



**Informationstechnik für Luft- und Raumfahrt**

Lehrstuhl für Informatik 8

Prof. Dr. Sergio Montenegro

---

# Bachelorarbeit

Positionsbestimmung eines autonomen  
Quadroopters durch Bildverarbeitung

Vorgelegt von

Eric Reinalt

Matr.-Nr.: 1791981

Prüfer:

Prof. Dr. Sergio Montenegro

Betreuer wissenschaftlicher Mitarbeiter:

Dipl.-Ing. Nils Gageik



# 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, 17.03.2014

---

Eric Reinald

# Aufgabenstellung

Die Fortschritte im Bereich Sensorik und Mikrotechnik ermöglichen heutzutage den kostengünstigen Bau kleiner unbemannter Luftfahrzeuge (UAV, unmanned aerial vehicle, Drohne) wie Quadrokopter. 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 Aerospace Information Technology der Uni Würzburg erforscht und erprobt (AQopterI8 Projekt).

Das aktuelle System verwendet einen optischen Flusssensor (OF-Sensor) zur Positionsbestimmung sowie eine auf den Boden gerichtete WebCam zur Objekterkennung und kann damit autonom Objekte suchen. Ein Problem des OF-Sensors sind Drehungen um die z-Achse (Gier-Rotationen). Der Sensor interpretiert diese fälschlicherweise als Positionsänderungen. Somit lässt sich im autonomen Flug nicht gieren, was den Betrieb einschränkt. Insbesondere, wenn das System eine Vorzugsrichtung hat, z.B. eine nach vorne montierte PMD-Kameras für eine hochgenaue Hinderniserkennung verwendet. Dann ist es in bestimmten Situationen erforderlich, durch Gieren diese Kamera auszurichten. Denkbar ist hier der Fall des autonomen Fliegens durch ein Fenster, das sich rechts vom Quadrocopter befindet.

Eine Lösung des Problems wäre es, die Daten des OF-Sensors während des Gierens nicht zu verwenden. Dabei tritt jedoch ein Positionsfehler auf, da der Quadrocopter als fliegendes System natürlich seine Position während des Gierens verändert. Eine Verbesserung kann durch eine optische Korrektur mit Hilfe der bereits mitgeführten WebCam erfolgen. Vor und nach dem Gieren wird ein Bild vom Boden aufgenommen. Die beiden Bilder werden mit Hilfe der bekannten Rotation und Höhe ineinander überführt und die Positionsverschiebung während der Drehung soll auf diese Weise errechnet werden.

Hauptaugenmerk dieser Arbeit ist die Bestimmung der Positionsänderung mit Mitteln der Bildverarbeitung. Als Eingabe dienen die Bilder der Kamera, die gemessene Höhe sowie die Rotation. Es bieten sich Lösungen basierend auf dem Prinzip des optischen Flusses an, jedoch ist das verwendete Prinzip frei wählbar. Es soll eine möglichst optimale Lösung für diesen Anwendungsfall unter Berücksichtigung der Nebenbedingungen Rechenleistung, Zeit, Bewegung und erforderliche Genauigkeit implementiert werden.

Die Implementierung ist in das bestehende System zu integrieren und an diesem zu evaluieren. Zur Arbeit gehört eine ausführliche Dokumentation.

Aufgabenstellung (Stichpunktartig):

- Konzept: Positionsbestimmung aus iterativen, verschobenen/transformierten (4DOF) Bildern
- Bestimmung Position unter Verwendung Orientierung und Höhe
- Bestimmung Rotation (optional: Rotation nicht gegeben, sondern auch gesucht)
- Integration der Software in bestehendes System
- Optimierung auf Flugbetrieb und Erprobung im Flug
- Evaluierung
- Dokumentation (Quellcode & Wiki)



---

# Zusammenfassung

Ziel dieser Arbeit ist die Entwicklung eines Systems zur rotations-invarianten Positionserfassung für den Forschungs-Quadrocopter *AQopterI8* der Universität Würzburg. Das bisherige Positionserfassungssystem ist nicht in der Lage, Rotationen zu erkennen, wodurch die Navigationsmöglichkeiten eingeschränkt sind. Die hier entwickelte Positionserfassung arbeitet mit einer senkrecht auf den Boden gerichteten Kamera. Durch kontinuierliche Bildregistrierung im Spektralbereich werden die affinen Transformationen aufgenommener Kamerabilder berechnet. Dies lässt einen Rückschluss auf die Eigenbewegung des Quadrocopters zu. Das implementierte System ist in der Lage, Translation, Rotation und Höhenänderung des Quadrocopters selbstständig zu berechnen. Die Evaluierung zeigte, dass die Positionsbestimmung im Flug zuverlässig arbeitet und die Anforderungen an die gewünschte Genauigkeit erfüllt.

# Inhaltsverzeichnis

<b>1 Einleitung .....</b>	<b>1</b>
<b>2 Stand der Technik .....</b>	<b>3</b>
2.1 Methoden zur Positionsbestimmung .....	3
2.1.1 Fremdortung .....	3
2.1.2 Eigenortung.....	3
2.2 Optischer Fluss .....	5
2.2.1 Definition des optischen Flusses.....	5
2.2.2 Rückschluss auf die tatsächliche Bewegung .....	6
2.2.3 Methoden zur Berechnung .....	8
2.3 Bilder als Signale .....	9
2.3.1 Interpretation als Signal.....	9
2.3.2 Koordinatensysteme .....	10
2.3.3 System.....	11
2.3.4 Diskrete Filter.....	12
2.3.5 Faltung .....	13
2.3.6 Anwendung von FIR-Filtern .....	14
2.3.7 Affine Transformationen auf Bildern.....	16
2.4 Bilder im Frequenzspektrum.....	16
2.4.1 Herleitung der Fouriertransformation auf diskreten Signalen .....	17
2.4.2 Eigenschaften der Fouriertransformation .....	20
2.4.3 Randeffekte.....	21
2.4.4 Die schnelle Fouriertransformation.....	23
2.5 Bildregistrierung.....	24
2.5.1 Verfahren der Registrierung .....	24
2.5.2 Phasenkorrelation.....	25
<b>3 Konzept.....</b>	<b>28</b>
3.1 Überblick .....	28
3.2 Grundlagen.....	29
3.3 Registrierung vor und nach dem Gieren .....	31
3.4 Registrierung unter Beachtung von Translation, Rotation und Skalierung .....	35
3.5 Kontinuierliche Registrierung .....	39

---

<b>4 Implementierung.....</b>	<b>41</b>
4.1 Anforderungen.....	41
4.2 Positionsbestimmungsalgorithmus.....	41
4.2.1 Vorverarbeitung der Bilder .....	42
4.2.2 Fouriertransformation .....	44
4.2.3 Phasenkorrelation.....	45
4.2.4 Hochpass-Filter .....	46
4.2.5 Log-Polar-Transformation.....	48
4.3 Kontrollsoftware .....	50
4.3.1 Bestehendes System.....	50
4.3.2 Positionsbestimmung mittels Bildregistrierung .....	51
4.4 Kamera .....	56
4.5 Kalibrierung.....	57
4.5.1 Schwellenwert der Phasenkorrelation .....	57
4.5.2 Skalierungsfaktor der Translation.....	58
<b>5 Evaluierung .....</b>	<b>60</b>
5.1 Vergleich der Kameras .....	60
5.2 Framerate und Auflösung .....	61
5.2.1 Trade-Off-Problem.....	61
5.2.2 Statisches Verhalten .....	61
5.2.3 Dynamisches Verhalten .....	63
5.2.4 Positionsbestimmung im Vergleich .....	64
5.2.5 Fazit.....	66
5.3 Rotation und Höhenänderung .....	67
5.4 Lichtverhältnisse .....	68
5.5 Tests im Flug.....	69
5.5.1 Abfliegen einer Wegpunktliste .....	69
5.5.2 Rotation .....	73
<b>6 Diskussion und Ausblick.....</b>	<b>76</b>
<b>Anhang.....</b>	<b>79</b>
<b>Abkürzungsverzeichnis .....</b>	<b>84</b>
<b>Abbildungsverzeichnis .....</b>	<b>85</b>
<b>Literaturverzeichnis .....</b>	<b>87</b>



# 1 Einleitung

In den letzten Jahren fanden unbemannte Flugdrohnen in verschiedenen Bereichen eine starke Verbreitung. Bei den auch Unmanned Aerial Vehicle (UAV) genannten Flugobjekten etablierten sich vor allem Quadrocopter. Quadrocopter sind Drohnen mit vier senkrecht zum Boden wirkenden Rotoren. Durch diese Anordnung erreicht man einen hohen Grad an Steuerbarkeit, aber gleichzeitig ist die Fluglage instabil. Daher benötigen Quadrocopter eine kontinuierliche Regelung, um waagrecht in der Luft zu bleiben.

Dank Fortschritten in der Mikro-Elektronik gibt es heute eine Vielzahl an Anwendungsmöglichkeiten für Quadrocopter. Neben einer Verbreitung im Hobby-Bereich beschäftigen sich immer mehr kommerzielle und wissenschaftliche Projekte mit Quadrocoptern. Drohnen werden unter anderem für Luftaufnahmen, Überwachungsaufgaben und Kartierungen eingesetzt.

Ein weiteres Anwendungsfeld ist der Einsatz bei Katastrophen. Quadrocopter können schnell in unwegsames oder gefährliches Gelände vordringen und einen Lagebericht liefern. Durch den Einsatz der Drohne muss kein Mensch der Gefahr in der zu erkundenden Umgebung ausgesetzt werden.

Um unbekanntes Gelände selbstständig erkunden zu können, muss der Quadrocopter in der Lage sein, autonom zu fliegen. Ein Großteil der heutigen autonomen Drohnen orientiert sich Anhand von GPS-Daten. Diese Drohnen benötigen zur Navigation eine ständige Satellitenverbindung, wodurch eine Verwendung in geschlossenen Gebäuden ausgeschlossen ist. Allerdings ergeben sich gerade in Gebäuden interessante Anwendungsmöglichkeiten für autonome Quadrocopter. Der autonome Flug in Umgebungen ohne GPS-Empfang ist jedoch noch nicht ausreichend erforscht.

Im Rahmen des *AQopter18*-Projekts der Julius-Maximilians-Universität Würzburg wird ein autonomer Indoor-Quadrocopter für Rettungsmissionen entwickelt. Ziel des Projekts ist, dass der Quadrocopter unbekannte Gebäude autonom erkunden und kartografieren kann. So soll er in der Lage sein, in brennenden Gebäuden nach Personen zu suchen und die Standorte der gefundenen Personen an die Bodenstation zu übermitteln. [1]

Ohne akkurate Positionsinformationen ist eine Lokalisierung erkannter Objekte nicht möglich. Daher muss der Quadrocopter stets seine Position bestimmen können, um unbekanntes Gebiet autonom erkunden und kartografieren zu können. Die Genauigkeit und Zuverlässigkeit

der Positionsbestimmung spielt eine entscheidende Rolle. Für den Quadrocopter sind im Flug sechs Freiheitsgrade zu bestimmen: drei für den Ort und drei für die Orientierung im Raum.

Nach aktuellem Entwicklungsstand des Projekts verfügt der Quadrocopter über eine Positionsbestimmung auf Basis des optischen Flusses. Über eine senkrecht auf den Boden gerichteten Kamera wird mit der differentiellen Methode der optische Fluss gemessen und in die Eigenbewegung des Quadrocopters übersetzt. Dieses System kann keine Rotation erkennen. Somit ist die Bewegung des Quadrocopters auf reinen Translations-Flug beschränkt.

In dieser Arbeit wird ein Positionsbestimmungssystem entwickelt, welches sowohl Translation als auch Rotation erkennen kann. Dadurch ist es möglich, den Quadrocopter bei jedem Manöver in Flugrichtung zu drehen, um mit einer nach vorn gerichteten Kamera die Umgebung zu erkunden. Obwohl es für die differentielle Methode des optischen Flusses Algorithmen zur Rotationserkennung gibt, baut das Konzept dieser Arbeit auf einer anderen Methode auf. Dadurch erhält man einen Vergleich verschiedener Systeme in Bezug auf Genauigkeit und Zuverlässigkeit. In dieser Arbeit wird durch Bildregistrierung auf die affine Transformation zweier Bilder geschlossen. Die Parameter der Transformation lassen einen Rückschluss auf die Eigenbewegung des Quadrocopters zu. Diese Positionserfassung arbeitet unabhängig von Rotationen und Höhenänderungen im Flug, wodurch man mehr Freiheit bei der Navigation des Quadrocopters erhält.

## 2 Stand der Technik

### 2.1 Methoden zur Positionsbestimmung

Die Methoden der Positionsbestimmung fasst man in zwei unterschiedliche Kategorien. Ist ein Roboter auf Informationen externer Systeme angewiesen, um seine Position zu erfassen, spricht man von Fremdortung. Erfolgt die Lokalisierung hingegen ausschließlich mit den Informationen eigener Sensoren, kann der Roboter seine Position autonom bestimmen. Dies wird Eigenortung genannt.

#### 2.1.1 Fremdortung

Die verbreitetste Methode unter autonomen (Flug-)Robotern ist die satellitengestützte Ortung (z.B. GPS<sup>1</sup>). Durch Kontakt zu mindestens vier Satelliten, welche ständig ihren exakten Ort und die Uhrzeit senden, lässt sich die eigene Position im Raum mittlerweile weltweit (mit Ausnahme der Pole) auf wenige Zentimeter genau errechnen. In den vorgesehenen Anwendungsszenarien des Quadrocopters ist diese Ortung aber nicht möglich, da in Gebäuden oftmals keine ausreichende Satellitenverbindung besteht und der Quadrocopter für akkurate Ortung ein durchgehendes Signal benötigt. [2]

Beim Navigieren einer Drohne in bekanntem, vorher festgelegtem Gebiet hat sich außerdem das kameragestützte Tracking durchgesetzt. Diese Methode wird vor allem in Gebäuden angewendet und liefert ein hohes Maß an Genauigkeit. Dazu werden die autonomen Roboter mit Markern versehen, welche von den Kameras erfasst werden. Ein Computer errechnet aus den Signalen der Kameras eine dreidimensionale Position und schickt diese an die Roboter. Trotz hoher Präzision besitzt das System nur begrenzte Anwendungsmöglichkeiten, da der Bewegungsradius räumlich auf den Sichtbereich der fest installierten Kameras eingeschränkt ist. Beim *AQopter18* wird das kameragestützte Tracking zur Evaluierung eigener Positionsbestimmungssysteme benutzt.

#### 2.1.2 Eigenortung

Verfügt der Roboter über ein System zur Positionsbestimmung, welches nicht auf externe Signale zurückgreifen kann, spricht man von Eigeneortung. Häufig kann die aktuelle Position nur relativ zur zuletzt bestimmten Position ermittelt werden. Dieses Verfahren wird Koppelnavigation (engl. Dead Reckoning) genannt. Im Gegensatz zur Fremdortung, bei der

---

<sup>1</sup> Global Positioning System

man meist über eine absolute Ortsinformation verfügt, erfolgt die autonome Ortsbestimmung mittels Koppelnavigation also nur relativ zu einem Referenzpunkt (meist Startposition). Außerdem ist bei der Koppelnavigation die Orientierung des Roboters von entscheidender Rolle, da neue Positionsdaten relativ zu den Koordinatenachsen des Robotersystems gewonnen werden. Es gibt verschiedene Ansätze zur Bestimmung der Position, wobei häufig optische Systeme eingesetzt werden. Möglich ist auch die Berechnung aus den Beschleunigungswerten einer IMU<sup>2</sup>. Ausreichende Genauigkeiten können Aufgrund der doppelten Integration dieser Methode aber nur mit einer lasergestützten IMU erzielt werden. Diese sind für den Quadrocopter *AQopter18* zu groß und zu schwer, was eine Integration in das bestehende System ausschließt.

Bei der bildbasierten Koppelnavigation wird die Bewegung aus dem optischen Fluss ermittelt. Entweder wird der optische Fluss direkt ermittelt, oder durch Transformation zweier aufeinanderfolgender Bilder die relative Positionsänderung bestimmt. Vom optischen Fluss wird dann auf die tatsächliche Bewegung geschlossen.

Einen anderen Ansatz verfolgt die SLAM-Methode<sup>3</sup>. Meist wird mit einem Laserscanner oder einem Infrarotsensor ein dreidimensionales Abbild der Umgebung erstellt. Dabei baut der Roboter iterativ eine Karte seiner Umgebung auf. Gleichzeitig lokalisiert er sich, denn nur mit dem Wissen über seine relative Position in der Karte kann diese erweitert werden. Die Lokalisierung erfolgt zum Beispiel über erkannte Objekte in der Karte. Bewegen sich diese Objekte, oder sind keine vorhanden (Flug im Freien ohne Hindernisse), kann keine Positionsbestimmung erfolgen. Statt Lasern kann auch eine Kinect-Kamera<sup>4</sup> verwendet werden. Dabei entsteht ein sogenanntes RGBD-Profil<sup>5</sup> der Umgebung. Die Bewegung der Kamera wird über das Tiefenprofil berechnet, gleichzeitig wird die Karte durch Bildregistrierung erstellt. Diese Methode lässt sich jedoch Aufgrund der benötigten Hardware kaum auf einem Low-Cost-Quadrocopter implementieren. [3] [4]

---

<sup>2</sup> Inertial Measurement Unit

<sup>3</sup> Simultaneous Localization and Mapping

<sup>4</sup> Kamera von Microsoft mit RBG- und Infrarot-Sensor

<sup>5</sup> Rot, Grün, Blau, Tiefe

## 2.2 Optischer Fluss

### 2.2.1 Definition des optischen Flusses

Der optische Fluss ist die Helligkeitsänderung (auch Intensitätsänderung) einer Szene im sichtbaren Spektrum und wird durch ein Vektorfeld beschrieben. Bewegen sich zwei Objekte relativ zueinander, repräsentiert der optische Fluss die scheinbare Bewegung dieser Objekte durch visuelle Beobachtung.

Der optische Fluss wird mit Hilfe einer Kamera ermittelt. Zur Beschreibung des optischen Flusses definiert man für jeden Pixel die zeitabhängige Intensität  $I(x, y, t)$ , wobei  $x$  und  $y$  die Koordinaten im Bild sind. Die zeitabhängige Bewegung der Pixel entspricht den Vektoren des Feldes des optischen Flusses.

Bewegt sich ein Objekt im Zeitintervall  $\Delta t$  um  $\Delta x$  bzw.  $\Delta y$  translatorisch durch die Szene, gilt für die korrespondierenden Pixel des Objekts:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \quad (2.1)$$

Geht man von einer kleinen Verschiebung aus, kann der rechte Teil mithilfe einer Taylor-Reihe entwickelt werden.

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

Daraus folgt mit Formel (2.1):

$$\frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t = 0 \quad (2.3)$$

Nach Division durch  $\Delta t$  erhält man für die Geschwindigkeiten  $u$  und  $v$  der Pixel in  $x$ - bzw.  $y$ -Richtung, welche dem Vektorfeld des optischen Flusses entsprechen, folgende Formel:

$$\frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v = -\frac{\partial I}{\partial t} \quad (2.4)$$

Formel (2.4) wird Optical Flow Constraint Equation (OFCE) genannt und besitzt ohne weitere Einschränkungen keine eindeutige Lösung. Durch das sogenannte Blendenproblem des optischen Flusses kann die zweidimensionale Verschiebung nicht eindeutig bestimmt werden. In Abbildung 2.1 erkennt man, dass nur der  $b$ -Vektor bestimmt werden kann, wenn die blaue Linie an den grauen Pfeilen entlang verschoben wird.

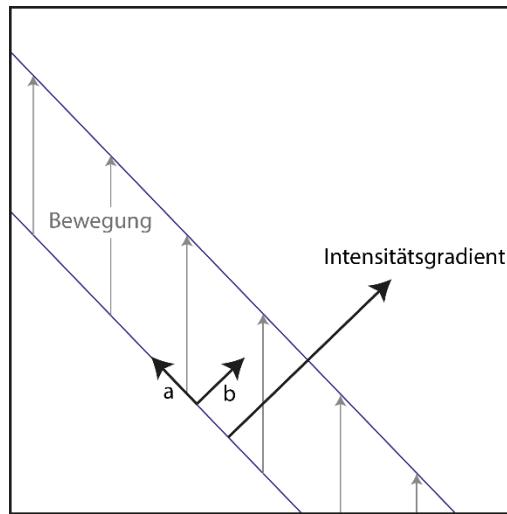


Abbildung 2.1 Blendenproblem des optischen Flusses

Es kann nur die Bewegung parallel zum Helligkeitsgradienten bestimmt werden, da eine Bewegung senkrecht zum Helligkeitsgradienten ( $a$ -Vektor im Bild) keinen optischen Fluss verursacht. Das Vektorfeld des optischen Flusses kann also nur korrekt erstellt werden, wenn in der Szene linear unabhängige (oder keine) Bewegungen auftreten. Dies ist für eine reale Szene nahezu immer der Fall, da z.B. an verschiedenen Kanten eines Objekts das Gradientenfeld niemals parallel verlaufen kann.

## 2.2.2 Rückschluss auf die tatsächliche Bewegung

Der Quadrocopter bestimmt den optischen Fluss mit einer senkrecht auf den Boden gerichteten Kamera. Der Boden wird als  $xy$ -Ebene definiert. Nimmt man die Szene als statisch an (keine beweglichen Objekte am Boden), ergeben sich Intensitätsunterschiede aufeinanderfolgender Bilder aus Translation und Rotation der Kamera. Der gesamte optische Fluss kann daher auf die Bewegung der Kamera (des Quadrocopters) zurückgeführt werden.

Bleibt die Kamera konstant senkrecht zum Boden, liegt das Sichtfeld der Kamera in einer Ebene mit dem Boden. Diese Ausrichtung entspricht dem Idealfall, da das Vektorfeld der Bewegung und das Vektorfeld des optischen Flusses durch die gemeinsame Ebene übereinstimmen. Der optische Fluss lässt sich in diesem Fall direkt in die Bewegung des Quadrocopters übersetzen.

Ohne Änderung der Orientierung des Quadrocopters in Bezug auf die  $xy$ -Ebene kann jede mögliche Bewegung und das daraus resultierende Vektorfeld einer der affinen Transformationen aus Abbildung 2.2 zugeordnet werden. Bewegt sich die Kamera (der Quadrocopter) auf den Boden zu oder von ihm weg, was einer Translation in  $z$ -Richtung gleichkommt, spricht man von Skalierung.

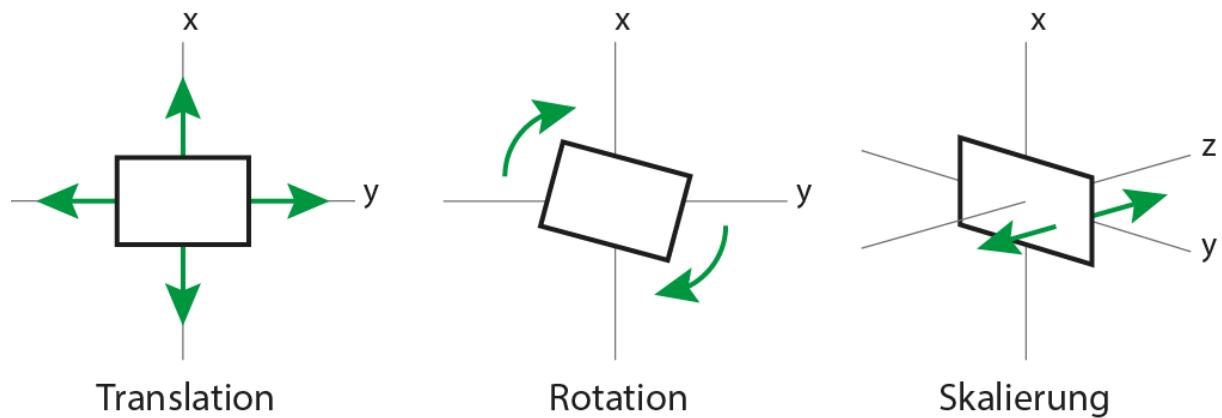


Abbildung 2.2 Affine Transformationen

Die direkte Übersetzung des optischen Flusses in die Bewegung der Kamera ist nur mit den oben genannten Einschränkungen möglich.

Ändert sich der Winkel der Kamera zum Boden, bewegt sich das Sichtfeld der Kamera über den Boden hinweg, ohne dass sich die Kamera bewegt. Die Kamera registriert einen optischen Fluss in Richtung der Bewegung des Sichtfeldes, optischer Fluss und Bewegung der Kamera stimmen nicht überein. Grund dafür sind die nicht übereinstimmenden Ebenen von Sichtfeld und Boden. Nick- und Rollwinkel der Kamera des Quadrocopters sollten daher bei der Positionsbestimmung mittels optischen Flusses berücksichtigt werden, um Änderungen der Orientierung nicht in falsche Bewegungen zu übersetzen.

Die Barber-Pole-Illusion (Abbildung 2.3) verdeutlicht das Problem der nicht parallelen Ebenen. Dreht man den Zylinder um die z-Achse, scheint es von vorne betrachtet, als ob die Streifen sich nach oben bewegen. Die Vektorfelder der Bewegung und des optischen Flusses stehen senkrecht aufeinander.

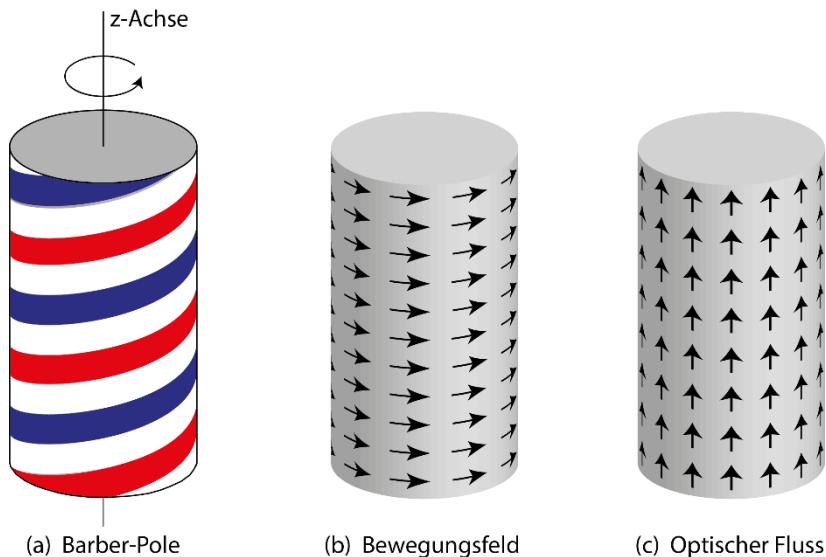


Abbildung 2.3 Barber-Pole-Illusion

Die beobachtete Szene der Kamera wurde außerdem als statisch angenommen. Trifft dies nicht zu, entsteht zusätzlicher optischer Fluss in der Szene, welcher nicht der Kamerabewegung entspricht. Da der Quadrocopter den Boden filmt, kann die Gesamtheit der Szene statisch betrachtet werden. Bewegen sich einzelne Objekte über den Boden, erzeugen diese eine lokale Störung des Vektorfeldes des optischen Flusses in ihrer Spur. Ein Algorithmus zur Bestimmung der Bewegung sollte diese Störungen erkennen, um das Ergebnis nicht zu verfälschen. [5] [6]

### 2.2.3 Methoden zur Berechnung

Der optische Fluss lässt sich durch mehrere grundverschiedene Methoden berechnen. Dabei wird nicht zwingend eine Lösung für die OFCE gesucht, sondern teilweise direkt die Transformation bestimmt.

Der optische Fluss lässt sich differentiell über den pixelgenauen Intensitätsgradienten eines Bildes berechnen. Diese aktuell auf dem Quadrocopter für rein translatorische Bewegungen implementierte Methode hat sich im Flug gut bewährt. Obwohl es auch hier eine Reihe von Algorithmen zur Erkennung der Rotation gibt, wurde dieser Ansatz nicht beachtet, um einen Vergleich verschiedener Methoden zu erhalten. [7]

Ein weiterer Ansatz ist die Minimierung der Summe der absoluten Differenzen (SAD<sup>6</sup>) zweier aufeinanderfolgender Bilder. Beim sogenannten Template-Matching wird eine Schablone (Template) im Suchbild verschoben, bis der SAD-Wert minimal ist. Dies entspricht einer Maximierung der Kreuzkorrelation. Das Verfahren arbeitet umso zuverlässiger, je höher der

<sup>6</sup> Sum of Absolute Differences =  $\sum_{\text{alle Pixel}} |[\text{Pixelwert Bild 1}] - [\text{Pixelwert Bild 2}]|$

globale Intensitätsgradient im Bild ist. Diese Technik eignet sich aber nur bedingt für den Einsatz am Quadrocopter, da beim Flug über eine Wiese oder schwach gemusterten Boden der Gradient minimal wird. [8]

Betrachtet man zwei aufeinanderfolgende Bilder einer statischen Szene, können diese durch affine Transformation in ein gemeinsames Koordinatensystem transformiert werden. Die Parameter der Transformation lassen dann einen Rückschluss auf die gesuchten Größen zu, wodurch die Bewegung rekonstruiert werden kann. Dieses Verfahren wird als Bildregistrierung bezeichnet und besitzt eine ganze Reihe verschiedener Lösungsansätze. Einige der Algorithmen sind sehr robust gegenüber den Bildeigenschaften (Helligkeitsverteilung). Um dem Quadrocopter eine zuverlässige Positionsbestimmung über Terrain aller Art zu ermöglichen, wird dieser Ansatz umgesetzt und an den Quadrocopter angepasst. Häufig greift die Registrierung auf die Spektraldarstellung der Bilder zurück.

## 2.3 Bilder als Signale

Wird einer physikalisch messbaren Größe eine Information zugeordnet, spricht man von einem Signal. Signale werden meist zeitabhängig beschrieben, die Interpretation der Variablen ist aber für die mathematische Beschreibung nicht relevant.

### 2.3.1 Interpretation als Signal

Bilder werden in der digitalen Bildverarbeitung als ortsabhängige zweidimensionale Signale behandelt. Eine Kamera wandelt während der Aufnahme die kontinuierliche Intensitätsverteilung eines gewissen Frequenzbandes in ein diskretes zweidimensionales Signal um, wobei jeder Pixel einem Messpunkt (Sample) entspricht.

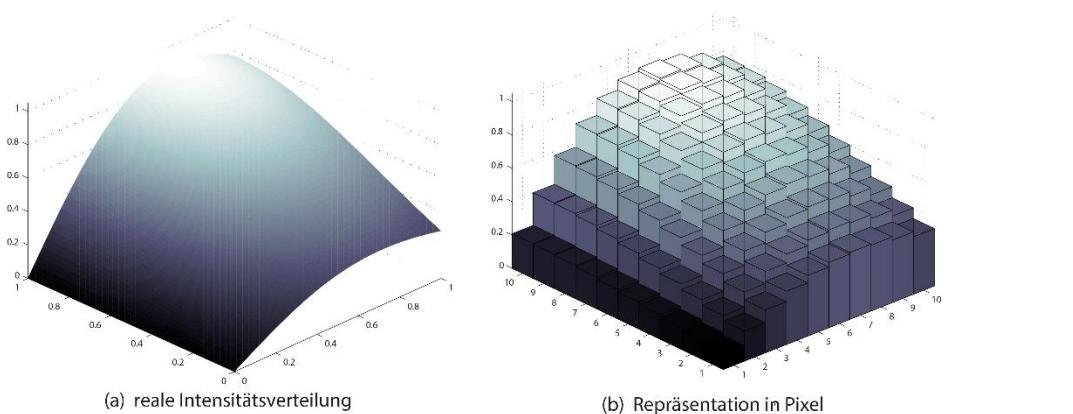


Abbildung 2.4 Vom realen Bild zum 2D-Signal

Ein Sample ist dabei nicht zeitabhängig, sondern ortsabhängig charakterisiert (siehe Abbildung 2.4). Die Folgenden Betrachtungen beziehen sich nur auf monochrome Bilder (Graustufenbilder). Ein RGB-Bild entsteht aus der Überlagerung der Intensitätsverteilungen dreier Frequenzbänder. Die Intensität (Pixelwerte) wird bei der Aufnahme meist in das diskrete Intervall  $[0,1, \dots, 255]$  eingeteilt (8-Bit Farbtiefe). Vor der Verarbeitung von Bildern erfolgt eine Umwandlung in das kontinuierliche Intervall  $(0, \dots, 1)$ , da Berechnungen dann wesentlich genauer ausgeführt werden können. Falls nicht anderes angegeben, beziehen sich die Pixelwerte im Folgenden immer auf dieses kontinuierliche Intervall.

### 2.3.2 Koordinatensysteme

Die zweidimensionale Position jedes Pixels eines Bilds wird mit Hilfe des Koordinatensystems eindeutig beschrieben. Abbildung 2.4 zeigt ein Bild in kartesischen Koordinaten. Seien  $m$  bzw.  $n$  die beiden Koordinatenachsen. Dann bezeichnet  $i[m_0, n_0]$  die Intensität des Pixels an der Position  $\langle m_0, n_0 \rangle$ . Falls nicht anders angegeben, beziehen sich alle Darstellungen auf kartesische Koordinaten.

Für einige mathematische Betrachtungen ist eine Transformation des Bilds in Polar-Koordinaten notwendig. In Polar-Koordinaten ist die Position eines Pixels durch Abstand  $r$  zum Mittelpunkt und den Winkel  $\theta$  gegeben (siehe Abbildung 2.5).

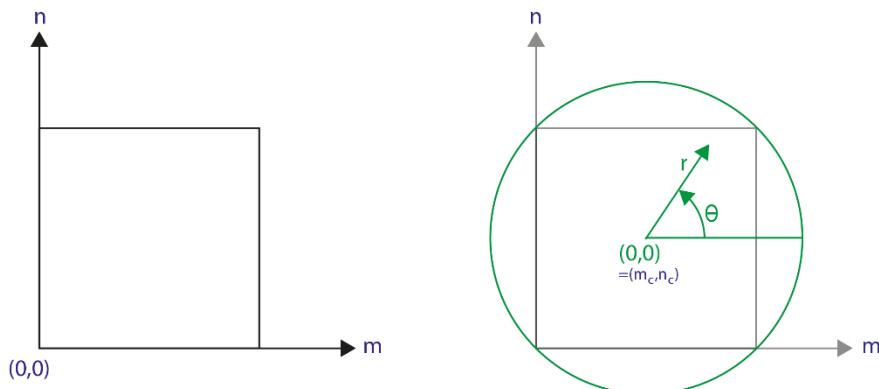


Abbildung 2.5 Kartesische und Polar-Koordinaten

Transformiert man ein Bild mit kartesischen Koordinaten  $\langle m, n \rangle$  in die Polar-Koordinaten  $\langle r, \theta \rangle$ , erhält man eine ungleichförmige Repräsentation desselben. Der Mittelpunkt des Bildes in kartesischen Koordinaten sei gegeben durch die Position  $\langle m_c, n_c \rangle$ . Die Transformation folgt der Vorschrift

$$r = \sqrt{(m - m_c)^2 + (n - n_c)^2} \quad (2.5)$$

$$\theta = \tan^{-1} \frac{n - n_c}{m - m_c} \quad (2.6)$$

für jedes Pixel  $(m, n)$  des Bilds. Dadurch ist der Informationsgehalt im Zentrum des Bilds am höchsten, die Randbereiche werden größer aufgelöst. Pixel mit gleichem  $r$  liegen radial mit gleichem Abstand um den Mittelpunkt  $(m_c, n_c)$ . Trägt man  $r$  und  $\theta$  auf der horizontalen bzw. vertikalen Achse auf, erhält man das transformierte Bild. Dabei hat die neue  $m$ -Achse einen Definitionsbereich von 0 bis  $R$ , die  $n$ -Achse wird in das Intervall  $[0, \dots, 2\pi]$  aufgeteilt.  $R$  ist der Maximalwert des Radius  $r$  und entspricht üblicherweise der halben Diagonalen des Bilds, um alle Pixel zu erfassen (siehe grüner Kreis in Abbildung 2.5 b). Abbildung 2.6 zeigt ein Bild in kartesischen Koordinaten und dessen Transformation in Polar-Koordinaten. Die schwarzen Bereiche entstehen durch die „ungültigen“ Pixel, welche innerhalb des Kreises  $R$ , aber außerhalb des Originalbildes liegen.



(a) Originalbild



(b) Polar-Transformation von Bild a

Abbildung 2.6 Polar-Transformation

### 2.3.3 System

Ein System  $T$  (siehe Abbildung 2.7) ist jede Einheit, welche aus einem gegebenen Signal ein anderes Signal als Ausgabe generiert. Da als Eingabe nur diskrete Signale zur Verfügung stehen, ist die Ausgabe wieder ein diskretes Signal. Man spricht von zeitdiskreten Systemen.



Abbildung 2.7 Modellierung eines Systems

Seien  $x[m, n]$  und  $y[m, n]$  die Ein- bzw. Ausgabesequenzen eines zweidimensionalen Systems ( $m$  und  $n$  bezeichnen den Ort des Samples im Signal). Die mathematische Beschreibung eines Systems lautet

$$y[m, n] = T\{x[m, n]\} \quad (2.7)$$

Das System ist also eine über das Eingangssignal definierte diskrete Funktion. Es handelt sich um ein lineares System, wenn jede Linearkombination zweier Eingangssignale dieselbe Linearkombination der Ausgangssignale zur Folge hat. Es muss folgende Gleichung (Superpositionsprinzip) gelten:

$$T\{ax_1[m, n] + bx_2[m, n]\} = aT\{x_1[m, n]\} + bT\{x_2[m, n]\} \quad (2.8)$$

Weiterhin ist ein System zeitinvariant, wenn es folgender Bedingung genügt:

$$T\{x[m - m_0, n - n_0]\} = y[m - m_0, n - n_0] \quad (2.9)$$

Das bedeutet, dass ein System bei gleicher Eingabe stets die gleiche Ausgabe liefert, egal wann (oder bei Bildern: wo) die Eingabe stattfindet. [9]

### 2.3.4 Diskrete Filter

Ist ein diskretes System linear und zeitinvariant, wird es diskreter Filter genannt. Häufig wird ein Filter nur durch seine Impulsantwort  $h[m, n]$  beschrieben, welche ihn eindeutig charakterisiert. Als Antwort auf die Delta-Funktion

$$\delta[m, n] = \begin{cases} 1 & \text{falls } m = 0, n = 0 \\ 0 & \text{falls } m \neq 0, n \neq 0 \end{cases} \quad (2.10)$$

liefert der Filter die Impulsantwort

$$h[m, n] = T\{\delta[m, n]\} \quad (2.11)$$

Ist diese örtlich beschränkt, handelt es sich um einen Filter mit endlicher Impulsantwort (FIR-Filter<sup>7</sup>). Dieser kann niemals instabil werden oder sich selbst zu Schwingungen anregen, da jede Antwort mit endlicher Ausbreitung abklingt. Es gilt:

$$h[k, l] = 0 \quad \forall k, l > x_0 \quad x_0 \in \mathbb{N} \quad (2.12)$$

---

<sup>7</sup> Finite Input Response Filter

Häufig wird die Impulsantwort mit dem Filter gleichgesetzt, man spricht vom „Filter  $h[m, n]$ “. Da  $h[m, n]$  nach Definition die Ausgabe eines Systems (auf die Delta-Funktion) ist, ist ein Filter selbst auch ein diskretes zweidimensionales Signal. Ein Filter ist signaltechnisch identisch mit einem Bild, der Begriff ergibt sich nur aus der konkreten Anwendung. FIR-Filter besitzen eine zentrale Rolle in der Bildverarbeitung, da sehr viele Operationen auf Bildern durch sie beschrieben werden können, zum Beispiel Weichzeichnungsfilter oder Bandpassfilter. [9]

### 2.3.5 Faltung

Filter werden durch ihre Antwort auf die Delta-Funktion beschrieben. Um die Antwort eines Filters auf ein beliebiges Signal (Bild) zu berechnen, muss das Eingangssignal mit der Impulsantwort des Filters in Verbindung gebracht werden. Jedes diskrete zweidimensionale Signal  $x[m, n]$  kann mithilfe der Delta-Funktion wie folgt dargestellt werden:

$$x[m, n] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} x[k, l] \delta[m - k, n - l] \quad (2.13)$$

Für die Systemantwort  $y[m, n]$  auf das Signal  $x[m, n]$  ergibt sich

$$y[m, n] = T\{x[m, n]\} = T\left\{ \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} x[k, l] \delta[m - k, n - l] \right\} \quad (2.14)$$

Da  $x[k, l]$  nicht von den Primärvariablen  $m, n$  abhängt, kann dies mit der geforderten Linearität zu

$$y[m, n] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} x[k, l] T\{\delta[m - k, n - l]\} \quad (2.15)$$

umgeformt werden. Setzt man nun Gleichung (2.11) ein, erhält man Dank des zeitinvarianten Charakters eines FIR-Filters

$$y[m, n] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} x[k, l] h[m - k, n - l] \quad (2.16)$$

Formel (2.16) ist die Faltung (engl. convolution) des Eingangssignals mit der Impulsantwort des Filters. Mit dem Sternchen-Operator für die Faltung kann dies wie folgt ausgedrückt werden:

$$y[m, n] = x[m, n] * h[m, n] \quad (2.17)$$

Bilder und Filter sind jedoch stets räumlich beschränkt, weswegen es sich in der Praxis um eine endliche und damit berechenbare Summe handelt. Wie man an Gleichung (2.16) leicht zeigen kann, ergibt sich aus einem Bild der Größe  $M_0 \times N_0$  und einer Impulsantwort des Filters der Größe  $M_1 \times N_1$  für die Größe der Faltung  $M_R \times N_R$  (das durch den Filter bearbeitete Bild):

$$M_R = M_0 + M_1 - 1 \quad N_R = N_0 + N_1 - 1 \quad (2.18)$$

Bild und Filter werden dabei mit Nullen fortgesetzt (zero-padding), soweit es die Berechnung erfordert.

Die Faltung kann durch Matrizenmultiplikationen dargestellt werden. Zur leichteren Veranschaulichung werden die eindimensionalen Signale  $x[n], n \in \mathbb{Z}_3$  und  $h[n], n \in \mathbb{Z}_2$  betrachtet.<sup>8</sup> Für das Ausgangssignal  $y[n]$  der Länge 4 (vgl. Formel (2.18)) ergibt sich:

$$y[n] = \begin{bmatrix} y[0] \\ y[1] \\ y[2] \\ y[3] \end{bmatrix} = \begin{bmatrix} h[0] & 0 & 0 \\ h[1] & h[0] & 0 \\ 0 & h[1] & h[0] \\ 0 & 0 & h[1] \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \end{bmatrix} \quad (2.19)$$

[9] [10]

### 2.3.6 Anwendung von FIR-Filtern

Mithilfe der Faltung können einige wichtige Bildbearbeitungsoperationen durchgeführt werden. Ein Filter wird normalerweise auf einem zweidimensionalen Fenster  $W$  der Mächtigkeit  $N_W = M_1 N_1$  definiert und verrechnet für ein Eingangsbild jeweils jeden Pixel mit der lokalen Umgebung. Statt eines bestimmten Filters können auch zwei Bilder miteinander verrechnet werden, zum Beispiel wird die Korrelation zweier Bilder durch deren Faltung berechnet.

Einen einfachen Weichzeichnungsfilter (Mittelwertfilter) der Größe  $3 \times 3$  beschreibt die Funktion

$$h[k, l] = \frac{1}{N_W} = \frac{1}{9} \quad k, l \in \{-1, 0, 1\} \quad (2.20)$$

---

<sup>8</sup>  $\mathbb{Z}_N$  beschreibt die Menge  $0, 1, \dots, N - 1$

Setzt man diesen Filter in Formel (2.17) ein, wird jedem Pixel der Mittelwert seiner acht Nachbarpixel und ihm selbst zugewiesen, wodurch das Bild unschärfer wird: jedem Pixel des Ursprungsbildes wird der Filter an Position  $k, l = 0$  überlagert.

Der Filter

$$h[k, l] = \begin{cases} -1 & l = -1 \\ 0 & l = 0 \\ 1 & l = 1 \end{cases} \quad k, l \in \{-1, 0, 1\} \quad (2.21)$$

berechnet den Gradienten des Bildes in n-Richtung. Der Wert eines jeden Pixels richtet sich nach der Differenz seiner Nachbarpixel. An klaren, senkrechten Kanten hat das Ergebnis helle Bereiche, an langsamen Übergängen Dunkle.

Abbildung 2.8 zeigt die räumliche Darstellung der beiden Filter, Abbildung 2.9 deren Anwendung.

	l			
	-1	0	1	
k	-1	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
	1	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
	1	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

	l			
	-1	0	1	
k	-1	1	0	-1
	0	1	0	-1
	1	1	0	-1

(a) Mittelwertfilter
(b) n-Gradient

Abbildung 2.8 räumliche Darstellung der Filter



Abbildung 2.9 Anwendung der Filter

### 2.3.7 Affine Transformationen auf Bildern

Die drei affinen Transformationen aus Abbildung 2.2 können für ein Bild  $i[m, n]$  mathematisch beschrieben werden. Verschiebt man das Bild um  $m_0$  bzw.  $n_0$ , entsteht das verschobene Bild  $i_t[m, n]$ . Es gilt der Zusammenhang

$$i[m, n] = i_t[m - m_0, n - n_0] \quad (2.22)$$

Die Rotation in der Bildebene um den Winkel  $\theta$  wird mit der Rotationsmatrix

$$R = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \quad (2.23)$$

beschrieben. Für ein um  $\theta$  rotiertes Bild  $i_r[m, n]$  gilt daher:

$$i[m, n] = i_r[m \cos \theta + n \sin \theta, -m \sin \theta + n \cos \theta] \quad (2.24)$$

Die Stauchung oder Streckung eines Bildes um die Faktoren  $\alpha$  bzw.  $\beta$  in  $m$ - bzw.  $n$ -Richtung skaliert die beiden Koordinatenachsen. Der Zusammenhang zwischen Originalbild und skaliertem Bild  $i_s[m, n]$  ist gegeben durch

$$i[m, n] = i_s[\alpha m, \beta n] \quad (2.25)$$

Die Kombination dieser Formeln erlaubt eine beliebige affine Transformation der Bilder.

## 2.4 Bilder im Frequenzspektrum

Die Faltung von Bildern mit Filtern (oder anderen Bildern) ist sehr rechenintensiv, da für jeden Pixel  $N_W$  Multiplikationen durchgeführt werden müssen. Für quadratische Bilder und Filter der Größen  $N \times N$  bzw.  $M \times M$  liegt die Komplexität der direkten Berechnung in der Größenordnung  $O(N^2 M^2)$ . Im Frequenzspektrum wird die Faltung jedoch zu einer einfachen Multiplikation, wodurch sich die Faltung wesentlich schneller berechnen lässt. Dies ist für die Robustheit der Lokalisation mittels optischen Flusses sehr wichtig ist.

Hinweis:  $i$  bezeichnet die imaginäre Zahl.  $j$  und  $k$  sind Laufvariablen, falls nicht anders angegeben.

### 2.4.1 Herleitung der Fouriertransformation auf diskreten Signalen

Die Fouriertransformation  $\mathcal{F}$  einer zweidimensionalen Funktion  $f(x, y)$  ist wie folgt definiert:

$$\mathcal{F}\{f\} = F(s_1, s_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-i2\pi(xs_1+ys_2)} dx dy \quad (2.26)$$

Die Transformierte  $F$  hängt nicht mehr von den Variablen  $x$  und  $y$  der Ursprungsfunktion ab, sondern von den Frequenzen  $s_1$  und  $s_2$ . Diese trigonometrische Repräsentation von  $f$  ist eindeutig und lässt sich mit der gleichen Formel (bis auf einen Vorfaktor) wieder in die Ursprungs-Domäne zurücktransformieren.

Jedes Signal lässt sich mit der Fouriertransformation in sein komplexes Frequenzspektrum übertragen. Bei diskreten Signalen (Bildern) kann Formel (2.26) nicht direkt angewendet werden, da die Funktion  $f$  nicht bekannt ist. Wie in Kapitel 2.3.1 gezeigt, ist jedes Pixel eines Bildes ein Sample einer unbekannten Intensitätsfunktion, also der gesuchten Funktion  $f$ . Gelingt es nun, diese durch ihre Stützstellen (den Samples) gegebene Intensitätsfunktion  $f$  im Frequenzspektrum darzustellen, hat man die Transformierte des Bildes gefunden. Abbildung 2.10 veranschaulicht dies für den eindimensionalen Fall. Für das diskrete Signal (Reihe aus zehn Pixeln) wird eine Funktion  $f(x)$  gesucht, die mit allen Pixelwerten übereinstimmt.

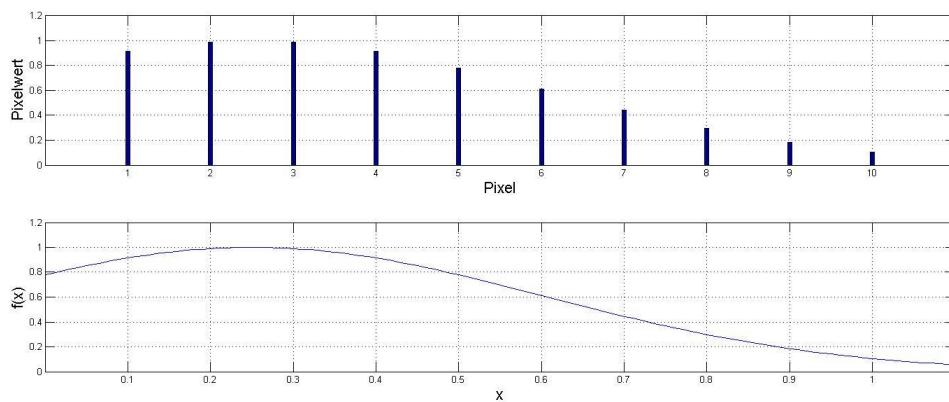


Abbildung 2.10 Intensitätsfunktion eines diskreten Bildes

Da die Intensitätsfunktion diskret vorliegt, ist eine diskrete Spektralfunktion gesucht. Die zweidimensionale Transformation ist nur eine Komposition zweier eindimensionaler Spektralfunktionen. Daher wird die Herleitung nur in einer Variablen betrachtet. Geht man von  $2N$  ( $N \in \mathbb{N}$ ) äquidistanten Stützstellen (Pixel)  $x_j$  für  $f(x)$  aus und projiziert diese auf das Intervall  $[-\pi, \pi]$ , gilt für die  $x_j$ :

$$x_j = -\pi + j \frac{\pi}{N} \quad j = 0, 1, \dots, 2N \quad (2.27)$$

$f(x_j)$  wird für die Herleitung  $2\pi$ -periodisch fortgesetzt, wodurch  $x_0$  und  $x_{2N}$  identisch sind. Dies stellt keine Einschränkung dar, da von  $2N$  Pixeln ausgegangen wurde. Die gesuchte spektrale Repräsentation von  $f(x_j)$  entspricht der Approximation durch trigonometrische Interpolation.

Für die Approximation an  $2N$  Stützstellen ist die Interpolation definiert durch ein trigonometrisches Polynom vom Grad  $n = N$  der Form

$$p(x) = \sum_{k=-n}^n c_k e^{ikx} \quad (2.28)$$

Die Interpolationsaufgabe ist gegeben durch die Gleichung

$$f(x_j) = p(x_j) \quad j = 0, 1, \dots, 2N - 1 \quad (2.29)$$

und beschränkt sich somit auf das Bestimmen der  $2N$  Koeffizienten  $c_k$ . Da die Anzahl der Stützstellen und der Koeffizienten (=Freiheitsgrade) identisch ist, ist diese Aufgabe eindeutig lösbar (eingeschränkt auf Frequenzen unter der Nyquist-Frequenz, für diese Arbeit nicht relevant). Wird Formel (2.28) in Formel (2.29) eingesetzt, erhält man durch Nutzen der Periodizität

$$f(x_j) = \sum_{k=-N+1}^N c_k e^{ikx_j} \quad (2.30)$$

Zur Bestimmung der  $c_k$  ist das lineare Gleichungssystem zu lösen. Eine Lösung (Beweis siehe Anhang) ist gegeben durch folgende Formel:

$$c_k = \frac{1}{2N} \sum_{j=0}^{2N-1} f(x_j) e^{-ikx_j} \quad (2.31)$$

Die Integrale zur Berechnung der Koeffizienten  $c_k$  der Fourierreihe einer Funktion  $f$  lauten

$$c_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{-ikx} dx \quad (2.32)$$

Nähert man diese Integrale mit der zusammengesetzten Trapezregel, folgt direkt wieder Formel (2.31). Durch die trigonometrische Interpolation der Pixel eines Bildes erhält man also

deren eindeutig bestimmte, diskrete Spektralfunktion. Daher kann mathematisch auf die Eigenschaften der kontinuierlichen Fouriertransformation zurückgegriffen werden. Hierbei müssen jeweils nur die  $2N$  Koeffizienten  $c_k$  bestimmt werden, um die Fouriertransformierte eines diskreten Signals der Länge  $2N$  zu erhalten. Für den zweidimensionalen Fall werden zwei Transformationen über jeweils eine Raumrichtung hintereinander ausgeführt, da die Dimensionen unabhängig voneinander sind. Die Spektralfunktion eines Bildes der Größe  $M \times N$  ist durch die  $MN$  Koeffizienten  $c_{k,l}$  gegeben. Dadurch besteht die Möglichkeit, für beliebige Bilder eine Fouriertransformation anzugeben, um Bildoperationen im Frequenzspektrum durchzuführen. Oft lässt sich dies deutlich effizienter gestalten als die direkte Verarbeitung der Bildinformationen.

Im Folgenden werden Funktionen mit kleinen Buchstaben und deren Fouriertransformierten mit großen Buchstaben gekennzeichnet. Die Transformierte des Bildes  $i_k[m, n]$ ,  $\mathcal{F}\{i_k\}$ , wird zum Beispiel mit  $I_k[s_1, s_2]$  beschrieben. Zu beachten ist der Variablenwechsel vom Bild- in den Frequenzbereich. Abbildung 2.11 zeigt die Transformation der Rechteckfunktion in einer und in zwei Dimensionen. Man erkennt deutlich, dass  $\mathcal{F}\{rect(m, n)\}$  aus der Verkettung zweier Transformationen jeweils einer Raumrichtung hervorgeht. [9] [10] [11]

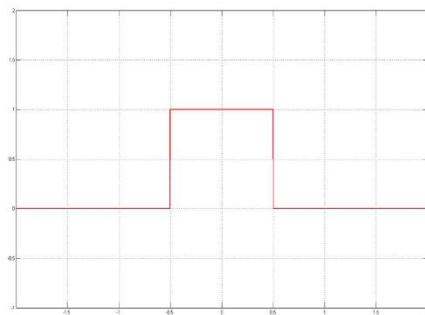
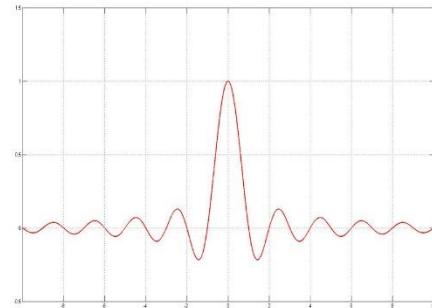
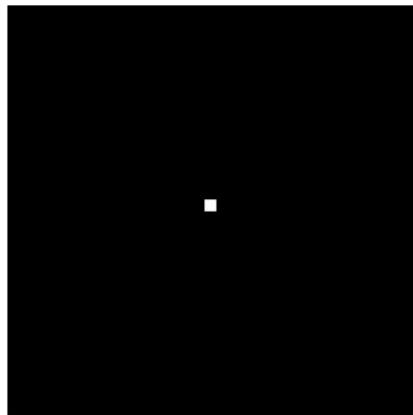
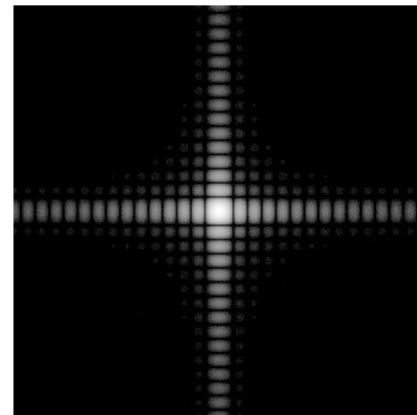
(a)  $rect(x)$ (b)  $\mathcal{F}\{rect(x)\}$ (c)  $rect(m, n)$ (d)  $\mathcal{F}\{rect(m, n)\}$ 

Abbildung 2.11 Fouriertransformation der Rechteckfunktion

## 2.4.2 Eigenschaften der Fouriertransformation

Aus der Herleitung wird ersichtlich, dass die Fouriertransformierte eines Bildes die Darstellung der zugrundeliegenden Intensitätsfunktion im Frequenzbereich ist. Hohe Frequenzen sind auf starke Intensitätsdifferenzen zurückzuführen, was sich im Ursprungsbild durch klare Kanten äußert. So lassen sich zum Beispiel in einem Bild die Kanten hervorheben, indem man bei der Transformierten des Bildes alle niedrigen Frequenzen unterhalb einer bestimmten Schwellenfrequenz auf null setzt. Nach der Rücktransformation in den Bildbereich sind bis auf eindeutige Kanten alle Details in dem Bild verschwunden. Im Gegensatz dazu dämpft die Transformation des Weichzeichnungsfilters (Formel (2.20)) die hohen Frequenzen.

Die Fouriertransformation besitzt weitere wichtige Eigenschaften, welche man sich für die Bildverarbeitung zunutze machen kann. Für ein Bild  $i[m, n]$  sei die Fouriertransformierte gegeben durch

$$\mathcal{F}\{i\} = \mathcal{F}\{i[m, n]\} = I[s_1, s_2] \quad (2.33)$$

Im speziellen sind folgende fünf Eigenschaften wichtig:

1. Die Fouriertransformation ist allgemein für komplexe Eingabedaten definiert. Ist die Eingabesequenz der Länge  $M \times N$  real, gilt für die Fouriertransformation<sup>9</sup>:

$$I[s_1, s_2] = I[M - s_1, N - s_2]^* \quad (2.34)$$

Die Transformation ist symmetrisch um den Mittelpunkt der Daten mit dem komplex Konjugierten.

2. Das Shift-Theorem besagt für verschobene Bilder (siehe Formel(2.22)):

$$\mathcal{F}\{i[m + m_0, n + n_0]\} = e^{-i2\pi(m_0 s_1 + n_0 s_2)} I[s_1, s_2] \quad (2.35)$$

Die Spektralfunktion stimmt bis auf einen Vorfaktor mit der Transformierten aus Formel (2.33) überein. So lassen sich translatorisch verschobene Bilder im Phasenspektrum miteinander vergleichen. Der Betrag der Spektralfunktion ist identisch zur Transformierten des nicht verschobenen Bildes, da der Betrag der komplexen  $e$ -Funktion immer 1 ist. Für die Beträge gilt

$$|\mathcal{F}\{i[m, n]\}| = |\mathcal{F}\{i[m + m_0, n + n_0]\}| \quad (2.36)$$

---

<sup>9</sup>  $a^*$  bezeichnet das komplex Konjugierte von  $a$

3. Die Rotation eines Bildes (vgl. Formel (2.24)) hat eine identische Rotation der Transformierten zur Folge. Dreht man das Bild  $i$  um den Winkel  $\theta$  gegen den Uhrzeigersinn, gilt:

$$\begin{aligned} \mathcal{F}\{i[m \cos \theta + n \sin \theta, -m \sin \theta + n \cos \theta]\} \\ = I[s_1 \cos \theta + s_2 \sin \theta, -s_1 \sin \theta + s_2 \cos \theta] \end{aligned} \quad (2.37)$$

4. In den Algorithmen zur Registrierung wird weiterhin auf die Skalierung der Transformation zurückgegriffen. Streckt oder staucht man ein Bild in  $m$ - bzw.  $n$ -Richtung um die Faktoren  $\alpha$  bzw.  $\beta$  (Formel (2.25)), gilt:

$$\mathcal{F}\{i[\alpha m, \beta n]\} = \frac{1}{|\alpha\beta|} I\left[\frac{S_1}{\alpha}, \frac{S_2}{\beta}\right] \quad (2.38)$$

5. Wie zu Beginn dieses Kapitels angesprochen, wird die Faltung im Frequenzspektrum zu einer Multiplikation. Die Faltung von zwei Bildern  $x$  und  $h$  nach Formel (2.17) kann also umgeschrieben werden zu

$$Y[s_1, s_2] = X[s_1, s_2] \odot H[s_1, s_2] \quad (2.39)$$

Der Operator  $\odot$  bezeichnet das Hadamard-Produkt, die elementweise Multiplikation der Matrizen. Formel (2.39) ist eine reine Array-Multiplikation und kann sehr schnell berechnet werden. Abbildung 2.12 zeigt die direkte Faltung sowie die Berechnung im Frequenzspektrum (F steht für die Fouriertransformation,  $F^{-1}$  für die Rücktransformation).

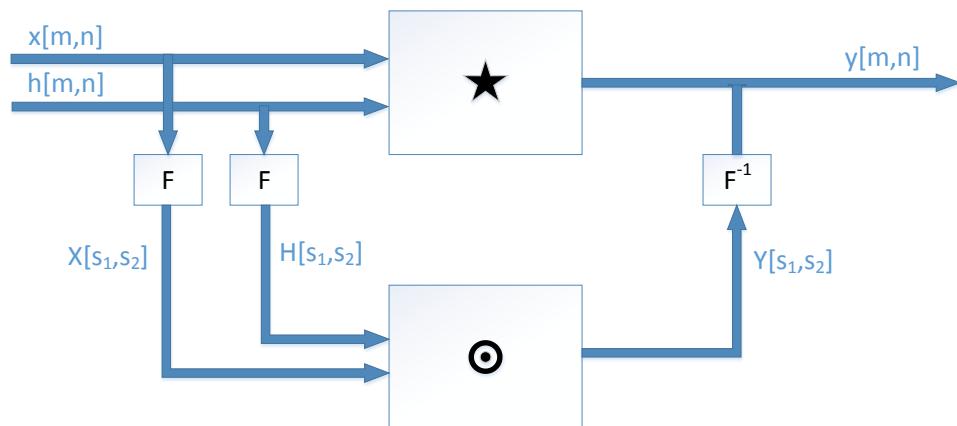


Abbildung 2.12 Filter im Frequenzspektrum

Die Beweise zu den Eigenschaften der Fouriertransformation finden sich in [11].

### 2.4.3 Randeffekte

Bei der Fouriertransformation von realen Bildern treten Randeffekte auf, da die Bilder örtlich beschränkt sind. Im Spektrum resultieren hohe Frequenzen aus schnellen

Intensitätsänderungen. Die Intensitätswerte eines Bildes sind in beide Dimensionen örtlich beschränkt (auf die Größe des Bildes), außerhalb dieses beschränkten Bereichs werden sie als 0 angenommen. Bei der Transformation des Bilds entstehen hohe Frequenzen entlang der Ränder, da die Fouriertransformation für die gesamte Bildebene definiert ist. Diese Frequenzen sind nicht Teil der Bildinformation und werden Randeffekte genannt.

Multipliziert man die Bilder vor der Fouriertransformation mit einem Hanning-Fenster, können die Randeffekte minimiert werden. Die diskrete Hanning-Funktion  $w(n)$  der Länge  $N$  ist definiert mit

$$w(n) = 0.5 \left( 1 - \cos \left( 2\pi \frac{n}{N-1} \right) \right) \quad n = 0, 1, \dots, N-1 \quad (2.40)$$

Abbildung 2.13 stellt  $w(n)$  und das zweidimensionale Hanning-Fenster

$$w(m, n) = w(m) w(n)^T \quad (2.41)$$

grafisch dar.

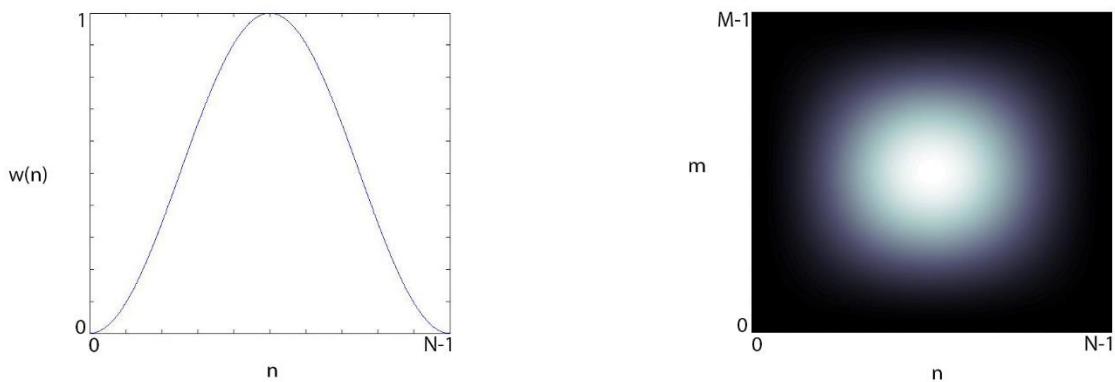


Abbildung 2.13 Hanning-Fenster

Multipliziert man die Intensitätswerte eines Bildes mit dem zweidimensionalen Hanning-Fenster, wird das Bild zum Rand hin schwarz. Abbildung 2.14 zeigt die Transformation für ein Bild mit und ohne Hanning-Fenster (FFT steht für Fouriertransformation – siehe nächstes Kapitel). Die Randeffekte äußern sich durch ein helles Kreuz entlang der Mittellinien des Spektralbilds. Es ist deutlich sichtbar, dass das helle Kreuz in Abbildung 2.14 (d) verschwunden ist.

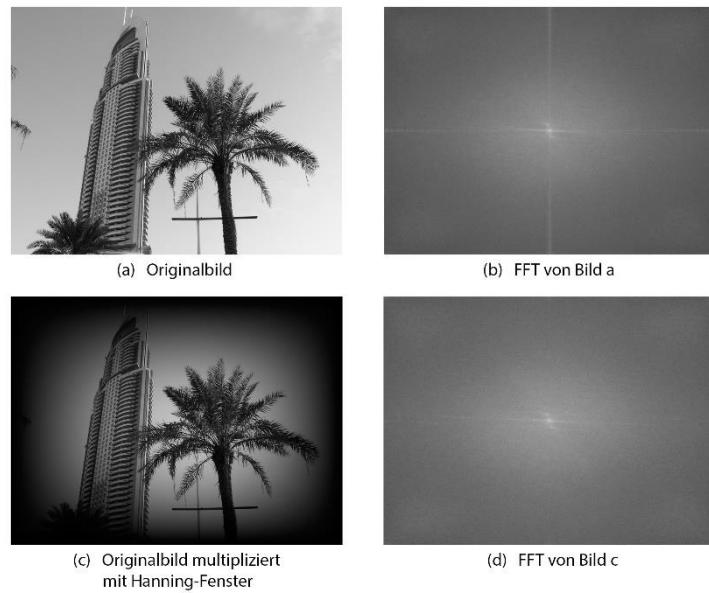


Abbildung 2.14 Hanning-Fenster in der Anwendung

Statt einem Hanning-Fenster könnte auch eine einfache Dreiecks- oder Kosinus-Funktion auf das Bild multipliziert werden. Das Hanning-Fenster besitzt aber eine stetig differenzierbare Fortsetzung an den Bildrändern (andernfalls würde die Fouriertransformation erneut Kanten aufweisen) und steigt dann zur Bildmitte hin schnell an, um möglichst viele Details des Bildes zu berücksichtigen. Damit ist das Hanning-Fenster die optimale Wahl zur Reduzierung der Randeffekte. [12]

#### 2.4.4 Die schnelle Fouriertransformation

Die schnelle Fouriertransformation (FFT<sup>10</sup>) ist ein Verfahren zur Berechnung der Transformierten eines diskreten Signals. Die Transformation wird dabei nicht direkt, sondern nach dem Teile-und-herrsche-Verfahren berechnet. Die FFT geht zurück auf einen Algorithmus von J. W. Cooley und J. Tukey. Heute ist eine ganze Reihe optimierter Methoden verfügbar. Während die direkte Berechnung der diskreten Fouriertransformation für ein Signal der Länge  $N$  die Zeit  $\mathcal{O}(N^2)$  benötigt, ist die FFT in  $\mathcal{O}(N \log_2 N)$  berechenbar. Dadurch ist zum Beispiel die Berechnung der Faltung (vgl. Abbildung 2.12) im Spektralbereich wesentlich schneller, als die herkömmliche Variante.

In den Algorithmen zur Registrierung wird die Fouriertransformation daher häufig nur mit FFT, die Rücktransformation mit FFT<sup>-1</sup> oder IFFT angegeben. Eine ausführliche Beschreibung sowie die Herleitung der FFT ist in [10] gegeben.

---

<sup>10</sup> Fast Fourier Transform

## 2.5 Bildregistrierung

Unter Bildregistrierung versteht man das Bestimmen der geometrischen Beziehung zweier Bilder. Registrierte Bilder können in ein einheitliches Koordinatensystem gebracht werden. Für die Berechnung des optischen Flusses mit Hilfe der Registrierung genügt die Betrachtung verschobener, rotierter und skalierter Bilder. Mit speziellen Registrierungstechniken können auch Verzerrungen, Neigungen und Verformungen der Bilder erkannt werden.

Die Betrachtungen in diesem Kapitel beschränken sich auf Rotation, Skalierung und Translation. Bewegt sich die Kamera an einer statischen Szene vorbei, nimmt sie zwei leicht verschobene Bilder auf. Die Registrierung bestimmt die translatorische Differenz zwischen beiden Bildern. Analog werden Rotation und Skalierung bestimmt, wenn sich die Kamera dreht oder auf die Szene zubewegt (vgl. Abbildung 2.2). Berechnet man die affinen Transformationen einer Bilderreihe jeweils Bild für Bild, kann die komplette Bewegung der Kamera rekonstruiert werden. Eine wichtige Eigenschaft der betrachteten Algorithmen ist die Laufzeit. Nur bei sehr schnellen Bilderfolgen ist eine Szene zweier Bilder als statisch anzunehmen. Außerdem darf die translatorische Verschiebung nicht zu groß sein, da sonst die Überlappung zu gering wird. Ohne ausreichende Überlappung kann keine Korrespondenz hergestellt werden. Die betrachteten Algorithmen beschränken sich alle auf nur einen Farb-Kanal (Graustufen-Bild). Kompliziertere Verfahren beachten durchaus mehrere Kanäle, eignen sich aber aufgrund ihrer Komplexität nicht für die Anwendung auf dem Quadrocopter.

### 2.5.1 Verfahren der Registrierung

Die Bildregistrierung besitzt ein sehr breites Anwendungsfeld, wodurch es mittlerweile eine große Menge verschiedener Registrierungs-Algorithmen gibt.

Die diskrete Methode arbeitet mit Objekterkennung. Durch affine Transformation korrespondierender Objekte in zwei Bildern erhält man deren Transformation. Je deutlicher im Sichtfeld der Kamera Objekte zu erkennen sind, desto besser arbeitet diese Methode. Einen interessanten Ansatz bei dem die konvexen Hüllen der Objektbegrenzungen verglichen werden, liefern Z. Yang und F. S. Cohen. [13]

Betrachtet man die Fourier-Transformierten zweier Bilder, lassen sich die Phasenspektren miteinander vergleichen. Aufgrund der Eigenschaften der Fouriertransformation (siehe Kapitel 2.4.2) können nur durch Translation versetzte Bilder mit der Phasenkorrelation direkt registriert werden. Eine Transformation der Koordinatensysteme vor der Fouriertransformation erlaubt außerdem eine Registrierung beliebig transformierter Bilder. Die spektrale Betrachtungsweise liefert eine hohe Robustheit gegenüber

Beleuchtungsänderungen. Außerdem erfordert sie keine klaren Objekte im Bild. Wegen diesen wichtigen Eigenschaften und effizienten Möglichkeiten zur Berechnung wird ein Algorithmus auf dem Quadrocopter implementiert, welcher auf die Phasenkorrelation zurückgreift.

Es könnte auch eine Methode von Q. Zheng und R. Chellappa auf dem Quadrocopter eingesetzt werden. Mit einem Gabor-Filter, der den Winkel des einfallenden Lichts eines Bildes berechnet, wird zuerst die Rotation annäherungsweise bestimmt. Anschließend werden Translation und Skalierung durch einige Feature-Points (korrespondierende, markante Punkte) berechnet. Damit erzielen sie gute Ergebnisse beim Flug mit einem Ballon über Böden verschiedener Beschaffenheit. In wieweit sich dieses Verfahren für den Quadrocopter eignet, müsste gesondert evaluiert werden, da die Systeme sich im Flugverhalten deutlich unterscheiden (z.B. Vibrationen beim Quadrocopter Aufgrund der Rotoren). [14]

Einen umfassenden Überblick über verfügbare Methoden liefert das Buch *Image Registration* [15]. Hier werden vor Allem Point- und Feature-Matching ausführlich beleuchtet, sowie Methoden zur Berechnung der Korrelation von Bildern vorgestellt.

## 2.5.2 Phasenkorrelation

Die Phasenkorrelation nutzt die Invarianz der Fouriertransformation gegenüber Translation. Unterscheiden sich zwei Bilder nur durch Translation um  $\Delta m$  bzw.  $\Delta n$ , kann diese Verschiebung berechnet werden. Das Ausgangsbild sei  $i_1[m, n]$ , das Verschobene  $i_2[m, n]$ . Für die überlappenden Bereiche der Bilder gilt:

$$i_2[m, n] = i_1[m + \Delta m, n + \Delta n] \quad (2.42)$$

Nach dem Shift-Theorem (2.35) gilt für die Transformierten:

$$I_2[s_1, s_2] = e^{-i2\pi(s_1\Delta m + s_2\Delta n)} I_1[s_1, s_2] \quad (2.43)$$

Die Korrelation von Signalen kann über den Vergleich der spektralen Leistungsdichten berechnet werden. Die Formel zur Berechnung der Korrelation wird Cross-Power-Spectrum (CPS) genannt. Nach [16] gilt<sup>11</sup>:

$$CPS = \frac{I_1[s_1, s_2] I_2^*[s_1, s_2]}{|I_1[s_1, s_2] I_2[s_1, s_2]|} \quad (2.44)$$

---

<sup>11</sup>  $a^*$  bezeichnet das komplexe Konjugierte von  $a$

Setzt man Formel (2.43) ein, zeigt sich der Zusammenhang zwischen CPS und Translation:

$$\frac{I_1[s_1, s_2]I_2^*[s_1, s_2]}{|I_1[s_1, s_2]I_2[s_1, s_2]|} = \frac{I_1[s_1, s_2]I_1^*[s_1, s_2]e^{-i2\pi(s_1\Delta m + s_2\Delta n)}}{|I_1[s_1, s_2]I_1[s_1, s_2]e^{-i2\pi(s_1\Delta m + s_2\Delta n)}|} \quad (2.45)$$

Der Betrag der imaginären Exponentialfunktion ist 1. Da die Phase einer komplexen Zahl multipliziert mit seinem komplex Konjugierten immer 0 ist, kann dieser Ausdruck gekürzt werden, es bleibt:

$$\frac{I_1[s_1, s_2]I_2^*[s_1, s_2]}{|I_1[s_1, s_2]I_2[s_1, s_2]|} = e^{-i2\pi(s_1\Delta m + s_2\Delta n)} \quad (2.46)$$

Daraus folgt, dass die Phase des CPS der Phasenverschiebung der Bilder  $I_1$  und  $I_2$  entspricht (vgl. Formel (2.43)). Berechnet man vom CPS die inverse Fourier-Transformation, wird genau diese Phase in die räumliche Translation  $\langle \Delta m | \Delta n \rangle$  übersetzt:

$$\mathcal{F}^{-1}\{CPS\} = \mathcal{F}^{-1}\{e^{-i2\pi(s_1\Delta m + s_2\Delta n)}\} = \delta[m - \Delta m, n - \Delta n] \quad (2.47)$$

Bei realen Bildern ist Formel (2.42) nie exakt erfüllt, da sich zum Beispiel durch Bewegung immer auch die Schatten leicht ändern. Außerdem stimmen die Randbereiche der Bilder durch die Translation nicht überein. Die inverse Transformation des CPS ( $\mathcal{F}^{-1}\{CPS\}$ ) ist daher keine Impulsfunktion. Vielmehr ist die Translation durch das Maximum (Peak) dieser Korrelationsoberfläche gegeben. Für die Bestimmung der Translation zwischen zwei Bildern mittels Phasenkorrelation (Abbildung 2.15) gilt:

$$\langle \Delta m | \Delta n \rangle = \max_{m,n} \mathcal{F}^{-1}\{CPS\} \quad (2.48)$$

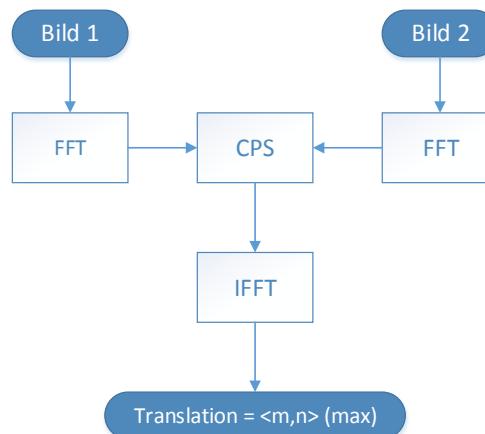


Abbildung 2.15 Phasenkorrelation

Je höher die Übereinstimmung der Bilder ist, desto höher ist auch der Peak-Wert. Bei genauer Betrachtung der Formel des CPS (Formel (2.44)) stellt man fest, dass es sich um einen Filter

handelt. Der Nenner dient nur der Normierung. Die beiden Bilder werden also miteinander gefaltet, das CPS berechnet dabei die Übereinstimmung für jede mögliche Verschiebung der Bilder zueinander. Je höher die Übereinstimmung für eine bestimmte Verschiebung ist, desto „heller“ ist das Ergebnis an den Koordinaten der Verschiebung. Man könnte die Registrierung translatorisch versetzter Bilder auch mit der direkten Faltung der Bilder lösen. Allerdings ist der Weg über das CPS, neben deutlich geringerem Rechenaufwand, robuster gegenüber dynamischen Lichtverhältnissen. Wird die komplette Szene (oder nur einzelne Bereiche) zwischen den beiden Aufnahmen heller oder dunkler, nimmt die pixelgenaue Übereinstimmung stark ab. Das Frequenzmuster bleibt hingegen nahezu konstant, es ändern sich hauptsächlich die Amplituden der Spektralfunktion. [17]

Abbildung 2.16 zeigt die inverse Fouriertransformation des CPS für zwei verschobene Bilder. Bild (b) wurde etwas Rauschen hinzugefügt, um eine realitätsnähere Verteilung zu erhalten. Die Translation des hellen Impulses in Bild (c) entspricht, von der Bildmitte aus gerechnet, genau der Verschiebung nach links unten von Bild (b).



Abbildung 2.16 Anwendung der Phasenkorrelation

# 3 Konzept

Das bestehende Positionsbestimmungssystem des Quadrocopters liefert eine zuverlässige Ortung, wenn der Quadrocopter sich nicht um die Gier-Achse dreht. In dieser Arbeit wird das System erweitert, um diese Einschränkung zu beseitigen. Eine Rotation um die Gier-Achse ist zum Beispiel erforderlich, wenn auf dem Quadrocopter eine nach vorne gerichtete Kamera zur Hinderniserkennung eingesetzt wird, welche immer in Flugrichtung zeigen soll.

## 3.1 Überblick

Der Quadrocopter besitzt an seiner Unterseite eine Kamera, welche senkrecht auf den Boden gerichtet ist. Durch Bildregistrierung wird auf den optischen Fluss geschlossen, welcher eine iterative Positionsbestimmung ermöglicht. Die Bildregistrierung erweitert das aktuelle System. Das bestehende Positionsbestimmungssystem berechnet den optischen Fluss nach der differentiellen Methode und wird im Folgenden schlicht „optischer Fluss“ genannt.

Es werden drei Konzepte zur iterativen Positionsbestimmung mittels Registrierung vorgestellt. Abbildung 3.1 zeigt die grundlegende Integration aller Konzepte in das bestehende System.

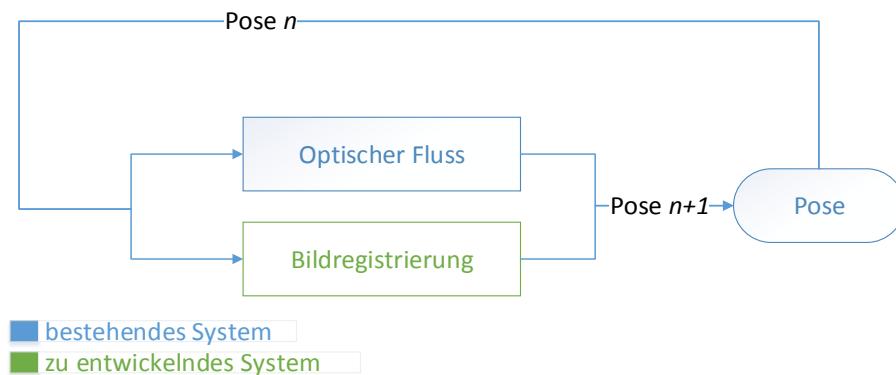


Abbildung 3.1 Allgemeines Konzept

Die Konzepte bauen aufeinander auf und unterscheiden sich dabei in den benötigten Eingabedaten der Registrierung, dem Ausführungsintervall und der Art, wie die Pose aus den zwei unterschiedlichen Systemen gewonnen wird (Abbildung 3.2). Der optische Fluss sowie Konzept 1 der Registrierung sind nicht in der Lage, die Orientierungsänderung zu ermitteln. Der Winkel-Anteil der Pose wird hier durch andere Sensoren ermittelt.



Abbildung 3.2 Übersicht der Konzepte

Im ersten Konzept übernimmt die Registrierung die Positionsbestimmung während eines jeden Giervorgangs. Solange der Quadrocopter nicht giert, kommt der bestehende optische Fluss zum Einsatz. Das neue System nimmt ein Bild zu Beginn und eines nach dem Giervorgang auf. Durch Information über Höhenänderung und Rotationswinkel kann mit der Phasenkorrelation die Positionsänderung während des Giervorgangs berechnet werden.

Um Genauigkeit und Verlässlichkeit zu erhöhen, wird im zweiten Konzept der Registrierungs-Algorithmus erweitert. Er berechnet die Rotation sowie Skalierung zwischen den zwei Bildern. Dadurch können mit der Phasenkorrelation deutlich bessere Ergebnisse erzielt werden. Man kann dieses Verfahren dazu einsetzen, für ein Gier-Manöver neben Translation auch den Gierwinkel zu bestimmen.

Im dritten Konzept wird die Position kontinuierlich durch Registrierung bestimmt. Sobald aktuelle Positionsinformationen vorliegen, wird ein neues Bild aufgenommen. Dieses wird dann mit dem zuletzt aufgenommenen Bild registriert, um die Position zu aktualisieren. Erst dadurch wird das Verfahren auf dem realen System einsatzfähig. Der deutlich höhere zeitliche Abstand zweier Bilder bei Konzept 2 führte im Flug zu komplett versetzten Bildausschnitten, was eine Registrierung unmöglich macht.

## 3.2 Grundlagen

Die Koordinatenachsen werden so festgelegt, dass der Boden der  $xy$ -Ebene entspricht (Definition nach Kapitel 2.2).  $x$ - und  $y$ -Achse sind dabei fest im Raum und drehen sich nicht mit dem Quadrocopter mit. Die Position des Quadrocopters wird mit dem 3D-Vektor  $\langle x, y, z \rangle$  angegeben. Da oft eine Projektion der Position in die  $xy$ -Ebene ausreicht, gleichzeitig aber die Orientierung wichtig ist, wird auch die vereinfachte Pose  $\langle x, y, \varphi \rangle$  verwendet.  $\varphi$  entspricht

der Drehung einer festgelegten Vorausrichtung des Quadrocopters um die  $z$ -Achse (Gierwinkel).

Die Registrierung berechnet die affine Transformation zweier Bilder zueinander. Um daraus ein System zur iterativen Positionsbestimmung zu entwickeln, muss das Koordinatensystem der Bilder mit dem Koordinatensystem des Quadrocopters in Verbindung gebracht werden. Sind zwei Bilder rein translatorisch verschoben, lässt sich mit der Phasenkorrelation die Verschiebung des Quadrocopters bestimmen. Unter der Annahme, dass Bewegungsfeld und optischer Fluss identisch sind (siehe Kapitel 2.2), benötigt man einen Umrechnungsfaktor für Pixelanzahl zu tatsächlicher Bewegung in Metern. Die Größe des Sichtfelds der Kamera ist direkt proportional zur Flughöhe. Daraus ergibt sich ein konstanter Umrechnungsfaktor, der mit der Flughöhe multipliziert werden muss.

Alle drei Konzepte gehen von einer waagrechten Fluglage des Quadrocopters aus. Diese Annahme ist im realen Flug nicht immer erfüllt und führt zu Fehlern in der Positionsbestimmung. Ändert sich der Roll- oder Nick-Winkel bei konstanter Position, verschiebt sich das Sichtfeld der Kamera über den Boden. Der Registrierungsalgorithmus misst eine Translation, die in Bewegung übersetzt wird. Den Einfluss der Neigung (Roll- und Nick-Winkel) zeigt Abbildung 3.3.

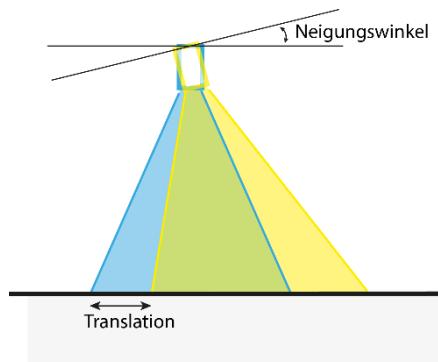


Abbildung 3.3 Einfluss der Neigung auf die Positionsbestimmung

Im Gegensatz zum senkrecht nach unten gerichteten Sichtfeld (blaue Fläche im Bild) ist das Sichtfeld der geneigten Kamera (gelbe Fläche) verschoben. Diese Translation entspricht aber nicht der Bewegung, da die Kameras sich am gleichen Ort befinden. Bei höheren Neigungswinkeln wird das Bild außerdem verzerrt, da das Sichtfeld schräg auf den Boden trifft.

Über die Winkeländerung zwischen zwei Bildern kann der Roll- oder Nick-Winkel-Fehler theoretisch herausgerechnet und damit kompensiert werden (sog. Neigungskompensation). Im Flugbetrieb hatte dies jedoch keine positiven Auswirkungen auf die Genauigkeit der Positionsbestimmung.

### 3.3 Registrierung vor und nach dem Gieren

Abbildung 3.4 zeigt die schematische Darstellung des ersten Konzepts. Benötigte Eingangsgrößen für die Registrierung sind blau dargestellt, die Ausgabe grün.

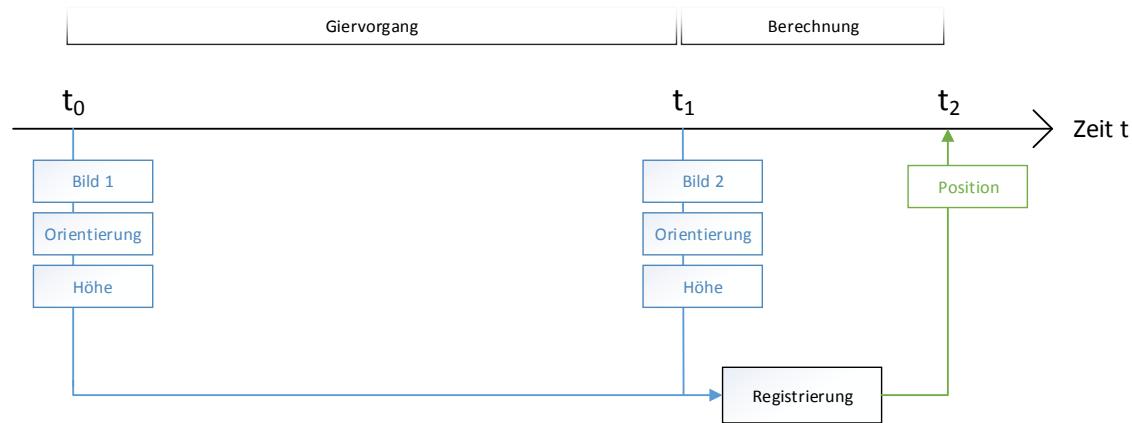


Abbildung 3.4 Schema Konzept 1

Startet der Quadrocopter einen Giervorgang (Zeitpunkt  $t_0$ ), wird die herkömmliche Positionsbestimmung pausiert und ein Bild mit der Kamera aufgenommen. Während des Gierens erhält der Quadrocopter keine neuen Positionsinformationen. Nach der Drehung ( $t_1$ ) wird ein zweites Bild aufgenommen. Der Quadrocopter greift ab  $t_1$  wieder auf den optischen Fluss zurück. Sobald die Registrierung der beiden Bilder abgeschlossen ist ( $t_2$ ), wird die Positionsänderung des Giervorgangs zur aktuellen Position addiert. Zusätzlich zu den Bildern dienen die Orientierung und die Höhe zu den Zeitpunkten  $t_0$  und  $t_1$  als Eingabeparameter für den Registrierungs-Algorithmus. Die Registrierung folgt dem Schema aus Abbildung 3.5.

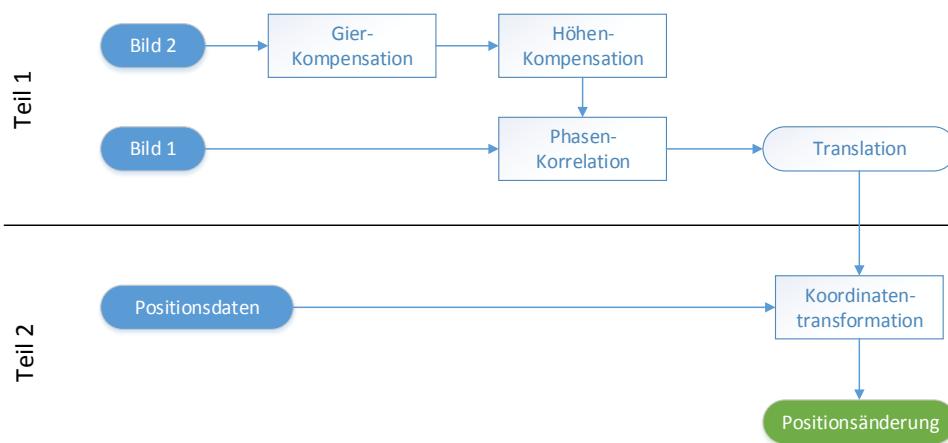


Abbildung 3.5 Schema der Registrierung

Das zweite Bild wird um den negativen Gierwinkel rotiert. Skaliert man es zusätzlich auf die relative Höhenänderung während des Manövers, sind die beiden Bilder nur translatorisch verschoben. Die Verschiebung der Bilder kann direkt mit der Phasenkorrelation (Abbildung

2.15) bestimmt werden. Damit ist die eigentliche Registrierung abgeschlossen (Teil 1). Anschließend wird aus der Translation der Bilder die Positionsänderung während des Giervorgangs berechnet (Teil 2).

Um den Einfluss der Randeffekte zu minimieren, kommen nur quadratische Bildausschnitte zum Einsatz. So verschiebt sich der Bildausschnitt bei Rotation nicht.

Abbildung 3.6 stellt den ersten Teil des Algorithmus von Konzept 1 grafisch dar. Er ist eine Erweiterung der Phasenkorrelation aus Kapitel 2.5.2. Benötigte Eingabegrößen sind blau dargestellt, die Ausgabe grün.

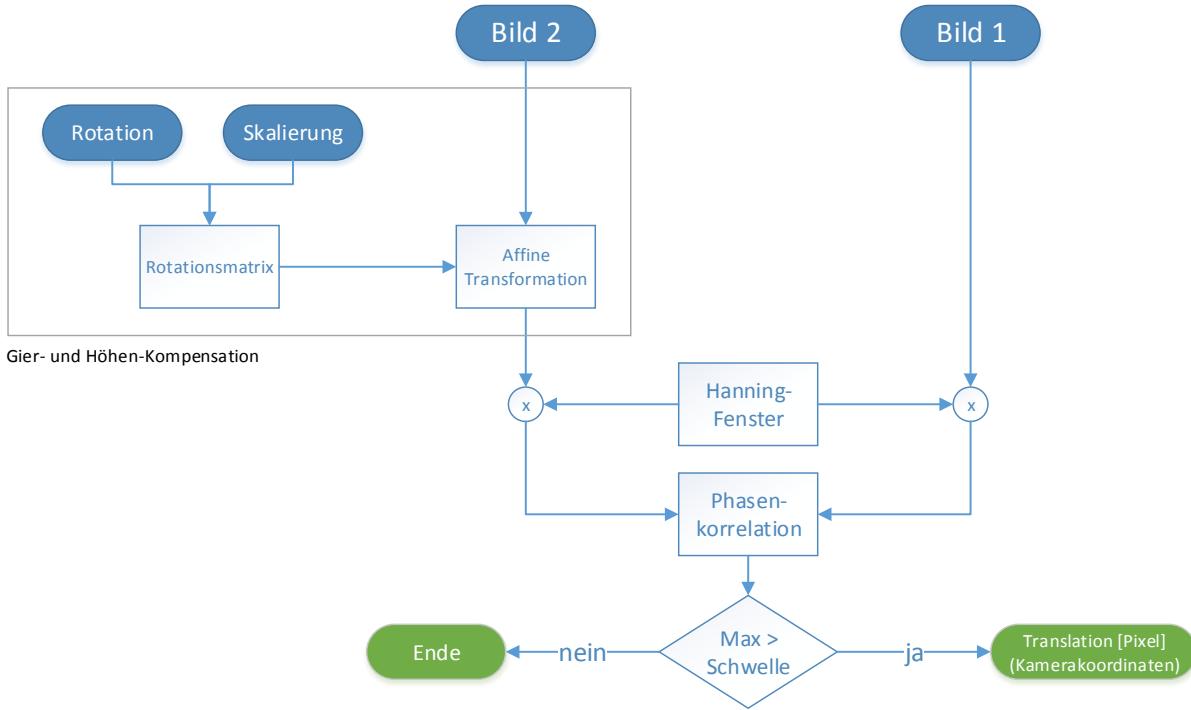


Abbildung 3.6 Algorithmus von Konzept 1 – Teil 1

Die Transformationsmatrix setzt sich aus Rotation (Formel (2.23)) und Skalierung (Formel (2.25)) zusammen. Den Skalierungsfaktor  $\alpha$  bestimmt man über die relative Höhenänderung. Sei  $h(t_0)$  die Höhe vor dem Giervorgang und  $h(t_1)$  die Höhe danach, so gilt für  $\alpha$ :

$$\alpha = \frac{h(t_1)}{h(t_0)} \quad (3.1)$$

Der Gierwinkel  $\theta$  ist die Differenz der Orientierungen  $\varphi(t_1)$  und  $\varphi(t_0)$ , wenn  $\varphi$  die eindimensionale Orientierung des Quadrocopters in der Bildebene ist. Bei einem Giervorgang um den Winkel  $\theta$  muss Bild 2 mit der Matrix  $T$  transformiert werden.

$$T = \frac{1}{\alpha} \begin{bmatrix} \cos -\theta & \sin -\theta \\ -\sin -\theta & \cos -\theta \end{bmatrix} \quad (3.2)$$

Die Phasenkorrelation arbeitet mit den Fouriertransformierten der Bilder. Um den Einfluss der Randeffekte der Transformation zu minimieren, werden die Bilder vor der Phasenkorrelation mit einem Hanning-Fenster multipliziert (siehe Kapitel 2.4.3). Der Vergleich zweier Spektren mit der Phasenkorrelation liefert ohne Hanning-Fenser oft das Ergebnis  $\langle 0,0 \rangle$  für die Verschiebung, wenn die Randeffekte den Großteil der Intensität darstellen.

Die Phasenkorrelation berechnet die Übereinstimmung der Bilder für jede mögliche Verschiebung (siehe Abbildung 2.16). Aus Formel (2.48) folgt, dass die Translation der Position des Maximalwerts (des Peak-Werts) entspricht. Der Wert des Peaks ist abhängig vom Maß der Übereinstimmung. Unterschreitet der Peak-Wert einen gewissen Schwellenwert, kann davon ausgegangen werden, dass die Registrierung fehlgeschlagen ist. Im Kapitel Kalibrierung wird gezeigt, dass ein konstanter Schwellenwert den Erfolg der Phasenkorrelation bestimmen kann. Somit verfügt der Algorithmus über eine Fehlererkennung: Peaks unter dem Schwellenwert (welche aufgrund fehlgeschlagener Registrierung meist am falschen Ort sind) werden der Verschiebung  $\langle 0,0 \rangle$  zugeordnet. Damit geht zwar die Positionsänderung während des Gierens verloren, der Quadrocopter wird jedoch nicht an eine falsche Position versetzt. Diese Fehlererkennung ist vor Allem für das dritte Konzept vonnöten, um zuverlässige Ergebnisse erzielen zu können.

Abbildung 3.7 zeigt die Berechnung der neuen Position des Quadrocopters aus den Pixelwerten der Phasenkorrelation.

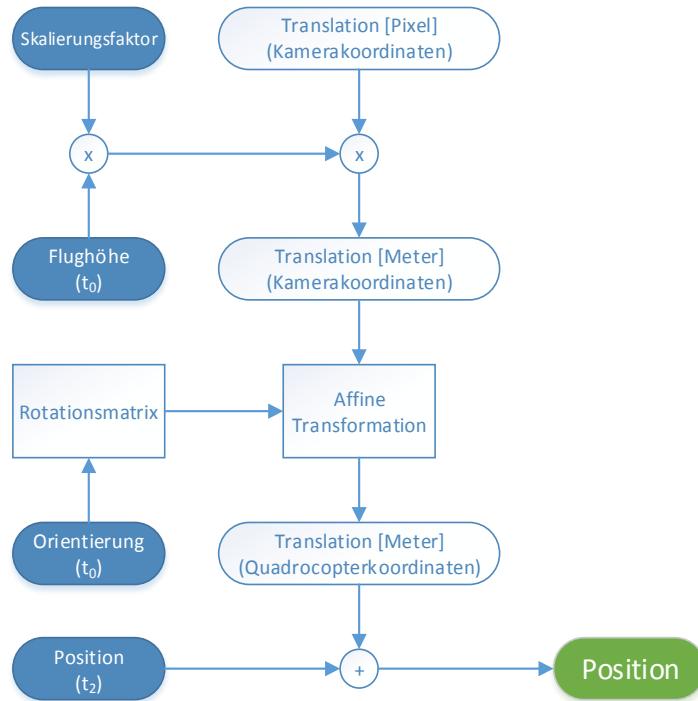


Abbildung 3.7 Algorithmus von Konzept 1 - Teil 2

Durch die affine Transformation von Teil 1 des Algorithmus bezieht sich die Translation auf Orientierung und Höhe zum Zeitpunkt  $t_0$ . Die Translation von  $\langle m_{max}, n_{max} \rangle$  Pixeln wird unter Beachtung der Höhe  $h(t_0)$  in Meter umgerechnet (siehe Kapitel 3.2).

Sei  $\langle x, y \rangle$  die Position des Quadrocopters zum Zeitpunkt  $t_2$  und  $\langle x_i, y_i \rangle$  die Translation der Bilder in Meter. Die neue Position  $\langle x', y' \rangle$  errechnet sich durch Rotation von  $\langle x_i, y_i \rangle$  um den Winkel  $\varphi(t_0)$ , um Kamera-Koordinatensystem und Quadrocopter-Koordinatensystem in Übereinstimmung zu bringen. Mit der Rotationsmatrix (2.23) gilt für die neue Position:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \cos \varphi(t_0) & \sin \varphi(t_0) \\ -\sin \varphi(t_0) & \cos \varphi(t_0) \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (3.3)$$

Der Algorithmus dieses Konzepts kann nur translatorisch verschobene Bilder korrelieren.

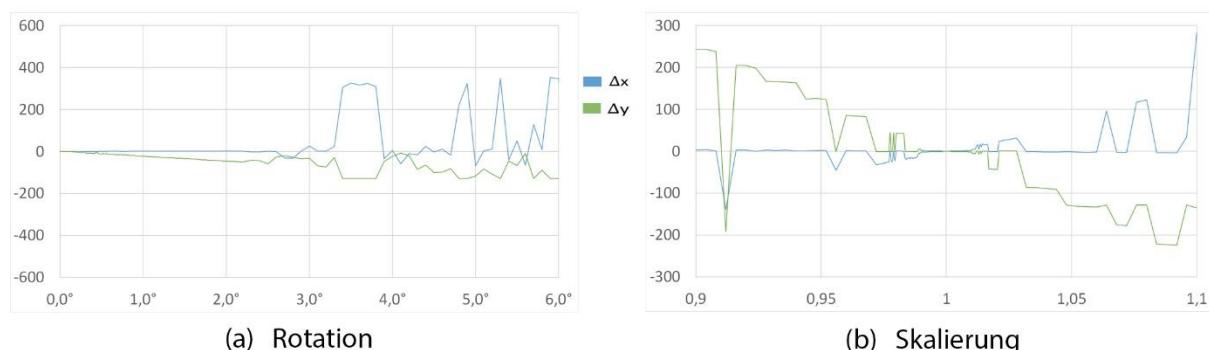


Abbildung 3.8 Fehler bei der Phasenkorrelation

Rotation und Skalierung müssen bekannt sein, um die Bilder entsprechend zu transformieren. Die beiden Größen werden von Sensoren des Quadrocopters erfasst und besitzen daher einen gewissen Fehler. Wie sehr die Korrelation von der Genauigkeit der Rotations- und Skalierungs-Werte abhängt, zeigt Abbildung 3.8. Die beiden Bilder aus Abbildung 2.16 wurden korreliert, wobei das zweite Bild um einen gewissen Winkel bzw. Skalierungsfaktor transformiert wurde.  $\Delta x$  und  $\Delta y$  stellen die Abweichung der wahren Translation zur erkannten Translation in Pixeln dar. Man sieht, dass die Phasenkorrelation schon bei Fehlern unter  $1^\circ$  bzw. Skalierungsänderungen ab 1% fehlerhafte Ergebnisse liefert. Für ein zuverlässiges Verfahren zur Positionsbestimmung muss ein anderes Konzept entwickelt werden.

### 3.4 Registrierung unter Beachtung von Translation, Rotation und Skalierung

Damit die Zuverlässigkeit der Positionsmessung nicht von der Rotations- bzw. Höhenmessung abhängt, wird der Registrierungs-Algorithmus um die Erkennung dieser Größen erweitert. Es wird ein Algorithmus von B. S. Reddy und B. N. Chatterji zur Erkennung von Rotation und Skalierung eingesetzt, der auf die Polarkoordinatendarstellung der Bilder zurückgreift. [18]

Der Algorithmus von Konzept 2 kann daher die Gesamte Pose sowie die Höhe für den Zeitpunkt  $t_1$  berechnen (Abbildung 3.9).

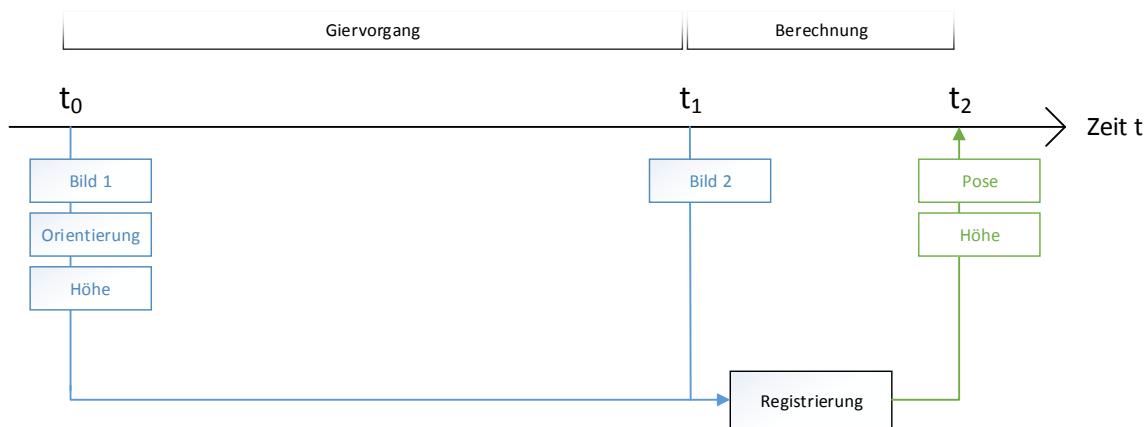


Abbildung 3.9 Schema Konzept 2

Rotations- und Skalierungserkennung werden erst einzeln betrachtet und anschließend kombiniert.

Für die rotations-invariante Bildregistrierung eignet sich die Darstellung der Bilder in Polarkoordinaten (vgl. Kapitel 2.3.2). Ein Bild in Polarkoordinaten sei gegeben durch  $i[r, \theta]$ .

Bei Rotation um den Mittelpunkt des Bildes bleibt  $r$  für alle Pixel konstant, da sich der Abstand der Pixel zur Bildmitte nicht ändert. Ist ein Bild  $i_2$  zu Bild  $i_1$  um  $\theta_0$  rotiert, gilt:

$$i_1[r, \theta] = i_2[r, \theta - \theta_0] \quad (3.4)$$

In den Polar-Transformationen der Bilder entspricht dieser Zusammenhang einer Translation in  $n$ -Richtung, da  $\theta$  auf der  $n$ -Achse aufgetragen ist. Abbildung 3.10 zeigt die Transformation für zwei zueinander rotierte Bilder. Bild (b) und (d) sind bis auf eine Verschiebung in  $n$ -Richtung identisch. Diese Verschiebung entspricht dem Winkel der Rotation, die durch Phasenkorrelation bestimmt werden kann.

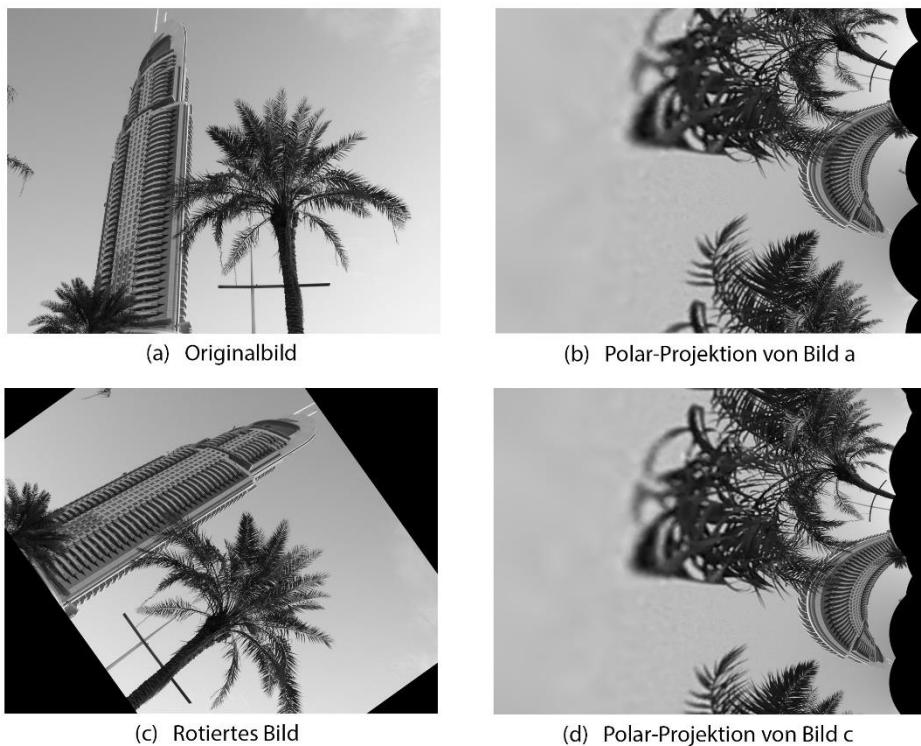


Abbildung 3.10 Polar-Transformation

Die Rotationserkennung funktioniert nur, falls das Rotationszentrum der Mittelpunkt des Bildes ist (Ursprung der Polarkoordinaten), da sich sonst die Radien der Pixel ungleichförmig ändern. Ist das Bild zusätzlich verschoben, ist auch der Rotationsmittelpunkt um den gleichen Betrag verschoben. Um verschobene rotierte Bilder zu registrieren, wird das Spektrum der Bilder betrachtet. Ist  $i_2[u, v]$  eine verschobene und rotierte affine Transformation von  $i_1[m, n]$  um  $\langle m_0, n_0 \rangle$  bzw.  $\theta_0$  in kartesischen Koordinaten, gilt:

$$u = m \cos \theta_0 + n \sin \theta_0 - m_0 \quad (3.5)$$

$$v = -m \sin \theta_0 + n \cos \theta_0 - n_0 \quad (3.6)$$

Nach Formel (2.35) gilt für die Transformierten von  $i_1$  und  $i_2$ :

$$I_1[s_m, s_n] = e^{-i2\pi(s_m m_0 + s_n n_0)} I_2[s_u, s_v] \quad (3.7)$$

Durch die Rotations-Eigenschaft der Fouriertransformation (Formel (2.37)) gilt für die transformierten Variablen die gleiche Relation wie für die ursprünglichen Variablen:

$$s_u = s_m \cos \theta_0 + s_n \sin \theta_0 \quad (3.8)$$

$$s_v = -s_m \sin \theta_0 + s_n \cos \theta_0 \quad (3.9)$$

Die Beträge (Formel (2.36)) der beiden Spektren sind also bis auf ihre Rotation zueinander identisch, wobei der Rotationsmittelpunkt die Bildmitte ist. Analog zu Formel (3.4) ist der Zusammenhang der spektralen Beträge in Polarkoordinaten gegeben durch

$$|I_1[s_r, s_\theta]| = |I_2[s_r, s_\theta - \theta_0]| \quad (3.10)$$

Mit der Phasenkorrelation der Beträge kann der Rotationswinkel  $\theta_0$  unabhängig von möglicher Translation bestimmt werden.

Zur Bestimmung der Skalierung zwischen zwei Bildern macht man sich die Eigenschaften des Logarithmus zunutze. Sei  $i_2[m, n]$  im Vergleich zu  $i_1[m, n]$  in Richtung beider Koordinatenachsen um den Faktor  $\alpha$  gestaucht oder gestreckt. Für die Polarkoordinaten-Darstellungen  $i_1[r_1, \theta_1]$  und  $i_2[r_2, \theta_2]$  der zueinander skalierten Bilder  $i_1[m, n] = i_2[\alpha m, \alpha n]$  lässt sich mit den Formeln (2.5) und (2.6) zeigen:

$$r_2 = \alpha r_1 \quad (3.11)$$

$$\theta_2 = \theta_1 \quad (3.12)$$

Die Skalierung wirkt sich in Polarkoordinaten also nur auf den Radius aus, der Winkel der Pixel bleibt konstant. Aus dem Zusammenhang

$$i_1[r, \theta] = i_2[\alpha r, \theta] \quad (3.13)$$

folgt für die Fouriertransformierten nach Formel (2.38) in Polarkoordinaten:

$$I_1[s_r, s_\theta] = \frac{1}{\alpha} I_2\left[\frac{s_r}{\alpha}, s_\theta\right] \quad (3.14)$$

Vernachlässigt man den Proportionalitätsfaktor  $\alpha^{-1}$  und teilt die  $s_r$ -Achse logarithmisch ein, gilt:

$$I_1[\log s_r, s_\theta] = I_2[\log s_r - \log \alpha, s_\theta] \quad (3.15)$$

Substituiert man ferner die Variablen mit  $x = \log s_r$  und  $c = \log \alpha$ , folgt eine Formel, mit der durch Phasenkorrelation auf den Skalierungsfaktor  $\alpha = e^c$  geschlossen werden kann.

$$I_1[x, s_\theta] = I_2[x - c, s_\theta] \quad (3.16)$$

Bringt man die Verfahren für Skalierungs- und Rotations-Erkennung in Verbindung, können beliebig (in der  $mn$ -Ebene) transformierte Bilder registriert werden.

Die logarithmische Einteilung der  $r$ -Achse der Polarkoordinaten wird Log-Polar-Transformation (LPT) genannt. Die  $r$ -Achse ist für das Verfahren der Rotationserkennung nicht von Bedeutung, weswegen die Rotation auch mit der LPT berechnet werden kann. Transformiert man beide Bilder in das Fourierspektrum und berechnet die Log-Polar-Transformation der Beträge, können Rotation und Skalierung berechnet werden. Für rotierte und skalierte Bilder folgt aus den Formeln (3.10) und (3.16) für die Beträge der Spektralfunktionen in Log-Polar-Koordinaten:

$$|I_1[x, s_\theta]| = |I_2[x - c, s_\theta - \theta_0]| \quad (3.17)$$

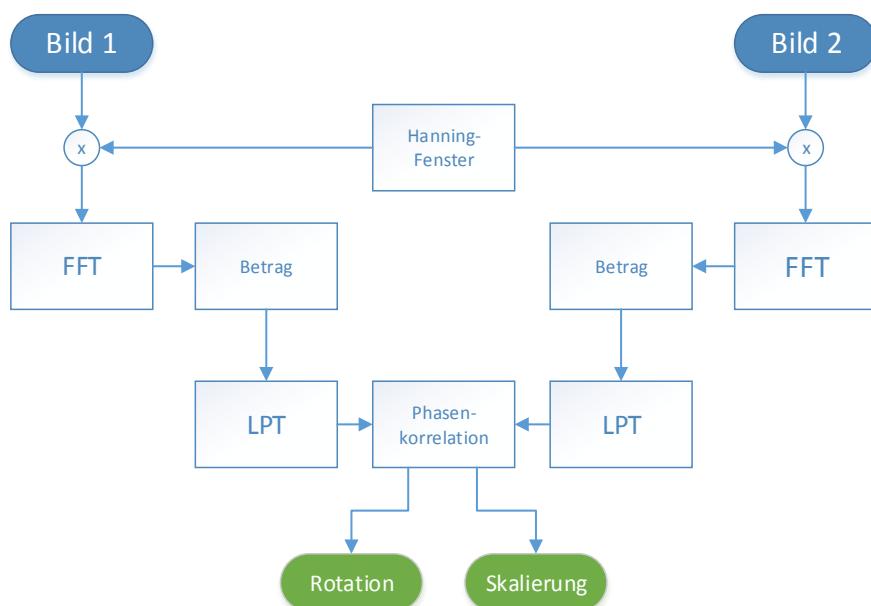


Abbildung 3.11 Algorithmus von B. S. Reddy und B. N. Chatterji

Rotation und Skalierung können durch Formel (3.17) mit nur einer Phasenkorrelation bestimmt werden (siehe Abbildung 3.11). Die Verschiebung  $c$  der  $x$ -Koordinate lässt auf die Skalierung  $\alpha = e^c$  schließen. Die Verschiebung der  $s_\theta$ -Koordinate entspricht dem Rotationswinkel  $\theta_0$ . Vor der Fouriertransformation werden die Bilder wieder mit einem Hanning-Fenster multipliziert (siehe Kapitel 3.3). Die Autoren des Algorithmus wenden außerdem einen Hochpass-Filter auf die Log-Polar-Transformierten an. In Kapitel 4.2.4 wird die Notwendigkeit dieses Filters gezeigt.

Bis auf die Berechnung von Rotation und Skalierung sind Konzept 2 und Konzept 1 identisch. Mit diesen Größen können Flughöhe sowie Orientierung für den Zeitpunkt  $t_1$  bestimmt werden, da Rotation und Skalierung die Änderung der zum Zeitpunkt  $t_0$  bekannten Werte Orientierung und Flughöhe sind.

Die Fehlererkennung mittels eines konstanten Schwellenwerts ist durch die doppelte Phasenkorrelation bei diesem Konzept sehr verlässlich. Schlägt die erste Phasenkorrelation fehl oder greift die Fehlererkennung nicht, ist das zweite Bild in Bezug auf das Erste falsch rotiert und skaliert. Dadurch liefert die zweite Phasenkorrelation einen sehr niedrigen Peak-Wert, da das Cross-Power-Spectrum keine hohe Übereinstimmung finden kann. Gelingt die erste Phasenkorrelation, stimmen Rotation und Skalierung der Bilder exakt überein. Diese Übereinstimmung hat einen hohen Peak-Wert der zweiten Phasenkorrelation zur Folge.

Die Anwendung zeigte, dass sich dieses Konzept nicht im realen Flugbetrieb eignet. Der Erfolg der Registrierung hängt maßgeblich von der Größe des übereinstimmenden Anteils der Bilder ab. Je nach zeitlicher Dauer des Giervorgangs ist die Positionsänderung bei Konzept 2 mitunter so groß, dass die Bilder nicht mehr genügend überlappen.

## 3.5 Kontinuierliche Registrierung

Registriert man kontinuierlich je zwei aufeinanderfolgende Bilder, wird die Positionsänderung zwischen zwei Frames minimiert. Dadurch arbeitet die Positionserkennung wesentlich zuverlässiger. Bei der Registrierung kommt der Algorithmus aus Konzept 2 zum Einsatz. Sobald zwei Bilder registriert sind, wird der aktuelle Kamera-Frame mit dem zuletzt verwendeten Frame registriert. Der folgende Pseudocode beschreibt den Algorithmus von Konzept 3:

```

Bild 1 << Kamera
Bild 2 << Kamera
Pose = Startpose
Höhe = Starthöhe
while (Positionsbestimmung aktiv)
    [Posenänderung, Höhenänderung] = Registrierung(Bild1, Bild 2)
    Pose += Posenänderung

```

```

Höhe += Höhenänderung
Bild 1 = Bild 2
Bild 2 << Kamera
end while

```

In dieser Form kann die Positionserkennung mittels Bildregistrierung nicht nur zum Gieren, sondern für den gesamten Flug eingesetzt werden. Abbildung 3.12 veranschaulicht Konzept 3.

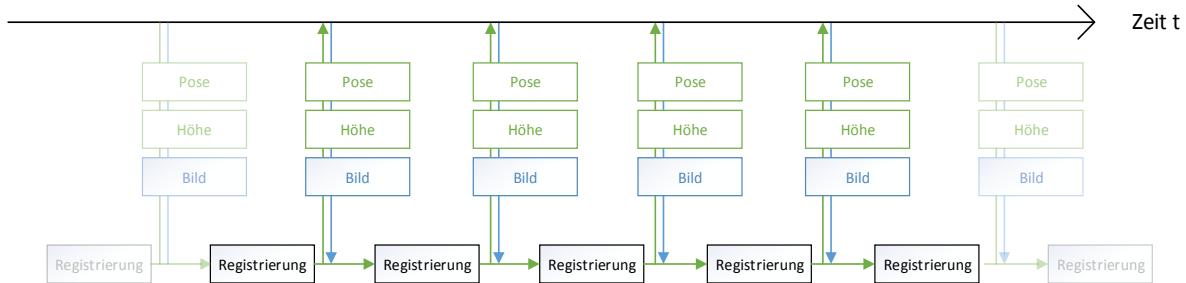


Abbildung 3.12 Schema Konzept 3

Höhe und Orientierung können iterativ bestimmt werden und besitzen daher einen Kopplungsfehler. Da die Sensoren des Quadrocopters Höhe und Orientierung absolut messen, liefern diese auf lange Sicht zuverlässigere Werte. Die Registrierung kann mit diesen oder den selbstgewonnenen Daten durchgeführt werden. Das Konzept wird daher mit beiden Möglichkeiten evaluiert.

Die Bereits gezeigte Fehlererkennung ist wichtig für die Zuverlässigkeit der kontinuierlichen Registrierung. Aufgrund der zeitlichen Nähe zweier Frames ist die erwartete Translation sehr klein im Vergleich zur gesamten Bildgröße. Schlägt bei einzelnen Frames die Registrierung fehl, geht nur eine kleine Positionsänderung verloren. Ohne die Fehlererkennung könnte jedoch eine einzelne Registrierung einen großen Positionsfehler verursachen, da sich der Peak-Wert bei fehlgeschlagener Registrierung im gesamten Bildbereich befinden kann.

In dieser Form bietet die Bildregistrierung ein flugfähiges System zur Positionsbestimmung. Es kann parallel zum optischen Fluss eingesetzt werden oder ihn während der Giervorgänge ersetzen. In der Evaluierung wird auf die Kombination von optischem Fluss und Registrierung eingegangen. Auf dem Quadrocopter wird dieses Konzept implementiert, da sich die beiden vorhergehenden Konzepte aufgrund des großen zeitlichen Abstands zweier Bilder nicht für die Anwendung im Flug eignen.

# 4 Implementierung

Dieses Kapitel beschreibt die Implementierung von Konzept 3 auf dem Quadrocopter *AQopter18*. Der Registrierungs-Algorithmus wird auf das Zielsystem angepasst und unter Beachtung der Anforderungen als Positionsbestimmungssystem eingesetzt.

## 4.1 Anforderungen

Je schneller und genauer die Positionsbestimmung arbeitet, desto akkurate können Positionen angeflogen und gehalten werden. Der Algorithmus zur Registrierung muss also an die Rechenleistung des Quadrocopters angepasst werden, da Positionsinformationen erst nach einem Durchlauf der Registrierung vorliegen (vgl. Abbildung 3.12). Je kürzer der Zeitraum ist, desto schneller kann die Regelung auf eine Positionsabweichung reagieren.

Der Algorithmus sollte weiterhin im statischen und dynamischen Regelverhalten einsetzbar sein. Damit der Quadrocopter eine bestimmte Position auf längere Zeit halten kann, muss der Drift minimiert werden. Weiterhin ist der Algorithmus an hohe Fluggeschwindigkeiten anzupassen. Die Überlappung zweier Bilder muss bei Höchstgeschwindigkeit ausreichend groß sein, um eine erfolgreiche Registrierung zu gewährleisten.

## 4.2 Positionsbestimmungsalgorithmus

Der zugrundeliegende Algorithmus aus Konzept 3 berechnet kontinuierlich die Transformation zweier aufeinanderfolgender Bilder (Abbildung 3.12). Alle a priori-Informationen des Algorithmus werden bei der Initialisierung berechnet, um eine möglichst hohe Framerate zu erzielen. Die Registrierung greift auf die *OpenCV*-Bildverarbeitungsbibliothek zurück. [19] Abbildung 4.1 fasst den gesamten Algorithmus zur Positionsbestimmung zusammen.

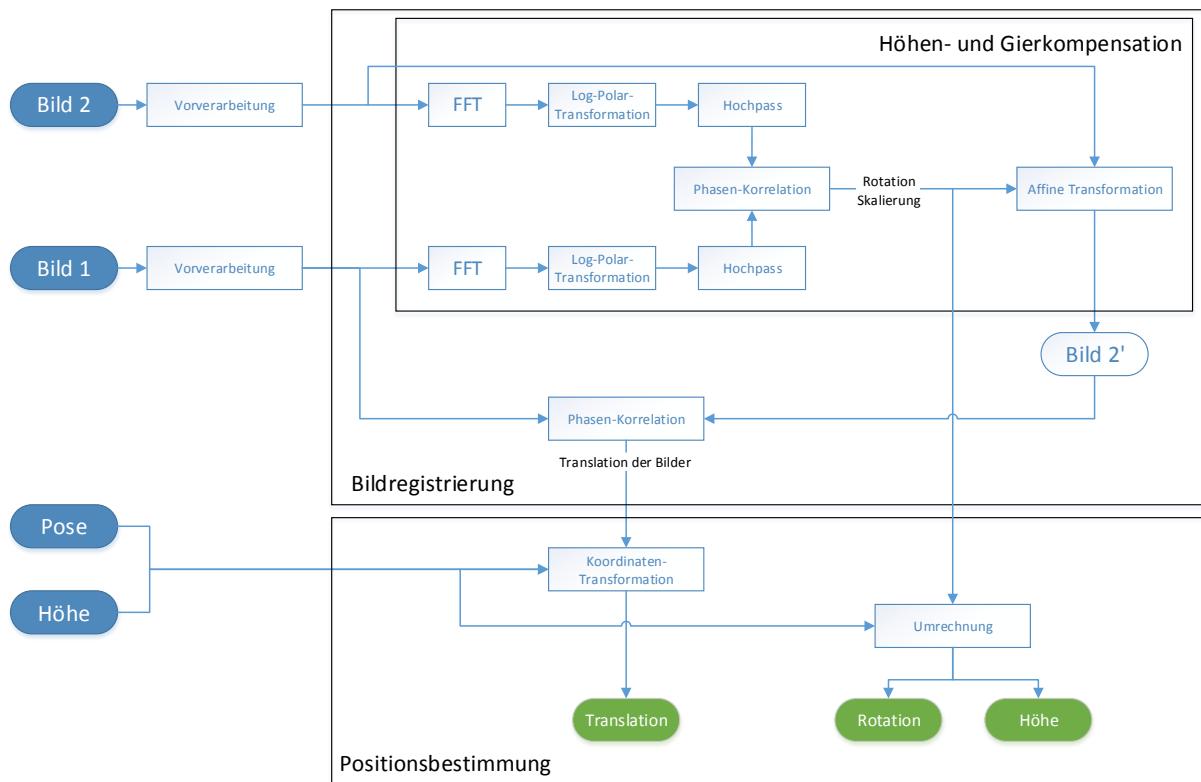


Abbildung 4.1 Positionsbestimmungsalgorithmus

Nach der Vorverarbeitung der Bilder wird die Höhen- und Gierkompensation durchgeführt. Sind Skalierung und Rotation der Bilder mit der ersten Phasenkorrelation (vgl. Abbildung 2.15) berechnet, wird Bild 2 mit den beiden Werten affin Transformiert. Bild 1 und Bild 2' unterscheiden sich nur durch Translation. Diese Translation wird mit der zweiten Phasenkorrelation bestimmt. Damit ist die Bildregistrierung abgeschlossen und Translation, Rotation und Skalierung der Eingangsbilder sind bekannt. Mit der Pose und der Flughöhe des Quadrocopters kann aus den berechneten Daten die Position bestimmt werden. Die Koordinaten-Transformation zur Positionsbestimmung erfolgt nach Abbildung 3.7. Im Folgenden werden die Implementierungsdetails der einzelnen Blöcke ausführlich erläutert.

#### 4.2.1 Vorverarbeitung der Bilder

Die angeschlossene Kamera liefert 8-bit RGB-Bilder (Wertebereich  $[0,1, \dots, 255]$ ). Die Bilder werden vor der Registrierung in Graustufenbilder umgewandelt. Der Grauwert ergibt sich aus dem Mittelwert der drei RGB-Werte.

Je breiter das Grau-Spektrum der Bilder ist, desto besser arbeitet die Registrierung. Mit einem Histogrammausgleich werden häufig vorkommende Grauwertbereiche auseinandergezogen, wodurch der globale Kontrast im Bild erhöht wird und die Fouriertransformation des Bildes ein stärker ausgeprägtes Frequenzspektrum besitzt. Die Histogrammfunktion eines Bildes

beschreibt die relative Häufigkeit jeder Graustufe. Abbildung 4.2 zeigt ein Kamerabild mit zugehöriger Histogrammfunktion. Auf der  $x$ -Achse sind die 256 Graustufen (von schwarz bis weiß) und auf der  $y$ -Achse die relativen Häufigkeiten aufgetragen.

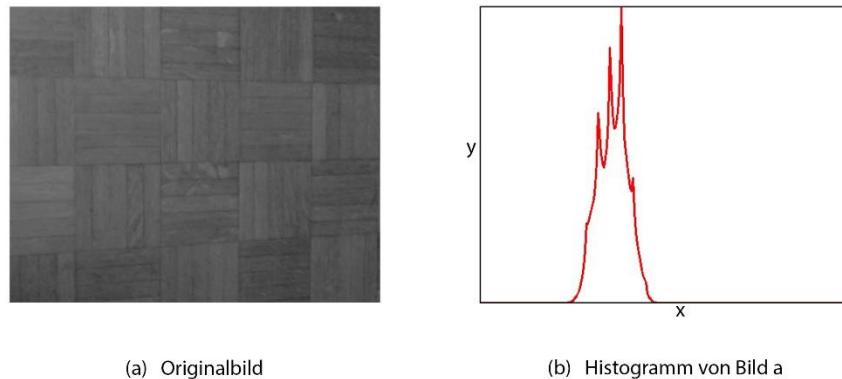


Abbildung 4.2 Kamerabild mit Histogramm

In Abbildung 4.2 (a) kommen nur dunkle Grautöne vor, weswegen die relativen Häufigkeiten in der rechten Hälfte (mittleres grau bis weiß) und am linken Rand (schwarz bis dunkelgrau) des Histogramms gleich Null sind. Wendet man einen Histogrammausgleich auf Bild (a) an, werden die Grauwerte auf das gesamte Spektrum verteilt. Der Histogrammausgleich erfolgt mit der *OpenCV*-Funktion `cv::equalizeHist()`. Die Funktion verteilt alle Grauwerte prozentual nach der kumulativen Histogrammfunktion des Bildes. [19]

Das Ergebnis des Histogrammausgleichs zeigt Abbildung 4.3.

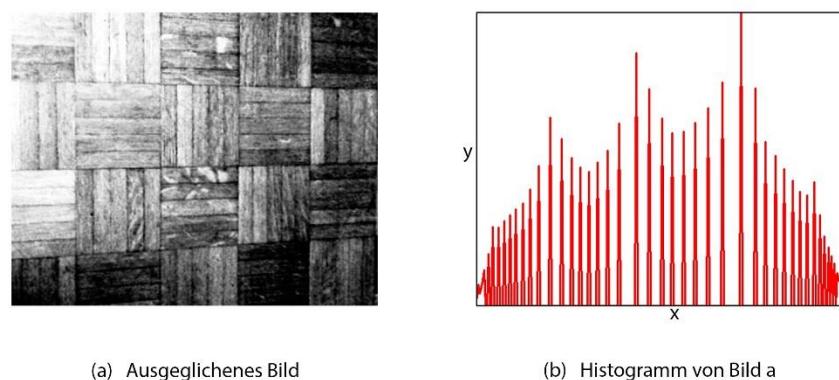


Abbildung 4.3 Ausgeglichenes Kamerabild mit Histogramm

Die Histogrammfunktion ist jetzt über das gesamte Grauwertspektrum verteilt, wobei die grundlegende Form des Ursprungshistogramms nicht verändert wurde. Zu erkennen ist dies an den drei Peaks des Histogramms, welche lediglich weiter voneinander entfernt wurden. Durch den Ausgleich enthält das Bild alle Töne von schwarz bis weiß. Der stark erhöhte Kontrast führt zu deutlich besseren Ergebnissen beim Korrelieren der Spektren.

Außerdem werden Beleuchtungsunterschiede der beiden zu korrelierenden Bilder eliminiert. Die Kamera passt die Belichtungszeit automatisch der Helligkeit der Szene an. Die zwei Bilder Abbildung 4.4 (a) und (b) wurden leicht versetzt aufgenommen. Durch die automatische Helligkeitsanpassung der Kamera sind jedoch keine übereinstimmenden Bereiche zu erkennen. Nach einem Histogrammausgleich (Bilder (c) und (d)) können sie miteinander korreliert werden.

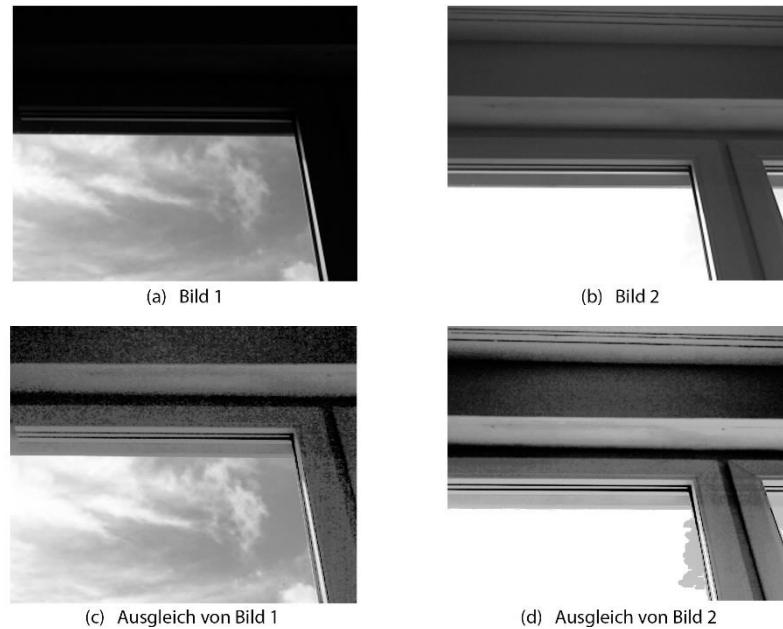


Abbildung 4.4 Histogrammausgleich bei Beleuchtungsänderung

Vor der Weiterverarbeitung werden die Bilder in kontinuierliche Grauwertbilder (Wertebereich (0, ..., 1)) umgewandelt (vgl. Kapitel 2.3.1).

## 4.2.2 Fouriertransformation

Vor jeder Fouriertransformation werden die Bilder mit einem Hanning-Fenster multipliziert (vgl. Kapitel 2.4.3 und 3.3). Dieses wird bei der Initialisierung einmalig nach Formel (2.41) mit einer *OpenCV*-Funktion erstellt und als 2D-Array für die spätere Verwendung gespeichert. Bei der Anwendung des Hanning-Fensters im Registrierungsalgorithmus müssen die Bilder nur mit dem gespeicherten Array multipliziert werden. Dadurch benötigt man keine rechenintensiven Kosinus-Funktionen während der Registrierung und der Algorithmus wird schneller.

Für jede Bildregistrierung werden acht Fouriertransformationen ausgeführt. Diese stellen den zeitaufwändigsten Teil der Registrierung dar. Um eine hohe Framerate zu erzielen, wird ein möglichst schneller Algorithmus zur Berechnung der Fouriertransformation implementiert.

Die *OpenCV*-Bibliothek stellt nur eine Funktion zur direkten Berechnung der Transformation bereit. Sie enthält keine Implementierung von FFT-Algorithmen.

Mit der schnellen Fouriertransformation (FFT) kann die Berechnungszeit signifikant verkürzt werden (vgl. Kapitel 2.4.4). Für ein Bild der Größe  $256 \times 256$  Pixel benötigt die FFT ca. 0,02% der Berechnungen der direkten Fouriertransformation und ist damit um ein Vielfaches schneller.

Zur Berechnung der FFT wird die *FFTW*-Bibliothek<sup>12</sup> in die Kontrollsoftware integriert. Diese freie Bibliothek greift auf eine große Anzahl von FFT-Algorithmen zurück, um für beliebige Daten und Parameter stets eine der schnellsten Implementierungen der FFT zu liefern. Dazu werden sogenannte Pläne erstellt, die mit festen Parametern initialisiert werden. Auf einem Plan können dann beliebig viele Fouriertransformationen mit den gewählten Parametern ausgeführt werden. Bei der Initialisierung werden alle für diese Parameter unveränderlichen Daten der Transformation im Voraus berechnet und der schnellste Algorithmus ausgewählt. Je mehr Zeit der Bibliothek zur Initialisierung eines Plans gegeben wird, desto schneller wird jede Transformation nach diesem Plan ausgeführt. Dazu stehen verschiedene Modi bereit. Mit der Standardeinstellung dauert die Initialisierung wenige Sekunden. Im Optionsmenü des Positionsbestimmungs-Moduls kann die Option *FFTW Patient* ausgewählt werden. Wenn aktiviert, dauert die Initialisierung bis zu drei Minuten. Der Plan wird mit einer sehr großen Anzahl der verfügbaren Algorithmen durchgerechnet, um den Schnellsten zu finden. Jede Registrierung wird dann ca. 13% schneller ausgeführt. Ob dies von Vorteil ist, wird in Kapitel 5 evaluiert.

Es besteht die Möglichkeit, generierte Pläne zu speichern. Hat man die optimalen Einstellungen für den Registrierungsalgorithmus gefunden, kann so der zugehörige Plan für die spätere Verwendung gespeichert werden, um die vergleichsweise lange Initialisierungszeit auf ein Minimum zu reduzieren. [20]

### 4.2.3 Phasenkorrelation

Die *FFTW* ist für beliebige komplexe Eingabesequenzen konzipiert. Da die Bilder aus realen Datenpunkten bestehen, ist das Ergebnis der Fouriertransformation mit dem komplex Konjugierten symmetrisch um den Mittelpunkt der Daten (siehe Formel (2.34)). Für die Phasenkorrelation wird daher nur eine Hälfte der Frequenzspektren der Bilder berechnet, da das Cross-Power-Spectrum zweier realer Eingabesequenzen die gleiche Symmetrie aufweist.

---

<sup>12</sup> Fastest Fourier Transformation in the West

Diese sogenannte real-zu-komplex-Transformation wird über die Eigenschaften des *FFTW*-Plans definiert. Es wird also nur eine Hälfte des CPS berechnet.

Transformiert man ein komplex konjugiertes Spektrum zurück in den Bildbereich, entsteht wieder ein reales Bild mit den Maßen der Eingabebilder. Die *FFTW* nimmt die Eingabesequenz dabei als die Hälfte einer komplex konjugierten Sequenz an. Dies ist nötig, da nur die eine Hälfte des komplex konjugierten CPS berechnet wurde.

Diese Nutzung der Symmetrie halbiert die Berechnungszeit des Cross-Power-Spectrums, da sowohl die Fouriertransformationen als auch das CPS selbst nur halb so viele Rechenoperationen benötigen.

Formel (2.48) berechnet die Pixel-Position des Peak-Werts der Phasenkorrelation. Die Genauigkeit der Translationsbestimmung ist durch die gewählte Auflösung der Kamera begrenzt. Bei der kontinuierlichen Registrierung akkumuliert sich dies zu einem nicht zu vernachlässigenden Fehler.

Zur Präzisierung der Translationsmessung wird die Peak-Position mit Sub-Pixel-Genauigkeit ermittelt. Die Dokumentation der *OpenCV*-Bibliothek beschreibt ein Verfahren zur Bestimmung des gewichteten Schwerpunkts einer Peak-Position, welches sich gut eignet. [19]

Es wird ein  $5 \times 5$ -Pixel-Fenster um den Peak-Wert betrachtet. Jedes der 25 Pixel geht unter Berücksichtigung der Intensität des Pixels in die Ortsbestimmung ein. Sei  $\binom{m_p}{n_p}$  die pixelgenaue Peak-Position. Der gewichtete Schwerpunkt liefert für die sub-Pixel-genaue Peak-Position  $\binom{m_s}{n_s}$ :

$$\binom{m_s}{n_s} = \frac{\sum_{n=-2}^2 \sum_{m=-2}^2 \binom{m_p+m}{n_p+n} i[m_p+m, n_p+n]}{\sum_{n=-2}^2 \sum_{m=-2}^2 i[m_p+m, n_p+n]} \quad (4.1)$$

Die Größe von 25 Pixeln hat sich in der Praxis als optimale Wahl herausgestellt, da nur die unmittelbare Nachbarschaft der Peak-Position für den gewichteten Schwerpunkt berücksichtigt wird. Bei größeren Fenstern kann die ermittelte Position durch Neben-Peaks und die ungleichförmige Korrelationsoberfläche verfälscht werden.

#### 4.2.4 Hochpass-Filter

Der Phasenkorrelation zur Bestimmung von Rotation und Skalierung geht ein Hochpass-Filter voraus (vgl. Kapitel 3.4). Die Notwendigkeit dieses Filters beruht auf der Anwendung der Log-Polar-Transformation. Die Log-Polar-Transformation bildet die niedrigen Frequenzen auf einen wichtigen Teil des transformierten Bildes ab. Rotation und Skalierung werden

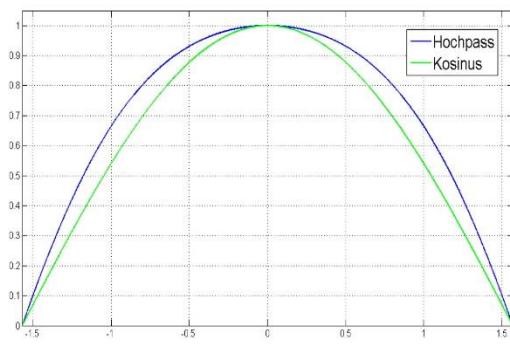
hauptsächlich an der Übereinstimmung der höheren Frequenzen erkannt. Die niedrigen Frequenzen sind meist verzerrt und stimmen in den Transformationen kaum überein. Mit einem Hochpassfilter wird verhindert, dass die niedrigen Frequenzen die Phasenkorrelation verfälschen. Der von B. S. Reddy und B. N. Chatterji vorgeschlagene Hochpass-Verstärkungs-Filter minimiert den Einfluss der niedrigen Frequenzen und verstärkt gleichzeitig die höchsten Frequenzen. Die Zuverlässigkeit der Registrierung konnte so enorm gesteigert werden. [18]

Der Filter ist charakterisiert durch seine Impulsantwort:

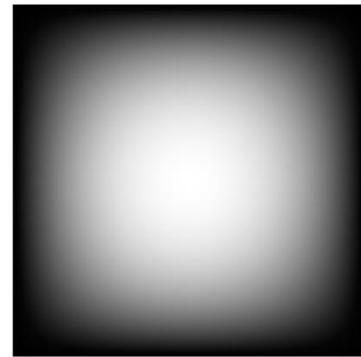
$$H[s_1, s_2] = X[s_1, s_2](1.5 - 0.5X[s_1, s_2]) \quad (4.2)$$

mit

$$X[s_1, s_2] = \cos s_1 \cos s_2, \quad \frac{-0.5}{\pi} \leq s_1, s_2 \leq \frac{0.5}{\pi} \quad (4.3)$$



(a) 1D-Hochpass und Kosinus-Funktion



(b) 2D-Hochpass

Abbildung 4.5 Hochpassfilter

Abbildung 4.5 stellt den Filter sowie seine 1D-Projektion grafisch dar. Bild (a) zeigt zudem einen Vergleich zur Kosinus-Funktion. Ein reiner Kosinus-Filter weist ein ähnliches Verhalten auf, die mittleren Frequenzen werden aber etwas schwächer berücksichtigt. Der Hochpass-Filter nach Formel (4.2) ist bei leicht verschwommenen Bildern robuster, da dann die Intensität der höchsten Frequenzen gegen null geht und die Registrierung sich hauptsächlich an den mittleren bis hohen Frequenzen orientiert.

Analog zum Hanning-Fenster wird auch der Hochpass-Filter bei der Initialisierung des Registrierungsalgorithmus einmalig angelegt und für die spätere Anwendung abgespeichert, um Rechenzeit zu sparen.

#### 4.2.5 Log-Polar-Transformation

Abbildung 4.6 zeigt die Log-Polar-Transformation (vgl. Kapitel 3.4 -  $\langle m_c, n_c \rangle$  ist die Bildmitte) von  $\langle m, n \rangle$  zu  $\langle e^r, \theta \rangle$  nach den Formeln:

$$e^r = \sqrt{(m - m_c)^2 + (n - n_c)^2} \quad (4.4)$$

$$\theta = \tan^{-1} \frac{n - n_c}{m - m_c} \quad (4.5)$$

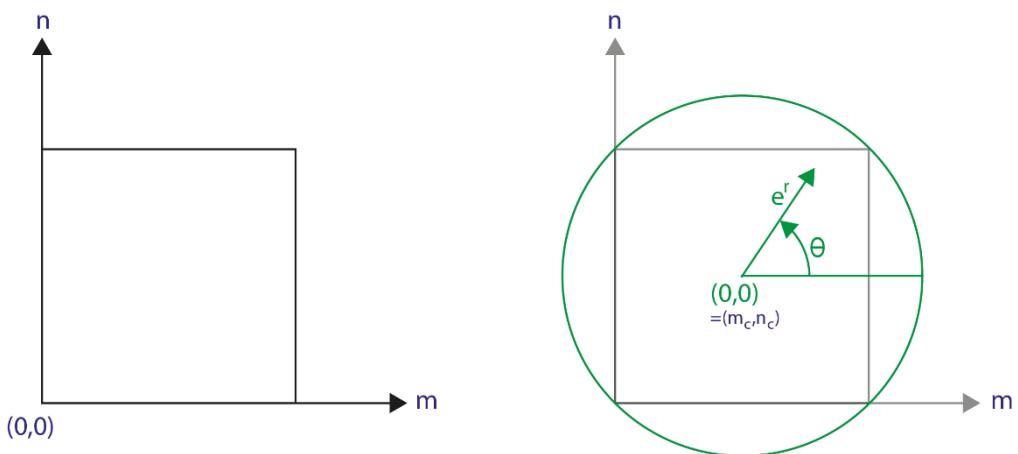


Abbildung 4.6 Kartesische und Log-Polar-Koordinaten

Der Maximalwert für  $r$  ergibt sich aus dem Umkreisradius des Ursprungsbilds:

$$\max(r) = \ln \sqrt{m_c^2 + n_c^2} \quad (4.6)$$

Der Umkreis berücksichtigt alle Pixel des Ursprungsbilds und besitzt gleichzeitig ungültige Bereiche außerhalb des Bilds. Diese werden als null angenommen.

Die Transformation der Pixel kann nicht eins-zu-eins erfolgen, da die Koordinatensysteme ungleichförmig sind. Jede  $\langle e^r, \theta \rangle$ -Koordinate im Log-Polar-Bild entspricht einer Position  $\langle x, y \rangle$  im Ursprungsbild nach der Vorschrift:

$$x = e^r \cos \theta + m_c \quad (4.7)$$

$$y = e^r \sin \theta + n_c \quad (4.8)$$

Diese  $\langle x, y \rangle$ -Positionen sind reell-wertig und können zwischen den Pixeln des Ursprungsbilds liegen. Der Intensitätswert  $i[x, y]$  an der Position  $\langle x, y \rangle$  wird mit bilinearer Interpolation ermittelt. Eine andere Variante ist die weitaus rechenintensivere bikubische Interpolation. Diese liefert zwar eine etwas genauere Abbildung, zeigte in der Praxis jedoch keine Vorteile für die Bildregistrierung.

Die bilineare Interpolation ist eine Verkettung zweier linearer Interpolationen in beide Raumrichtungen. Seien  $i_{11}[x_1, y_1]$ ,  $i_{12}[x_1, y_2]$ ,  $i_{21}[x_2, y_1]$  und  $i_{22}[x_2, y_2]$  die Intensitätswerte der vier Pixel, die die Position  $\langle x, y \rangle$  einschließen. Die Interpolation in  $x$ -Richtung für beide  $y$ -Koordinaten liefert:

$$i[x, y_1] = \frac{x_2 - x}{x_2 - x_1} i_{11} + \frac{x - x_1}{x_2 - x_1} i_{21} \quad (4.9)$$

$$i[x, y_2] = \frac{x_2 - x}{x_2 - x_1} i_{12} + \frac{x - x_1}{x_2 - x_1} i_{22} \quad (4.10)$$

Die Intensität  $i[x, y]$  ergibt sich aus linearer Interpolation in  $y$ -Richtung:

$$i[x, y] = \frac{y_2 - y}{y_2 - y_1} i[x, y_1] + \frac{y - y_1}{y_2 - y_1} i[x, y_2] \quad (4.11)$$

Diese bilineare Interpolation wird für jede Log-Polar-Koordinate ausgeführt. Folgender Pseudocode beschreibt die Log-Polar-Transformation eines  $N \times N$ -Bildes.

```

delta_theta = 2*PI/N
log_basis = (N*sqrt(0.5))^(1/N)
for i = 0 to N-1
    for j = 0 to N-1
        radius = log_basis^i
        theta = j*delta_theta
        x = radius*cos(theta)+N/2
        y = radius*sin(theta)+N/2
        log-polar-bild[i, j] = bilineare_interpolation(bild[x, y])
    end for
end for

```

Das Log-Polar-Bild besitzt die selbe Größe  $N \times N$ . Als Basis  $b$  für den Logarithmus wird nicht die eulersche Zahl, sondern die Größe  $b = (N\sqrt{0.5})^{1/N}$  verwendet. Dadurch wird das Bild komplett in den Log-Polar-Bereich abgebildet, denn mit dem maximalen  $i = N - 1$  (vgl. Pseudocode) gilt für  $r$ :

$$\max_{i=N-1} (r) = b^{N-1} \approx \sqrt{(0.5N)^2 + (0.5N)^2} \quad (4.12)$$

Der Maximalwert von  $r$  entspricht dem Umkreisradius des Ursprungsbilds (vgl. Abbildung 4.6) und wird im Log-Polar-Bild in die Spalte  $i = N - 1$  abgebildet. Damit ist die Fläche des Log-

Polar-Bilds optimal genutzt. Die schwarzen Flächen am rechten Rand des Log-Polar-Bilds (hohe  $r$ -Werte), welche aus den ungültigen Bereichen des Umkreises entstehen, sind mit dem gewählten Wert für  $b$  minimal (siehe Abbildung 3.10).

*OpenCV* stellt für die Log-Polar-Transformation die Funktion `cv::cvLogPolar(m,...)` mit dem logarithmischen Zusammenhang

$$e^r = m \ln \sqrt{(0.5N)^2 + (0.5N)^2} \quad (4.13)$$

bereit. Der Wert für  $m$  wird so gewählt, dass Formel (4.12) erfüllt ist:

$$\log_b \max(r) = \frac{1}{\ln b} \ln \max(r) \rightarrow m = \frac{1}{\ln b} = \frac{N}{\ln(N\sqrt{0.5})} \quad (4.14)$$

[19]

## 4.3 Kontrollsoftware

Das System der Positionsbestimmung mittels Bildregistrierung ist in die bestehende Kontrollsoftware *I8QuatPlay* des Quadrocopters zu integrieren. Dazu zählt die Anzeige der Kamerabilder sowie die Aufbereitung aller aus der Registrierung gewonnenen Daten, um das System evaluieren zu können.

### 4.3.1 Bestehendes System

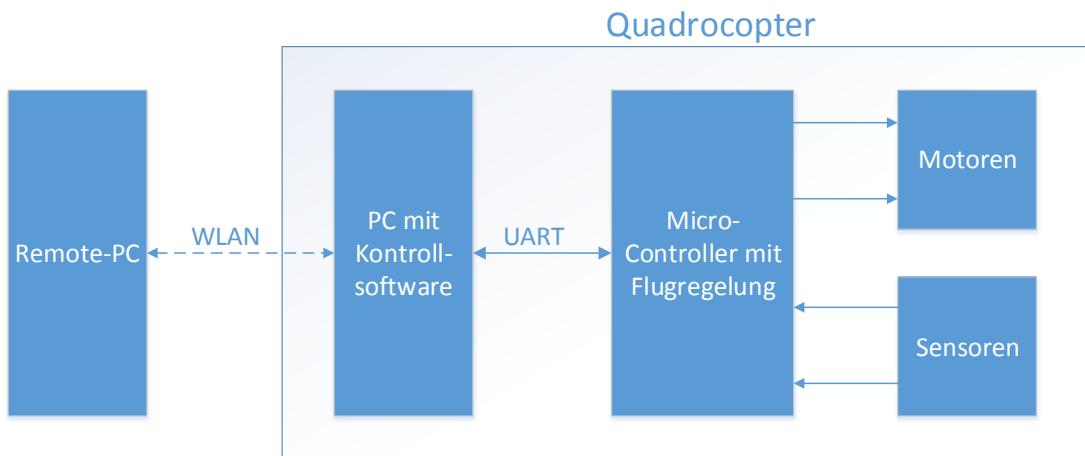


Abbildung 4.7 Schematische Darstellung des Quadrocopters

Der Quadrocopter verfügt über einen Micro-Controller vom Typ *AVR-32UC3A* und einen *Commell LP-180 x86*-Computer (1,65 GHz Dual-Core) mit Windows 7, auf dem die Kontrollsoftware *I8QuatPlay* läuft (Abbildung 4.7). [21] [22] Der Micro-Controller ist mit den Sensoren und Aktuatoren des Quadrocopters verbunden und ist für eine stabile

Fluglageregelung sowie Flugführung verantwortlich. Er ist über eine bidirektionale serielle Schnittstelle (UART<sup>13</sup>) mit dem PC verbunden. So können Sensorwerte und Motorstellwerte in der Kontrollsoftware angezeigt werden und Stellgrößen vom PC an die Flugregelung übermittelt werden.

Die modulbasierte Kontrollsoftware besitzt eine grafische Benutzeroberfläche und ist für alle nicht flugrelevanten Aufgaben zuständig. Sie basiert auf dem *QT*-Framework und ist in C++ geschrieben. Über eine WLAN-Verbindung zu einem Remote-PC kann die Oberfläche der Kontrollsoftware während des Fluges auf einem stationären PC angezeigt werden. Im Modul *Debug* der Kontrollsoftware können alle verfügbaren Daten des Quadrocopters mit zeitlichem Verlauf angezeigt und verglichen werden.

### 4.3.2 Positionsbestimmung mittels Bildregistrierung

Die Positionsbestimmung mittels Bildregistrierung wird als Teilmodul der Kontrollsoftware *I8QuatPlay* entwickelt. Dazu wird ein neuer Reiter *Image Registration* im Hauptfenster der Software angelegt. Das Modul greift auf die Bildverarbeitungsbibliothek *OpenCV* zurück, da diese bereits Bestandteil der Software ist und alle nötigen Anforderungen für die Registrierung erfüllt. Zusätzlich benötigt der Quadrocopter eine senkrecht auf den Boden gerichtete Kamera.

Das Modul stellt eine Positionsbestimmung auf Grundlage der kontinuierlichen Registrierung zur Verfügung. Über ein Interface (Schnittstelle) kann diese Positionsbestimmung komplett automatisiert aus anderen Modulen heraus gesteuert werden. So kann zum Beispiel die Navigationskontrolle die Optionen des Registrierungsalgorithmus ändern und die Positionsbestimmung starten und stoppen.

#### 4.3.2.1 Grafische Oberfläche

Neben dem Interface für die Steuerung der automatisierten Positionsbestimmung besitzt das Modul *Image Registration* drei verschiedene Oberflächen (Bereiche). Allen gemein ist die Schaltfläche *Open Camera / Close Camera* am oberen Bildrand, um eine Verbindung zur angeschlossenen Kamera aufzubauen / zu beenden. Im Drop-Down-Menü links daneben kann zwischen den angeschlossenen Kameras gewählt werden.

---

<sup>13</sup> Universal Asynchronous Receiver Transmitter

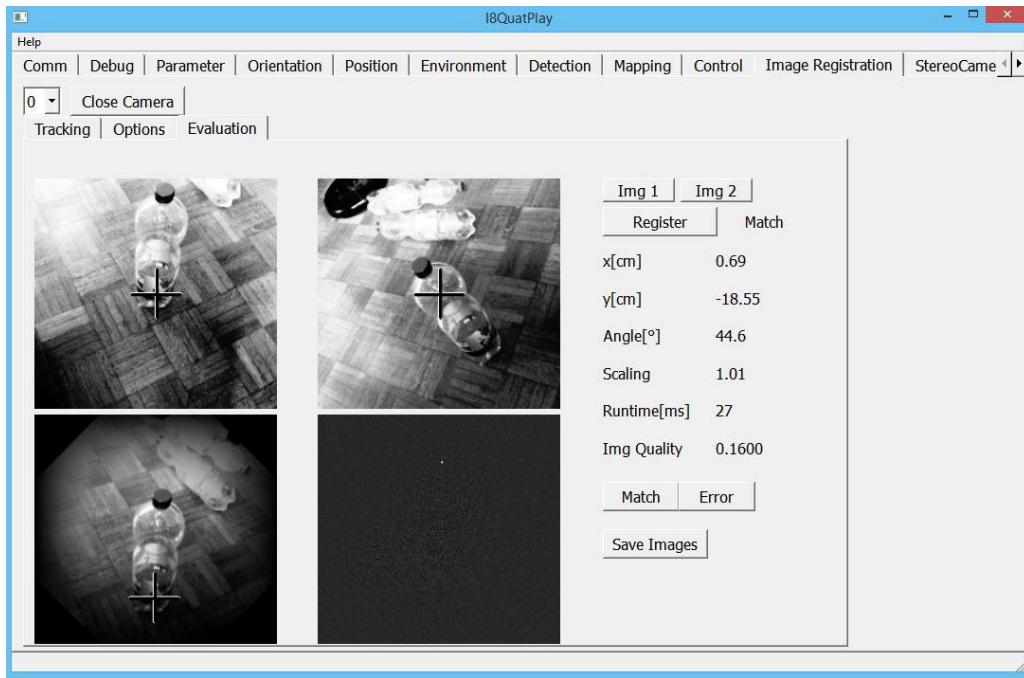


Abbildung 4.8 Screenshot des Evaluations-Modus

Im Bereich *Evaluation* können zwei Kamerabilder miteinander registriert werden. Dieser Bereich dient zur Evaluierung des Registrierungs-Algorithmus. Dabei kommt der Algorithmus aus Konzept 2 zum Einsatz. Über zwei Schaltflächen (*Img 1* bzw. *Img 2*) werden Bilder manuell aufgenommen und anschließend miteinander verglichen. Abbildung 4.8 zeigt einen Screenshot der Oberfläche dieses Bereichs.

Die aufgenommenen Bilder werden in der oberen Reihe angezeigt. Nach einem Klick auf *Registration* werden Rotation, Skalierung und Translation der Bilder berechnet und die Ergebnisse auf der rechten Seite angezeigt. Links unten wird eine affine Transformation des rechten Bildes angezeigt, die bei korrekter Registrierung dem Bild links oben entspricht. Die dunklen Ränder des linken unteren Bildes resultieren aus der Multiplikation mit einem Hanning-Fenster. Die Kreuze dienen der Überprüfung der Registrierung. Bei den Quellbildern befinden sie sich in der Bildmitte, im transformierten Bild ist das Kreuz um den Wert der Translation versetzt. Bei korrekter Registrierung stimmen die Positionen der Kreuze der linken Bilder überein.

Zur Kalibrierung der Schwellenwerte der Fehlererkennung können die Schaltflächen *Match* und *Error* benutzt werden. Bei einem Klick werden der Peak-Wert der zweiten Phasenkorrelation sowie das Ergebnis (erfolgreich oder fehlerhaft) in der Textdatei *evalRegistration.txt* abgespeichert. Ob eine Registrierung erfolgreich war, kann man an den Kreuzen der linken Bilder ablesen. Jeder Klick erzeugt eine neue Zeile in der Form:

[Peak-Wert],[0|1]

Im Kapitel Kalibrierung wird auf die Auswertung dieser Daten eingegangen.

Rechts neben der Schaltfläche *Registration* ist das Ergebnis der Fehlererkennung mit dem eingestellten Schwellenwert sichtbar. Der Text *Match* deutet auf eine erfolgreiche Registrierung hin, wie man auch an den Bildern auf der linken Seite sehen kann. Dies bedeutet, dass der Korrelationswert größer als der Schwellenwert ist. Mit der Schaltfläche *Save Images* können die vier angezeigten Bilder zusätzlich abgespeichert werden. Das Bild unten rechts zeigt die inverse Fouriertransformation des Cross-Power-Spectrums der zweiten Phasenkorrelation.

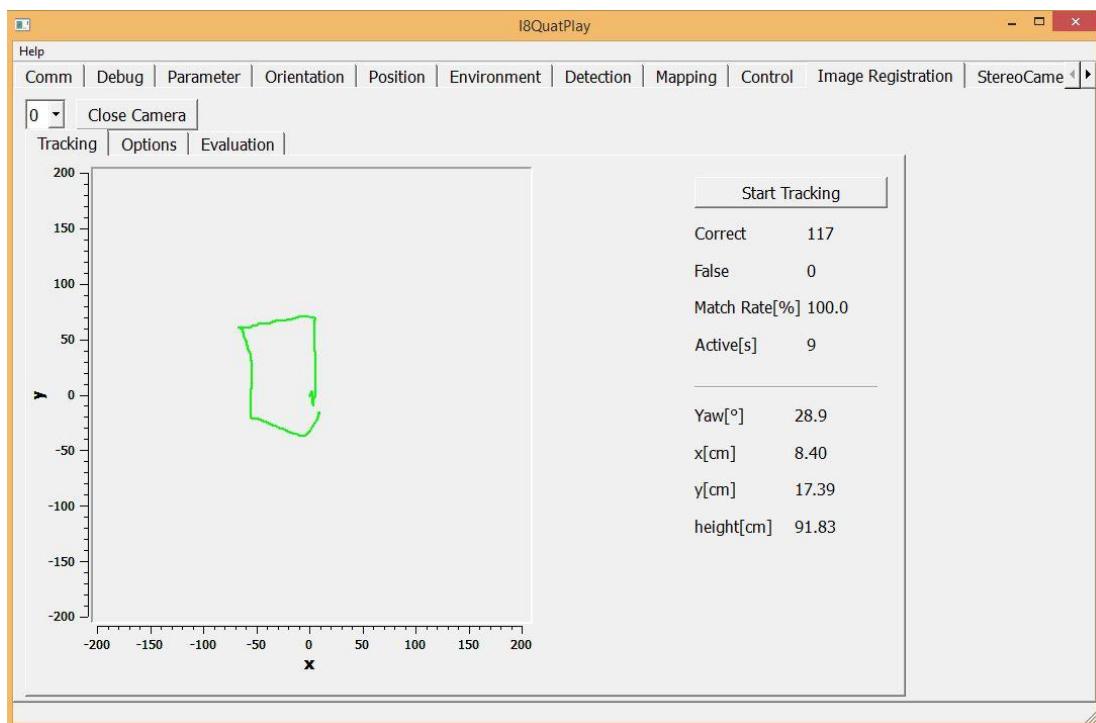


Abbildung 4.9 Screenshot des Trackings

Im Bereich *Tracking* werden Informationen der Positionsbestimmung angezeigt, falls die kontinuierliche Registrierung aktiviert ist (Abbildung 4.9).

Über die Schaltfläche *Start Tracking / Stop Tracking* wird die kontinuierliche Registrierung manuell gestartet und gestoppt. Während das Tracking aktiv ist, wird die Bodenspur des Quadrocopters, beginnend bei der Position  $\langle 0,0 \rangle$ , im Diagramm auf der linken Seite angezeigt. Direkt unter der Schaltfläche *Start Tracking* wird die Anzahl der korrekt und falsch erkannten Frames angezeigt, wobei fehlerhafte Frames nicht in die Positions berechnung mit einfließen (siehe Kapitel 3.5). Darunter werden aktuelle Informationen über Position, Orientierung und Flughöhe angezeigt.

Im Bereich *Options* werden die Parameter für die Registrierung gesetzt. Folgende Optionen stehen zur Auswahl:

- *Console Output*: Wenn aktiviert, gibt der Registrierungsalgorithmus laufend Informationen und Variablenwerte auf der Konsole aus. Diese Funktion wird für Debug-Zwecke benutzt.
- *FFT Patient*: Option für die Fouriertransformation. Wenn aktiviert, dauert die Initialisierung der Positionsbestimmung länger, jede Fouriertransformation wird aber schneller ausgeführt (siehe Kapitel 4.2.2). Dadurch kann die Framerate erhöht werden.
- *Use Height*: Sollte im Flugbetrieb immer aktiviert sein. Wird die Software auf einem externen System getestet, stehen keine Höheninformationen zur Verfügung. Durch Deaktivierung von *Use Height* wird eine Standardhöhe (100 cm) bei allen Berechnungen verwendet.
- *Use Yaw*: Deaktivierung führt zur Verwendung einer Standardorientierung von 0° (identische Wirkung wie *Use Height* auf den Gierwinkel).
- *Iterative Height*: Wenn aktiviert, greift die Registrierung in jedem Frame auf die aktuelle Höhe der Quadrocopter-Sensoren zurück. Andernfalls wird die Höhe iterativ aus der Skalierung der Bilder berechnet.
- *Iterative Yaw*: Wenn aktiviert, greift die Registrierung in jedem Frame auf den aktuellen Gierwinkel der Quadrocopter-Sensoren zurück. Andernfalls wird der Gierwinkel iterativ aus der Rotation der Bilder berechnet.
- *GUI Output*: Legt fest, ob die Benutzeroberfläche im Bereich *Tracking* bei aktiver Positionserfassung laufend aktualisiert wird.
- *SampleTime*: Legt die Wiederholrate der Registrierung fest. Unterschreitet der Wert die maximal mögliche Laufzeit, läuft die kontinuierliche Registrierung mit maximaler CPU-Auslastung. Die Laufzeit für die gewählten Optionen wird im Bereich *Evaluation* bei jeder Registrierung angezeigt.
- *Resolution*: Legt die Auflösung der Kamerabilder fest. Höhere Auflösung bedeutet höhere Genauigkeit, aber niedrigere Framerate. Es stehen die Auflösungen 480x480, 240x240 und 120x120 Pixel zur Verfügung.

Nach Änderung der Optionen müssen diese über die Schaltfläche *Apply* übernommen werden. Alle von den Optionen abhängigen Daten werden dann berechnet. Dadurch lässt sich bei kontinuierlicher Registrierung eine höhere Framerate erzielen. Über die Schaltflächen *Save Options* / *Load Options* können die aktuellen Einstellungen sowie die aktiven Pläne der FFTW gespeichert und später wieder geladen werden. Dadurch ist es möglich, die lange Initialisierungszeit beim Start bei aktiviertem *FFT Patient* zu vermeiden.

Die vier Schaltflächen auf der rechten Seite im Options-Menü dienen zum Testen der Schnittstelle zu anderen Modulen (siehe nächster Abschnitt). Bei einem Klick rufen sie die entsprechenden *Slots* der Schnittstelle auf. *ITest* legt die Optionen fest und öffnet die Kamera-Verbindung. Ist die Initialisierung abgeschlossen, wird unterhalb der Schaltflächen die Anzahl falsch übergebener Optionsparameter angezeigt. Im Normalfall sollte hier eine null stehen.

#### 4.3.2.2 Schnittstelle zu anderen Modulen

Das Interface ist für die automatisierte Benutzung der Positionsbestimmung aus anderen Modulen heraus konzipiert. Es basiert auf der Slot-Signal-Kommunikation des *QT*-Frameworks. Folgende Funktionen stehen zur Verfügung:

```
slots:  
reg_open_camera()  
reg_close_camera()  
reg_start_tracking()  
reg_stop_tracking()  
reg_set_option(OPTION, VALUE)  
  
signals:  
reg_send_position(POSITION)  
reg_send_relative_position(POSITION)
```

Insbesondere kann das Navigationsmodul mit diesem Interface die Registrierung steuern. Bei aktiverter Positionsbestimmung wird nach jedem Frame die Positionsänderung emittiert (*reg\_send\_relative\_position(POSITION)*). Bei einem Aufruf von *reg\_stop\_tracking()* wird die gesamte Positionsänderung seit dem Start durch *reg\_start\_tracking()* emittiert (*reg\_send\_position(POSITION)*). Die Funktionen stellen alle Möglichkeiten bereit, die auch über die grafische Oberfläche des Moduls ausgeführt werden können (bis auf den Bereich *Evaluation*). Eine Liste der Parameter zum Setzen der Optionen findet sich in der Dokumentation.

#### 4.3.2.3 Debug-Datenstrom

Ist die Positionsbestimmung aktiv, werden kontinuierlich Daten an das bestehende *Debug*-Modul der Kontrollsoftware geschickt. Im Debug-Modul können einzelne Datenströme ausgewählt und mit Daten anderer Module verglichen werden. Alle aktiven Datenströme werden grafisch mit zeitlichem Verlauf ausgegeben. Außerdem lassen sich aktive Datenströme als CSV-Daten abspeichern. Folgende Daten der Positionsbestimmung werden übertragen:

- Position ( $x[m], y[m]$ )
- Orientierung ( $\varphi[^{\circ}]$ )
- Höhe ( $z[m]$ )
- Qualität der Messung (Peak-Wert der zweiten Phasenkorrelation)

- Erfolgsrate (Prozentualer Anteil erfolgreicher Registrierungen)

## 4.4 Kamera

Das entwickelte System kann mit einer beliebigen Kamera genutzt werden. Das Positionsbestimmungs-Modul liest die Bilder von der *OpenCV*-Kamera-Schnittstelle ein. USB-Webcams werden von dieser automatisch erkannt. Die Schnittstelle kann im Quellcode aber auch an andere Kameras angepasst werden (z.B. Gig-Ethernet- oder *FireWire*-Kameras).

Der Quadrocopter verfügt aktuell über zwei fest installierte senkrecht nach unten gerichtete Logitech-USB-Webcams unterschiedlicher Preisklassen (*Logitech c270 USB HD Webcam* und *Logitech HD PRO Webcam c920*). Die höhere Bildqualität der teureren Kamera *c920* wirkt sich positiv auf die Zuverlässigkeit der Registrierung aus. Die Aufnahme-Einstellungen beider Kameras können über die Logitech-Software *Logitech Camera Controller* verändert werden. Bei schnellen Bewegungen liefern die Kameras sehr unscharfe Bilder. Durch Deaktivierung der Optionen *Right Light* und *Auto White Balance* wird die Unschärfe reduziert und die Zuverlässigkeit der Registrierung erhöht. Die dadurch stark verfälschte Farbtemperatur der Bilder ist nur für die Bildbetrachtung wichtig und hat keinen Einfluss auf die Positionsbestimmung. Abbildung 4.10 zeigt die Oberfläche der Logitech-Software mit optimalen Einstellungen für die Registrierung. Bei der Webcam *c920* muss zusätzlich der Autofokus im Hauptmenü der Kamerasoftware deaktiviert werden, da dieser aufgrund der Vibrationen im Flug oft versagt. Mit unendlichem Fokus liefert die Kamera in allen Flughöhen ausreichend scharfe Bilder. Die Webcam *c270* besitzt keinen Autofokus.

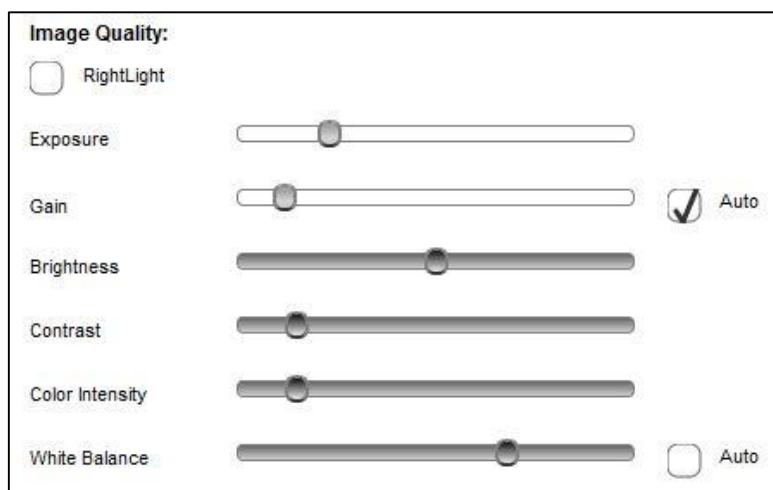


Abbildung 4.10 Screenshot der Logitech-Kamera-Software

Beide Kameras unterstützen ausreichend hohe Auflösungen sowie Framerates, um sie mit allen Parametern zur Positionsbestimmung zu evaluieren.

## 4.5 Kalibrierung

### 4.5.1 Schwellenwert der Phasenkorrelation

Der Erfolg der Registrierung kann durch den Peak-Wert der Phasenkorrelation bestimmt werden (vgl. Kapitel 3.3). Überschreitet der Wert eine gewisse Schwelle, kann von einer erfolgreichen Registrierung ausgegangen werden. Diese Schwelle hängt von der gewählten Auflösung ab. Zur Bestimmung dieser Schwellenwerte werden für alle drei verfügbaren Auflösungen einige Bilder im Evaluierungs-Modus registriert und das Ergebnis (Peak-Wert und Erfolg/Misserfolg) gespeichert (vgl. Kapitel 4.3.2.1). Abbildung 4.11 zeigt, dass ein konstanter Schwellenwert eine sehr gute Vorhersage liefert.

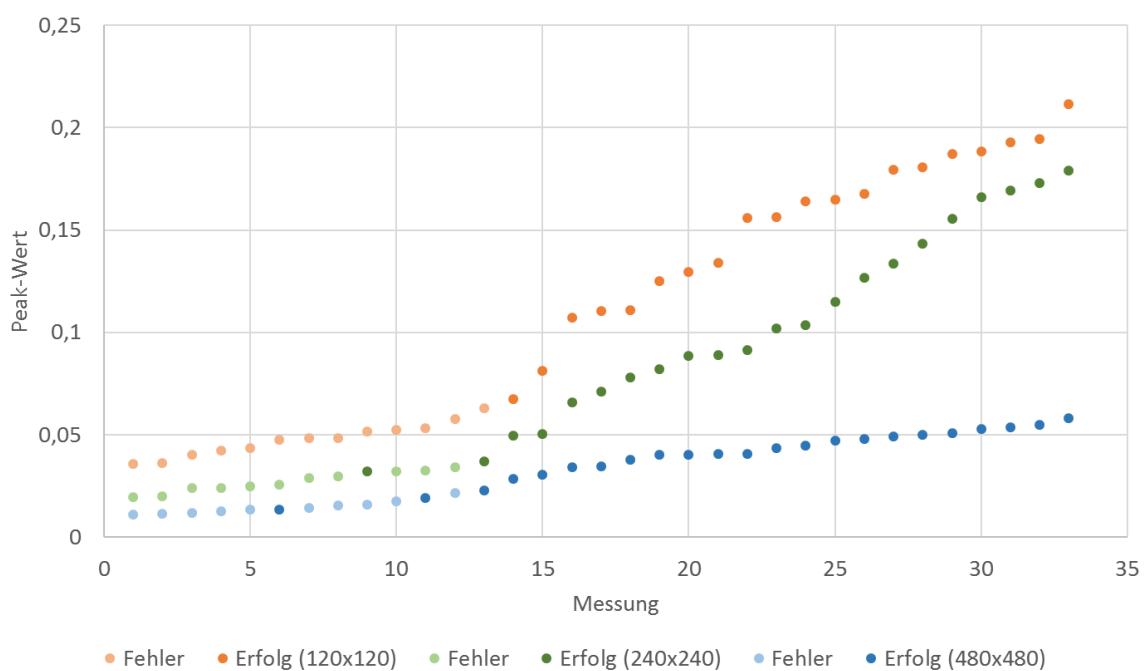


Abbildung 4.11 Ermittlung des Schwellenwerts der Fehlererkennung

Für jede der drei Auflösungen (120x120 / 240x240 / 480x480 Pixel) sind die fehlerhaften Registrierungen mit hellen und die erfolgreichen Registrierungen mit dunklen Punkten dargestellt. Die Messungen sind nach Peak-Werten sortiert. Als Schwellenwert dient jeweils der niedrigste Datenpunkt jeder Reihe, ab dem nur noch erfolgreiche Registrierungen auftraten. Dies ergibt die drei Schwellenwerte:

Schwellenwert(120x120 Pixel): 0.067  
 Schwellenwert(240x240 Pixel): 0.040  
 Schwellenwert(480x480 Pixel): 0.024

Weitere Registrierungen mit den drei Auflösungen haben gezeigt, dass die gewählten Schwellenwerte zu 97,5% korrekt über Erfolg und Misserfolg der Registrierung entscheiden.

Außerdem liefert jeder Schwellenwert unabhängig von der gewählten Kamera vergleichbare Resultate. Im Fehlerfall wird deutlich häufiger ein richtig registriertes Bildpaar als falsch angenommen, als fehlerhafte Registrierungen als richtig. In den meisten Fällen geht die Positionsänderung eines einzelnen Frames verloren, der Quadrocopter erhält jedoch kaum falsche Positionsinformationen. Im Vergleich zu Tests mit niedrigeren Schwellenwerten war die Positionsbestimmung genauer, da deutlich seltener fehlerhafte Positionsänderungen berücksichtigt werden.

#### 4.5.2 Skalierungsfaktor der Translation

Der Registrierungsalgorithmus berechnet die Translation zweier Bilder in Pixeln. Dieser Pixelwert muss unter Beachtung der Höhe in Meter umgerechnet werden, um die Positionsänderung des Quadrocopters zu erhalten (vgl. Kapitel 3.2). Der Skalierungsfaktor ist von der Auflösung und der gewählten Kamera abhängig. Die Kameras haben leicht unterschiedliche Brennweiten, weswegen für jede Kamera ein eigener Skalierungsfaktor bestimmt werden muss.

Die Auflösung geht indirekt proportional in den Skalierungsfaktor ein, da die Bildausschnitte der Kameras bei Änderung der Auflösung konstant bleiben. Bei Verringerung der Auflösung auf die halbe Kantenlänge entsprechen nur halb so viele Pixel der gleichen Strecke in Metern. In diesem Fall wird der Skalierungsfaktor verdoppelt.

Für die Positionsänderung gilt:

$$\text{Positionsänderung[m]} = \text{Translation[Pixel]} * \text{Skalierungsfaktor} * \text{Flughöhe[m]} * \text{Auflösungsfaktor}$$

Der Skalierungsfaktor wird für jede Kamera mit einer Referenzauflösung bestimmt. Für den Auflösungsfaktor gilt:

$$\text{Auflösungsfaktor} = \text{Kantenlänge(Referenzauflösung)} / \text{Kantenlänge(aktive Auflösung)}$$

Als Referenz wird die höchste Auflösung verwendet (480x480 Pixel), um ein möglichst genaues Ergebnis zu erhalten. Zur Bestimmung des Skalierungsfaktors wird der Quadrocopter bei konstanter Höhe  $h$  per Hand langsam (ca.  $0,1 \text{ ms}^{-1}$ ) zwei Mal um genau zwei Meter hin und her bewegt. Die Position wird während der Bewegung vom externen optischen Tracking-System (vgl. Kapitel 5) aufgezeichnet, um die Daten verifizieren zu können. Der Quadrocopter steht dabei auf einem fahrbaren Tisch, um Fehler durch Neigung zu vermeiden. Der Tisch hat eine Aussparung in der Mitte, um der Kamera freie Sicht auf den Boden zu gewähren. Man erhält eine Ortskurve, die der Form der Bewegung ähnlich ist, jedoch der Bewegung in Pixeln entspricht.

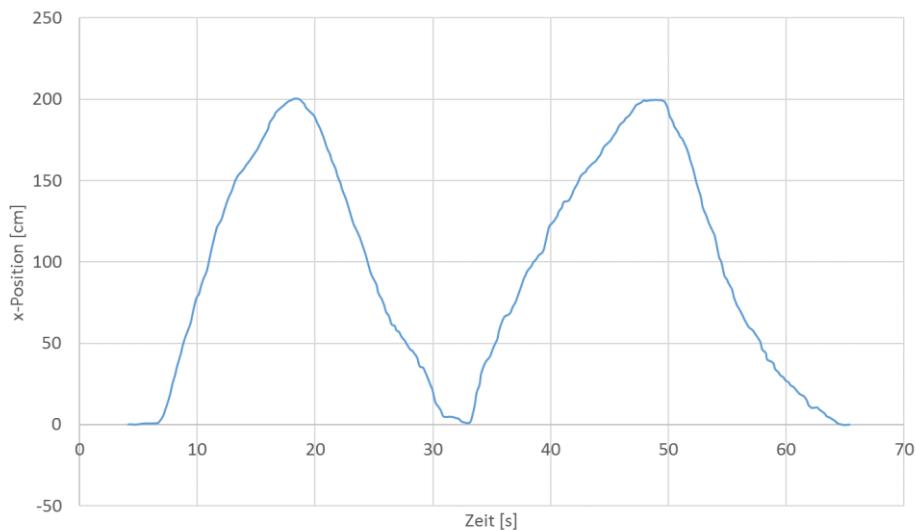


Abbildung 4.12 Kurve zur Berechnung des Skalierungsfaktors

Trägt man die  $x$ -Position über die Zeit auf, erhält man die Ortskurve der  $x$ -Koordinate. Skaliert man die lokalen Maxima und Minima dieser Kurve auf die tatsächliche Bewegung von zwei Metern, erhält man den Skalierungsfaktor. Abbildung 4.12 zeigt eine skalierte Ortskurve für die Webcam *c920*. Nach insgesamt acht Metern Bewegung erreicht die Position wieder den Nullpunkt und die Werte der Maxima stimmen nahezu genau überein. Dies lässt eine eindeutige Bestimmung des Skalierungsfaktors zu. Die ermittelte Skalierung wird durch die Höhe  $h$  geteilt, um einen höhenunabhängigen Faktor zu erhalten. Aus jeweils vier Messkurven für jede Kamera wurden durch Mittelwertbildung folgende Skalierungsfaktoren bestimmt:

$$\begin{aligned} \text{Skalierungsfaktor(c920)} &: 1,563 \cdot 10^{-3} \\ \text{Skalierungsfaktor(c270)} &: 1,389 \cdot 10^{-3} \end{aligned}$$

Der Abweichungen der Messungen liegt bei beiden Werten in der vierten gültigen Stelle. Dies entspricht einer theoretischen Genauigkeit im Millimeterbereich bei einem Meter Bewegung. Im Test ergaben diese Werte bei variabler Höhe vergleichbar genaue Ergebnisse. Deswegen werden diese Skalierungsfaktoren als konstante Werte im Programm verwendet.

# 5 Evaluierung

Ein wichtiges Ziel der Evaluierung ist die Bestimmung der optimalen Parameter für die kontinuierliche Registrierung. Dazu zählt auch die Analyse des Verhaltens unter verschiedenen Umwelteinflüssen und Umgebungen. Abschließend wird die Positionsbestimmung im Flug getestet.

Die Flüge sowie die manuellen Tests (Bewegen des Quadrocopters von Hand) wurden in der Quadrocopter-Testhalle der Universität Würzburg durchgeführt. Dort stand das optische Tracking-System *PPT X* der Firma *WorldViz* bereit, welches die 3D-Position des Quadrocopters jederzeit millimetergenau bestimmen kann. Die Daten des optischen Trackings dienen als Referenz für alle Positionsmessungen. [23]

## 5.1 Vergleich der Kameras

Theoretisch kann die Positionsbestimmung mittels Bildregistrierung mit einer beliebigen Kamera durchgeführt werden. Zur Evaluierung stehen zwei fest installierte Logitech-Webcams zur Verfügung, das Modell *c270* und das teurere Modell *c920*. Im Evaluierungsmodus der Kontrollsoftware ließen sich subjektiv keine Unterschiede zwischen den Kameras feststellen. Die Webcam *c270* liefert in allen getesteten Auflösungen eine ausreichende Bildqualität, um affin transformierte Bilder zu registrieren.

In statischen Tests zeigte sich allerdings, dass Bilder der *c920* bei identischen Einstellungen genauer registriert werden können. Weiterhin wirkt sich die höhere Leistung der *c920* bei Bewegung positiv auf die Erfolgsrate der kontinuierlichen Registrierung aus. Dies beruht wahrscheinlich auf der höherwertigen Bildstabilisierung der *c920*. In unscharfen Aufnahmen verschwinden die hohen Frequenzen des Spektrums fast vollständig, wodurch die Registrierung eher fehlschlägt.

Ein Großteil der folgenden Tests wurde mit beiden Kameras durchgeführt, um qualitative Aussagen über den Einfluss der Kamera auf die Testresultate treffen zu können.

## 5.2 Framerate und Auflösung

### 5.2.1 Trade-Off-Problem

Den größten Einfluss auf die Leistungsfähigkeit der Positionsbestimmung mittels kontinuierlicher Registrierung besitzen die ambivalenten Faktoren Framerate (FPS<sup>14</sup>) und Auflösung. Je höher die Auflösung ist, desto mehr Rechenzeit nimmt eine Registrierung in Anspruch und die maximal mögliche Framerate sinkt. Je niedriger die Framerate ist, desto größer ist der Versatz zwischen zwei Bildern. Die Erfolgswahrscheinlichkeit der Phasenkorrelation sinkt proportional zum Versatz (Translation) der Bilder. Niedrigere Auflösungen verringern gleichzeitig die Genauigkeit und Robustheit der Registrierung. Deswegen muss ein optimaler Trade-Off zwischen Laufzeit und Auflösung gefunden werden.

Auf dem Quadrocopter wurden die Auflösungen 480x480, 240x240 und 120x120 Pixel getestet. Berücksichtigt man ca. 10% Pufferzeit für jede Berechnung, lassen sich unter voller CPU-Last minimale Laufzeiten von 300 ms, 100 ms und 50 ms (respektive 3,3, 10 und 20 FPS) auf dem System erzielen. Die Laufzeiten setzen sich zusammen aus ca. 20 ms „Basisberechnungen“ und einem variablen Anteil, der linear mit der Auflösung (quadratisch mit der Kantenlänge der Bilder) steigt. Da die Positionsänderungen entlang der beiden Koordinatenachsen (xy-Ebene) unabhängig voneinander sind, bedeutet der eben genannte Zusammenhang vierfache Laufzeit für doppelte Genauigkeit.

### 5.2.2 Statisches Verhalten

Das statische Verhalten beschreibt die Positionsabweichung in Abhängigkeit der Zeit ohne Bewegung des Systems. Zur Evaluierung wurden beide Kameras jeweils drei Minuten in einem Meter Abstand vor einer konstanten Szene platziert. Im Idealfall sollte keine Abweichung vom Nullpunkt gemessen werden. Der Versuch wurde mit allen drei Auflösungen durchgeführt. Er zeigte, dass der Algorithmus eine zeitabhängig steigende Abweichung vom Nullpunkt errechnet. Alle Registrierungen wurden mit der konstanten Framerate 3,3 FPS durchgeführt, da der Fehler in jedem Frame steigt. So sind die Ergebnisse vergleichbar.

Abbildung 5.1 stellt den zeitlichen Verlauf der gemessenen Abweichung für die Auflösungen 240x240 und 480x480 Pixel grafisch dar.

---

<sup>14</sup> Frames per Second

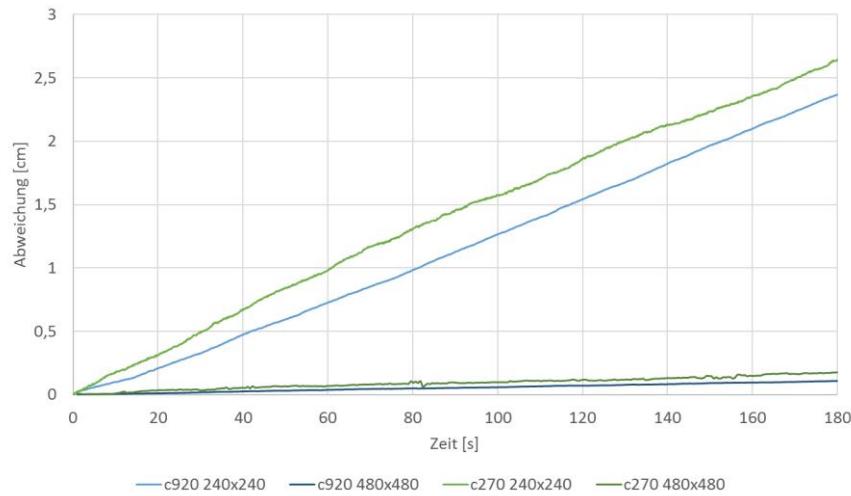


Abbildung 5.1 Positionsabweichung im statischen Test (mittlere und hohe Auflösung)

Während die Abweichung in der höchsten Auflösung vernachlässigbar ist, stellt sich bei beiden Kameras auf mittlere Auflösung nach drei Minuten bereits ein Fehler von ca. 2,5 cm ein. Die Abweichung ist direkt proportional zur Frame-Anzahl. Berücksichtigt man die kürzere Laufzeit der Registrierung unter niedrigerer Auflösung (240x240 Pixeln / 10 FPS), erreicht die Abweichung im dargestellten Zeitraum ca. 7,5 cm.

Weiterhin kann man der Grafik entnehmen, dass die Abweichung bei der höherwertigen Kamera geringer ausfällt. Der Versuch wurde für jede Testreihe drei Mal wiederholt und lieferte reproduzierbare Ergebnisse. Ein Test verschiedener Oberflächen führte zu einem qualitativ vergleichbaren Ergebnis. Die Genauigkeit der Registrierung hängt nicht davon ab, ob im Bild viele oder wenige Details (Szene mit vielen Objekten im Vergleich zu Hallenboden mit schwacher Musterung) zu erkennen sind.

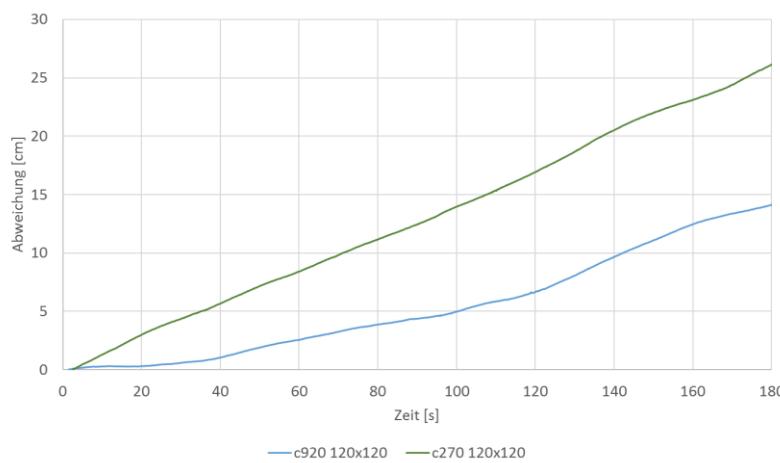


Abbildung 5.2 Positionsabweichung im statischen Test (niedrige Auflösung)

Betrachtet man das Ergebnis für die niedrigste Auflösung 120x120 Pixel, erkennt man eine noch stärkere Abweichung (Abbildung 5.2). Unter Nutzung der maximalen Wiederholrate von

50 ms akkumuliert sich der Fehler nach drei Minuten auf ca. 84 bzw. 156 cm. Die Unterschiede beider Kameras sind bei dieser Auflösung weitaus größer. Aber selbst die mit der Webcam c920 erzielten Ergebnisse sind mit der Auflösung 120x120 Pixel zu ungenau für eine akkurate Positionsbestimmung, da man davon ausgehen muss, dass diese Abweichungen in gleicher Größenordnung als systematische Fehler in die Positionsbestimmung während des Fluges eingehen.

### 5.2.3 Dynamisches Verhalten

Das dynamische Verhalten analysiert die Positionsbestimmung unter Bewegung. Der Quadrocopter wurde, auf einem Tisch stehend, mit einer Geschwindigkeit von  $0,5 \text{ ms}^{-1}$  durch den Raum geschoben. Dieser Wert entspricht ca. der Fluggeschwindigkeit des aktuell implementierten Positionsreglers. Für jede Auflösung wurde die höchste Framerate (3,3, 10 und 20 FPS) eingestellt, um eine Aussage über die am besten geeignete Auflösung treffen zu können. Stellt man hingegen für alle Auflösungen die Wiederholrate von 300 ms ein, erzielt man mit der höchsten Auflösung stets die besten Ergebnisse.

Abbildung 5.3 zeigt die Peak-Werte der Phasenkorrelation über die Zeit für das eben beschriebene Experiment unter Einsatz der Webcam c920. Der Peak-Wert der Phasenkorrelation enthält eine sehr zuverlässige Information über Erfolg oder Misserfolg einer einzelnen Registrierung, wenn er mit dem Schwellenwert der Registrierung verglichen wird (siehe Kapitel 4.5.1). Je öfter Misserfolge auftreten, desto ungenauer wird die Positionsbestimmung, da jedes Mal die Positionsänderung eines einzelnen Frames verloren geht.

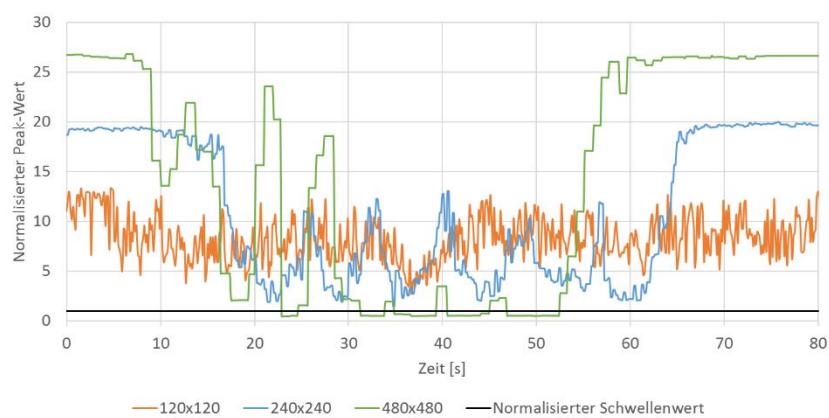


Abbildung 5.3 Peak-Werte der Phasenkorrelation unter verschiedenen Auflösungen

Die Peak-Werte sind auf den jeweiligen Schwellenwert der Auflösung normalisiert, um eine gemeinsame Hochachse zu erhalten. Zu Beginn und Ende der Zeitskala wurde der Quadrocopter nicht bewegt. Die Auflösung 480x480 Pixel und damit eine Framerate von 3,3

FPS erweist sich als ungeeignet für die Positionsbestimmung. Die grüne Linie im Graph zeigt deutlich, dass ein großer Teil der Frames nicht registriert werden konnte (Bereiche unterhalb des in schwarz dargestellten Schwellenwerts). Dies lässt sich mit der großen Translation der Bilder erklären, welche aus der niedrigen Framerate resultiert. Bei mittlerer und niedriger Auflösung konnten hingegen alle Frames richtig erkannt werden. Es fällt auf, dass bei der Auflösung 120x120 Pixel der Peak-Wert kaum von der Bewegung abhängt, jedoch stark schwankt. Auch bei schnelleren Bewegungen fiel der Peak-Wert selten unter den Schwellenwert. Der Registrierungserfolg mit der Auflösung 240x240 Pixel hängt hingegen stärker von der Bewegungsgeschwindigkeit ab, bei schnelleren Bewegungen fallen einige Frames unter die Schwelle.

Um den Einfluss der Kamera-Modelle auf das dynamische Verhalten zu evaluieren, wurden zwei Instanzen der Kontrollsoftware auf einem schnelleren System parallel ausgeführt. Jede Instanz steuerte eine der beiden Kameras an, welche fest miteinander verbunden waren und mit gleichem Sichtfeld durch den Raum bewegt wurden. Das Experiment wurde für beide Kameras mit 240x240 Pixeln und 100 ms Wiederholrate durchgeführt. Abbildung 5.4 zeigt die aufsteigend sortierten Peak-Werte für beide Kameras sowie den Schwellenwert.

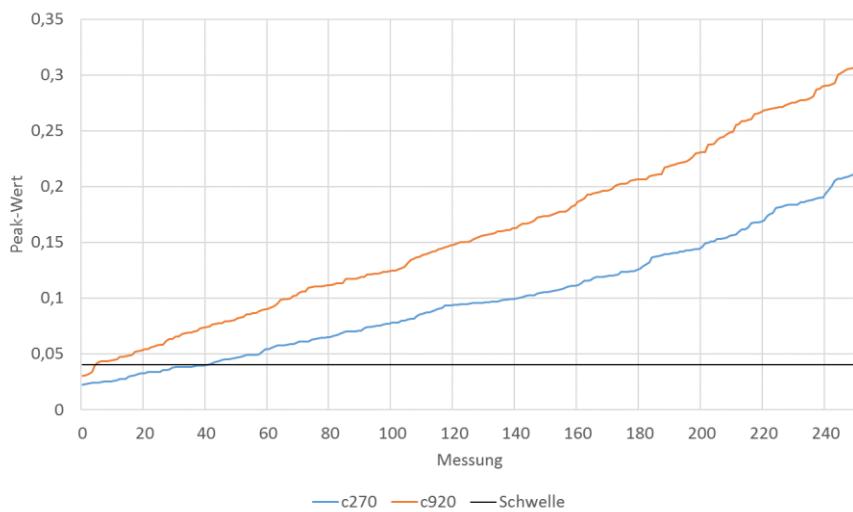


Abbildung 5.4 Vergleich der Bildqualitäten unter identischer Bewegung

Die Webcam c270 liefert bei identischer Bewegung deutlich häufiger nicht registrierbare Bilder. Im Flug führt das zu größeren Positionsfehlern.

## 5.2.4 Positionsbestimmung im Vergleich

Die Analyse von statischem und dynamischem Verhalten legt nahe, dass die mittlere Auflösung 240x240 Pixel am besten für die Positionsermittlung geeignet ist. Abbildung 5.5 und Abbildung 5.6 zeigen die  $x$ -Position des Quadrocopters während wiederholter hin- und

her-Bewegungen, gemessen durch die Kameras *c270* bzw. *c920*. Als Referenz dient das optische Tracking. Beide Versuche wurden mit einer Auflösung von  $240 \times 240$  Pixel und einer Wiederholrate von  $100 \text{ ms}$  durchgeführt. Die Versuche wurden wiederholt durchgeführt und lieferten reproduzierbare Resultate.

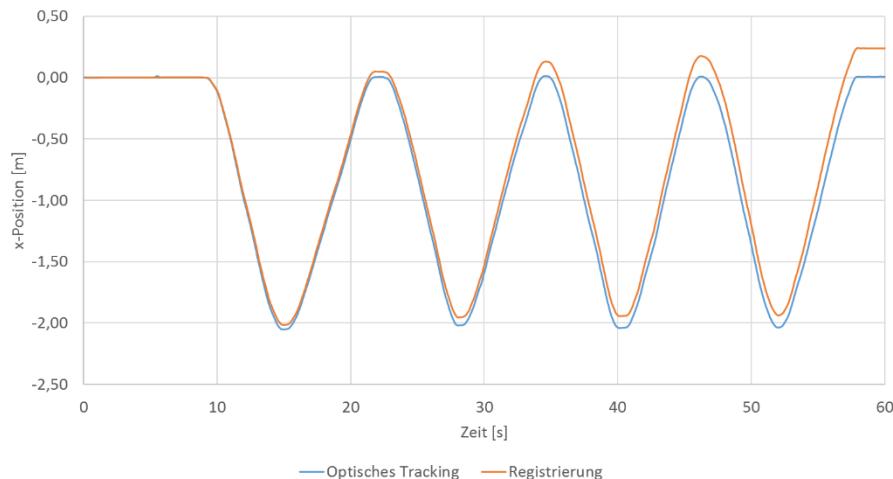


Abbildung 5.5 Positionsermittlung mit Webcam c270

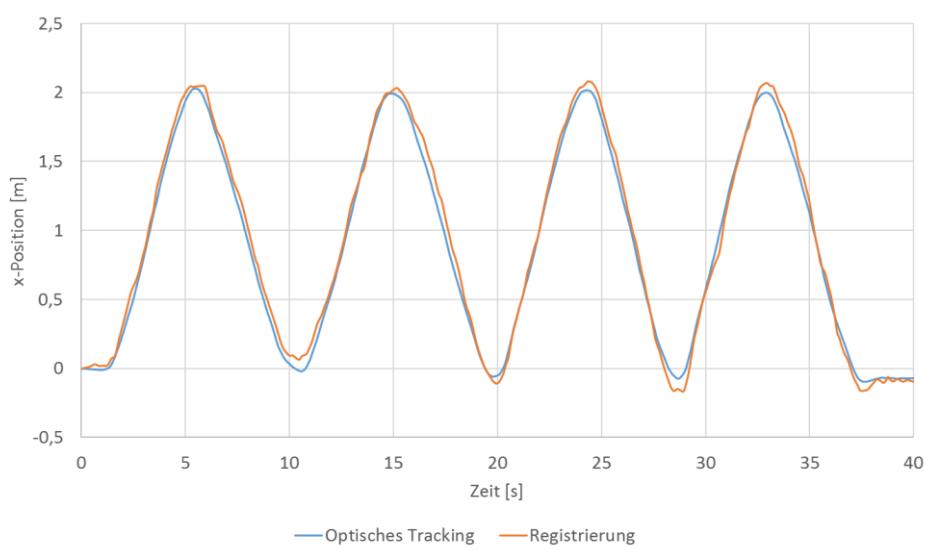


Abbildung 5.6 Positionsermittlung mit Webcam c920

Die Positionsbestimmung mittels *c920* liefert nach 16 Meter Bewegung in 40 Sekunden ein zentimetergenaues Ergebnis, mit der Webcam *c270* haben sich bei langsamerer Bewegung knapp 25cm Abweichung eingestellt. Mit der *c920* stellten sich im Mittel der Versuche 5cm Abweichung ein. Das grafisch dargestellte Ergebnis der *c920* ist sogar genauer, als durch den statischen Fehler zu erwarten war. Die leichten Unregelmäßigkeiten der Bewegungskurve sind wahrscheinlich auf Neigungseffekte zurückzuführen, da der Quadrocopter beim *c920*-Experiment getragen wurde. Die Neigungseffekte haben in diesem Fall keinen negativen

Einfluss auf das Experiment, da der Neigungswinkel stets sehr klein war und sich entgegengesetzte Neigungen in der Translation teilweise aufheben.

Mit einer Auflösung von  $120 \times 120$  Pixel lag die Abweichung bei diesem Experiment stets über  $70 \text{ cm}$ . Mit einer Auflösung von  $480 \times 480$  Pixel wurden bei jeder Wiederholung zahlreiche Frames nicht erkannt und die Abweichung betrug nach der Strecke von  $16$  Metern über  $40 \text{ cm}$ .

### 5.2.5 Fazit

Die Qualität der Positionsermittlung hängt stark von der Framerate ab. Eine Framerate von  $3,3 \text{ FPS}$  ist zu niedrig, um die Bilder bei Bewegung zuverlässig zu registrieren. Daher ist eine Auflösung von  $480 \times 480$  Pixel (oder mehr) nicht geeignet. Auf der anderen Seite ist die Auflösung  $120 \times 120$  Pixel zu gering, um akkurate Translationen zu berechnen. Der systematische Fehler ist bei dieser Auflösung zu groß, als dass diese angewendet werden kann. Im Test zeigte sich, dass die Auflösung  $240 \times 240$  Pixel ein optimaler Trade-Off zwischen Laufzeit und Genauigkeit ist. Andere Auflösungen zwischen den drei Evaluierten wären auch denkbar. In diesem Bereich gibt es jedoch keine weiteren nativen Unterstützungen der Webcams und eine Interpolation der Bilder führt zu erhöhter Laufzeit.

Das Experiment aus Abbildung 5.6 wurde mit anderen Laufzeiten wiederholt, um den Einfluss der Framerate noch genauer bestimmen zu können. Eine Verringerung der Wiederholrate um  $20 \text{ ms}$  auf  $80 \text{ ms}$  hatte erst bei schnelleren Bewegungen leichte Verbesserungen zur Folge, da mit der Bewegungsgeschwindigkeit aus Abbildung 5.6 bereits alle Frames korrekt erkannt wurden. Bei einer Wiederholrate von  $150 \text{ ms}$  erhält man bei obigen Bedingungen ungenauere Ergebnisse, da dann einzelne Frames nicht erkannt werden. Eine quantitative Aussage zur Beschreibung des Zusammenhangs zwischen Framerate und Resultat lässt sich nicht treffen, da genaue Vergleichsmöglichkeiten fehlen. Die Anwendung der *FFTW-Patient*-Option (siehe Kapitel 4.2.2) zur Verringerung der Laufzeit um ca.  $13\%$  erscheint auf jeden Fall sinnvoll unter oben genannten Aspekten.

Die Experimente zeigten, dass mit einer Auflösung von  $240 \times 240$  Pixel und der Webcam *c920* die Translation sehr genau bestimmt werden kann, wenn die Bewegungsgeschwindigkeit unter der Grenze von  $0,5 \text{ ms}^{-1}$  bleibt. Wie sich das System im realen Flugbetrieb verhält, zeigt Kapitel 5.5. Falls nicht anders angegeben, werden alle nachfolgenden Experimente mit folgenden optimalen Parametern durchgeführt:

Kameramodell:	c920
Auflösung:	$240 \times 240$ Pixel
Framerate:	10 FPS

## 5.3 Rotation und Höhenänderung

Die Registrierung kann dazu benutzt werden, den Gierwinkel sowie die Höhe iterativ zu berechnen.

Der Quadrocopter wurde per Hand in einer Höhe von ca. einem Meter mehrmals um bis zu  $90^\circ$  gedreht. Abbildung 5.7 zeigt den Vergleich der Gierwinkel-Messungen von Quadrocopter-internen Sensoren und der Registrierung. Die Kurven weisen ein sehr ähnliches Profil auf. Unter idealen Bedingungen eignet sich die Registrierung gut zur Messung des Gierwinkels, auch eine Translationsbewegung während des Versuchs verschlechterte das Resultat nicht.

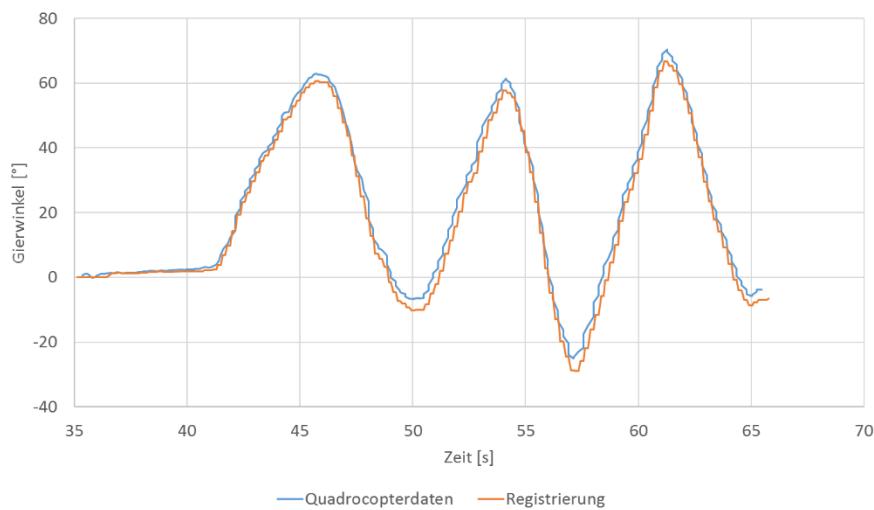


Abbildung 5.7 Gierwinkel-Messung

Zur Evaluierung der iterativen Höhenbestimmung wurde die Höhe des Quadrocopters per Hand verändert. In Abbildung 5.8 ist der zeitliche Verlauf der Höhendaten aus Registrierung und Quadrocopter-internen Messungen dargestellt.

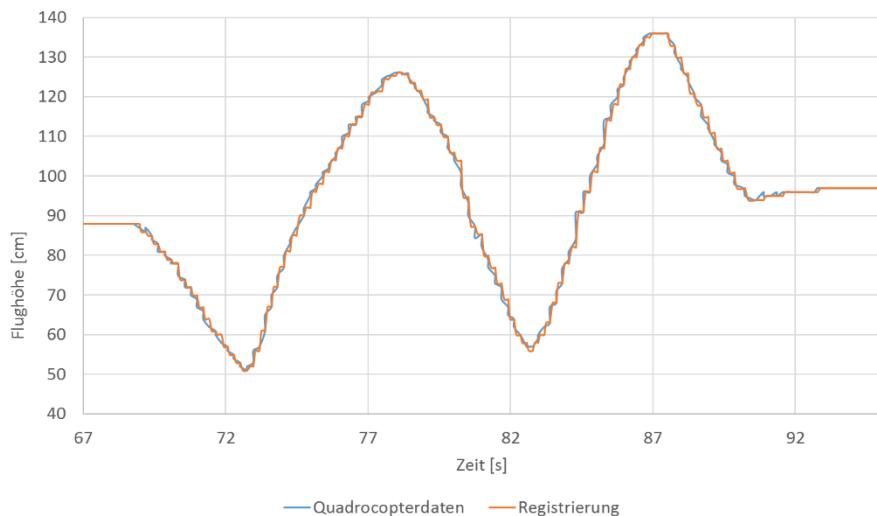


Abbildung 5.8 Höhenmessung

Werden alle Frames erkannt, können die Flughöhe sowie der Gierwinkel sehr genau mit der Registrierung bestimmt werden. Der Versuch aus Abbildung 5.6 (mehrmaliges Hin- und Hertragen des Quadrocopters) wurde mit starken Veränderungen der Höhe und des Gierwinkels während des Tragens wiederholt. Das Resultat war eine ähnlich genaue Positionsbestimmung. Folglich kann der Registrierungsalgorithmus unter idealen Bedingungen eine akkurate Positionsbestimmung unter gleichzeitiger Translation, Rotation und Skalierung errechnen.

## 5.4 Lichtverhältnisse

Die Registrierung greift auf optische Daten der angeschlossenen Webcam zurück. Der Algorithmus wurde unter verschiedenen Lichtverhältnissen getestet, um seine Robustheit gegenüber Helligkeit und Helligkeitsänderungen zu evaluieren.

In völliger Dunkelheit schlagen erwartungsgemäß alle Registrierungen fehl. Bereits bei schwacher Beleuchtung ist kein Unterschied zur Registrierung unter hellem Licht festzustellen. Erst bei direkter Sonneneinstrahlung oder künstlichen Lichtquellen im Sichtfeld der Kamera sind keine Kontraste mehr wahrnehmbar und die Bilder sind nicht registrierbar. Da die Kamera senkrecht auf den Boden gerichtet ist, sollte dieser Punkt das Positionsbestimmungssystem nicht einschränken.

Eine starke Veränderung der Helligkeit hatte keine negativen Einflüsse auf die Registrierung. Zur Evaluierung wurde die Kamera langsam durch den Raum bewegt. Gleichzeitig wurde eine helle künstliche Lichtquelle mehrmals ein- und ausgeschalten. Eine weitere, sehr schwache Lichtquelle sorgte für eine durchgehende Grundhelligkeit. Im Versuch konnten alle Bilder korrekt registriert werden.

Die hohe Robustheit gegenüber wechselnden Lichtverhältnissen liegt an der Histogramm-Anpassung vor der Registrierung (Kapitel 4.2.1) und den Eigenschaften der Fourier-Transformation. Die Bilder werden im Spektralbereich korreliert. Dadurch stellen die relativen Helligkeitsunterschiede im Bild den Kern der Information dar, die absoluten Helligkeitswerte bleiben unbeachtet.

## 5.5 Tests im Flug

Bei diesen Versuchen wurde das Verhalten der Positionsbestimmung im realen Flug untersucht. Als Vergleich dienen das optische Tracking sowie die Positionsdaten des bestehenden Systems (Messung des optischen Flusses).

### 5.5.1 Abfliegen einer Wegpunktliste

Der Quadrocopter flog in einer Höhe von ca. einem Meter eine durch Wegpunkte gegebene, knapp zehn Meter lange Strecke ab (siehe Abbildung 5.9). Der Start befindet sich an der Position  $\langle 0,0 \rangle$  (gelber Wegpunkt).

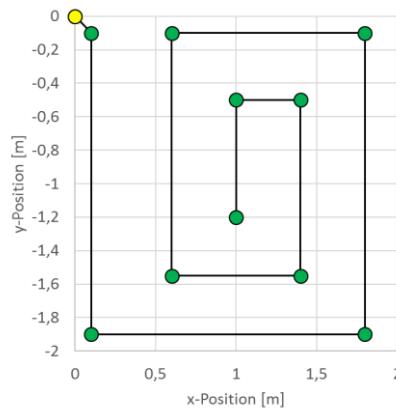


Abbildung 5.9 Wegpunktliste

Die Positionsregelung erfolgte nach den Daten des optischen Flusses. Im Idealfall sollten die Positionsdaten der Registrierung daher mit den Positionen des optischen Trackings übereinstimmen, nicht mit dem Pfad der Wegpunktliste. Die Positionserfassungssysteme wurden vor dem Start aktiviert und zeichneten den kompletten Flug bis zum Ende des Landevorgangs auf.

Der Versuch wurde mit den optimalen Parametern aus Kapitel 5.2.5 und der Webcam *c920* durchgeführt. Abbildung 5.10 zeigt die gemessenen Trajektorien aller drei Positionserfassungssysteme. Die reine Flugzeit vom Start- bis zum Endpunkt betrug ca. 30 Sekunden, die Höchstgeschwindigkeit  $0,7 \text{ ms}^{-1}$ .

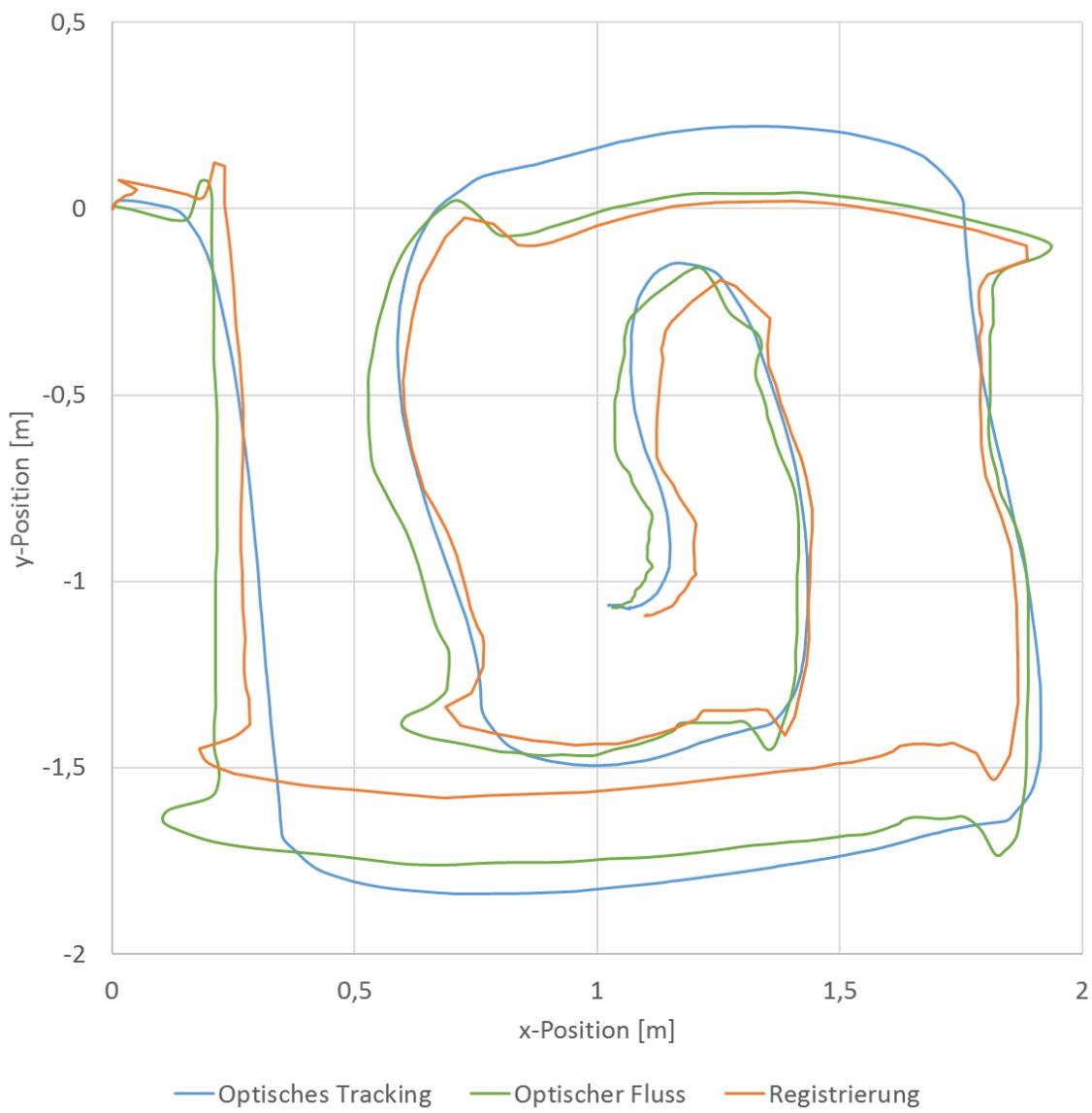


Abbildung 5.10 Trajektorien beim Abfliegen der Wegpunktliste (Webcam c920)

An der orangenen Linie erkennt man, dass die Bildregistrierung im Flug ebenfalls gute Resultate liefert. Die Abweichung zwischen optischem Tracking und Registrierung beträgt nach der Landung 7 cm. Die drei Trajektorien stimmen qualitativ überein, wobei optischer Fluss und Registrierung eine sehr ähnliche Form aufweisen. Zur genaueren Analyse der Messung sind in Abbildung 5.11 die zeitlichen Verläufe der x- und y-Positionen getrennt dargestellt (größere Grafik siehe Anhang).

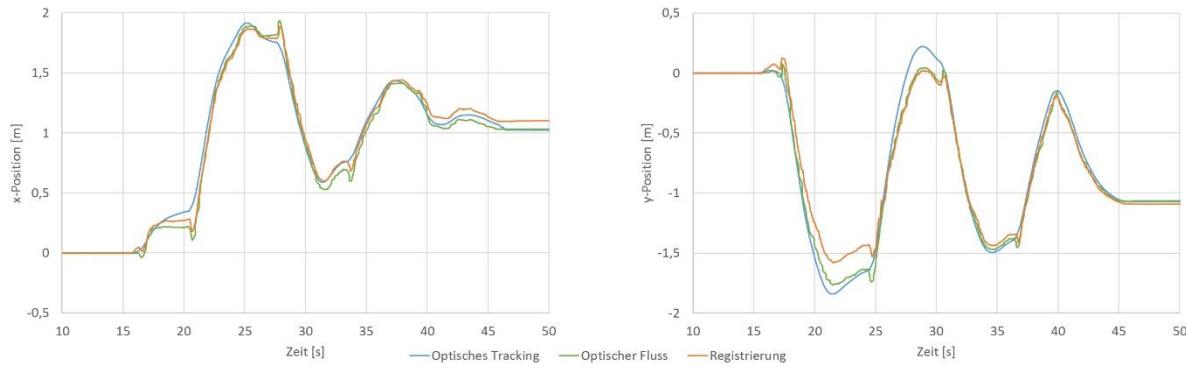


Abbildung 5.11 Positionsmessung beim Abfliegen der Wegpunktliste

Der Start erfolgte nach 15 s der Zeitskala und die Landung in Sekunde 46. In der Auf- und Abstiegsphase (je ca. eine Sekunde) treten nur leichte Differenzen zwischen optischem Tracking und den On-Board-Systemen auf. Eine niedrige Flughöhe verkleinert den Sichtkegel der Kamera. Jedoch kompensieren die vergleichsweise langsamen translatorischen Bewegungen in diesen Phasen das Problem des verkleinerten Sichtbereichs. Ab ca. 30 cm Flughöhe konnten die Bilder korrekt registriert werden, unter dieser Höhe lieferte die Kamera zu unscharfe Bilder. Da der optische Fluss nur über einer Höhe von 30 cm Positionsdaten liefert, ähnelt er bei Start und Landung der Trajektorie der Registrierung.

Während geradliniger Bewegungen ist keine Differenz zwischen den drei Positionserfassungssystemen festzustellen. Der Skalierungsfaktor der Registrierung (siehe Kapitel 4.5.2) erweist sich als guter Wert, da eine Stauchung oder Streckung der orangenen Kurven aus Abbildung 5.11 die absolute Differenz zu den Referenzwerten des optischen Trackings vergrößert.

Zu den Zeitpunkten der Richtungsänderungen weichen optischer Fluss und Registrierung vom optischen Tracking ab. Oft ist bei beiden On-Board-Systemen ein kleiner Ausschlag in die entgegengesetzte Bewegungsrichtung festzustellen. Dieser Fehler ist auf die Neigung des Quadrocopters zurückzuführen, da keinem der Systeme eine Neigungskompensation zur Verfügung steht. Der Quadrocopter kippt zum Anfliegen einer neuen Position in Richtung der Bewegung. Der Sichtkegel der optischen Systeme bewegt sich dadurch entgegen der Bewegungsrichtung und die optischen Systeme messen eine fehlerhafte Translation. Normalisiert sich die Lage zur Waagrechten, wird eine zur ursprünglichen Kippbewegung ausgleichende Translation gemessen und der Positionsfehler korrigiert.

Bei größeren Neigungswinkeln wird das Bild der optischen Systeme zusätzlich verzerrt und das Verhältnis von Pixel zu Meter ändert sich. Dadurch kann die zeitweise Abweichung der Systeme vom optischen Tracking erklärt werden. Die Systeme werden von der Verzerrung offenbar unterschiedlich stark beeinflusst. Betrachtet man die y-Positionen in Abbildung 5.11

zwischen 20 s und 25 s, weicht die Registrierung knapp 20 cm vom optischen Fluss ab. Ab dem Anfliegen eines neuen Wegpunkts stimmen die Kurven wieder überein.

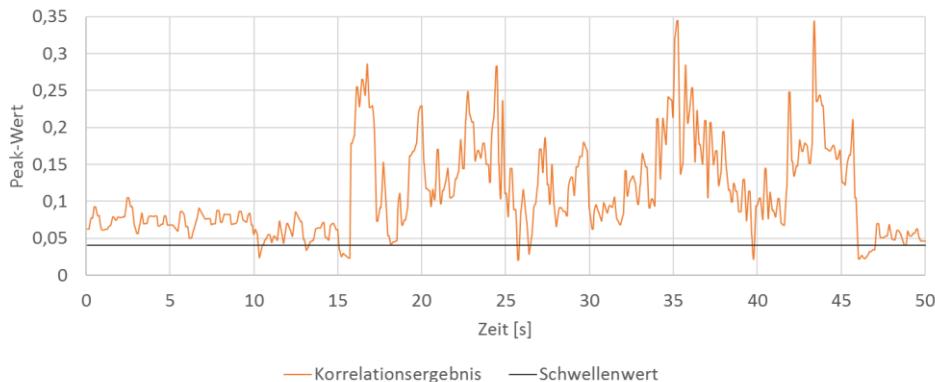


Abbildung 5.12 Korrelationswerte im Flug

Im Flug konnten 99% der Frames korrekt registriert werden (Abbildung 5.12). In der Flugphase (16 s bis 45 s) liegen nur drei einzelne Registrierungen unterhalb des Schwellenwertes. Diese fehlenden Informationen verursachen wahrscheinlich die Abweichung an der letzten Position. Deutlich zu erkennen sind die fehlerhaften Registrierungen bei Start und Landung. Wie bereits erwähnt erhält man erst ab einer Höhe von ca. 30 cm verlässliche Ergebnisse.

Das Experiment zeigt, dass die Positionsbestimmung mittels Bildregistrierung unter realen Flugbedingungen eingesetzt werden kann. Wiederholte Versuche zeigten, dass die Genauigkeit mit den Ergebnissen des optischen Flusses vergleichbar ist. Bei beiden Systemen führt jedoch die Neigung zu Fehlern in der Positionsmessung.

Die gleiche Wegpunktliste (Abbildung 5.9) wurde ebenfalls unter Einsatz der Webcam c270 abgeflogen. Nach den Ergebnissen aus Kapitel 5.2 war ein akzeptables, aber ungenaueres Ergebnis zu erwarten. In Abbildung 5.13 sind die Trajektorien der drei Positionserfassungssysteme dargestellt. Die Kurve der Registrierung weicht deutlich von den anderen Kurven ab und bestätigt diese Annahme. Im Versuch wurden nur 80% der Bilder korrekt registriert. Dadurch stimmt die Kurve der Registrierung nur qualitativ mit der Bewegung überein. Die vielen nicht beachteten Positionsänderungen verfälschen die Entfernung signifikant und führen zu einer Abweichung von 30 cm am Ende des Fluges. Eine qualitativ vergleichbare Positionserfassung zum optischen Fluss lässt sich also nur mit der höherwertigen Kamera c920 erzielen.

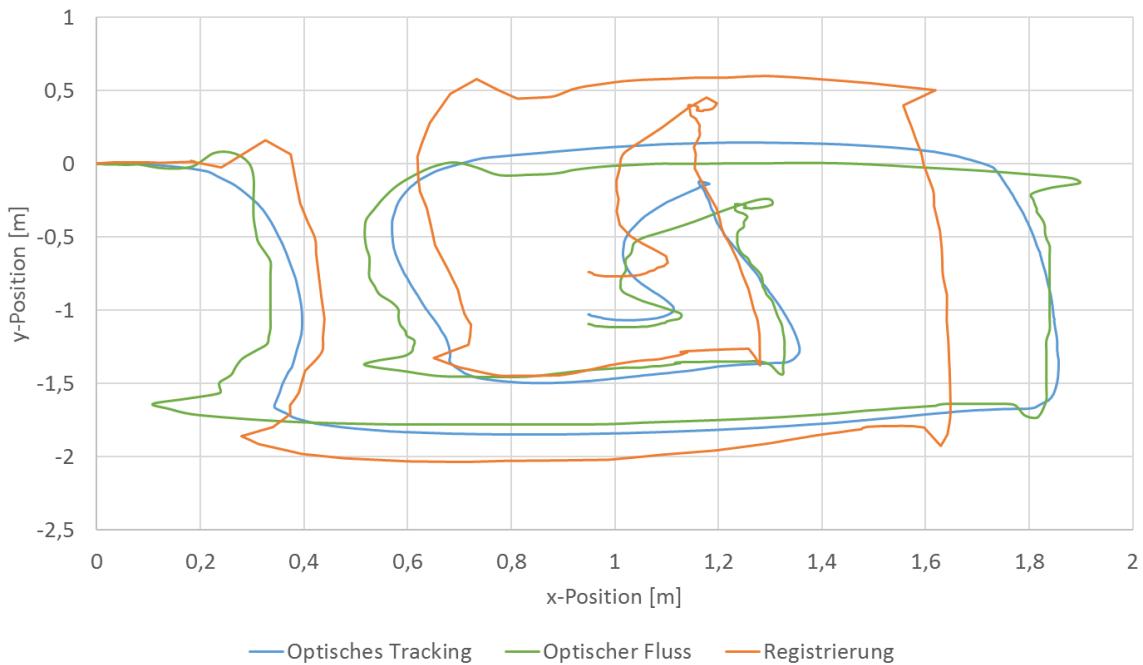


Abbildung 5.13 Trajektorien beim Abfliegen der Wegpunktliste (Webcam c270)

## 5.5.2 Rotation

Motivation der Arbeit ist die zuverlässige Positionsbestimmung während einer Rotation des Quadrocopters. Der Quadrocopter wurde in einem Meter Höhe um die Gier-Achse gedreht, um das System bei Rotation zu testen. Der aktuelle Gier-Regler des Quadrocopters ist in der Lage, eine Drehung um  $180^\circ$  in unter einer Sekunde auszuführen. Diese Dreh-Geschwindigkeit ist zu hoch für das Registrierungssystem. Zum einen können die unscharfen Bilder während der Rotation nicht registriert werden. Des Weiteren führt die sehr schnelle Bewegung zu einer Neigung des Quadrocopters. Während der Positionsfehler bei leichten Neigungen im Translations-Flug nach der Rückkehr in die Waagrechte verschwindet, bleibt der Fehler beim Rotations-Flug bestehen. Rotiert der Quadrocopter unter Neigung, sind die Rotationsachsen von Quadrocopter und Kamerabild nicht identisch. Die Rotation des Quadrocopters um die Gier-Achse führt zu einer kreisförmigen Translation in der Messung des Registrierungssystems.

Eine langsamere Rotation minimiert die Kippbewegung des Quadrocopters und führt zu registrierbaren Bildern während der Drehung. Der Rotationsversuch wurde mit einer Drehgeschwindigkeit von  $4^\circ s^{-1}$  wiederholt. Im Versuch drehte der Quadrocopter erst  $90^\circ$  in positive Drehrichtung, dann  $180^\circ$  entgegen, und anschließend in die Ausgangsposition zurück. Abbildung 5.14 zeigt die gemessenen x- und y-Positionen während der Drehung mit

der Kamera *c920* (größere Grafik siehe Anhang). Der Registrierungsalgorithmus berechnete den Gierwinkel iterativ. Die grauen Kurven zeigen, dass die Rotationsmessung der Registrierung auch im Flug exakte Ergebnisse liefert. Nach Erreichen von  $90^\circ$  und  $-90^\circ$  kehrt der Winkel wieder sehr genau zum Startwinkel  $0^\circ$  zurück.

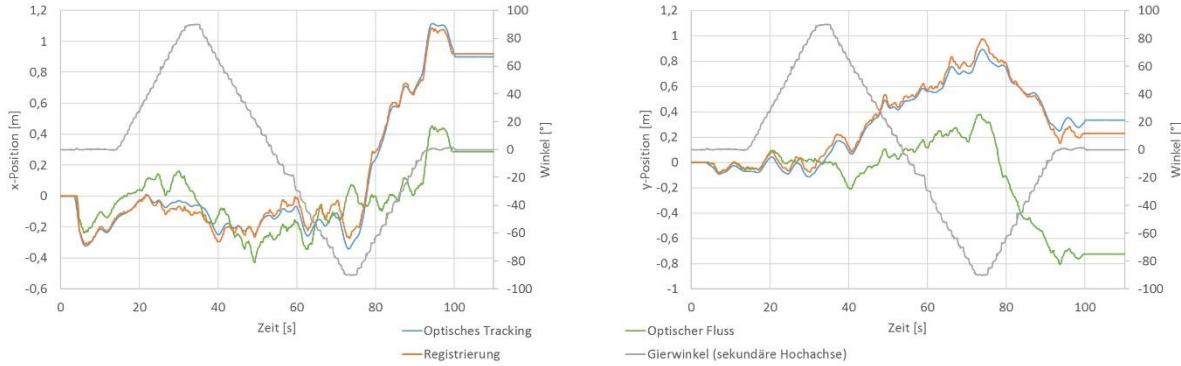


Abbildung 5.14 Positionsmeßung bei Rotation

Die orangenen Positionskurven belegen, dass die Registrierung im Rotations-Flug akkurate Ergebnisse liefert. Erst am Ende der Drehung tritt eine Abweichung zum optischen Tracking in  $y$ -Richtung von 10 cm auf. Dieser Fehler ist mit drei fehlgeschlagenen Registrierungen zu erklären. Kleinere Abweichungen im Verlauf der Drehung sind wahrscheinlich wieder auf minimale Neigungswinkel des Quadrocopters zurückzuführen.

Der optische Fluss ist nicht für Gier-Manöver programmiert und berechnet dementsprechend falsche Positionsdaten. Es fällt auf, dass die Kurve des optischen Flusses zum Teil die gleiche Form wie die anderen beiden Positionscurven aufweist. Möglicherweise führt eine Transformation der optischen Fluss-Positionen um den Gierwinkel zu korrekten Ergebnissen.

In Abbildung 5.15 sind die gemessenen Trajektorien grafisch dargestellt. Die Registrierung erfolgte mit den gleichen Parametern wie in Kapitel 5.5.1 beim Translations-Flug. Die Auflösung von 240x240 Pixeln in Kombination mit einer Wiederholrate von 100 ms ist geeignet, eine verlässliche Positionsbestimmung für alle Flugmanöver (in der Waagrechten) des Quadrocopters zu liefern. Kleinere Positionsunterschiede zum optischen Tracking resultieren aus dem statischen Messfehler, einzelnen Fehlregistrierungen und der Neigung im Flug.

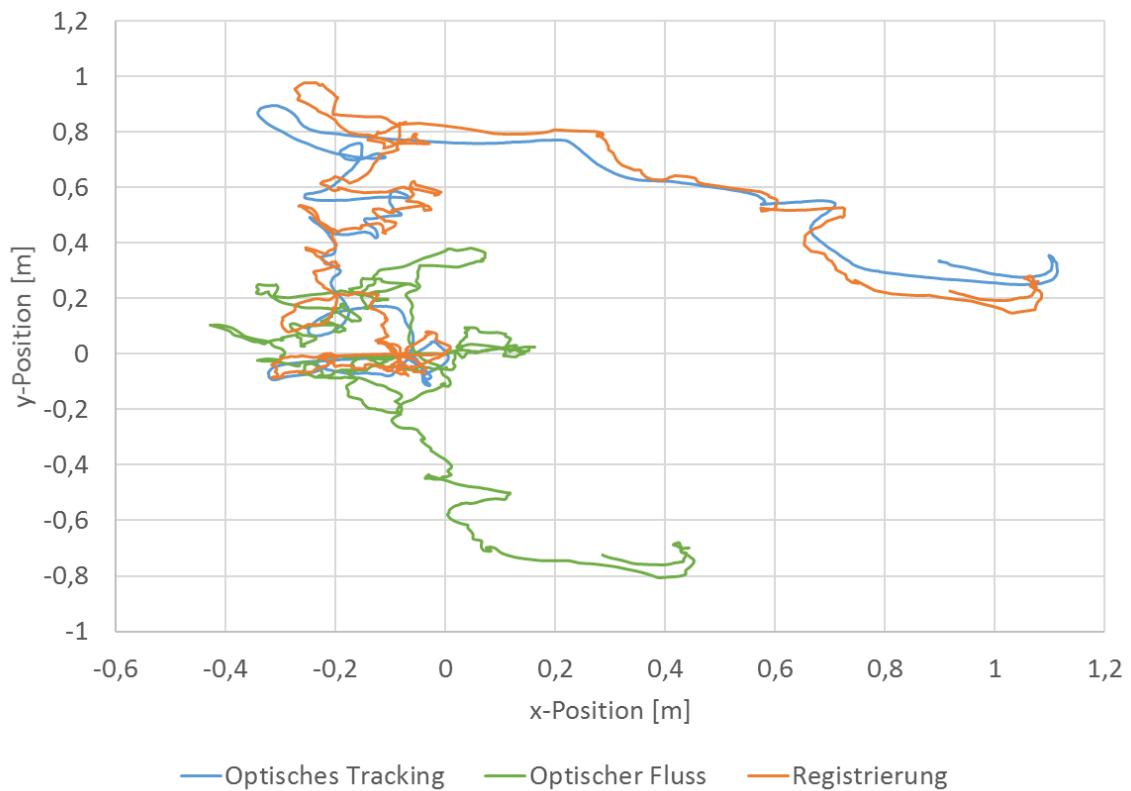


Abbildung 5.15 Trajektorien bei Rotation (Webcam c920)

## 6 Diskussion und Ausblick

Das implementierte Konzept ist gut geeignet, um Positionsänderungen des Quadrocopters für alle vorgesehenen Flugmanöver akkurat zu erfassen. Insbesondere ist der Quadrocopter mit dem entwickelten System in der Lage, seine Position im Rotationsflug zu bestimmen. Die Evaluierung zeigte, dass das System in manuellen Tests sehr genau und zuverlässig arbeitet. Im Translations-Flug erzielt die Registrierung ein zum optischen Fluss vergleichbar genaues Ergebnis. Die Registrierung ist außerdem in der Lage, den Gierwinkel sowie die Flughöhe iterativ zu bestimmen.

Das entwickelte Positionsbestimmungssystem lastet einen Kern der Dual-Core-CPU des Computers voll aus. Nach aktuellen Entwicklungsstand des Gesamtsystems erscheint es sinnvoll, die Registrierung nur bei Rotationsmanövern einzusetzen. Im Translations-Flug kann auf die Positionsbestimmung des optischen Flusses zurückgegriffen werden und der Computer kann andere Aufgaben (z.B. Objekterkennung) ausführen. Das Starten und Stoppen der Registrierung übernimmt die Navigationskontrolle des Quadrocopters. Die modulare Aktivierung der Registrierung erwies sich in Tests gegen Ende des Projekts als zuverlässige Positionserfassung für alle Flugmanöver.

Die verwendete Kamera hat einen großen Einfluss auf die Positionsbestimmung. Die Evaluierung wurde mit zwei Webcams durchgeführt, wobei nur die Logitech-Webcam *c920* ausreichend scharfe Bilder lieferte. Das Modell *c270* liefert bei Bewegung unscharfe Bilder, welche teilweise nicht registriert werden können.

Ein Teil des Fehlers der Positionserfassung ist auf die Neigung des Quadrocopters zurückzuführen. Fliegt der Quadrocopter eine Position an, neigt er sich in Flugrichtung und es wird eine fehlerhafte Translation gemessen. Dieses Problem tritt in gleicher Weise bei der Messung des optischen Flusses auf. Eine kardanische Aufhängung der nach unten gerichteten Sensoren mit einer Regelung zu einer konstant waagrechten Lage würde dieses Problem beheben und die Präzision der Positionserfassung erhöhen. Statt einer kardanischen Aufhängung könnte auch eine Fusion von IMU-Daten und Registrierung den Neigungsfehler kompensieren. Ein weiterer Ansatz ist die Berechnung von Nick- und Rollwinkel in der Registrierung. S. Zokai und G. Wolberg stellen in [24] ein Verfahren zur Berechnung der Verzerrung (Neigung) im Spektralbereich vor.

Zur Präzisierung der Positionsmessung bietet sich im Translations-Flug eine Datenfusion zwischen optischem Fluss und Registrierung an. In diesem Zusammenhang erscheint eine zu entwickelnde Multi-Core-Unterstützung der Kontrollsoftware sinnvoll. Der Registrierungs-

algorithmus könnte auf den zweiten Kern der CPU ausgelagert werden, sodass den anderen Modulen der Kontrollsoftware mehr Rechenzeit zur Verfügung stünde. Eine Fusion der Gierwinkel-Daten könnte den Sensordrift der IMU kompensieren oder weitere Referenzdaten für die Magnetometer-basierte Orientierungserkennung bereitstellen.

Abschließend betrachtet erfüllt das entwickelte System alle gestellten Anforderungen. Darüber hinaus besteht Potenzial für Erweiterungen, insbesondere in Kombination mit anderen Sensordaten.



# Anhang

## Beweis zu Formel (2.31): Lineares Gleichungssystem der trigonometrischen Interpolation

Gesucht ist eine Lösung für die  $c_k$  aus Formel (2.30):

$$f(x_j) = \sum_{k=-N+1}^N c_k e^{ikx_j} \quad (6.1)$$

Zur Berechnung der  $c_k$  muss das lineare Gleichungssystem gelöst werden. Dazu wird erst gezeigt, dass folgende zwei Formeln gelten<sup>1</sup>:

$$\sum_{j=0}^{2N-1} e^{i(l-k)x_j} = \begin{cases} 2N, & l = k \\ 0, & l \neq k \end{cases} = 2N\delta_{k,l} \quad (6.2)$$

$$\sum_{k=-N+1}^N e^{ik(x_j - x_l)} = \begin{cases} 2N, & j = l \\ 0, & j \neq l \end{cases} = 2N\delta_{l,j} \quad (6.3)$$

Beweis zu Formel (6.2):

Fall  $l = k$

$$\sum_{j=0}^{2N-1} e^{i(l-k)x_j} = \sum_{j=0}^{2N-1} 1 = 2N$$

Diese Lösung gilt für alle  $l - k = 2Nm$ ,  $m \in \mathbb{Z}$ . Durch die geforderten Grenzen von  $k$  und  $l$  ist aber nur  $l = k$  möglich.

Fall  $l \neq k$

mit Formel (2.27):

$$\sum_{j=0}^{2N-1} e^{i(l-k)x_j} = \sum_{j=0}^{2N-1} e^{i(l-k)(-\pi + j\frac{\pi}{N})}$$

Nach Ausmultiplizieren des Exponenten kann ein Teil aus der Summe herausgenommen werden

---

<sup>1</sup>  $\delta_{k,l}$  bezeichnet das Kronecker-Delta:  $\delta_{k,l} = \begin{cases} 1, & l = k \\ 0, & l \neq k \end{cases}$

$$= e^{i\pi(k-l)} \sum_{j=0}^{2N-1} \left( e^{\frac{i\pi}{N}(l-k)} \right)^j$$

Anwendung der geometrischen Summenformel ergibt

$$= e^{i\pi(k-l)} \frac{1 - \left( e^{\frac{i\pi}{N}(l-k)} \right)^{2N}}{1 - \left( e^{\frac{i\pi}{N}(l-k)} \right)}$$

Es ist also folgendes zu zeigen:

$$e^{i\pi(k-l)} \left( 1 - \left( e^{\frac{i\pi}{N}(l-k)} \right)^{2N} \right) = 0$$

Auflösen ergibt

$$\left( e^{\frac{i\pi}{N}(l-k)} \right)^{2N} = 1$$

$$e^{i2\pi(l-k)} = 1$$

Dies ist eine Kosinus-Funktion mit einem beliebigen Vielfachen von  $2\pi$  im Argument (dadurch verschwindet der Sinus-Anteil), ist also immer wahr.

■

Formel (6.3) beweist man analog, da nur der Exponent anders aufgelöst wird. Jetzt kann das lineare Gleichungssystem (6.1) für die  $c_k$  gelöst werden. Nimmt man an, dass eine Lösung für  $c_{-N+1}, \dots, c_N$  existiert, dann gilt nach Einsetzen von Formel (6.1) für  $k = -N+1, \dots, N$ :

$$\begin{aligned} \sum_{j=0}^{2N-1} f(x_j) e^{-ikx_j} &= \sum_{j=0}^{2N-1} \sum_{l=-N+1}^N c_l e^{i(l-k)x_j} \\ &= \sum_{l=-N+1}^N c_l \sum_{j=0}^{2N-1} e^{i(l-k)x_j} \end{aligned}$$

Einsetzen von Formel (6.2):

$$= \sum_{l=-N+1}^N c_l 2N \delta_{k,l} = 2N c_k$$

Setzt man dieses  $c_k$  in Formel (6.1) ein, wird die Korrektheit von  $c_k$  gezeigt:

$$\sum_{k=-N+1}^N c_k e^{ikx_j} = \frac{1}{2N} \sum_{k=-N+1}^N \sum_{l=0}^{2N-1} f(x_l) e^{ik(x_j-x_l)}$$

$$= \frac{1}{2N} \sum_{l=0}^{2N-1} f(x_l) \sum_{k=-N+1}^N e^{ik(x_j-x_l)}$$

Einsetzen von Formel (6.3):

$$= \frac{1}{2N} \sum_{l=0}^{2N-1} f(x_l) 2N \delta_{j,l} = f(x_j)$$

Also ist durch diese  $c_k$  eine Lösung gegeben. Löst man die Annahme noch nach  $c_k$  auf, erhält man die Lösung des linearen Gleichungssystems. Die Koeffizienten für das eindeutig bestimmte trigonometrische Interpolationspolynom zur Funktion  $f$  lauten:

$$c_k = \frac{1}{2N} \sum_{j=0}^{2N-1} f(x_j) e^{-ikx_j} \quad (6.4)$$

**Grafiken aus Kapitel 5.5 im Großformat**

Abbildung 5.11 Positionsmessung beim Abfliegen der Wegpunktliste

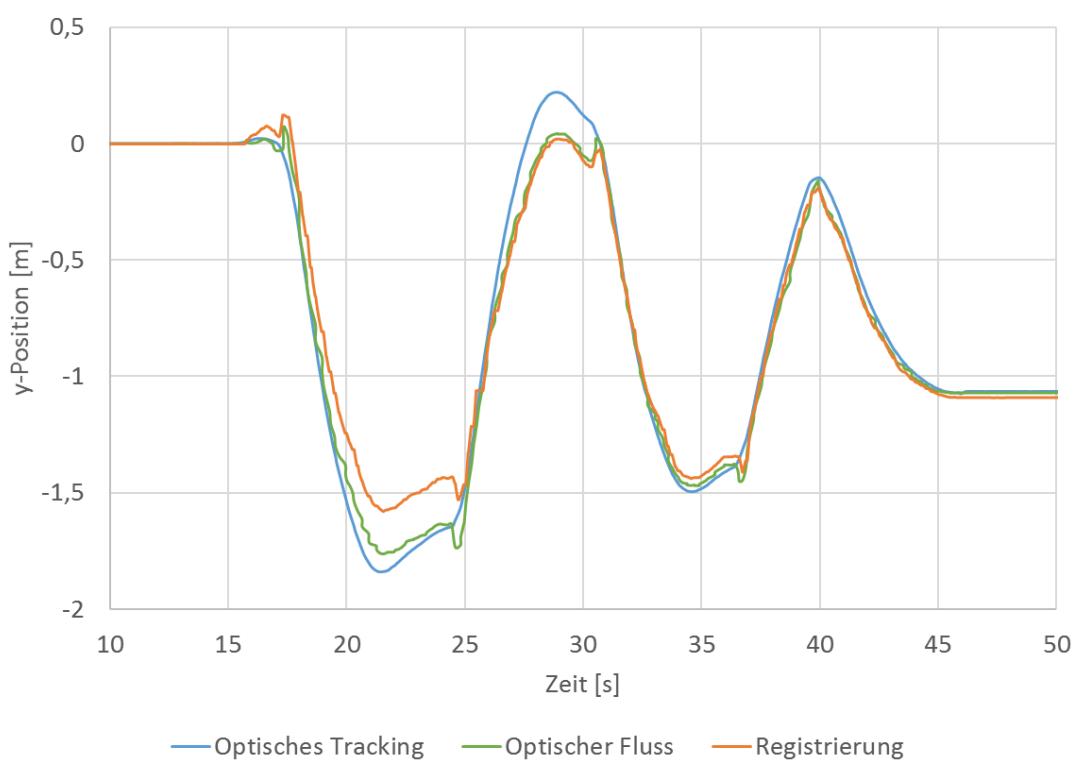
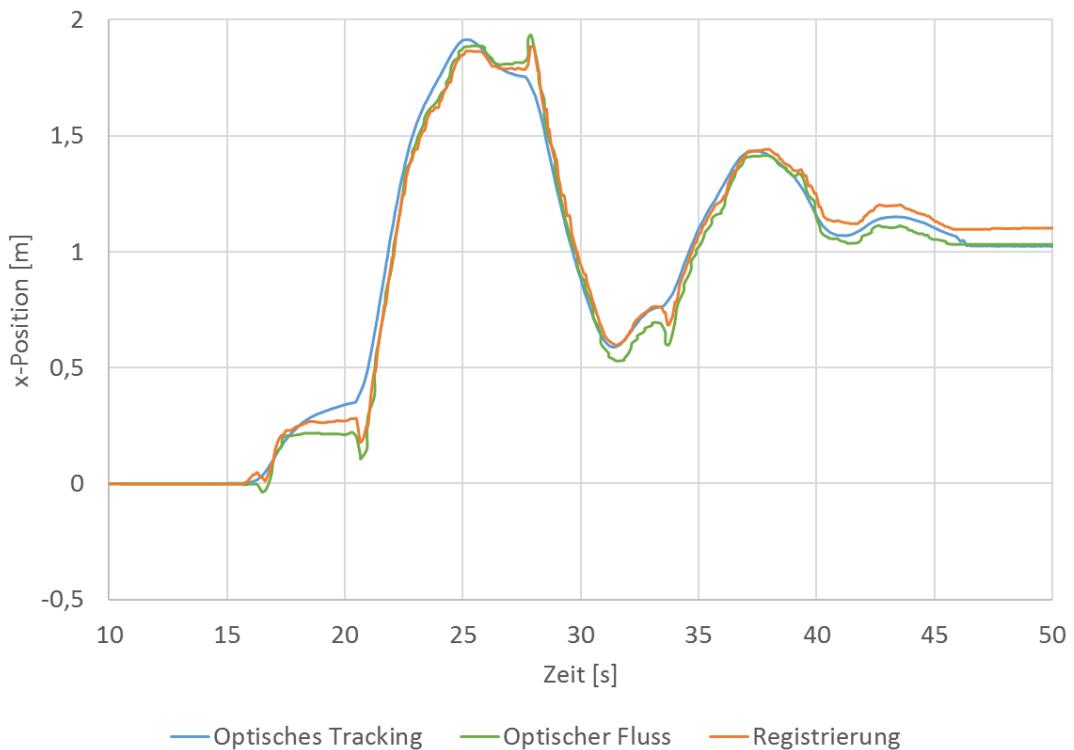
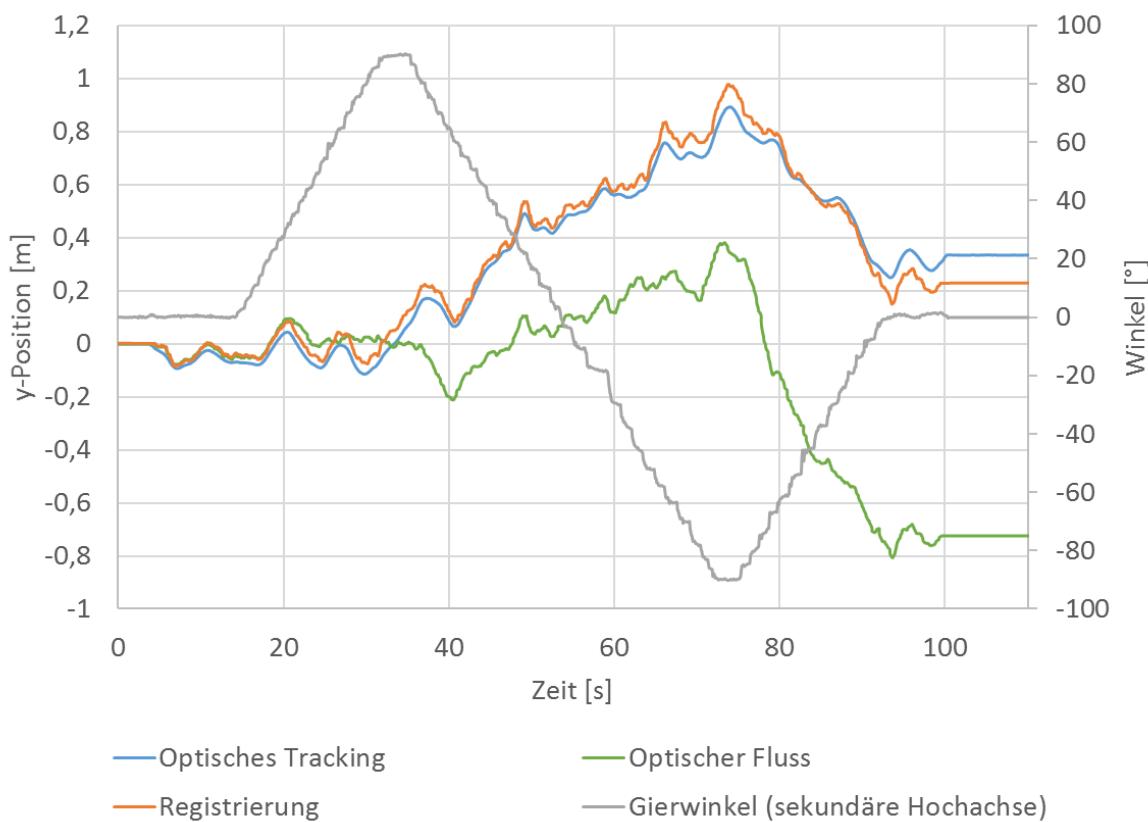
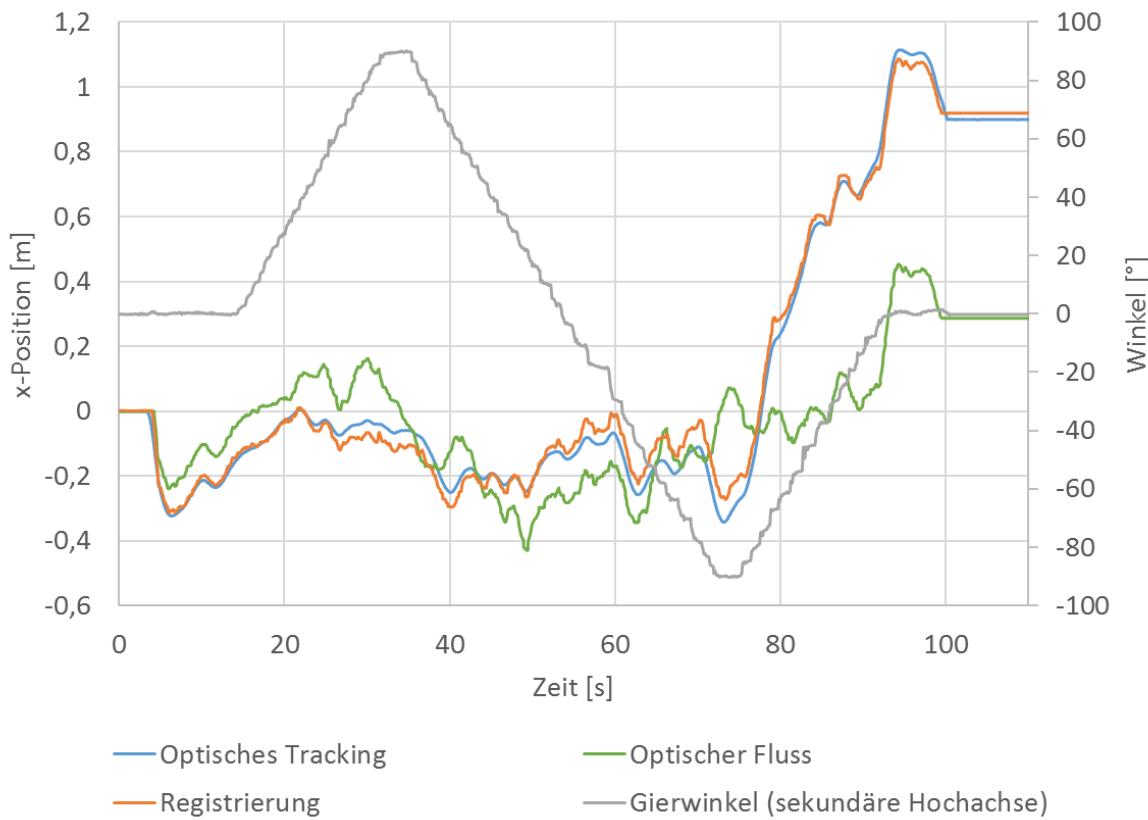


Abbildung 5.14 Positionsmessung bei Rotation



# Abkürzungsverzeichnis

CPS	Cross Power Spectrum
DFT	Discrete Fourier Transformation
FFT	Fast Fourier Transformation
FIR	Finite Input Response
FPS	Frames per Second
GPS	Global Positioning System
IFFT	Inverse Fast Fourier Transformation
IMU	Inertial Measurement Unit
LPT	Log-Polar-Transformation
OFCE	Optical Flow Constraint Equation
SAD	Sum of Absolute Differences
SLAM	Simultaneous Localization and Mapping

# Abbildungsverzeichnis

Abbildung 2.1 Blendenproblem des optischen Flusses .....	6
Abbildung 2.2 Affine Transformationen .....	7
Abbildung 2.3 Barber-Pole-Illusion .....	8
Abbildung 2.4 Vom realen Bild zum 2D-Signal .....	9
Abbildung 2.5 Kartesische und Polar-Koordinaten.....	10
Abbildung 2.6 Polar-Transformation .....	11
Abbildung 2.7 Modellierung eines Systems.....	11
Abbildung 2.8 räumliche Darstellung der Filter.....	15
Abbildung 2.9 Anwendung der Filter .....	15
Abbildung 2.10 Intensitätsfunktion eines diskreten Bildes .....	17
Abbildung 2.11 Fouriertransformation der Rechteckfunktion.....	19
Abbildung 2.12 Filter im Frequenzspektrum .....	21
Abbildung 2.13 Hanning-Fenster .....	22
Abbildung 2.14 Hanning-Fenster in der Anwendung .....	23
Abbildung 2.15 Phasenkorrelation .....	26
Abbildung 2.16 Anwendung der Phasenkorrelation .....	27
Abbildung 3.1 Allgemeines Konzept .....	28
Abbildung 3.2 Übersicht der Konzepte .....	29
Abbildung 3.3 Einfluss der Neigung auf die Positionsbestimmung .....	30
Abbildung 3.4 Schema Konzept 1 .....	31
Abbildung 3.5 Schema der Registrierung .....	31
Abbildung 3.6 Algorithmus von Konzept 1 – Teil 1.....	32
Abbildung 3.7 Algorithmus von Konzept 1 - Teil 2 .....	34
Abbildung 3.8 Fehler bei der Phasenkorrelation .....	34
Abbildung 3.9 Schema Konzept 2 .....	35
Abbildung 3.10 Polar-Transformation .....	36
Abbildung 3.11 Algorithmus von B. S. Reddy und B. N. Chatterji.....	38
Abbildung 3.12 Schema Konzept 3 .....	40
Abbildung 4.1 Positionsbestimmungsalgorithmus .....	42
Abbildung 4.2 Kamerabild mit Histogramm .....	43
Abbildung 4.3 Ausgeglichenes Kamerabild mit Histogramm .....	43

Abbildung 4.4 Histogrammausgleich bei Beleuchtungsänderung.....	44
Abbildung 4.5 Hochpassfilter.....	47
Abbildung 4.6 Kartesische und Log-Polar-Koordinaten.....	48
Abbildung 4.7 Schematische Darstellung des Quadrocopters .....	50
Abbildung 4.8 Screenshot des Evaluations-Modus .....	52
Abbildung 4.9 Screenshot des Trackings .....	53
Abbildung 4.10 Screenshot der Logitech-Kamera-Software .....	56
Abbildung 4.11 Ermittlung des Schwellenwerts der Fehlererkennung.....	57
Abbildung 4.12 Kurve zur Berechnung des Skalierungsfaktors .....	59
Abbildung 5.1 Positionsabweichung im statischen Test (mittlere und hohe Auflösung) .....	62
Abbildung 5.2 Positionsabweichung im statischen Test (niedrige Auflösung).....	62
Abbildung 5.3 Peak-Werte der Phasenkorrelation unter verschiedenen Auflösungen .....	63
Abbildung 5.4 Vergleich der Bildqualitäten unter identischer Bewegung .....	64
Abbildung 5.5 Positionsermittlung mit Webcam c270 .....	65
Abbildung 5.6 Positionsermittlung mit Webcam c920.....	65
Abbildung 5.7 Gierwinkel-Messung.....	67
Abbildung 5.8 Höhenmessung .....	68
Abbildung 5.9 Wegpunktliste .....	69
Abbildung 5.10 Trajektorien beim Abfliegen der Wegpunktliste (Webcam c920) .....	70
Abbildung 5.11 Positionsmeßung beim Abfliegen der Wegpunktliste.....	71
Abbildung 5.12 Korrelationswerte im Flug .....	72
Abbildung 5.13 Trajektorien beim Abfliegen der Wegpunktliste (Webcam c270) .....	73
Abbildung 5.14 Positionsmeßung bei Rotation .....	74
Abbildung 5.15 Trajektorien bei Rotation (Webcam c920) .....	75

# Literaturverzeichnis

- [1] "AQopterI8-Projekt (Lehrstuhl für Informatik VIII)," Julius-Maximilians-Universität, Würzburg, 2014. [Online]. Available: <http://www8.informatik.uni-wuerzburg.de/wissenschaftforschung/aqopteri8/>. [Accessed 10 03 2014].
- [2] "Navstar GPS User Equipment Introduction," Navstar, 1996. [Online]. Available: <http://www.navcen.uscg.gov/pubs/gps/gpsuser.pdf>. [Accessed 7 1 2014].
- [3] A. Nüchter, 3D Robotic Mapping: The Simultaneous Localization and Mapping Problem with Six Degrees of Freedom (Springer Tracts in Advanced Robotics), Berlin: Springer-Verlag, 2010.
- [4] "Visual Odometry For GPS-Denied Flight And Mapping Using A Kinect," MIT Robust Robotics Group, 2011. [Online]. Available: <http://groups.csail.mit.edu/rrg/index.php?n>Main.VisualOdometryForGPS-DeniedFlight>. [Accessed 10 1 2014].
- [5] D. A. D. Marshall, "Vision Systems: Optical Flow Lecture Notes," 1994. [Online]. Available: [http://www.cs.cf.ac.uk/Dave/Vision\\_lecture/node46.html](http://www.cs.cf.ac.uk/Dave/Vision_lecture/node46.html). [Accessed 12 1 2014].
- [6] R. Owens, "Optical Flow," 1997. [Online]. Available: [http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/OWENS/LECT12/](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT12/). [Accessed 12 1 2014].
- [7] M. Strohmeier, *Implementierung und Evaluierung einer Positionsregelung unter Verwendung des optischen Flusses*, Würzburg, 2012.
- [8] P. A. Devijver and J. Kittler, Pattern Recognition: A Statistical Approach, Prentice Hall, 1982.

- [9] A. K. Jain, *Fundamentals of Digital Image Processing*, New Jersey: Prentice-Hall, Inc., 1989.
- [10] T. Arens, F. Hettlich, C. Karpfinger, U. Kockelkorn, K. Lichtenegger und H. Stachel, *Mathematik*, Heidelberg: Spektrum Akademischer Verlag, 2008.
- [11] J. W. Goodman, *Fourier Optics*, Greenwood Village: Roberts & Company Publishers, 2005.
- [12] B. Jähne, *Practical Handbook on Image Processing for Scientific Applications*, Boca Raton: CRC Press LLC, 1997.
- [13] Z. Yang and F. S. Cohen, "Image Registration and Object Recognition Using Affine Invariants and Convex Hulls," vol. 8, no. 7, 1999.
- [14] Q. Zheng and R. Chellappa, "A Computational Vision Approach to Image Registration," vol. 2, no. 3, 1993.
- [15] A. A. Goshtasby, *Image Registration*, London: Springer London Limited, 2012.
- [16] National-Instruments, "The Fundamentals of FFT-Based Signal Analysis and Measurement in LabVIEW and LabWindows/CVI," 2009. [Online]. Available: <http://www.ni.com/white-paper/4278/en/pdf>. [Accessed 22 1 2014].
- [17] A. Averbuch and Y. Keller, "A Unified Approach to FFT Based Image Registration," Tel Aviv University, Tel Aviv.
- [18] B. S. Reddy and B. N. Chatterji, "An FFT-Based Technique for Translation, Rotation, and Scale-Invariant Image Registration," vol. 5, no. 8, 1996.
- [19] "OpenCV API Reference," [Online]. Available: <http://docs.opencv.org/>. [Accessed 5 February 2014].
- [20] "FFTW Library Documentation," [Online]. Available: <http://fftw.org/>. [Accessed 10 February 2014].

- [21] Commell, "Pico-ITX Computer LP-180," Taiwan Commate Computer Inc., [Online]. Available: <http://www.commell.com.tw/Product/SBC/LP-180.HTM>. [Accessed 02 03 2014].
- [22] Atmel, "Atmel 32-bit AVR UC3 Microcontrollers," [Online]. Available: <http://www.atmel.com/products/microcontrollers/avr/32-bitavruc3.aspx>. [Accessed 02 03 2014].
- [23] WorldViz, "PPT X Optical Tracking System," [Online]. Available: <http://www.worldviz.com/products/ppt/ppt-x>. [Accessed 21 02 2014].
- [24] S. Zokai and G. Wolberg, "Image Registration Using Log-Polar Mappings for Recovery of Large-Scale Similarity and Projective Transformations," vol. 14, no. 10, 2005.