Julius-Maximilians-Universität Würzburg
Fakultät für Mathematik und Informatik

Informationstechnik für Luft- und Raumfahrt
Lehrstuhl für Informatik 8 Prof. Dr. Sergio Montenegro





Bachelorarbeit

Implementierung und Evaluierung einer Positionsregelung unter Verwendung des optischen Flusses

Vorgelegt von

Michael Strohmeier

Matr.-Nr.: 1703406

Prüfer: Prof. Dr. Sergio Montenegro

Betreuende wissenschaftliche Mitarbeiter: Dipl.-Ing. Nils Gageik

Würzburg, 14. 09. 2012

Erklärung

Ich versichere, dass ich die vorliegende Arbeit einschließlich aller beigefügter Materialien selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe.
Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Werken entnommen sind, sind in jedem Einzelfall unter Angabe der Quelle deutlich als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form noch nicht als Prüfungsarbeit eingereicht worden.
Mir ist bekannt, dass Zuwiderhandlungen gegen diese Erklärung und bewusste Täuschungen die Benotung der Arbeit mit der Note 5.0 zur Folge haben kann.

Würzburg, 14. 09. 2012

_____

Michael Strohmeier

Aufgabenstellung

Die Fortschritte im Bereich Sensorik und Mikrotechnik ermöglichen heutzutage den kostengünstigen Bau kleiner unbemannter Luftfahrzeuge (engl. **U**nmanned **A**erial **V**ehicle (UAV), Drohne) wie Quadrocopter. Die Forschung und Entwicklung dieser Systeme wurde in den letzten Jahren aufgrund der vielfältigen Anwendungsmöglichkeiten stark vorangetrieben. Wenngleich im Bereich UAV viel geforscht wurde, ist das Thema Autonomes Flugobjekt längst noch nicht vollständig behandelt. Insbesondere der Indoor-Betrieb ist aufgrund fehlender absoluter Positionsstützung durch GPS problematisch. Der Aufbau eines eigenen autonomen Systems wird daher am Lehrstuhl VIII Informationstechnik für Luft- und Raumfahrt der Julius-Maximilians-Universität Würzburg erforscht und erprobt. Im Rahmen dieses Forschungsvorhabens ist ein System zur Positionsregelung zu entwickeln.

Hauptaugenmerk dieser Arbeit ist die Bestimmung einer Positionsänderung über Boden durch Bestimmung des optischen Flusses mit Hilfe einer Kamera. Teil der Arbeit ist die treibermäßige Anbindung einer Kamera sowie die Berechnung des optischen Flusses aus Echtzeitbildern. Im zweiten Schritt soll ein Regler zur Positionsregelung implementiert werden. Die zu entwickelte Software ist in das bestehende System einzubinden. Schließlich ist das System zur Positionsbestimmung und -regelung umfangreich zu evaluieren. Zur Aufgabe gehört eine ausführliche Dokumentation der Software.

Aufgabenstellung (Stichpunktartig):

- Treiber Kamera
- Bestimmung optischer Fluss
- Ausführliche Erklärung Algorithmus Optischer Fluss
- Positionsregelung
- Dokumentation

Zusammenfassung

Der Schwerpunkt dieser Arbeit liegt in der Entwicklung eines Systems zur Positionsregelung eines autonomen Quadrocopters zur Innenraumerkundung. Der Quadrocopter wird im Rahmen des AQopterI8 Projektes am Lehrstuhl VIII Informationstechnik für Luft- und Raumfahrt der Julius-Maximilians Universität Würzburg entwickelt. Da der Quadrocotper in Gebäuden, also in einer für GPS unzugänglichen Umgebung, zum Einsatz kommen soll, kann die Positionsbestimmung nicht mit Hilfe von GPS durchgeführt werden. Es wurde daher ein optisches System entwickelt und an der Unterseite des Quadrocopters befestigt. Das optische Systems nimmt Bildsequenzen auf, aus denen durch die Berechnung des optischen Flusses die Eigenbewegung des Quadrocopters ermittelt wird. Diese Informationen werden zur relativen Positionsbestimmung des Quadrocopters verwendet. Auf Basis dieser Positionsdaten wird ein PID-Regler implementiert, der das autonome Anfliegen und Halten einer beliebigen Position ermöglicht. Bei der Realisierung des optischen Systems wurden verschiedene Ansätze untersucht und bewertet. Der zuverlässigste Ansatz wurde schließlich in den Quadrocopter integriert und ausführlich evaluiert. Die abschließende Evaluierung des Gesamtsystems zeigt, dass der Quadrocopter selbstständig Positionen anfliegen und für mindestens sechs Minuten halten kann.

Stichworte: Quadrocopter, Innenraumerkundung, Optischer Fluss, Positionsregelung

Inhaltsverzeichnis

1 Einleitung **1**

2 Grundlagen **3**

2.1 Optischer Fluss 4
2.2 Beziehung zum Bewegungsfeld 5
2.3 Das Blendenproblem 6
2.4 Kontinuitätsgleichung des optischen Flusses 6
2.5 Methoden zur Bestimmung des optischen Flusses 8
2.5.1 Frequenzbasierte Methoden 8
2.5.2 Korrelationsmethoden 8
2.5.3 Differentielle Methoden 8
2.5.3.1 Globale Methode: Die Horn-Schunck Methode 9
2.5.3.2 Lokale Methode: Die Lucas-Kanade Methode 9
2.6 Verwendeter Algorithmus 10
2.6.1 Theoretischer Ansatz 11
2.6.2 Praktische Umsetzung 12

3 Stand der Technik **15**

3.1 Systeme zur Positionsbestimmung 15
3.1.1 Global Positioning System 16
3.1.2 Externe Tracking Systeme 16
3.1.3 Integrierte optische Systeme 16
3.2 Implementierung integrierter optischer Systeme 17
3.2.1 Optische Flusssensoren 17
3.2.2 System aus Bildsensor und Prozessor 18
3.2.2.1 CMOS Sensoren 18
3.2.2.2 CCD Sensoren 19
3.2.2.3 Spezielle Vision Chips 19
3.2.2.4 Mikrokontroller 20
3.2.2.5 Field Programmable Gate Arrays 20

3.2.3 Konsequenzen für die vorliegende Arbeit 21

4 Konzept 23

4.1 Überblick 23
4.1.1 Systemanforderungen 25
4.1.2 Entwicklungsphasen 25
4.2 Systemkomponenten 26
4.2.1 Die Kontrollsoftware 26
4.2.2 Das Optische System zur Positionsbestimmung 26
4.2.3 Der Positionsregler 27

5 Implementierung 29

5.1 Kontrollsoftware zur Evaluierung und Steuerung 29
5.1.1 Evaluierungsfunktionen 30
5.1.2 Steuerungsfunktionen 31
5.2 Implementierung eines optischen Systems 32
5.2.1 Centeye Vision Chip Tam2 32
5.2.1.1 Implementierung der Schnittstelle 32
5.2.1.2 Berechnung des optischen Flusses 33
5.2.1.3 Bewertung des Systems 33
5.2.2 4D Systems uCAM 34
5.2.2.1 Implementierung der Schnittstelle 34
5.2.2.2 Bewertung des Systems 35
5.2.3 CMUCam 4 35
5.2.4 OV 7670 und STM32 F4 36
5.2.4.1 Implementierung der Schnittstellen 36
5.2.4.2 Berechnung des optischen Flusses 38
5.2.4.3 Speicherauslastung und Bildqualität 38
5.2.4.4 Bewertung des Systems 39
5.2.5 ADNS 3080 40
5.2.5.1 Implementierung der Schnittstelle 41
5.2.5.2 Berechnung des optischen Flusses 41
5.2.5.3 Bewertung des Systems 41
5.2.6 Auswahl und Integration des optischen Systems 42
5.3 Implementierung des Positionsreglers 43
5.3.1 Programmablauf des Positionsreglers 43
5.3.1.1 Berechnung der P, I und D Anteile 44
5.3.1.2 Berechnung der Sollwinkel für die Lageregelung 45

5.3.2 Einstellen des Positionsreglers 45
5.4 Kalibrierung des fertigen Systems 46
5.4.1 Problematik bei der Kalibrierung des Systems 46
5.4.2 Messung zur Kalibrierung des Systems 47
5.4.2.1 Versuchsbeschreibung 47
5.4.2.2 Auswertung 47

6 Evaluierung 49
6.1 Evaluierung der optischen Systeme 49
6.1.1 Versuchsbeschreibung 49
6.1.2 Auswertung OV 7670 und STM32 F4 50
6.1.3 Auswertung ADNS 3080 und AVR UC3A 51
6.1.4 Vergleich der Systeme 52
6.2 Evaluierung des fertigen Systems 52
6.2.1 Statisches Regelverhalten 52
6.2.1.1 Versuchsbeschreibung 52
6.2.1.2 Auswertung 53
6.2.2 Dynamisches Regelverhalten - Einfacher Positionsanflug 55
6.2.2.1 Versuchsbeschreibung 55
6.2.2.2 Auswertung 55
6.2.3 Dynamisches Regelverhalten - Komplexer Positionsanflug 57
6.2.3.1 Versuchsbeschreibung 57
6.2.3.2 Auswertung 58

7 Diskussion und Ausblick 61

8 Inhalt der CD 63

9 Literaturverzeichnis 65

10 Quellenverzeichnis 69

11 Abkürzungsverzeichnis 71

12 Abbildungsverzeichnis 73

1 Einleitung

Am Lehrstuhl VIII Informationstechnik für Luft- und Raumfahrt der Julius-Maximilians-Universität Würzburg wird im Rahmen des AQopterI8 Projektes ein Quadrocopter zur Innenraumerkundung entwickelt (vgl. Abbildung 1.1). Ein Quadrocopter ist eine Drohne (engl. **U**nmanned **A**erial **V**ehicle (UAV)) mit vier Propellern, die sich manuell per Funksteuerung oder autonom im dreidimensionalen Raum bewegen kann. In den letzten Jahren haben sich zahlreiche Anwendungsfelder für den Einsatz von Drohnen ergeben. Neben der zunehmenden Beliebtheit im Hobby- und Modellflugbereich werden Quadrocopter vor allem zur Aufnahme von Kamerabildern aus der Luft verwendet. In einigen Bundesländern werden Drohnen bereits zur Überwachung größerer Menschenansammlungen durch die Polizei eingesetzt[SpiegelOnline, 2012]. Bei den meisten Drohnen, die zur Zeit im Einsatz sind, handelt es sich überwiegend um manuell gesteuerte Quadrocopter, die im Freien fliegen.



Abbildung 1.1: Quadrocopter des AQopterI8 Projektes

Ziel des AQopterI8 Projektes ist „*die Entwicklung eines autonomen Quadrocopters, der sich auch in a priori unbekannten Gebäudekomplexen zurechtfindet und in der Lage ist, Erkundungs- und Suchaufgaben durchzuführen, sowie die Ergebnisse an den Nutzer (per Funk) zu schicken.*“[AQopterI8, 2012]. Autonome Quadrocopter könnten beispielsweise zur Unterstützung der Feuerwehr, des technischen Hilfswerks oder in Krisengebieten, die für den Menschen nur schwer zugänglich sind, eingesetzt werden.

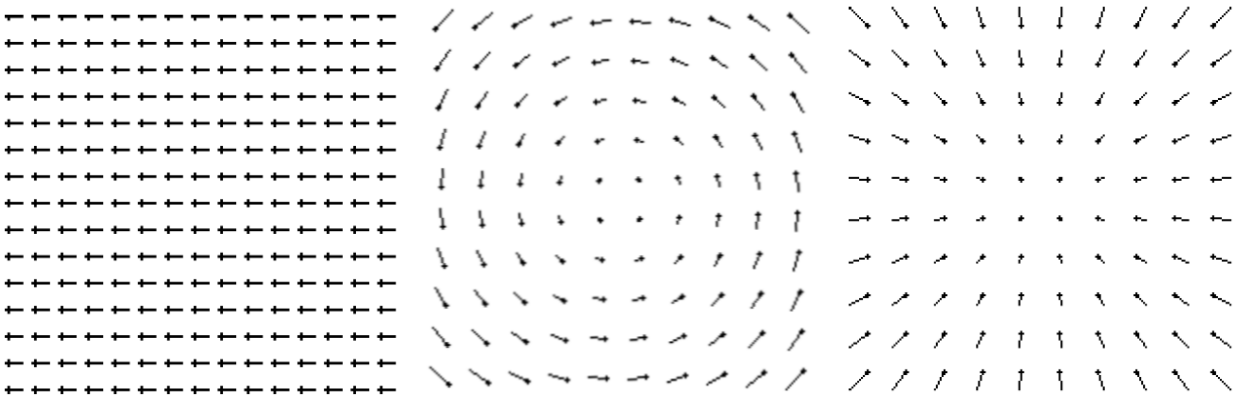
Damit ein Quadrocopter selbstständig in einem unbekannten Raum fliegen kann, muss er seine Position exakt bestimmen und regeln können. Eine zuverlässige Positionsregelung ist besonders zur Driftkompensation beim Positionshalten und zum autonomen Anfliegen von Positionen wichtig. Anders als bei vielen bereits existierenden Systemen darf ein autonomer Quadrocopter dabei nicht auf externe Hilfsmittel, wie beispielsweise ein optisches Tracking System, zurückgreifen. Es muss daher ein System entwickelt werden, das es dem Quadrocopter ermöglicht, sich selbst im Raum zu lokalisieren. Bei der Realisierung eines solchen Systems ist neben einer Koppelnavigation (engl. Dead Reckoning) mit Hilfe von Inertialsensoren auch eine Abstandsmessung durch Ultraschall- oder Infrarotsensoren zu den Wänden des Raumes denkbar, um die Position des Quadrocopters zu bestimmen. Eine weitere Möglichkeit ist die Verwendung eines optischen Systems, das sich an der Unterseite des Quadrocopters befindet und dessen Bewegungsrichtung optisch bestimmt. Im Gegensatz zu den Interialsensoren einer Koppelnavigation gibt es bei der Verwendung von optischen Sensoren keinen Sensordrift, was eine sehr genaue Messung der Eigenbewegung des Quadrocopters erlaubt. Darüberhinaus wird, im Gegensatz zur Positionsbestimmung durch Abstandssensoren, nur ein einziger Sensor für das optische System benötigt.

Aufgrund der eben genannten Vorteile eines optischen Systems wird der Quadrocopter des AQopterI8 Porjektes in dieser Arbeit um ein solches System zur Positionsbestimmung und Positionsregelung erweitert. Durch eine zum Boden gerichtete Kamera an der Unterseite der Drohne wird während des Flugs eine Bildsequenz aufgenommen. Aus dieser Bildsequenz wird anschließend die Eigenbewegung des Systems berechnet. Diese Berechnung basiert auf dem Prinzip des optischen Flusses, einem Verfahren zur Approximation von dreidimensionalen Objektbewegungen in einem zweidimensionalen Bild. Das fertige System ist in der Lage, autonom eine Position über mehrere Minuten zu halten und selbstständig verschiedene Positionen anzufliegen.

2 Grundlagen

Im Bereich der Bildanalyse und -verarbeitung ermöglichen Bildsequenzen die zeitliche Veränderung von Objekten und Szenen zu erkennen und zu charakterisieren. Bildsequenzanalysen eröffnen ein breites Spektrum an wissenschaftlichen und technischen Anwendungen. Zur Untersuchung biologischer Wachstumsprozesse, der Überwachung von industriellen Prozessen, sowie bei der Steuerung autonomer Fahrzeuge, Roboter und insbesondere von UAVs erzielt dieses Verfahren gute Ergebnisse[Jiang, 2005].

Ziel der Bildsequenzanalyse ist es, das sogenannte Bewegungsfeld aus einer Folge von Bildern zu extrahieren. Unter einem Bewegungsfeld versteht man die Projektion von dreidimensionalen, physikalischen Bewegungen mit Hilfe eines optischen Systems in eine zweidimensionale Bildebene. Das Bewegungsfeld beschreibt also dreidimensionale Bewegungen durch Vektoren für jedes einzelne Bildelement innerhalb einer Ebene. Abbildung 2.1 veranschaulicht exemplarisch die Bewegungsfelder von elementaren Bewegungen. Abbildung 2.1(a) entsteht durch eine translatorische Bewegung parallel zu einer Ebene, Abbildung 2.1(b) durch die Rotation senkrecht zu einer Ebene. Das Bewegungsfeld in Abbildung 2.1(c) entsteht durch eine senkrechte Bewegung in Richtung der betrachteten Ebene. Da das Bewegungsfeld jedoch nicht direkt beobachtet werden kann, solange nicht bekannt ist, wie sich einzelne Bildelemente bewegen, wird der optische Fluss als Approximation des Bewegungsfeldes eingeführt [Westfeld, 2004].



(a) Translation (b) Rotation (c) Zoom in

Abbildung 2.1: Bewegungsfelder von elementaren Bewegungen [Jiang, 2005]

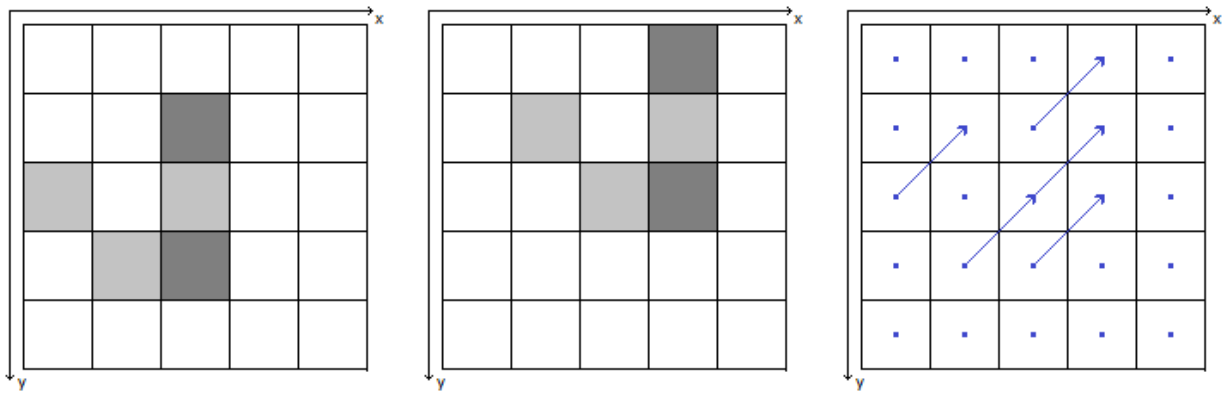
2.1 Optischer Fluss

Horn und Schunck [1980] beschreiben den optischen Fluss wie folgt:

> „Optical flow is the distribution of apparent velocities of movement of brightness patterns in an image.“

Der optische Fluss ist demnach als die Verteilung von Geschwingigkeitsvektoren definiert, die durch die scheinbare Bewegung von Helligkeitsmustern in einem Bild entstehen. Er kann analog zu dem Bewegungsfeld durch ein Vektorfeld repräsentiert werden, das für jeden betrachteten Bildpunkt Bewegungsrichtung und -geschwindigkeit beschreibt. Eine Änderung der Helligkeitsmuster im Bild kann unterschiedliche Ursachen haben. Einerseits kann eine Veränderung durch die Bewegung betrachteter Objekte erzeugt werden, andererseits auch durch die Bewegung des optischen Systems selbst. Eine Veränderung der Umgebungsbeleuchtung führt zwangsweise zu einer Änderung der beobachteten Helligkeitsmuster und vermittelt so den Eindruck von Bewegung in der beobachteten Szene.

Betrachtet man nun die Helligkeitsverteilungen $I(x, y, t)$ und $I(x, y, t + \Delta t)$, wobei $I(x, y, t)$ die Helligkeit I des Bildpunktes (x, y) zum Zeitpunkt t beschreibt, zweier aufeinanderfolgender Bilder einer Bildsequenz, so ist der optische Fluss äquivalent zu dem Verschiebungsvektorfeld, mit dem $I(x, y, t)$ auf $I(x, y, t + \Delta t)$ abgebildet wird. Abbildung 2.2 soll eine solche Bildfolge und den daraus resultierenden optischen Fluss beispielhaft verdeutlichen. In Abbildung 2.2(a) ist die Helligkeitsverteilung I für alle Bildpunkte x und y zum Zeitpunkt t zu sehen. Analog dazu ist in Abbildung 2.2(b) die Helligkeitsverteilung I für alle Bildpunkte x und y zum Zeitpunkt $t + \Delta t$ dargestellt. Die Abbildung 2.2(c) zeigt schließlich den aus der Bildsequenz resultierenden optischen Fluss als Vektorfeld.

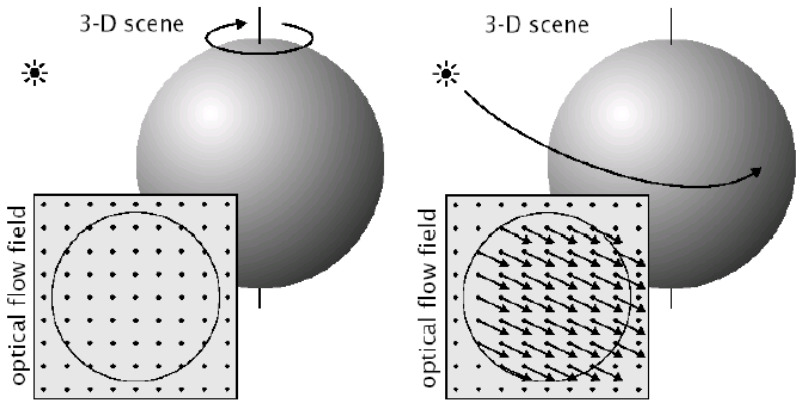


(a) $I(x, y, t)$ (b) $I(x, y, t + \Delta t)$ (c) Optischer Fluss

Abbildung 2.2: Einfache Darstellung des optischen Flusses

2.2 Beziehung zum Bewegungsfeld

Wie bereits zuvor erwähnt, kann der optische Fluss nicht nur durch die relative Bewegung von beobachteter Szene und optischen System entstehen, sondern auch durch eine Änderung der Szenenbelichtung. Helligkeitsänderungen können demnach auch ohne Objektbewegung in der beobachteten Szene entstehen. Im Gegensatz dazu muss nicht jede Objektbewegung eine Helligkeitsänderung im Bild bewirken. Steinmüller [2008] beschriebt ein Beispiel, das dieses Problem sehr anschaulich durch die Betrachtung von zwei extrem Situationen verdeutlicht. In Abbildung 2.3(a) ist eine rotierende Kugel mit konstanter Oberflächenbeschaffenheit zu sehen. In dieser Szene ist die Lichtquelle (Sonnensymbol) fixiert, die Belichtung somit also konstant. Obwohl das Bewegungsfeld von 0 verschieden ist, bleibt das optische Flussfeld gleich 0, da keine sichtbaren Helligkeitsänderungen wahrnehmbar sind. Abbildung 2.3(b) zeigt die selbe Kugel ohne Rotation. Anders als zuvor ist die Lichtquelle nicht fixiert, sondern dreht sich um die Kugel. Da offensichtliche keine Objektbewegung stattfindet, ist in diesem Fall das Bewegungsfeld gleich 0. Da sich die Umgebungsbeleuchtung jedoch ändert, verändern sich auch die beobachteten Helligkeitsmuster. Das optische Flussfeld ist also von 0 verschieden.



(a) Optischer Fluss $= 0$, Bewegungsfeld $\neq 0$

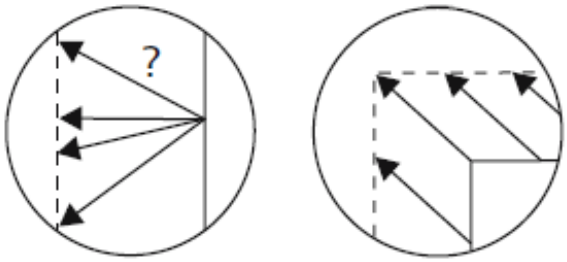
(b) Optischer Fluss $\neq 0$, Bewegungsfeld $= 0$

Abbildung 2.3: Beziehung zwischen Bewegungsfeld und optischen Fluss [ComputerVision, 2002]

Damit der optische Fluss als aussagekräftige Schätzung für das wahre Bewegungsfeld verwendet werden kann, müssen zuvor mehrere Annahmen erfüllt werden. Beauchemin und Barron [1995] fordern an dieser Stelle eine einheitliche, konstante Beleuchtung, eine Lambert'sche Oberflächenreflexion, eine reine translatorische Bewegung parallel zur Bildebene und ausreichend kontrastreiche Texturen. Diese Bedingungen können allerdings in der Realität niemals ganzheitlich erfüllt werden. Stattdessen geht man davon aus, dass sie lokal in der Szene und damit auch lokal in der Bildebene eingehalten werden.

2.3 Das Blendenproblem

Ein weiteres, gravierendes Problem bei der Approximation des Bewegungsfeldes durch den optischen Fluss ist das sogenannte Blenden- oder Aperturproblem. Ein durch das optische System beschränkter Bildausschnitt liefert nicht immer ausreichend Informationen über die momentan stattfindende Bewegung. Abbildung 2.4(a) zeigt die möglichen Verschiebungen einer Kante entlang verschiedener Vektoren. Die Verschiebung ist nicht eindeutig, da lediglich der Betrag des Vektors senkrecht zur Kante bestimmt werden kann, nicht aber wie weit das Objekt parallel zur Kante verschoben wurde. Ein Punkt der durchgezogenen Kante kann nicht eindeutig einem Punkt der gestrichelten Kante zugeordnet werden. Erst wenn zusätzliche Informationen zu der beobachteten Szene hinzugefügt werden, wie beispielsweise eine Ecke des Objektes (vgl. Abbildung 2.4(b)), lässt sich die Verschiebung eindeutig bestimmen[Jähne, 2005].



(a) Mehrdeutige Verschiebung (b) Eindeutige Verschiebung

Abbildung 2.4: Veranschaulichung des Blendenproblems [Jähne, 2005]

2.4 Kontinuitätsgleichung des optischen Flusses

Um den optischen Fluss aus einer Bildsequenz zu bestimmen, werden zunächst die Helligkeitsverteilungen $I(x, y, z)$ und $I(x+\delta x, y+\delta y, t+\delta t)$ zweier aufeinander folgender Bilder betrachtet. Unter der Annahme, dass die Helligkeitsmuster der zeitveränderlichen Bildbereiche für zumindest eine kurze Dauer annähernd konstant sind, ergibt sich für zueinander korrespondierende Bildpunkte folgende Gleichung [Horn und Schunck, 1980]:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t) \tag{2.1}$$

Eine Taylorreihenentwicklung der rechten Seite der Gleichung 2.1 unter Vernachlässigung von Termen zweiter und höherer Ordnung führt zu:

$$I(x, y, t) = I(x, y, t) + \delta x \frac{\partial I}{\partial x} + \delta y \frac{\partial I}{\partial y} + \delta t \frac{\partial I}{\partial t} \tag{2.2}$$

Durch weitere Äquivalenzumformungen erhält man schließlich mit $u = \frac{\delta x}{\delta t}$ und $v = \frac{\delta y}{\delta t}$ die Kontinuitätsgleichung des optischen Flusses (engl.: optical flow contraint equation, OFCE):

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0 \tag{2.3}$$

Ersetzt man die partiellen Ableitungen durch die Abkürzungen I_x, I_y und I_t erhält man:

$$I_x u + I_y v + I_t = 0 \tag{2.4}$$

Die partiellen Ableitungen I_x, I_y und I_t aus Gleichung 2.4 lassen sich direkt aus den zwei Ausgangsbildern berechnen. I_x und I_y beschreiben die räumliche, I_t hingegen die zeitliche Änderung des Helligkeitsmusters. u und v beschreiben den optischen Fluss, also die Verschiebungsgeschwindigkeit der Bildpunkte in x- bzw. y-Richtung. Um den optischen Fluss zu erhalten, müssen also die Größen u und v bestimmt werden. Es handelt sich hierbei um ein unterbestimmtes Problem, da eine Gleichung mit zwei Unbekannten zu lösen ist. Die Gleichung 2.4 beschränkt den optischen Fluss auf eine Gerade (engl.: constraint line), liefert aber keine eindeutige Lösung (vgl. Abbildung 2.5).

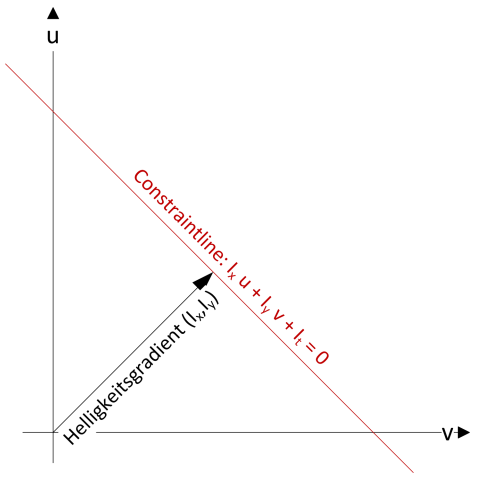


Abbildung 2.5: Grafische Darstellung der Kontinuitätsgleichung des optischen Flusses

Das bedeutet, dass lediglich die Bewegungskomponente $\mathbf{v}_\perp$ parallel zum Helligkeitsgradienten (also senkrecht zur constraint line) bestimmt werden kann [Beauchemin und Barron, 1995]. Die Komponente des optischen Flusses parallel zum Helligkeitsgradienten (I_x, I_y) beträgt demnach:

$$\mathbf{v}_\perp = -\frac{I_t}{\sqrt{I_x^2 + I_y^2}} \tag{2.5}$$

Dieses Phänomen ist äquivalent zu dem in Kapitel 2.3 beschriebenen Blendenproblem.

2.5 Methoden zur Bestimmung des optischen Flusses

Um den optischen Fluss schließlich zu bestimmen, existieren verschiedene Methoden und Algorithmen. Je nach Autor werden diese unterschiedlich kategorisiert (vgl. Barron et al. [1994]; Jähne [2005]). Prinzipiell unterscheidet man jedoch zwischen frequenzbasierten, korrelationsbasierten und differentiellen Methoden:

2.5.1 Frequenzbasierte Methoden

Frequenzbasierte Methoden beruhen auf der Fouriertransformation einer translatorischen Helligkeitsverschiebung, wie in Gleichung 2.1 beschrieben. Sie benutzen im Fourierbereich Filter, die sowohl an die Geschwindigkeit als auch die Orientierung angepasst sind [Beauchemin und Barron, 1995]. Allerdings sind diese Methoden nur schwer realisierbar [Ebers, 2005] und mit einem hohen Rechenaufwand verbunden [Barron et al., 1994].

2.5.2 Korrelationsmethoden

Die Korrelationsmethoden basieren hingegen auf einem völlig anderen Prinzip. Das erste Bild der Bildsequenz wird hierbei innerhalb eines bestimmten Suchbereichs mit dem zweiten, verschobenen Bild verglichen. Es wird nach der Position mit der optimalen Ähnlichkeit zwischen beiden Bildern innerhalb des Suchbereichs gesucht. Das Ähnlichkeitsmaß wird unempfindlich gegenüber Beleuchtungsänderungen gewählt. Analog zum frequenzbasierten Ansatz ist diese Methode ebenfalls mit einem hohen Rechenaufwand verbunden[Jähne, 2005]. Die Korrelationsmethode eignet sich besonders dann, wenn der zeitliche Abstand zwischen den Bildern groß ist[Beauchemin und Barron, 1995].

2.5.3 Differentielle Methoden

Die differentiellen Methoden bilden den klassischen Ansatz zur Approximation der Objektbewegung aus zwei aufeinander folgenden Bildern. Ausgangspunkt dieses Ansatzes ist die Kontinuitätsgleichung 2.4. Da diese jedoch, wie in Abschnitt 2.4 zuvor erläutert, nicht eindeutig lösbar ist, müssen weitere Annahmen getroffen werden.

In den folgenden zwei Unterpunkten werden mit der globalen und der lokalen Methode die zwei wohl bekanntesten, differentiellen Methoden näher beschrieben. Bei der globalen Methode wird als zusätzliche Bedingung globale Glattheit gefordert. Glattheit ist hier im mathematischen Sinn zu verstehen. Das heißt, dass die erste Ableitung des optischen Flussfeldes an allen Stellen gleich Null sein muss. Die lokale Methode basiert stattdessen auf der Annahme, dass sich benachbarte Pixel annährend gleich bewegen [Beauchemin und Barron, 1995].

2.5.3.1 Globale Methode: Die Horn-Schunck Methode

Die globale Methode wurde von Horn und Schunck 1980 erstmals beschrieben. Geht man davon aus, dass benachbarte Oberflächenpunkte eines sich bewegenden Objektes annährend gleiche Verschiebungsvektoren besitzen, kann das Vektorfeld des optischen Flusses global als glatt angenommen werden. Diese zusätzliche Annahme der Glattheit beschränkt die Größe und Richtung des Verschiebungsvektors eines Punktes auf die seiner Nachbarn und kann für einen Pixel mathematisch wie folgt formuliert werden [Frey et al., 2010]:

$$E_s(u, v) = |\nabla u|^2 + |\nabla v|^2 \tag{2.6}$$

Zusammen mit der unterbestimmten Kontinuitätsgleichung

$$E_d(u, v) = I_x u + I_y v + I_t \tag{2.7}$$

verfügt man nun über zwei Gleichungen, die den optischen Fluss beschreiben. Die beiden Funktionen werden in einer Kostenfunktion über alle Bildpunkte zusammengefasst. Anschließend muss die Kostenfunktion minimiert werden, um eine optimale Lösung für den optischen Fluss zu erhalten.

$$E(u, v) = \int\int (E_d + \alpha E_s) dx dy \tag{2.8}$$

α ist hierbei ein Gewichtungsfaktor, der die Glattheit des Flusses quantifiziert und somit den Einfluss der beiden Bedingungen reguliert. Dieses zu minimierende Funktional ist Grundlage für viele verschiedene Lösungsansätze. Steinmüller [2008] beschreibt mit der Variationsmethode und einem Verfahren basierend auf einer diskreten Iteration zwei Ansätze zur Lösung des Minimierungsproblems. Da in dieser Arbeit ein Algorithmus, der auf dem Ansatz der lokalen Methode von Lucas und Kanade basiert, verwendet wird, wird bei der globalen Methode lediglich der grundsätzliche Ansatz erläutert und auf weiterführende Literatur verwiesen.

2.5.3.2 Lokale Methode: Die Lucas-Kanade Methode

Lucas und Kanade gehen in ihrer Arbeit davon aus, dass in der Nähe des betrachteten Bildpunktes die Verschiebung zwischen den zwei betrachteten Bildern klein und ungefähr konstant ist. Betrachtet man demnach eine $m \times m$ Nachbarschaft N_m um einen Bildpunkt (k, l) so erhält man ein überbestimmtes System von m^2 linearen Gleichungen basierend auf der Kontinuitätsgleichung 2.4 [Jiang, 2005]:

$$\forall (i, j) \in N_m : I_x(i, j) \cdot u_{kl} + I_y(i, j) \cdot v_{kl} = -I_t(i, j) \tag{2.9}$$

Ein überbestimmtes System besitzt in der Regel keine Lösung. Mit Hilfe der Methode der kleinsten Quadrate kann dennoch eine Lösung gefunden werden, die für das entsprechende Gleichungssystem bestmöglichst geeignet ist. Die Methode der kleinsten Quadrate liefert folgendes Ergebnis:

$$\begin{bmatrix} u_{kl} \\ v_{kl} \end{bmatrix} = \begin{bmatrix} \sum\limits_{(i,j)\in N_m} I_x^2(i,j) & \sum\limits_{(i,j)\in N_m} I_x(i,j)I_y(i,j) \\ \sum\limits_{(i,j)\in N_m} I_x(i,j)I_y(i,j) & \sum\limits_{(i,j)\in N_m} I_y^2(i,j) \end{bmatrix}^{-1} \cdot \begin{bmatrix} -\sum\limits_{(i,j)\in N_m} I_x(i,j)I_t(i,j) \\ -\sum\limits_{(i,j)\in N_m} I_y(i,j)I_t(i,j) \end{bmatrix} \tag{2.10}$$

Je größer die Nachbarschaft N_m ist, desto größer ist auch der Fehler, der durch die Annahme bedingt mit in die jeweilige Lösung eingeht. Mit steigender Distanz von den benachbarten Bildpunkten zum zu betrachtenden Bildpunkt, steigt auch die Wahrscheinlichkeit, dass sich ein benachbarter Pixel nicht mit dem gleichen Verschiebungsvektor wie der betrachtete Bildpunkt bewegt. Um diesen Fehler zu vermeiden, sollte der Einfluss der Nachbarn entsprechend ihrer Entfernung zum momentan betrachteten Pixel gewichtet werden. Eine derartige Gewichtungsfunktion w_{ij} kann beispielsweise durch die Gauß'sche Normalverteilung realisiert werden.

$$\begin{bmatrix} u_{kl} \\ v_{kl} \end{bmatrix} = \begin{bmatrix} \sum\limits_{(i,j)\in N_m} w_{ij}I_x^2(i,j) & \sum\limits_{(i,j)\in N_m} w_{ij}I_x(i,j)I_y(i,j) \\ \sum\limits_{(i,j)\in N_m} w_{ij}I_x(i,j)I_y(i,j) & \sum\limits_{(i,j)\in N_m} w_{ij}I_y^2(i,j) \end{bmatrix}^{-1} \cdot \begin{bmatrix} -\sum\limits_{(i,j)\in N_m} w_{ij}I_x(i,j)I_t(i,j) \\ -\sum\limits_{(i,j)\in N_m} w_{ij}I_y(i,j)I_t(i,j) \end{bmatrix} \tag{2.11}$$

2.6 Verwendeter Algorithmus

Der in dieser Arbeit verwendete Algorithmus ist eine Version der differentiellen Methode und basiert, wie bereits erwähnt, auf dem Ansatz von Lucas und Kanade. Der Algorithmus wurde von Srinivasan [1994] vorgestellt. Es handelt sich hierbei um ein einstufiges, nicht iteratives Verfahren, das die Verschiebung zwischen zwei aufeinanderfolgenden Bildern auf der Basis mehrerer Referenzbilder bestimmt.

2.6.1 Theoretischer Ansatz

Betrachtet werden erneut zwei aufeinander folgende Bilder einer Bildsequenz, die zu den Zeitpunkten t_0 und t aufgenommen werden. Ihre Helligkeitsverteilung sei durch $I_0(x, y)$ und $I(x, y)$ gegeben. Die Verschiebung zwischen den Bildern wird durch Δx und Δy in x- bzw. y-Richtung beschrieben und soll nun innerhalb eines bestimmten Bildbereiches geschätzt werden. Die Größe und Form dieses Bereiches wird durch eine Fensterfunktion Ψ definiert, beispielsweise durch eine zweidimensionale Normalverteilung:

$$\Psi(x, y) = e^{\left(-\frac{2.772}{p^2}\cdot\left(x^2+y^2\right)\right)} \tag{2.12}$$

Im ersten Schritt müssen vier Referenzbilder I_1,I_2,I_3 und I_4 mit den folgenden Eigenschaften bestimmt werden. I_1 und I_2 sind zwei Versionen des ersten Bildes I_0, die um einen Referenzwert Δx_{ref} entlang der x-Achse in negativer bzw. positiver Richtung verschoben sind. Sie sind damit wie folgt definiert:

$$I_1(x, y) = I_0(x + \Delta x_{ref}, y) \tag{2.13}$$

$$I_2(x, y) = I_0(x - \Delta x_{ref}, y) \tag{2.14}$$

I_3 und I_4 sind zwei weitere Versionen des ersten Bildes I_0. Sie sind um einen Referenzwert Δy_{ref} entlang der y-Achse in negativer bzw. positiver Richtung verschoben und analog zu den vorherigen Referenzbildern definiert:

$$I_3(x, y) = I_0(x, y + \Delta y_{ref}) \tag{2.15}$$

$$I_4(x, y) = I_0(x, y - \Delta y_{ref}) \tag{2.16}$$

Alle Referenzbilder können entweder rechnerisch durch Softwareverschiebungen oder durch den Gebrauch mehrerer Kameras ermittelt werden. Bei der Verwendung von Kameras sind diese so anzuordnen, dass ihre Bilder den entsprechenden Verschiebungen und damit den Referenzbildern von I_0 gleichkommen.

Unter der Annahme, dass das Bild sich kontinuierlich, linear und rein translatorisch von I_0 zu I verschiebt, lässt sich I ausgehend von I_0 mit Hilfe der Referenzbilder I_{1-4} durch $\hat{I}$ approximieren:

$$\hat{I} = I_0 + 0.5\left(\frac{\Delta x}{\Delta x_{ref}}\right)(I_2 - I_1) + 0.5\left(\frac{\Delta y}{\Delta y_{ref}}\right)(I_4 - I_3) \tag{2.17}$$

Srinivasan [1994] zu Folge ist diese Annahme wahr, vorausgesetzt, die Verschiebung zwischen den beiden Bildern und die Verschiebung der Referenzbilder ist bezüglich des Ausgangsbildes entsprechend klein. Um die Bewegung zwischen den beiden Bildern zu bestimmen, muss der Fehler zwischen der interpolierten Approximation $\hat{I}$ und dem wirklich gemessenen Bild I klein

sein. Um eine möglichst genaue Approximation $\hat{I}$ zu erhalten, wird daher der Unterschied zwischen der Approximation und der wahren Verschiebung innerhalb des definierten Bereichs Ψ betrachtet. Dieser Fehler wird mit Hilfe der Methode des kleinsten quadratischen Fehlers minimiert:

$$E = \int\int \Psi \cdot \left[I - \hat{I} \right]^2 dxdy \tag{2.18}$$

Durch Einsetzten von Gleichung 2.17 für $\hat{I}$ erhält man:

$$E = \int\int \Psi \cdot \left[I - \left\{ I_0 + 0.5 \left(\frac{\Delta x}{\Delta x_{ref}} \right) (I_2 - I_1) + 0.5 \left(\frac{\Delta y}{\Delta y_{ref}} \right) (I_4 - I_3) \right\} \right]^2 dxdy \tag{2.19}$$

Setzt man die partiellen Ableitung von E nach Δx bzw. Δy gleich Null, erhält man folgende zwei Gleichungen:

$$\begin{aligned} \left(\frac{\Delta x}{\Delta x_{ref}} \right) \int\int \Psi \cdot (I_2 - I_1)^2 \, dxdy + \left(\frac{\Delta y}{\Delta y_{ref}} \right) \int\int \Psi \cdot (I_4 - I_3)(I_2 - I_1) \, dxdy \\ = 2 \int\int \Psi \cdot (I - I_0)(I_2 - I_1) \, dxdy \end{aligned} \tag{2.20}$$

$$\begin{aligned} \left(\frac{\Delta x}{\Delta x_{ref}} \right) \int\int \Psi \cdot (I_2 - I_1)(I_4 - I_3) \, dxdy + \left(\frac{\Delta y}{\Delta y_{ref}} \right) \int\int \Psi \cdot (I_4 - I_3)^2 \, dxdy \\ = 2 \int\int \Psi \cdot (I - I_0)(I_4 - I_3) \, dxdy \end{aligned} \tag{2.21}$$

Durch die Gleichungen 2.20 und 2.21 lassen sich Δx und Δy direkt ermitteln. Da es zwei Gleichungen mit zwei Unbekannten gibt, ist die Lösung eindeutig, sofern sie existiert. Es kann genau dann keine Lösung gefunden werden, wenn der betrachtete Bereich keinerlei Konturen enthält, also homogen ist oder das Blendenproblem (vgl. 2.3) auftritt.

Der hier beschriebene, theoretische Ansatz ist eine Vereinfachung des von Srinivasan beschriebenen Verfahrens. In seiner Arbeit [Srinivasan, 1994] ist dieser Ansatz um rotatorische Bewegungen erweitert. Zusätzlich präsentiert der Artikel Testreihen und Ergebnisse bei denen dieses Verfahren für synthetisch erzeugte und reale Bildsequenzen verwendet wurde.

2.6.2 Praktische Umsetzung

Bei der Implementierung dieses Verfahrens wurden weitere Vereinfachungen getroffen. Der betrachtete Bereich wird nicht durch eine zweidimensionale Normalverteilung beschrieben, sondern

durch ein Fadenkreuz, bestehend aus einem Centerpixel und seinen vier Nachbarn. Zusätzlich sollen die Referenzwerte Δx_{ref} und Δy_{ref} zur rechnerischen Ermittlung der Referenzbilder I_{1-4} lediglich einen Pixel betragen. Dadurch lassen sich die Referenzbilder einfach ermitteln, indem die vier benachbarten Pixel des Centerpixels betrachtet werden, wodurch also kein zusätzlicher Rechenaufwand entsteht. Abbildung 2.6 zeigt das Fadenkreuz beim Start des Algorithmus. Der Rand des Bildes wird für die Centerpixel ausgelassen, da sich offensichtlich keine Referenzverschiebungen außerhalb des Bildes befinden können.

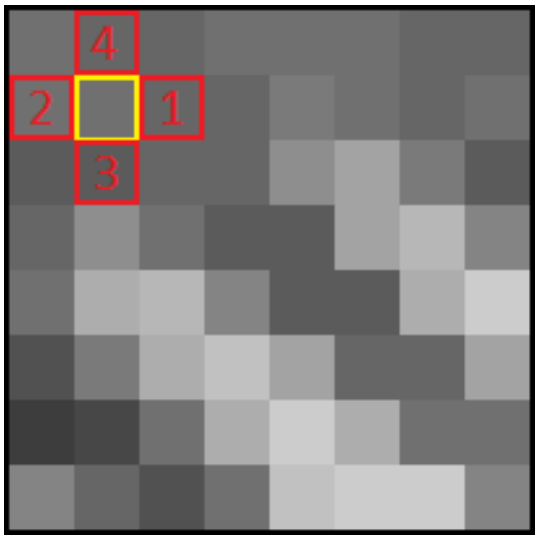


Abbildung 2.6: Betrachteter Bildbereich: Centerpixel (gelb) und benachbarte Pixel (rot)

Somit lassen sich die Gleichungen 2.20 und 2.21 vereinfachen. Drückt man beide Gleichungen durch Matrizen aus, erhält man in diskretisierter Form:

$$\begin{aligned} \begin{bmatrix} \sum\sum (I_2 - I_1)^2 & \sum\sum (I_4 - I_3)(I_2 - I_1) \\ \sum\sum (I_2 - I_1)(I_4 - I_3) & \sum\sum (I_4 - I_3)^2 \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \\ = 2 \begin{bmatrix} \sum\sum (I - I_0)(I_2 - I_1) \\ \sum\sum (I - I_0)(I_4 - I_3) \end{bmatrix} \end{aligned} \tag{2.22}$$

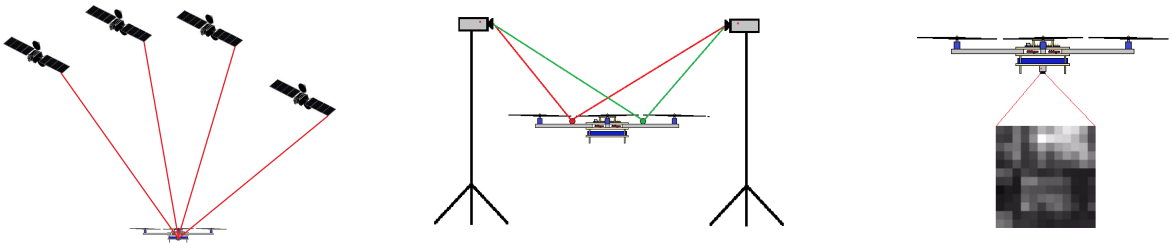
Diese Gleichung lässt sich einfach implementieren. Multipliziert man das Inverse der 2×2 Matrix mit der rechten Seite der Gleichung, erhält man eine eindeutige Lösung für den optischen Flussvektor $(\Delta x, \Delta y)^T$. Zusätzlich werden die zwei Konfigurationsparameter OfScale und ΔMax eingeführt. OfScale gewichtet, wie stark die Verschiebung eines Pixels in den errechneten optischen Flusswert eingeht. ΔMax bestimmt, wann das Referenzbild ersetzt werden muss. Dies ist genau dann der Fall, wenn der Betrag des optischen Flussvektors größer als ΔMax ist.

3 Stand der Technik

Damit ein UAV autonom fliegen kann, muss es zu jeder Zeit seine Position kennen und regeln können. In den letzten Jahren wurden verschiedene Methoden zur Positionsbestimmung von UAVs entwickelt. Die mit Abstand am weitesten verbreitete Methode führt die Lokalisierung der Drohne mit Hilfe des **G**lobal **P**ositioning **S**ystem (GPS) durch. Daneben werden häufig auch optische Systeme verwendet. Hierbei kann man zwischen externen Tracking Systemen und optischen Systemen, die in die Drohne selbst integriert sind, unterscheiden. Im ersten Abschnitt werden die Vor- und Nachteile sowie die Anwendungsmöglichkeiten der eben genannten Systeme erläutert. Anschließend werden verschiedene Ansätze zur Umsetzung eines integrierten, optischen Systems zur Positionsbestimmung dargestellt.

3.1 Systeme zur Positionsbestimmung

Je nach Anwendungsbereich werden unterschiedliche Anforderungen an die Positionssysteme gestellt. Im Rahmen des AQopterI8-Projektes der Universität Würzburg soll ein autonomer Quadrocopter zur Innenraumerkundung entwickelt werden[AQopterI8, 2012]. Der fertige Quadrocopter muss sich also selbstständig, ohne die Hilfe von externen Systemen, innerhalb eines Gebäudes lokalisieren können. Abbildung 3.1 zeigt verschiedene Systeme zur Positionsbestimmung.



(a) Quadrocopter mit GPS Satellitenkontakt

(b) Quadrocopter mit Markern und optischem Tracking System

(c) Quadrocopter mit integriertem optischen System

Abbildung 3.1: Übersicht über die Systeme zur Positionsbestimmung eines Quadrocopters

3.1.1 Global Positioning System

Fast alle autonomen Drohnen verfügen über GPS. Das **G**lobal **P**ositioning **S**ystem ermittelt die Position durch die Laufzeitmessung von Satellitensignalen. Mit Hilfe von Differenzmethoden lässt sie sich sogar bis auf wenige Zentimeter genau bestimmen. Zusätzlich zu aktuellem Längen- und Breitengrad kann auch die Höhe erfasst werden. Damit diese Art der Lokalisierung funktioniert, ist es folglich notwendig, ausreichend guten Kontakt zu den GPS-Satelliten herzustellen. Allerdings ist dies beim Gebäudeflug nicht möglich. Somit ist eine Positionsregelung mittels GPS insbesondere für einen Indoor-Quadrocopter nicht geeignet.

3.1.2 Externe Tracking Systeme

Im Gegensatz zu GPS kommen externe Tracking Systeme fast ausschließlich im Innenbereich zum Einsatz. Das zu beobachtende Flugobjekt wird dabei mit Markern versehen und von mehreren Kameras gefilmt. Die gesammelten Informationen werden durch einen eigenen Rechner verarbeitet und ermöglichen so eine dreidimensionale Lokalisierung. Gerade im Forschungsbereich der UAVs gibt es zahlreiche Anwendungsmöglichkeiten. Sie reichen von der einfachen Positionsbestimmung über die Trajektorieverfolgung bis hin zur Objekterkennung. Da man allerdings stets auf externe Hilfsmittel angewiesen ist, eignet sich diese Methode nicht für den Bau eines autonomen Indoor-Quadrocopters, der sich selbstständig in einem fremden Raum lokalisieren soll.

3.1.3 Integrierte optische Systeme

Anders als die bisher genannten Systeme sind integrierte, optische Systeme nicht zwangsweise auf externe Hilfsmittel angewiesen. Sie bestehen meist aus einer oder mehreren Kameras, die an der Drohne angebracht werden. Mit Hilfe der Kameras und geeigneter Verfahren zur Auswertung der gesammelten Bildinformationen lassen sich so Bewegungen im Raum erkennen, die anschließend Rückschlüsse auf die Position erlauben. Durch eine senkrecht zum Boden gerichtete Kamera kann beispielsweise der optische Fluss, der durch die Eigenbewegung der Drohne entsteht, gemessen werden. Durch den gemessenen optischen Fluss kann die Positionsänderung approximiert werden. Ist die Anfangsposition der Drohne bekannt, verfügt sie somit stets über genaue Positionsangaben. Ein derartiges System ist sowohl für den Einsatz innerhalb, als auch außerhalb eines Gebäudes geeignet. Da keine externen Informationen benötigt werden, kann das System auch in völlig neuen Umgebungen eingesetzt werden.

3.2 Implementierung integrierter optischer Systeme

Die Verfahren, die aus einer Folge von Bildern Informationen über die Eigenbewegung des Systems in Echtzeit berechnen, sind häufig sehr komplex und in den meisten Fällen mit einem hohen Rechenaufwand verbunden. Die Bilddaten des optischen Sensors müssen mit einer entsprechend hohen Frequenz ausgelesen und anschließend schnell verarbeitet werden, um die Echtzeitfähigkeit des Gesamtsystems zu gewährleisten. Es existieren bereits verschiedene Ansätze für die Implementierung integrierter optischer Systeme zur Bestimmung der Eigenbewegung von UAVs mit Hilfe des optischen Flusses[Ding et al., 2009][Hrabar et al., 2005]. Eine exakte Kategorisierung der Ansätze ist aufgrund ihrer Vielzahl allerdings nicht möglich. In den folgenden Abschnitten soll die Vielfältigkeit der unterschiedlichen Ansätze an Hand von Beispielen veranschaulicht werden.

3.2.1 Optische Flusssensoren

Um ein optisches System zur Positionsbestimmung von UAVs zu realisieren, können fertige Sensoren verwendet werden. Bei den Sensoren handelt es sich um optische Flusssensoren, die normalerweise in optischen Computermäusen verbaut werden. Durch entsprechende Konfigurationen der Optik können sie jedoch auch zur Positionsregelung von UAVs eingesetzt werden. Das OpenSource Projekt Arducopter [Arducopter, 2012] und der SkeyeCopter der technischen Universität Chemnitz [Lange und Sünderhauf, 2012] verwenden mit dem ADNS 3080 von Avago Technologies einen solchen Sensor (siehe Abbildung 3.2).

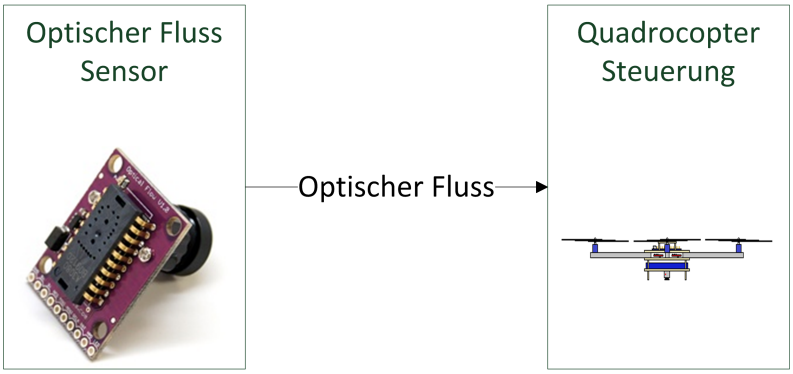


Abbildung 3.2: Schematische Darstellung bei Verwendung eines ADNS 3080 Sensors

Die Vorteile dieses Ansatzes sind offensichtlich. Der fertige Sensor muss lediglich in das bestehende System eingebunden werden und ist anschließend sofort einsatzbereit. Die spezialisierte Hardware ermöglicht zudem extrem hohe Bildraten und liefert bei guten Lichtverhältnissen genaue Messwerte. Allerdings bringt die Verwendung eines solchen Sensors auch Nachteile mit sich. So ist er in seiner Funktionalität stark eingeschränkt. Im Falle des ADNS 3080 können beispielsweise keine Rotationen gemessen werden, da er für Computermäuse und damit für rein

translatorische Bewegungen konzipiert ist. Zusätzlich muss eine ausreichend gute Beleuchtung der Szene sichergestellt werden, um verlässliche Ergebnisse zu produzieren.

3.2.2 System aus Bildsensor und Prozessor

Bei diesem Ansatz wird ein digitaler Bildsensor verwendet, um das Bild zu erfassen. Anschließend wird das Bild mit Hilfe eines Prozessors bearbeitet und ausgewertet. In den folgenden Unterpunkten werden die gängigsten Komponenten, die zur Umsetzung dieses Ansatzes verwendet werden, erläutert. Sowohl bei den Bildsensoren, als auch bei den Prozessoren kann auf unterschiedliche Technologien zurückgegriffen werden. Bei den Bildsensoren unterscheidet man zwischen **C**omplementary **M**etal **O**xide **S**emiconductor (CMOS) Sensoren, **C**harged **C**oupled **D**evice (CCD) Sensoren und speziellen Vision Chips. Der Prozessor zur Bildverarbeitung kann durch einen Mikrokontroller oder ein **F**ield **P**rogrammable **G**ate **A**rray (FPGA) realisiert werden.

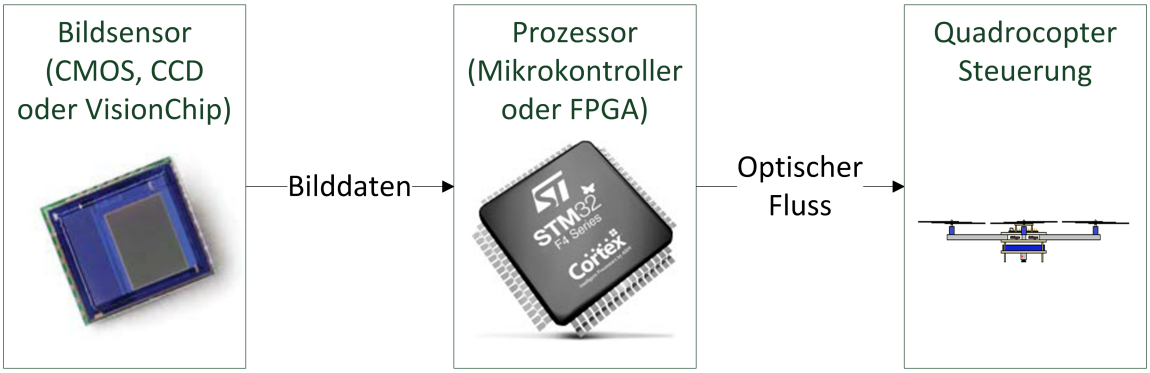


Abbildung 3.3: Schematische Darstellung bei Verwendung eines Bildsensors und Prozessors

3.2.2.1 CMOS Sensoren

CMOS Sensoren sind digitale Bildsensoren, die heutzutage in fast allen digitalen Foto- und Videokameras verbaut werden. Im einfachsten Fall besteht ein Bildelement des Sensors aus einer in Sperrrichtung geschaltenen Photodiode und drei Feldeffekttransistoren. Die Spannung der Photodiode wird zu Beginn mit Hilfe des Resetsignals auf ein definiertes Level gesetzt. Während der Belichtungszeit wird die Diode über den Photostrom entladen. Die verbleibende Spannung wird ausgelesen, verstärkt und an einen Analog-Digital-Wandler weitergegeben (vgl. Abbildung 3.4)[Ohta, 2010].

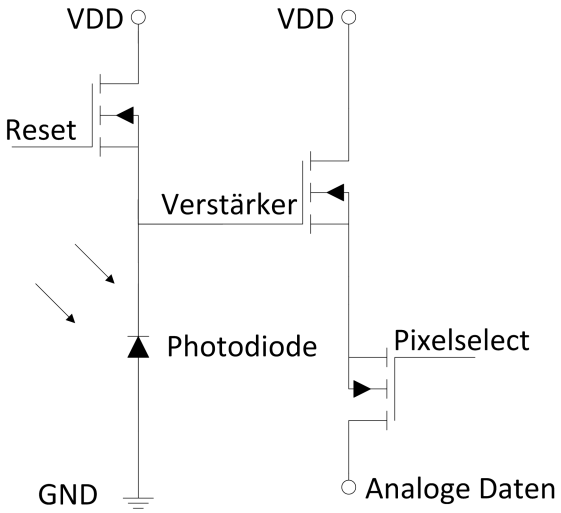


Abbildung 3.4: Aktives Bildelement eines CMOS Sensors [Ohta, 2010]

Anders als bei CCD Sensoren kann so die Helligkeit jedes einzelnen Bildelements ausgelesen werden. Ihre einfache Produktion macht CMOS Sensoren deutlich billiger als CCD Sensoren und erlaubt zusätzlich die Integration weiterer Funktionen[AXISCommunications, 2010].

3.2.2.2 CCD Sensoren

CCD Sensoren bestehen ebenfalls aus lichtempfindlichen Photodioden. Bei Lichteinfall findet eine Ladungsverschiebung innerhalb der Photodioden statt. Die Ladungsverschiebungen werden dann in eine Spannung umgewandelt und als analoges Signal weitergeleitet. Das analoge Signal wird anschließend verstärkt und mittels eines Analog-Digital-Wandlers ausgegeben[AXISCommunications, 2010]. Anders als CMOS Sensoren basieren CCD Sensoren nicht auf digitalen, sondern auf analogen Bauteilen. CCD Sensoren werden bereits seit 1975 in Fotokameras verwendet[Wikipedia, 2012]. Ursprüngliche Vorteile gegenüber CMOS Sensoren, wie eine bessere Lichtempfindlichkeit und geringere Rauschanfälligkeit, existieren auf Grund der rassanten Entwicklung in der Halbleitertechnologie der letzten Jahre nicht mehr. Die Nachteile von CCD Sensoren sind dafür umso gravierender gegenüber CMOS Sensoren. Da analoge Komponenten verwendet werden, sind sie in der Herstellung sehr teuer und benötigen im Betrieb deutlich mehr Strom. Der hohe Stromverbrauch führt darüberhinaus zu ungewollter Wärmeentwicklung im Sensor[AXISCommunications, 2010].

3.2.2.3 Spezielle Vision Chips

Neben den herkömmlichen CMOS und CCD Sensoren gibt es auch spezielle Sensoren auf CMOS Basis, die eigens für die Bestimmung des optischen Flusses entwickelt wurden. Sie werden als Vision Chips bezeichnet. Es handelt sich hierbei um schwarz-weiß Sensoren mit einer einfach

anzusprechenden Schnittstelle. Die geringe Auflösung dieser Sensoren ermöglicht höhere Bildraten als bei herkömmlichen Sensoren. Einige Sensoren verfügen zusätzlich über die Möglichkeit selbstständig Bildbearbeitungen, wie beispielsweise eine Bildglättung durch intelligentes Zusammenfassen von Pixeln, durchzuführen. Das Robobee Projekt der Harvard University benutzt einen von der Firma Centeye entwickelten Vision Chip [Harvard, 2012][Centeye, 2012]. Die speziell entwickelten CMOS Vision Chips sind allerdings im Vergleich zu herkömmlichen CMOS Sensoren deutlich teurer.

3.2.2.4 Mikrokontroller

Die aktuelle Generation von Mikrokontrollern ist leistungsstark genug, um in Echtzeit Bilder von einem herkömmlichen Bildsensoren auszulesen und anschließend zu verarbeiten. Mikrokontroller mit speziellen Schnittstellen für die Bildsensoren eignen sich hierbei besonders. Es existieren bereits fertige Einplatinenlösungen, zum Beispiel die CMUCam 4 der Carnegie Mellon University[CMUCam, 2012]. Weitere, vielversprechende Einplatinenlösungen befinden sich momentan noch in der Entwicklung, wie das PX4FLOW Smart Camera Projekt von px4Autopilot[px4Autopilot, 2012]. Der große Vorteil dieses Ansatzs liegt, anders als bei der Verwendung von optischen Flusssensoren, in der Flexibilität des Systems. Die Algorithmen zur Bildverarbeitung können jederzeit angepasst und die Funktionalität beliebig erweitert werden. Darüber hinaus können die Algortihmen so modifiziert werden, dass das optische System auch bei schlechteren Lichtverhältnissen arbeiten kann. Mit den Bildraten eines optischen Flusssensors kann ein System aus Bildsensor und Mikrokontroller jedoch nicht einmal annähernd mithalten. Beispielsweise arbeitet mit einem solchen System der von der Humboldt-Universität Berlin entwickelte Quadrocopter[SpiegelOnline, 2008][Hafner et al., 2010], ebenso der Flapping Wing Robot der University of Carlifornia [Bermudez und Fearing, 2009].

3.2.2.5 Field Programmable Gate Arrays

Aktuelle FPGAs verfügen analog zu den Mikrokontrollern über ausreichend große Leistungskapazitäten. Gängige Schnittstellen für Bildsensoren lassen sich leicht implementieren. Dieser Implementierungsansatz besitzt alle Vorteile des Mikrokontrolleransatzes. Darüber hinaus können Bildraten bis zu 1000 Bilder pro Sekunde durch einen hohen Grad an Parallelisierung erreicht und verarbeitet werden [Ishii et al., 2010]. Zahlreiche Veröffentlichungen beschreiben derartige Systeme, z.B. [Abutaleb et al., 2010] und [Wei et al., 2007]. Nachteile dieser Lösung sind vergleichsmäßig teure Hardware und der große Entwicklungsaufwand bis zu einem fertigen System.

3.2.3 Konsequenzen für die vorliegende Arbeit

Wie im ersten Abschnitt dieses Kapitels (vgl. Abschnitt 3.1) gezeigt, existieren verschiedene Ansätze zur Positionsbestimmung eines Quadrocopters. Damit ein Quadrocopter sich auch in einem unbekannten Raum zurechtfinden kann, muss er auf integrierte Systeme zur Positionsbestimmung zurückgreifen. Die Positionsbestimmung mit Hilfe eines optischen Systems eignet sich hierzu besonders, da diese Methode sehr viele Vorteile gegenüber anderen Messmethoden besitzt. Optische Systeme besitzen eine deutlich höhere Genauigkeit als Inertialsensoren oder Sensoren, die zur Abstandsmessung, wie beispielsweise Infrarot- und Ultraschallsensoren, verwendet werden. Darüberhinaus entsteht in einem optischen System im Gegensatz zu Systemen, die auf Inertialsensoren basieren, kein Sensordrift, der die Messwerte verfälscht. Damit die Eigenbewegung und damit die Position des Systems aus einer Bildsequenz durch das optische System bestimmt werden kann, wird der optische Fluss berechnet. Das Thema optischer Fluss ist, wie der vorherige Abschnitt und Kapitel 2 zeigen, bereits sehr intensiv erforscht worden. Doch obwohl zahlreiche Veröffentlichungen zur Entwicklung eines integrierten optischen Systems zur Messung des optischen Flusses existieren, gibt es nur wenige Projekte, die die Integration eines optischen Systems in einen Quadrocopter beschreiben. Die wenigen Veröffentlichungen, die sich mit diesem Thema beschäftigen, beschreiben allerdings nur in Ansätzen die Umsetzung eines solchen Systems [Hafner et al., 2010]. Besonders über kostengünstige Quadrocopter mit einem geringen Energieverbrauch gibt es keine detaillierten Veröffentlichungen, die die technischen Anforderungen an Autonomie und Flexibilität des Systems erfüllen. Um dem Quadrocopter des AQopterI8 Projekts der Universität Würzburg einen möglichst hohen Grad an Autonomie und Flexibilität zu verleihen, ist die Entwicklung eines zuverlässigen Systems zur Positionsbestimmung und -regelung daher unerlässlich.

4 Konzept

In diesem Kapitel wird das der Arbeit zugrundeliegende Konzept vorgestellt. Im ersten Teil wird ein Überblick über das Gesamtkonzept vermittelt. Es werden Anforderungen an das fertige System formuliert und die geplanten Schritte auf dem Weg zur Fertigstellung des Systems erläutert. Im zweiten Teil werden die zu entwickelnden Systemkomponenten ausführlich beschrieben.

4.1 Überblick

Abbildung 4.1 zeigt eine vereinfachte Darstellung des Gesamtkonzeptes.

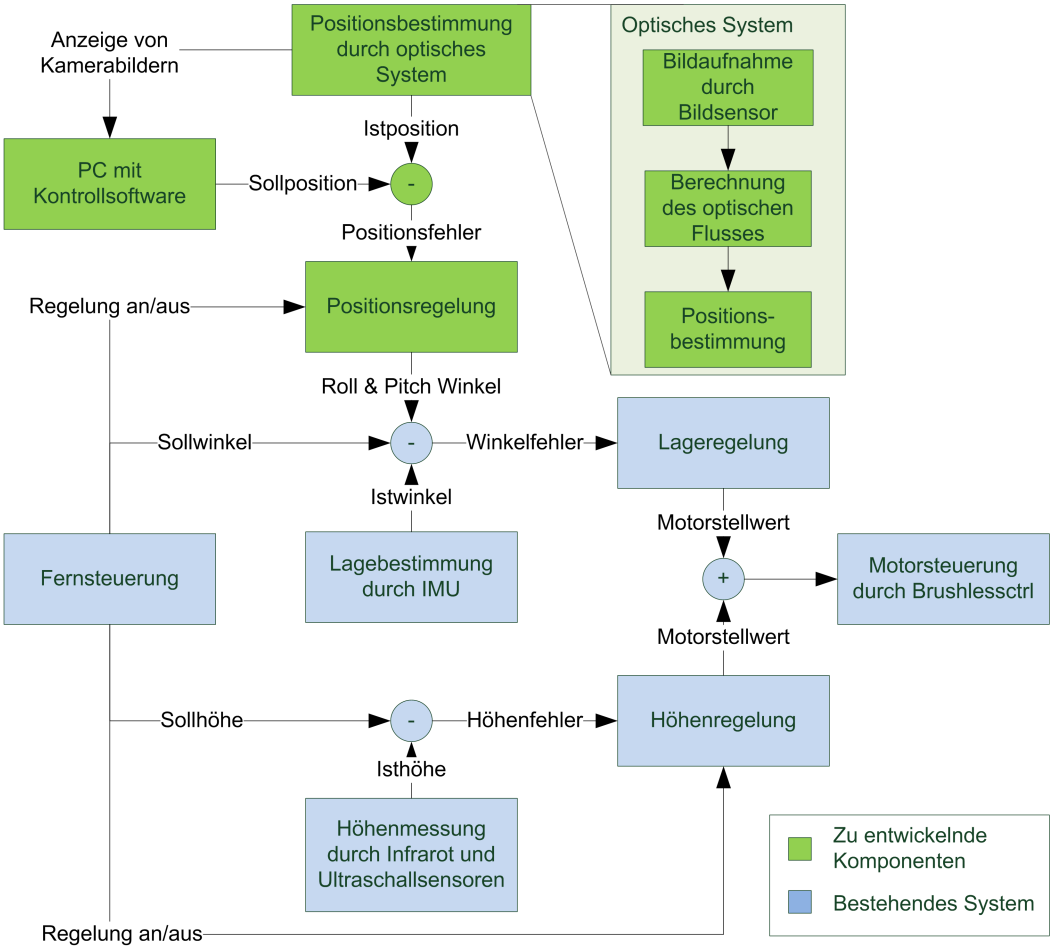


Abbildung 4.1: Gesamtkonzept

Da es, wie in Abschnitt 3.2.3 beschrieben, keine detaillierten Veröffentlichung zu diesem Thema gibt, muss ein eigenes Konzept entwickelt werden. Abbildung 4.1 visualisiert das entwickelte Gesamtkonzept. Die blaugefärbten Komponenten bilden das bisher bestehende System, die grünen Komponenten stellen die notwendigen Erweiterungen des Systems dar. Das bisherige System besteht im wesentlichen aus vier Teilen:

- **Fernsteuerung**: Die Fernsteuerung wird zum Ein- und Ausschalten einzelner Komponenten sowie zur Steuerung des Quadrocopters verwendet.
- **Lagereglung mit entsprechender Sensorik**: Die Lageregelung hält den Quadrocopter stabil in der Luft und setzt die Steuerbefehle der Fernsteuerung um. Zur Messung der Quadrocopterlage wird eine **I**nertial **M**easurement **U**nit (IMU) verwendet.
- **Höhenregelung mit entsprechender Sensorik**: Die Aufgabe der Höhenregelung besteht darin, eine bestimmte Höhe autonom zu halten. Die aktuelle Höhe des Quadrocopters wird mit Hilfe von Infrarot- und Ultraschallsensoren ermittelt.
- **Motorsteuerung**: Die Motorsteuerung erhält die Stellwerte der einzelnen Regler und und gibt diese an die einzelnen Motoren weiter.

In dieses bestehende System müssen nun die notwendigen Erweiterungen integriert werden. Diese Erweiterungen lassen sich in die folgenden drei Systemkomponenten unterteilen:

- **Kontrollsoftware**: Über die Kontrollsoftware kann die Sollposition des Quadrocopters geändert und die aktuelle Position jederzeit ausgelesen werden. Darüber hinaus kann sie die Bilder des optischen Systems anzeigen.
- **Optisches System zur Positionsbestimmung**: Die Aufgabe des optischen Systems zur Positionsbestimmung ist, wie der Name bereits sagt, die aktuelle Position des Quadrocopters zu ermitteln. Dazu muss zuerst mit einem optischen Sensor eine Bildsequenz aufgenommen werden. Aus dieser wird anschließend durch die Berechnung des optischen Flusses die Eigenbewegung des Quadrocopters approximiert. Aus der Eigenbewegung wird dann durch eine einfache Integration die aktuelle Position des Quadrocopters relativ zu einem Startpunkt ermittelt.
- **Positionsregler**: Der Positionsregler vergleicht die Soll- und Istposition des Quadrocopters. Er berechnet dann Stellwerte für die Roll- und Pitchwinkel der Lageregelung so, dass die gewünschte Sollposition angeflogen und gehalten wird.

4.1.1 Systemanforderungen

Das fertige System soll mit Hilfe des optischen Flusses selbstständig seine Position regeln. Um dies zu erreichen, müssen zunächst einige Anforderung an das System formuliert werden. Neben den zuvor beschriebenen Funktionen soll das fertige System:

- in Echtzeit seine Position relativ zu einem Startpunkt bestimmen,
- eine angeflogene Position mit möglichst kleiner Abweichung für eine gewisse Zeit halten
- und eine per Kontrollsoftware übergebene Position selbstständig anfliegen und halten.

4.1.2 Entwicklungsphasen

Um das Gesamtkonzept umzusetzen, werden als nächstes einzelne Etappenziele festgesetzt. Das Flussdiagramm in Abbildung 4.2 veranschaulicht die einzelnen Schritte der Systementwicklung. Im ersten Schritt muss ein geeignetes optisches System zur Bestimmung des optischen Flusses gefunden werden. Dieses System wird dann in den Quadrocopter integriert und anschließend umfassend evaluiert.

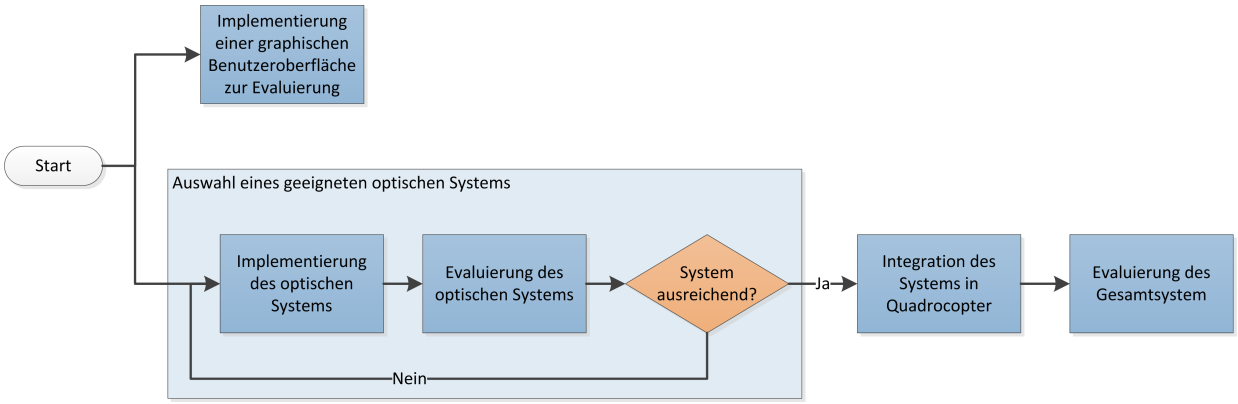


Abbildung 4.2: Phasen während der Systementwicklung

4.2 Systemkomponenten

Wie aus dem Gesamtkonzept (vgl. Abbildung 4.1) hervorgeht, muss das System um drei Komponenten erweitert werden. Bei diesen drei Komponenten handelt es sich um die Kontrollsoftware zum Evaluieren und Steuern des System, das optische System zur Positionsbestimmung und den eigentlichen Positionsregler.

4.2.1 Die Kontrollsoftware

Zur Evaluierung und Steuerung des entwickelten Systems wird eine speziell entwickelte Kontrollsoftware benötigt. Dazu wird ein bereits bestehendes Tool um zusätzliche Funktionen erweitert. Zur Evaluierung sollen die Bilder des optischen Systems in Echtzeit angezeigt werden können, um den Fokus des Objektives und die Güte der Bilder zu beurteilen. Die Positionsänderungen des Quadrocopters soll ebenfalls visualisiert werden, um das Regelverhalten zu bewerten. Zur Steuerung des Systems muss die aktuelle Sollposition an den Quadrocopter übertragen werden können. Wichtige Parameter des optischen Systems, wie beispielsweise die Bildrate oder die Belichtungsdauer, sollen auch über die Software angepasst werden können.

4.2.2 Das Optische System zur Positionsbestimmung

Das optische System zur Positionsbestimmung besteht aus einem integrierten, optischen System, mit dessen Hilfe der optische Fluss $(u, v)^T$ als Approximation der Eigenbewegung des Quadrocopters berechnet wird und dem Mikrokontroller, der die Steuerung des Quadrocopters übernimmt. Der berechnete optische Fluss wird an den Mikrokontroller weitergegeben, der daraufhin die aktuelle Position $(x, y)^T$ bezüglich einer bekannten Anfangsposition $(x_0, y_0)^T$ wie folgt berechnet:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \int \begin{pmatrix} u \\ v \end{pmatrix} dt + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \tag{4.1}$$

In Abbildung 4.3 wird die Aufgabenverteilung der beteiligten Systeme bei Positionsbestimmung und der Positionsregelung noch einmal verdeutlicht.

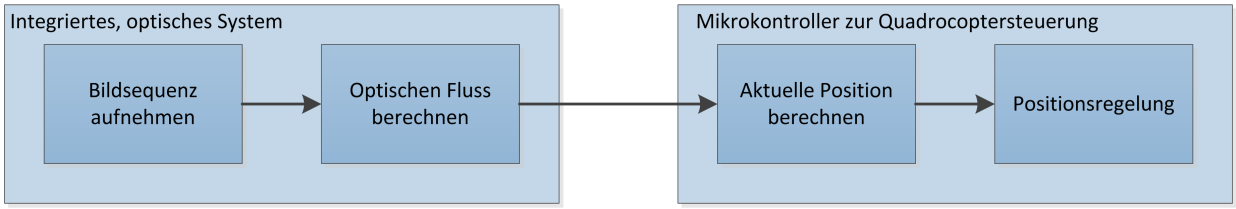


Abbildung 4.3: Aufgabenverteilung bei der Positionsregelung

Zur Umsetzung des integrierten, optischen Systems bieten sich die zwei, in Abschnitt 3.2 beschriebenen Möglichkeiten an. Da dieses mittig, an der Unterseite des Quadrocopters montiert werden soll, muss nach einer möglichst platzsparenden Lösung gesucht werden.

4.2.3 Der Positionsregler

Wie Abbildung 4.3 zeigt, soll der Positionsregler auf dem Mikrokontroller zur Steuerung des Quadrocopters implementiert werden. Der Positionsregler besteht aus zwei überlagerten **P**roportional-**I**ntegral-**D**ifferential (PID)-Reglern für den Roll bzw. Pitchwinkel. Die vom Positionsregler berechneten Sollwerte für die Roll- und Pitchwinkel werden der Lageregelung des Quadrocopters übergeben. Diese berechnet wiederum die entsprechenden Stellsignale zur Motorsteuerung. Abbildung 4.4 zeigt ein detailliertes Blockschaltbild des Positionsregelkreises.

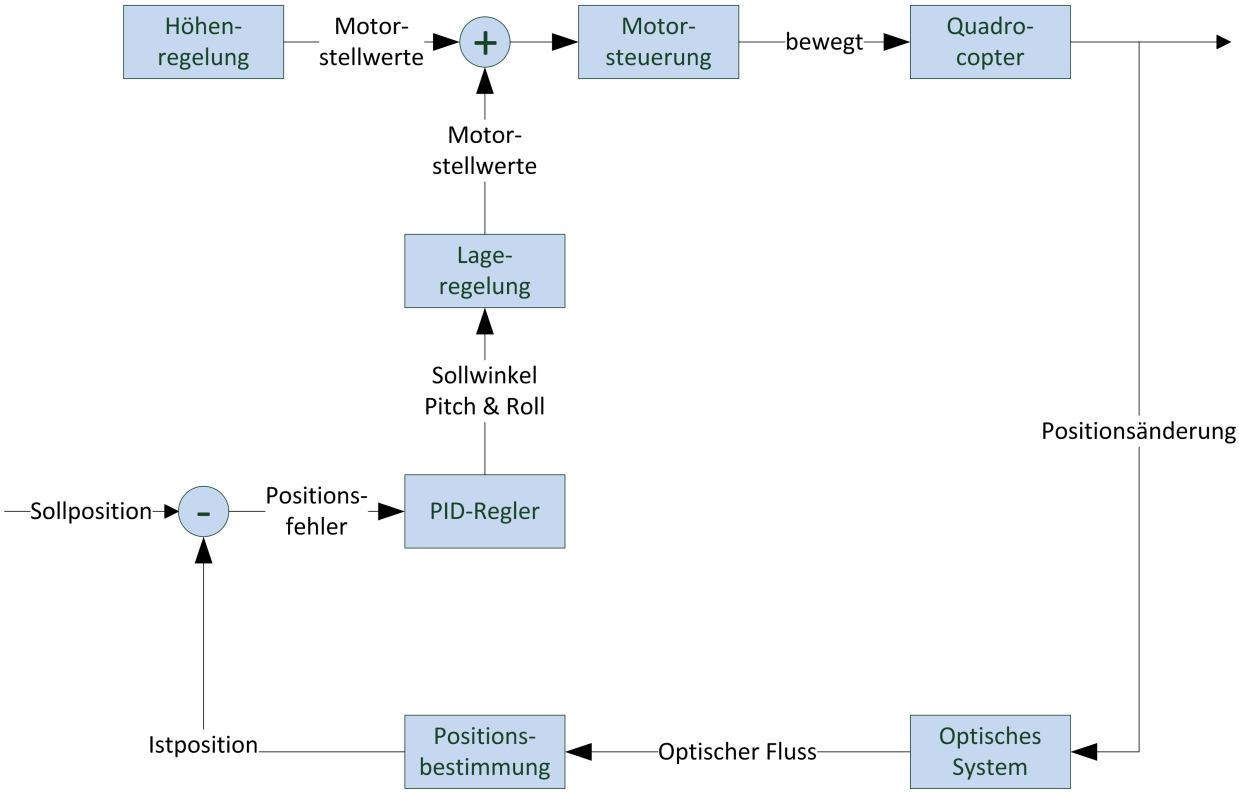


Abbildung 4.4: Positionsregelkreis

5 Implementierung

Dieses Kapitel beschreibt die Entwicklung der notwendigen Erweiterungen, bestehend aus der Kontrollsoftware, dem optischen System zur Positionsbestimmung und dem Positionsregler. Zuerst wird die Funktionalität der Kontrollsoftware zur Steuerung und Evaluierung des Systems erläutert. Anschließend wird der Aufbau verschiedener optischer Systeme zur Bestimmung des optischen Flusses geschildert. Am Ende dieses Kapitel wird das optische System, das sich am Besten für den Einsatz an einem Quadrocopter eignet, in das bestehende Quadrocoptersystem integriert und der eigentliche Positionsregler implementiert.

5.1 Kontrollsoftware zur Evaluierung und Steuerung

Die Kontrollsoftware und ihre graphische Benutzeroberfläche (engl. **G**raphical **U**ser **I**nterface (GUI)) wird mit Hilfe des QT Creators, einer integrierten Entwicklungsumgebung (engl. **I**ntegrated **D**evelopment **E**nvironment (IDE)) von Nokia, in C++ entwickelt. Als Basis wird das QT Steuerprogramm des AQopterI8 Projektes verwendet, bei dem unter anderem folgende Funktionen bereits implementiert sind:

- Serielle **U**niversal **A**synchronous **R**eceiver **T**ransmitter (UART) Schnittstelle
- Darstellung des zeitlichen Verlaufs von frei auswählbaren Messwerten
- Schnittstelle für ein externes, optisches Tracking System
- erweiterbares Protokoll zur Kommunikation mit dem Quadrocopter

Es wird ein zusätzlicher Reiter *Kamera* zu der Benutzeroberfläche hinzugefügt. Das Fenster dieses Reiters unterteilt sich in zwei Bereiche. Im rechten Teil kann das aktuelle Kamerabild angezeigt werden. Der linke Teil besteht aus drei weiteren Reitern: *Wegpunkte*, *ADNS 3080* und *OV 7670*. Der Inhalt des Reiters *Wegpunkte* dient zur zweidimensionalen Visualisierung der Quadrocoptertrajektorie und dem Setzen der Sollposition des Positionsreglers. In den beiden anderen Reitern können Kameraeinstellungen für die entsprechenden, optischen Systeme vorgenommen werden.

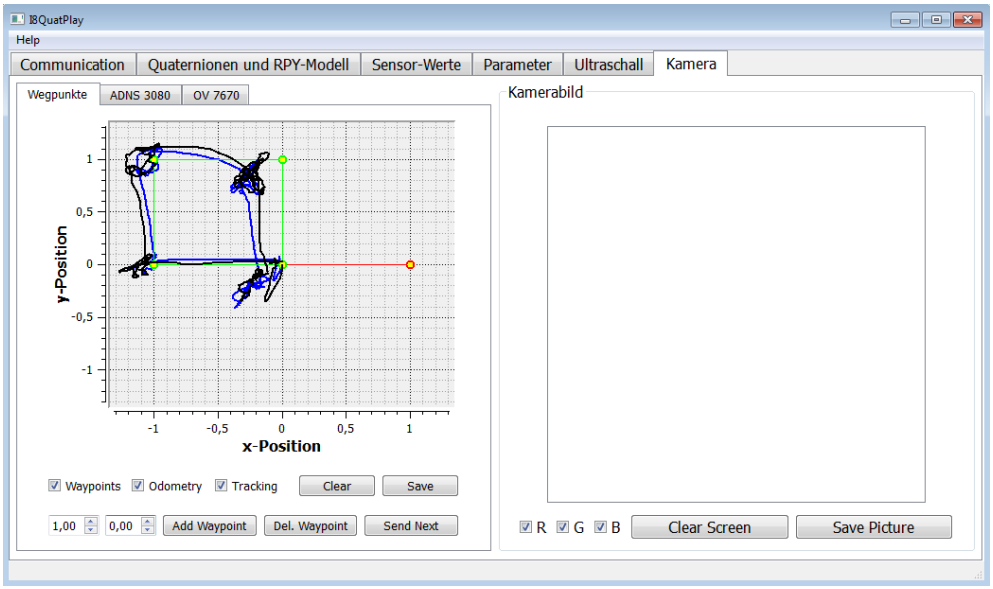


Abbildung 5.1: Benutzeroberfläche der Kontrollsoftware bei der Trajektorieverfolgung

5.1.1 Evaluierungsfunktionen

Um die Messwerte des optischen Flusses zeitabhängig darzustellen, muss lediglich das in der *protocol.h* definierte Protokoll ergänzt werden. Ein zusätzlicher Eintrag in dieser Headerdatei reicht zur Darstellung weiterer Messwerte aus. Im Reiter Sensor-Werte können dann die entsprechenden Messwerte angezeigt werden.

Zur Darstellung der Quadrocoptertrajektorie wird ein Koordinatensystem im Reiter *Wegpunkte* erstellt. Über die Checkboxen *Waypoints*, *Odometry* und *Tracking* lassen sich die einzelnen Graphen ein- bzw. ausblenden. Über den Button *Clear* können die gespeicherten Messwerte der selektierten Graphen gelöscht werden. Der Button *Save* speichert die aktuellen Messwerte aller sichtbaren Graphen. Bereits übertragene Sollpositionen werden grün dargestellt, die anderen rot. Der blaue Pfad ist die mit Hilfe des optischen Trackingsystems gemessene Trajektorie. Der schwarze Pfad wurde durch den optischen Fluss bestimmt (vgl. Abbildung 5.1, linker Teil).

Die Darstellung des Kamerabildes wird für zwei Bildformate implementiert. Es können Bilder im 8bit Graustufenformat und im 16bit RGB565 Format angezeigt werden. Das Format wird durch die Selektion eines Kamerareiters automatisch geändert. Für den Reiter *ADNS 3080* wird das 8bit Graustufenformat bei einer festen Auflösung von 30×30 Pixel gewählt. Öffnet man den Reiter *OV 7670* wird das RGB565 Format eingestellt. Die Auflösung kann hier im Reiter verändert werden. Falls das RGB565 Format aktiv ist, lassen sich durch die Checkboxen *R*, *G* und *B* einzelne Farbkanäle an- bzw abwählen. Der Button *Clear Screen* löscht das aktuelle Bild. Mit *Save Picture* lässt sich das aktuelle Bild speichern (vgl. Abbildung 5.2, rechter Teil).

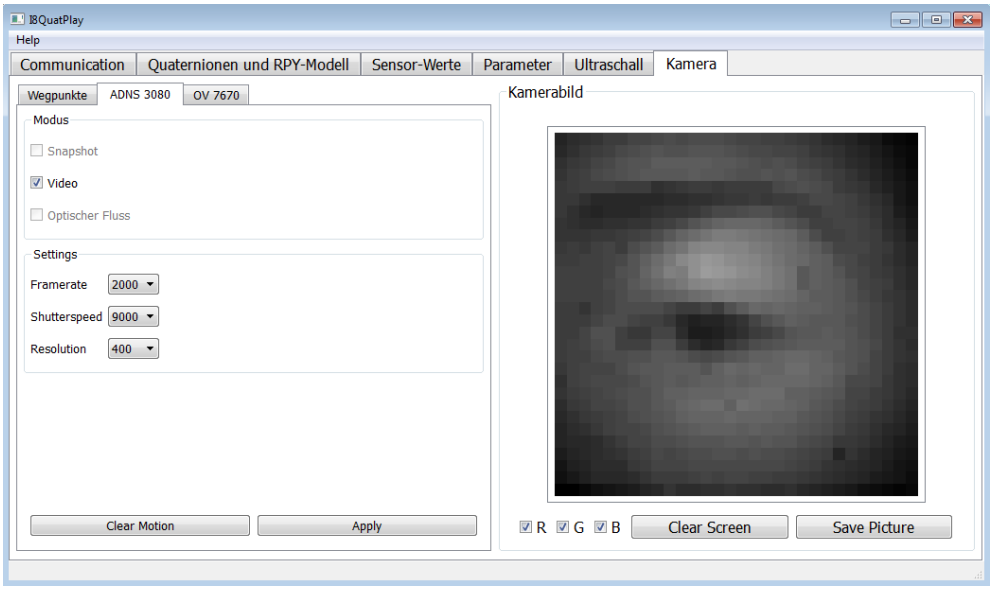


Abbildung 5.2: Benutzeroberfläche der Kontrollsoftware im Videomodus des ADNS 3080

5.1.2 Steuerungsfunktionen

Die Sollposition für den Positionsregeler lässt sich auf zwei Arten durch sogenannte Waypoints in der Kontrollsoftware einstellen. Durch einen Mausklick mit der linken Taste im Koordinatensystem des *Wegpunkte*-Reiters wird ein Wegpunkt gesetzt. Alternativ kann er manuell mit Hilfe der zwei Spinboxen exakt platziert und mit dem Button *Add Waypoint* schließlich gesetzt werden. Der jeweils nächste Wegpunkt wird durch Klicken des Buttons *Send Next* an den Quadrocopter übertragen. Durch einen Mausklick mit der rechten Taste oder durch Betätigen des Buttons *Del. Waypoint* wird der zuletzt gesetzte Wegpunkt gelöscht (vgl. Abbildung 5.1, linker Teil).

In den Kamerafenstern *ADNS 3080* und *OV 7670* lassen sich der Parameter der optischen Systeme konfigurieren. Sie werden in beiden Fenstern durch den *Apply* Button übertragen. Es kann ebenfalls in beiden Fenstern zwischen den unterschiedlichen Arbeitsmodi gewechselt werden. Zur Auswahl stehen *Snapshot*, *Video* und *Optischer Fluss*. Im Snapshotmodus wird ein einzelnes Bild des jeweiligen Kamerasystems übertragen. Im Videomodus werden kontinuierlich Bilder des optischen Systems gesendet. Wählt man dagegen *Optischer Fluss*, so berechnet das optische System den optischen Fluss und gibt die aktuelle Position an die Kontrollsoftware zurück. Der *Clear Motion* Button setzt die aktuelle Position auf $(0, 0)$ zurück. Je nach optischem System lassen sich unterschiedliche Parameter setzen. Im *ADNS 3080* Reiter lassen sich die Bildrate (*Framerate*), die Belichtungszeit (*Shutterspeed*) und die Auflösung (*Resolution*) ändern. Im *OV 7670* Reiter kann neben der *Höhe* und der *Breite* auch die Bildrate (*Framerate*) eingestellt werden. Zusätzlich können hier die zwei Konfigurationsparameter *Of Scale* und Δ *Max* zur Berechnung des optischen Flusses verändert werden(vgl. Abbildung 5.2, linker Teil).

5.2 Implementierung eines optischen Systems

Um die Position des Quadrocopters zu bestimmen muss, wie in Abschnitt 4.1.2 beschrieben, als erstes ein geeignetes optisches System zur Berechnung des optischen Flusses gefunden werden. Dazu werden zunächst verschiedene Ansätze untersucht und anschließend bewertet. Das System soll bei einem Abstand von einem Meter zur betrachteten Oberfläche zuverlässige Werte für den optischen Fluss berechnen. Darüber hinaus soll es einfach in das bestehende System integriert werden können. Idealerweise sollte das System bereits auf dem AVR UC3A Eval Board laufen, da dieses momentan zur Steuerung des Quadrocopters verwendet wird.

5.2.1 Centeye Vision Chip Tam2

Der Vision Chip Tam2 ist ein speziell entwickelter CMOS Sensor von Centeye. Das verwendete Breakoutboard des Chips wurde für Arduinoboards entwickelt und besitzt ein festangebrachtes Weitwinkelobjektiv. Der Sensor besitzt eine maximale Auflösung von 16×16 Pixeln. Auf der Produktseite von Centeye wird der Sensor inklusive Quellcode angeboten[Centeye, 2012]. Aus diesem Grund und der einfachen Handhabung von Arduinoboards eignet sich dieser Sensor zum Einstieg in das Thema optischer Fluss. Abbildung 5.3 zeigt den Vision Chip Tam2 und ein Arduinoboard.

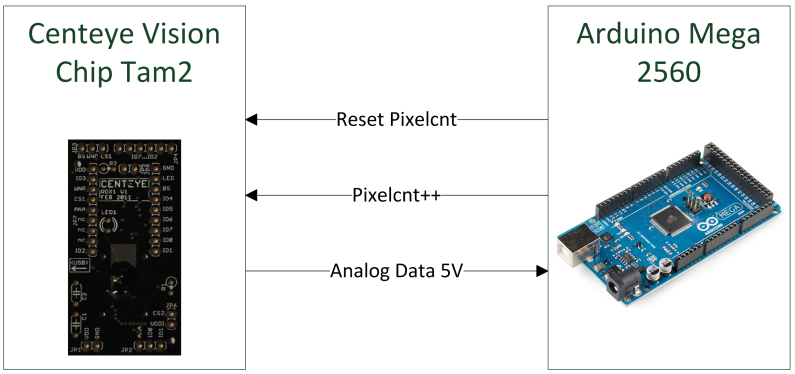


Abbildung 5.3: Vision Chip Tam2 und Arduino Mega 2560

5.2.1.1 Implementierung der Schnittstelle

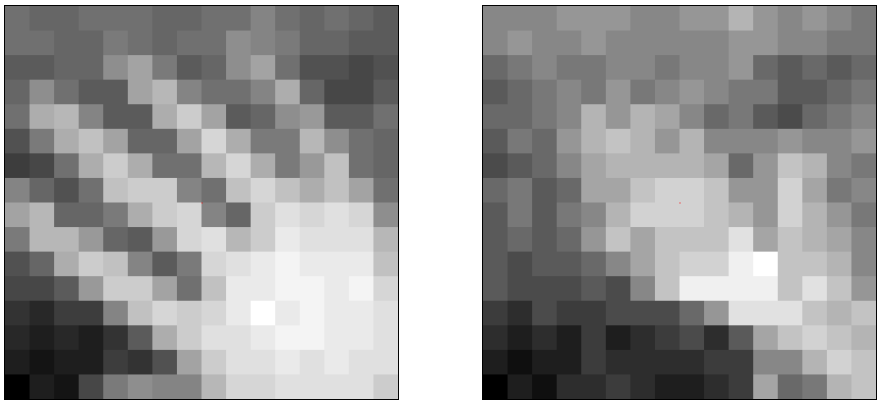
Die Schnittstelle des Vision Chips Tam2 besteht aus drei Signalen. Im Sensor ist ein Zähler integriert, der als Pointer auf den jeweils auszulesenden Pixel fungiert. Über das *Analog Data 5V* Signal wird der analoge Spannungswert des jeweiligen Pixels ausgelesen. Mit Hilfe des Analog-Digital-Wandlers des Arduinoboards erhält man schließlich den digitalen Wert des Pixels. Über das *Reset Pixelcnt* Signal wird der Pointer auf 0, also den ersten Pixel zurück gesetzt. Mit dem Signal *Pixelcnt++* lässt sich der Pointer auf die Pixel inkrementieren. Um ein vollständiges Bild auszulesen benötigt man mit dem Arduino Mega 2560 Board ungefähr 35ms.

5.2.1.2 Berechnung des optischen Flusses

Der optische Fluss wird mit Hilfe des in Abschnitt 2.6 beschriebenen Algorithmus berechnet. Das Auslesen eines Bildes, inklusive der anschließenden Verarbeitung, benötigt auf dem Arduino Mega 2560 bei einer Taktung von 16MHz ungefähr 40ms. Das entspricht einer Bildrate von 25fps[1].

5.2.1.3 Bewertung des Systems

Da die Schnittstelle und die Berechnung des optischen Flusses bereits im Quellcode des Herstellers implementiert sind, kann das optische System schnell in Betrieb genommen und getestet werden. Abbildung 5.4(a) zeigt eine Hand in ca. 10cm, Abbildung 5.4(b) in ca. 25cm Entfernung.



(a) Hand in ca. 10cm Entfernung (b) Hand in ca. 25cm Entfernung

Abbildung 5.4: Bild des Vision Chips Tam2 bei unterschiedlichen Entfernungen

Das festangebrachte Weitwinkelobjektiv besitzt offensichtlich eine zu geringe Brennweite und ein zu großes Blickfeld um in einer Höhe von einem Meter effizient zu arbeiten. Feine Texturen auf der Bodenoberfläche können mit diesem System nicht erkannt werden. Einfache Tests mit Objekten, deren Abstand zum Objektiv wenige cm klein ist, aber zeigen, dass sich der verwendete Algorithmus bei der momentanen Bildrate von ca. 25fps gut eignet, um Bewegungen zu erkennen und den optischen Fluss zu berechnen. Da das Weitwinkelobjektiv festangebracht ist und sich das optische System somit nicht für den Einsatz an einem Quadrocopter eignet, wird an dieser Stelle auf eine ausführliche Evaluierung des optischen Systems verzichtet. Abschließend liefert dieses System aber wichtige Kenntnisse bei der Berechnung des optischen Flusses. Da der verwendete Algorithmus gut funktioniert, sollte eine Bildrate im Bereich von mindestens 25fps angestrebt werden. Darüberhinaus ist es wichtig, dass das verwendete Objektiv eine möglichst große Brennweite hat, um feine Texturen auch bei größeren Abständen zu erkennen. Mit 50U$ ist

[1] **F**rames **p**er **S**econd (FPS)

der Tam2 Vision Chip allerdings im Vergleich zu anderen CMOS Chips teuer. Damit eine Systemintegration möglich ist, wird ein zusätzlicher 5V Analog-Digital-Wandler benötigt, da das AVR Board nur 3V Wandler besitzt.

5.2.2 4D Systems uCAM

Die uCAM von 4D Systems ist ein integriertes, serielles Kameramodul. Als optischer Sensor wird ein OmniVision VGA Color CMOS Sensor verwendet. Über eine serielle **T**ransistor-**T**ransistor-**L**ogik (TTL) Schnittstelle kann zwischen verschiedenen Formaten gewählt. Neben verschiedenen Auflösungen kann auch das Bildformat eingestellt werden. Die Bilder können in Graustufen mit unterscheidlicher Tiefe, im 12- und 16-bit RGB Format und als JPEG ausgegeben werden. Die Bildausgabe erfolgt ebenfalls über die serielle TTL Schnittstelle [4DSystems, 2011]. Abbildung 5.5 zeigt die Anbindung des uCam-Moduls an das AVR UC3A Eval Board.

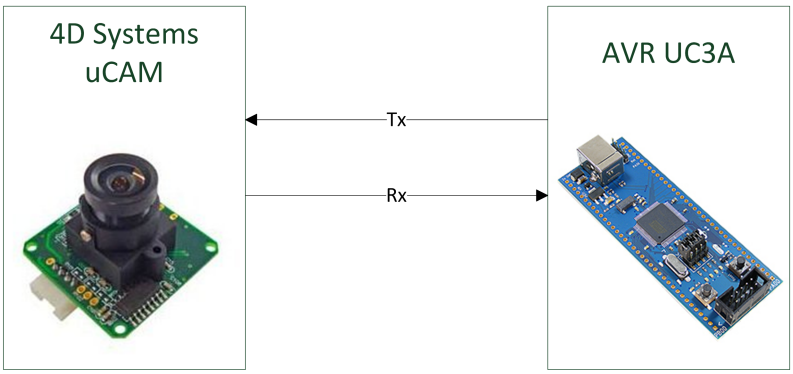


Abbildung 5.5: Mögliche Integration der uCam

5.2.2.1 Implementierung der Schnittstelle

Die serielle TTL Schnittstelle der Kamera wird mit einer UART Schnittstelle des AVR UC3A Boards verbunden. Das uCam-Modul wird anschließend, wie im Datenblatt [4DSystems, 2011] beschrieben, so konfiguriert, dass die aufgenommenen Bilder mit einer minimalen Auflösung von 80×60 im 8-bit Graustufen Format übertragen werden. Damit fällt pro Bild eine Datenmenge von 4,8kB an. Bei einer maximalen Baudrate der Kamera von 921600 Baud mit 8 Datenbits und je einem Start- bzw. Stopbit beträgt die theoretische Übertragungsdauer für ein Bild demnach:

$$4800Byte \cdot \frac{1}{92160Byte/s} = 52.08ms \tag{5.1}$$

Messungen zeigen jedoch, dass pro Bild zwischen 78 und 94ms vergehen. Die zusätzliche Zeit wird zur Aufnahme und Verarbeitung (evtl. Formatanpassung durch das Kameramodul) benötigt. Damit ergibt sich eine obere Schranke der Bildrate bei 13fps.

5.2.2.2 Bewertung des Systems

Die uCam ist einfach anzuschließen und leicht in ein beliebiges System zu integrieren. Die maximal mögliche Bildrate reicht jedoch nicht aus, um den optischen Fluss zu berechnen. Hinzu kommt, dass die maximal mögliche Baudrate des AVR UC3A bei 368640 Baud liegt. Bei Verwendung des AVR UC3A würde sich die Bildrate also nochmals drastisch verringern. Der Preis der uCam liegt bei ca. 40U$.

5.2.3 CMUCam 4

Die CMUCam 4 verfügt, wie die uCam, über einen eigenen Mikrokontroller auf der Platine. Bei dem Mikrokontroller handelt es sich um einen Propeller Chip der Firma Parallax. Ein Propeller Chip ist im Prnzip ein Mikrokontroller mit acht parallel arbeitenden Prozessorkernen (sogenannten Cogs), die in SPIN, einer C und Assembler ähnlichen Sprache, programmiert werden. Der verwendete Bildsensor ist der OV 9665 CMOS Sensor von OmniVision. Die CMUCam 4 verfügt zusätzlich über einen MicroSD-Kartenslot, der die Speicherung von Bildern erlaubt. Die aktuelle Firmware ermöglicht das Auslesen und Speichern von Bildern. Ein Colortrackingalgorithmus ist ebenfalls implementiert. Auf der Herstellerseite[CMUCam, 2012] kann man den Quellcode zur aktuellen Firmware finden. Abbildung 5.6 veranschaulicht eine mögliche Integration der CMUCam 4 in das aktuelle System. Analog zur uCam kann die CMUCam 4 über eine serielle Schnittstelle mit dem AVR UC3A verbunden werden.

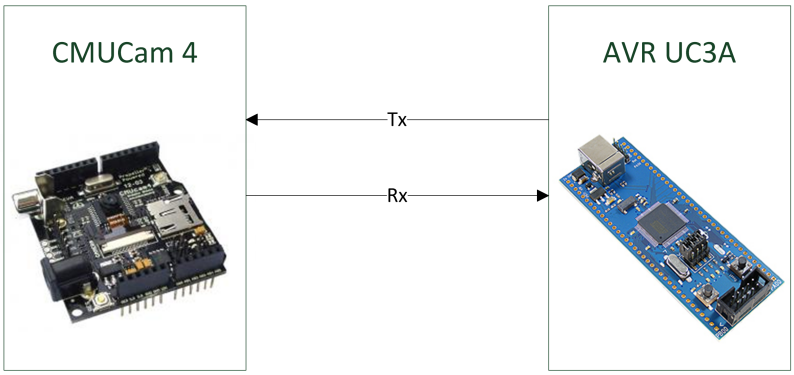


Abbildung 5.6: Mögliche Integration der CMUCam 4

Damit mit diesem System eine hohe Bildrate bei der Berechnung des optischen Flusses erzielt werden kann, muss die Berechnung mit Hilfe des Propeller Chips durchgeführt werden. Eine Übertragung der Bilddaten an einen anderen Mikrokontroller wäre nur über die UART Schnittstelle möglich und damit wieder zu langsam. Um den optischen Fluss auf dem Propeller Chip zu berechnen, muss allerdings eine komplett neue Firmware entwickelt werden. Die Entwicklung einer solchen Firmware ist aufgrund der mangelhaften Dokumentation seitens der Hersteller und

der eigenen Programmiersprache des Propeller Chips mit einer sehr langen Einarbeitungszeit verbunden. Erschwerend kommt der geringe Arbeitsspeicher von 32kB hinzu. In ihm müssen sämtliche Variablen, die notwendigen Bilder zur Berechnung des optischen Flusses sowie der gesamte Programmcode Platz finden. Auch eine Verwendung des zusätzlichen MicroSD-Kartenslots könnte das Speicherproblem nicht lösen, da dadurch der Programmablauf deutlich verlangsamt wird. Aufgrund dieser Mängel eignet sich die CMUCam 4 nicht zum Berechnen des optischen Flusses. Darüberhinaus ist die CMUCam 4 mit knapp 100U$ das teuerste System, das in dieser Arbeit betrachtet wird.

5.2.4 OV 7670 und STM32 F4

Das optische System aus dem OV 7670 CMOS Sensor von OmniVision und dem STM32 F4 Discovery Board versucht die Probleme der bisher beschriebenen Systeme zu überwinden. Das Objektiv des OV 7670 Breakoutboards kann, anders als beim Vision Chip Tam2 Breakoutboard, beliebig gewechselt und die Brennweite dadurch entsprechend angepasst werden. Im Gegensatz zur uCam erfolgt die Bildübertragung zwischen Bildsensor und Mikrokontroller zur Berechnung des optischen Flusses nicht seriell, sondern über eine parallele Schnittstelle, wodurch höhere Bildraten möglich sind. Zusätzlich wird das STM32 F4 Discovery Board anders als die CMUCam 4 in C programmiert. Abbildung 5.7 zeigt die Anbindung des OV 7670 Sensors an das STM32 F4 Discovery Board.

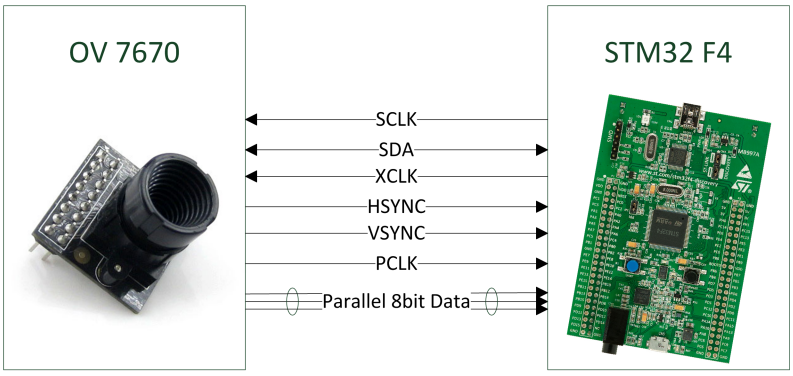


Abbildung 5.7: Anbindung des OV 7670 an STM32 F4 Discovery Board

5.2.4.1 Implementierung der Schnittstellen

Um mit dem OV 7670 Bildsensor arbeiten zu können, müssen verschiedene Schnittstellen des STM32 F4 Discovery Boards implementiert werden. Zur Kommunikation mit einem externen System, z.B. einem PC zum Debuggen und Anzeigen der Bilder, wird eine UART Schnittstelle implementiert. Zur Initialisierung des CMOS Sensors müssen die Kameraregister über ein **I**nter-**I**ntegrated **C**ircuit (I2C) ähnliches Protokoll beschrieben werden. Daher wird ebenfalls die I2C

Schnittstelle des Boards implementiert. Das eigentliche Auslesen der Kamerabilder findet über eine parallele Schnittstelle, der **D**igital **Cam**era **I**nterface (DCMI) Schnittstelle, statt. Die DCMI Schnittstelle besteht aus einem externen Taktsignal (*XCLK*), einem Referenztakt für jeden übertragenen Pixel (*PCLK*), einem Signal, das das Ende einer Bildzeile anzeigt (*HSYNC*) und einem Signal, das das Ende der gesamten Übertragung eines Bildes signalisiert (*VSYNC*). Die Pixel Daten werden dabei über acht parallele Signale übertragen. Jedes Signal steht hierbei für ein bit des aktuell übertragenen Bytes.

Abbildung 5.8 zeigt das horizontale Timing der DCMI Schnittstelle bei der Übertragung einer Bildzeile. Ein übertragenes Byte wird jeweils bei steigender Flanke des *PCLK* Signals gelesen. *HSYNC* ist am Anfang und am Ende der Übertragung einer Bildzeile high, während der Übertragung low.

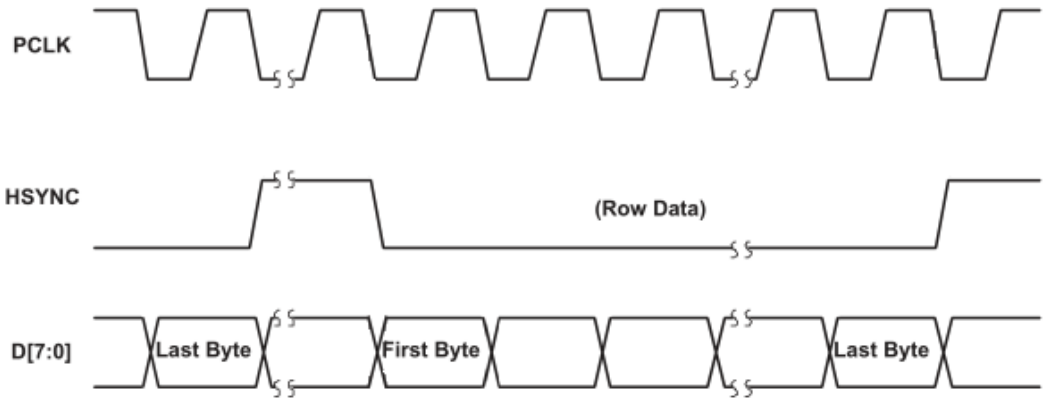


Abbildung 5.8: Horizontales Timingdiagramm der DCMI Schnittstelle[Omnivision, 2006]

In Abbildung 5.9 ist das gesamte Timing bei einer Bildübertragung mit n Pixeln pro Zeile und insgesamt m Zeilen zu sehen. Das *VSYNC* Signal wechselt von low auf high, sobald das gesamte Bild übertragen wurde.

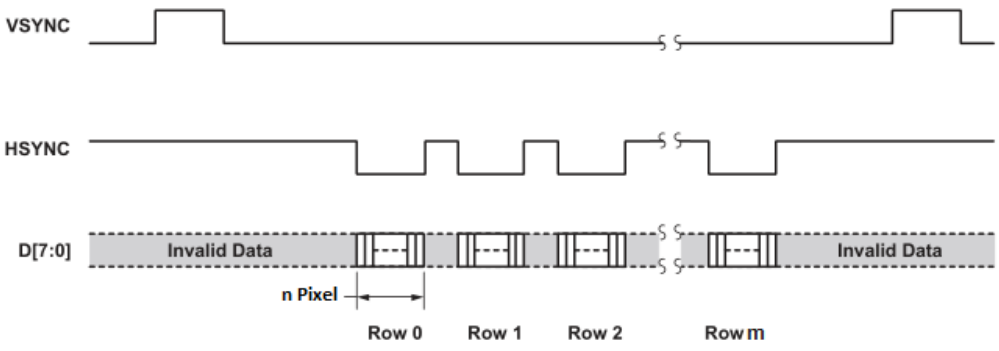


Abbildung 5.9: Gesamtes Timingdiagramm der DCMI Schnittstelle[Omnivision, 2006]

Um das Bild, das über die DCMI Schnittstelle empfangen wird, zu speichern, bietet sich der **D**irect **M**emory **A**ccess (DMA)-Controller des Discovery Boards an. Dabei werden die Bilder von der DCMI Schnittstelle, ohne Umweg über die **C**entral **P**rocessing **U**nit (CPU), direkt in den **R**andom **A**ccess **M**emory (RAM) des Mikrokontrollers gelegt.

5.2.4.2 Berechnung des optischen Flusses

Zur Berechnung des optischen Flusses wird erneut der in Abschnitt 2.6 beschriebene Algorithmus verwendet, da er bereits gute Ergebnisse mit dem Vision Chip Tam2 erzielt hat. Die Bilder des CMOS Sensors werden im RGB565 Format übertragen, da der OV 7670 die Ausgabe im 8bit Graustufenformat nicht unterstützt. Für die Farbkanäle Rot, Grün und Blau wird der optische Fluss einzeln bestimmt und dann der Mittelwert über die drei Farbkanäle berechnet und zurückgegeben.

5.2.4.3 Speicherauslastung und Bildqualität

Da die Bilder per DMA in den Speicher gelegt werden, muss sichergestellt werden, dass ein Bild, das gerade zur Berechnung des optischen Flusses verwendet wird, nicht während der Berechnung überschrieben wird. Anders ausgedrückt dürfen CPU und DMA niemals auf den selben Speicherbereich gleichzeitig zugreifen. Daher sollte der für den DMA Controller reservierte Speicher für mindestens zwei Bilder ausreichen. Per DMA werden die empfangenen Bilder nacheinander in den reservierten Speicher gelegt. Die CPU wartet bis das erste Bild vollständig gespeichert ist und beginnt mit dessen Hilfe die Berechnung des optischen Flusses. Währenddessen wird das zweite Bild über den DMA Controller gespeichert. Nachdem die CPU den optischen Fluss berechnet hat, wartet sie erneut, bis das zweite Bild vollständig im Speicher ist. Anschließend berechnet sie den optischen Fluss mit Hilfe des zweiten Bildes, während per DMA ein drittes Bild an der ehemaligen Speicherstelle des ersten Bildes gespeichert wird. Zusätzlich benötigt der verwendete Algorithmus stets noch ein aktuelles Referenzbild und deshalb Platz für mindestens ein weiteres Bild. Tabelle 5.1 zeigt die Auslastung des 192 kB Arbeitsspeichers für die Speicherung von drei Bildern im RGB565 Format bei verschiedenen Standardauflösungen des OV 7670s:

Tabelle 5.1: Auslastung des Arbeitsspeichers bei verschiedenen Standardauflösungen

Auflösung	Breite	Höhe	Pixel	Speicher	Auslastung RAM
VGA	640	480	307200	1843,2 kB	960,00%
QVGA	320	240	76800	460,8 kB	240,00%
CIF	352	288	101376	608,256 kB	316,80%
QCIF	176	144	25344	152,064 kB	79,20%

Offensichtlich sind die gängigen Standardauflösungen des OV 7670 Sensors zu groß, um drei Bilder im Arbeitsspeicher zu speichern. Im QCIF Format würden sie zwar in den Speicher passen,

aber mit einer Auslastung von fast 80% deutlich zu viel Platz belegen. Zusätzlicher, externer Speicher ermöglicht zwar das Speichern besserer Bilder, verlangsamt aber auch gleichzeitig das System durch längere Speicherzugriffszeiten und einen höheren Rechenaufwand bei der Bestimmung des optischen Flusses. Natürlich könnten die Bilder durch den CMOS Sensor entsprechend herunterskaliert werden, sie leiden dabei jedoch an einem gravierenden Qualtitätsverlust, da mehrere Pixel mathematisch zusammengefasst werden und Rundungsfehler entstehen. Eine Verschlechterung der Bildqualtität bei den ohnehin sehr feinen Texturen einer Bodenoberfläche wäre fatal bei der Berechnung des optischen Flusses. Eine Lösung für dieses Problem ist das Crop Feature des Discovery Boards. Mit seiner Hilfe lassen sich Bilder mit hoher Auflösung empfangen, wobei aber nur kleinere Teilbilder gespeichert werden.

Abbildung 5.10 zeigt zwei Bilder. Beide wurden aus der selben Position, ca. 50cm von der betrachteten Szene entfernt, aufgenommen. Bild (a) wurde im QVGA Format aufgenommen und über das Crop Feature mit einer Auflösung von 100×100 Pixeln gespeichert. Bild (b) wurde im QQVGA Format aufgenommen und ebenfalls mit einer Auflösung von 100×100 Pixeln gespeichert. Beim QQVGA Format skaliert der Bildsensor intern runter, daher ist der betrachtete Ausschnitt der aufgenommen Szene auch größer. Der Qualitätsverlust durch das Herunterskalieren ist sehr deutlich sichtbar.



(a) QVGA Format (b) QQVGA Format

Abbildung 5.10: Bilder des OV7670 Sensors in unterschiedlichen Formaten

5.2.4.4 Bewertung des Systems

Eine entscheidende Schlüsselstelle bei der Implementierung dieses Systems ist die Initialisierung der Kameraregister. Das Datenblatt beschreibt nur ansatzweise die umfangreichen Funktionen der einzelnen Register. Um den Sensor dennoch vernünftig zu initialisieren werden schließlich die Registerwerte aus dem Beispielcode des Sensors übernommen[2]. Mit einem externen Taktsignal

[2]Beispielcode zu OV7670 Kamera und ATmega32 Board von egochina, `www.egochina.net.cn/e-shop/eBay/Datasheet/ATmega32_tft_ov7670_REV4.1.rar`, aufgerufen am 03.09.2012

von 16MHz wird dabei eine Bildrate von 20fps erreicht. Durch eine manuelle Änderung des internern Prescalers und der **P**hase-**L**ocked **L**oop (PLL) des CMOS Sensors kann die Bildrate auf 30fps erhöht werden. Durch die ausreichend gute Bildqualität und die hohe Bildrate erfüllt dieses System die zwei Voraussetzungen, die zur Berechnung des optischen Flusses notwendig sind. Da bereits einfache Tests zeigen, dass es möglich ist, den optischen Fluss mit diesem System zu bestimmen, ist an dieser Stelle eine genaue Evaluierung des Systems notwendig. Abschnitt 6.1.2 behandelt die Evaluierung dieses Systems ausführlich. Mit ca. 15U$ für die Kamera und 20U$ für das Board ist dies die günstigste Lösung.

5.2.5 ADNS 3080

Der ADNS 3080 Sensor von Avago Technologies ist im Gegensatz zu den anderen, bereits vorgestellten optischen Systemen, ein fertiger optischer Flusssensor. Für gewöhnlich werden solche Sensoren in optischen Mäusen verbaut. Da auf dem Breakoutboard des Sensors verschiedene Objektive einfach angebracht werden können, kann er auch auf größere Distanzen verwendet werden. Die Kommunikation mit dem Sensor findet über das Standard **S**erial **P**eripheral **I**nterface (SPI) statt. Der Sensor arbeitet mit einer programmierbaren Bildrate von bis zu 6400fps. Neben der Bildrate kann auch die Belichtungszeit frei eingestellt werden. Beide Werte müssen aber nicht zwangsweise manuell eingestellt werden, da der Sensor für beide Größen jeweils die Option zum automatischen Einstellen zur Verfügung stellt. Zusätzlich zum optischen Fluss berechnet der Sensor einen Wert zur Quantifizierung der betrachteten Oberfläche. Dieser Wert beschreibt, wie gut sich die Oberfläche eignet, um den optischen Fluss zu bestimmen. Da zur Berechnung des optischen Flusses kein weiterer Mikrokontroller notwendig ist, kann der Sensor direkt an den Mikrokontroller zur Steuerung des Quadrocopters angeschlossen werden. Abbildung 5.11 skizziert den Anschluss des Sensors an das AVR UC3A Eval Board.

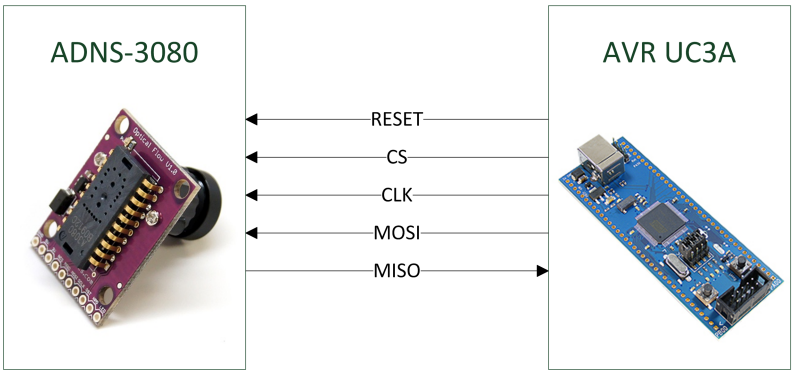


Abbildung 5.11: Mögliche Integration des ADNS 3080 Sensors

5.2.5.1 Implementierung der Schnittstelle

Die Schnittstelle des Sensors ist mit Hilfe des AVR Frameworks schnell implementiert. Über das *CS* Signal wird der Sensor zur Kommunikation ausgewählt. Synchron zum Taktsignal (*CLK*) werden Daten vom Sensor über das *MISO* Signal gelesen und über das *MOSI* Signal zum Sensor geschickt. Bevor man mit dem Sensor arbeiten kann, ist ein Neustart über das *Reset* Signal durchzuführen.

5.2.5.2 Berechnung des optischen Flusses

Die genaue Berechnung des optischen Flusses ist nicht bekannt. Aus der Produktinformation des Herstellers [AvagoTechnologies, 2010] geht nur hervor, dass in zwei aufeinander folgenden Bildern einer Sequenz gemeinsame Merkmale gesucht werden und die Verschiebung dieser Merkmale berechnet wird. Es wird also ein spezieller Algorithmus basierend auf der Korrelationsmethode (vgl. Abschnitt 2.5) verwendet.

5.2.5.3 Bewertung des Systems

Der Sensor ermöglicht eine einfache Integration in das bestehende System bei einer gleichzeitig sehr hohen Bildrate. Die Auflösung des Sensors beträgt dabei nur 30×30 Pixel (vgl. Abbildung 5.12.

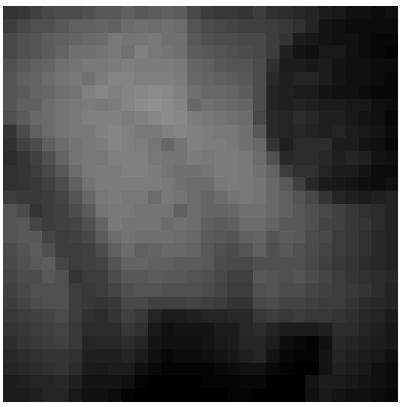


Abbildung 5.12: Bild des ADNS 3080 Sensors

Schnelle, einfache Tests zeigen auch hier, dass mit diesem System sehr einfach der optische Fluss bestimmt werden kann. Neben dem optischen Fluss ermittelt der Sensor auch einen Wert zur Beschreibung der Oberflächenqualität. An dieser Stelle ist ebenfalls eine genaue Evaluierung des optischen Systems notwendig. In Abschnitt 6.1.3 wird die Evaluierung dieses Systems ausführlich behandelt. Der Preis des Sensor liegt bei knapp 40U$.

5.2.6 Auswahl und Integration des optischen Systems

Um in das Gesamtsystem integriert zu werden, stehen damit zwei optische Systeme zur Auswahl. Zum einen das selbstentwickelte System aus dem OV 7670 Sensor und dem STM32 F4 Discovery Board und zum anderen der fertige optische Flusssensor ADNS 3080 von Avago Technologies. Aufgrund der Evaluierungsergebnisse aus Abschnitt 6.1 und der Tatsache, dass nach einem möglichst kleinen, einfach zu integrierenden optischen System gesucht wird, fällt die Wahl auf den optischen Flusssensor. Der optische Flusssensor wird also an der Unterseite des Quadrocopters montiert und an den Mikrokontroller zur Steuerung des Quadrocopters angeschlossen. Der bei der Entwicklung und Evaluierung des Systems entstandene Code wird angepasst und in das bestehende Programm zur Steuerung des Quadrocopters eingepflegt. Der Mikrokontroller des Quadrocopters muss den optischen Flusssensor regelmäßig über SPI auslesen und die Messwerte für den optischen Fluss aufintegrieren, um die Position relativ zu einem Startpunkt zu berechnen. Abbildung 5.13 zeigt die Sensorik am Boden des Quadrocopters. Außen befinden sich zwei Infrarotsensoren für unterschiedliche Distanzen. In der Mitte ist links ein Ultraschallsensor und rechts der optische Flusssensor zu sehen.

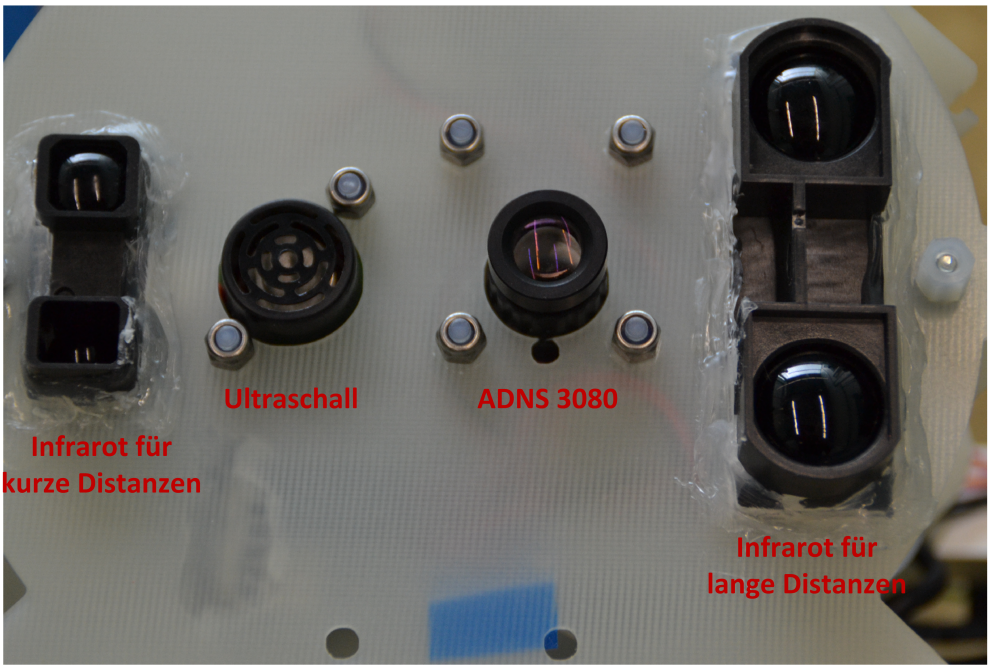


Abbildung 5.13: Sensorik am Boden des Quadrocopters

5.3 Implementierung des Positionsreglers

Der Positionsregler besteht im wesentlichen aus zwei überlagerten PID-Reglern. Je ein Regler berechnet die Stellwerte für eine Neigungsachse des Quadrocopters. Diese Stellwerte werden dann an die Lageregelung übergeben, welche sie dann wiederum in Signale für die Motorsteuerung umrechnet. In den folgenden Abschnitten werden die groben Programmabläufe des gesamten Positionsreglers und die Feinheiten beim Berechnen der Stellwerte durch den PID-Regler erläutert.

5.3.1 Programmablauf des Positionsreglers

Abbildung 5.14 gibt einen vereinfachten Überblick über den Programmablauf des Positionsreglers:

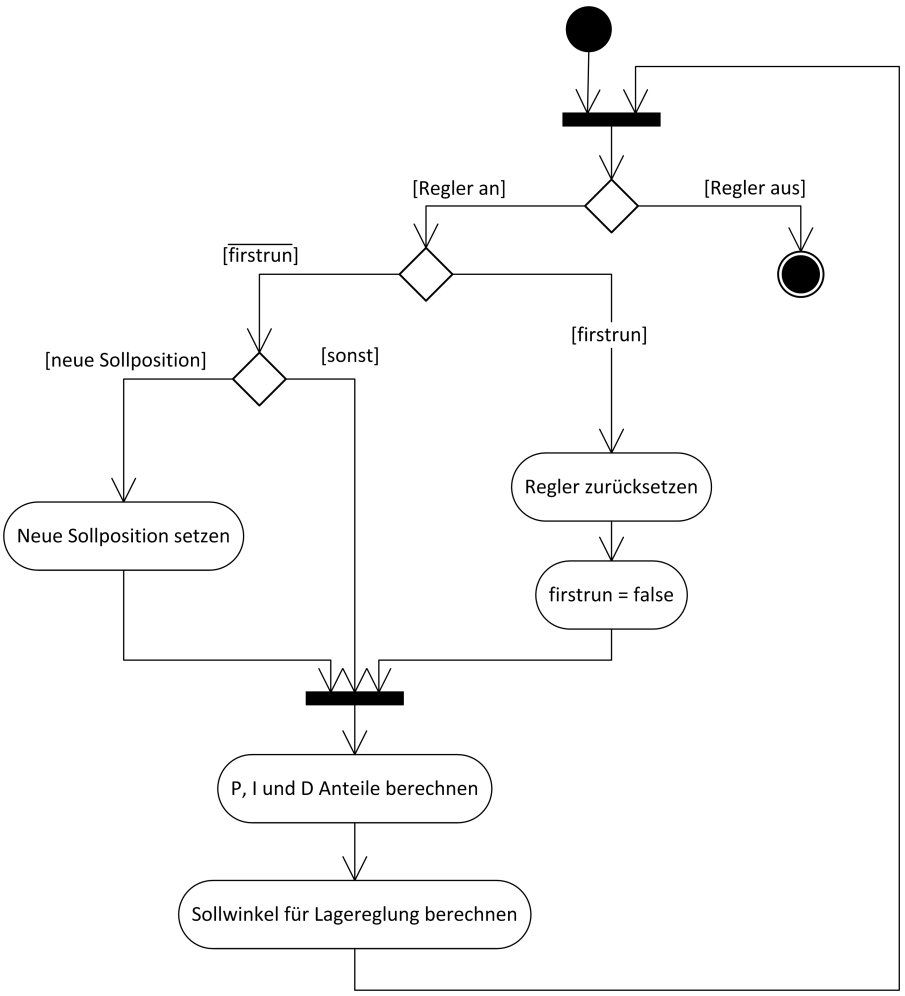


Abbildung 5.14: Vereinfachtes Aktivitätsdiagramm des Positionsreglers

Über einen Schalter an der Fernsteuerung kann der Positionsregler ein- bzw. ausgeschaltet werden. Die Position wird nur dann geregelt, wenn der entsprechende Schalter an der Fernsteuerung aktiviert ist. Jedesmal, wenn der Regler eingeschaltet wird, wird die Variable *firstrun* auf *true*

gesetzt. Falls *firstrun* auf *true* gesetzt ist, wird der Positionsregler zurückgesetzt. Hierbei wird die Fehlerhistorie des Reglers, die aktuelle Position und die Sollposition auf Null gesetzt. *firstrun* wird nach dem Zurücksetzten des Reglers auf *false* gesetzt. Anschließend werden die P, I und D Anteile und die Sollwinkel für die Lageregelung berechnet. Der Programmablauf beginnt danach von vorne, falls der Positionsregler noch aktiviert ist. Da *firstrun* jetzt *false* ist, wird dieser Schritt übersprungen. Falls an dieser Stelle bereits eine neue Sollposition über die Kontrollsoftware an den Quadrocopter übermittelt wurde, wird die neue Sollposition des Reglers auf die übermittelte Sollposition gesetzt. Anschließend folgt erneut die Berechnung die P, I und D Anteile und der Sollwinkel.

In den nächsten zwei Unterpunkten werden die Schritte zur Bestimmung der P, I und D Anteile, sowie zur Berechnung der Sollwinkel detailliert beschrieben.

5.3.1.1 Berechnung der P, I und D Anteile

Wie bereits zu Beginn des Abschnitts 5.3 erläutert, handelt es sich beim Positionsregler um zwei überlagerte PID-Regler. Beide Regler sind exakt identisch, bis auf die Richtung, in die sie arbeiten. Es reicht daher eine allgemeine Beschreibung des Reglers, ohne die einzelnen Richtungskomponenten näher zu erläutern. Im ersten Schritt wird der Positionsfehler e_{pos} aus der Differenz der Istposition i_{pos} und der Sollposition s_{pos}:

$$e_{pos}(t) = i_{pos}(t) - s_{pos}(t) \tag{5.2}$$

Anschließend werden die P, I und D Anteile bestimmt. Anders als bei der Standardimplementierung eines PID-Reglers wird bei dieser Implementierung der D-Anteil zusätzlich geglättet, um große Spitzen im D-Anteil und damit eine ruckartige Regelung zu vermeiden. Diese wird realisiert, indem der Mittelwert über die letzten n Änderungen des Positionsfehlers gebildet wird. Die Berechnung der einzelnen Teile ergibt sich damit wie folgt:

$$p = K_p \cdot e_{pos}(t) \tag{5.3}$$

$$i = K_i \cdot \int e_{pos}(t)dt \tag{5.4}$$

$$d = K_d \cdot \frac{1}{n} \cdot \sum_{i=t-n}^{t} \frac{d}{dt} e_{pos}(i) \tag{5.5}$$

5.3.1.2 Berechnung der Sollwinkel für die Lageregelung

Die Sollwinkel Φ_{Roll} und Θ_{Pitch} für die Lageregelung werden aus der Summe der Stellwinkel des Positionsreglers u_{Roll} und u_{Pitch} und den Stellwinkeln der Fernsteuerung v_{Roll} und v_{Pitch} berechnet:

$$\Phi_{Roll} = u_{Roll} + v_{Roll} \tag{5.6}$$

$$\Theta_{Pitch} = u_{Pitch} + v_{Pitch} \tag{5.7}$$

Da die Stellwinkel der Fernbedienung bekannt sind, müssen nur die Stellwinkel des Positionsreglers berechnet werden. Um einen maximalen Grad an Stabilität zu gewährleisten, werden dabei die Summe der P- und D-Anteile und der I-Anteil in x- bzw. y-Richtung begrenzt. Diese Begrenzung ist aus verschiedenen Gründen sinnvoll. Große Kippwinkel, die zu einem Absturz oder unkontrollierbaren Verhalten des Quadrocopters führen könnten, werden auf diese Weise vermieden. Darüberhinaus kann durch verschiedene Begrenzungen der beiden Teile gesteuert werden, in welchem Verhältnis der PD-Anteil und der I-Anteil zur Regelung beitragen. Die Begrenzung des PD-Anteils wurde auf

$$|p + d| \leq 5^\circ$$

festgesetzt. Die Begrenzung des I-Anteils wird größer gewählt als die des PD-Anteils, da der I-Anteil zur Driftkompensation verwendet wird

$$|i| \leq 10^\circ$$

Die Stellwinkel ergeben sich dann aus der Summe der P-, I- und D-Anteile:

$$u_{Roll} = K_i \cdot \int e_{posx}(t)dt + K_d \cdot \frac{d}{dt} e_{posx}(t) + K_d \cdot \frac{1}{n} \cdot \sum_{i=t-n}^{t} \frac{d}{dt} e_{posx}(i) \tag{5.8}$$

$$u_{Pitch} = -\left(K_i \cdot \int e_{posy}(t)dt + K_d \cdot \frac{d}{dt} e_{posy}(t) + K_d \cdot \frac{1}{n} \cdot \sum_{i=t-n}^{t} \frac{d}{dt} e_{posy}(i) \right) \tag{5.9}$$

5.3.2 Einstellen des Positionsreglers

Um die Regelparameter K_i,K_p und K_d zu bestimmen, wird der Positionsregler im Flug aktiviert und die entsprechenden Parameter durch gezieltes Probieren, empirisch ermittelt. Folgende Werte haben sich bei einem verwendeten Objektiv mit 16mm Brennweite in einer Höhe von einem Meter sehr gut geeignet:

$$P = 0.0125, I = 0.000015, D = 3.5$$

5.4 Kalibrierung des fertigen Systems

Das optische System am Boden des Quadrocopters bestimmt dessen Position abhängig davon, wie sich einzelne Pixel im Bild des Sensors verschieben. Die errechnete Verschiebung liegt also in Pixeln pro Sekunde vor, die Position dementsprechend in Pixeln. Zur Umrechnung in das metrische System muss ein Skalierungsfaktor gefunden werden, der die Positionsdaten ineinander umrechnet.

5.4.1 Problematik bei der Kalibrierung des Systems

Dieser Skalierungswert ist allerdings zum einen von der aktuellen Höhe und zum anderen von den Kameraparametern abhängig. Abbildung 5.15 zeigt den Einfluss unterschiedlicher Höhen auf das Bild des Quadrocopters. Je höher sich der Quadrocopter befindet, desto größer ist der betrachtete Bildbereich. Gleichgroße Bewegungen des Quadrocopters bewirken je nach Höhe unterschiedlich große Helligkeitsverschiebungen und damit unterschiedliche Messungen des optischen Flusses. Bewegt man beispielsweise einen Quadrocopter mit optischem System in einem Meter Höhe um 10cm nach links, verschiebt sich die äußerste Pixelreihe am linken Rand einmal durch das ganze Bild. Bewegt man den selben Quadrocopter dagegen in zwei Meter Höhe um 10cm nach links, verschiebt sich der linke Pixelrand lediglich in die Mitte des Bildes.

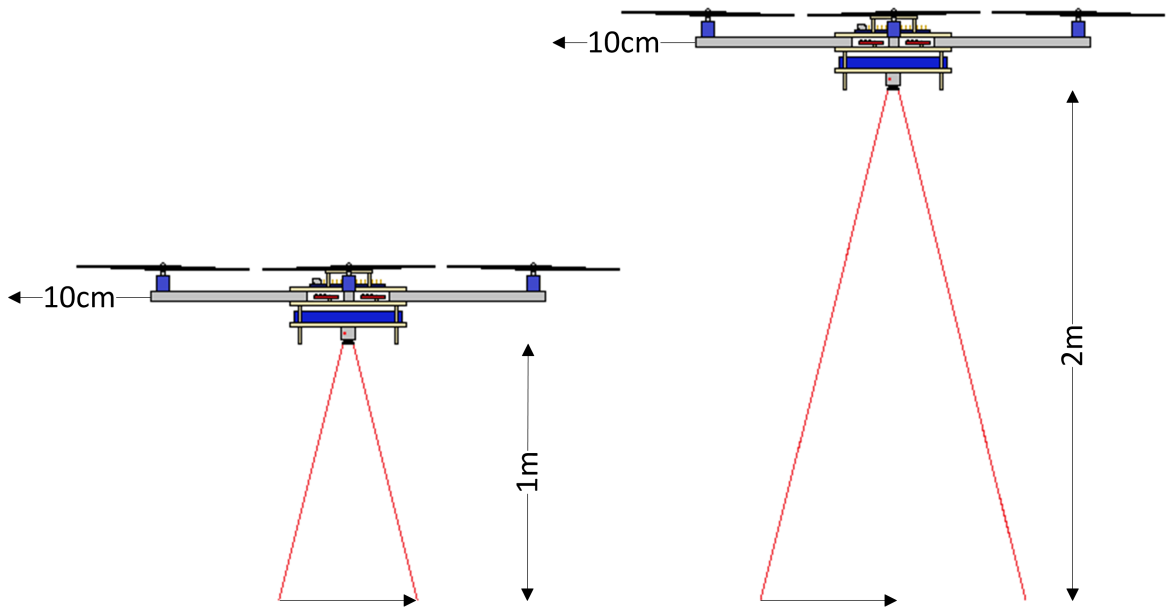
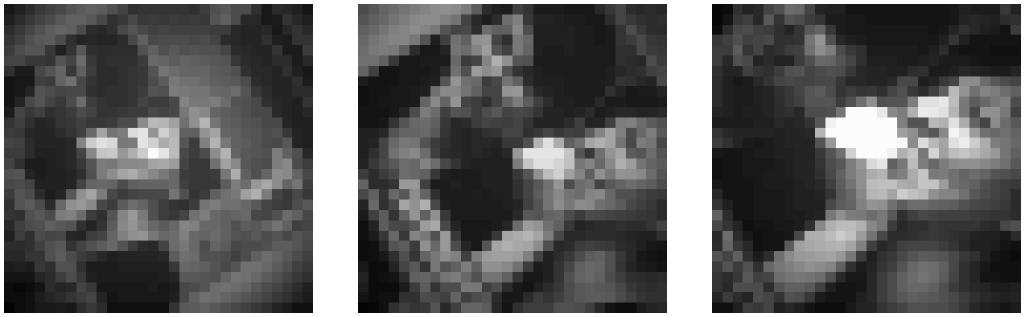


Abbildung 5.15: Einfluss der Höhe auf die Messwerte des optischen Sensors[Arducopter, 2012]

Ein ähnlicher Effekt tritt durch die Verwendung verschiedener Objektive auf. Je höher die Brennweite des benutzten Objektives ist, desto kleiner ist der betrachtete Bildbereich. Abbildung 5.16 zeigt das Bild des ADNS 3080 Sensor bei der Verwendung verschiedener Objektive.



(a) 8mm Brennweite, 11° Bildwinkel (b) 12mm Brennweite, 29° Bildwinkel (c) 16mm Brennweite, 21° Bildwinkel

Abbildung 5.16: Bild des ADNS 3080 Sensor bei Verwendung unterschiedlicher Objektive

5.4.2 Messung zur Kalibrierung des Systems

Um den Skalierungswert zu bestimmen, muss also ein Versuch in einer vorher bestimmten Flughöhe mit festen Kameraparametern durchgeführt werden.

5.4.2.1 Versuchsbeschreibung

Die Flughöhe wird auf einen Meter festgelegt und es wird ein Objektiv mit einer Brennweite von 16mm verwendet. Der Quadrocopter wird per Hand entlang einer Achse mehrmals zwei Meter hin- und herbewegt. Das optische Tracking stellt dabei sicher, dass eine Höhe von einem Meter eingehalten wird. Es werden sowohl die Positionsdaten des optischen Flusses (Odometrie), als auch die Positionsdaten, die durch das optische Tracking ermittelt werden, aufgezeichnet. Um ungefähr gleich viele Messpunkte bei beiden Verfahren zu erhalten, wird für beide Messungen eine identische Samplerate von 100ms eingestellt. Eine anschließende Auswertung der Messwerte liefert den gewünschten Skalierungsfaktor.

5.4.2.2 Auswertung

Um den Skalierungsfaktor zu bestimmen wurde eine Messreihe mit mehreren Messzyklen analysiert. Ein Zyklus besteht dabei aus einer zwei Meter Bewegung entlang einer Achse und wieder zurück. Da das optische Tracking und die Positionsbestimmung des Quadrocopters zwei voneinander unabhängige Systeme sind, ist es nicht möglich, die Messzeitpunkte der beiden Systeme zu synchronisieren. Dies macht eine exakte Bestimmung des Skalierungsfaktor nicht einfach. Es werden daher verschiedene Methoden angewendet, um den Faktor zu bestimmen. Anschließend

wird ein Mittelwert über diese Methoden gebildet und der Fehler des Skalierungsfaktors als Maximalfehler angegeben.

Abbildung 5.17 zeigt die aufgenommene Messreihe bei diesem Versuch.

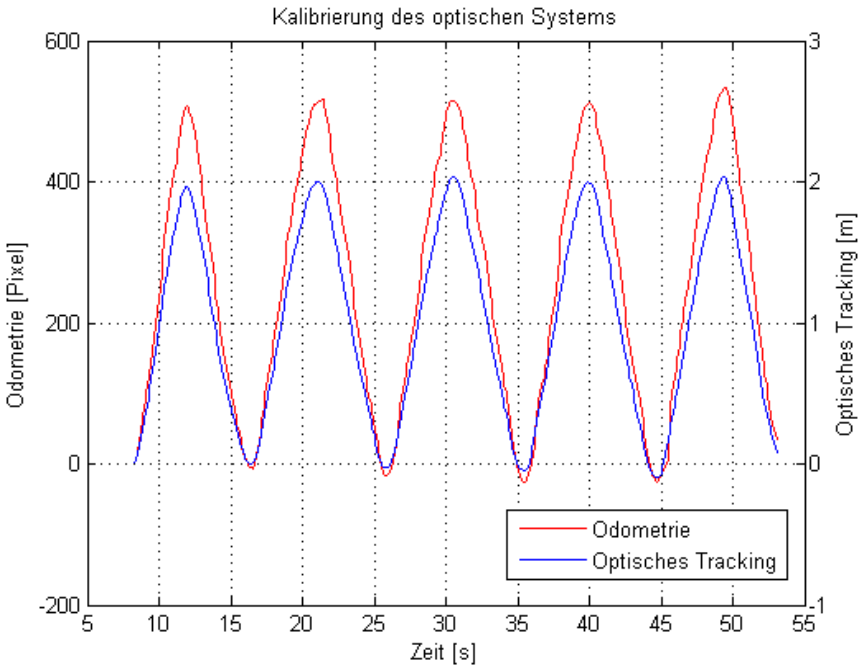


Abbildung 5.17: Messreihe zur Kalibrierung des Systems

Die **1. Methode** zur Bestimmung des Skalierungsfaktors basiert auf der Methode der kleinsten Fehlerquadrate. Es wird hierbei versucht, einen Skalierungsfaktor s zu bestimmen, sodass die Summe der quadratischen Fehler zwischen den skalierten Messwerten der Odometriekurve $o(i)$ und dem optischen Tracking $t(i)$ minimal ist. i ist dabei der i-te Messpunkt der Messreihe

$$\sum_{i \in Messreihe} (o(i) \cdot s - t(i))^2 = Minimal \tag{5.10}$$

Bei der **2. Methode** wird der Mittelwert beider Kurven gebildet und durch eine anschließende Division der Faktor bestimmt.

Die **3. Methode** berechnet zuerst die Differenzen zwischen je zwei benachbarten lokalen Minima und lokalen Maxima der einzelnen Kurven. Anschließend wird erneut der Mittelwert über alle Differenzen gebildet und durch eine Division der Skalierungsfaktor bestimmt.

Die anschließende Auswertung der einzelnen Methoden ergibt folgende Tabelle:

Tabelle 5.2: Auswertung der einzelnen Methoden zur Bestimmung des Skalierungsfaktors

	1. Methode	**2. Methode**	**3. Methode**	**Ergebnis**	**Fehler**
Pixel in Meter	0,00385	0,00388	0,00386	0,00386	±0,00002
Meter in Pixel	259,51	257,64	259,21	258,79	±1,15

6 Evaluierung

In diesem Kapitel werden die optischen Systeme, die sich für den Einsatz an einem Quadrocopter eignen und das entwickelte Gesamtsystem evaluiert. Im ersten Teil dieses Kapitels findet ein Vergleich der beiden optischen Systeme statt, bei dem diese unter gleichen Bedingungen den optischen Fluss messen müssen. Im zweiten Teil wird das fertige Gesamtsystem auf seine Genauigkeit beim Halten und beim Anfliegen von Positionen geprüft.

6.1 Evaluierung der optischen Systeme

Bei der Implementierung der verschiedenen optischen Systeme haben zwei Systeme die notwendigen Voraussetzungen zur Bestimmung des optischen Flusses erfüllt. Das eine System besteht aus dem OV 7670 CMOS Sensor und dem STM 32 F4 Discovery Board. Der optische Flusssensor ADNS 3080 und das zur Steuerung des Quadrocotpers verwendete AVR UC3A Eval Board bilden das zweite optische System. In diesem Abschnitt wird ein Versuch durchgeführt, um die Güte des jeweils berechneten optischen Flusses zu bestimmen.

6.1.1 Versuchsbeschreibung

Damit ein Vergleich zwischen den gemessenen optischen Flüssen möglich ist, müssen bei der Durchführung die selben Bedingungen für die beiden optischen Systeme gewährleistet werden. Abbildung 6.1 zeigt den Versuchsaufbau für den optischen Flusssensor ADNS 3080.

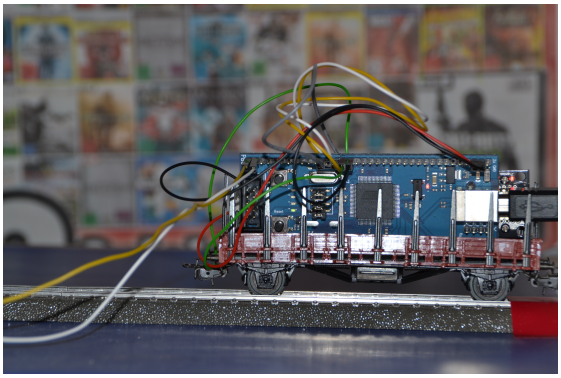


Abbildung 6.1: Versuchsaufbau für den ADNS 3080 Sensor

Die Systeme werden im Abstand von 55 cm parallel zu einer Wand bewegt. Die Wand ist dabei mit einer Seite aus einem Werbekatalog beklebt, um einen ausreichend hohen Kontrast bei der Messung des optischen Flusses sicherzustellen. Die bei der Messung zurückgelegte Strecke beträgt 20 cm. Damit eine rein translatorische Bewegung entlang einer Achse durchgeführt werden kann, wird die Kamera des jeweiligen optischen Systems an einem Waggon einer Modelleisenbahn befestigt und auf passenden Schienen mehrmals bewegt. Bei einem Versuch wird der Waggon mit dem optischen System wiederholt mit unterschiedlicher Geschwindigkeit hin und her geschoben und dabei die Position mit Hilfe des optischen Flusses bestimmt.

6.1.2 Auswertung OV 7670 und STM32 F4

Der Versuch mit dem optischen System aus OV 7670 und STM32 F4 wird exakt analog zur Versuchsbeschreibung durchgeführt. Abbildung 6.2 zeigt die mit Hilfe des optischen Flusses approximierte Position des optischen Systems bei der Versuchsdurchführung.

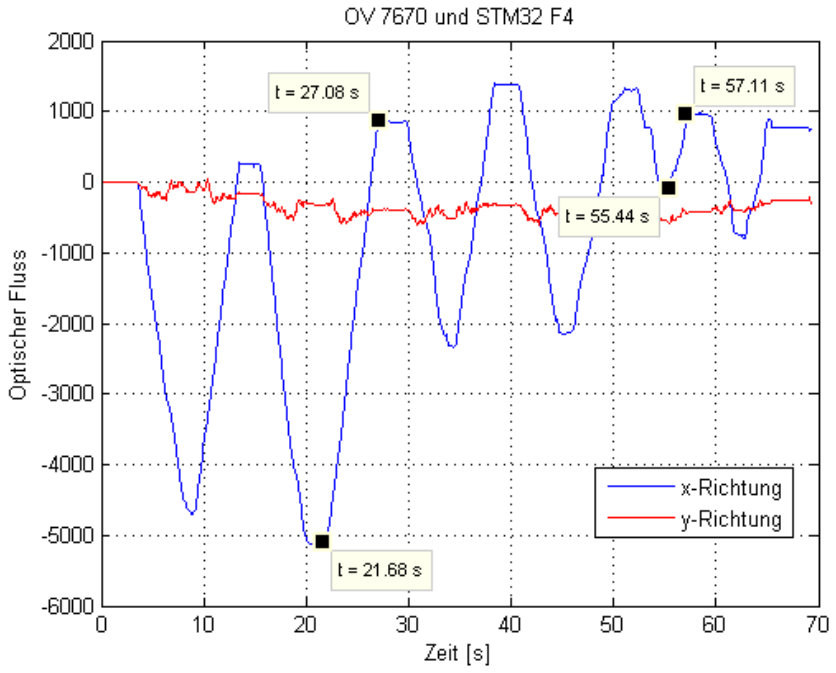


Abbildung 6.2: Positionsbestimmung des optischen Systems aus OV 7670 und STM32 F4

Das Diagramm zeigt, dass das entwickelte optische System prinzipiell funktioniert und zuverlässig die Bewegungsrichtung erkennt. Es wird aber auch sofort ersichtlich, dass der gemessene optische Fluss nicht immer gleich groß ist, obwohl exakt die selbe Strecke von 20 cm zurückgelegt wird. Eine genaue Betrachtung der Zeitachse lässt erkennen, dass die Messung des optischen Flusses von der Bewegungsgeschwindigkeit des Systems abhängt. Je kürzer der Zeitraum ist, in dem die Stecke von 20 cm zurückgelegt wird, desto geringer ist die durch den optischen Fluss bestimmte Positionsänderung. Demnach ist bei gleicher Strecke der Betrag des optischen Fluss

indirekt proportional zur Bewegungsgeschwindigkeit des Systems. Die Ursache hiefür liegt in der Bildrate von 30fps. Bei schnellen Bewegungen sind die Verschiebungen der Helligkeitsmuster zwischen zwei aufeindander folgenden Bildern größer, da eine größere Strecke zwischen den beiden Aufnahmezeitpunkten zurückgelegt wird. Da der verwendete Algorithmus aber von Verschiebungen, die kleiner gleich einem Pixel sind, ausgeht, liefert er keine verlässlichen Ergebnisse mehr. Um diesem Problem entgegen zu wirken muss entweder eine höhere Bildrate erreicht oder der Algorithmus auf größere Verschiebungen erweitert werden.

6.1.3 Auswertung ADNS 3080 und AVR UC3A

Der zweite Versuch mit dem optischen System, bestehend aus ADNS 3080 und AVR UC3A, wird analog zur Versuchsbeschreibung aus Abschnitt 6.1.1 durchgeführt. Der bei der Versuchsdurchführung bestimmte Positionsverlauf ist in Abbildung 6.3 zu sehen.

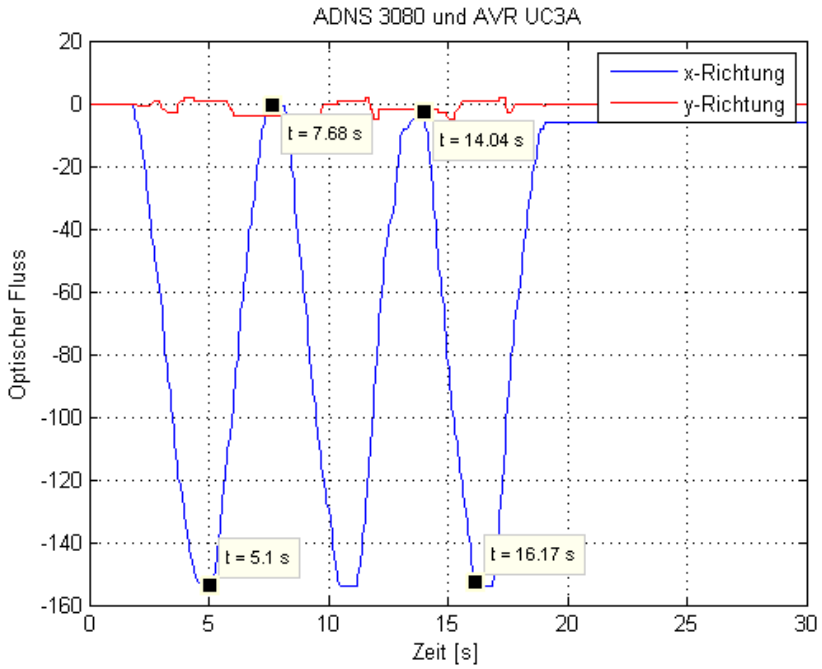


Abbildung 6.3: Positionsbestimmung des optischen Systems aus ADNS 3080 und AVR UC3A

Das Diagramm aus Abbildung 6.3 zeigt, dass der mit Hilfe des optischen Flusssensors bestimmte Fluss sowohl in der Richtung als auch im Betrag bei unterschiedlichen Geschwindigkeiten übereinstimmt. Aufgrund der sehr hohen Bildrate funktioniert der optische Flusssensor auch bei schnellen Bewegungen sehr gut.

6.1.4 Vergleich der Systeme

Ein Vergleich der Evaluierungsergebnisse beider Versuche zeigt, dass mit der aktuellen Implementierung die Messung des optischen Flusses mit beiden Systemen möglich ist. Anders als das optische System aus OV 7670 und STM32 F4 liefert der optische Flusssensor ADNS 3080 auch sehr exakte Werte bei hohen Bewegungsgeschwindigkeiten. Die sehr hohen Bildraten durch die extrem spezialisierte Hardware des optischen Flusssensors ermöglicht dieses gute Ergebnis trotz der geringeren Auflösung. Obowhl das optische System aus OV 7670 und STM32 F4 ein deutlich flexibleres System ist, wird auf Grund seiner hohen Genauigkeit im Weiteren der optische Flusssensor ADNS 3080 verwendet.

6.2 Evaluierung des fertigen Systems

Im zweiten Teil dieses Kapitels wird, wie bereits zu Beginn erwähnt, das Gesamtsystem evaluiert. Dabei wird zunächst untersucht, wie genau der Quadrocopter mit Hilfe des optischen Flusses eine Position halten kann. Anschließend wird evaluiert, wie genau einfache und komplexere Positionsanflüge durchgeführt werden können.

6.2.1 Statisches Regelverhalten

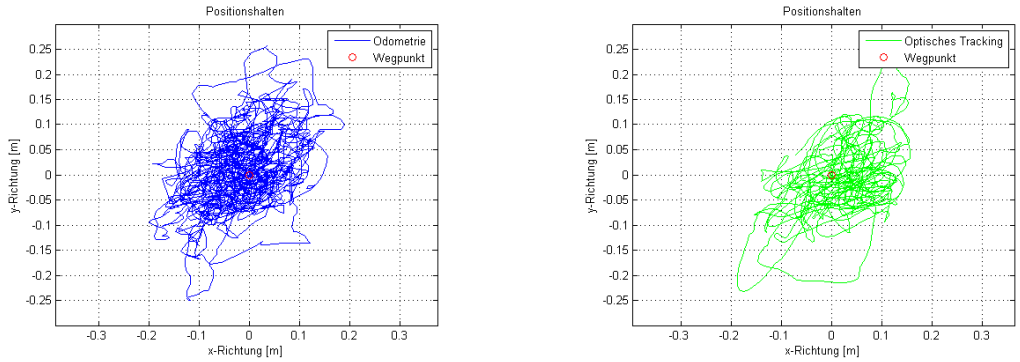
In diesem Versuch wird ermittelt, wie exakt eine Position gehalten werden kann.

6.2.1.1 Versuchsbeschreibung

Zu Beginn des Versuchs befindet sich der Quadrocopter auf einer markierten Stelle am Boden. Der Quadrocopter wird gestartet und in einem Meter Höhe manuell über die markierte Stelle geflogen. Sobald der Quadrocopter diese Stelle erreicht, wird der Positionsregler aktiviert. Über einen Zeitraum von sechs Minuten werden anschließend die Trajektorie der Positionsdaten des optischen Flusses (Odometrie) aufgezeichnet. Um die aufgezeichneten Positionsdaten zu verifizieren wird ein externes optisches Präzisionstracking Systems der Firma WorldViz verwendet[WorldViz, 2012]. Der Ursprung des Trackingkoordiantensystems befindet sich an der zuvor markierten Stelle. Der Ursprung des Odometriekoordiantensystems ist an der Stelle, an der der Positionsregler des Quadrocopters aktiviert wird. Um einen einfachen Vergleich zwischen den Trackingdaten und der Odometrie des Quadrocopters zu ermöglichen, ist es daher wichtig, den Positionsregler exakt über der markierten Stelle zu aktivieren.

6.2.1.2 Auswertung

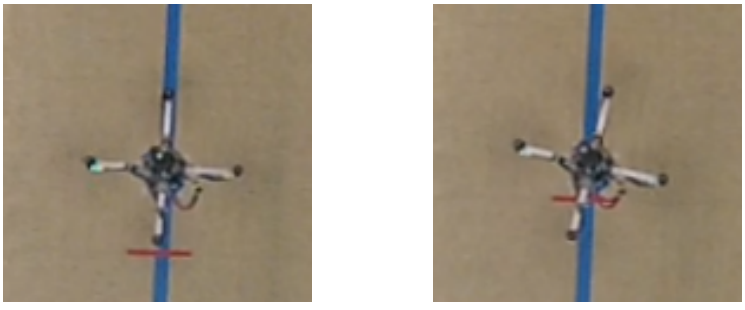
Abbildung 6.4 zeigt die Messergebnisse dieses Versuchs.



(a) Odometrie (b) Optisches Tracking

Abbildung 6.4: Vergleich von Odometrie und optischem Tracking im statischen Regelverhalten

Ein Vergleich beider Diagramme zeigt, dass die gemessenen Trajektorien im Wesentlichen gut übereinstimmen. Bei beiden Messungen liegt der Großteil der Messpunkte dicht um den Nullpunkt und es lassen sich nur wenige Ausreißer erkennen. Bei genauerer Betrachtung der Ausreißer lässt sich erkennen, dass beide Trajektorien nicht exakt übereinstimmen. Hierfür kommen mehrere Ursachen in Frage. So können Schwankungen in der Höhe den Skalierungsfaktor zur Umrechnung der Odometrierohdaten in Meter beeinflussen(vgl. Abschnitt 5.4). Darüberhinaus können Kippbewegungen des Quadrocopters die Messwerte des optischen Flusses verfälschen. Am wahrscheinlichsten für den Unterschied der beiden Trajektorien aber ist die fehlende Gierkompensation des optischen Systems. Während des Fluges nimmt der Sensordrift der IMU stetig zu. Da der optische Flusssensor jedoch keine Rotationen messen kann, wird die Rotation des Quadrocopters nicht durch das optische System erkannt. Da das optische Tracking diese Rotationen jedoch erkennt, kommt es zu einer Verdrehung der beiden gemessenen Trajektorien. Abbildung 6.5 zeigt die Position zu Beginn der Messung und ca. sechs Minuten später.



(a) $t = 0s$ (b) $t = 365s$

Abbildung 6.5: Unterschiedliche Rotationen um die Gierachse im statischen Regelverhalten

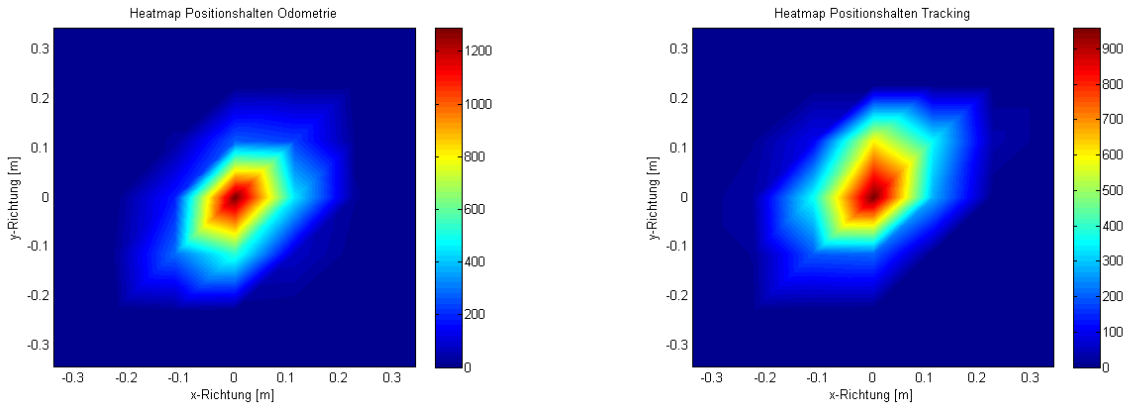
Tabelle 6.1 zeigt eine genaue Analyse der Verteilung der Messwerte. 68,3% der Messwerte besitzen einen Abstand von kleiner gleich 10cm zum Nullpunkt. 95,4% der Odometriemesspunkte und des optischen Trackings liegen in einem Kreis mit einem Radius von 16cm um den Nullpunkt. 99,7% der Odometriedaten befinden sich in einem Kreis mit einem 24cm Radius. Die selbe Punkteanzahl wird beim optischen Tracking breits in einem Kreis mit einem 22cm Radius erreicht.

Tabelle 6.1: Verteilung der Messpunkte im statischen Regelverhalten

Prozentualer Anteil der Messpunkte	68,3%	95,4%	99,7%
Abstand zum Nullpunkt optisches Tracking [m]	0,10	0,16	0,22
Abstand zum Nullpunkt Odometrie [m]	0,10	0,16	0,24

Die Verteilung der Messpunkte des optischen Trackings stimmt also annährend mit der durch die Odometriedaten ermittelten Verteilung überein.

Abbildung 6.6(a) visualisiert die interpolierte Punktdichte der Odometriedaten für das statische Regelverhalten. Abbildung 6.6(b) zeigt die interpolierte Punktdichte des optischen Trackings. Ein optischer Vergleich der Punkteverteilung wird dadurch möglich. Zur Erstellung dieses Diagramms wird ein zweidimensionales Histogramm aus den Messdaten berechnet und farblich dargestellt. Das Histogramm wird anschließend mit Matlab interpoliert. Vergleicht man beide Diagramme, lässt sich eine leichte Verschiebung der Trackingdaten in y-Richtung erkennen. In beiden Diagrammen ist die höchste Punktdichte im Nullpunkt erreicht und nimmt nach außen hin ab.



(a) Odometrie (b) Optisches Tracking

Abbildung 6.6: Interpolierte Punktedichte im statischen Regelverhalten

6.2.2 Dynamisches Regelverhalten - Einfacher Positionsanflug

Bei diesem Versuch wird das Regelverhalten bei einem Positionsanflug untersucht. Der Quadrocopter startet an einer Position und bekommt über die Kontrollsoftware einen zweiten Wegpunkt gesendet, den er anschließend anfliegt.

6.2.2.1 Versuchsbeschreibung

Der Quadrocopter startet, analog zu dem Versuch zuvor, auf einer markierten Stelle am Boden. Die markierte Stelle dient wieder als Referenzpunkt für das optische Tracking System. Nach dem Start begibt sich der Quadrocopter in eine Höhe von einem Meter und wird manuell über die markierte Position gesteuert. Der Regler wird aktiviert und, sobald die Position annähernd stabil gehalten wird, bekommt der Quadrocopter einen zweiten Wegpunkt gesendet. Dieser wird daraufhin angeflogen und für eine kurze Zeit gehalten. Neben den Trackingdaten werden erneut die Odometriedaten des Quadrocopters aufgezeichnet.

6.2.2.2 Auswertung

Abbildung 6.7 zeigt das Regelverhalten in x-Richtung. Der Quadrocopter startet an der Position $(0, 0)$ und fliegt auf die Position $(2, 0)$. Zur besseren Auswertung wird der Offset zwischen dem optischen Tracking und den Odometriedaten ausgeglichen, indem der Anfangspunkt des optischen Trackings mit dem Nullpunkt der Odometriedaten gleichgesetzt wird.

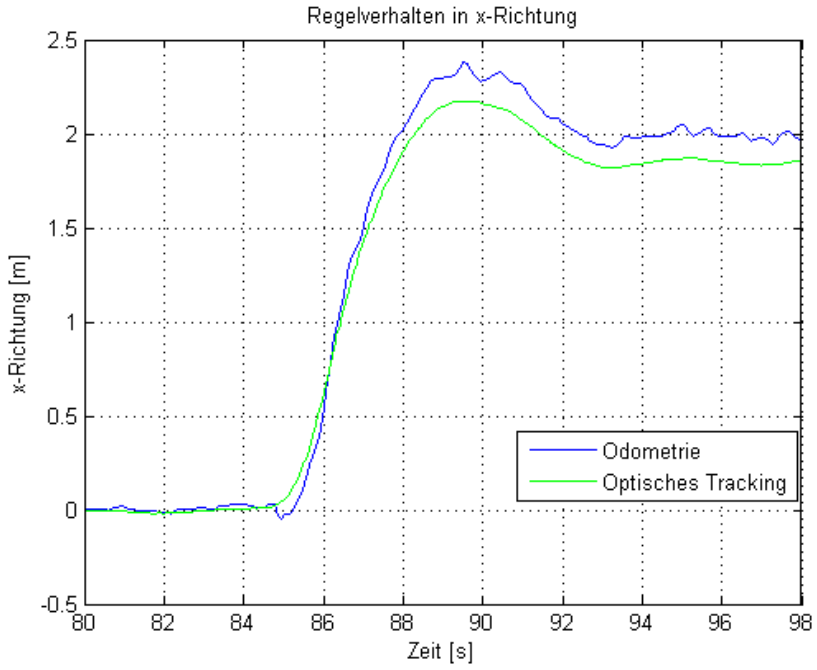


Abbildung 6.7: Dynamisches Regelverhalten in x-Richtung

Aus der Abbildung 6.7 geht hervor, dass der Quadrocopter innerhalb von ca. 2 Sekunden 90% des Sollwerts erreicht hat. Nach weiteren 5 Sekunden hält er, nach kurzem Überschwingen den Sollwert stabil. Die Überschwungweite variert je nachdem, welche Messkurve man betrachtet. Die tatsächliche Überschwingweite liegt zwischen 17 und 35 cm, kann allerdings nicht exakt bestimmt werden. Auf den ersten Blick scheint es ein Skalierungsproblem zwischen Odometrie- und optischen Trackingdaten zu geben. Eine genaue Betrachtung des Regelverhaltens in y-Richtung hilft hier weiter (vgl. Abbildung 6.8).

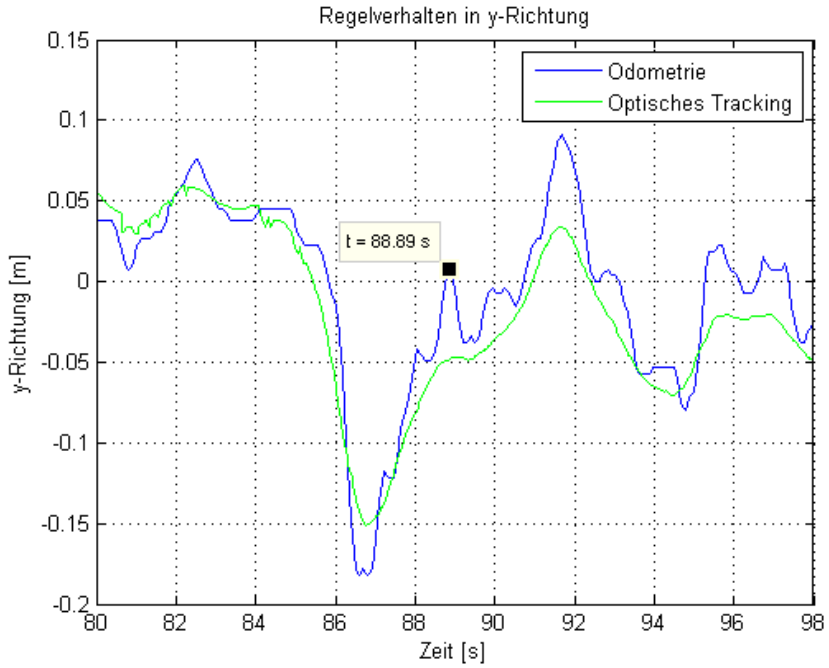


Abbildung 6.8: Dynamisches Regelverhalten in y-Richtung

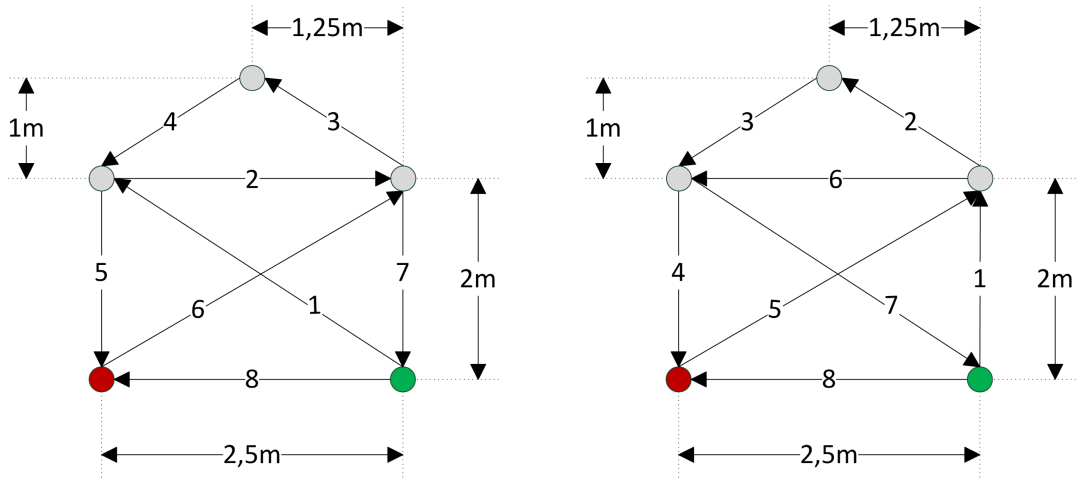
Falls ein Skalierungsproblem vorliegt, gibt es einen Faktor, der sowohl in x- als auch in y-Richtung beide Messreihen annähernd in Deckung bringt. Ein solcher Faktor gleicht allerdings Peaks, wie in Diagramm 6.8 bei $t = 88,89s$ zu sehen, nicht aus. Dieser Fakt und die Tatsache, dass in Abbildung 6.7 beide Graphen in der Einschwingphase beinahe parallel verlaufen, machen ein Skalierungsproblem sehr unwahrscheinlich. Viel wahrscheinlicher ist dagegen eine falsche Messung des optischen Flusses, bedingt durch eine Kippbewegung des Quadrocopters. Diese Vermutung wird dadurch verstärkt, dass der Fehler beim Anfliegen der neuen Position, also in einem Moment in dem der Quadrocopter mit Sicherheit stark geneigt ist, in x-Richtung auftritt.

6.2.3 Dynamisches Regelverhalten - Komplexer Positionsanflug

In diesem Abschnitt wird ein weiterer Versuch zur Evaluierung des dynamischen Regelverhaltens des entwickelten Systems durchgeführt. Anders als im vorherigen Abschnitt, besteht dieser Versuch nicht aus einem einzigen Positionsanflug, sondern aus einer Kombination von mehreren Positionsanflügen in unterschiedlichen Richtungen und Abständen zur Quadrocopterposition.

6.2.3.1 Versuchsbeschreibung

Der Quadrocopter soll in diesem Versuch das „Haus vom Nikolaus“ nachfliegen. Das „Haus vom Nikolaus“ ist ein Zeichenspiel, bei dem ein Haus aus acht Linien gezeichnet werden muss, ohne eine Linie doppelt zu malen. Es werden folgende zwei Zeichenvarianten getestet (vgl. Abblidung 6.9):



(a) Variante 1 (b) Variante 2

Abbildung 6.9: Zeichenvarianten beim „Haus vom Nikolaus“ Flug

Zu Beginn des Versuchs befindet sich der Quadrocopter an der grün markierten Stelle. Sie ist gleichzeitig der Nullpunkt des Koordinatensystems des optischen Trackings. Wie bei den anderen Versuchen zuvor, wird der Quadrocopter manuell gestartet und mit Hilfe des Höhenreglers in einem Meter Höhe über den Startpunkt gesteuert. Der Positionsregler wird aktiviert und die Sollpositionen, analog zu den oben stehenden Varianten, nacheinander an den Quadrocopter geschickt. Sobald der Quadrocopter eine Sollposition erreicht hat, folgt das Senden der nächsten Sollposition, bis er schließlich alle acht Linien geflogen und am roten Endpunkt angekommen ist. Es werden erneut die optischen Trackingdaten, sowie die mit Hilfe des optischen Flusses ermittelten Odometriedaten aufgezeichnet.

6.2.3.2 Auswertung

Beide Flugvarianten werden mehrfach getestet und liefern vergleichbare, reproduzierbare Ergebnisse. Abbildung 6.10 zeigt die vom optischen Tracking und dem optischen System des Quadrocopters aufgezeichneten Trajektorien bei einem Flug nach Variante 1.

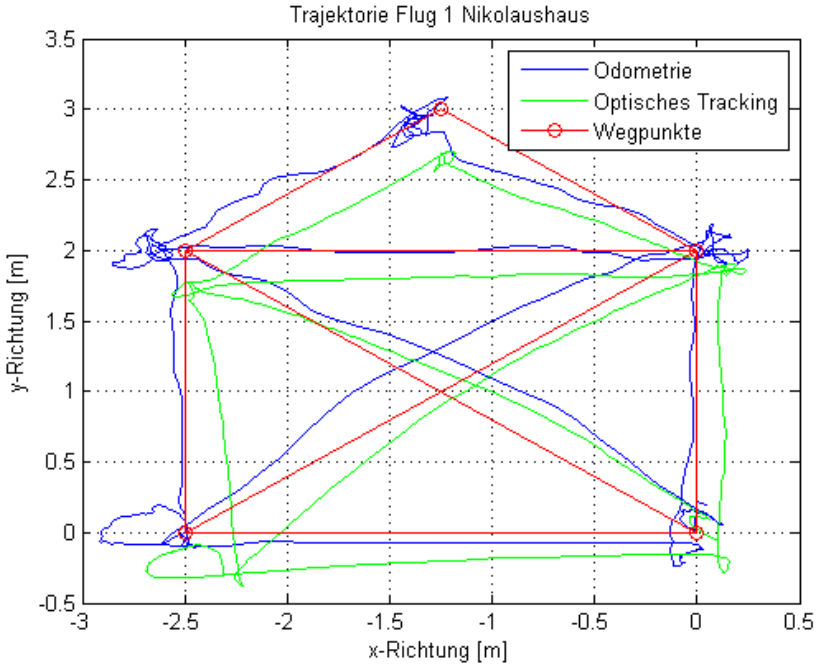
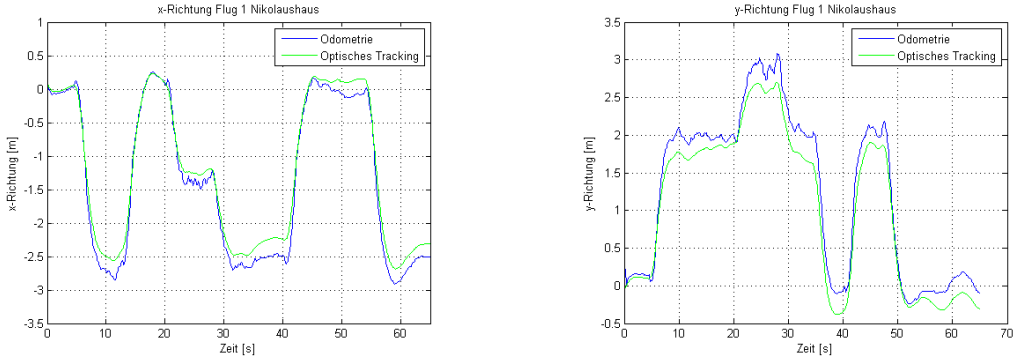


Abbildung 6.10: Trajektorie beim Flug eines Nikolaus Hauses nach Variante 1

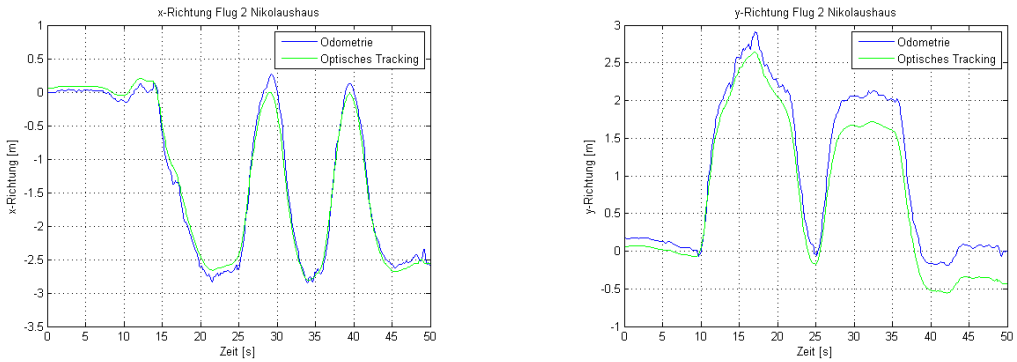
Eine erste Analyse der Abbildung 6.10 zeigt, dass die Trajektorien nicht exakt übereinstimmen. Die durch die Odometrie bestimmte Trajektorie folgt den zuvor festgesetzten Wegpunkten und fliegt diese mit leichtem Überschwingen an. Die Trajektorie des optischen Trackings ist leicht nach unten versetzt und etwas nach links verdreht. Der Offset nach unten hängt mit den ungenauen Synchronisation der beiden Koordinatensysteme zusammen. Die Verdrehung kann durch eine Schrägstellung des Quadrocopters zu Beginn der Messung entstehen. Darüber hinaus erkennt man, dass die Trajektorie beim Anfliegen einer neuen Sollpositionen im allgemeinen sehr geradlinig verläuft. Eine Ausnahme ist hierbei jedoch der Anflug der Dachspitze des Hauses. Da sich in der Versuchsumgebung oberhalb der Spitze gestapelte Bodenmatten befinden, kommt es an dieser Stelle zu starken Luftverwirbelungen, die zu einer Instabilität des Quadrocopters führen. Trotz der erschwerten Bedingungen kann der Quadrocopter jedoch die gewünschte Position anfliegen und halten. Um weitere Erkenntnisse aus diesem Versuch zu gewinnen, muss der zeitliche Verlauf entlang der beiden Achsen einzeln betrachtet werden. Abbildung 6.11 stellt den zeitlichen Verlauf der Positionen in x- und y-Richtung getrennt dar.



(a) x-Richtung (b) y-Richtung

Abbildung 6.11: Zeitlicher Positionsverlauf beim Flug eines Nikolaus Hauses nach Variante 1

In y-Richtung (vgl. Abbildung 6.11(b)) ist es nur einen Offset zwischen dem optischen Tracking und den Odometriedaten zu geben. In x-Richtung (vgl. Abbildung 6.11(a)) erkennt man jedoch ein ähnliches Verhalten wie beim einfachen Positionsanflug. Der Abstand zwischen den beiden Trajektorien variiert stark. Zum Zeitpunkt $t \approx 45s$ ist dieser Abstand fast nicht zu erkennen. Nach der nächsten Positionsänderung ist die Differenz der beiden Kurven bei $t \approx 50s$ dagegen schon sichtbar und wächst dann mit der nächsten Positionsänderung erneut (vgl. $t \approx 60$) an. Dieses Phänomen lässt sich erneut nur durch einen falschen Skalierungsfaktor oder eine Fehlmessung erklären. Die Analyse weiterer Messungen zeigt, dass eine Fehlmessung wahrscheinlich ist. Abbildung 6.12 zeigt den zeitlichen Positionsverlauf in x- und y-Richtung bei einem weiteren Flug nach Variante 2. In x-Richtung stimmen beide abgebildeten Trajektorien bis auf kleine Abweichungen überein. In y-Richtung ist dagegen erneut ein plötzlich auftretender Abstand zwischen beiden Kurven zu erkennen. Da beide Kurven in y-Richtung vor dem ersten markierten Zeitpunkt $t = 25s$ bis auf einen kleinen Offset in etwa gleich sind und beim zweiten Zeitpunkt $t = 28.5s$ bereits eine deutliche Differenz zwischen beiden Trajektorien zu erkennen ist, muss ein Messfehler während der Positionsänderung auftreten.



(a) x-Richtung (b) y-Richtung

Abbildung 6.12: Zeitlicher Positionsverlauf beim Flug eines Nikolaus Hauses nach Variante 2

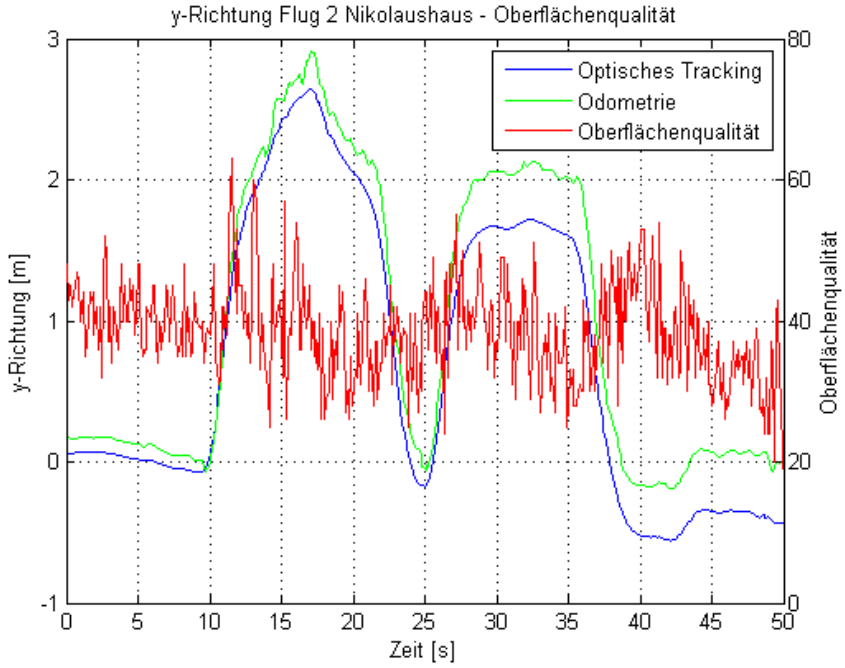


Abbildung 6.13: Zeitlicher Verlauf von Oberflächenqualität und Position in y-Richtung

Ein Vergleich der Messwerte mit der vom optischen Sensor ermittelten Oberflächenqualität schließt Messfehler durch schlechte Lichtverhältnisse aus (vgl. Abbildung 6.13), da sich keine gravierenden Verschlechterungen in der Oberflächenqualität erkennen lassen. Eine Verfälschung der Messwerte durch eine Kippbewegung des Quadrocopters ist deshalb auch hier sehr wahrscheinlich. Andere Ursachen können allerdings nicht vollständig ausgeschlossen werden. Veränderungen der Oberflächentextur des Bodens oder helle Lichtreflexion im Boden können ebenfalls Ursachen für eine Fehlmessung sein.

7 Diskussion und Ausblick

Von den verschiedenen, erprobten optischen Systemen eignen sich nur zwei zur Bestimmung des optischen Flusses. Das selbstentwickelte, optische System aus dem OV7670 CMOS Sensor und dem STM32 F4 Discovery Board erlaubt eine gute Messung des optischen Flusses bei langsamen, gleichgroßen Bewegungsgeschwindigkeiten. Sobald man das System allerdings unterschiedlich schnell bewegt, werden falsche Werte für den optsichen Fluss berechnet. Da der verwendete Algorithmus von maximal einem Pixel als Verschiebung zwischen zwei aufeinander folgenden Bildern ausgeht, reicht die erzielte Bildrate von 30fps bei schnelleren Bewegungen nicht aus. Durch eine entsprechende Modifikation des Algorithmus könnte dieses Problem jedoch kompensiert werden. Das optische System aus dem optischen Flusssensor ADNS 3080 und dem AVR UC3A Eval Board zur Steuerung des Quadrocopters eignet sich, dank der sehr hohen Bildrate von bis zu 6400fps, dagegen sehr gut, die Translationen auch bei sehr schnellen Bewegungen zu bestimmen. Da das zweite optische System gegenüber dem ersten sehr viel kompakter und robuster bei schnellen Bewegungen ist, wird es in das bestehende System integriert.

Sowohl das statische als auch das dynamische Regelverhalten des entwickelten Gesamtsystems funktionieren gut. Positionen können trotz Systemdrift für mehrere Minuten stabil gehalten und darüberhinaus auch zuverlässig angeflogen werden. Das entwickelte System ist stabil gegenüber äußeren Einwirkungen, wie selbsterzeugte Luftverwirbelungen oder kleinere Schubsbewegungen. Allerdings treten Fehler bei der Bestimmung des optischen Flusses auf, wenn Positionen angeflogen werden. Ursache hierfür ist vermutlich die Kippbewegung, die notwendig ist, um die Position des Quadrocopters zu ändern. Es ist daher sinnvoll, eine genaue Evaluierung des Systemverhaltens im gekippten Flug anzustellen. Das Problem selbst kann auf zwei unterschiedliche Arten gelöst werden. Zum einen kann eine kardanische Aufhängung für das optische System verwendet werden. Eine solche Aufhängung richtet die Kamera stets parallel zum Boden aus. Eine Schrägstellung des Quadrocopters hat damit keinen Einfluss auf die Messung des optischen Flusses. Die ebenfalls an der Unterseite des Quadrocopters angebrachten Sensoren zur Höhenmessung würden von dieser Lösung auch profitieren. Zum anderen kann das Problem durch eine intelligente Sensorfusion mit der IMU des Quadrocopters gelöst werden. Indem die aktuellen Roll- und Pitchwinkel des Quadrocopters bei der Berechnung des optischen Flusses berücksichtigt werden, könnten diese Messfehler vermieden werden.

Während der Systementwicklung und -evaluierung haben sich noch zwei weitere Nachteile bei der Verwendung des optischen Flusssensors ADNS 3080 herausgestellt. Da der Quadrocopter später im Innenbereich zum Einsatz kommen soll, muss davon ausgegangen werden, dass nicht immer ausreichend gute Lichtverhältnisse vorhanden sind. Um zuverlässige Werte für den optischen Fluss zu bestimmen, ist der optische Flusssensor auf sehr gute Lichtverhältnisse angewiesen. Damit der Sensor in einer unbekannten Umgebung bei wenig Licht auch arbeiten kann, sollte daher eine eigene Lichtquelle zusätzlich an der Unterseite des Quadrocopters befestigt werden. Super Helle, fokusierte LEDs oder der Kopf einer Taschenlampe haben in kleinen Versuchen gegen Ende dieser Arbeit bereits gute Ergebnisse erzielt. Eine weitere Verbesserung des Systems kann erzielt werden, indem durch das optische System weitere Bewegungen gemessen werden. Da sich das optische System an der Unterseite des Quadrocopters befindet, könnte es auch zur Messung von Rotationen um die Gierachse und zur Abstandsmessung entlang dieser Achse verwendet werden. Durch die Messung der auftretenden Rotationen kann das Driften um die Gierachse kompensiert werden. Mit Hilfe einer weiteren Messung des optischen Flusses entlang der Gierachse können zusätzlich Zoombewegungen zum Boden hin oder vom Boden weg erkannt werden. Diese zusätzlichen Informationen über die Höhenänderung des Quadrocopters könnten für eine Start- und Landeregelung verwendet werden.

8 Inhalt der CD

1. **Bachelorarbeit**: Diese Arbeit im .pdf Format
2. **Quellcode**: Der Quellcode der Systemerweiterung
   a) Kontrollsoftware
   b) Optische Systeme
   c) Gesamtsystem
3. **Kalibrierung**: Messreihe zur Kalibrierung des ADNS 3080 Sensors
4. **Evaluierung**: Messreihen zur Evaluierung mit Quellcode für Matlab
   a) Optische Systeme
   b) Statisches Regelverhalten
   c) Dynamisches Regelverhalten
5. **Kamerabilder**: Kamerabilder der optischen Systeme
   a) Tam2
   b) OV7670
   c) ADNS 3080

9 Literaturverzeichnis

[Abutaleb et al. 2010] ABUTALEB, M. M. ; HAMDY, A. ; ABUELWAFA, M. E. ; SAAD, E. M.: A Reliable FPGA-based Real-time Optical-flow Estimation. In: *International Journal of Electrical and Electronics Engineering* (2010)

[AQopterI8 2012] AQOPTERI8: *Autonomous Quadrotor for Indoor Exploration*. Webseite. 2012. – URL `http://www8.informatik.uni-wuerzburg.de/wissenschaftforschung/aqopteri8/projektskizze/`

[AvagoTechnologies 2010] AVAGOTECHNOLOGIES: *Optical Mice and How They Work*. Produktinformation. 2010. – URL `http://www.avagotech.de/docs/AV02-1649EN`

[AXISCommunications 2010] AXISCOMMUNICATIONS: *CCD and CMOS technology*. Website. 2010. – URL `http://www.axis.com/files/whitepaper/wp_ccd_cmos_40722_en_1010_lo.pdf`

[Barron et al. 1994] BARRON, J.L. ; FLEET, D.J. ; BEAUCHEMIN, S.S.: Performance of optical flow techniques. In: *International Journal of Computer Vision* (1994)

[Beauchemin und Barron 1995] BEAUCHEMIN, S.S. ; BARRON, J.L.: The Computation of Optical Flow. In: *ACM Computing Surveys (CSUR)* (1995)

[Bermudez und Fearing 2009] BERMUDEZ, Fernando G. ; FEARING, Ronald: Optical Flow on a Flapping Wing Robot. In: *IROS'09 Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems* (2009)

[Ding et al. 2009] DING, Weidong ; WANG, Jinling ; HAN, Songlai ; ALMAGBILE, Ali ; GARRATT, Matthew A. ; LAMBERT, Andrew ; WANG, Jack J.: Adding Optical Flow into the GPS/INS Integration for UAV navigation. In: *International Global Navigation Satellite Systems Society Symposium* (2009)

[Ebers 2005] EBERS, Olga: *Bildverarbeitende Objekterkennungssysteme: Analyse der mittleren Ebene*, Technische Universität Berlin, Diplomarbeit, 2005

[Frey et al. 2010] FREY, Daniel ; ULRICH, Markus ; HINZ, Stefan: Evaluierung effizienter Methoden zur Berechnung des optischen Flusses. In: *Photogrammetrie Fernerkundung Geoinformation* (2010)

[Hafner et al. 2010] HAFNER, Verena V. ; BACHMANN, Ferry ; BERTHOLD, Oswald ; SCHULZ, Michael ; MÜLLER, Mathias: An autonomous flying robot for testing bio-inspired navigation strategies. In: *Joint conference of the 41st International Symposium on Robotics* (2010)

[Horn und Schunck 1980] HORN, Berthold K. ; SCHUNCK, Brian G.: Determining Optical Flow. In: *A.I. Memo No.572, Massachusetts Institute of Technology* (1980)

[Hrabar et al. 2005] HRABAR, Stefan ; SUKHATME, Gaurav S. ; CORKE, Peter ; USHER, Kane ; ROBERTS, Jonathan: Combined Optic-Flow and Stereo-Based Navigation of Urban Canyons for a UAV. In: *Proceedings of IROS* (2005)

[Ishii et al. 2010] ISHII, Idaku ; TANIGUCHIA, Taku ; YAMAMOTOA, Kenkichi ; TAKAKIA, Takeshi: 1000 fps Real-Time Optical Flow Detection System. In: *Proc. of SPIE-IS&T Electronics Imaging 2010 Meeting* (2010)

[Jähne 2005] JÄHNE, Bernd: *Digitale Bildverarbeitung*. 6. Auflage. Springer-Verlag Berlin Heidelberg, 2005

[Jiang 2005] JIANG, Xiaoyi: *Kapitel 10 - Bewegungsanalyse*. Vorlesung. 2005. – URL `http://cvpr.uni-muenster.de/teaching/ws05/bildanalyseWS05/`

[Lucas und Kanade 1981] LUCAS, Bruce D. ; KANADE, Takeo: An Iterative Image Registration Technique with an Application to Stereo Vision. In: *DARPA Proceedings of Image Understanding Workshop* (1981)

[Ohta 2010] OHTA, Jun: *Smart CMOS Image Sensor and Applications*. CRC Press, 2010

[Srinivasan 1994] SRINIVASAN, M.V.: An image-interpolation technique for the computation of optic flow and egomotion. In: *Biological Cybernetics* (1994)

[Steinmüller 2008] STEINMÜLLER, Johannes: *Bildanalyse*. Springer-Verlag Berlin Heidelberg, 2008

[Wei et al. 2007] WEI, Zhaoyi ; LEE, Dah-Jye ; NELSON, Brent E.: FPGA-based Real-time Optical Flow Algorithm Design and Implementation. In: *Journal of Multimedia* (2007)

[Westfeld 2004] WESTFELD, Patrick: *Monokulare menschliche Bewegungsanalyse in Bildsequenzen*, Technische Universität Dresden, Diplomarbeit, 2004

[Wikipedia 2012] WIKIPEDIA: *Wikipedia - CCD-Sensor*. Website. 2012. – URL `http://de.wikipedia.org/wiki/CCD-Sensor`

10 Quellenverzeichnis

[ComputerVision 2002] : *Computer Vision - Bewegungsdetektion*. Vorlesung. 2002. – URL http://www.ra.cs.uni-tuebingen.de/lehre/uebungen/ws02/ComputerVision/kapitel3.pdf. – Zugriffsdatum: 15.08.2012

[4DSystems 2011] 4DSYSTEMS: *uCam - Serial JPEG Camera Modul Data Sheet*. Datenblatt. 2011. – URL http://www.4dsystems.com.au/downloads/microCAM/Docs/uCAM-DS-rev7.pdf. – Zugriffsdatum: 07.09.2012

[Arducopter 2012] ARDUCOPTER: *Optical Flow Sensor*. Webseite. 2012. – URL http://code.google.com/p/arducopter/wiki/AC2_OptFlow. – Zugriffsdatum: 07.09.2012

[Centeye 2012] CENTEYE: *Custom Vision Chips for Roboticists*. Website. 2012. – URL http://centeye.com/. – Zugriffsdatum: 26.08.2012

[CMUCam 2012] CMUCAM: *CMUCam*. Website. 2012. – URL http://www.cmucam.org/. – Zugriffsdatum: 16.08.2012

[Harvard 2012] HARVARD: *Robobees - A convergence of body, brain, and colony*. Website. 2012. – URL http://robobees.seas.harvard.edu/. – Zugriffsdatum: 26.08.2012

[Lange und Sünderhauf 2012] LANGE, Sven ; SÜNDERHAUF, Niko: *SkyeCopter*. Webseite. 2012. – URL http://www.tu-chemnitz.de/etit/proaut/forschung/quadrocopter.html. – Zugriffsdatum: 26.08.2012

[Omnivision 2006] OMNIVISION: *OV7670/OV7171 CMOS VGA (640x480) Camera Chip - Preliminary Datasheet*. Datasheet. 2006

[px4Autopilot 2012] PX4AUTOPILOT: *PX4FLOW Smart Camera Projekt*. Website. 2012. – URL https://pixhawk.ethz.ch/px4/modules/px4flow. – Zugriffsdatum: 15.08.2012

[SpiegelOnline 2008] SPIEGELONLINE: *Drohnen im Hörsaal - Studenten entwickeln autonomen Flugroboter*. Webvideo. 2008. – URL http://www.spiegel.de/video/drohnen-im-hoersaal-studenten-entwickeln-autonomen-flugroboter-video-37124.html. – Zugriffsdatum: 26.08.2012

[SpiegelOnline 2012] SPIEGELONLINE: *Der Dohneneinsatz für die Polizei ist umstritten*. Webvideo. 2012. – URL http://www.spiegel.de/video/der-drohneneinsatz-fuer-die-polizei-ist-umstritten-video-1220453.html. – Zugriffsdatum: 10.09.2012

[WorldViz 2012] WORLDVIZ: *Precision Position Tracker*. Webseite. 2012. – URL http://www.worldviz.com/products/ppt/index.html. – Zugriffsdatum: 13.09.2012

11 Abkürzungsverzeichnis

CCD	**C**harged **C**oupled **D**evice
CIF	*C*ommon **I**ntermediate **F**ormat
CMOS	**C**omplementary **M**etal **O**xide **S**emiconductor
CPU	**C**entral **P**rocessing **U**nit
DCMI	**D**igital **Cam**era **I**nterface
DMA	**D**irect **M**emory **A**ccess
FPGA	**F**ield **P**rogrammable **G**ate **A**rray
FPS	**F**rames **p**er **S**econd
GPS	**G**lobal **P**ositioning **S**ystem
GUI	**G**raphical **U**ser **I**nterface
I2C	**I**nter-**I**ntegrated **C**ircuit
IDE	**I**ntegrated **D**evelopment **E**nvironment
IMU	**I**nertial **M**easurement **U**nit
PID	**P**roportional-**I**ntegral-**D**ifferential
PLL	**P**hase-**L**ocked **L**oop
QCIF	**Q**uarter **C**ommon **I**ntermediate **F**ormat
QQVGA	**Q**uarter **Q**uarter **V**ideo **G**raphics **A**rray
QVGA	**Q**uarter **V**ideo **G**raphics **A**rray
RAM	**R**andom **A**ccess **M**emory
SPI	**S**erial **P**eripheral **I**nterface
TTL	**T**ransistor-**T**ransistor-**L**ogik
UART	**U**niversal **A**synchronous **R**eceiver **T**ransmitter
UAV	**U**nmanned **A**erial **V**ehicle
VGA	**V**ideo **G**raphics **A**rray

12 Abbildungsverzeichnis

1.1 Quadrocopter des AQopterI8 Projektes 1

2.1 Bewegungsfelder von elementaren Bewegungen 3
2.2 Einfache Darstellung des optischen Flusses 4
2.3 Beziehung zwischen Bewegungsfeld und optischen Fluss 5
2.4 Veranschaulichung des Blendenproblems 6
2.5 Grafische Darstellung der Kontinuitätsgleichung des optischen Flusses 7
2.6 Betrachteter Bildbereich: Centerpixel (gelb) und benachbarte Pixel (rot) 13

3.1 Übersicht über die Systeme zur Positionsbestimmung eines Quadrocopters 15
3.2 Schematische Darstellung bei Verwendung eines ADNS 3080 Sensors 17
3.3 Schematische Darstellung bei Verwendung eines Bildsensors und Prozessors . . . 18
3.4 Aktives Bildelement eines CMOS Sensors 19

4.1 Gesamtkonzept 23
4.2 Phasen während der Systementwicklung 25
4.3 Aufgabenverteilung bei der Positionsregelung 26
4.4 Positionsregelkreis 27

5.1 Benutzeroberfläche der Kontrollsoftware bei der Trajektorieverfolgung 30
5.2 Benutzeroberfläche der Kontrollsoftware im Videomodus des ADNS 3080 31
5.3 Vision Chip Tam2 und Arduino Mega 2560 32
5.4 Bild des Vision Chips Tam2 bei unterschiedlichen Entfernungen 33
5.5 Mögliche Integration der uCam 34
5.6 Mögliche Integration der CMUCam 4 35
5.7 Anbindung des OV 7670 an STM32 F4 Discovery Board 36
5.8 Horizontales Timingdiagramm der DCMI Schnittstelle 37
5.9 Gesamtes Timingdiagramm der DCMI Schnittstelle 37
5.10 Bilder des OV7670 Sensors in unterschiedlichen Formaten 39
5.11 Mögliche Integration des ADNS 3080 Sensors 40
5.12 Bild des ADNS 3080 Sensors 41
5.13 Sensorik am Boden des Quadrocopters 42

5.14 Vereinfachtes Aktivitätsdiagramm des Positionsreglers 43
5.15 Einfluss der Höhe auf die Messwerte des optischen Sensors 46
5.16 Bild des ADNS 3080 Sensor bei Verwendung unterschiedlicher Objektive 47
5.17 Messreihe zur Kalibrierung des Systems 48

6.1 Versuchsaufbau für den ADNS 3080 Sensor 49
6.2 Positionsbestimmung des optischen Systems aus OV 7670 und STM32 F4 50
6.3 Positionsbestimmung des optischen Systems aus ADNS 3080 und AVR UC3A . 51
6.4 Vergleich von Odometrie und optischem Tracking im statischen Regelverhalten . 53
6.5 Unterschiedliche Rotationen um die Gierachse im statischen Regelverhalten . . . 53
6.6 Interpolierte Punktedichte im statischen Regelverhalten 54
6.7 Dynamisches Regelverhalten in x-Richtung 55
6.8 Dynamisches Regelverhalten in y-Richtung 56
6.9 Zeichenvarianten beim „Haus vom Nikolaus“ Flug 57
6.10 Trajektorie beim Flug eines Nikolaus Hauses nach Variante 1 58
6.11 Zeitlicher Positionsverlauf beim Flug eines Nikolaus Hauses nach Variante 1 . . 59
6.12 Zeitlicher Positionsverlauf beim Flug eines Nikolaus Hauses nach Variante 2 . . 59
6.13 Zeitlicher Verlauf von Oberflächenqualität und Position in y-Richtung 60