

Bachelorarbeit

Johannes Meyer

Indoornavigation eines Quadcopters

Johannes Meyer
Indoornavigation eines Quadcopters

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Technische Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. rer. nat. Gunter Klemke
Zweitgutachter : Prof. Dr. Franz Korf

Abgegeben am 12. Mai 2010

Johannes Meyer

Thema der Bachelorarbeit

Indoornavigation eines Quadcopters

Stichworte

Quadcopter, MikroKopter, Webcam, Bilderkennung, Indoor-Navigation, Lageregelung

Kurzzusammenfassung

Im Rahmen dieser Arbeit wird ein Hard- und Softwaresystem zur Positionserkennung des Quadcopters in geschlossenen Räumen entwickelt, sowie die automatische Lageregelung. Mittels einer stationären Webcam an der Decke und drei LEDs am Quadcopter wird die Position bestimmt. Über eine Bluetooth Funkverbindung werden vom Rechner automatisch Steuerbefehle ausgesendet um den Quadcopter stabil auf Position zu halten, bzw. eine neue Position anzusteuern.

Johannes Meyer

Title of the paper

Indoornavigation with a quadcopter

Keywords

quadcopter, mikrokopter, webcam, image recognition, indoor navigation, position control

Abstract

In this thesis, a hardware and software system for position recognition in closed rooms for quadcopters is developed, and the automatic position control. Using a stationary webcam on the ceiling and three LEDs on the quadcopter, the position is determined. The computer automatically sends via Bluetooth wireless connection, commands out to the quadcopter to keep it stable in position, or to navigate to a new position.

Danksagung

Ich möchte mich bei meinen Eltern bedanken, die es mir ermöglicht haben zu studieren. Nicht nur finanziell, sondern auch moralisch stand mir meine Familie zur Seite und war stets Basis und Ruhepunkt, aus der ich Kraft schöpfen konnte. Des weiteren möchte ich mich bei William Thielicke bedanken, der mir durch sein Projekt erst die Idee für diese Bachelorarbeit gegeben hat. Besonderer Dank geht an Gunter Klemke, meinen betreuenden Professor, der mich bei meinem Vorhaben unterstützt hat und sich von dem Projekt ebenso begeistern ließ wie ich. Er hat mir die Gelegenheit gegeben, meinen Kindheitstraum auszuleben, mit einem Modellhubschrauber zu arbeiten. Olaf Rempel möchte ich dafür danken, dass er den Quadcopter für seine damalige Bachelorarbeit zusammen gebaut hat, welcher mir nun als Grundlage für diese Arbeit dient.

Inhaltsverzeichnis

| | |
|--|-----------|
| Tabellenverzeichnis | 8 |
| Abbildungsverzeichnis | 9 |
| 1. Einführung | 11 |
| 1.1. Motivation | 11 |
| 1.2. Zielsetzung | 12 |
| 1.3. Gliederung | 12 |
| 2. Grundlagen | 14 |
| 2.1. Positionsbestimmung | 14 |
| 2.2. Navigation | 14 |
| 2.3. Eigenschaften eines Quadcopters | 15 |
| 2.3.1. Grundlegende Funktion | 15 |
| 2.3.2. Besonderheiten | 16 |
| 2.3.3. Fluglageregelung | 17 |
| 2.3.4. Mögliche Erweiterungen | 18 |
| 3. Indoor Positionsbestimmung | 19 |
| 3.1. GPS – Global Positioning System | 19 |
| 3.2. Cricket/IMAPS | 20 |
| 3.3. Ubisense | 21 |
| 3.4. Paric | 23 |
| 3.5. HiBall | 23 |
| 3.6. WLAN | 25 |
| 3.6.1. RADAR | 25 |
| 3.6.2. MagicMap | 25 |
| 3.7. Bilderkennung | 27 |
| 3.7.1. Kamerabasierten Erkennung von frequenzcodierten Infrarotsendern | 27 |
| 3.7.2. Tracking mittels einer stationären Kamera | 28 |
| 3.8. Zusammenfassung | 29 |
| 4. Analyse | 30 |

| | |
|---|-----------|
| 4.1. Aufgabenstellung | 30 |
| 4.2. Grundlegende Anforderungen an das System | 30 |
| 4.3. Begrenzende Faktoren | 31 |
| 4.4. Abwägung der verschiedenen Positionsbestimmungsverfahren | 31 |
| 4.4.1. GPS | 31 |
| 4.4.2. Cricket/IMAPS | 32 |
| 4.4.3. Ubisense | 32 |
| 4.4.4. Paric | 33 |
| 4.4.5. HighBall | 33 |
| 4.4.6. WLAN | 33 |
| 4.4.7. Bilderkennung | 33 |
| 4.5. Entscheidung für ein Positionsbestimmungsverfahren | 34 |
| 4.6. Zusammenfassung | 35 |
| 5. System-Design | 36 |
| 5.1. Hardware | 36 |
| 5.1.1. Quadcopter | 37 |
| 5.1.2. Positionierung der LEDs | 38 |
| 5.1.3. Positionierung der Kamera | 39 |
| 5.1.4. IR-Filter | 41 |
| 5.2. Software | 42 |
| 5.2.1. Bilderkennung | 43 |
| 5.2.2. Berechnung | 44 |
| 5.2.3. Regelung (PID) | 48 |
| 5.2.4. Steuerbefehle | 50 |
| 6. Realisierung | 51 |
| 6.1. Hardware | 51 |
| 6.1.1. Quadcopter | 51 |
| 6.1.2. Kamera | 52 |
| 6.1.3. LED | 53 |
| 6.1.4. Sicherheitsmaßnahmen | 55 |
| 6.2. Software | 55 |
| 6.2.1. MATLAB Code | 55 |
| 6.2.2. Latenzzeit zwischen Realität und Kamerabild | 57 |
| 6.2.3. Bugs in der Programmierumgebung | 58 |
| 6.2.4. Firmwareanpassung | 58 |
| 6.3. Zusammenfassung | 60 |
| 7. Fazit | 61 |
| 7.1. Verbesserungsvorschläge | 61 |

| | |
|--|-----------|
| 7.2. Ausblick in die Zukunft | 62 |
| 7.3. Abschließende Worte | 63 |
| Literaturverzeichnis | 64 |
| Glossar | 70 |
| A. Inhalt der CD | 72 |
| A.1. Quellcode | 72 |
| A.1.1. Matlab | 72 |
| A.1.2. Mikroopter | 72 |
| A.1.3. Latenzmessung | 72 |
| A.2. PDF Dokumente | 72 |

Tabellenverzeichnis

| | |
|--|----|
| 3.1. Auswertungsgeschwindigkeiten | 28 |
| 5.1. Protokoll für die Steuerbefehle (Ingo Busker (2010a)) | 50 |

Abbildungsverzeichnis

| | |
|---|----|
| 2.1. Beispiel eines Quadcoptermodells (Ansicht von oben) [Agner (2010)] | 15 |
| 2.2. Antriebsschema eines Quadcopter [Wikipedia (2006)] | 16 |
| 2.3. Quadcopter mit Digitalkamera [FH Bielefeld (2010)] | 18 |
| 3.1. Cricket Modul [MIT Computer Science and Artificial Intelligence Lab Cambridge, MA 02139 (2005a)] | 20 |
| 3.2. Cricket Beacons an der Decke montiert [MIT Computer Science and Artificial Intelligence Lab Cambridge, MA 02139 (2005b)] | 21 |
| 3.3. Ubisense Sender [ubisense (2007b)] | 22 |
| 3.4. Ubisense Location Engine [ubisense (2007c)] | 22 |
| 3.5. Paric Aufbau [Paric Measurement (2006b)] | 23 |
| 3.6. HiBall Beacon Array [3rdtech (2010b)] | 24 |
| 3.7. HiBall Sensor [3rdtech (2010c)] | 24 |
| 3.8. MagicMap Architektur [MagicMap (2010a)] | 26 |
| 3.9. MagicMap Karte [MagicMap (2010b)] | 26 |
| 3.10. Original Bild vor der Bearbeitung [Hahn (2008)] | 27 |
| 3.11. Bild nach der Binarisierung [Hahn (2008)] | 28 |
| 3.12. Versuchsaufbau [Thielicke (2009a)] | 29 |
| 3.13. Kamerabild mit identifizierten Leuchtmarken [Thielicke (2009a)] | 29 |
| 5.1. Schematischer Aufbau | 36 |
| 5.2. Quadcopter vor Umbau | 37 |
| 5.3. ARM7 Steuereinheit | 37 |
| 5.4. Quadcopter Schema | 38 |
| 5.5. Kamerasicht mit Infrarot Filter | 41 |
| 5.6. Softwarehierarchie | 42 |
| 5.7. Flussdiagramm der Software | 43 |
| 5.8. Berechnung der Entfernung zwischen zwei Punkten | 44 |
| 5.9. Bestimmung der Entfernung zur Kamera | 45 |
| 5.10. Aufsplitten der Entfernung in Nick- und Roll-Komponenten | 47 |
| 6.1. Quadcopter nach Umbau | 52 |
| 6.2. Philips Webcam an der Decke montiert | 53 |

| | |
|--|----|
| 6.3. 1. LED-Version mit 30° Abstrahlwinkel | 54 |
| 6.4. 2. LED-Version mit 60° Abstrahlwinkel | 54 |
| 6.5. Positionierung der LEDs | 54 |
| 6.6. Testboard für die Latenzzeitmessung | 57 |
| 7.1. 360° Kamera Q24M mit automatischer Entzerrung [MOBOTIX (2010b)] | 62 |

1. Einführung

1.1. Motivation

Die Positionsbestimmung von Objekten und Personen mittels GPS ist in der heutigen Zeit selbstverständlich geworden. Vor einigen Jahren war ein Navigationsgerät im Auto noch eine sehr teure Anschaffung und galt als etwas besonderes. Aufgrund der gestiegenen Nachfrage und großer Stückzahlen ist ein mobiles Navigationsgerät heutzutage für nahezu Jedermann erschwinglich geworden. Der Umgang mit dieser Technik ist für viele selbstverständlich geworden und Straßenkarten auf Papier geraten in Vergessenheit.

Die Gesellschaft befindet sich in einem ständigen Wandel auf der Suche nach neuer Technologie, welche das Leben vereinfacht. So ist es nicht nur im Freien interessant zu wissen, wo sich etwas befindet und wie man dort hin gelangt, sondern auch innerhalb von Gebäuden. Das GPS-System benötigt jedoch einen freien Blick zum Himmel (Satelliten). Bauwerke schwächen die Signale teilweise so stark ab, dass eine präzise Lokalisierung nicht mehr gegeben ist oder total ausfällt.

Basierend auf einer Positionsbestimmung innerhalb von Gebäuden, kann eine Navigationslösung geschaffen werden. Diese ist nicht nur für Personen interessant, die sich in einem Gebäude orientieren wollen, sondern auch für autonome Roboter, die ihren Weg durch Gänge und an Hindernissen vorbei finden müssen.

An selbstständig navigierende Modellfahrzeuge hat man sich inzwischen schon fast gewöhnt, doch für Veranstaltungen wie z.B. die „Nacht des Wissens“ [[Behörde für Wissenschaft und Forschung Hamburg \(2009\)](#)] wäre ein autonom fliegender Modellhubschrauber eine Besonderheit. Dieser könnte in einem Raum unterschiedliche Ziele anfliegen, ohne dass ein Pilot die Steuerung übernehmen muss. Doch nicht nur für Showzwecke ist ein Modellhubschrauber interessant, so wäre ein weiteres Anwendungsgebiet die Raumüberwachung.

Bei den Modellhubschraubern hat sich der Quadcopter als interessante Plattform für Experimente erwiesen. So ist dieser für sein besonders ruhiges Flugverhalten bekannt und ermöglicht es somit auch in Räumen zu fliegen. Dank seiner hohen Zuladungskapazität kann zusätzliche Elektronik wie z.B. eine Kamera montiert werden. Im kommerziellen und

auch militärischen Outdoorbereich werden Quadcopter schon länger als Aufklärungsdrohnen verwendet. [microdrones GmbH (2010)] Für die Zukunft wird der Einsatz dieser System in Gebäuden interessant. Hierfür ist es wichtig eine passende Indoornavigationslösung zu entwickeln.

1.2. Zielsetzung

Nachfolgend gilt es ein System zu finden, welches es ermöglicht, die Position eines Modellhubschraubers so präzise zu erfassen, dass dieser autonom im Raum bewegt werden kann. Hierfür ist eine Art der Positionsbestimmung nötig, wie sie im Outdoorbereich durch das GPS-System bekannt ist. Es gibt bereits einige Lösungsansätze und Realisierungen, die sich mit diesem Thema beschäftigen. Diese gilt es zu vergleichen und ein brauchbares Positionsbestimmungssystem zu wählen.

Nach der Auswahl des Systems wird dieses in einem Testaufbau realisiert. In einem weiteren Schritt soll, basierend auf den Positionsdaten, ein Quadcopter zunächst seine Position im Raum halten (Position-Hold). Die Steuerung wird hierbei von einem Computer übernommen. Eventuelle Abweichungen von der ursprünglichen Position sollen automatisch ausgeglichen werden.

Im letzten Schritt soll dann die Grundlage für eine Navigation geschaffen werden, indem ein Zielpunkt innerhalb des Flugbereiches des Testaufbaus angeflogen wird, welcher zuvor eingespeichert worden ist. Da es sich um eine kontrollierte Testumgebung handelt, in welcher keine Störfaktoren (Gegenstände) sind, wird eine Navigationslösung entwickelt, welche auf einer Punk-zu-Punkt-Navigation basiert.

Wie sich im Verlauf dieser Bachelorarbeit zeigen wird, fällt die Wahl auf die visuellen Positionsbestimmung. Mittels einer fest installierten Kamera an der Decke und einem PC mit einer Analysesoftware¹, wird der Quadcopter im Raum erkannt. LEDs dienen hierbei als eindeutige Markierung. Mit Hilfe der ermittelten Position werden automatisch Steuerbefehle an einen Quadcopter gesendet um diesen im Raum zu bewegen.

1.3. Gliederung

Die Arbeit ist in drei große Teilbereiche eingeteilt. Der erste Teil beschäftigt sich zunächst mit den benötigten Grundlagen. Anschließend folgt die Analyse der Aufgabenstellung. Nachdem

¹MATLAB [MathWorks (2010)]

verschiedene Positionsbestimmungsmöglichkeiten vorgestellt wurden, werden diese gegeneinander abgewogen und ein Positionierungsverfahren ausgewählt. Dieses wird dann im späteren Verlauf umgesetzt.

Der zweite Teil behandelt das Systemdesign. Dieses umfasst die Bereiche Hardware² und Software. Die benötigten Eigenschaften der technischen Umsetzung, der Bilderkennung und der Steuerung werden hier erörtert.

Der dritte Teil befasst sich mit der Realisierung der Arbeit. Eventuell auftretende Schwierigkeiten bei der Umsetzung im Bereich der Hard- und Software werden hier behandelt.

Abschließend wird ein Fazit gezogen, in dem die Arbeit nochmals analysiert wird. Dies beinhaltet mögliche Verbesserungsvorschläge und einen Ausblick in die Zukunft.

²Systemteile

2. Grundlagen

2.1. Positionsbestimmung

Eine Position ist die Lage eines Punktes im Raum. Hierfür gibt es verschiedene Bezugssysteme und Maßeinheiten.

Ein bekanntes Beispiel für eine Positionsbestimmung ist GPS. Es basiert auf der Ausbreitungsgeschwindigkeit von Funkwellen und der daraus resultierenden Entfernungsberechnung. So ist mittels der Trilateration eine Positionsbestimmung möglich. GPS liefert eine absolute Position, z.B. im WGS84-Koordinatenformat¹.

Bei der Positionsbestimmung in Räumen bietet es sich an, eine relative Position zu verwenden. Der Bezugspunkt ist meistens ein markanter Punkt, von dem aus die Position in X-, Y- und Z-Raumanteilen angegeben wird. Ein markanter Punkt wäre z.B. die Kamera, welche für die Bilderkennung zuständig ist. Die Kameraposition im Raum spielt dabei keine Rolle, da die Kamera das Bezugssystem ist und sich für die Berechnung nichts ändert. Anhand der Leuchtmarken am Quadcopter wird der Nullpunkt kalibriert, die Software berechnet ausgehend von diesem Punkt die entsprechenden Entfernungen und Winkel. Gleichzeitig ist dieser Nullpunkt das Ziel der Navigation, an welchem der Quadcopter stabilisiert werden soll.

Doch nicht nur die Position ist interessant, sondern auch die Ausrichtung des Gegenstandes, seine Geschwindigkeit oder Beschleunigung.

2.2. Navigation

In einem dreidimensionalen Raum ist die kürzeste Entfernung zwischen Start- und Zielpunkt somit eine gerade Linie. Dies ist der effektivste Weg um zum Ziel zu gelangen. Vorausgesetzt, dass sich auf diesem Weg keine Störgrößen wie z.B. Gegenstände befinden. In der kontrollierten Testumgebung dieser Arbeit können solche Bedingungen geschaffen werden. Solange der Quadcopter vom Ziel entfernt ist, werden Steuerbefehle gesendet, die ihn auf kürzestem Weg zum Bestimmungsort führen.

¹World Geodetic System 1984

2.3. Eigenschaften eines Quadcopters

Ein Quadcopter gehört wie ein Hubschrauber zu den Drehflüglern und besitzt sechs Freiheitsgrade. Diese setzen sich zusammen aus drei Translations- und drei Rotationsfreiheitsgraden. (Bittner, 2009, S.150ff)

2.3.1. Grundlegende Funktion

Der Auftrieb wird durch vier senkrecht zum Boden gerichtete Propeller erreicht. (Abb. 2.1) Zwei Rotoren drehen in Uhrzeigersinn, die beiden anderen entgegengesetzt. Sofern die



Abbildung 2.1.: Beispiel eines Quadcoptermodells (Ansicht von oben) [Agner (2010)]

Drehzahlen gleich sind, hebt sich die Kraft um die Hochachse auf. Der Giermoment ist ausgeglichen. Je nach Drehzahlsumme, steigt, sinkt oder hält der Quadcopter die Höhe. In folgender Grafik ist das Antriebsschema und die Drehrichtung der Rotoren eines Quadcopters abgebildet (Abb. 2.2)

Um eine Drehung (zukünftig „Gier“ genannt) zu erreichen, wird die Drehzahl eines gleich drehenden Rotorpaares erhöht, während das andere Paar langsamer dreht, die Summe des Schubes wird so konstant gehalten, die Höhe ändert sich nicht, jedoch entsteht ein Drehmoment um die Hochachse.

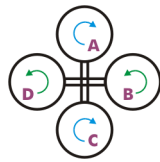


Abbildung 2.2.: Antriebsschema eines Quadcopter [Wikipedia (2006)]

Um eine Bewegung in X- oder Y-Richtung zu erzielen, wird die Propellerebene bzw. der gesamte Quadcopter verkippt (Rotation um Längs- oder Querachse). Dies erreicht man durch Drehzahländerung der sich gegenüberliegenden Rotorpaare.

Die Verkipfung um eine Vorwärts- bzw. Rückwärtsbewegung zu erhalten, nennt man „Nick“. Die seitliche Verkipfung wird „Roll“ genannt. Eine Veränderung der Lage aus der Horizontalen führt somit immer zu einer Beschleunigung in X- und Y-Richtung. [Ingo Busker (2010e)]

2.3.2. Besonderheiten

Da der Auftrieb eines Propellers über die Drehzahl gesteuert wird, und nicht wie bei einem Hubschrauber über den Anstellwinkel der Rotorblätter (welcher auch negativ werden kann), ist es bei einem Quadcopter praktisch nicht möglich, aus dem Flug mittels eines halben Loopings auf dem Kopf weiter zu fliegen, da die Propeller ihre Drehrichtung schlagartig ändern müssten. Aufgrund des herrschenden Drehimpulses würde dies einige Zeit in Anspruch nehmen. In dieser Zeit befindet sich der Quadcopter schon im freien Fall und müsste diese Beschleunigung zusätzlich abfangen. Des Weiteren wird ein Rückwärtslauf der Motoren von den meisten Motorreglern - die für Quadcopter verwendet werden - nicht unterstützt. Loopings sind hingegen möglich, wobei diese eher einem Überschlagen einer Achse entsprechen.

Bei einem Flug in Bodennähe ist es wichtig die entstehenden Turbulenzen zu beachten. Durch die Abwinde der Rotorblätter kommt es in Bodennähe zum sogenannten Bodeneffekt, dies äußert sich durch ein sehr instabiles Schwebeverhalten. Der Quadcopter driftet durch die entstehenden Turbulenzen in beliebige Nick- und Roll-Richtung. Um diesen Effekt zu minimieren, sollte man einen gewissen Abstand zum Boden² überschreiten. Ebenso sollte davon abgesehen werden, nahe an Wände zu fliegen, da es neben „normalen“ Turbulenzen auch zu einem Strömungsabriss beim - der Wand zugewandten - Rotor führen kann.

²Erfahrungswerte zeigen, dass eine Flughöhe von 50cm ausreichend sind.

2.3.3. Fluglageregelung

Ein Drehflügler ist ein instabiles System, das ständiger Steuereingriffe bedarf, um eine Position im Raum zu halten. Wie oben schon erwähnt, reichen schon kleinste Abweichungen aus der horizontalen Rotorebene, um eine Beschleunigung zu verursachen. Dies kann z.B. durch Wind oder Drehzahlunterschiede kleiner der Regelauflösung passieren. Ist der Quadcopter erst einmal verkippt, so kehrt er nicht von alleine in die ursprüngliche Lage zurück, die Beschleunigung hält somit an.

Ist bei einem Hubschrauber der Ausgleich dieser Störfaktoren komplett dem Piloten überlassen, so regelt bei einem Quadcopter eine Elektronik diese nahezu vollständig aus. Um einen Quadcopter zu stabilisieren benötigt man Sensoren, die Auskunft über die aktuelle Ausrichtung und Rotationsgeschwindigkeit geben.

Jeweils ein Gyroskop misst die Rotationsänderung um die drei Achsen. Mit diesem Sensor kann somit die Winkelgeschwindigkeit gemessen werden und eine Rotation um die Achsen kann dementsprechend gestoppt werden.

Da das Stoppen der Rotationsbewegung noch nicht ausreicht um dem Quadcopter stabil an einer Position zu halten (es wird eine waagerechte Lage der Rotoren benötigt), ist es nötig, auch die absolute Ausrichtung bezüglich der Horizontalen zu ermitteln. Mittels eines kalibrierten Beschleunigungssensors kann die absolute Verkippung um die drei Achsen bestimmt werden, indem die Erdbeschleunigung als Referenz benutzt wird. Zusätzliche Beschleunigungen werden ebenfalls gemessen, in der Steuerelektronik ausgewertet und diesen gegebenenfalls entgegen geregelt. [Ingo Busker (2010f)]

Bei einer Abweichung wird somit automatisch die Rotordrehzahl entsprechend geändert, um den Quadcopter wieder in die (kalibrierte) Horizontale zu bringen. Je nach Softwareeinstellung kann auch der beim Flug aufgenommenen Geschwindigkeit entgegen gesteuert werden. Somit ist im idealen Fall ein Schweben auf der Stelle möglich. In der Praxis ist dies jedoch selten der Fall, da die Gyroskope und Beschleunigungssensoren nur eine endliche Auflösung haben. Hinzu kommt, dass weder ein Gyroskop noch ein Beschleunigungssensor eine Geschwindigkeit relativ zum Boden misst. Bewegt sich der Quadcopter schon mit konstanter Geschwindigkeit, so kann nur durch Integration (über die Zeit) der vorhergegangenen Beschleunigungen auf die Geschwindigkeit relativ zum Boden geschlossen werden. Hierbei sorgen kleinste Messfehler mit der Zeit für große Abweichungen, was ein Drift des Quadcopters zur Folge hat.

2.3.4. Mögliche Erweiterungen

Mit zusätzlichen Sensoren, wie z.B. GPS kann diesem Effekt entgegen gewirkt werden. Das GPS bestimmt eine absolute Position bezüglich der Erde. Eine - von den internen Sensoren unbemerkte - relative Geschwindigkeit zum Boden wird nun durch die Änderung der GPS-Position bemerkt und kann (der Messgenauigkeit entsprechend) entgegengeregelt werden. Somit kann eine Position gehalten werden, ohne dass ein Eingreifen des Piloten notwendig ist. Der in dieser Arbeit verwendete Quadcopter besitzt kein GPS-Modul, da dieses in geschlossenen Räumen nicht ausreichend präzise arbeitet. (Näheres siehe Kapitel 4.4.1)

Da das GPS-System keine besonders gute Auflösung der Höhe liefert, kann ein zusätzlicher Luftdrucksensor verwendet werden. Abhängig von der Höhenlage ändert sich der Luftdruck. Je nach Auflösung des Sensors kann eine Änderung der Höhe von ca. 5cm registriert werden. [Ingo Busker (2010c)] Bei dem verwendeten Quadcopter ist zwar dieser Höhensensor verbaut, wird jedoch nicht für die Höhenregelung verwendet.

Dank der hohen Zuladungskapazität kann auch eine Digitalkamera am Quadcopter montiert werden. (Abb. 2.3) Durch die Fernsteuerung können dann jeweils Einzelaufnahmen ausgelöst werden. Dieses System ist für Fotoaufnahmen aus der Vogelperspektive gedacht.



Abbildung 2.3.: Quadcopter mit Digitalkamera [FH Bielefeld (2010)]

Eine weitere Möglichkeit besteht darin, eine Videokamera zu montieren, welche ihre Aufnahmen per Funksignal an eine Bodenstation sendet. Mit einer entsprechenden Videobrille ist damit ein sogenannter FPV-Flug möglich. D.h. der Pilot steuert den Quadcopter aus der Ich-Perspektive, als wäre er an Bord der Quadcopters. Auf der Webseite von „RCMovie.de“ wird in einem Video, ein Immersionsflug mit einem Quadcopter dargestellt. [Kilrah (2008)] Mit diesem System ist eine viel natürlichere Flugsteuerung möglich, da man sozusagen „im Quadcopter sitzt“. Eine direkte Sicht zum Quadcopter ist nicht nötig und erweitert die Einsatzmöglichkeiten.

3. Indoor Positionsbestimmung

Da für die Indoornavigation zunächst eine genaue Positionsbestimmung nötig ist, muss hierfür ein passendes System gefunden werden. In folgendem Kapitel werden verschiedene, bereits existierende Lokalisierungsmethoden für geschlossene Räume vorgestellt. In einem späteren Schritt (4) werden die Systeme analysiert, inwieweit sie für die Positionsbestimmung des Quadcopters in Frage kommen.

3.1. GPS – Global Positioning System

Das GPS-Netzwerk benötigt 21 Satelliten (2008 befanden sich ca. 30 aktiven Satelliten in der Umlaufbahn, um eine Redundanz zu gewährleisten).[kowoma (2008)] Für eine Positionsbestimmung sind jedoch vier Satelliten ausreichend. An Bord jedes Satelliten befindet sich eine sehr präzise Atomuhr. Das Zeitsignal und andere Informationen werden per Funk ausgestrahlt. Mit einem Empfänger werden mehrere dieser Signale empfangen. Die Zeitdifferenz zwischen Aussenden und Empfang lässt Rückschlüsse auf die sogenannte (Pseudo-)Entfernung¹ zu. Mittels der Trilateration wären drei Satelliten mit dazugehörigen Entfernungen ausreichend für eine Positionsbestimmung. Dies würde aber eine sehr präzise Uhr im Empfänger voraussetzen. Um dieses Problem zu umgehen, wird ein vierter Satellit hinzugezogen, um den Uhrenfehler im Empfänger zu bestimmen und auf die korrekte Position zu schließen. Um so mehr Satellitensignale empfangen werden können, um so besser ist die Empfangsgenauigkeit, da das Gleichungssystem überbestimmt ist und man z.B. mittels Kalman-Filterung² fehlerbehaftete Messgrößen verbessern kann.

Handelsübliche Handempfänger, wie z.B. das Garmin Vista HCx haben eine Auflösung im Meterbereich (<3m).[Garmin (2007)] Der Anschaffungspreis von ca. 180 € ist erschwinglich.

An Stellen, an denen die direkte Sicht zu den Satelliten nicht möglich ist und die Feldstärke abgeschwächt wird, werden Empfänger empfohlen, die eine hohe Empfangsempfindlichkeit

¹ Nicht synchron laufende Uhren zwischen Sender und Empfänger führen zu einer falschen Berechnung der Entfernung

² vgl. [Ramachandra (2000)]

haben. SuperSense Empfänger sollen laut Hersteller auch in geschlossenen Räumen noch die Position ermitteln können. [u-blox (2010)]

Für den professionellen Bereich (z.B. Bauvermessung) gibt es Systeme die dank zusätzlicher Technik, eine Auflösung im Zentimeterbereich erreichen [fischinger-messen (2010)]. Die Preise steigen hier aber sofort in astronomische Höhen³.

3.2. Cricket/IMAPS

Cricket wurde 2005 vom „MIT Computer Science and Artificial Intelligence Lab“ in Cambridge entwickelt und ist für die Positionsbestimmung im Raum konzipiert. Die Position wird mittels Raumkoordinaten im kartesischen System ausgegeben. Das System hat laut Hersteller eine Auflösung von 1-3cm und besteht aus Sender und Empfänger, die in eine Einheit kombiniert sind (Abb. 3.1). Mehrere Einheiten werden als aktive Transmitter geschaltet (sogenannte Beacons) und an der Decke befestigt (Abb. 3.2). Eine Einheit wird als Empfänger benutzt (sogenannter Listener oder auch Host) und wertet die von den Beacons ausgesendeten Daten aus. Der Empfänger ist auf dem zu lokalisierenden Objekt befestigt und bewegt sich mit diesem durch den Raum. Periodisch sendet jeder Beacon seine ID und Position per Funk. Gleichzeitig wird ein Ultraschall-Signal ausgesandt. Listener, die Sichtkontakt zum jeweiligen Beacon haben und in Reichweite der Schallwellen sind, können beide Signale empfangen und auswerten.

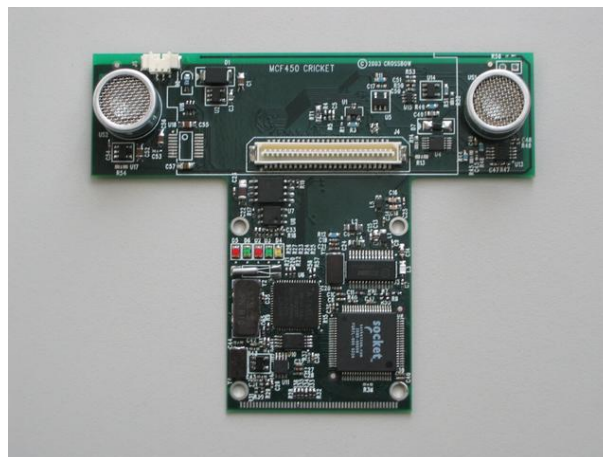


Abbildung 3.1.: Cricket Modul [MIT Computer Science and Artificial Intelligence Lab Cambridge, MA 02139 (2005a)]

³Leica Komplettsystem GPS900 Basis + Rover für 19.990,00 € [grad-gon (2010)]



Abbildung 3.2.: Cricket Beacons an der Decke montiert [MIT Computer Science and Artificial Intelligence Lab Cambridge, MA 02139 (2005b)]

Aufgrund der unterschiedlichen Ausbreitungsgeschwindigkeit von Funkwellen und Schallwellen, kann der Listener beim Empfang des Funksignals eine Zeitmessung starten, beim Empfang der Schallwelle und mit Hilfe der bekannten Ausbreitungsgeschwindigkeit von Schall, die Entfernung des Senders berechnen. Mit mehreren dieser Entfernungen kann, ähnlich wie bei GPS [3.1], auf die Position geschlossen werden.[MIT Computer Science and Artificial Intelligence Lab Cambridge, MA 02139 (2005c)] Ein ähnliches Time-of-Arrival-System⁴ ist IMAPS⁵, welches an der HAW bereits verwendet wird. [Gregor (2006)]

3.3. Ubisense

Ubisense wurde 2002 gegründet. Die britische Firma, hat sich auf Echtzeit-Ortungssysteme spezialisiert und gilt als Marktführer auf diesem Gebiet. Das Unternehmen beschreibt sich selbst als „nächste Generation des computergestützten Gebäudemanagements (CAFM, Computer Aided Facility Management)“ [ubisense (2010)] Ihr Produkt „Series 7000 System“ basiert auf aktiven Sende- und Empfängermodulen (Tags und Sensoren). Die batteriebetriebenen Sender (Maße: 83mm x 42mm x 11mm) wiegen gerade mal 32 Gramm und senden in einstellbaren Intervallen, zwischen 0,01 – 20Hz, ein Funksignal auf dem UWB (Im Bereich von 6-8GHz) aus. (Abb. 3.3) [ubisense (2007b)]

Mehrere Sensoren werden in sogenannten Cells zusammengefasst, wobei ein Sensor die Steuerung und Koordination der restlichen Sensoren übernimmt. Die Funksignale des Senders werden von den Sensoren empfangen und daraus die Position bestimmt. Dies erfolgt

⁴siehe [(Pfaff, 2007, S. 19)]

⁵„indoor distance measurement and positioning system“ [(Gregor, 2006, S. 30)]



Abbildung 3.3.: Ubisense Sender [ubisense (2007b)]

mittels AOA⁶, TDOA⁷ und anderen Bestimmungsmethoden, auf die der Hersteller nicht weiter eingeht. [ubisense (2007a)] Die berechnete Position wird über eine Ethernetverbindung zur "Location Engine" gesendet, welche diese wiederum gesammelt zur Verfügung stellt⁸ (Abb. 3.4). Das System hat eine Latenzzeit von 100ms und einer Lokalisierungspräzision von bis zu 15cm. Es ist nahezu unbegrenzt skalierbar und auf die Erfassung und Verwaltung von vielen Tags gleichzeitig ausgelegt. [ubisense (2007c)]

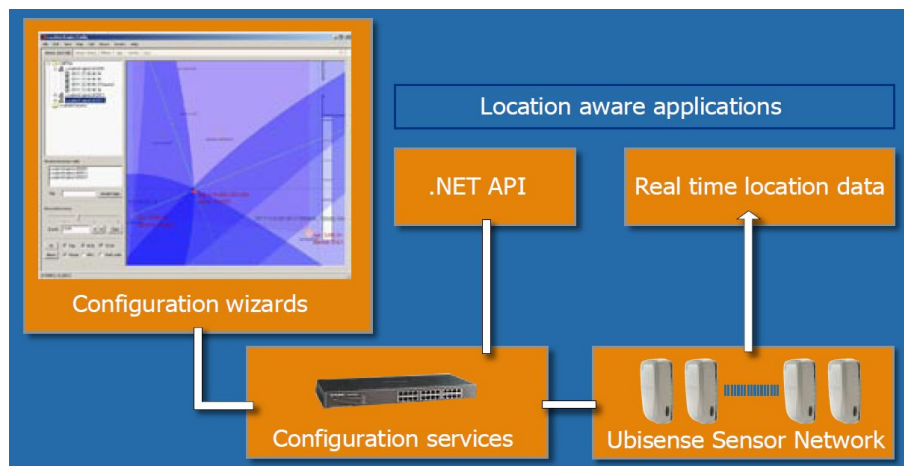


Abbildung 3.4.: Ubisense Location Engine [ubisense (2007c)]

⁶Angle of Arrival – Einfallswinkel der Funksignale wird ausgewertet

⁷Time Difference of Arrival – Die unterschiedlichen Signallaufzeiten werden ausgewertet

⁸.NET 2.0 API

3.4. Paric

Die neuseeländische Firma Paric wurde 2001 gegründet und bietet ein Trackingsystem für den Indoorbereich an. Ähnlich wie Ubisense (3.3), besteht dieses System aus aktiven Sendern (Transmitter Tag) und mehreren Empfängern.

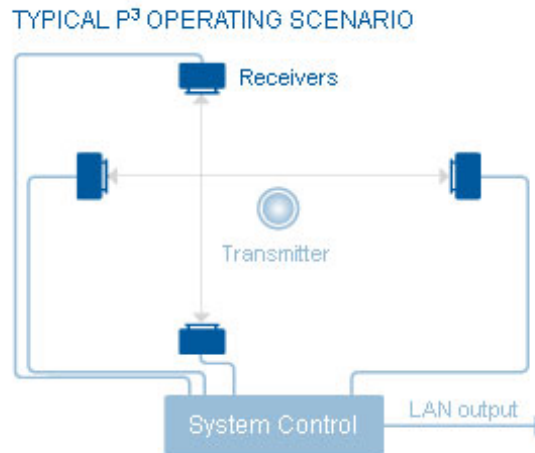


Abbildung 3.5.: Paric Aufbau [Paric Measurement (2006b)]

Um einen Raum abzudecken, werden mindestens 4 Empfänger benötigt⁹. Jeweils zwei sind genau auf der gegenüber liegenden Seite angebracht (Abb. 3.5). Dieser Aufbau und eine spezielle Software ermöglicht eine besonders hohe Lokalisierungsgenauigkeit von bis zu 10mm, da Reflexionen ausgefiltert werden können¹⁰. Das System ist für Räume kleiner als 200m² ausgelegt. Die Aktualisierungsrate beträgt 150Hz. [Paric Measurement (2006a)]

3.5. HiBall

Die Firma 3rdTech bietet mit ihrem Produkt „HiBall“ eine visuelle Positionsbestimmung an. An der Decke wird ein 2D Beacon Array bestehend aus Infrarot-LEDs installiert. (Abb. 3.6)

An der zu bestimmenden Position ist der sogenannte Scanner montiert. Dieser besteht aus 6 Linsen mit Fotodioden (Abb. 3.7), die es ihm ermöglichen die Beacons an der Decke zu lokalisieren. Anhand der berechenbaren Winkel zwischen HiBall und der bekannten Position der LEDs, kann auf die Position geschlossen werden¹¹.

⁹Aus Gründen der Redundanz, werden in der Praxis jedoch mehr als 4 Empfänger verwendet

¹⁰Nur das Signal der direkten Sichtlinie wird verwendet. Andere Mehrfachreflexionen werden ignoriert.

¹¹vgl. Triangulation und Multilateration



Abbildung 3.6.: HiBall Beacon Array [3rdtech (2010b)]



Abbildung 3.7.: HiBall Sensor [3rdtech (2010c)]

Laut Hersteller bietet das System folgende Eigenschaften:

- Gewicht: 6 OZ (ca. 170 Gramm)
- Skalierbar von 144 sq. ft. bis 1,600 sq. ft. (ca. 45qm bis ca. 500qm)
- Präzise (Ortsauflösung von 0,2mm)
- Unempfindlich gegenüber Metall, Magnetfeldern oder Geräuschen
- Hohe Aktualisierungsrate mit sehr kurzer Verzögerung (2kHz)
- Automatische Kalibrierung (keine präzise Ausrichtung der Beacons nötig) [3rdtech (2006)]

Dies macht das System zu einem der schnellsten und präzisesten auf dem Markt. [3rdtech (2010a)]

3.6. WLAN

In folgendem Kapitel werden zwei Positionsbestimmungsmethoden vorgestellt, die auf WLAN-Infrastrukturen basieren.

3.6.1. RADAR

Im Jahr 2000 entwickelte das Microsoft Research Lab das RADAR System. Anhand der Signalstärke verschiedener WLAN Basisstationen wird die Position ermittelt. Das System kann eine bestehende WLAN Netzwerksstruktur nutzen, muss jedoch „angelern“ werden. Hierfür müssen an zufällig gewählten Stellen, die jeweiligen Signalstärken gemessen werden. Diese Information und die zugehörige Position wird gespeichert. Je mehr Messungen erfolgen, um so mehr Vergleichswerte hat man für eine spätere Lokalisierung. Nach der Kalibrierung kann das System genutzt werden, indem das mobile Gerät die jeweiligen Signalstärken zu den Basisstationen misst und diese an das RADAR System sendet. Dieses vergleicht die Werte wiederum mit den zuvor gespeicherten Referenzwerten. Aus der besten Übereinstimmung wird dann die ungefähre Position abgeleitet. Die Präzision der Lokalisierung liegt zwischen 3-4m. [[Paramvir Bahl \(2000\)](#)]

3.6.2. MagicMap

MagicMaps ist eine Entwicklung der Humboldt-Universität zu Berlin. Die Software wertet die Signalstärke drahtloser Kommunikationssysteme aus (z.B. WLAN Access Points, Bluetooth, o.a) und bestimmt mittels Trilateration die Position.

Dieses System besteht aus mehreren Komponenten. (Abb. 3.8) Der sog. „Stumbler“ misst die Signalstärken der jeweiligen Funkinfrastruktur. Ein Server stellt die gesammelten Daten zur Verfügung. Eine P2P Kommunikation erlaubt den Datenaustausch, auch wenn kein Server zur Verfügung steht. Die „Positioning Engine“ ist für die Berechnung der Position verantwortlich. Dies kann beim Client stattfinden, oder verteilt im System (z.B. Server mit stärkerer Rechenleistung). Der Tracker reagiert auf bestimmte Ereignisse (z.B. eine berechnete Position liegt innerhalb eines festgelegten Bereichs oder das Bewegungsprofil entspricht einem besonderen Muster). In einer GUI können die berechneten Positionen auf einem Grundriss der Karte angezeigt werden. (Abb. 3.9) [[MagicMap \(2010c\)](#)]

Die Genauigkeit ist sehr stark von der jeweiligen Systemarchitektur abhängig. Nach der Einmessung und Kalibrierung sind in einem Raum mit Signalhindernissen, Ortsgenauigkeiten von bis zu 3m realisierbar. [[MagicMap \(2008\)](#)]

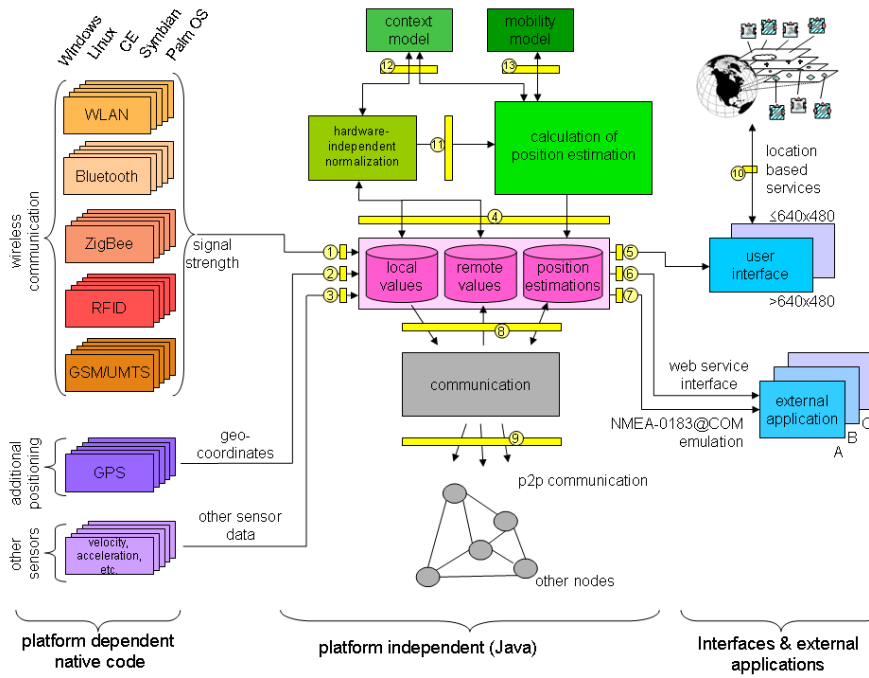


Abbildung 3.8.: MagicMap Architektur [MagicMap (2010a)]

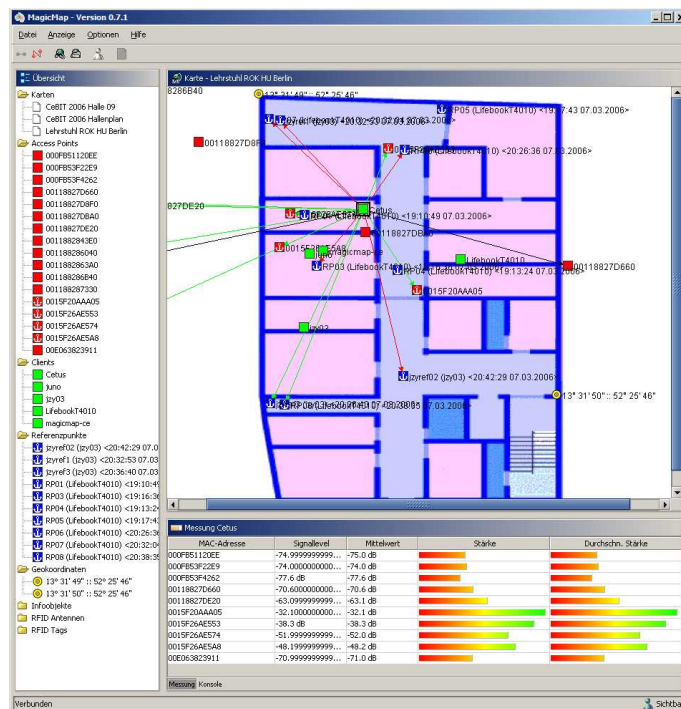


Abbildung 3.9.: MagicMap Karte [MagicMap (2010b)]

3.7. Bilderkennung

3.7.1. Kamerabasierten Erkennung von frequenzcodierten Infrarotsendern

Tobias Hahn hat in seiner Bachelorarbeit, die Erkennung von frequenzcodierten Infrarotsendern behandelt. Ein mobiler Tag (Abb. 3.10) sendet mittels einer gepulsten IR-Diode seine Identifikationsnummer (ID) aus.

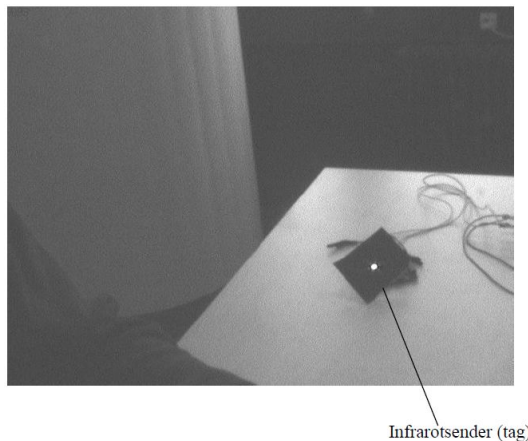


Abbildung 3.10.: Original Bild vor der Bearbeitung [Hahn (2008)]

Eine fest installierte Kamera erfasst die mobilen IR-Sender und ein PC wertet die Bilder aus, um die Tags zu identifizieren. Als Ergebnis erhält man die Bildkoordinaten der erkannten Tags. Hierbei handelt es sich zunächst nur um eine 2D Positionsbestimmung. Erst durch mehrere Tags, welche in einem definierten Abstand zueinander liegen, kann auf die Entfernung zur Kamera geschlossen werden.

Je nach Codelänge der ID, können mehrere Punkte erkannt und eindeutig zugeordnet werden¹². Um eventuelles Rauschen der Kamera zu minimieren, sollte ein Tiefpassfilter auf das Bild angewendet werden. Die Bearbeitungsgeschwindigkeit wird erhöht, indem das Bild entsprechend eines Schwellwertes binarisiert wird. (Abb. 3.11)

Durch zusätzliche Auswertung der Bilder über die Zeit, wird das Blinkmuster registriert. Somit können die statischen Störungen (wie z.B. das Sonnenlicht im Bild) ausgefiltert werden und nur die Tags werden erkannt.

¹²In diesem Fall ist der Code 6 Pulse lang, daraus resultiert eine maximale Anzahl von 64 möglichen Kombinationen und somit eindeutigen IDs

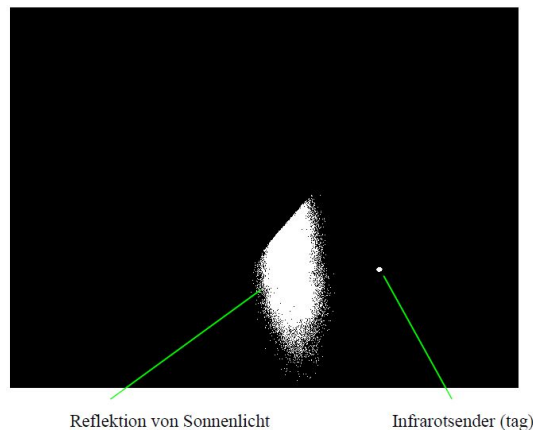


Abbildung 3.11.: Bild nach der Binarisierung [Hahn (2008)]

Bei einer Bildwiederholrate von 30 fps entspricht die verfügbare Auswertungsgeschwindigkeit ca. 33ms. Die von der Software benötigte Bearbeitungszeit dauert maximal 5,0ms (siehe Tabelle 3.1) und ist somit ausreichend schnell.[Hahn (2008)]

| Funktion | Zeiten ohne Sender | Zeiten mit 2 Sendern |
|------------------|--------------------|----------------------|
| filterBackground | 1,4ms-1,5ms | 1,4ms-1,5ms |
| detectBlobs | 4,9ms-5,0ms | 4,9ms-5,0ms |

Tabelle 3.1.: Auswertungsgeschwindigkeiten

3.7.2. Tracking mittels einer stationären Kamera

Ein weiteres Projekt der optischen Positionserkennung, von William Thielicke [Thielicke (2010a)] zeigt die Möglichkeit, ein mit Leuchtmarkern versehenes Objekt, mittels einer fest installierten Kamera zu identifizieren und die Position zu berechnen. (Abb. 3.12)

In seinem Video [Willa1980 (2009)] wird veranschaulicht, wie diese Positionsbestimmung genutzt wird, um einen Tricopter zu stabilisieren und anschließend zu navigieren. (Abb. 3.13) Die Bilderkennung findet mit Hilfe von MATLAB¹³, auf einem externen Rechner statt. Von dort werden die Steuersignale via Bluetooth an den Tricopter gesendet.

¹³Die durchschnittliche Auswertungsgeschwindigkeit beträgt 12Hz

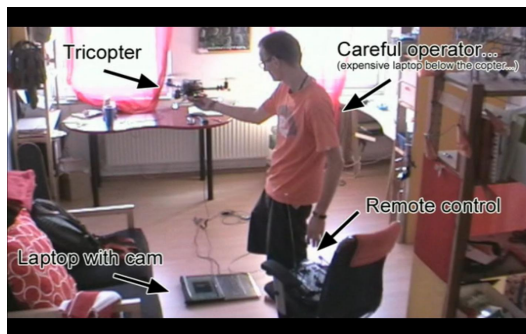


Abbildung 3.12.: Versuchsaufbau [Thielicke (2009a)]

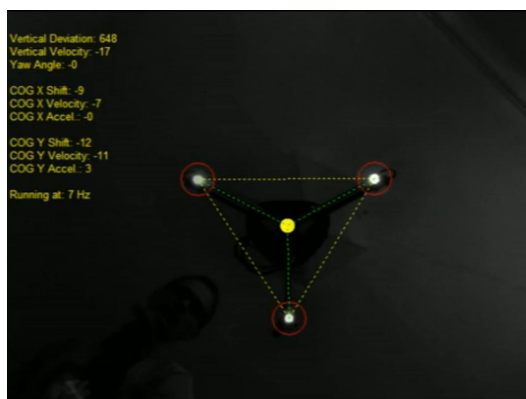


Abbildung 3.13.: Kamerabild mit identifizierten Leuchtmarken [Thielicke (2009a)]

3.8. Zusammenfassung

In diesem Kapitel wurden mehrere Positionsbestimmungsmethoden vorgestellt. Diese basieren auf unterschiedlichen Techniken und reichen von der akustischen Ortung, über Triangulation von Funksignalen, bis hin zur optischen Erkennung. Dementsprechend vielfältig ist die jeweils benötigte Hardware. Diese hat wiederum ihre Vor- und Nachteile in einzelnen Bereichen. Im nächsten Kapitel (4) werden deshalb die Positionsbestimmungsmethoden, auf ihre Tauglichkeit für das Projekt, analysiert.

4. Analyse

In folgendem Kapitel werden zunächst die grundlegenden Anforderungen an das System erläutert, gefolgt von den begrenzenden Faktoren. Anschließend werden die Positionsbestimmungsverfahren aus Kapitel 3 analysiert und auf Tauglichkeit, sowie Realisierbarkeit geprüft. Daraufhin wird eine Entscheidung für ein Positionsbestimmungsverfahren getroffen, welches davon am besten geeignet scheint, einen Quadcopter in einem Raum zu lokalisieren und die Positionsdaten nutzbar zur Verfügung zu stellen.

4.1. Aufgabenstellung

Wie in der Einleitung schon angedeutet, ist das Ziel dieser Arbeit, einen Quadcopter im Raum fliegen zu lassen. Die Steuerung soll hierbei nicht einem Menschen überlassen werden, sondern von einer Software übernommen werden. Dieser Software obliegt die Aufgabe, den Quadcopter auf Position zu halten und eventuelle Abweichungen selbstständig auszugleichen. Hierfür wird ein Positionsbestimmungsverfahren benötigt, welches periodisch die aktuelle Position des Quadcopters an die Software meldet. Zunächst soll der Quadcopter im Raum seine Position halten. In einem späteren Schritt soll dann auch eine Punkt-zu-Punkt Navigation möglich sein.

4.2. Grundlegende Anforderungen an das System

Da sich der Quadcopter in einem geschlossenen Raum bewegen soll, ist es zwingend erforderlich, dass die Positionsbestimmung dort jederzeit einwandfrei möglich ist. Sollte die Erkennung auch nur kurzzeitig ausfallen, besteht die Gefahr, dass sich die Position unkontrolliert verändert und die Rotoren mit Gegenständen (z.B. der Wand) in Berührung kommen. Ein Absturz ist dann unvermeidbar und Personenschaden kann nicht ausgeschlossen werden. Für die Lageregelung ist es entscheidend, wie schnell in Folge, und vor allem, wie präzise die Position des Quadcopters ermittelt werden kann. Umso schneller eine Abweichung vom Sollwert registriert wird, umso eher und feinfühlicher kann dieser entgegen geregelt werden.

4.3. Begrenzende Faktoren

Bei der Auswahl eines geeigneten Positionsbestimmungsverfahrens darf jedoch nicht nur das Augenmerk auf der Präzision und Geschwindigkeit liegen, sondern auch auf Faktoren, die das Projekt unter Umständen negativ beeinflussen können, oder sogar unvereinbar machen.

Die Erkennungspräzision und Aktualisierungsrate der aktuellen Position muss im Zentimeterbereich liegen und sollte 12Hz nicht unterschreiten. Dies basiert auf den Erfahrungen von Thomas Wetzel ([Wetzel, 2007](#), S.86) und William Thielicke [[Thielicke \(2010a\)](#)], bezüglich der Lagestabilisierung eines Modellhubschraubers.

Der Quadcopter hat eine sehr begrenzte Zuladungskapazität. Schon bei einem zusätzlichen Gewicht von ca. 250 Gramm wird das Flugverhalten merkbar schwerfällig, denn die interne Lageregelung hat weniger Spielraum für die Drehzahlregelung¹. Ein hoher Energiebedarf ist ebenso nachteilhaft, da dies die Flugzeit des Quadcopters reduziert. Daher ist es nötig, bei der Auswahl des passenden Systems, auf das Gewicht und den Stromverbrauch des mobilen Tags zu achten.

Auch wenn die Quellcodes der internen Steuerung des Quadcopters verfügbar sind und es daher möglich ist, Code hinzuzufügen, der die Positionsdaten auf dem Mikroprozessor auswertet, sollte man für die Berechnung nicht zu viel Rechenzeit verbrauchen, da diese an anderer Stelle fehlen könnte und eine rechtzeitige Auswertung der Sensordaten verhindern kann.

4.4. Abwägung der verschiedenen Positionsbestimmungsverfahren

In folgendem Abschnitt werden die zuvor in Kapitel 3 vorgestellten Positionsbestimmungsverfahren auf ihre Tauglichkeit für das Projekt untersucht.

4.4.1. GPS

Die Empfangsgenauigkeit von GPS liegt bei ca. 10m. Mit zusätzlichen Hilfsmitteln wie DGPS werden Fehlerfaktoren wie z.B. Störungen in der Ionosphäre, Uhr- und Bahnfehler der Satel-

¹Durch das Zusatzgewicht müssen die Rotoren insgesamt schneller drehen, um genügend Auftrieb zu erzeugen

liten korrigiert. Damit lässt sich die Genauigkeit auf den Dezimeterbereich verbessern. Eine Schwankung im Dezimeterbereich ist für die Indoornavigation jedoch nicht optimal.

GPS benötigt eine freie Sicht zum Himmel. Mit leistungsfähigen Empfängern kann man auch an abgeschatteten Stellen noch Empfang haben (im Wald oder in Städten). Bei einem geschlossenen Raum ist es aber nicht mehr möglich, die Signale ohne Reflexionen zu empfangen. Dies wirkt sich negativ auf die Genauigkeit aus und verhindert eine sinnvolle Auswertung.

Erschwerend kommt die Aktualisierungsrate hinzu. Eine Erfassung der Position erfolgt sekundlich und ist für einen fliegenden, instabilen Körper viel zu langsam. GPS fällt für diesen Anwendungsbereich demzufolge völlig aus.

4.4.2. Cricket/IMAPS

Dieses System sieht zunächst vielversprechend aus. Es wurde speziell für die Positionsbestimmung in Räumen entwickelt. Eine ähnliche Bachelorarbeit hat dieses System verwendet, um einen Modellhubschrauber im Raum zu navigieren. Es war jedoch nicht möglich, mit einer nutzbaren Frequenz von 3Hz und einer Genauigkeit von 1cm, eine stabile Fluglage herzustellen. ([Welzel, 2007](#), S.86)

Aufgrund der auftretenden Schwingungen beim Flug, wurden die Ultraschallmesswerte zu sehr verfälscht. Ebenso führten Steuerimpulse dazu, dass die Sensoren aus der – für die Messung notwendigen - horizontalen Lage bewegt wurden. Es konnte keine ausreichende Präzision, bei entsprechend hoher Abtastrate erreicht werden.

4.4.3. Ubisense

Da dieses System schon an der HAW verbaut ist, gibt es bereits praktische Erfahrungen. Die von dem Hersteller erwähnte Präzision von 15cm ist nur unter optimalen Bedingungen möglich und erfordert eine aufwändige Kalibrierung, welche nur von Fachpersonal durchgeführt werden kann. Die tatsächliche Genauigkeit des Systems an der HAW liegt bei ca. 40cm. Sollte eine Kalibrierung durchgeführt werden und die vom Hersteller genannten 15cm erreicht werden, so ist dies für die Positionsbestimmung des Quadcopters noch immer viel zu ungenau.

4.4.4. Paric

Auch bei diesem System ist eine genaue Kalibrierung nötig, um die versprochene Genauigkeit zu erzielen.

Da dieses System an der HAW nicht verbaut ist, müsste es angeschafft und installiert werden. Dies scheidet jedoch aus Gründen der Wirtschaftlichkeit aus.

4.4.5. HighBall

Dieses System hat eine phänomenale Genauigkeit und noch dazu eine hohe Abtastrate. Jedoch ist der Anschaffungspreis viel zu hoch. Das System ist fertig entwickelt und wäre ähnlich wie ein GPS Empfänger, eine Blackbox, welche die Positionsdaten liefert. Die Eigenleistung der Bachelorarbeit würde sich somit lediglich auf die Installation des Systems und der Programmierung der Navigationssoftware beschränken.

4.4.6. WLAN

Ähnlich wie GPS, besitzt die Positionsbestimmung per WLAN-Signalstärkenauswertung nicht die benötigte Genauigkeit. Da die Signalstärke nicht nur von der Entfernung zu den einzelnen Sendern abhängt, sondern auch von anderen Störfaktoren wie z.B. Wänden, ist die Lokalisierung nicht an allen Stellen des Raumes gewährleistet und nicht linear.

Weder RADAR (3.6.1) noch MagicMaps (3.6.2) schaffen es durch ihre Verfahren, eine brauchbare Ortsauflösung bei hoher Abtastrate zur Verfügung zu stellen.

4.4.7. Bilderkennung

Die Positionsbestimmung mittels Bilderkennung hat einen sehr eingeschränkten Aktionsradius, der sich auf den Sichtbereich der Kamera beschränkt. Um eine akzeptable Genauigkeit zu gewährleisten, muss die Auflösung der Kamera hoch genug sein. Bei entsprechender Bildwiederholfrequenz ist somit eine stabile Fluglagensteuerung möglich. Besonders vorteilhaft ist die günstige Hardware. Eine Webcam ist meistens vorhanden, ansonsten für weniger als 50€ zu bekommen. Die LEDs sind ebenfalls günstig in der Anschaffung, besitzen ein äußerst geringes Gewicht und haben einen relativ niedrigen Stromverbrauch. Eine aufwändige Kalibrierung entfällt hierbei ebenfalls.

Kamerabasierten Erkennung von frequenzcodierten Infrarotsendern

Dieses System hat eine ausreichend hohe Abtastrate. Es wäre aber nicht ausreichend, nur eine LED an den Quadcopter zu befestigen, da so die Ausrichtung nicht erkannt werden kann. Da die einzelnen Leuchtmarken frequenzcodiert sind, und somit eindeutig identifizierbar sind, würden zwei LEDs ausreichend sein um die Position und Ausrichtung zu erkennen.

Tracking mittels einer stationären Kamera

Wie das Projekt der optischen Positionserkennung von William Thielicke gezeigt hat, ist es möglich, einen Tricopter, mittels Bilderkennung zu lokalisieren und zu steuern. Die Auswertung und Regelung erfolgt durch MATLAB, welches eine gute Dokumentation besitzt.

4.5. Entscheidung für ein Positionsbestimmungsverfahren

Die meisten Indoor Positionsbestimmungsverfahren sind für diese Arbeit nicht geeignet, da sie viel zu ungenau sind und somit den Anforderungen nicht gerecht werden.

GPS und WLAN scheiden aufgrund der hohen Ungenauigkeit sofort aus. HighBall wäre eine Möglichkeit, ist aber aus Wirtschaftlichkeit nicht realisierbar. In die enge Wahl kommen somit nur noch die Bilderkennung und das IMAPS-Verfahren. Letzteres ist schon an der HAW verbaut, somit würden die Installationskosten entfallen. Da ein ähnliches Projekt jedoch gescheitert ist, wäre die Hauptaufgabe dieser Arbeit, die einzelnen Fehlerquellen weiter zu verringern. Ob dies überhaupt technisch lösbar ist, ist dabei nicht abzuschätzen und die Gefahr eines Fehlschlages ist hoch.

Die Positionsbestimmung mittels Bilderkennung ist hingegen schon erfolgreich umgesetzt worden und sieht sehr vielversprechend aus. Da das Projekt von William Thielicke bereits erfolgreich war und sehr ähnlich zu der Problemstellung dieser Arbeit ist, und die Programmierung mit MATLAB dank der Dokumentation und fertigen Toolboxen vieles erleichtert, wird dieses System bevorzugt. Der technische Aufwand ist erstaunlich gering. Lediglich ein paar Leuchtdioden, eine Webcam und ein PC mit entsprechender Software sind nötig. Hierin liegt ein weiterer Vorteil, denn die rechenintensive Bildanalyse findet nicht im Quadcopter oder einem mobilen Zusatzsystem statt², sondern auf einem leistungsstarken, stationären PC. Die

²Dies hätte zusätzliches Gewicht und erhöhten Stromverbrauch zur Folge

Fehleranfälligkeit erscheint hier demzufolge sehr gering. Eine Portierung in verschiedene Räume ist zudem mit wenig Aufwand möglich.

4.6. Zusammenfassung

Aufgrund der vielen positiven Faktoren, fiel die Wahl sehr leicht auf die Positionsbestimmung mittels Bilderkennung. Hierbei wird eine fest an die Decke montierte Kamera verwendet. An den Quadcopter werden drei Leuchtdioden befestigt, die nach oben zur Kamera ausgerichtet sind. Die Auswertung und Berechnung erfolgt an einem PC mittels MATLAB. Von dort werden die Steuerbefehle über eine Funkstrecke an den Quadcopter übertragen.

5. System-Design

In folgendem Kapitel wird ein Konzept erarbeitet, mit dessen Hilfe es möglich sein soll, einen Quadcopter im Raum zu stabilisieren und zu navigieren. Zunächst wird die benötigte Hardware betrachtet, anschließend die Software, welche für die Auswertung und Steuerung zuständig ist.

5.1. Hardware

In Kapitel 4.5 fiel die Entscheidung zugunsten der Positionsbestimmung per Webcam, welche an der Decke montiert ist. Der Quadcopter wird mittels Leuchtmarken im Raum erkannt und von einem PC über die serielle Schnittstelle gesteuert. (Abb. 5.1)

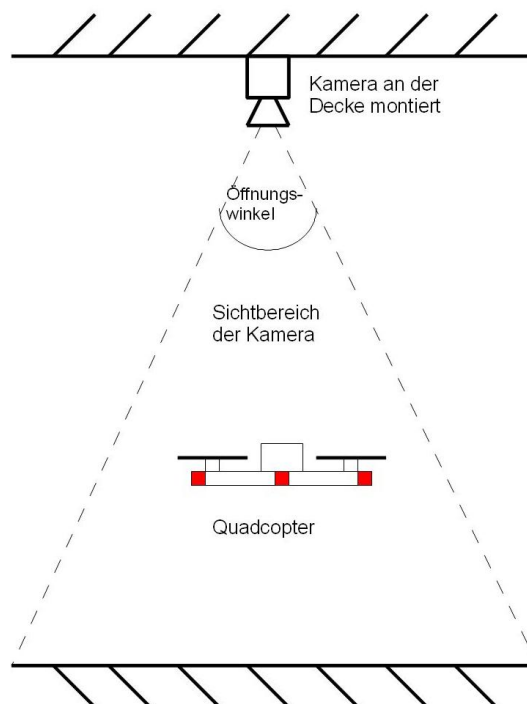


Abbildung 5.1.: Schematischer Aufbau

5.1.1. Quadcopter

An der HAW steht bereits ein Quadcopter zur Verfügung. Dieser wurde im Rahmen einer Bachelorarbeit konstruiert und ist mit einem ARM7 Prozessor für die Lageregelung ausgestattet. (Abb. 5.2 und 5.3) [Rempel (2008)]



Abbildung 5.2.: Quadcopter vor Umbau

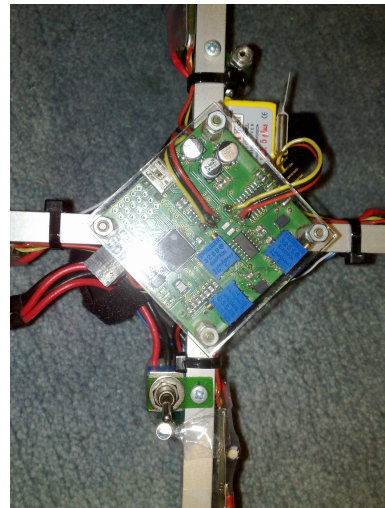


Abbildung 5.3.: ARM7 Steuereinheit

Um eine verbesserte Lageregelung für diese Arbeit zu erzielen, wird die Elektronik ersetzt. Es werden die zentrale Steuereinheit „Flight-Ctrl V2.0 ME“, ein Luftdrucksensor „MPX 4115A“ und vier Brushless-Regler¹ „BL-Ctrl V1.2“ von der Firma „MikroKopter.de“ verbaut.

Die Kommunikation zwischen PC und Quadcopter erfolgt mittels serieller Schnittstelle. Um eine direkte Kabelverbindung zu vermeiden, erfolgt die Übertragung durch eine Bluetooth-Funkstrecke. Diese wandelt die Signale für die Funkübertragung automatisch um. Hierfür sind zwei Bluetooth Module vom Typ „F2M03GXA“ und eine Bluetooth Adapter Platine „BT-AP“ nötig. [Ingo Busker (2010b)] Beim Quadcopter treffen dann wieder serielle Daten ein.

Die Verdrahtung muss für den Umbau ebenfalls komplett neu verlegt werden, da u.A. noch zusätzliche Leitungen für die LED Leuchtmarker, an drei der vier Ausleger benötigt werden.

¹Bürstenlose Gleichstrommotoren benötigen einen Controller, der die drei Phasen mittels PWM ansteuert [Ingo Busker (2009a)]

5.1.2. Positionierung der LEDs

Um die Position des Quadcopters zu bestimmen, würde eine LED ausreichen. Die Entfernung zur Kamera könnte man mit Hilfe von zwei LEDs ermitteln², indem man den Abstand zwischen den LEDs bestimmt. Um die Orientierung des Quadcopters zu bestimmen, benötigt man eine weitere (dritte) LED, welche in einem anderen Abstand - als die beiden anderen Leuchtmarken zueinander - angebracht ist. Hierbei bietet es sich an, drei Enden der Propellerausleger des Quadcopters zu verwenden, denn der größtmögliche Abstand verbessert die Auflösungsgenauigkeit der Position. Gleichzeitig wird so ein gleichschenkliges Dreieck aufgespannt, welches die unterschiedliche Entfernung der LEDs zueinander gewährleistet. (Abb. 5.4) Die Positionsbestimmung mittels drei LEDs geht davon aus, dass der Quadcopter

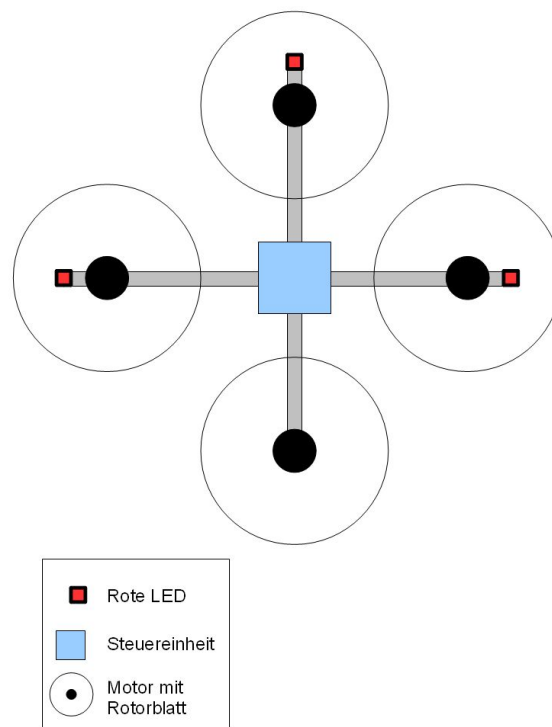


Abbildung 5.4.: Quadcopter Schema

stets waagrecht zur Kamera ausgerichtet ist. Sobald der Quadcopter verkippt wird, ändern sich die Abstände zwischen den Markern, was als Veränderung der Höhe interpretiert wird.

²Unter der Annahme, dass sich der Quadcopter stets in der Waagerechten befindet

Da der Quadcopter für die Steuerung maximal um 10° verkippt wird, hat dies jedoch wenig Auswirkung auf die Entfernung zwischen den LEDs.

Anhand folgender Beispielrechnung (5.1 und 5.2) wird die Abstandsänderung veranschaulicht. Die Entfernung zwischen zwei LEDs beträgt z.B. 400 Pixel. Die Verkipfung beträgt 10° .

$$\cos(10) * 400\text{Pixel} \approx 394\text{Pixel} \quad (5.1)$$

Bei einem Abstand von 100 Pixeln:

$$\cos(10) * 100\text{Pixel} \approx 98\text{Pixel} \quad (5.2)$$

Dies sind Abweichungen von 2-6 Pixeln und somit kaum von normalen Bilderkennungs-schwankungen zu unterscheiden. Aus diesem Grund kann die Auswertung der Verkipfung des Quadcopters vernachlässigt werden.

Demzufolge sind drei LEDs ausreichend, um die Position und Ausrichtung zu bestimmen.

5.1.3. Positionierung der Kamera

Da die Steuerung und Position des Quadcopters relativ zur Kameraposition ist, ist es nicht nötig, die Kamera an eine bestimmte Stelle im Raum zu befestigen. Es sollte lediglich darauf geachtet werden, dass der Flugbereich nicht von Gegenständen oder Wänden eingeschränkt wird. Eine Kalibrierung entfällt bei dieser Art der Positionsbestimmung, da nicht mit absoluten Höhen- und Längenangaben in Metern gerechnet wird, sondern mittels Pixelanzahl zwischen den Leuchtmarken. Abhängig vom Sichtbereich der Kamera wird der Aktionsradius von diesem eingegrenzt.

Zur Berechnung des Öffnungswinkel der Kamera wird die Raumhöhe und Aufnahmebereich gemessen. Bei einer Deckenhöhe von 2,10m überblickt die Kamera einen Bereich von 1,70m. Dies ergibt ein rechtwinkliges Dreieck mit der Ankathete der Länge 210cm und der Gegenkathete der Länge 85cm.

Der halbe Öffnungswinkel ergibt sich aus dem inversen Tangens (5.3 und 5.4):

$$\tan(\alpha) = \frac{85\text{cm}}{210\text{cm}} \quad (5.3)$$

$$\alpha = \arctan\left(\frac{85\text{cm}}{210\text{cm}}\right) \approx 22^\circ \quad (5.4)$$

Daraus ergibt sich der Öffnungswinkel von ca. 44° . Basierend auf der Optik und den Strahlengesetzen, bewegt man sich in einem pyramidenförmigen Flugraum, der sich zur Kamera hin verjüngt. Aufgrund der fixen Abmessungen des Quadcopters ist dieser Flugraum zusätzlich eingeschränkt, da alle Leuchtmarker gleichzeitig erfasst werden müssen. Die größte Entfernung zwischen zwei LEDs beträgt ca. 50cm.

Dies ergibt eine minimal zulässige Entfernung zwischen Quadcopter und Kamera von (5.5):

$$\frac{25cm}{\tan(22^\circ)} \approx 62cm \quad (5.5)$$

Hierbei ist jedoch zu beachten, dass keinerlei Sicherheitsabstand zum Kamerarand existiert. Kleinste Driftbewegungen führen sofort dazu, dass ein oder mehrere Leuchtmarker nicht mehr im FOV sind und eine Positionsbestimmung unmöglich ist.

Die optimale Flughöhe über dem Boden³ beträgt 60cm, und mindestens 1,50m von der Kamera entfernt.

Bei dieser Entfernung ist genügend Platz um die LEDs zu erfassen, verschiedene Navigationsziele anzufliegen und dabei ausreichend Sicherheitsabstand zum Kamerarand zu halten.

³Ein gewisser Abstand zum Boden ist nötig, um den Bodeneffekt zu minimieren.

5.1.4. IR-Filter

Da die automatische Blende vieler Kameras ungenügend ist und nicht nur die Leuchtmarken als hellste Punkte erkannt werden könnten, ist es nötig, einen Filter vor das Objektiv zu setzen. Für die Leuchtmarken sind rote LEDs vorgesehen, diese besitzen neben dem sichtbaren roten Spektrum auch noch einen Infrarotanteil. Im Gegensatz zum menschlichen Auge, haben die meisten Kameras eine gewisse Aufnahmeempfindlichkeit im IR-Spektrum. Ziel ist es also das sichtbare Licht auszufiltern und das Infrarotspektrum zu erhalten. Dies lässt sich einfach, mittels eines Disketten-Datenträgers realisieren, der eine ausreichende Infrarotfilterung⁴ besitzt. [(Bernin, 2009, S. 5)]

Mit diesem Filter ist im optimalen Fall das gesamte Kamerabild schwarz, bis auf die Leuchtpunkte, die als einzige helle Punkte sichtbar sind. (Abb. 5.5)



Abbildung 5.5.: Kamerasicht mit Infrarot Filter

⁴In diesem Fall ist jene Filterung gemeint, die nur infrarot-lichtdurchlässig ist

5.2. Software

Dieses Kapitel beschäftigt sich mit der Programmierung der Bilderkennung⁵ und Steuerung. In folgender Grafik ist der Programmablauf dargestellt (Abb. 5.6). Die grafische Benutzeroberfläche (GUI) ermöglicht es, das Tracking incl. Steuerung⁶ zu starten. Daraufhin läuft periodisch ein Timer ab, der die nötigen Berechnungen sequentiell abarbeitet. Ein Timer mit definierter Verzögerung ist erforderlich, da MATLAB bei einer Endlosschleife ohne Verzögerung keine Benutzereingabe mehr ermöglichen würde. Die benötigten Einstellungen für die Kamera und die PID-Regelung können über die GUI eingegeben werden. Um das Tracking zu beenden, wird über die GUI der Timer gestoppt.

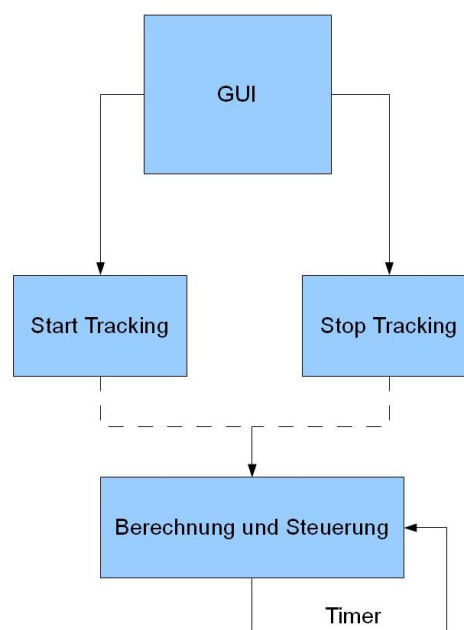


Abbildung 5.6.: Softwarehierarchie

In Abbildung 5.7 ist der Ablauf des Timers dargestellt. Zunächst wird das Bild der Webcam angefordert, verarbeitet und vereinfacht (Im Abschnitt 5.2.1 wird hierauf näher eingegangen). Anschließend folgt die Berechnung der aktuellen Position des Quadcopters und der Vergleich mit der Zielposition. Daraus ergibt sich eine Abweichung vom Sollwert. Diese wird im nächsten Schritt an die PID-Regelung übergeben, welche daraus die Steuerbefehle berechnet und im abschließenden Schritt, mittels serieller Schnittstelle an den Quadcopter sendet.

⁵Der Programmcode für die Bilderkennung basiert größtenteils auf dem Code von William Thielicke. [Thielicke (2009b)]

⁶Der Code für die Generierung der Steuerbefehle basiert auf den Angaben aus dem Mikrokoopter Wiki [Ingo Busker (2010a)] und dem DUBwise-Projekt [ligi (2010)]

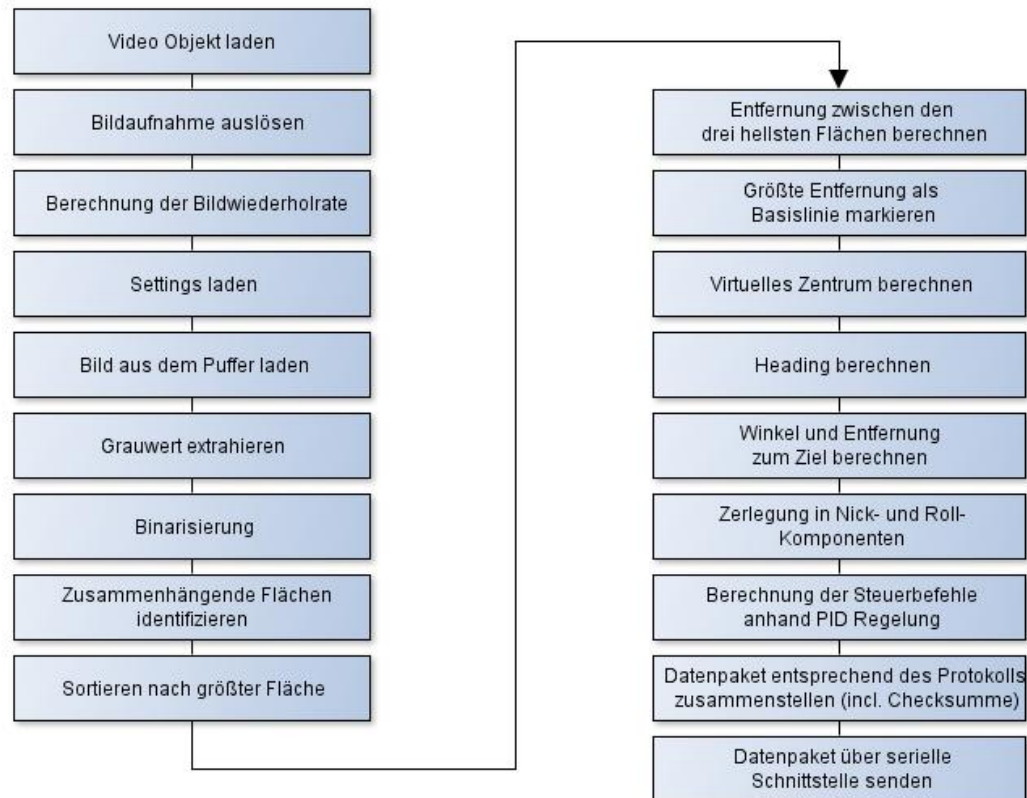


Abbildung 5.7.: Flussdiagramm der Software

5.2.1. Bilderkennung

Für die Bilderkennung benötigt man zunächst das Bild der Webcam. Dieses erhält man durch die „Image Acquisition Toolbox“, welche Teil von MATLAB ist. Diese Toolbox ermöglicht den Zugriff auf eine Webcam und kann je nach Einstellung, zyklisch eine Bildaufnahme auslösen und diese in einer Variablen zur weiteren Verarbeitung zur Verfügung stellen. Um die Bearbeitungsgeschwindigkeit zu erhöhen, wird das Bild mittels der Funktion „im2bw“ auf ein Binärbild reduziert (der benötigte Schwellwert wird experimentell bestimmt).

```
1 A = im2bw(imgl, (settings.thresh/255));
```

Da die meisten Elemente des Arrays gleich null sind, können sie mit Hilfe der Sparse-Funktion effizienter gespeichert werden. Da wir an den drei hellsten Punkten interessiert sind, werden zusammenhängende Flächen gesucht, gruppiert und mittels der Mean-

Funktion(arithmetisches Mittel) die Zentren bestimmt. Die einzelnen Einträge werden dann nach Größe der Flächen sortiert.

Die ersten drei Einträge der Matrix entsprechen nun den drei hellsten Punkten, die eventuell restlichen erkannten Punkte⁷ werden verworfen.

```
1 Ainfo(nrpoints+1:end, :)=[];
```

Am Ende der Vereinfachung erhält man eine Matrix mit den jeweiligen X- und Y-Koordinaten der drei hellsten Punkte im Bild.

5.2.2. Berechnung

Um den Abstand zwischen zwei Punkte zu berechnen, wendet man den Satz des Pythagoras an. (Abb. 5.8)

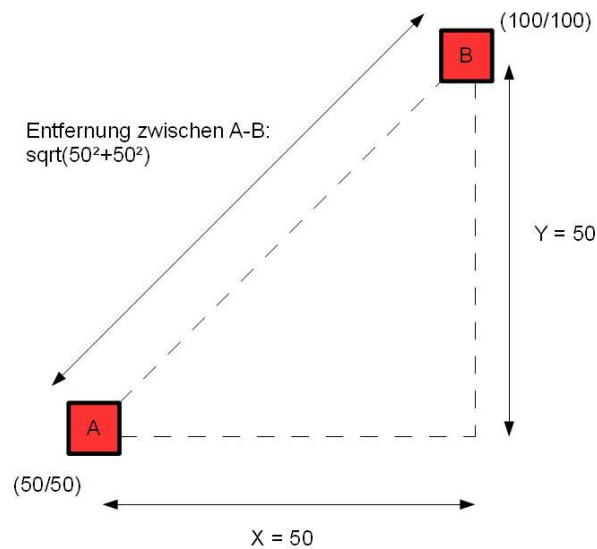


Abbildung 5.8.: Berechnung der Entfernung zwischen zwei Punkten

⁷Streulicht oder Reflexionen die nicht komplett durch die schwarz-weiß-Reduktion ausgefiltert wurden

Position

Die drei LEDs sind jeweils an einem Ausleger befestigt. Die längste Strecke zwischen ihnen ist die Basislinie, sie verläuft durch das Zentrum des Quadcopters. (vgl. Abb. 5.4) Es bietet sich an, den Mittelpunkt zwischen den am weitest entfernten Markern, als den virtuellen Mittelpunkt zu wählen, da dieser dem tatsächlichen Mittelpunkt des Quadcopters am ehesten entspricht. Die dritte LED ist am vorderen Ausleger angebracht, und markiert somit die Front des Quadcopters. Diese Information ist wichtig, damit die Nick- und Roll-Steuerbefehle nicht verwechselt werden.

Die Höhenberechnung erfolgt durch Berechnung des Pixelabstandes zwischen den LEDs der Basislinie. Je weiter entfernt der Quadcopter von der Kamera ist, desto kleiner wird dieser Abstand. (Abb. 5.9)

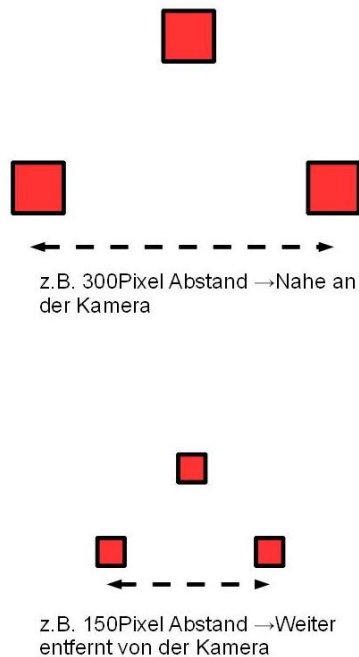


Abbildung 5.9.: Bestimmung der Entfernung zur Kamera

Heading

Um die Ausrichtung des Quadcopter zu bestimmen, wird die Front LED und das virtuelle Zentrum verwendet. Ausgehend von dieser Geraden durch die beiden Punkte, wird die Ausrichtung berechnet.

Mit der Funktion „atan2“ berechnet man den Winkel zwischen dem Koordinatenursprung und einem beliebigen Punkt. Um den Winkel zwischen zwei Punkten zu bestimmen, muss das Punktepaar so verschoben werden, dass ein Punkt im Koordinatenursprung liegt. Dies erreicht man, indem von der Front-Koordinate, die Zentrums-Koordinate abgezogen wird. $\text{atan2}(\text{„Y Front“} - \text{„Y Zentrum“}, \text{„X Front“} - \text{„X Zentrum“})$

```
1 heading = 180/pi*atan2(front(2)-center(2), front(1)-center(1));
```

Entfernung zum Ziel

Anhand der Position (5.2.2) und dem Heading (5.2.2) ist es möglich, den Abstand zum Ziel zu berechnen. Da sich der Quadcopter in einem beliebigen Winkel zum Ziel befinden kann, muss die Entfernung in Nick- und Roll-Komponenten aufgesplittet werden. Die Grafik 5.10 veranschaulicht die Aufteilung in die Nick (y-Achse) und Roll (x-Achse) Komponente, bezüglich dem Quadcopter-Zentrum und Zielpunkt.

Zunächst wird der Winkel zwischen dem Quadcopter und dem Ziel berechnet. Der relative Winkel ergibt sich aus der Differenz zwischen dem Heading und dem gerade berechneten Winkel.

Anschließend wird die kürzeste Entfernung zwischen Quadcopter und Ziel berechnet. (vgl. 5.2.2) Mit Hilfe des relativen Winkels und der Entfernung, können die Nick⁸- und Roll⁹-Komponenten bestimmt werden.

```
1 target_center_angle = 180/pi*atan2(center(2)-settings.target_y,center
  (1)-settings.target_x);
2 target_center_distance = sqrt( (center(1)-settings.target_x)^2+(center
  (2)-settings.target_y)^2 );
3 x_rel = sind(heading-target_center_angle)*target_center_distance;
4 y_rel = cosd(heading-target_center_angle)*target_center_distance;
```

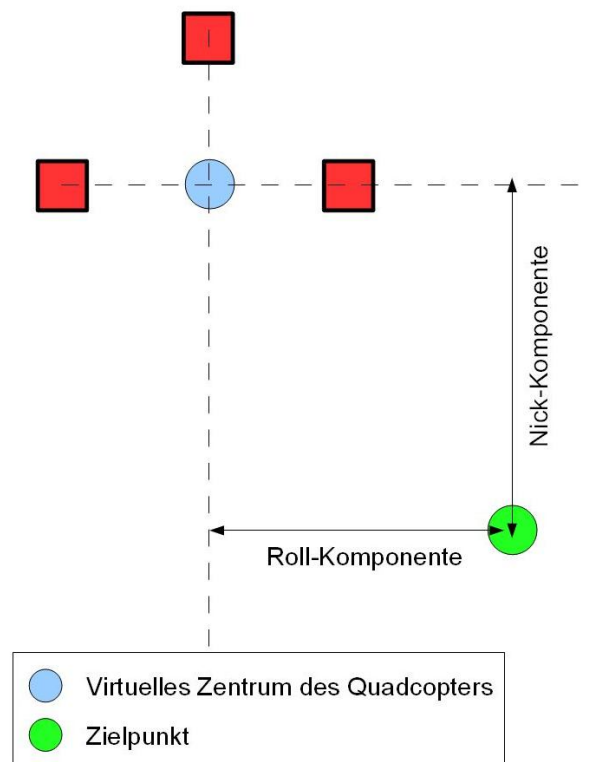


Abbildung 5.10.: Aufsplitten der Entfernung in Nick- und Roll-Komponenten

⁸Y-Relativ

⁹X-Relativ

5.2.3. Regelung (PID)

Da nun die auszuregelnden Abstände bekannt sind, können daraus die Steuerbefehle (5.2.4) generiert werden. Die einfachste Art dies umzusetzen, wären statische Steuerbefehle, die je nach Abweichungsrichtung nur das Vorzeichen ändern¹⁰. Der Nachteil dieser Regelung ist bei diesem instabilen Flugkörper offensichtlich. Es ist sehr unwahrscheinlich, dass die Abweichung vom Ziel genau null ist. Doch selbst der kleinste Wert würden einen Steuerbefehl zur Folge haben, welcher ggf. dafür sorgt, dass der Quadcopter über sein Ziel hinaus schießt und bei der nächsten Messung eine entgegengesetzte Regelabweichung vorliegt. Der Quadcopter würde somit nie zur Ruhe kommen.

Das grundlegende Problem dieser Regelung besteht darin, dass es technisch nicht möglich ist, die Regelabweichung ohne Verzögerung¹¹ zu bestimmen. Ebenso können die Steuerbefehle nicht unverzüglich umgesetzt werden, da diese zunächst übertragen und ausgewertet werden müssen. Des Weiteren handelt es sich beim Quadcopter um eine träge Masse, diese kann nicht sofort die Bewegungsrichtung oder Geschwindigkeit ändern. Je nach Beschleunigungsstärke, benötigt ein Flugmanöver eine gewisse Zeit, bis es umgesetzt wird. Das bedeutet, dass eine Latenzzeit zwischen Regelabweichung und Korrektur vorliegt. Zudem kann die Auswertung nicht beliebig schnell hintereinander ausgeführt werden, da sie Rechenzeit benötigt. D.h. Regelabweichungen werden nicht nur verzögert wahrgenommen, sondern auch nur mit einer gewissen zeitlichen Auflösung¹². Je nach Steuerbefehl kann dies zu einer Mitkopplung führen. Dies äußert sich in einem Aufschwingen des Systems und kann im ungünstigsten Fall außer Kontrolle geraten und zum Absturz führen.

Dieses Verhalten ist ein bekanntes Problem in der Regelungstechnik. Um diesem bestmöglich zu begegnen wird ein linearer Regler benutzt. Bei diesem Regelkreis bietet es sich an einen PID Regler zu verwenden, da es auf schnelle Reaktion und genaue Ausregelung ankommt. [Böge (2007)] Der Mehraufwand (im Vergleich zu PD- oder PI-Reglern) für die Berechnung kann bei den heutigen Rechenleistungen vernachlässigt werden.

Der Proportionalanteil ändert sich abhängig von der Größe der Regelabweichung. Ist der Quadcopter weiter von seinem Ziel entfernt, wird stärker geregelt als wenn er nah am Ziel ist. Der Verstärkungsfaktor sorgt für die gewünschte Verstärkung, proportional zur Regelabweichung.

```
1 output = (pid.p.nick_roll*y_rel); %Proportionalanteil
```

¹⁰z.B. Y-Abweichung=+60Pixel, Steuerbefehl=Nick -5. Y-Abweichung=-10Pixel, Steuerbefehl=Nick +5

¹¹Bildaufnahme, Verarbeitung und Berechnung benötigen Zeit

¹²Gemessen in Bilder pro Sekunde, bzw. Frames per second (fps)

Der Integralanteil nimmt stetig zu, so lange eine Regelabweichung vorliegt. Mit jedem Berechnungsdurchgang wird die Abweichung (in Abhängigkeit von der Bildwiederholrate) aufsummiert. Die Summe ist nach oben und unten hin begrenzt um zu extreme Flugmanöver zu verhindern. Mit Hilfe des Integralanteils ist es möglich, die Regelabweichung vollständig auszuregeln, da der Ausgangswert ständig zunimmt, bis keine Abweichung mehr vorhanden ist.

```
1      %Integralanteil
2      dt = 1/runningat;          %runningat = fps
3      pid.sum.nick = pid.sum.nick + (y_rel*dt);
4      if(pid.sum.nick>500)
5          pid.sum.nick = 500;
6      else if(pid.sum.nick<-500)
7          pid.sum.nick = -500;
8          end
9      end
10     output = (pid.i.nick_roll*pid.sum.nick);
```

Der Differentialanteil reagiert auf die Änderungsgeschwindigkeit der Regelabweichung. Eine große Änderung der Abweichung, im Vergleich zur letzten Messung, führt zu einem starken Regelausschlag. Die Annäherung des Differentialwertes wird dadurch erreicht, dass die Differenz der letzten beiden Abweichungen verglichen wird. [Wikipedia (2010a)]

```
1      %Differentialanteil
2      dt = 1/runningat;
3      derivative_nick = (y_rel - pid.old_error.nick)/dt;
4      output = (pid.d.nick_roll*derivative_nick);
5      pid.old_error.nick = y_rel;
```

Die Einstellung der jeweiligen Faktoren geschieht für jeden Steuerwert getrennt. Lediglich Nick und Roll benutzen die selben PID-Faktoren, da sie sich sehr ähnlich verhalten. Die optimalen Werte werden experimentell bestimmt.

5.2.4. Steuerbefehle

Beim Start der Anwendung wird der Serielle Port auf eine Bitrate von 57600 Baud initialisiert¹³ und bleibt während der Programmlaufzeit geöffnet. Das Protokoll ist folgendermaßen aufgebaut: (5.1)

| Start-Byte | Address Byte | ID-Byte | n Data-Bytes | CRC1 | CRC2 | Stop-B. |
|------------|--------------|-------------|-------------------|----------|----------|---------|
| '#' | 'a'+ Addr | 'V','D' etc | „modified-base64“ | variable | variable | '\r' |

Tabelle 5.1.: Protokoll für die Steuerbefehle (Ingo Busker (2010a))

Der Quadcopter empfängt die Steuerbefehle mittels des oben gezeigten Protokolls. Die Adresse der FlightControl ist fest vorgegeben. (Das Adress Byte ist somit „a + 1“). Das ID-Byte legt fest, welches Kommando gesendet wird, z.B. „r“ für einen Reset oder wie es für diese Arbeit benötigt wird „b“ um einen externen Steuerbefehl zu übermitteln. Nick, Roll, Gier und Gas werden durch eine modifizierte base64 Codierung zusammengefasst. Abgeschlossen wird das Datenpaket mit mit einer Checksumme. Dies hat zur Folge, dass jeweils alle Steuerbefehle gesendet werden müssen, auch wenn z.B. nur der Nick-Wert geändert wurde. Nachdem die Werte zusammengefasst wurden, wird das Datenpaket versendet. Beim Empfänger angekommen, wird es entpackt und die entsprechenden Steuerwerte umgesetzt.

¹³Dieser Wert ist durch das Protokoll und Konfiguration der Quadcopter Firmware fest vorgegeben.

6. Realisierung

In diesem Kapitel werden die zuvor geplanten Schritte umgesetzt und getestet. Das Kapitel ist in zwei Teile gegliedert. Zunächst wird die Hardware betrachtet, anschließend erfolgt die Behandlung der Software. Schwierigkeiten bei der Realisierung werden erläutert, Verbesserungsmöglichkeiten vorgeschlagen und gegebenenfalls umgesetzt.

6.1. Hardware

Da der Quadcopter bisher mit einer ARM7 Eigenentwicklung bestückt ist, welche noch keinen ausreichend stabilen Schwebeflug ermöglicht, wird der Quadcopter mit einer neuen Steuereinheit umgerüstet, die in Modellbauszene als Standard anerkannt ist und als ausgereift gilt. Der Umbau des Quadcopters mit der neuen Steuereinheit erweist sich als sinnvoll, da sich nun ein sehr ruhiges Flugverhalten einstellt. Bei der Realisierung der Hardware treten im weiteren Verlauf jedoch einige Schwierigkeiten mit der Webcam und den LEDs auf. Diese werden nachfolgend erläutert.

6.1.1. Quadcopter

Wie im Kapitel [5.1.1](#) beschrieben, wird die Elektronik und Verdrahtung komplett getauscht um eine stabilere Fluglage zu erzielen. (Abb. [6.1](#)) Die neuen Gyroskope und die Steuereinheit erweisen sich in Flugtests als deutliche Verbesserung für die grundlegenden Lagestabilität.



Abbildung 6.1.: Quadcopter nach Umbau

6.1.2. Kamera

Bei der verwendeten Webcam handelt es sich um das Modell „Microsoft LifeCam VX 5000“, welche per USB an einen Laptop angeschlossen ist. Die maximale Auflösung beträgt 640x480 Pixel. Die Kamera ist fest an die Decke montiert und senkrecht zum Boden ausgerichtet.

Wie sich im Verlauf der Realisierung herausstellt, liefert die MATLAB-Berechnung, in Verbindung mit der Webcam von Microsoft, eine zu geringe Bildwiederholrate von 4 fps. Daher wird sie gegen eine Philips Webcam (SPC230NC) ausgetauscht. (Abb. 6.2) Genau dieses Modell wurde auch von William Thielicke verwendet.

Durch den Tausch der Webcam wurde die Bildwiederholrate verbessert. Leider werden nicht die erhofften 12Hz¹ erreicht, sondern lediglich eine Verbesserung von 4Hz auf 6Hz.

¹Diesen Wert erreichte William Thielicke mit seinem Projekt



Abbildung 6.2.: Philips Webcam an der Decke montiert

6.1.3. LED

Zunächst werden roten Leuchtdioden² mit einem Abstrahlwinkel von 30° und 4500mcd verwendet. (Abb. 6.3)

Diese erweisen sich aufgrund ihres geringen Abstrahlwinkels jedoch als unbrauchbar, da im Randbereich der Kamera und in bestimmten Fluglagen die Leuchtintensität stark zurück geht und somit von der Software nicht mehr erkannt werden können. Daraufhin werden neue SMD-LEDs³ mit einem Abstrahlwinkel von 60° und 5600mcd verbaut. (Abb. 6.4)

²Typ „LED 5-4500 RT“ [Reichelt Elektronik (2010)]

³Typ „OSRAM Power TOPLED, Linse,5600mcd,super-rot“ [Reichelt Elektronik (2010)]



Abbildung 6.3.: 1. LED-Version mit 30° Abstrahlwinkel



Abbildung 6.4.: 2. LED-Version mit 60° Abstrahlwinkel

Diese sind nun aus nahezu jeder Fluglage für die Kamera sichtbar und ermöglichen eine durchgängige Lokalisierung. (Abb. 6.5)

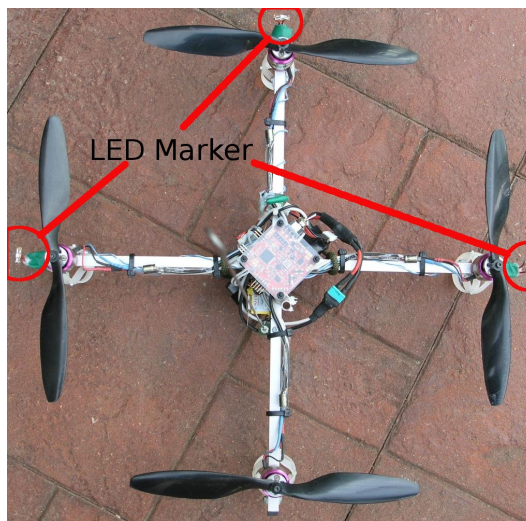


Abbildung 6.5.: Positionierung der LEDs

6.1.4. Sicherheitsmaßnahmen

Da für die PID-Regelung zunächst die passenden Parameter gefunden werden müssen, ist es nötig eine Auffangvorrichtung zu bauen, damit der Quadcopter während der Testphase nicht unkontrolliert ausbrechen kann. Hierfür wird eine Nylonschnur zwischen Decke und Boden gespannt. Ein an den Quadcopter befestigtes Rohr dient als Führungsstange, welche verhindert, dass die Schnur bei Schräglage in die Rotoren gerät. Je nach Spannkraft der Schnur ist der Aktionsradius in Nick- und Rollrichtung variabel, aber soweit begrenzt, dass der Quadcopter nicht aus dem Sichtbereich der Kamera fliegen kann. Um ein unkontrolliertes Ausbrechen in Richtung Decke zu verhindern, wird eine weitere Schnur am Boden und Quadcopter befestigt, welche die maximale Flughöhe begrenzt. Dieser Aufbau ermöglicht es nun, die PID-Werte in Ruhe einzustellen.

6.2. Software

In folgendem Teil wird die Softwarerealisierung behandelt. Bei der Umsetzung treten unvorhergesehene Schwierigkeiten mit der Bildwiederholfrequenz und der Programmierumgebung auf. Wie im Hardwareteil schon beschrieben, kann durch eine neue Webcam die Frequenz immerhin um 50 Prozent gesteigert werden. Die 6Hz sind aber für die Lagestabilisierung nicht ausreichend. Der Quadcopter beginnt sich aufzuschwingen, es erweckt den Anschein, dass die Auswertung der Abweichung zu selten erfolgt und die Steuerbefehle zu lange anliegen. Nachfolgend wird versucht, dieses Problem softwaretechnisch zu lösen.

6.2.1. MATLAB Code

Es werden Verbesserungen am Code vorgenommen, um die Rechengeschwindigkeit zu erhöhen.

Initialisierung fester Arraygrößen

Durch Initialisierung von Arrays mit statischer Größe wird verhindert, dass das Array für jeden neuen Wert erweitert und der Inhalt umkopiert werden muss.

```
1   %Initialize
2   numberofpixels=zeros (maxXM, 1) ;
3   iind=zeros (maxXM, 1) ;
4   jind=zeros (maxXM, 1) ;
```

Diese Änderung führt jedoch zu keinem messbaren Geschwindigkeitszuwachs.

Verlegen des Triggerevents

Weiter wird versucht die Geschwindigkeit zu erhöhen, indem die Auslösung (Trigger) des nächsten Bildes vor die Berechnung verlegt wird. Der Ablauf der Routine läuft nun folgendermaßen ab:

- Timergesteuerter Eintritt in die Berechnungsroutine
- Video Objekt laden
- Bild aus dem Puffer laden
- Puffer der Kamera leeren
- Bildaufnahme auslösen (für den nächsten Aufruf der Routine)
- Berechnungen
- Routine verlassen

Diese Änderung führt zu einer Steigerung der Bildwiederholrate von 6Hz auf 8Hz.

Optimierung der grafischen Ausgabe

Da die 8Hz noch nicht für einen stabilen Flug ausreichen, wird der Code weiter optimiert. Als großer Zeitfresser wird die Plot- und Text-Funktion identifiziert. Die visuellen Einblendungen werden durch „hold on“ übereinander gelegt. Dies führt zu einer erheblichen Geschwindigkeitseinbuße. Werden diese Einblendungen weggelassen und nur das Bild der Webcam angezeigt, so erhöht sich die Auswertungsgeschwindigkeit von 8Hz auf 11-12Hz. Wird gar kein Bild ausgegeben, so sind Geschwindigkeiten von bis zu 24-25Hz möglich⁴.

Wechsel auf ein 64Bit System

Die oben genannten 24-25Hz scheinen durch die Rechenleistung des PCs begrenzt zu sein. Die CPU Auslastung ist am Limit. Es wird daher versuchsweise ein 64Bit Rechner auf höhere Performanz getestet. Nach der Portierung auf ein 64Bit Windows7 System⁵ sind Geschwindigkeiten von 29-30Hz möglich. Dies entspricht der physikalischen Grenze der Webcam, welche bei 30 Bildern pro Sekunde liegt.

⁴Windows XP 32Bit, MATLAB 32Bit, 2GB RAM, Intel Core 2 - T7200 @ 2GHz

⁵Windows 7 64Bit, MATLAB 64Bit, 2GB RAM, Intel Core 2 Duo - L9400 @ 1,86Ghz

6.2.2. Latenzzeit zwischen Realität und Kamerabild

Trotz der, nun erreichten, hohen Auswertungsgeschwindigkeit besteht weiterhin das Problem, dass die Bilder verzögert beim PC eintreffen⁶. Um die Latenz zu messen wird eine periodisch blinkende LED verwendet. Die Frequenz wird so lange erhöht, bis sich zwei Blinkimpulse überlagern. D.h im Bild ist der Impuls „n“ zu sehen, die LED sendet in diesem Augenblick den Impuls „n+1“ aus. Die Zeit zwischen den Impulsen ist somit die gesuchte Verzögerung.

Diese Schaltung wurde mit einem ATM18 Testboard realisiert. (Abb. 6.6)

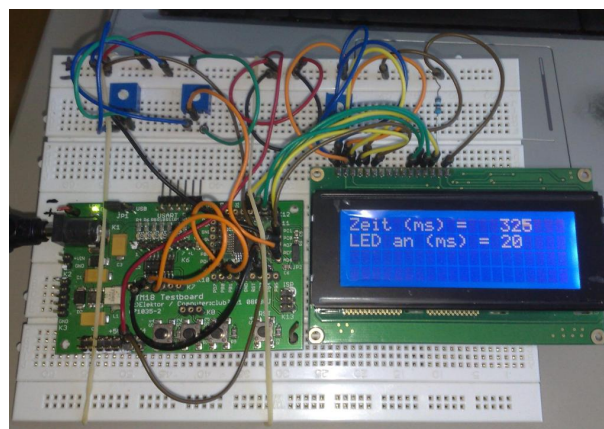


Abbildung 6.6.: Testboard für die Latenzzeitmessung

Der Quellcode ist in Bascom geschrieben:

```
1 Do
2   Zeit = Getadc(7)
3   Ledan = Getadc(6)
4   Ledan = Ledan / 5
5   Portb.1 = 1
6   Portb.2 = 1
7   Portb.3 = 1
8   Waitms Ledan
9   Portb.1 = 0
10  Portb.2 = 0
11  Portb.3 = 0
12
13  Ledaus = Zeit - Ledan
14  If Ledaus < 0 Then
```

⁶Auch bekannt unter capture-to-display latency (CDL)

```
15     Ledaus = 0
16     End If
17
18     Cls
19     Locate 1 , 1
20     Lcd "Zeit (ms) ="
21     Locate 1 , 15
22     Lcd Zeit
23
24     Locate 2 , 1
25     Lcd "LED an (ms) ="
26     Locate 2 , 15
27     Lcd Ledan
28
29     Waitms Ledaus
30 Loop
```

Die gemessene Latenzzeit zwischen Realität und dem Eintreffen in MATLAB, von ca. 325ms ist dafür verantwortlich, dass die Steuerbefehle für eine Position berechnet werden, an welcher sich der Quadcopter in Wirklichkeit gar nicht mehr befindet. Dieses Verhalten lässt sich nicht ohne weiteres beschleunigen, da dies einen Eingriff in den Kameratreiber zur Folge hätte und den Rahmen dieser Arbeit sprengen würde.

6.2.3. Bugs in der Programmierumgebung

Aufgrund verschiedener Fehler in der Programmierumgebung selbst kommt es zu erheblichen Verzögerungen. Es stellt sich heraus, dass der Programmcode fehlerfrei ist, sich jedoch nicht jedes Mal der gewünschte Effekt einstellt. Um einen PID-Wert in der GUI zu übernehmen, ist es normalerweise ausreichend, das Eingabefeld zu wechseln oder mit „Enter“ zu bestätigen. Dies geschieht aber entgegen der Dokumentation von MATLAB nicht jedes mal zuverlässig. Zur Sicherheit muss man mehrmals (ca. 2-4x) den Wert bestätigen, bevor dieser übernommen wird. Laut Dokumentation ist der Programmcode korrekt, daher wird angenommen, dass es sich hierbei um einen Bug seitens der Programmierumgebung handelt. Aufgrund dieses Fehlverhaltens kann es passieren, dass nur Teile der PID-Faktoren übernommen werden. Dies hat extreme Auswirkungen auf das Flugverhalten des Quadcopters und macht die oben genannten Sicherheitsmaßnahmen nötig.

6.2.4. Firmwareanpassung

Da die Firmware mit dem Quellcode des Quadcopters zur Verfügung steht, ist es möglich, nötige Anpassungen vorzunehmen. Für Flüge im Freien ist folgendes Verhalten als Sicher-

heitsfeature gedacht: Per Fernbedienungsschalter kann die externe Kontrolle an den PC vergeben werden. Damit der Quadcopter bei einem fehlerhaften Signal kontrollierbar bleibt, wird der maximale Gaswert von der Fernsteuerung vorgegeben. D.h. die PC Steuerung kann nur so viel Gas geben wie momentan an der Fernsteuerung geregelt ist. So ist es möglich, den Quadcopter bei Fehlverhalten sehr schnell auf den Boden zurück zu bringen. Problematisch wird diese Sicherheitsregelung aber in geschlossenen Räumen. Folgendes Szenario:

Die Fernsteuerung hat die Kontrolle an den PC abgegeben, damit der Quadcopter fliegen kann und genug Reserven zum Regeln hat, muss der Gaswert an der Fernsteuerung relativ hoch ausgegelt werden. Solange der PC regelmäßig Steuerbefehle sendet, ist der Quadcopter unter Kontrolle und lässt sich steuern. Bleiben die externen Steuerbefehle aus (z.B. durch Störungen in der Übertragung oder ein Softwareabsturz), so wird aus Sicherheitsgründen, nach einem Timeout, die Steuerkontrolle wieder an die Fernbedienung zurück gegeben. Da dort ein relativ hoher Gaswert eingestellt ist, beschleunigt der Quadcopter sehr stark nach oben und kollidiert mit der Decke und sehr wahrscheinlich auch der Kamera. Dies hat eine Beschädigung am Quadcopter und eventuell auch weiteren Gegenständen zur Folge.

Daher wird der Quellcode der Quadcopter-Steuereinheit angepasst und die Bedingung entfernt, dass von der Fernbedienung der maximale Gaswert vorgegeben werden muss. [Ingo Busker (2009b)]

```
1  if(ExternControl.Config & 0x01 && Parameter_ExternalControl > 128)
2  {
3    StickNick += (int) ExternControl.Nick * (int) EE_Parameter.Stick_P;
4    StickRoll += (int) ExternControl.Roll * (int) EE_Parameter.Stick_P;
5    StickGier += ExternControl.Gier;
6    ExternHoehenValue = (int) ExternControl.Hight * (int)EE_Parameter.
        Hoehe_Verstaerkung;
7    StickGas = ExternControl.Gas; //alter Code: if(ExternControl.Gas <
        StickGas) StickGas = ExternControl.Gas;
8  }
```

6.3. Zusammenfassung

Bei der Realisierung wurde mehr Zeit benötigt als geplant. Die Erkennungsschwierigkeit der LEDs konnte durch einen anderen LED-Typ behoben werden.

Das Problem der niedrigen Auswertungsgeschwindigkeit konnte durch den Kameratausch und den Optimierungen an der Software ausreichend gesteigert werden. Die Höhenregelung konnte durch die passenden PID-Werte stabilisiert werden. Ebenso ist die Gierkontrolle stabil. Bei der Nick- und Rollsteuerung konnten jedoch keine passenden PID-Werte bestimmt werden, die das Aufschwingen ausreichend gedämpft haben. Als Ursache für dieses Problem wurde die Latenzzeit zwischen Realität und dem Eintreffen des Bildes im PC identifiziert.

7. Fazit

Die Umsetzung der Aufgabenstellung, eine Indoornavigation für einen Quadcopter zu realisieren ist nur teilweise erfolgreich. Die Bilderkennung und damit verbundene Positionsbestimmung funktioniert einwandfrei. Ebenso die draus abgeleiteten Steuerbefehle werden fehlerfrei vom Quadcopter umgesetzt. Eine grundlegende Wegpunktnavigation ist möglich. Die Bildwiederholrate konnte von anfänglich 4Hz auf 12-30Hz gesteigert werden (Je nachdem ob mit, oder ohne grafischer Ausgabe). Lediglich die Latenzzeit konnte nicht ausgeglichen werden, welche weiterhin zu einem Aufschwingen des Systems führt und einen freien, selbstständigen Flug verhindert. Die grundlegenden Steuerbefehle, basierend auf der Regelabweichung sind korrekt, werden jedoch für eine Position berechnet, die um einige Hundert Millisekunden in der Vergangenheit liegt und nicht mehr der realen Position entspricht.

7.1. Verbesserungsvorschläge

Um das Problem des Aufschwingens zu kompensieren, könnte man zusätzlich zum Differentiell-Regler, welcher die Änderungsgeschwindigkeit berücksichtigt, einen weiteren Regler implementieren, welcher die Beschleunigung des Quadcopters bestimmt. Aus dieser Beschleunigung wird dann ein Regelwert berechnet.

Da die Latenzzeit bekannt ist, könnte man versuchen die Real-Position durch Interpolation anzunähern und anhand dieser, die aktuell benötigten Steuerbefehle zu berechnen. Basierend auf der letzten Position, der Geschwindigkeit und Beschleunigung kann eine wahrscheinliche Position mit dem zeitlichen Versatz t interpoliert werden.

Um die Latenzzeit zu verringern, könnte man auch versuchen, direkt auf den Datenstrom der Kamera zuzugreifen und den Treiber dementsprechend selber schreiben. Da die Verzögerung von der Art der Umsetzung des Treibers abhängt, könnten andere Kameramodelle (mit entsprechenden Treibern) zu einem besseren Ergebnis führen.

Eine weitere Möglichkeit wäre es, die Auswertung des Bildes und Bestimmung der Position, einem FPGA zu überlassen. Dieser ermöglicht, durch die parallele Verarbeitung, eine hohe Auswertungsgeschwindigkeit und je nach Zugriff auf die Kamera, eine niedrigere Latenz. Es wäre noch zu diskutieren, wie die Kommunikation zwischen FPGA und PC umgesetzt wird,

um dort möglichst geringe Verzögerungen zu generieren, oder ob es dann nicht sogar sinnvoller wäre, die Bildanalyse, Berechnung und Steuerung komplett im FPGA umzusetzen.

Alternativ kann man versuchen eine „Wii Remote“ zur Bildanalyse zu verwenden. Diese bestimmt hardwaremäßig die Koordinaten von bis zu 4 Punkten, bei einer Frequenz von 100Hz. Johnny Chung Lee hat mit seinem Projekt gezeigt, dass es möglich ist, mehrere Punkte unabhängig voneinander, mit eine „Wii Remote“ zu erkennen. [Lee (2007)]

7.2. Ausblick in die Zukunft

Sobald das Problem der Latenzzeit gelöst ist, und eine stabile Positionierung hergestellt werden kann, stellt sich die Frage, wie man das System nutzen oder erweitern könnte. Zunächst fällt einem der äußerst begrenzte Flugraum auf, der bisher auf den Sichtbereich der Kamera beschränkt ist.

Als eine Möglichkeit könnte man ein anderes Kameramodell verwenden. Es gibt Kameras mit einem 360° Sichtbereich. [MOBOTIX (2010a)] Die entstehende Verzerrung kann hierbei von der Software nach Bedarf korrigiert werden. (Abb. 7.1)

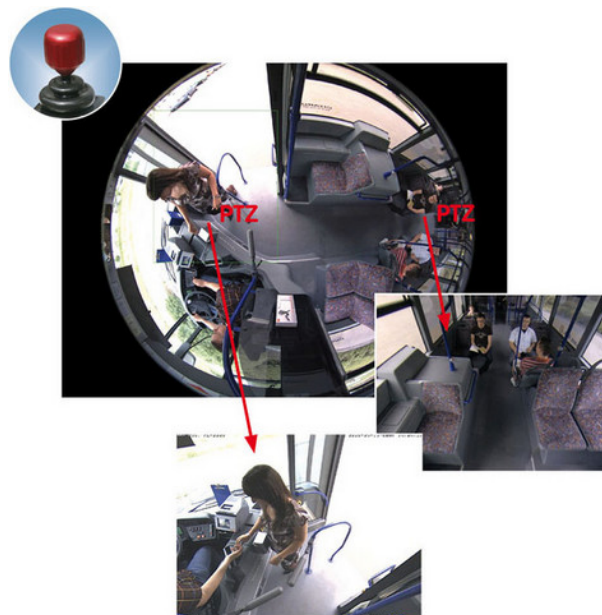


Abbildung 7.1.: 360° Kamera Q24M mit automatischer Entzerrung [MOBOTIX (2010b)]

Eine andere Möglichkeit diesen Raum zu erweitern, wären zusätzliche Kameras mit leicht überlappenden Sichtbereichen. Ähnlich dem Prinzip von Mobilfunkzellen, könnte ein Algo-

rhythmus entwickelt werden, der die Kontrolle über den Quadcopter von einer Zelle zur nächsten koordiniert und weitergibt. Somit wäre es möglich, den Flugbereich beliebig zu erweitern (je nach Rechenleistung des Systems).

Eine weitere Entwicklungsmöglichkeit bestünde in der Navigation. Im Moment ist nur eine rudimentäre Luftlinien-Punkt-zu-Punkt-Navigation realisiert. Diese könnte weiter verfeinert werden, indem zunächst eine Abfolge von Punkten angefliegen wird, die zuvor aufgenommen wurde. Weiterhin könnte man den Flugraum mit virtuellen Flugverbotszonen versehen um eine Kollision mit Gegenständen zu vermeiden. In einem weiteren Schritt könnte die Navigation diese Verbotszonen dann automatisch „umfliegen“ um an ein Ziel zu gelangen.

Da die Flugzeit der Quadcopters auf ca. 10min begrenzt ist, wäre es sehr vorteilhaft, wenn sich dieser bei geringem Akkuladestatus selbstständig zu einer Basisstation navigiert und dort auf einer Ladeplattform automatisch wieder aufgeladen wird. Somit könnte das autonome Verhalten des Quadcopters besser ausgenutzt werden.

7.3. Abschließende Worte

Die Arbeit mit dem Quadcopter hat mir sehr viel Spaß bereitet. Der Umbau und die ersten Flugversuche waren sehr herausfordernd und haben dafür gesorgt, dass die Erfolgserlebnisse um so intensiver waren. Vor allem war es schön die Fortschritte real zu sehen, anstatt nur die Theorie zu behandeln. Auch wenn das Ziel nicht ganz erreicht wurde, war es eine spannende Entwicklung und ich habe viel bei dieser Arbeit gelernt. Mein Interesse für das Themengebiet des Quadcopters wurde im Verlauf dieser Arbeit stetig gesteigert und wird mich auch weiterhin begeistern.

Literaturverzeichnis

- [3rdtech 2006] 3RDTECH: *HiBall - 3100 Wide-Area Tracker and 3D Digitizer*. 2006. – URL <http://www.3rdtech.com/images/hbds02v6forweb2.pdf>. – letzter Zugriff: 08.05.2010
- [3rdtech 2010a] 3RDTECH: *HiBall 6 DOF Tracker*. 2010. – URL <http://www.3rdtech.com/HiBall.htm>. – letzter Zugriff: 08.05.2010
- [3rdtech 2010b] 3RDTECH: *HiBall Ceiling*. 2010. – URL <http://www.3rdtech.com/images/ceilingplushiball114web.jpg>. – letzter Zugriff: 08.05.2010
- [3rdtech 2010c] 3RDTECH: *HiBall Key*. 2010. – URL <http://www.3rdtech.com/images/hiballkeycrop2web.jpg>. – letzter Zugriff: 08.05.2010
- [Agner 2010] AGNER, Stefan: *quadrocopter-schema-graphic*. 2010. – URL <http://uavp.agner.ch/wp-content/uploads/2010/02/quadrocopter-schema-graphic.png>. – letzter Zugriff: 08.05.2010
- [Behörde für Wissenschaft und Forschung Hamburg 2009] BEHÖRDE FÜR WISSENSCHAFT UND FORSCHUNG HAMBURG: *Nacht des Wissens*. 2009. – URL <http://nachtdeswissens.hamburg.de/>. – letzter Zugriff: 08.05.2010
- [Bernin 2009] BERNIN, Arne: *Räumliche Segmentierung mit Differenzbildern*. Hochschule für Angewandte Wissenschaften Hamburg, Projektausarbeitung. 2009. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2009-proj/bernin.pdf>
- [Bittner 2009] BITTNER, Walter: *Flugmechanik der Hubschrauber: Technologie, das flugdynamische System Hubschrauber, Flugstabilitäten, Steuerbarkeit*. Springer, 2009. – ISBN 354088971X, 9783540889717
- [Bluetooth SIG 2010] BLUETOOTH SIG: *Bluetooth*. 2010. – URL <http://www.bluetooth.com/German/Technology/Pages/default.aspx>. – letzter Zugriff: 08.05.2010
- [Bob O'Hara 2004] BOB O'HARA, Al P.: *IEEE 802.11 handbook: a designer's companion*. IEEE Standards Association, 2004. – ISBN 0738144495, 9780738144498

- [Böge 2007] BÖGE, Wolfgang: *Vieweg-Handbuch Elektrotechnik: Grundlagen und Anwendungen für Elektrotechniker*. Vieweg+Teubner Verlag, 2007. – ISBN 3834801364, 9783834801364
- [FH Bielefeld 2010] FH BIELEFELD: *Quadrokooper*. 2010. – URL http://www.lrh.fh-bielefeld.de/Labore/cad/pro_luftbild/Bilder/Quadrokooper.bmp. – letzter Zugriff: 08.05.2010
- [fischinger-messen 2010] FISCHINGER-MESSEN: *GPS 900 Bauvermessungssystem*. 2010. – URL <http://www.fischinger-messen.de/kap13.pdf>. – letzter Zugriff: 08.05.2010
- [Garmin 2007] GARMIN: *eTrex HC-Serie - Benutzerhandbuch*. 2007. – URL http://www8.garmin.com/manuals/eTrexLegendHCx_DEBenutzerhandbuch.pdf. – letzter Zugriff: 08.05.2010
- [grad-gon 2010] GRAD-GON: *Angebote*. 2010. – URL <http://www.grad-gon.de/angebot.htm>. – letzter Zugriff: 08.05.2010
- [Gregor 2006] GREGOR, Sebastian: *Entwicklung einer Hardwareplattform für die Ermittlung von Positionsdaten innerhalb von Gebäuden*, Hochschule für Angewandte Wissenschaften Hamburg, Bachelorarbeit, 2006. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/gregor.pdf>
- [Hahn 2008] HAHN, Tobias: *Techniken zur kamerabasierten Erkennung von frequenzcodierten Infrarotsendern für Indoorpositioningsysteme*, Hochschule für Angewandte Wissenschaften Hamburg, Bachelorarbeit, 2008. – URL <http://opus.haw-hamburg.de/volltexte/2008/627/pdf/Bachelorarbeit.pdf>
- [Ingo Busker 2009a] INGO BUSKER, Holger B.: *BrushlessCtrl*. 2009. – URL <http://www.mikrokooper.de/ucwiki/BrushlessCtrl>. – letzter Zugriff: 08.05.2010
- [Ingo Busker 2009b] INGO BUSKER, Holger B.: *MikroKopter - WebSVN - FlightCtrl - Revision 1378 - /tags/V0.76g/fc.c*. 2009. – URL <http://svn.mikrokooper.de/mikrowebsvn/filedetails.php?repname=FlightCtrl&path=%2Ftags%2FV0.76g%2Ffc.c>. – letzter Zugriff: 08.05.2010
- [Ingo Busker 2010a] INGO BUSKER, Holger B.: *en/SerialProtocol - Wiki: MikroKopter.de*. 2010. – URL <http://www.mikrokooper.de/ucwiki/en/SerialProtocol>. – letzter Zugriff: 08.05.2010
- [Ingo Busker 2010b] INGO BUSKER, Holger B.: *F2M03GXA*. 2010. – URL <http://www.mikrokooper.de/ucwiki/F2M03GXA>. – letzter Zugriff: 08.05.2010

- [Ingo Busker 2010c] INGO BUSKER, Holger B.: *Höhensensor*. 2010. – URL <http://www.mikrokopter.de/ucwiki/H%C3%B6hensensor>. – letzter Zugriff: 08.05.2010
- [Ingo Busker 2010d] INGO BUSKER, Holger B.: *Mikrokopter*. 2010. – URL <http://www.mikrokopter.de/ucwiki/>. – letzter Zugriff: 08.05.2010
- [Ingo Busker 2010e] INGO BUSKER, Holger B.: *Mikrokopter Einstieg*. 2010. – URL <http://www.mikrokopter.de/ucwiki/MikroKopterEinstieg#head-da41fa46384e734c1f5605363aba252c84aea1dc>. – letzter Zugriff: 08.05.2010
- [Ingo Busker 2010f] INGO BUSKER, Holger B.: *MK-Parameter / Zusammenfassung-Erläuterungen*. 2010. – URL http://www.mikrokopter.de/ucwiki/MK-Parameter/Zusammenfassung-Erl%C3%A4uterungen#ACC_.28Beschleunigungssensor.29. – letzter Zugriff: 08.05.2010
- [Kilrah 2008] KILRAH: *MikroKopter FPV tests*. 2008. – URL <http://www.rcmovie.de/video/924e16c772dcb718fcd0/MikroKopter-FPV-tests>. – letzter Zugriff: 08.05.2010
- [kowoma 2008] KOWOMA: *Die geschichtliche Entwicklung des GPS-Systems*. 2008. – URL <http://www.kowoma.de/gps/Geschichte.htm>. – letzter Zugriff: 08.05.2010
- [Lee 2007] LEE, Johnny C.: *Tracking Your Fingers with the Wiimote*. 2007. – URL <http://johnnylee.net/projects/wii/>. – letzter Zugriff: 08.05.2010
- [ligi 2010] LIGI: *DUBwise*. 2010. – URL <http://www.mikrokopter.de/ucwiki/en/DUBwise?action=show>. – letzter Zugriff: 08.05.2010
- [MagicMap 2008] MAGICMAP: *MagicMap Wiki*. 2008. – URL http://wiki.informatik.hu-berlin.de/nomads/index.php/MagicMap_Anwender_FAQs. – letzter Zugriff: 08.05.2010
- [MagicMap 2010a] MAGICMAP: *MagicMap Architektur*. 2010. – URL http://wiki.informatik.hu-berlin.de/nomads/images/0/00/MagicMapArchitektur_Schnittstellen.png. – letzter Zugriff: 08.05.2010
- [MagicMap 2010b] MAGICMAP: *MagicMap Demo*. 2010. – URL http://www2.informatik.hu-berlin.de/rok/MagicMap/_images/screenshots/MM_Screenshot070306.jpg. – letzter Zugriff: 08.05.2010
- [MagicMap 2010c] MAGICMAP: *MagicMap System*. 2010. – URL <http://www2.informatik.hu-berlin.de/rok/MagicMap/system.htm>. – letzter Zugriff: 08.05.2010

- [MathWorks 2010] MATHWORKS: *Matlab*. 2010. – URL <http://www.mathworks.de/products/matlab/>. – letzter Zugriff: 08.05.2010
- [microdrones GmbH 2010] MICRODRONES GMBH: *microdrones GmbH | your eye in the sky*. 2010. – URL <http://www.microdrones.com/startseite.php>. – letzter Zugriff: 08.05.2010
- [MIT Computer Science and Artificial Intelligence Lab Cambridge, MA 02139 2005a] MIT COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE LAB CAMBRIDGE, MA 02139: *Cricket Beacon*. 2005. – URL <http://cricket.csail.mit.edu/pictures/v2-cf.jpg>. – letzter Zugriff: 08.05.2010
- [MIT Computer Science and Artificial Intelligence Lab Cambridge, MA 02139 2005b] MIT COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE LAB CAMBRIDGE, MA 02139: *Cricket Deployment*. 2005. – URL http://cricket.csail.mit.edu/pictures/cricket_deployment.jpg. – letzter Zugriff: 08.05.2010
- [MIT Computer Science and Artificial Intelligence Lab Cambridge, MA 02139 2005c] MIT COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE LAB CAMBRIDGE, MA 02139: *Cricket v2 User Manual*. 2005. – URL <http://cricket.csail.mit.edu/v2man.pdf>. – letzter Zugriff: 08.05.2010
- [MOBOTIX 2010a] MOBOTIX: *MOBOTIX - Hemispheric Kamera - Die Q24 im Detail*. 2010. – URL https://www.mobotix.com/ger_DE/Produkte/Hemispheric-Kamera/Die-Q24-im-Detail. – letzter Zugriff: 08.05.2010
- [MOBOTIX 2010b] MOBOTIX: *Raumuebersicht*. 2010. – URL https://www.mobotix.com/var/storage/images/media/images/products/produktseiten/hemispheric/raumuebersicht.jpg/1660575-1-ger/raumuebersicht.jpg_formatVGA.jpg. – letzter Zugriff: 08.05.2010
- [Paramvir Bahl 2000] PARAMVIR BAHL, Venkata N. P.: *RADAR: An In-Building RF-based User Location and Tracking System*. 2000. – URL <http://research.microsoft.com/en-us/um/people/padmanab/papers/infocom2000.pdf>. – letzter Zugriff: 08.05.2010
- [Paric Measurement 2006a] PARIC MEASUREMENT: *Paric*. 2006. – URL <http://www.panic.co.nz/FAQ.html>. – letzter Zugriff: 08.05.2010
- [Paric Measurement 2006b] PARIC MEASUREMENT: *Paric System*. 2006. – URL http://www.panic.co.nz/images/tracker_rec.jpg. – letzter Zugriff: 08.05.2010

- [Pfaff 2007] PFAFF, Thomas: *Entwicklung eines PDA-basierten Indoor-Navigationssystems*, Hochschule für Angewandte Wissenschaften Hamburg, Bachelorarbeit, 2007. – URL http://opus.haw-hamburg.de/volltexte/2007/383/pdf/BA_TP_ohne.pdf
- [Ramachandra 2000] RAMACHANDRA, K. V.: *Kalman filtering techniques for radar tracking*. CRC Press, 2000. – ISBN 0824793226, 9780824793227
- [Reichelt Elektronik 2010] REICHELT ELEKTRONIK: *Reichelt*. 2010. – URL <http://www.reichelt.de>. – letzter Zugriff: 08.05.2010
- [Rempel 2008] REMPEL, Olaf: *Entwicklung einer inertialen Lageregelung für einen Quadrocopter auf ARM7 Basis*, Hochschule für Angewandte Wissenschaften Hamburg, Bachelorarbeit, 2008. – URL <http://opus.haw-hamburg.de/volltexte/2008/591/pdf/bachelorarbeit.pdf>
- [Thielicke 2009a] THIELICKE, W.: *Tricopter position-hold via webcam + target tracking*. 2009. – URL <http://vimeo.com/5905003>. – letzter Zugriff: 08.05.2010
- [Thielicke 2009b] THIELICKE, William: *Tracking IR LEDs with Matlab*. 2009. – URL <http://www.villalachouette.de/william/krim/tricopter/tricoptertracking.htm>. – letzter Zugriff: 12.10.2009
- [Thielicke 2010a] THIELICKE, William: *Shrediquette - a tricopter by William Thielicke*. 2010. – URL <http://shrediquette.blogspot.com/>. – letzter Zugriff: 08.05.2010
- [Thielicke 2010b] THIELICKE, William: *Shrediquette - HowTo*. 2010. – URL <http://www.villalachouette.de/william/krim/tricopter/Shrediquette%20HowTo%20by%20W.%20Thielicke.pdf>. – letzter Zugriff: 08.05.2010
- [u-blox 2010] U-BLOX: *SuperSense - the ultimate in indoor GPS tracking*. 2010. – URL <http://u-blox.com/en/supersense.html>. – letzter Zugriff: 08.05.2010
- [ubisense 2007a] UBISENSE: *System Overview*. 2007. – URL <http://ubien.mangowebserver.com/media/pdf/Ubisense%20System%20Overview%20May%202007.pdf>. – letzter Zugriff: 08.05.2010
- [ubisense 2007b] UBISENSE: *Ubisense - SlimTag*. 2007. – URL <http://www.ubisense.de/media/pdf/Ubisense-SlimTag-GER.pdf>. – letzter Zugriff: 08.05.2010
- [ubisense 2007c] UBISENSE: *Ubisense Location Engine*. 2007. – URL <http://www.ubisense.de/media/pdf/Ubisense%20Software%20LocEn%20GER%20V1.0.pdf>. – letzter Zugriff: 10.04.2010

- [ubisense 2010] UBISENSE: *Arbeitsplatz :: Ubisense German Site*. 2010. – URL <http://www.ubisense.de/content/25.html>. – letzter Zugriff: 08.05.2010
- [Uwe Brinkschulte 2007] UWE BRINKSCHULTE, Theo U.: *Mikrocontroller und Mikroprozessoren*. Springer, 2007. – ISBN 3540468013, 9783540468011
- [Welzel 2007] WELZEL, Thomas: *Entwicklung einer autonomen Flugsteuerung für einen Indoor-Koaxialhubschrauber unter Verwendung von IMAPS*, Hochschule für Angewandte Wissenschaften Hamburg, Diplomarbeit, 2007. – URL <http://users.informatik.haw-hamburg.de/~kvl/welzel/Diplomarbeit.pdf>
- [Wikipedia 2006] WIKIPEDIA: *Quadrocopter Schema*. 2006. – URL http://upload.wikimedia.org/wikipedia/commons/e/e2/Quadrocopter_schema.png. – letzter Zugriff: 08.05.2010
- [Wikipedia 2010a] WIKIPEDIA: *Pseudocode*. 2010. – URL http://en.wikipedia.org/wiki/PID_controller. – letzter Zugriff: 08.05.2010
- [Wikipedia 2010b] WIKIPEDIA: *Quadrocopter*. 2010. – URL <http://de.wikipedia.org/wiki/Quadrocopter>. – letzter Zugriff: 08.05.2010
- [Willa1980 2009] WILLA1980: *Autonomous indoor waypoint navigation with a tricopter*. 2009. – URL http://www.youtube.com/watch?v=4ktM_wjd3JQ. – letzter Zugriff: 08.05.2010

Glossar

AOA Angle of Arrival – Einfallswinkel der Funksignale wird ausgewertet

Bluetooth vgl. [[Bluetooth SIG \(2010\)](#)]

Bug Fehler in einem Programm

FOV Abk. 'Field of View' (Sichtbereich der Kamera)

FPV engl. First-Persons-View = Immersionsflug

Gier Drehbewegung um die Hochachse

GPS Global Positioning System

GUI Graphic User Interface – Grafische Benutzeroberfläche zu Interaktion mit dem Programm

HAW Hochschule für Angewandte Wissenschaften Hamburg

Heading engl. Ausrichtung

Indoor engl. für 'in einem Gebäude'

IR Abk. Infrarot

LED Light Emitting Diode

Nick Drehbewegung um die Querachse

P2P Peer to Peer – Direkte Verbindung zwischen Clients ohne Verwaltung durch einen Server

PID engl. proportional–integral–derivative controller

PWM Pulsweitenmodulation, vgl. ([Uwe Brinkschulte, 2007](#), S. 170)

Quadcopter Abk. für 'Quadrocopter' [[Wikipedia \(2010b\)](#)], auch bekannt als 'Mikrokopter' [[Ingo Busker \(2010d\)](#)]

Roll Drehbewegung um die Längsachse

TDOA Time Difference of Arrival – Die unterschiedlichen Signallaufzeiten werden ausgewertet

Tricopter Modellhubschrauber mit drei Rotoren. [[Thielicke \(2010b\)](#)]

UWB Ultra Wide Band

WLAN Wireless Local Area Network – vgl. [[Bob O'Hara \(2004\)](#)]

A. Inhalt der CD

A.1. Quellcode

A.1.1. Matlab

- GUI2.m
- GUI2.fig
- StartCam.m
- independentTrack_timer.m
- settings.mat
- figureclosingTrack.m

A.1.2. Mikrokopter

- fc.c

A.1.3. Latenzmessung

- Latenz.bas

A.2. PDF Dokumente

- Diese Bachelorarbeit im PDF-Format

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 12. Mai 2010

Ort, Datum

Unterschrift