Gymnasium Wolbeck

Raumanalyse
durch einen Quadrokopter

Alexander Neuwirth, Jonathan Sigrist

Jugend forscht 2016

Inhaltsverzeichnis

1 Wir stellen uns vor 1
2 Einleitung 1
3 Vorüberlegungen 1
3.1 Erste Planungen und Ideen 1
3.2 Beschaffen der Drohne und Installation der Software 2
3.3 Ansteuerung der Drohne / Analyse der Daten 2
4 Projektablauf 3
4.1 Testen der Drohne 3
4.2 Erstes Programmieren der Drohne 3
4.3 Die Programmierelemente von Node.js® 3
5 Die Drohne programmieren 3
5.1 Steuerung über die Konsole 3
5.2 Eine Website für Interaktion mit dem Piloten 4
5.3 Entfernungsmessung mit Ultraschall 4
5.4 Stabilisierung 4
5.5 Bildanalyse 5
6 Streckenauswertung 7
7 Darstellung am Computer 8
8 Probleme und Lösungen 8
8.1 Kamerafokus 8
8.2 Videostream 8
8.3 Horizontale Entfernungsmessung 9
9 Ausblick 9
9.1 Automatische Raumanalyse 9
9.2 Rundumanalyse 9
9.3 Erkennen von Hindernissen 10
10 Zusammenfassung 10
11 Anhang: Vorschriften für die Nutzung von Quadrokoptern 11
12 Literaturverzeichnis 12

1 Wir stellen uns vor

Wir sind Schüler der Jahrgangsstufe Q2 am Gymnasium Wolbeck. Beide haben wir Interesse an Informatik, Physik und Technik und arbeiten auch mal außerhalb der Schulzeit gerne an Projekten. Wir sind 17 Jahre alt und haben beide unabhängig voneinander bereits an einem Jugend forscht Wettbewerb teilgenommen.

Alexander Neuwirth hat im Jahr 2015 mit einer LEGO®-Mindstorms Roboter Programmierung (NXT Android Control Internet basierte Steuerung eines mit Smartphone ausgestatteten Legoroboters) bei Jugend forscht teilgenommen und den 3.Preis im Landeswettbewerb NRW erlangt. Außerdem hat er beim KICK Wettbewerb (kluge Ideen, clevere Köpfe) der Stadtwerke Münster mit seinem Jugend-forscht-Projekt teilgenommen. Seine Facharbeit hat er in Informatik geschrieben.

Jonathan Sigrist hat bereits im Jahre 2011 bei Jugend forscht und anderen Wettbewerben mit einem Projekt zur effizienten Nutzung von Solarzellen(automatische Solarzellenausrichtung) in der Physik beigetragen und einen Sonderpreis für Umweltbewusstsein erlangt. Seine Facharbeit hat er im Fach Physik geschrieben. Bereits mehrere Male hat er sich an außerschulischen Camps beteiligt.

2 Einleitung

Ziel des Projektes ist die Vermessung und grafische Darstellung eines geschlossenen Raumes mit Hilfe einer Parrot® AR.Drone 2.0 (Quadrokopter).

Die Steuerung des Quadrokopters (kurz: Drohne) soll mittels des *ar-drone* Moduls in Node.js® (serverseitige Plattform zum Betrieb von Netzwerkanwendungen basierend auf einer JavaScript-Laufzeitumgebung) realisiert werden.

Die Drohne erhält Steuerkommandos von einem Computer über eine WLAN-Verbindung und übermittelt ihrerseits Daten an diesen, beispielsweise einen Videostream von der integrierten HD-Kamera. Empfangene Daten sollen dann vom Computer ausgewertet und zu einer grafischen Darstellung des Raums aufbereitet werden (Erste Phase).

Des Weiteren soll der Quadrokopter im Auto-Modus, d.h. möglichst ohne durch Menschen gesteuert zu werden, ein Ziel ansteuern und dabei den Raum analysieren und Hindernissen ausweichen können (Zweite Phase).

Am Ende soll eine vollständige, möglichst fehlerfreie und automatische Analyse eines Raumes ohne Zielvorgabe möglich sein (Dritte Phase: vollständiger Auto-Modus).

3 Vorüberlegungen

3.1 Erste Planungen und Ideen

Unsere Idee verfolgt die Realisierung einer vollautomatischen Analyse eines Raumes mittels einer Drohne. In der ersten Phase soll die Drohne von einem Piloten gesteuert werden. Hierbei erfolgt die Steuerung über First Person View (FPV), also mittels Kameratechnik aus der Perspektive des ferngesteuerten Modells.Dazu war das von Alexander durchgeführte Projekt mit dem LEGO®-NXT Roboter natürlich eine gute Vorlage. Allerdings ging es dort hauptsächlich um die Verbindung und die allgemeine Steuerbarkeit und nicht um die Automatisierung und Analyse erhaltener Daten.

Nach längerer Recherche im Internet haben wir uns für die AR.Drone 2.0 von Parrot® entschieden, da sie mittels Node.js® einfach ansteuerbar ist und gute Flugeigenschaften besitzt. Außerdem hat sie einen USB-Anschluss zum Datenaustausch (dieser wird für das Projekt allerdings nicht verwendet). Auch ist die Reichweite der Drohne lediglich durchdas WLAN-Netz begrenzt und lässt sich somit durch den Einsatz eines WLAN-Routers schnell vergrößern.

Den Ultraschallsensor der AR.Drone wollen wir nach vorne richten, um damit einen Raum zu vermessen. Aus diesen Daten kann man den Grundriss des gesamten Raumes berechnen und sich auf einem Computer grafisch anzeigen lassen. In der zweiten Phase kann der Pilot einen beliebigen Punkt im Raum als Ziel angeben (Koordinaten). Die Drohne fliegt dann eine geeignete Flugroute zum Zielort ohne mit Hindernissen zu kollidieren. In der dritten Phase soll die Drohne (ohne Zielvorgabe) selbstständig den Raum analysieren (Vollständiger Auto-Modus).

3.2 Beschaffen der Drohne und Installation der Software

Um die Kosten möglichst klein zu halten, haben wir uns eine gebrauchte und generalüberholte AR.Drone von Parrot® gekauft.

Ausstattung der Drohne:

- 3-achsiger Beschleunigungssensor
- 3-achsiges Gyroskop
- Drucksensor (für Höhe)
- 3-achsiges Magnetometer
- Ultraschallsensor
- 60 fps Bodenkamera
- 30 fps Frontkamera

Da der mitgelieferte Akku nur eine Laufzeit von zwölf Minuten hat, haben wir noch einen weiteren Akku mit 2000mAh gekauft, um längere Testflüge absolvieren zu können. Mit der kostenlosen Handy/Tablet-App "AR.FreeFlight" von Parrot® kann die Drohne sehr leicht geflogen werden. Um sie mit dem Computer zu steuern, wird Node.js® und das frei verfügbare Modul *ar-drone* von Felix Geisendörfer benötigt. Als Editoren haben wir notepad++ und vim genutzt. Um parallel programmieren zu können (Synchronisieren des Projektes) haben wir git (Software zur verteilten Versionsverwaltung von Dateien) verwendet.

3.3 Ansteuerung der Drohne / Analyse der Daten

Die Drohne baut beim Anschalten ein WLAN-Netz auf, in welches sich ein Computer mit WLAN-Schnittstelle einloggen kann. Der Computer kann dann Befehle an die Drohne schicken und die Drohne kann Navigations- und Sensordaten an den eingeloggten Computer übermitteln.

Die auszuführenden Befehle werden über einen Browser ausgewählt, d.h. die Software auf dem Computer ist ein HTTP-Server, der die gewünschten Befehle des Anwenders (Pilots) an die Drohne weiterleitet und die Sensordaten der Drohne für die Darstellung aufbereitet.

Hauptbestandteil der Website ist ein Videostream der Frontkamera oder Bodenkamera der Drohne, welcher in Echtzeit analysiert wird. Ebenso werden die Position und die zurückgelegte Strecke der Drohne auf der Website dargestellt.

4 Projektablauf

4.1 Testen der Drohne

Als die Drohne angekommen ist, haben wir die Flugeigenschaften, Module und Sensoren mit Hilfe der Handy-App getestet und ein Gefühl dafür entwickelt, wie man die Drohne ansteuern muss, um eine möglichst genaue Flugbahn zu erhalten. Die App wurde auch später oft verwendet, um Probleme (siehe Kapitel 8 Probleme und Lösungen) aufzudecken und zu beheben.

4.2 Erstes Programmieren der Drohne

Der erste Schritt war, die Drohne über den Computer zu steuern. Dies ermöglichte die direkte Implementierung des Moduls *ar-drone* in eine JavaScript Datei, welche dann mit Node.js® ausgeführt wird. Über die Konsole können direkte Befehle an die Drohne geschickt werden (zum Beispiel: "takeoff()", "forward(0.1)").

4.3 Die Programmierelemente von Node.js®

Bei Node.js® handelt es sich um eine serverseitige Plattform zum Betrieb von Netzwerkanwendungen basierend auf einer JavaScript-Laufzeitumgebung. Somit kann auf Client- sowie Serverseite in JavaScript programmiert werden. Auf der Serverseite ergeben sich Verwaltungsvorteile. Insbesondere die einfache Möglichkeit eine Website mit in das Programm einzubinden wurde von uns verwendet.

Um auf die Drohne zuzugreifen, muss das Modul *ar-drone* in das Projekt eingebunden werden. Danach können alle Funktionen frei genutzt werden (siehe Kapitel 4.2 Erstes Programmieren der Drohne). Zum Beispiel liefert die Drohne die Möglichkeit auf einen Videostream. Diesen kann man mit dem Modul *dronestream*, das auf *ar-drone* aufbaut, an den Browser übertragen und in Echtzeit aufbereiten lassen.

Auch sollen Befehle vom Webinterface möglichst ohne Verzögerung von der Drohne ausgeführt werden. Das Modul *socket.io* bietet die Möglichkeit einer intuitiven Implementation von WebSockets und somit einer direkten Datenübertragung vom Browser zum Server.

5 Die Drohne programmieren

Im folgenden Kapitel wird erklärt, wie die Drohnensteuerung programmiert ist und wie sich durch ein Programm oder durch einen Benutzer komplexe Abläufe realisieren lassen. Dabei werden Grundfunktionen programmiert, auf die später aufgebaut werden kann.

5.1 Steuerung über die Konsole

Wie in Kapitel 4.2 Erstes Programmieren der Drohne bereits erwähnt, lässt sich die Drohne nach Implementierung des Moduls *ar-drone* über die Konsole steuern. Damit konnten wir schon einfache Programme zur Realisierung von Grundfunktionen schreiben, wie „Drohne starten“, „Drohne hochfliegen“ und „Drohne landen“. Die Anforderungen der ersten Phase sind somit erfüllt.

5.2 Eine Website für Interaktion mit dem Piloten

Für die weitere Arbeit war eine erweiterte Benutzeroberfläche nötig. Eine Website (siehe Abbildung 5: Website) bietet zwei weitere große Vorteile gegenüber der Konsole. Zum einen kann der Videostream abgebildet werden und die spätere grafische Darstellung der Daten ist möglich. Zum anderen kann man sich aber auch mit mehreren Clients (mehre Piloten) mit dem Server verbinden.

Wir haben also einen Webserver programmiert, welcher sich mit der Drohne verbindet. Mit diesem Server kann man sich dann mittels eines Internetbrowser verbinden. Damit erhält man die Kontrolle über die Drohne. Es ist so auch möglich, sich mit mehreren Clients auf einem Server zu verbinden und so die Drohne parallel anzusteuern, jedoch kann es zu Konflikten beim Steuern kommen, da beide Clients gleichberechtigt sind.

5.3 Entfernungsmessung mit Ultraschall

Vertikale Entfernungsmessung

Die AR.Drone hat an ihrer Unterseite einen Ultraschallsensor, welcher die Entfernung durch Schallrückwurf zum nächsten Objekt recht genau angibt. Als erstes haben wir untersucht, in welchem Bereich der Sensor die Entfernung messen kann und wie stark sich die Streuung des Schalls auf das Ergebnis auswirkt. Dabei kam ein erstaunlich genauer Wert heraus, so dass Entfernungen von 5 cm bis 6m mit einer Messungenauigkeit im Zentimeterbereich ermittelt werden.

Horizontale Entfernungsmessung

Für die horizontale Entfernungsmessung hat die Drohne standardmäßig keinen Sensor. Um die Entfernung nach vorne anstatt nach unten zu messen, haben wir erst einen Schallspiegel in einem 45° Winkel angebracht. Bei dem Versuch ist allerdings die Flugfähigkeit der Drohne sehr stark beeinträchtigt worden, da diese alle vorhandenen Sensoren selbst verwendet um beispielweise die Höhe halten zu können. Darüber hinaus sind die durch Spiegelung ermittelten Werte ungenau oder häufig über lange Perioden komplett ungültig Siehe auch Kapitel 8 Probleme und Lösungen.

5.4 Stabilisierung

In der zweiten Phase muss die Drohne eigenständig zu Zielpositionen fliegen können. Das Programm steuert dabei mit dem Kompass der Drohne, dem Neigungswinkel und der Verschiebung des Bildes der Bodenkamera die Drohne so, dass sie von der Startposition aus in eine bestimmte Richtung zum Ziel fliegt. Hierzu muss die Drohne Hindernisse erkennen, um ihnen auszuweichen zu können.

Mit dem Drehbefehl lässt sich die Drohne um 360° drehen. Während der Drehung kann die horizontale Entfernung laufend gemessen werden (siehe Kapitel 8.3 Horizontale Entfernungsmessung). Dabei ist es wichtig, dass die Drohne auf einer Position stehen bleibt und nicht weggleitet. Das ist automatisch durch die Verfolgung des Bildes der Bodenkamera gewährleistet. Auch durch die Frontkamera kann die Position stabilisiert werden. Durch Gleiten nach rechts oder links, sowie nach oben oder unten verschiebt sich das Bild dementsprechend. Gleitet die Drohne nach vorne oder nach hinten, wird die Darstellung der aufgenommenen Gegenstände kleiner oder größer (Perspektive).

Wir entschieden uns das Modul *ardrone-autonomy (*auf *ar-drone* basierend*)* zu verwenden, da dieses den Extended Kalman Filter implementiert (Algorithmus zur Zustandsschätzung, zum Beispiel Position, Geschwindigkeit), um die Position der Drohne zu stabilisieren. Das Modul verwendet hierfür Navigationsdaten der Drohne. Somit kann weiterhin der Videostream der Frontkamera oder der Bodenkamera auf der Webseite anzeigt werden.

Außerdem bietet dieses Modul die Möglichkeit Flugbefehle in Metern anzugeben (zum Beispiel „fliege x Meter vor“).

5.5 Bildanalyse

Die Bildanalyse findet auf der Client-Seite statt, um den die Drohne steuernden Server zu entlasten. *JSFeat* ist eine speziell für Computer Vision auf der Client-Seite unter Verwendung von JS/HTML5 ausgelegte Bibliothek. Wir haben in unser Projekt hieraus zwei Algorithmen zur Kantendetektion eingebunden:

1. Canny edge detector.
2. scharr_derivatives

Beide Algorithmen basieren auf Grauwertbildern.

Abbildung 1: Orginalbild

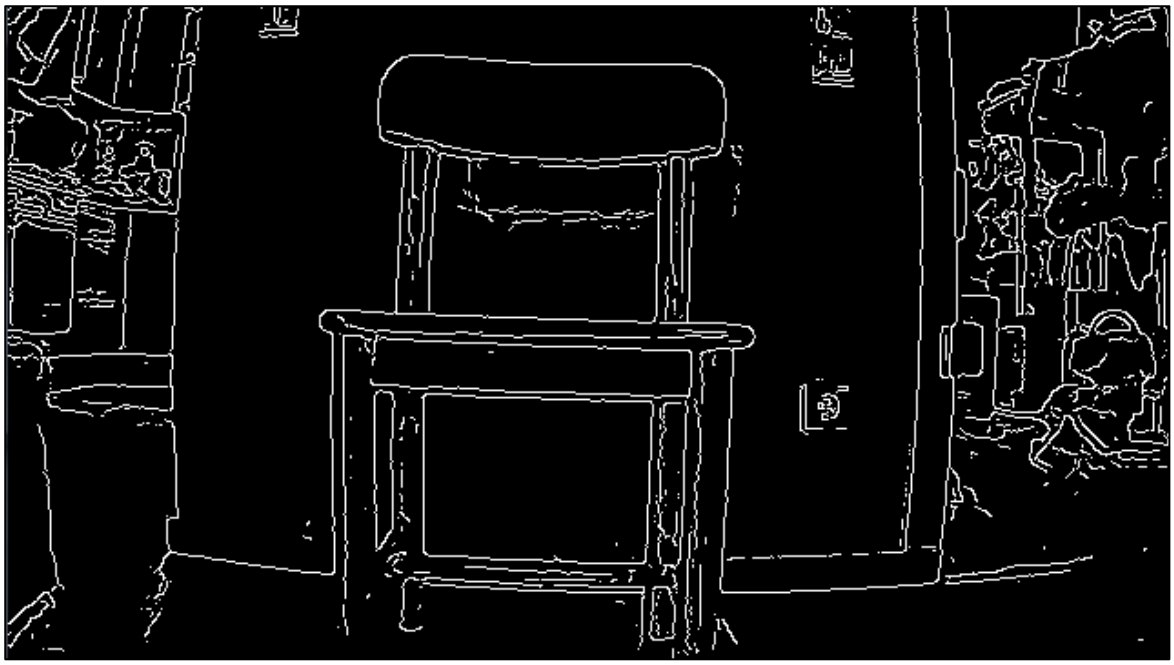

Abbildung 2: Canny edge detector

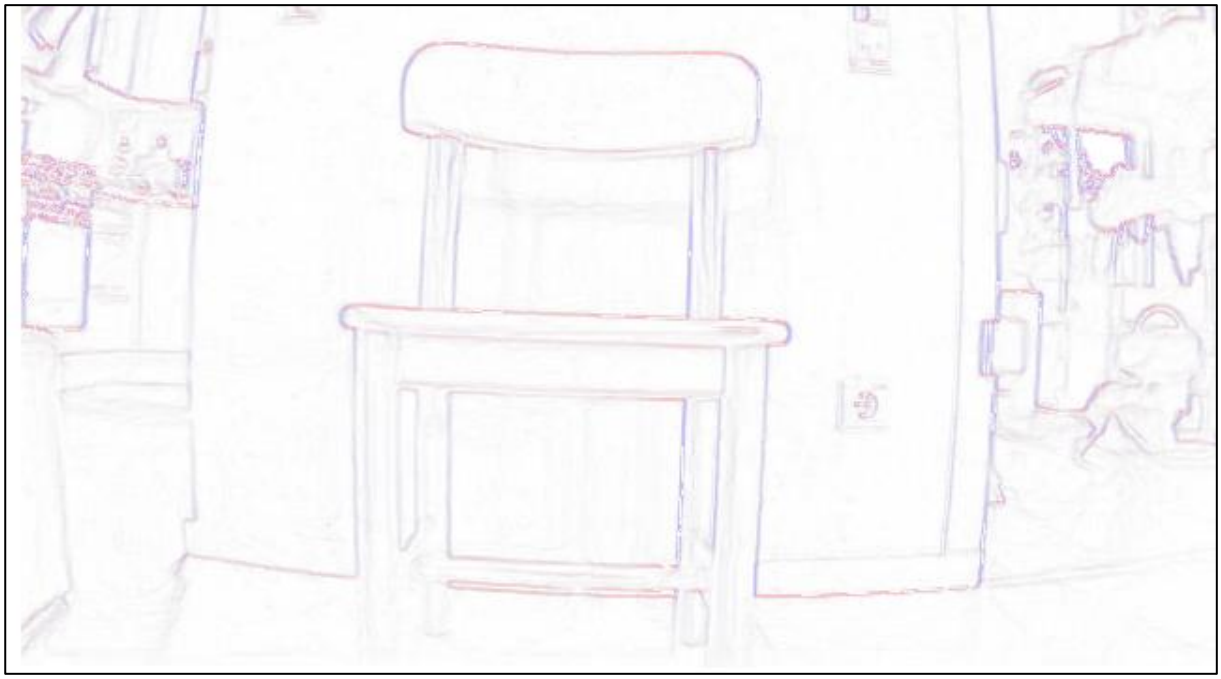

Abbildung 3: scharr_derivatives

6 Streckenauswertung

Die mit Hilfe des Moduls *ardrone-autonomy* aufbereiteten Positionsdaten der Drohne werden grafisch dargestellt (siehe Abbildung 4: Steckenverlauf der Drohne).

Zusätzlich könnten über den horizontalen Entfernungsmesser (siehe Kapitel 8.3) ermittelte Hindernisse in die grafische Darstellung aufgenommen werden.

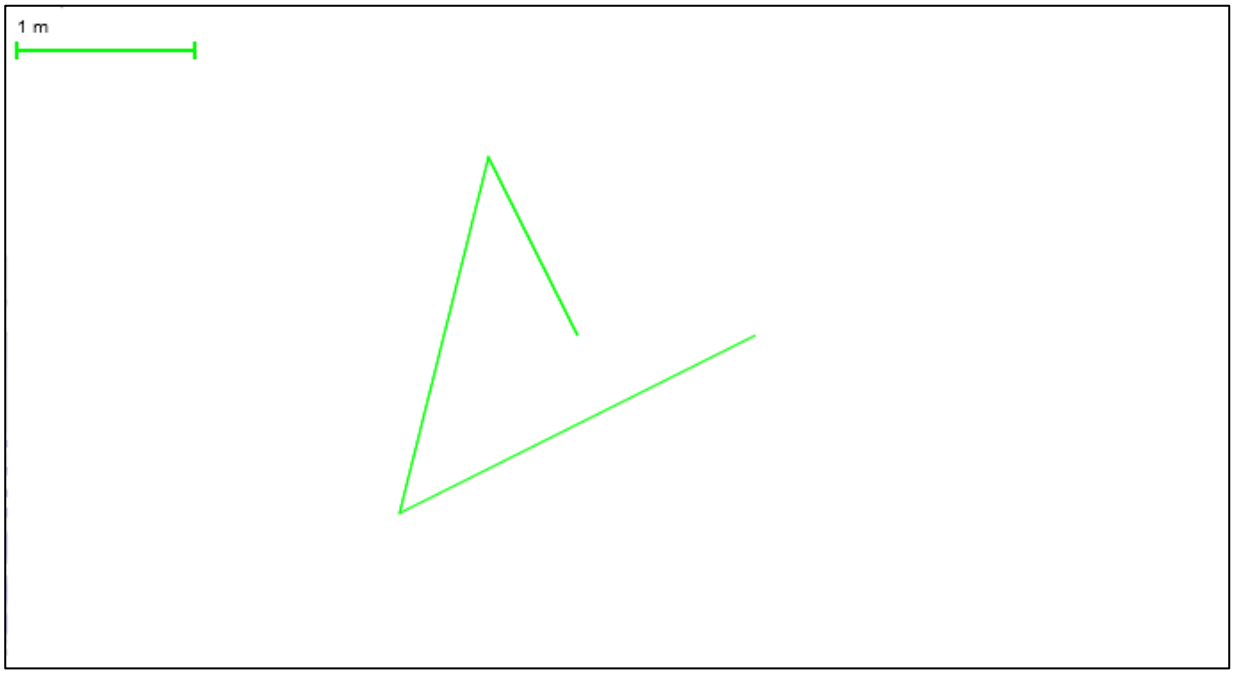

Abbildung 4: Steckenverlauf der Drohne

7 Darstellung am Computer

Abbildung 5: Website zeigt das Frontkamerabild, das die Drohne in Startposition aufgenommen hat. Daneben sieht man die bearbeitete Canny edge detector bzw. scharr_derivatives Darstellung. Der Streckenverlauf wird noch nicht angezeigt, da die Drohne in Startposition sitzt. Im unteren Bereich werden die Steuerbuttons dargestellt.

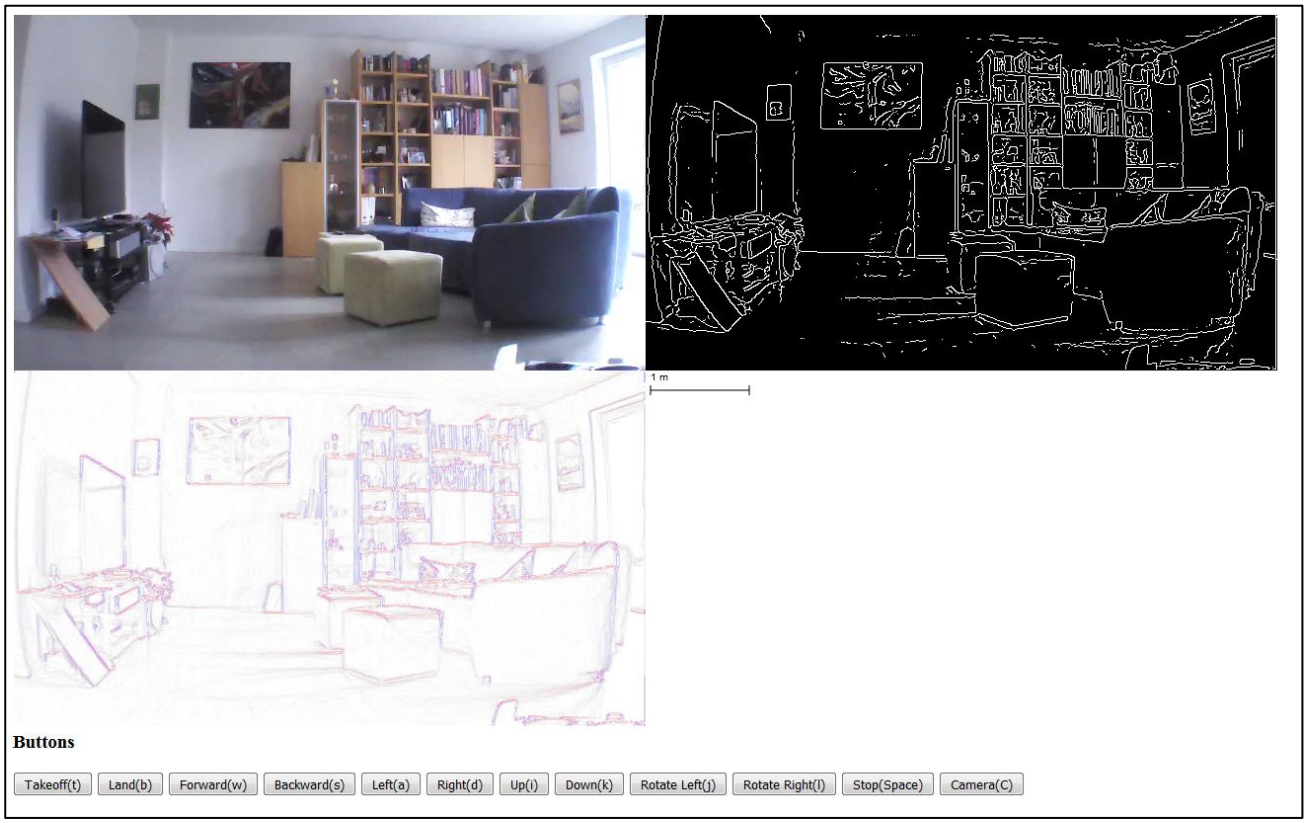

Abbildung 5: Website

8 Probleme und Lösungen

8.1 Kamerafokus

Während des Testens der Drohne (siehe Kapitel 4.1) ist uns aufgefallen, dass das Kamerabild unscharf war. Nach etwas Recherche im Internet haben wir herausgefunden, dass der Fokus der Linse der Frontkamera von Parrot® falsch eingestellt worden ist. Diesen konnte man verstellen, nachdem man den Klebstoff, der zum Festhalten der Linse gedacht ist, entfernt hat. Aufgrund weiterer Tests konnten wir die Position der Linse korrigieren und dauerhaft fixieren.

8.2 Videostream

Zuerst haben wir das Modul *ar-drone-png-stream* verwendet um einzelne PNG-Bilder an den Browser zu übertragen, jedoch hat sich gezeigt, dass die Übertragung eine Verzögerung von etwa 2 Sekunden hatte. Deshalb haben wir das komplexere Modul *dronestream* implementiert.

8.3 Horizontale Entfernungsmessung

In Kapitel 5.3 Entfernungsmessung mit Ultraschall wurde beschrieben, dass kein Sensor zur horizontalen Entfernungsmessung im Ausstattungsumfang der Drohne enthalten ist. Dieser wird aber für die in Kapitel 6 beschriebene Streckenauswertung benötigt.

Es stellen sich nun drei Lösungsmöglichkeiten für das Problem dar:

1. Eine Untersuchung des **Videostreams**.
    - Mit Hilfe der Videoübertragung und geeigneten Analysealgorithmen erhält man Daten über die Entfernung und Positionierung von Objekten vor der Drohne. Hier liegt das Problem darin, dass eine komplette dreidimensionale Untersuchung mit nur einem Kamerabild in Echtzeit nicht in der gewünschten Genauigkeit realisierbar ist.
    - Man könnte zwei Bilder des Videostreams, die sich geringfügig in der Position der Drohne unterscheiden, verwenden. Somit könnte sich ein ziemlich realistisches dreidimensionales Modell erstellen lassen, jedoch ist auch diese Bildauswertung in Echtzeit nicht realisierbar.
2. Ein **Laser** für exakte punktgenaue Messungen. Um wirklich genaue Werte zu analysieren, kann ein Lasermessgerät genutzt werden. Es ist allerdings recht teuer und muss extern an die Drohne angebaut werden. Des Weiteren muss die Messung so ausgerichtet werden, dass die Abmessungen der Drohne berücksichtigt werden, eine punktgenaue Messung reicht nicht aus.
3. Ein weiteres **Ultraschallmessgerät**. Ein separates Messgerät ermöglicht in Kombination mit einem Raspberry-Pi eine völlig ausreichende Genauigkeit. Zusätzlich werden so auch Objekte kurz unter oder über der Flughöhe miterkannt. Im Gegensatz zum Laser (siehe Punkt 2) werden Flächen analysiert und nicht nur einzelne Punkte. Das hat den Vorteil, dass Fehlermessungen mit herausragenden Objekten vermieden werden können.

Ein weiterer separater Ultraschallsensor hat also Vorteile gegenüber der Lasermessung.

9 Ausblick

9.1 Automatische Raumanalyse

Die automatischen Raumanalyse (Auto-Modus, zweite Phase) wird im Rahmen des Projektes als nächster Schritt realisiert. Hierzu ist noch ein Raspberry-Pi sicher auf der Drohne zu befestigen und ein Ultraschallsensor zu beschaffen.

9.2 Rundumanalyse

Bei der Rundumanalyse dreht sich die Drohne einmal um die eigene Achse und nimmt Messpunkte auf (siehe Kapitel 5.4 Stabilisierung). Dadurch erhält man einen zweidimensionalen Grundriss des Raumes. Die Rundumanalyse dient der Vorbereitung zur dritten Phase.

9.3 Erkennen von Hindernissen

Erkennt die Drohne (Ultraschallsensor) ein Hindernis, so kann dies lokalisiert und im Streckenverlauf dargestellt werden, indem man neben der Position der Drohne das Hindernis (gegeben durch die Messung der Richtung und der Entfernung) darstellt. Einzelne Messungen werden in Verbindung gebracht. Somit wird der Umriss eines Hindernisses aus mehreren Messungen ermittelt.. Liegen nacheinander gemessene Punkte zu weit auseinander (z.B. Außenecke eines Gegenstandes), so soll der Raum zwischen den Punkten (Längs- bzw. Querseite des Gegenstandes) weiter analysiert werden. Ist ein Punkt ein Ausreißer zwischen vielen Punkten (z.B. einzelnes Loch in einer Wand), wird er als ungültig gewertet und nicht mit den übrigen Messpuntken in Zusammenhang gebracht. Die Genauigkeit der Messung kann durch verschiedene Parameter beeinflusst werden:

1. Dreht sich die Drohne langsamer, erhält man einen kleineren Drehwinkel zwischen den Messungen, d.h. die Abtastrate und somit die Genauigkeit werden höher
2. Wählt man den maximal zulässigen Abstand zwischen zwei nacheinander gemessenen Punkten kleiner, so wird eine höhere Genauigkeit erzwungen und Störobjekte wie beispielsweise Stangen oder Kabel werden besser erkannt.

10 Zusammenfassung

In der Arbeit wurde die Ansteuerung des Quadrokopters realisiert. Hierzu wurde nach umfangreichen Tests der Drohne mit der freiverfügbaren App, die Ansteuerung der Drohne durch einen Computer(Konsole) implementiert. Dabei fiel uns auf, dass die von der Kamera gelieferten Bilder unscharf und für eine Bildanalyse nicht geeignet waren. Dieses Problem konnte durch einen manuelle Einstellung des Kamerafokus behoben werden.

Anschließend wurden die Module *ar-drone*, *dronestream* und *ardrone-autonomy* in die Steuerprogramme integriert. Dadurch wurde das Problem der zu langsamen Bildübertragung gelöst.

Die Interaktion zwischen Drohne und Pilot beziehungsweise zwischen Drohne und Piloten erfolgt über eine von uns erstellte Website. Auf dieser werden neben dem eigentlichen Kamerabild, das von der Drohne in Echtzeit übertragen wird, die aufgrund der Kantendetektion (canny edge detector, scharr_derivatives) berechneten Bilder dargestellt.

Wenn man mit der Drohne die Ränder eines Raumes abfliegt, kann der Grundriss eines Raumes mit der von uns programmierten Streckenauswertung dargestellt werden.

Da ein Sonar für die horizontale Entfernungsmessung nicht in der Ausstattung der Drohne enthalten ist, erfordert die Raumanalyse mehr Steuerung als geplant war. Deshalb konnte die Implementierung einer dreidimensionalen Darstellung noch nicht realisiert werden. Die Erweiterung der Drohne um ein Raspberry-Pi und einen Sonarsensor wird im Rahmen des Projektes als nächster Schritt realisiert werden.

11 Anhang: Vorschriften für die Nutzung von Quadrokoptern

Diese vier Bedingungen sind für unser Projekt relevant (Auszug):

1. Das deutsche Luftsicherheitsgesetz schließt eine Verwendung von Drohnen außerhalb des Sichtfeldes eines menschlichen Bedieners kategorisch aus.
2. Prinzipiell ist das fliegen von UAVs (Unmanned Areal Vehicle) versicherungspflichtig.
3. Innerhalb eines Abstandes von 1,5 km vom Flughafenzaun ist die Nutzung von Flugmodellen und unbemannten Flugsystemen grundsätzlich ganz verboten.
4. Über Menschenmengen, militärischen Objekten, Kraftwerken und Krankenhäusern darf grundsätzlich nicht geflogen werden.
5. Der Einsatz von Luftfahrzeugen mit Video- oder Fotomöglichkeiten wird als Beeinträchtigung der Privatsphäre betrachtet.

12 Literaturverzeichnis

bkw. 2016. node-dronestream. [Online] github.com, 13. Januar 2016. https://github.com/bkw/node-dronestream.

Castledine, Earle und Sharkie, Craig. 2010. *jQuery: Novice to Ninja.* Collingwood : SitePoint Pty. Ltd., 2010. ISBN 987-0-9805768-5-6.

drohnen.de. 2016. Vorschriften, Genehmigungen für die Nutzung von Drohnen und Multicoptern. [Online] 13. Januar 2016. http://www.drohnen.de/vorschriften-genehigungen-fuer-die-nutzung-von-drohnen-und-multicoptern/.

eschnou. 2016. ardrone-autonomy. [Online] github.com, 13. Januar 2016. https://github.com/eschnou/ardrone-autonomy.

felixge. 2016. node-ar-drone. [Online] github.com, 13. Januar 2016. https://github.com/felixge/node-ar-drone.

Geldner, Andreas. 2016. Unbemannte Fluggeräte: Gesetze bremsen den Milliardenmarkt für Drohnen. *Stuttgarter Zeitung.* [Online] 13. Januar 2016. • http://www.stuttgarter-zeitung.de/inhalt.unbemannte-fluggeraete-gesetze-bremsen-den-milliardenmarkt-fuer-drohnen.4ce7ef60-3279-4434-94b0-9078fa207dca.html.

inspirit. 2016. jsfeat. [Online] github.io; github.com, 13. Januar 2016. https://inspirit.github.io/jsfeat/; https://github.com/inspirit/jsfeat.

Node.js. 2016. Node.js. *Node.js.* [Online] 13. Januar 2016. https://nodejs.org/en/.

Wikipedia. 2016. Unbemanntes Luftfahrzeug. *Wikipedia.* [Online] 13. Januar 2016. https://de.wikipedia.org/wiki/Unbemanntes_Luftfahrzeug#Deutschland.