hätte ein Pilot mit Videobrille sowieso keine Sicht nach hinten und somit wäre ein Flug in eine unbekannte Region theoretisch gesehen zu riskant.

Die Steuerung wurde Prototyp-mäßig implementiert, ist jedoch noch nicht als ROS-Modul verpackt. Ebenfalls würde ich eine saubere Implementierung wie auch die vorherige Studienarbeit mit einem Fuzzy-Controller umsetzen und nicht mit der manuellen Abfrage von Werten. Die aktuelle Lösung erlaubt mir jedoch eine feinere Kalibrierung, welche vor Allem beim Vorwärtsflug von Vorteil ist. Die endgültige Lösung muss noch ausgearbeitet werden.

Planung:

Die Prototyp-mäßige Implementierung muss in ROS implementiert werden und anschließend in ein passendes Format für die Drohne / den Simulator umgewandelt werden. Ich werde den Fokus auf die Verwendung des Simulators legen, damit zügig eine lauffähige Demo entsteht.

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.

Zeitliche Planung:

Aktuell habe ich noch gar nicht an der eigentlichen Arbeit geschrieben, ich habe mich vorerst nur mit der Implementierung beschäftigt.