

Masterarbeit

**Analyse von Trägheitssensoren im Einsatz am Filmset
zur Speicherung der Kamerabewegung für die
Verwendung in der Postproduktion**

*Analyse of Inertial Measurement Units on a filmset to capture the camera
movement for post-production*

Erstprüfer: Prof. Katja Koepl

Zweitprüfer: Prof. Dr. Bernhard Eberhardt

Hochschule der Medien Stuttgart
Studiengang Elektronische Medien
Wintersemester 2010/11

vorgelegt von:

Verena Scheunemann

Matrikelnummer: 21119

Bebelstraße 52

70193 Stuttgart

Abgabedatum: 16.05.2011


HOCHSCHULE DER MEDIEN

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Thesis selbständig und ohne unzulässige fremde Hilfe angefertigt habe. Alle verwendeten Quellen und Hilfsmittel sind angegeben.

Ort, Datum

VERENA SCHEUNEMANN

Abstract

Visual Effect Shots wurden in frühen Zeiten mit statischer Kamera aufgenommen, um die ideale Integration von VFX-Elementen zu gewährleisten. Mit verbesserten Technologien ist das Tracking mittlerweile in der Postproduktion standardisiert und ermöglicht jegliche Objektintegration. Die Kamerabewegung wird auf Basis von optischen Trackingreferenzen von 3D Tracking Algorithmen berechnet und oft mit erheblichem Zeitaufwand durch die Notwendigkeit der manuellen Anpassung und Retusche von optischen Trackingmarkern vervollständigt.

Ein weiterer Ansatz über das Tracking in der Postproduktion hinaus, dokumentiert die Kamerabewegung bereits live am Set. Virtuelle Studios greifen mit Hilfe von kommerziellen Produkten auf Motion Control Techniken zurück. Die Previsualisierung von komplexen virtuellen Sets lässt sich über die Kombination von optischen Tracking Methoden und inertialen Sensor Systemen live realisieren.

Der Wunsch über die statische Studioumgebung hinaus, ist die Speicherung der Kamerabewegung auf freiem Feld mit einer Datenübertragung in die Postproduktion zur getreuen Abbildung einer 3D Kamera.

Welche Voraussetzungen sind notwendig um diesen Wunsch zu realisieren? Inertiale Sensoren bieten bereits die technischen Möglichkeiten, doch welcher Grad an Genauigkeit ist notwendig, um eine stabile Anpassung von visuellen Elemente zu gewährleisten? Lohnt sich der Aufwand in wirtschaftlicher Hinsicht im Vergleich zum traditionellen Matchmoving in der Postproduktion?

Diese Fragen sollen in dieser Masterarbeit systematisch beantwortet werden - beginnend mit der theoretischen und praktischen Analyse von inertialen Sensorsystemen in einem Evaluierungs-Set und dem anschließendem Einsatz in der Postproduktion mit dem Ziel, das Matchmoving zu optimieren bzw. gar überflüssig zu machen.

Schlagwörter: IMU, Inertiale Sensoren, Inertial System, Matchmoving, MEMS, Motion Capture, Postproduktion, Real-time tracking, Reverse Motion Control, Tracking, Trägheitssensor, VFX, Virtuelles Set, Visual Effects, Visuelle Effekte

Inhalt

1.Einführung.....	1
1.1.Motivation.....	1
1.2.Zielsetzung.....	3
1.3.Rahmenbedingungen.....	3
2.Positionsbestimmung durch Inertiale Messsysteme.....	5
2.1.Definition Inertiale Systeme.....	5
2.1.1. Beschleunigungssensor (Accelerometer).....	6
2.1.2. Drehratensensor (Gyroskop).....	8
2.1.3. Magnetfeldsensor.....	10
2.1.4. Sensorkombinationen.....	10
2.2.Marktanalyse.....	12
3.Konzept.....	14
3.1.Idee und Zielsetzung.....	14
3.2.Anforderungsanalyse.....	15
3.3.Qualitätsmaßstab.....	21
3.4.Fallunterscheidung.....	25
4.Prototypentwicklung.....	27
4.1.IMU Auswahl.....	27
4.2.IMU Programmierung mit Arduino.....	29
4.3.Datenspeicherung mit Processing.....	32
4.4.Datenauswertung mit Excel.....	35
5.Evaluation.....	37
5.1.Selbsttest.....	37
5.1.1. Sensor in Ruhestellung.....	37
5.1.2. Sensor in Bewegung.....	40
5.1.3. Zwischenfazit.....	44
5.2.Einsatz am Drehort.....	44
5.2.1. Drehergebnisanalyse.....	46
6.Fazit.....	53
6.1.Ausblick.....	55
Abbildungsverzeichnis.....	57
Tabellenverzeichnis.....	59
Literaturverzeichnis.....	61
Anhang: IMU Arduino Code.....	63
Anhang: Processing Code.....	69
Anhang: Sensor Datenblatt LY530ALH (Auszug).....	71
Anhang: Sensor Datenblatt ADXL345 (Auszug).....	73
Anhang: Daten-DVD.....	75

1. Einführung

1.1. Motivation

Visuelle Effekte bereichern die Filmindustrie mit atemberaubenden Bildern. Bis es eine Effektszene jedoch auf die Leinwand schafft sind eine Vielzahl an Arbeitsschritten und Arbeitsstunden notwendig.

Ein unerlässlicher und zeitaufwändiger Prozess in der professionellen VFX-Produktion ist das Matchmoving und Kameratracking, ein Prozess der es ermöglicht neue Bildelemente in eine bestehende Bildbewegung zu integrieren. Ziel des Matchmovings ist die exakte Nachempfindung der realen Kamera-bewegung auf eine virtuelle Kamera. Voraussetzung für eine nahtlose Integration in Bezug auf die Bewegung ist eine exakte Berechnung der Kamerabewegung im Verhältnis zum Bild. Obwohl spezielle Softwarepakete¹ für die Bewegungsanalyse in der Postproduktion zum Einsatz kommen ist die Arbeit bei weitem kein automatisierter Prozess. Es benötigt erfahrene Operatoren für den optimalen Umgang mit dieser Software und bei komplexen Szenarien einiges an Handarbeit.

Um den Zeitaufwand in der Postproduktion zu minimieren können bereits während des Drehs einige Vorhaben getroffen werden.

Standardisierte Optionen sind hier zum Beispiel die Platzierung von so genannten Trackingmarkern im Bildausschnitt oder die Vermessung des Filmsets.²

Der Ansatz dieser Masterarbeit sieht eine Optimierung des Matchmove-Aufwands unter Anwendung von Inertialen Sensor Systemen (Inertial Measurement Unit = IMU) am Filmset vor.

Bewegungssensoren, wie Beschleunigungsmesser und Gyroskope finden heutzutage als Micro Elektromechanische Systeme (MEMS) einen breiten Anwendungsmarkt.

1 Auszug professioneller Matchmoving Software: SynthEyes™ von Andersson Technologies; boujou von Vicon (ehemals 2d3); PFTrack von The Pixel Farm®

2 Weiterführende Informationen zum Thema Matchmoving bietet die Fachlektüre von Erica Hornung „The Art and Technique of Matchmoving: Solutions for the VFX Artist“ erschienen bei Focal Press 2010

Die besonders kleinen (<5 mm) und kostengünstigen Sensoren bereichern unter anderem Festplatten, Handys oder Navigationssysteme. In der Spielindustrie sind die Nintendo Wii™ Controller erfolgreiche Beispiele für den Konsumermarkt. Als Lebensretter kommen sie seit Jahren schon in unseren Autos zum Einsatz, um bei einem Unfall in Sekundenschnelle der Airbag auszulösen.

Mit der Kombination mehrerer Sensoren, als inertielle Sensoreinheit, soll die Trajektorie der Kamera während der Bildaufzeichnung gespeichert werden, um diese in die Postproduktion zu übertragen.

Das Speichern von Bewegungsabläufen findet bereits im Motion Capture Verfahren Anwendung, das vor allem das Ziel verfolgt, die Bewegung von Menschen oder Gesichtszügen nach zu empfinden.

Speziell die Bewegungsanalyse einer Kamera findet erfolgreich in Virtuellen Studios statt. So sei hier als Beispiel ein Nachrichtenstudio genannt.

Dieses System ist allerdings auf den stationären Einsatz im Studio beschränkt.

Selbst Motion Capture Systeme lassen sich wenn überhaupt im Freien, nur unter weiteren Einschränkungen in Reichweite, Genauigkeit und zu Lasten weiterem Equipments einsetzen.

Die neuesten Entwicklungen in diesem Technologiesektor, die auch in der Filmindustrie umgesetzt werden, gehen bisher so weit, dass in einer Studio-umgebung Bewegungen von Schauspielern synchron zur Kamerabewegung live in eine 3D Umgebung gebettet werden können.³ Voraussetzung ist immer eine Infrastruktur, die sich nachteilig auf Mobilität und Flexibilität in ihrer Anwendung auswirkt.

Darüber hinaus finden sich auf dem Markt keine kommerziellen Systeme zur Bewegungsanalyse einer Filmkamera ohne zusätzliche Infrastruktur.

³ Die neuesten Technologien im Bereich der Filmproduktion fanden zuletzt ihren Einsatz zum Dreh von „Avatar“ unter Regie von James Cameron, 2009.

Wünschenswert wäre die Möglichkeit zur Speicherung jeglicher Kamerabewegung im beliebigen Raum (sowohl Interieur als auch Exterieur), die eine exakte Nachbildung einer virtuellen Kamera zulässt, um das Matchmoving in der Postproduktion zu ersetzen.

1.2. Zielsetzung

Ziel dieser Masterarbeit soll die Speicherung von beliebigen Kamerabewegungen in allen sechs Freiheitsgraden (x-, y-, z-Position und Yaw-, Roll-, Pitch-Winkel) unter ausschließlicher Anwendung von Inertialen Sensoreinheiten sein. Inertiale Sensoren ermöglichen die absolute Positionsbestimmung im Raum durch Kombination von Beschleunigungs- und Neigungswinkeldaten. Diese Daten sollen im sinnvollen Verhältnis zu einem Koordinatensystem gebracht werden, um sie in ein Matchmoving Programm zu importieren. Es soll vor allem die Frage beantwortet werden, inwieweit die Genauigkeit der Sensoren hilfreich ist, um das Matchmoving zu optimieren.

Absolutes Qualitätskriterium für die Trajektorie ergibt sich aus dem standardisierte Verfahren in der Postproduktion, der Berechnung einer 3D Kamera in einem Matchmoving Programm. Diese 3D Kamera soll mit der aus dem Sensor entstandenen Kamerabewegung verglichen werden, um die Differenz der Bewegungspfade zu definieren.

1.3. Rahmenbedingungen

Im Rahmen dieser Masterarbeit sind allerdings einige Einschränkungen auf das Forschungsvorhaben zu legen, die im Folgenden beschrieben sind.

Eine Filmkamera darf in ihrer Bewegungsbeliebigkeit als Handkamera sowohl im Interieur als auch Exterieur eingesetzt werden. Jedoch stets in angemessener Bodennähe, sowie in Kamerakranhöhe. Helikopter-, Flugkameras und Russian-Arm-ähnliche Konstruktionen sind hier ausgeschlossen.

Die Reichweite der Bewegung soll sich auf bis zu fünf Meter beschränken und die Aufnahmedauer unter zwei Minuten bleiben.

Die Auflösung der Bilddaten soll sich mindestens im HD-Bereich befinden.

Über das theoretische Konzept hinaus wird diese Arbeit einen praktischen Anwendungstest mit einer Fallunterscheidung am Set ausführen.

Zum Einsatz kommen soll eine Sensoreinheit, die möglichst wenig programmierspezifische Eigeninvestition erfordert, um den Evaluationsfokus auf die Handhabung am Set und die Anwendung für Visuelle Effekte zu legen.

2. Positionsbestimmung durch Inertiale Messsysteme

Die Grundidee zur Verwendung eines Inertialen Messsystems im Zusammenhang mit einer Kamera liegt in der Gewinnung von Trajektorien im Raum.

Die Bewegung eines Objektes, wie der Kamera, wird aus Geschwindigkeit, Richtung und zurückgelegter Strecke gewonnen. Die absolute Positionsbestimmung ist stets auf die vorherige Position angewiesen. Ausgehend von einem bekannten Startpunkt kann der neue Aufenthaltsort berechnet werden.

Inertiale Sensoren fanden ursprünglich ihren ersten Einsatz in der Raumfahrt und sind seit den 1990er in jedem Autoairbag montiert, um das spontane Stoppmanöver des Autos zu erkennen und den Airbag auszulösen. Später tauchten sie in Smartphones, Laptops und Computer Spielen wie Nintendo's Wii auf. Als sich die Leistung der Sensoren über die Jahre hinweg verbesserte und die Kosten sanken, fand man neue Industriezweige. In der Filmindustrie ermöglicht die Sensortechnik die Abbildung von Schauspielerbewegungen auf 3D Charaktere mit dem so genannten Motion Captureverfahren; vgl. [DES10].

2.1. Definition Inertiale Systeme⁴

„**Inertialsystem**, *das; -s,-e PHYSIK* Bezugssystem zur Beschreibung von Bewegungen, in dem das Trägheitsprinzip gilt.“ [LAN11]

inert [lat.]: träge; unbeteiligt [DUD74]

„**Inertialsystem**, ein Bezugssystem, in dem Körper sich gemäß der newtonschen Axiome bewegen, d.h., es treten keine Scheinkräfte auf wie z.B. Fliehkräfte. Jedes Bezugssystem, das sich relativ zu einem Inertialsystem nach Betrag und Richtung mit konstanter Geschwindigkeit bewegt, ist ebenfalls ein Inertialsystem. In der Praxis verwendete Bezugssysteme sind stets nur näherungsweise Inertialsysteme. Insbesondere ist ein übl. Laborsystem aufgrund der Schwerkraft und der Erdrotation kein Inertialsystem. Der Übergang von der Beschreibung der Physik in Inertialsystemen zu einer

⁴ <http://www.analog.com/en/mems/products/glossary/term.html>

allgemein gültigen physikal. Theorie entspricht dem Übergang von der newtonschen Mechanik bzw. der speziellen Relativitätstheorie der allgemeinen Relativitätstheorie.“ [BRO02]

Inertiale Sensoren nutzen demnach das Prinzip der Trägheit um eine Messung durchzuführen. Trägheitssensoren lassen sich zwischen Beschleunigungssensoren und Neigungswinkelsensoren unterscheiden.

2.1.1. Beschleunigungssensor (Accelerometer)

Ein Beschleunigungssensor misst sowohl die statische (Gravitation) als auch dynamische Beschleunigung (Geschwindigkeitszu- bzw. abnahme) durch die Differenzwahrnehmung einer fixierten Masse am bewegten Objekt und einer unabhängigen Masse.

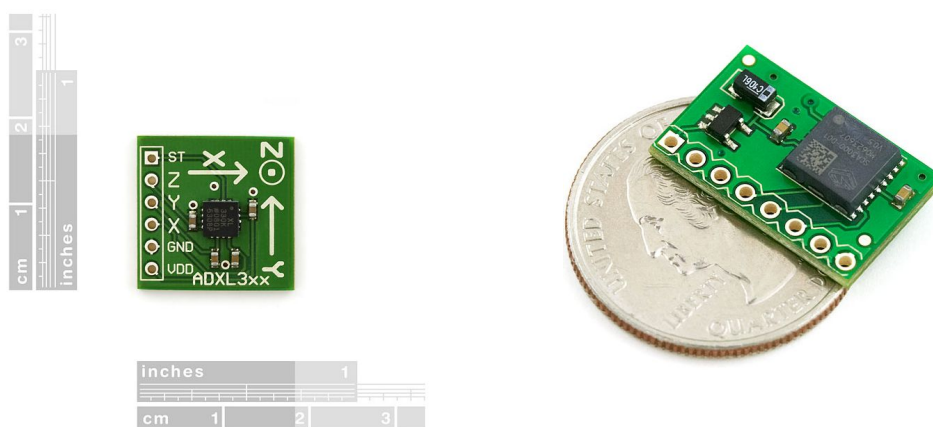


Abb. 2.1. : Beschleunigungssensoren

Miniaturisierte elektromechanische Systeme - MEMS (Micro Electro Mechanical Systems) beinhalten eine so genannte seismische Masse m , die an einer Feder mit konstanter Rückstellkraft befestigt ist und ausgelenkt werden kann. Durch die Auslenkung wird die elektrische Kapazität gemessen. Das zweite Newton'sche Gesetz findet hier seine Notwendigkeit, da dies besagt, dass die auftretende Kraft F abhängig von der Masse m und der Beschleunigung a ist.

$$F = m \cdot a$$

Die Einheit einer Beschleunigung definiert sich in Metern pro Sekunde im Quadrat (m/s^2), oder über die mittlere Erdbeschleunigung g .

$$1 g = 9,81 m/s^2$$

Werden kontinuierliche Beschleunigungsmessungen aufgezeichnet, so bezeichnet man diese Messdaten als Akzelerogramm und man erhält eine Trajektorie.

Beispielhafte Einsatzgebiete von Beschleunigungssensoren:

- in Handys und Kameras zur Lageerkennung
- in der Videospieleindustrie zur Steuerung
- in Fahrzeugen für Airbags

Aufgrund der Vielzahl von Sensortypen sind bestimmte Spezifikationen für eine projektspezifische Selektion zu unterscheiden.

Skalierung (Full-Scale-Range):

Einer der wichtigsten Faktoren ist der minimale und maximale Bereich, in dem Sensor in der Lage ist zu messen. Je kleiner der Full-Scale Bereich ist, desto genauer wird das Messergebnis ausfallen können. Für eine Beschleunigung zwischen $+2 g$ und $-2 g$ wird ein Full-Scale Wert von $\pm 250 g$ keine nutzbaren Ergebnisse liefern. Neben der fixen Skalierung lassen manche Sensortypen eine variable Wertedefinition zu.

Schnittstelle (Interface):

Ein weiteres Merkmal ist die Schnittstelle. Gemessene Sensorwerte werden entweder analog, digital oder als Pulsweitenmodulation (PWM) ausgegeben. Die, am einfachsten zu implementierende, die analoge Schnittstelle gibt eine Spannung V aus, die direkt proportional zur gemessenen Beschleunigung ist. $0 g$ entsprechen demnach dem Mittelwert einer Sensorspannung (z.B. 1,65 V).

Im Gegenzug sind die digitalen Ausgabeformate aufwändiger im System zu implementieren, lassen jedoch mehr Einstellmöglichkeiten offen und sind mit geringerer Rauschanfälligkeit belastet.

Anzahl der Messachsen:

Es gibt Sensoren, die sich darin unterscheiden ob sie in ein, zwei oder drei Achsen gleichzeitig messen können.

Bandbreite:

Die Bandbreite (in Hz) eines Sensors schränkt die Abtastfrequenz der Datenmessung ein.

Stromversorgung:

Dieses Merkmal lässt Rückschlüsse auf die notwendige Stromspeisung zu.

Accelerometer	Full-Scale Range	Bias stability	Orientation accuracy	Interface	Axes	Maximum Bandwidth	Power Requirements	Bonus Features
SCA3000	±2g	± 0.005g	±0.5°	SPI USB RS232 TTL serial	3	300Hz	2.35-3.6V, 300-650µA	Selectable frequency response, temperature output, interrupt features

Tabelle 2.1: Beispielhaftes Datenblatt eines Beschleunigungssensors

2.1.2. Drehratensensor (Gyroskop)

Im Gegensatz zu einem Beschleunigungssensor sind Gyroskope nicht von der Gravitation abhängig. Sie liefern Informationen über die Orientierung eines sich bewegenden Objektes im Raum durch Messung der Winkelveränderung entlang einer Messachse.

Das Gyroskop (gr. *gyro* „Kreis“ und *skopein* „sehen“; dies soll auf das Sichtbarmachen der Erddrehung hinweisen) wurde von Johann Gottlieb Friedrich von Bohnenberger erfunden. 1852 wurde der Begriff „Gyroskop“ von Jean Bernard Léon Foucault eingeführt.

Wirkt eine Kraft F auf einen Kreisel, weicht die Kreiselachse in Richtung des erzeugten Drehmoments T aus und beginnt eine Rotation ω' senkrecht zu T und der Kreiselachse.

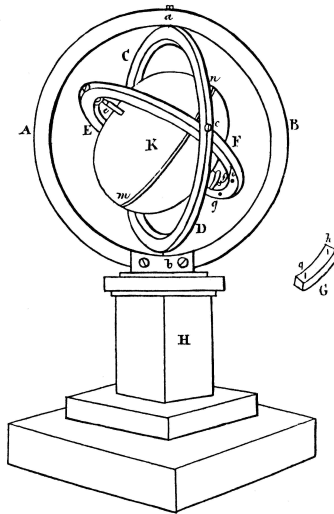


Abb. 2.2.: Gyroskop Bohnenberger

Wird der Kreisel an seinen beiden Enden fest verankert wirkt sich das Drehmoment T auf die Lage zum festen Bezugssystem aus, so dass dessen Veränderung gemessen werden kann.

Die Winkelfrequenz wird in der Einheit Rotation pro Minute (RPM) oder Grad pro Sekunde ($^{\circ}/s$) gemessen.

Die drei Rotationsachsen x, y, z werden auch als *roll*, *pitch* und *yaw* Winkel definiert.

Zu den gängigsten Drehratensensoren gehören MEMS-Kreisel⁵, Faserkreisel und Ringlaserkreisel.

Erstere zählen zu den kostengünstigsten, jedoch auch von geringster Güte.

Über einen längeren Zeitraum haben Gyroskope eine starke Drifteigenschaft, auch Bias genannt, die jedoch in Kombination von Magnetfeldsensoren kompensiert werden können.

Beispielhafte Einsatzgebiete von Drehratensensoren:

- zur der Flug- und Fahrzeugnavigation
- Motion-Control in der Filmindustrie
- im Tunnelbau

Weiterführende Informationen zu Drehratensensoren finden sich in [STR04].

Die Spezifikationen eines Gyroskopes ähneln die der Beschleunigungssensoren.

Accelerometer	Full-Scale Range	Sensitivity	Interface	Axes	Maximum Bandwidth	Power Requirements	Bonus Features
LPR530A L	$\pm 300^{\circ}/s$	3.33 $mV/^{\circ}/s$	I2C/SPI digital output interface	2	140Hz	2.4 V to 3.6 V	Access to power-down, self-test, and high-pass filter reset pins

Tabelle 2.2: Beispielhaftes Datenblatt eines Drehratensensors

⁵ MEMS = Micro Electro-Mechanical System

2.1.3. Magnetfeldsensor

Ein Sensor der nicht zur Kategorie der Trägheitssensoren zählt ist der Magnetfeldsensor. Dieser wird durch seine zuverlässige Erdmagnetfeldmessung und relativ preiswerte Herstellung gerne zur Unterstützung der Gyroskope eingesetzt.

2.1.4. Sensorkombinationen

Eine Kombination aus mehreren Sensoren wird als Trägheitsnavigation oder Inertialsensoreinheit, engl. Inertial Measurement Unit (IMU) bezeichnet. Hier steht die Sensorik in Verbindung mit der Elektronik zur Bereitstellung einer Schnittstelle im Vordergrund. Ein Inertial Sensor Assembly (ISA) bezeichnet dahingegen nur die eigentlichen Inertialsensoren. Vgl. [INT07]

Die inertielle Navigation gewann ab Mitte des 20. Jahrhunderts bereits an Bedeutung, wobei es sich hier rein um ein Koppelnavigationsverfahren (engl. dead reckoning) handelt, das ohne Information über Lageänderungen arbeitet und somit vom IMU zu unterscheiden ist.

Ein IMU erfasst die Bewegungsart, Richtung und Einheit in mehreren Freiheitsgraden - Degree of Freedom (DoF). Für eine vollständige Bewegungs- und Orientierungsanalyse eines Objektes im freien Raum sind sechs Freiheitsgrade notwendig. Demnach werden drei Neigungswinkelsensoren und drei Beschleunigungssensoren für Transversal-, Sagittal-, und Frontalebene jeweils orthogonal zueinander ausgerichtet. Je nach Sensoreinheit liefern IMUs Beschleunigungen und Drehraten oder Geschwindigkeits- und Winkelinkremente.

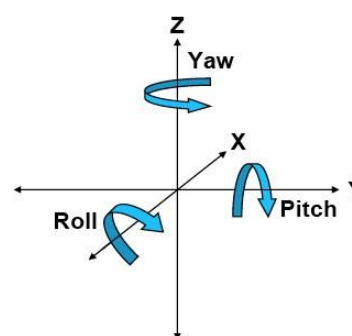


Abb. 2.3.: 6 Freiheitsgrade

Die fortlaufende Ermittlung der Bewegungsrichtung und Beschleunigung lässt, nach vergangener Zeit, Rückschlüsse auf die Position und Orientierung der Sensoreinheit bzw. des gekoppelten Objektes zu. Die zurückgelegte Strecke im Zusammenhang

mit Bewegungsrichtung wird im betrachteten Zeitintervall zur letzten Position hinzu addiert. Die körperfesten Achsen können dann in ein Koordinatensystem mit raumfesten Koordinatenrichtungen (x/y/z) umgewandelt werden.

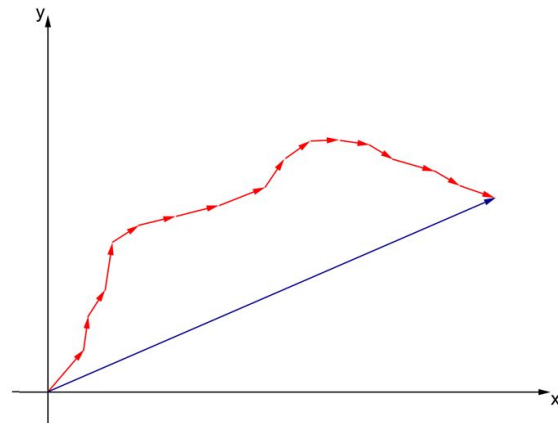


Abb. 2.4.: relative Positionsbestimmung aus Vektoren

Der mathematische Hintergedanke findet mit Hilfe des **Strapdown-Algorithmus** statt.

Die Strapdown-Rechnung lässt sich laut [INT07] S.45, grob in drei Schritte einteilen:

- Propagation der Lage durch Integration der Drehraten
- Propagation der Geschwindigkeit durch Integration der Beschleunigung
- Propagation der Position durch Integration der Geschwindigkeit.

Inertialsensoren messen die Beschleunigungen und Drehraten des körperfesten Koordinatensystems bezüglich eines globalen Inertialkoordinatensystems dessen z-Achse auf die Rotationsachse der Erde und die x/y-Achsen auf die Äquatorebene fällt. Die Betrachtung des erdfesten Koordinatensystems mit den Breiten- und Längengrad kann hier zu diesem Thema vernachlässigt werden, da im lokalen Koordinatensystem gearbeitet wird.

Inertiale Messsysteme haben die Eigenschaft nur kurzzeitig eine genaue Messung zu liefern. Über die Zeit wachsen Messfehler ohne geeignete Gegenmaßnahmen jedoch stets an.

Ursachen liegen z.B. in der endlichen Abtastrate der Sensorik oder der Temperaturfühligkeit. Des weiteren sind die Sensoren zwar idealerweise exakt orthogonal zueinander ausgerichtet, in der Realität weisen sie aber geringe Achsenfehlstellungen auf. Dieser Bias wird nach Vermessung der Teile im Nachhinein rechnerisch kompensiert.

siert und zwischen konstanten Wert bzw. zeitveränderlichen Wert (Drift) unterschieden.

Durch spezielle Algorithmen ist es möglich die Fehlercharakteristiken von Sensorrohdaten einzuschränken, diese sind in [INT07] näher beschrieben.

Darüber hinaus lassen sich Genauigkeit und Zuverlässigkeit eines IMUs erhöhen, indem sie zusätzlich von einem Magnetometer, Alimeter oder GPS (Global Positioning System) unterstützt werden. Durch diese Erweiterung bezeichnet man ein solches System als Attitude and Heading Reference Systems (AHRS). Der Hauptunterschied zwischen IMU und AHRS liegt in der On-board Verarbeitung der Sensorrohdaten im AHRS, das Lage und Richtungsmessungen direkt löst und ausgibt anstatt sie an eine Recheneinheit weiter zu leiten.⁶ Vgl. [AHR11].

2.2. Marktanalyse

Der Markt bietet Inertiale Sensoreinheiten, die auf die Bedürfnisse unterschiedlichster Industrien abgestimmt sind:

Industriezweige (beispielhafter Einsatz):

- Consumer Elektronik (Computerspielsteuerung, Mobilfunkgeräte)
- Automobilindustrie (Navigation, Airbag)
- Fertigungsindustrie (Qualitätsmessung, Zugführung, Robotik)
- Medizintechnik (kinematische Werkzeuge)
- Marine (unbemannte U-Boote)
- Verteidigung (Flugkörper, Raketen)
- Luft- und Raumfahrttechnik (zivile Flugbeförderung, Satelliten)

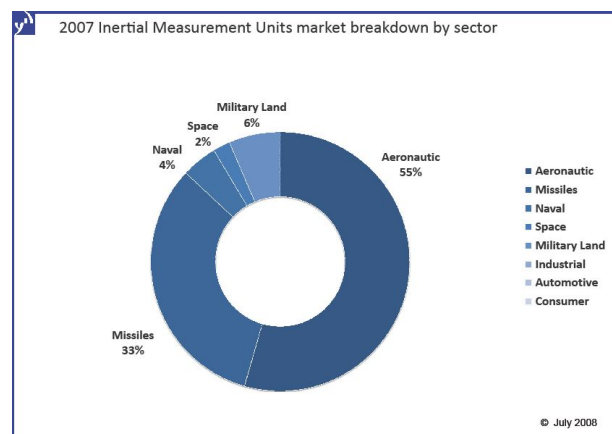


Abb. 2.5: IMU Marktsegmentierung

⁶ Ein interessantes Open Source Projekt mit weiterführenden Informationen zu AHRS Systemen im Einsatz in unbemannten Flugkörpern bietet das OpenPilot Projekt; vgl. [OPE11]

Die Anforderungen an Güte und Zuverlässigkeit der Systeme variieren. Stabilisierungs- und Navigationslösungen für Luft- und Raumfahrt sind weitaus teurer als die kleinen konsumerfreundlichen micro Systeme.

Um ein Gefühl für Sensorarten zu bekommen sei in Tab.2.3 ein Auszug an Herstellern und 6 DoF IMU Systemen vorgestellt.

Hersteller	iMAR	Summit Instruments	Analog Devices	MicroStrain	Xsense
Modell	iIMU-FCAI	65210ES	ADIS16405	3DM-GX1®	MTi
Einsatzgebiet	Fertigungsindustrie	Militär	u.a Robotik; Medizintechnik	u.a. UAV ⁷ ; Computerindustrie	u.a. Marine
Beschleunigung					
Messbereich (±)	5 g	55 g	18 g	1.7 g/ 5 g/ 18 g	5 g
Auflösung (±)	< 5 mg	60 mg	2 mg	3 mg/ 5 mg/ 10 mg	2 mg
Linearitätsfehler	< 0.03 %	0.2 %	0.5 %	0.2 %	0.2 %
Gyroskope					
Messbereich (±)	450 °/s	max. 1200 °/s	250 °/s	1200/ 300/ 50 °/s	150 °/s
Auflösung (±)	0.00027 °/s	2°/s - 6 °/s	0.007 °/s	0.7 °/s	1 °/s
Linearitätsfehler	< 0.03 %	0.1 %	0.1 %	0.2 %	0.1 %
Abmessungen					
Gewicht	130 x 115 x 140 mm	70 x 70 x 151 mm	32 x 23 + 23 mm	65 x 90 x 25 mm	58 x 58 x 22 mm
Typ	2200 g	823 g	k.A.	74.6 g	50 g
Typ	Faserkreiselsensoren	k.A.	MEMS	MEMS/AHRS	MEMS/AHRS
Kosten	k.A.	k.A.	550 EUR	2290 EUR	1750 EUR
Produktwebseite	http://www.imar-navigation.de/downloads/IMU_FCAI-03.pdf Vgl. [IMA11]	http://www.summitinstruments.com/products/products.asp?pg=41	http://www.analog.com/en/sensors/inertial-sensors/adis16405/products/product.html	http://www.microstrain.com/3dm-gx1.aspx	http://www.xsense.com/en/general/mti

Tabelle 2.3: Überblick IMUs

Die Anforderungen an die Sensoren sollen im Folgenden Kapitel vom Anwendungsgebiet einer Kamerabewegung hergeleitet werden.

7 UAV = Unmanned Aerial Vehicle (unbemannte Flugkörper)

3. Konzept

3.1. Idee und Zielsetzung

Mit dieser Abschlussarbeit wird die Idee verfolgt, mit inertialen Sensoren die Trajektorie einer Kamerabewegung zu speichern. Neben der Konzepterstellung soll auch ein Prototyp zur Verifikation erstellt werden, um das Konzept in der Praxis zu prüfen.

Ziel ist die Montage einer Sensoreinheit an eine Videokamera, um über einen kurzen Zeitraum die Kamerabewegung zu speichern, die letztendlich eine möglichst genaue Abbildung der Trajektorie liefern soll. Als Qualitätsmaßstab gilt die optisch getrackte Kamerafahrt aus einem Matchmovingprogramm.

Die rohen Sensordaten werden nach der Datenaufzeichnung über den Strapdown Algorithmus in absolute Positionskoordinaten gewandelt, um sie mit dem finalen Tracking zu vergleichen. Bereits die Gegenüberstellung der jeweiligen Werte über Graphen werden Differenzen deutlich. Ein Import in eine Matchmoving Software soll eine Analyse ermöglichen inwieweit die Daten dem internen Algorithmus zum Kamera Solving⁸ von Nutzen sind.

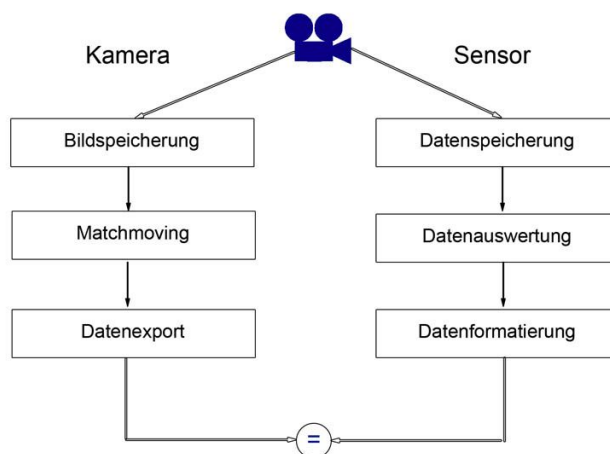


Abb. 3.1: Visualisierung der Vorgehensweise

⁸ Das Kamera Solving ist ein Begriff aus dem Matchmoving und bezeichnet den Prozess der Berechnung von Kameraposition und Neigung inkl. Brennweite nach Analyse mehrere Bildpunktvektoren.

Zunächst steht eine Analyse einer beliebig bekannten Kamerafahrt an, die bereits als 3D Objektpfad vorliegt, um die Anforderungen an das benötigte Sensorpaket zu definieren.

3.2. Anforderungsanalyse

Da inertielle Sensoren lediglich eine Objektbeschleunigung erfassen, wird die absolute Position über mehrere Rechenschritte postum erforderlich. Es stellt sich die Frage wie schnell eine Kamera während einer relevanten Aufnahme überhaupt beschleunigt? Kann man die minimale und maximale Beschleunigung bereits einschränken? Aus der Praxis lässt sich herleiten, dass sich eine auf dem Stativ befestigte Kamera eher subtiler als eine handgeführte Kamera bewegt, während eine Kamera an einem mobilen Fahrzeug befestigt, weitaus höhere Geschwindigkeiten und Beschleunigungen erreicht. Die Anwendung muss also differenziert werden.

Inertielle Sensoren werden für unterschiedlichste Anwendungsfälle konzipiert und haben demnach unterschiedliche Empfindlichkeiten und Laufzeitqualitäten. Für den Anwendungszweck mit einer Kamera soll zunächst die Anforderung an Messbereich und Empfindlichkeit untersucht werden.

Gesucht ist diesbezüglich die relative Beschleunigung einer Kamera. Um diese zu finden sind im Folgenden drei Kamerafahrten zu analysieren.

Als Fallbeispiele stehen neben einer virtuellen 3D Kamerafahrt (computer generiert) zwei reale Kamerafahrten zur Verfügung, die aus dem Matchmoving vorliegen. Da eine virtuelle Kamera zwar nicht mit Sensoren gemessen werden kann, soll sie dennoch als zusätzliches Beispiel herangezogen werden, um sie zu einem Vergleich zu den realen Kameras heranzuziehen.⁹

- Stativschwenk
- Handkameraführung in allen Achsen
- virtuelle 3D Kamerafahrt

Für die Verwendung in der Postproduktion müssen die Kamerakoordinaten in speziellen Dateiformaten vorliegen.

⁹ Daten der virtuellen Kamera kommen mit freundlicher Genehmigung von Mackevision Mediendesign GmbH

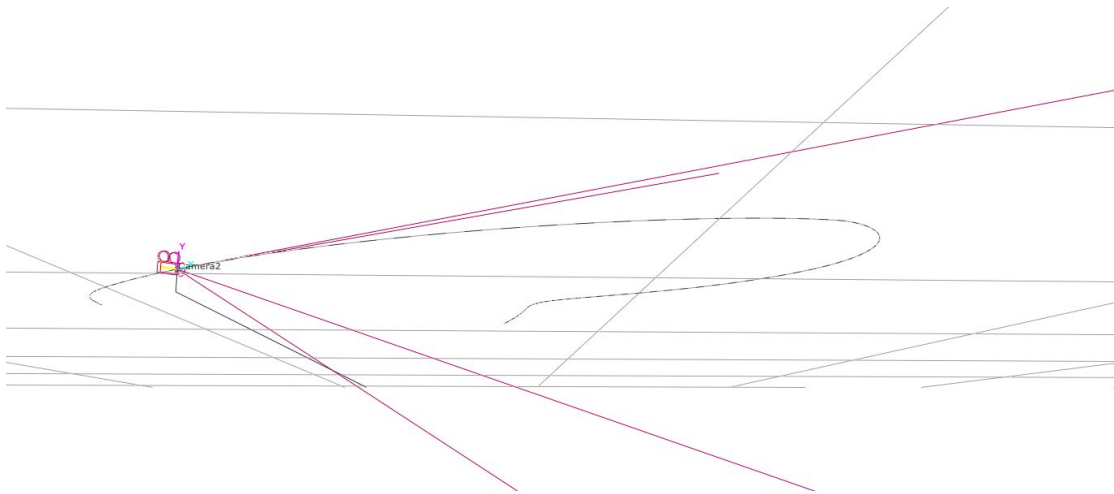


Abb. 3.2: virtueller Kamerapfad im 3D Raum

Gegeben ist hier das **Channel File**, das als Speicherformat für die 3D Kamera-
daten dient:

Eine Kamerafahrt kann in unterschiedlichsten Dateiformatierungen gespeichert
werden wobei nicht jedes Format mit jeder Software kompatibel ist. In dieser Arbeit
wird das universelle „Channel“ <.chan> Speicherformat verwendet.

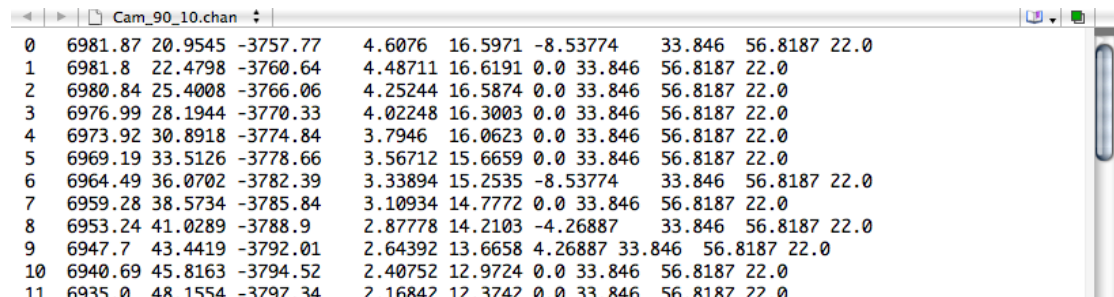


Abb. 3.3: Channel File <.chan>

Ein .chan Dateiformat gliedert die Positionswerte und Neigungswinkel zusammen
mit dem Bildwinkel und der Brennweite für jedes Frame in eine neue Zeile. Die Zah-
lenwerte werden lediglich durch Tabulatoren getrennt.

Frame	Translation (Position) in Pixel			Rotation Neigungswinkel Eulergrad (°/s)			Bildwinkel		Brennweite
	x	y	z	x	y	z	vertikal	horizontal	
1	6981.87	20.9545	-3757.77	4.6076	16.5971	-8.53774	33.8467	56.8187	22.0
2	6981.8	22.4798	-3760.64	4.48711	16.6191	0.0	33.8467	56.8187	22.0
3

Tabelle 3.1: Channel File Formatierung

Aus den gegebenen absoluten Positions- und Winkelkoordinaten der Kamerafahrt von je drei Achsen, folgt im nächsten Schritt die Berechnung der Beschleunigung.

Ermittlung der Beschleunigung zur Definition von Messbereich und Empfindlichkeit

Über die Berechnung der zurückgelegten Strecke s kann die Geschwindigkeit v pro Frame t propagiert werden und weiter die Beschleunigung a . Negative Ergebnisse sprechen für eine negative Beschleunigung, sprich einem Bremsmanöver. Die Einheit der Beschleunigung a soll letztendlich in g vorliegen, um einen Vergleich mit den Datenblättern der Sensoren zu ermöglichen.

$$s = \text{Strecke} = Pos_{neu} - Pos_{alt}$$

$$v = \text{Geschwindigkeit}$$

$$a = \text{Beschleunigung}$$

$$t = \text{Zeit}$$

$$v = Pos_t - Pos_{(t-1)} \quad (\text{Gl. 1})$$

$$a = v_t - v_{(t-1)} \quad (\text{Gl. 2})$$

$$\text{Umrechnung von } m/s^2 \text{ in } g = \frac{a \text{ in } m/s^2}{0,981 m/s^2} \quad (\text{Gl. 3})$$

Die Berechnung der Beschleunigung aus den absoluten Positionskoordinaten der drei besagten Channelfiles mit Gl. 1-3 findet mit Hilfe von Excel statt.¹⁰

Rechnauszug: Gesucht Beschleunigung a mit gegebenen Koordinaten:

$$x_{16} = -11.361;$$

$$x_{17} = -11.356;$$

$$x_{18} = -11.355$$

$$v_{17} = |-11.356 - (-11.361)| = 0.005$$

$$v_{18} = |-11.355 - (-11.356)| = 0.001$$

$$a = 0.005 - 0.001 = -0.004 m/s^2 / 9.81 = 0.0004 g$$

Die Ergebnisse sind in Tabelle 3.2 zusammengefasst.

¹⁰ Exceltabellen sind in elektronischer Form auf der Arbeits-DVD gespeichert „Anforderungsanalyse.ods“.

Konzept

max. Beschleunigung	x-Achse	y-Achse	z-Achse
virtuelle Kamera	0.295 g	0.155g	0.292 g
Stativschwenk	0 g	0 g	0 g
Handkamera	0.155 g	0.295 g	0.785 g
min. Beschleunigung je	0.001 g	0.001 g	0.001 g
Ø Beschleunigung	x-Achse	y-Achse	z-Achse
virtuelle Kamera	0.003 g	0 g	0.004 g
Stativschwenk	0 g	0 g	0 g
Handkamera	0g	0 g	0 g

Tabelle 3.2: Beschleunigungsergebnisse aus den Daten der drei Fallbeispiele

Der Kameraschwenk über das Stativ hat durch die Fixierung keinerlei Bewegungsfreiheit in den Translationsachsen, daher mit 0 g keine Beschleunigung. Die größte Beschleunigung findet mit 0.295 g statt, die geringste Beschleunigung ungleich 0 g misst 0.001 mg.

Eine durchschnittliche Beschleunigung (welche nach der Ruhestellung berechnet wird) von 0 g beweist, dass der Sensor wieder in eine Ruhestellung zu Bewegungsende kommt. Beschleunigungs- und Abbremsmanöver heben sich gegenseitig auf.

Die **Neigungswinkel** im Channel Dateiformat sind in Eulerwinkel beschrieben, entsprechend der Einheit wie sie aus einem Gyroskop ausgegeben werden können. Für die Empfindlichkeitsanalyse der Neigungswinkel ist lediglich die Differenz zwischen zwei Winkeln nach einem Frame zu berechnen.

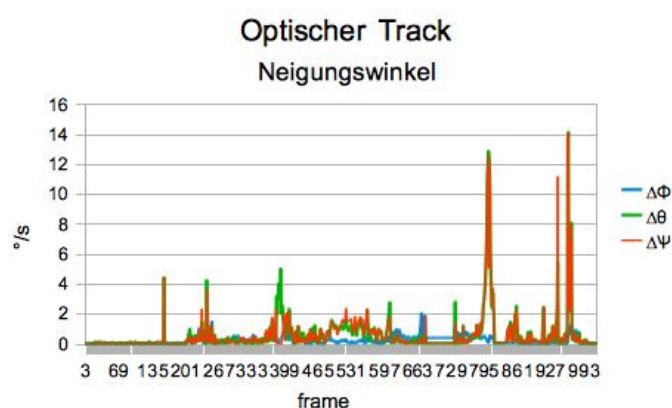


Abb. 3.4: Neigungswinkeldelta der Handkamera

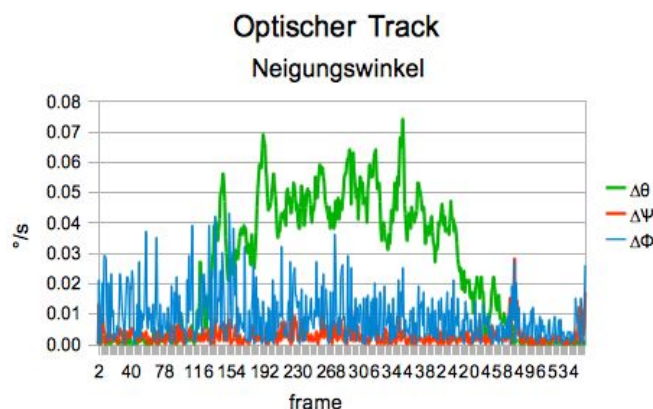


Abb. 3.5: Neigungswinkeldelta des Stativschwenks

Tabelle 3.3 listet für die drei Analysefälle die Maximalwerte der Winkeldifferenzen.

Die Minimal-Differenzen liegen im Ruhezustand bereits bei 0.001° .

max. Winkeldifferenzen	$\Delta\theta$ pitch	$\Delta\Psi$ yaw	$\Delta\Phi$ roll
virtuelle Kamera	0.268°	1.589°	0
Stativschwenk	0.074°	0.028°	0.043°
Handkamera	1.99°	14.04°	14.109°
Ø Winkeldifferenzen	$\Delta\theta$ pitch	$\Delta\Psi$ yaw	$\Delta\Phi$ roll
virtuelle Kamera	0.065°	0.948°	0°
Stativschwenk	0.022°	0.002°	0.006°
Handkamera	0.255°	0.638°	0.678°

Tabelle 3.3: Ergebnisse der Winkelempfindlichkeiten aus den Fallbeispielen

Die virtuelle Kamerafahrt lässt einen sehr homogenen Bewegungsablauf ohne Rauschen erkennen, was einer Messung in einem natürlichen Umfeld entgegengesetzt hervorhebt, dass immer minimale Schwankungen zu verzeichnen sind.

Nach der Auswertung drei typischer Kamerabewegungen lässt sich der IMU-Messbereich für die Beschleunigung zwischen -0.1 g und $+0.1\text{ g}$ einschränken. Die minimalste Beschleunigung liegt bei 0.001 g . Die Sensoren sollten folglich in der Lage sein, Beschleunigungen von 1 mg zu erfassen.

Anforderungen	Beschleunigungssensor	Gyroskop
Messbereich	$\pm 0.1\text{ g}$	$45\text{ }^{\circ}/s$
Auflösung	$\pm 1\text{ mg}$	$0.001^{\circ}/s/^{\circ}C$

Tabelle 3.4: Anforderungen an die Sensoren im Einsatz mit einer Kamera

Über die Anforderungen der Beschleunigungsmessung hinaus gilt es im Konzept noch weitere Punkte im Folgenden zu analysieren.

Abtastfrequenz

Ein weiterer entscheidender Wert für die Sensorwahl ist die Abtastrate. Wie oft kann der Sensor Messwerte erfassen und verarbeiten?

Die Abtastfrequenz lässt sich aus der Framerate (in fps) der Bildaufzeichnung ableiten, die sich häufig zwischen 20 und 60 Frames pro Sekunde bewegt. Andere durchaus übliche Frameraten für Slowmotion bzw. Hochgeschwindigkeits-Aufnahmen sollen im Rahmen dieser Arbeit nicht zur Anforderung zählen.

Übliche Frameraten:

60/50/30/29.979/25/24/23.976 fps

⇒ Abtastfrequenzbereich zwischen 20 und 60 Hz

$fps = Hz$

Synchronisation mit dem Timecode

Eine wichtige Frage stellt sich noch im Zusammenhang mit der Zeiterfassung: wie lässt sich ein Bild aus der Aufnahme mit dem Wert des Sensors synchronisieren? Wie ist sicherzustellen, dass Messwert x genau zu Bild x gehört? Kamera und Sensor arbeiten zwar im gleichen Zeitformat, benötigen aber noch eine korrekte Zuordnung eines Zeitstempels.

Dazu bietet sich am ehesten der Timecode der Kamera an, mit einem Frame als kleinste Einheit. Jedes Bild (hier 25 in jeder Sekunde) bekommt mit dem Timecode einen festen Zeitstempel.

Die Zeitsynchronisation findet bei Drehs mit zeitgleichem Einsatz mehrerer Kameras ihre Anwendung. In den meisten Fällen gibt ein „Timegenerator“ als Master den Timecode für alle Geräte vor, die daran angeschlossen werden. Die Übertragung des Master-Timecodes findet entweder über Kabelleitungen oder Funk statt.

Einige Kameras, insbesondere Broadcast- und Videokameras, sind auch in der Lage selbst als Master-Timecode-Generator zu agieren. Das Timecodesignal wird in diesem Fall standardgemäß über eine Kabelleitung übertragen.

Für den Thesis-Prototyp kommt ein Timecodesignal direkt aus der Kamera in Frage. Über ein Firewiresignal kann dieses in den Computer geschleust und über ein Programm abgefragt werden. Sobald der Sensor mit Strom versorgt wird, d.h. an den Computer angeschlossen wird, können beide Signale zusammengeführt werden.

Während die Kamera in Ruhestellung ist, steht der Timecode still und wird als solcher auch gespeichert. Wird eine Aufzeichnung gestartet läuft der Timecode weiter. Die Sensordaten außerhalb der Aufnahmezeit benötigen keiner weiteren Beachtung und können später extrahiert werden, sofern sie nicht für die spätere Berechnung zur Initialisierung benötigt werden.

3.3. Qualitätsmaßstab

Die Herleitung einer Kamera über Matchmoving Programme ist heutzutage, wie bereits in der Einleitung erwähnt, ein standardisierter Prozess mit dem eine hundertprozentige Replikation einer originalen Kamera möglich ist. Diese hundertprozentige Genauigkeit gilt es optimalerweise mit den Trägheitssensoren zu erreichen.

Doch in welchen Einheiten definiert sich eine hohe Genauigkeit? In welcher Größe darf sich davon gegebenenfalls entfernt werden, um noch stets „genau“ zu sein?

Das Ziel des Matchmovings ist die „unsichtbare“ Integration eines Elements in die getrackte Umgebung. Unsichtbar heisst in diesem Fall, dass sich das Element im gleichen Verhältnis zum originalen Bildausschnitt bewegt. Die Praxis zeigt folgende Erfahrung: Sobald sich ein Objekt im Vordergrund bereits ein Pixel unverhältnismäßig zum Rest des Bildes bewegt, fällt dem Zuschauer diese Unstimmigkeit negativ ins Auge, vgl. auch [DOB05], S. 73.

Die Abweichung der virtuellen Kamera dürfen im Bild demnach nicht mehr als ein Pixel ausmachen. Dies gilt sowohl für die Translationsergebnisse der x-, y- und z-Achsen als auch für die Neigungswinkel (Φ , θ , Ψ)

⇒ Fehleranfälligkeit < 1 Pixel

Ein Pixel in den Translationsachsen lässt sich im Pixel-Koordinatensystem leicht ins Verhältnis setzen, doch wie wirken sich Winkelfehler in Grad auf die Pixeleinheit aus?

Wie viel Grad entspricht ein Pixel im Bild?

Zur Beantwortung dieser Frage bedarf es der Kenntnis des horizontalen Bildwinkels, der sich je nach Kamera und Optik unterscheidet.

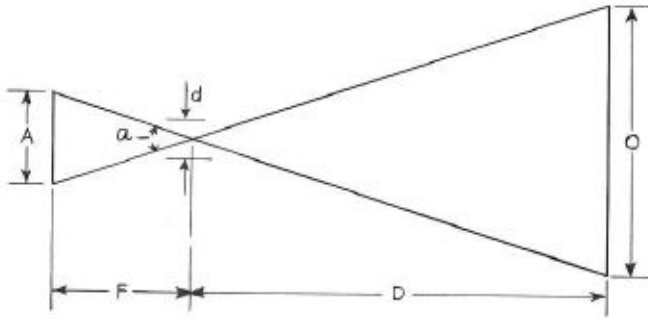


Abb. 3.6: Zusammenhang zwischen Objektgröße, Brennweite der Linse, Objektdistanz und Sensorgröße einer Kamera.

Der Bildwinkel, sog. Field of View FOV steht im Verhältnis zur Brennweite F und Sensorgröße A nach Gl. 4, vgl. [RYA93].

$$\frac{FOV}{2} = \arctan \frac{0.5 * A}{F} \quad (Gl. 4)$$

Ein Praxisbeispiel (1):

Mit einer horizontalen Sensorgröße von 4,4 mm ergibt sich nach Gl. 4 mit einer Brennweite von 32.5 mm ein horizontaler Bildwinkel von 7.7° .¹¹

$$\frac{FOV}{2} = \arctan \frac{0.5 * 4.4 \text{ mm}}{32.5 \text{ mm}} = \arctan(0.00676) = 3.87^\circ$$

$$FOV = 2 * 3.87 = 7.7^\circ$$

Ein Bildwinkel von 7.7° entspricht bei einer DVCPRO HD Auflösung von 1080i/25p, wie z.B. bei der Panasonic HVX 200E, 1440 Pixel in der Horizontalen.

$$7.7^\circ = 1440 \text{ px}$$

$$\Leftrightarrow 1^\circ = 187 \text{ px}$$

$$\Leftrightarrow 0.0053^\circ = 1 \text{ px}$$

Ein Pixelfehler im HD Bild entspricht demnach einem Winkelfehler von 0.0053° .

Bei einer gewählten 100 mm Brennweite entspricht der $FOV = 1.94^\circ$

¹¹ Die Variablen beziehen sich auf die Panasonic HVX 200E mit einer weitwinkligen Brennweite von 4.2 mm (äquivalenten 32.5 mm), die in der Praxis für diese Arbeit zum Einsatz kam.

$$\Leftrightarrow 0.0013^\circ = 1 \text{ px}$$

Je weitwinkliger die Brennweite, umso größer ist der Winkel, der 1 px repräsentiert.

Praxisbeispiel (2):

Betrachtung einer 35mm Auflösung mit der Sensorgröße von 24.92 mm x 18.67 mm und einer Pixelauflösung von 4153 x 3112 px:

$$\frac{FOV}{2} = \arctan \frac{0.5 * 24.92 \text{ mm}}{32.5 \text{ mm}}$$

$$FOV = 41.95^\circ$$

Ein größerer Sensor bietet gegenüber einem kleinen Sensor einen größeren Bildwinkel bei gleicher Brennweite. In Bezug auf das Pixelverhältnis entspricht 1 px gleich 0.01° .

Ergebnis bei einer horizontalen Auflösung von 4153 Pixeln:

$$41.95^\circ = 4153 \text{ px}$$

$$\Leftrightarrow 0.01^\circ = 1 \text{ px}$$

oder bei einer horizontalen Auflösung von 2048 Pixeln:

$$41.95^\circ = 2048 \text{ px}$$

$$\Leftrightarrow 0.02^\circ = 1 \text{ px}$$

Bei einer Brennweite von 32.5 mm entspricht in diesem Rechenfall ein Bildpixel bei einem Bildwinkel von 41.95° bei einer 4k Auflösung 0.01° und bei einer 2k Auflösung 0.02° .

Fazit: Je größer der Sensor und/oder je geringer die Pixeldichte, desto weitwinkliger wird ein Pixel repräsentiert, d.h. umso mehr Abbildungsinformation steckt in einem Pixel.

Nach der Erkenntnis des Verhältnisses von Aufnahmewinkel zum Bildpixel, stellt sich nun die Frage wie sich ein Positionsversatz in einer linearen Achse, sowohl in metrischer Einheit als auch im Bildpixel verhält?

Nachdem das Verhältnis von Pixelanzahl zur Sensorgröße der Kamera bekannt ist, lässt sich dieser Größenbezug auf das Verhältnis von Bildauflösung und reale Objektgröße definieren.

$$\frac{\text{Sensorgröße } A}{\text{Bildauflösung } Ab} = \frac{\text{Objektgröße } O}{\text{Objektauflösung } Oa} \quad (\text{Gl. 5})$$

Die Objektgröße O steht im Verhältnis zur Brennweite F, Sensorgröße A und Distanz D. Die Distanz bezeichnet die Entfernung des abzubildenden realen Objektpunktes auf den Kamerachip.

$$\frac{O}{A} = \frac{D}{F} \quad (\text{Gl. 6})$$

Wie groß ist ein Objekt, das mit einem Pixel reproduziert wird?

Welcher Größe entspricht ein reales Objekt in mm in der digitalen Abbildung auf einem Pixel?

$$O = \frac{D * A}{F} \quad (\text{Gl. 7})$$

Folgende Variablen sind gegeben:

Sensorgröße A = 4.4 mm

Brennweite F = 32.5 mm

Distanz D = 1000 mm

$$O = \frac{1000 \text{ mm} * 4.4 \text{ mm}}{32.5 \text{ mm}}$$

$$O = 135.3 \text{ mm}$$

Mit der horizontale Bildauflösung von 1440 Pixel, was in einem Meter Abstand 135.3 mm entspricht, kann die Objektgröße für ein Pixel berechnet werden.

$$\text{Objektgröße} = \frac{1 \text{ px} * 135.3 \text{ mm}}{1440 \text{ px}}$$

$$1 \text{ px} = 0.09 \text{ mm}$$

Ein Pixel-Versatz von einem Pixel bedeutet bei einer Objektentfernung von einem Meter 0.09 mm Versatz; analog dazu bei zehn Metern 0.9 mm¹²

Im Umkehrschluss bedeutet dies, wenn sich die Kamera um weniger als einen Millimeter in der x-Achse verschiebt, verschiebt sich ein Objekt in 10 m Entfernung um ein Pixel.

¹² Die Berechnung liegt den Variablen der Panasonic HVX 200E zu Grunde

3.4. Fallunterscheidung

Um eine Beurteilung über die Verwendungsmöglichkeiten der IMU Messdaten für die Postproduktion abzugeben, bedarf es unterschiedlicher Untersuchungen. Es soll getestet werden unter welchen Umständen eine Sensormessung sinnvoll wird bzw. überhaupt funktioniert. Die Tests werden am Evaluationsset mit einer Kamera durchgeführt. Die Bilder lassen sich danach in ein Matchmoving Programm importieren; eine virtuelle Kamera steht den aufbereiteten Messdaten zum Vergleich gegenüber.

Es sind zwei Testszenarien für die Evaluation im Studio mit Kamera vorgesehen, die im Folgenden beschrieben sind.

Im ersten Fall handelt es sich um eine kurzzeitige Aufnahme mit Stativ. Es soll ein subtiler Schwenk auf nahezu einer Achse sein. Dieser Fall soll verdeutlichen wie viel Bewegung über den Sensor wahrgenommen wird. Die Bewegungsanalyse mit einem Matchmovingprogramm unter diesen Umständen ist einfach und unkompliziert.¹³ Deshalb ist hier zusätzlich die Frage ob sich der IMU Einsatz in einem solcher Fall lohnt?

Ein schneller, reisserischer Schwenk dahingegen ist kaum eine gute Voraussetzung für ein einfaches Matchmoving, da hier meist nicht genügend Referenzpunkte über eine Bildabfolge verbleiben. Motionblur macht das Tracking zunehmend erschwerlicher. Gleichfalls sind wilde Kamerabewegungen ohne feste Achse und keine Gleichmäßigkeit im Bewegungsablauf eine Trackingherausforderung. Diese beiden Szenarien sollen als zweite Falluntersuchung den Einsatz von inertialen Sensoren am Filmset rechtfertigen. Eine lange Aufnahmedauer von ca. 50 Sekunden soll in diesem Fall zusätzlich fokussiert werden, da die bekannte Fehleranfälligkeit von inertialen Sensoren im Laufe der Zeit ansteigt.

Die zwei Szenarien decken letztlich extreme Einsatzmöglichkeiten der Sensoreinheit im Rahmen eines einfachen Filmdrehs und dieser Abschlussarbeit ab.

Fallunterscheidung für den Dreh:

1. Fall: subtiler Schwenk auf dem Stativ
2. Fall: Handkameraführung über 50 Sekunden inkl. Reisschwenk

¹³ Voraussetzung für ein einfaches Matchmoving sind offensichtlich gut erkennbare und in einer möglichst hohen Anzahl erforderlichen Trackingpunkte.

4. Prototypentwicklung

4.1. IMU Auswahl

Die Wahl des IMU für die Thesisevaluation fällt nach Analyse der Anforderungen in Kapitel 3.2 auf den „9 Degrees of Freedom - Razor IMU“ von Sparkfun¹⁴.

Produktbeschreibung nach [SPA11]:

„Die Sensoreinheit besteht aus vier Sensoren – ein 1-Achsen Gyroskop, 2-Achsen Gyroskop, 3-Achsen Beschleunigungssensor und einem 3-Achsen Magnetometer. Die Messungsergebnisse werden von einem integrierten Atmega328 Prozessor verarbeitet und über eine serielle Schnittstelle ausgegeben. Mit der Arbeit von u.a. Jordi Munoz¹⁵ lässt sich die Sensoreinheit zu einem Attitude and Heading Reference System (AHRS) erweitern. [...] Der 9DoF arbeitet mit einer 3.3VDC Spannung; jegliche Stromspeisung über den weißen JST Schnittstelle wird auf die Betriebsspannung heruntergeregelt. Der Ausgabekopf ist auf das 3.3V FTDI Basic Breakout Board zugeschnitten, um es unkompliziert über ein USB Kabel an den Computer anzuschließen.“



Abb. 4.1: IMU von Sparkfun

Die Bezeichnung mit neun Freiheitsgraden mag irritierend sein, da es physikalisch nur sechs Freiheitsgrade gibt. Die zusätzlichen drei Achsen resultieren eher aus dem Marketing über die Addition der drei Magnetfeldsensoren.

Die wichtigsten Eigenschaften des 9DoF Razor IMU's sind in Tabelle 4.1 zusammengefasst. Datenblätter der einzelnen Sensoren finden sich im Anhang dieser Arbeit. Das Grundrauschen des Beschleunigungssensors wird bei einer Datenrate von 100 Hz mit weniger als 1 LSB rms¹⁶ angegeben, d.h. mit weniger als 4 mg.

Die Nichtlinearität bezieht sich auf ein Prozentwert des gesamten Messbereichs von 2 g.

¹⁴ Sparkfun ist ein amerikanischer Händler für Elektronik <http://www.sparkfun.com>

¹⁵ Google Project Code Seite nach der Creative Commons 3.0 Lizenz; vgl.[MUN11]

¹⁶ rms = Root Mean Square

Hersteller	sparkfun		
Modell	9DOF Razor		
Sensorart	ADXL345 Beschleunigung	LY530ALH Gyroskop	LPR530AL Gyroskop
Messbereich	± 2 g	300 °/s	300 °/s
Rauschlevel	< 1.0 LSB rms x-,y-Achse (< 4 mg) < 1.5 LSB rms z-Achse (< 6 mg)	0.035 °/s/√Hz	0.035 °/s/√Hz
Linearitätsfehler vom Messbereich	< 0.5 %	±1 %	±1 %
Bandbreite ¹⁷	6.25 - 3200 Hz	140 Hz	140 Hz
Abmessungen	49.53 x 27.94 mm		
Typ	MEMS		
Kosten	91 EUR		
Quellverweis	http://www.sparkfun.com/products/9623 Vgl. [SPA11]		

Tabelle 4.1: Spezifikationen 9DoF Razor IMU

Das 9DOF Razor Board bietet einen 8MHz Arduino Bootloader inkl. Firmware zur Durchführung eines Selbsttests. Voraussetzung ist ein Breakoutboard für den Datentransport zum Computer und ein Arduino Programm Terminal.

„Die Arduino-Plattform ist eine aus Soft- und Hardware bestehende Physical-Computing-Plattform. Beide Komponenten sind im Sinne von Open Source quelloffen. Die Hardware besteht aus einem einfachen I/O-Board mit einem Mikrocontroller und analogen und digitalen Ein- und Ausgängen. Die Entwicklungsumgebung beruht auf Processing (einem Java-Dialekt) und Wiring (einem C-Dialekt), die insbesondere Künstlern, Designern, Hobbyisten und anderen Interessierten den Zugang zur Programmierung und zu Mikrocontrollern erleichtern soll. Arduino kann benutzt werden, um eigenständige interaktive Objekte zu steuern oder um mit Softwareanwendungen auf Computern zu interagieren. Arduino wird intensiv an Kunsthochschulen genutzt, um interaktive Installationen aufzubauen.“[ARD11]¹⁸

¹⁷ Die Bandbreite ist die Hälfte der Ausgabefrequenz

¹⁸ Weiterführende Informationen zu Arduino auf der offiziellen Webpräsenz; vgl. [AOW11]

Da in dieser Arbeit besonderer Fokus auf die Umsetzung und die Ergebnisanalyse liegt, und weniger auf die Programmierung einer Sensoreinheit, eignet sich der 9DoF Razor insbesondere im Zusammenhand mit dem vorhandenen AHRS Code. Die gute Dokumentation des Codes ermöglicht auch mit geringen Programmierkenntnissen eine Adaption der Programmierzeilen. Die Modifikation für den individuellen Einsatz an einer Filmkamera wird im folgenden Kapitel beschrieben.

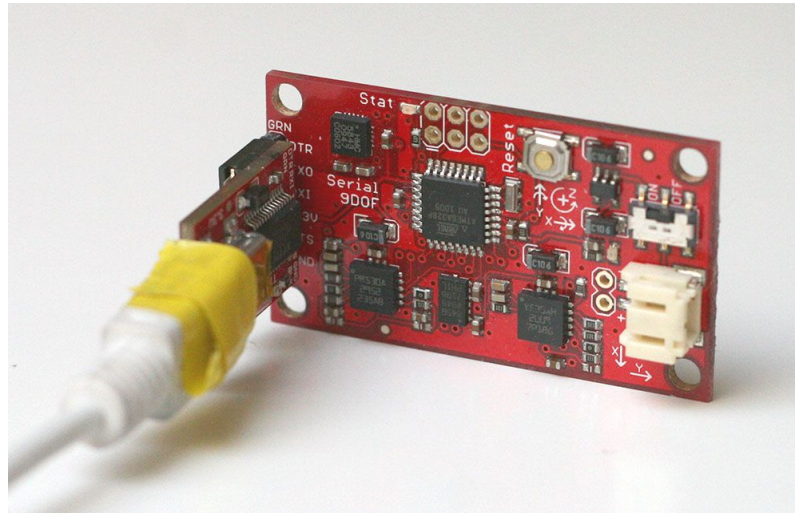


Abb. 4.2: Prototyp Sparkfun Sensor mit USB-Breakoutboard

4.2. IMU Programmierung mit Arduino

Zur Speicherung von Messergebnissen bedarf es einiger Programmierung bevor die Sensoreinheit einsatzbereit ist.

Für den 9DoF Razor wurde im Februar 2010 ein Arduino-Code unter der Creative Common Lizenz im Internet veröffentlicht; vgl. [MUN11]. Dieser Code ist insbesondere für die Steuerung von unbemannten Fahrzeugen und zur Bildstabilisierung vorgesehen. Für den Einsatz an einer Kamera sind lediglich die Messergebnisse der Sensoren interessant. Die rohen Sensordaten werden im AHRS-Code zusätzlich über die Magnetfeldmessung Fehler korrigiert.

Abänderung des Codes für den Einsatz an der Kamera:

Die Bildaufzeichnung mit der Kamera wird für die Evaluierung mit 25 Bildern pro Sekunde definiert. Die Abtastfrequenz der Sensoren ist folglich auf 25 Hz zu setzen, um zu jedem Bild einen Messwert zu erhalten.

SF9DOF_AHRS	ADC	Compass	DCM	I2C §	Output	Vector	matrix
-------------	-----	---------	-----	-------	--------	--------	--------

```

/* ***** */
/* I2C code for ADXL345 accelerometer */
/* and HMC5843 magnetometer */
/* ***** */

int AccelAddress = 0x53;
int CompassAddress = 0x1E; //0x3C //0x3D; //(0x42>>1);

void I2C_Init()
{
  Wire.begin();
}

void Accel_Init()
{
  Wire.beginTransmission(AccelAddress);
  Wire.send(0x2D); // power register
  Wire.send(0x08); // measurement mode
  Wire.endTransmission();
  delay(5);
  Wire.beginTransmission(AccelAddress);
  Wire.send(0x31); // Data format register
  Wire.send(0x08); // set to full resolution +-2g
  Wire.endTransmission();
  delay(5);
  Wire.beginTransmission(AccelAddress);
  Wire.send(0x2C); // Rate
  Wire.send(0x09); // set to 25Hz
  Wire.endTransmission();
  delay(5);
}

```

Abb. 4.3: Arduino Codeauszug Registerbelegung

Im Arduino Code wird diesbezüglich eine Codezeile bearbeitet, welche das Register für die Datenrate füllt.

Im Datenblatt des Sensors lässt sich anhand der Tabellen der dezimale Wert für 25 Hz ablesen und in eine Hexadezimalzahl konvertieren (00011000 = 0x08). Zu beachten sind in diesem Hex Wert auch die restlichen Register Bits, da jedes Bit einer Bedeutung zukommt! Die Bandbreite der Gyroskope ist fix, sie wird über die Zeitschleife von 140 Hz auf 25 Hz herabgestuft.

```
if((millis()-timer)>=40
```

Alle 40 Millisekunden (1/25 Sekunden) werden Messwerte abgefragt.

SF9DOF_AHRS_02	ADC	Compass	DCM	I2C	Output	Vector	matrix
----------------	-----	---------	-----	-----	--------	--------	--------

```

void loop() //Main Loop
{
  if((millis()-timer)>=40) // Main loop runs at 25Hz
  {
    counter++;
    timer_old = timer;
    timer=millis();
    if (timer>timer_old)
      G_Dt = (timer-timer_old)/1000.0; // Real time of loop run. We us
    else
      G_Dt = 0;

    // *** DCM algorithm
    // Data adquisition
    Read_adc_raw(); // This read gyro data
    Read_Accel(); // Read I2C accelerometer

    if (counter > 2) // Read compass data after 2 loops only|
    {
      counter=0;
      Read_Compass(); // Read I2C magnetometer
      Compass_Heading(); // Calculate magnetic heading
    }

    // Calculations...
    Matrix_update();
    Normalize();
    Drift_correction();
    Euler_angles();
    // ***

    printdata();
  }
}

```

Abb. 4.4: Main Loop – Schleife mit 25 Hz Abtastfrequenz

Die **Kalibrierung** des Boards findet zu Beginn der Stromspeisung statt. Hier ist es notwendig, das Board bereits vorher in Ruhestellung zu bringen. Die Software berechnet einen Mittelwert der ersten 32 Werte eines jeden Messwerts, um so das interne Rauschverhalten zu minimieren. Ausserdem wird die Schwerkraft der Erde berechnet, die ein Sensor natürlich als statische Beschleunigung wahrnimmt.

Offset: 380, 381, 384, 1, -4, -6

Der Sensor gibt Beschleunigungswerte in internen Einheiten „LSB“ aus. Die Empfindlichkeit des Sensors liegt laut Datenblatt bei einem Mittelwert von 3.9 mg pro LSB-Einheit.

Die Testmessung im Ruhezustand des IMUs ergibt nun auf der z-Achse (Vertikale) ein Wert von 248 LSB. In diesem Zustand ist davon auszugehen, dass nur die Gravitation gemessen wird, die sich nach Newtons Kraftgesetz auf 1 g (= 1000 mg) definiert.

Getesteter Wert: 248 LSB \approx 1000 mg

$$\frac{1000 \text{ mg}}{248 \text{ LSB}} = 4,03 \text{ mg}$$

Nach dieser Testmessung kann die Empfindlichkeit des Beschleunigungssensors auf 4 mg festgesetzt werden.

Um auf die Beschleunigungseinheit in g zu kommen wird jede LSB Einheit durch 248 dividiert. Diese Umrechnung soll jedoch erst später nach dem Messvorhaben

stattfinden, um die rohen Sensordaten beizubehalten.

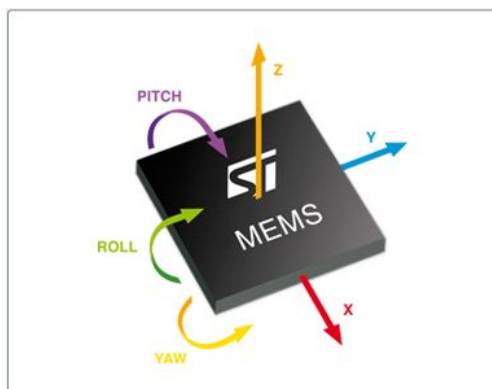


Abb. 4.5: MEMS-Achsenzuordnung

Die Neigungswinkel werden über eine diskrete Cosinus Matrix berechnet und über Magnetfeldmessungen Fehler korrigiert. Die finalen Ausgabewerte sind in Eulerwinkel beschrieben.

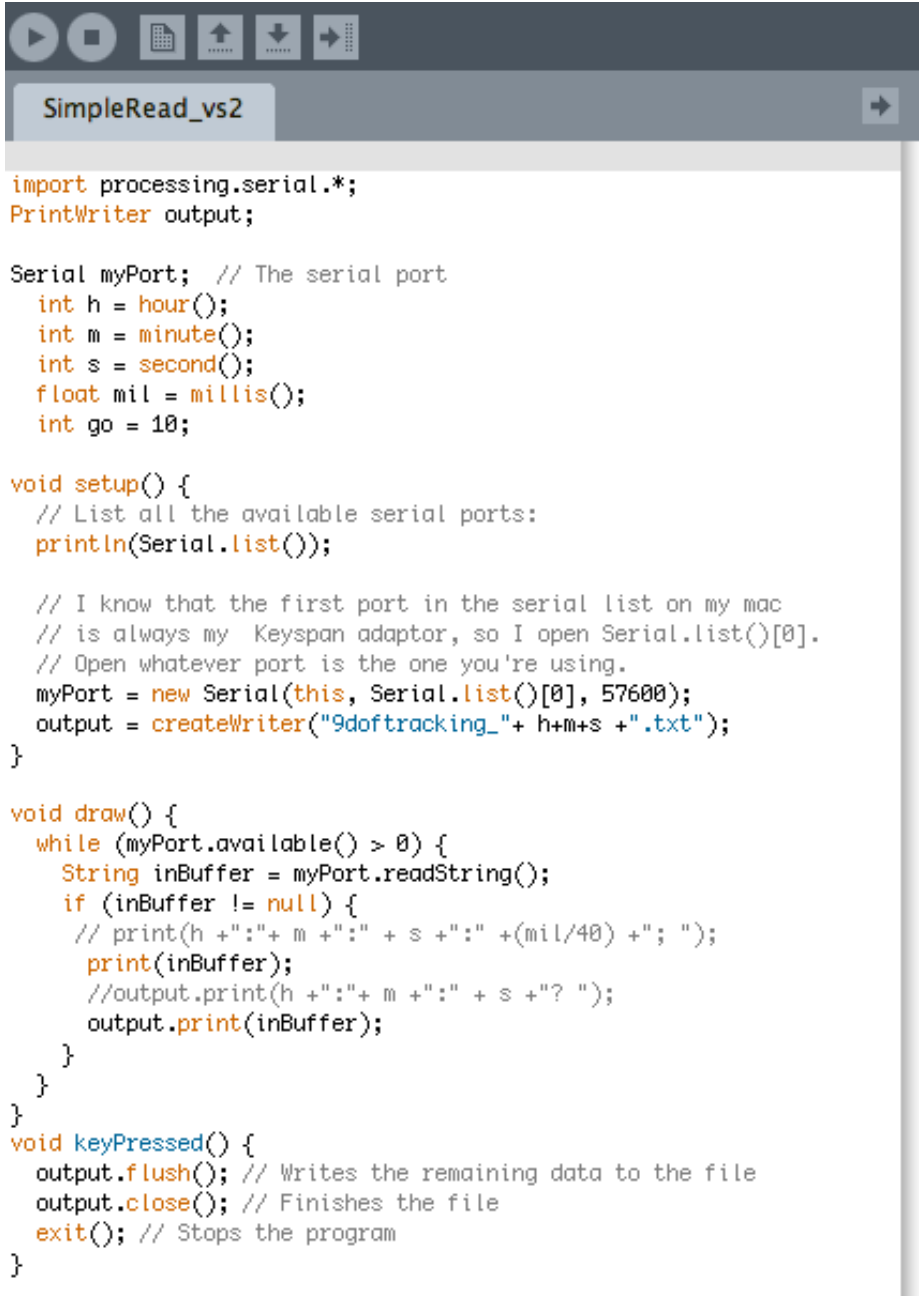
4.3. Datenspeicherung mit Processing

Arduino erhält über die serielle Schnittstelle, die über ein Terminal beobachtet wird, die Messwerte des Sensors. Diese Werte sollen in sinnvoller Formatierung als einfache Textdatei gespeichert werden. Da Arduino keine Möglichkeit bietet eigene Dateien anzulegen, wird die Programmiersprache Processing zusätzlich herangezogen. Ein kleines Programm „SimpleRead.pde“ überwacht die serielle Schnittstelle und speichert alle übertragenen Daten in eine Textdatei.

Wie soll nun die perfekte Formatierung der Werte aussehen?

Sinnvoll ist nach jeder Zeiteinheit – einem Frame eine neue Zeile zu eröffnen.

Einzelne Werte werden mit einem Komma getrennt, um sie in der Postproduktion leichter auszulesen.



```
import processing.serial.*;
PrintWriter output;

Serial myPort; // The serial port
int h = hour();
int m = minute();
int s = second();
float mil = millis();
int go = 10;

void setup() {
  // List all the available serial ports:
  println(Serial.list());

  // I know that the first port in the serial list on my mac
  // is always my Keyspan adaptor, so I open Serial.list()[0].
  // Open whatever port is the one you're using.
  myPort = new Serial(this, Serial.list()[0], 57600);
  output = createWriter("9doftracking_"+ h+m+s +".txt");
}

void draw() {
  while (myPort.available() > 0) {
    String inBuffer = myPort.readString();
    if (inBuffer != null) {
      // print(h + ":" + m + ":" + s + ":" + (mil/40) + "; ");
      print(inBuffer);
      //output.print(h + ":" + m + ":" + s + "? ");
      output.print(inBuffer);
    }
  }
}

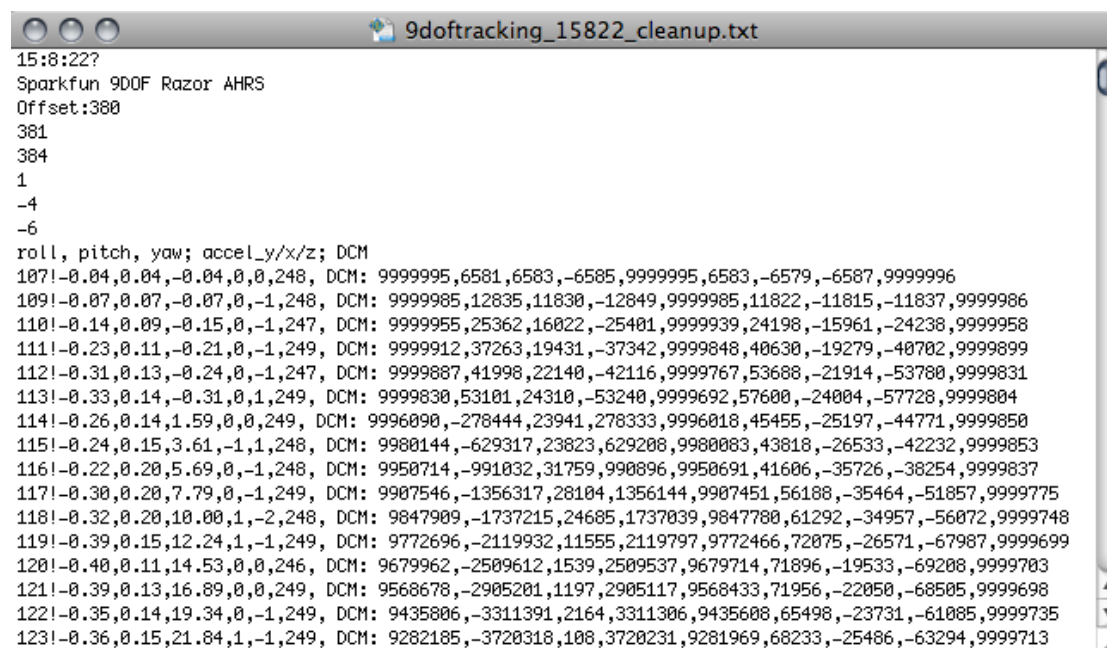
void keyPressed() {
  output.flush(); // Writes the remaining data to the file
  output.close(); // Finishes the file
  exit(); // Stops the program
}
```

Abb. 4.6: Processing Code zur Speicherung der Sensordaten in einer Textdatei

Zeilenformatierung der Ausgabe in einem Textfile:

```
1243!-0.58,-0.01,133.71,1,-2,249, DCM: -6910686,-7227533,-72061,7227891,  
-6910329,-70229,962,-100618,9999494
```

Die Beschleunigungs- und Neigungswinkelwerte sind nun in einer formatierten Textdatei inkl. Zeitstempel abgespeichert und bedürfen einer weiteren Bearbeitung, um auf die Trajektorie des IMUs zu schließen.



```
15:8:22?  
Sparkfun 9DOF Razor AHRS  
Offset:388  
381  
384  
1  
-4  
-6  
roll, pitch, yaw; accel_y/x/z; DCM  
107!-0.04,0.04,-0.04,0,0,248, DCM: 9999995,6581,6583,-6585,9999995,6583,-6579,-6587,9999996  
109!-0.07,0.07,-0.07,0,-1,248, DCM: 9999985,12835,11830,-12849,9999985,11822,-11815,-11837,9999986  
110!-0.14,0.09,-0.15,0,-1,247, DCM: 9999955,25362,16022,-25401,9999939,24198,-15961,-24238,9999958  
111!-0.23,0.11,-0.21,0,-1,249, DCM: 9999912,37263,19431,-37342,9999848,40630,-19279,-40702,9999899  
112!-0.31,0.13,-0.24,0,-1,247, DCM: 9999887,41998,22140,-42116,9999767,53688,-21914,-53780,9999831  
113!-0.33,0.14,-0.31,0,1,249, DCM: 9999830,53101,24310,-53240,9999692,57600,-24004,-57728,9999804  
114!-0.26,0.14,1.59,0,0,249, DCM: 9996090,-278444,23941,278333,9996018,45455,-25197,-44771,9999850  
115!-0.24,0.15,3.61,-1,1,248, DCM: 9980144,-629317,23823,629208,9980083,43818,-26533,-42232,9999853  
116!-0.22,0.20,5.69,0,-1,248, DCM: 9950714,-991032,31759,990896,9950691,41606,-35726,-38254,9999837  
117!-0.30,0.20,7.79,0,-1,249, DCM: 9907546,-1356317,28104,1356144,9907451,56188,-35464,-51857,9999775  
118!-0.32,0.20,10.00,1,-2,248, DCM: 9847909,-1737215,24685,1737039,9847780,61292,-34957,-56072,9999748  
119!-0.39,0.15,12.24,1,-1,249, DCM: 9772696,-2119932,11555,2119797,9772466,72075,-26571,-67987,9999699  
120!-0.40,0.11,14.53,0,0,246, DCM: 9679962,-2509612,1539,2509537,9679714,71896,-19533,-69208,9999703  
121!-0.39,0.13,16.89,0,0,249, DCM: 9568678,-2905201,1197,2905117,9568433,71956,-22050,-68505,9999698  
122!-0.35,0.14,19.34,0,-1,249, DCM: 9435806,-3311391,2164,3311306,9435608,65498,-23731,-61085,9999735  
123!-0.36,0.15,21.84,1,-1,249, DCM: 9282185,-3720318,108,3720231,9281969,68233,-25406,-63294,9999713
```

Abb. 4.7: Textdatei mit Rohdaten aus dem Sensor

4.4. Datenauswertung mit Excel

Für die Herleitung der absoluten Positionswerte im Koordinatensystem muss zunächst der Bewegungspfad ermittelt werden, bevor dieser mit dem Offset der Nullpunktkoordinaten addiert werden kann.

Mit Hilfe des Strapdown-Algorithmus wird die Lage durch Integration der Drehraten propagiert während die Position durch jeweilige Integration von Beschleunigung und Geschwindigkeit propagiert wird.

Die rohen Sensorwerte werden aus der Textdatei in die Exceltabelle kopiert.

Die Neigungswinkel werden bereits in Eulerwinkel ausgegeben und bedürfen keiner weiteren Berechnung.

Berechnung der Translationskoordinaten x/y/z:

Gegeben sind jeweils die Beschleunigungswerte in drei Achsen. Von der z-Beschleunigung wird zusätzlich die Gravitation (- 248 LSB) abgezogen, da diese im 3D Raum keiner Bedeutung zukommt. Der Anfangszustand definiert über eine Geschwindigkeit von 0 und einer Urposition von 0 im Ruhezustand.

Gesucht ist der zurückgelegte Weg D pro Frame in LSB bzw. Meter pro Sekunde im Quadrat.

Die Geschwindigkeit v kumuliert sich aus der Beschleunigung a mit

$$v = v_{(t-1)} + a_t \quad (Gl. 8)$$

und der Weg ergibt sich aus

$$D = v_{(t-1)} + v_t \quad (Gl. 9)$$

Die Umrechnung aus der LSB Einheit in m/s^2 erfolgt mit

$$d = \frac{D_{(x/y)}}{248} * 9.81 \text{ m/s}^2 \quad (Gl. 10)$$

Die Teilung durch 248 LSB geht auf den Skalenfaktor von 4 mg pro LSB zurück.

5. Evaluation

5.1. Selbsttest

Die Sensoreinheit wird über ein USB Kabel mit dem Computer verbunden. Processing speichert die über den seriellen Eingang kommenden LSB Werte in einer Textdatei. Die Schwerkraft ist aus der z-Achse noch nicht extrahiert und wird postum in Excel abgezogen (-248 LSB). Es werden drei gleichwertige Versuche durchgeführt.

Die Stromzufuhr wird vor jeder Testmessung unterbrochen, um den Sensor zu reseten, so werden Driftfehler die über einen langen Zeitraum entstehen ausgeschlossen.

5.1.1. Sensor in Ruhestellung

Um das Rauschverhalten der Sensoren zu messen, soll zunächst eine Messung ohne Bewegung stattfinden.

Die Laufzeitlängen der Tests bewegen sich zwischen 13 Sekunden und fünf Minuten.

Optimalerweise sollten die Ergebniswerte in allen Achsen bei 0 LSB liegen.

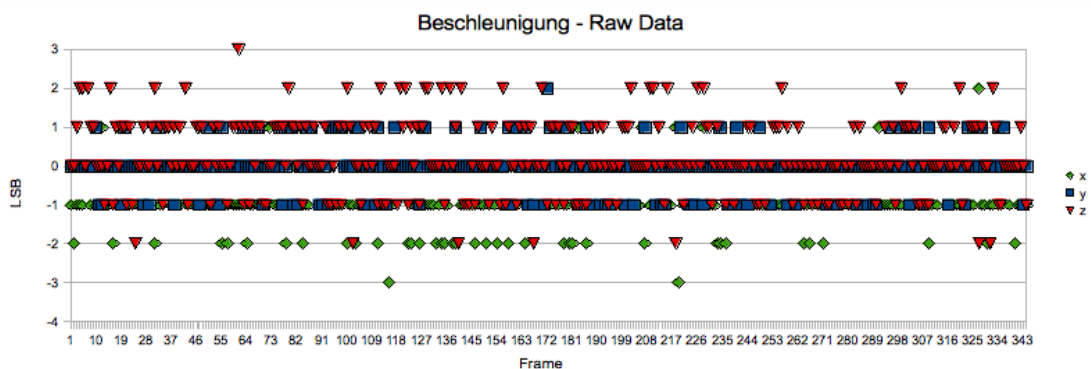


Abb. 5.1: Ruhestellung 2 – Beschleunigungsdaten (Laufzeit 13 Sekunden)

In der Realität sind in den Translationsergebnissen Abweichungen vom Optimalwert zu erkennen. Abb. 5.1 verdeutlicht als Auszug aus der Versuchsreihe Abweichungen von bis zu 3 LSB in jeder Achse. Diese Abweichungen sind konstant in jeder Testmessung zu verzeichnen und kann als Grundrauschen betrachtet werden.

Zum Verständnis ist in Formel 11 eine Umrechnung in g und m/s^2 vorgenommen.
 Es gilt $248 \text{ LSB} = 1 \text{ g} = 9.81 \text{ m/s}^2$

$$1 \text{ LSB} = \frac{1}{248} = 0.004 \text{ g} * 9.81 \text{ m/s}^2 = 0.04 \text{ m/s}^2 \quad (\text{Gl. 11})$$

Das Grundrauschen hat in der Berechnung der Trajektorie zur Folge dass sich diese Abweichungen kumulieren, vgl. 5.2.

Nach Ermittlung der Geschwindigkeit und der Positionskoordinaten pro Frame wird der Drift bereits nach weniger als zwei Sekunden sehr deutlich.

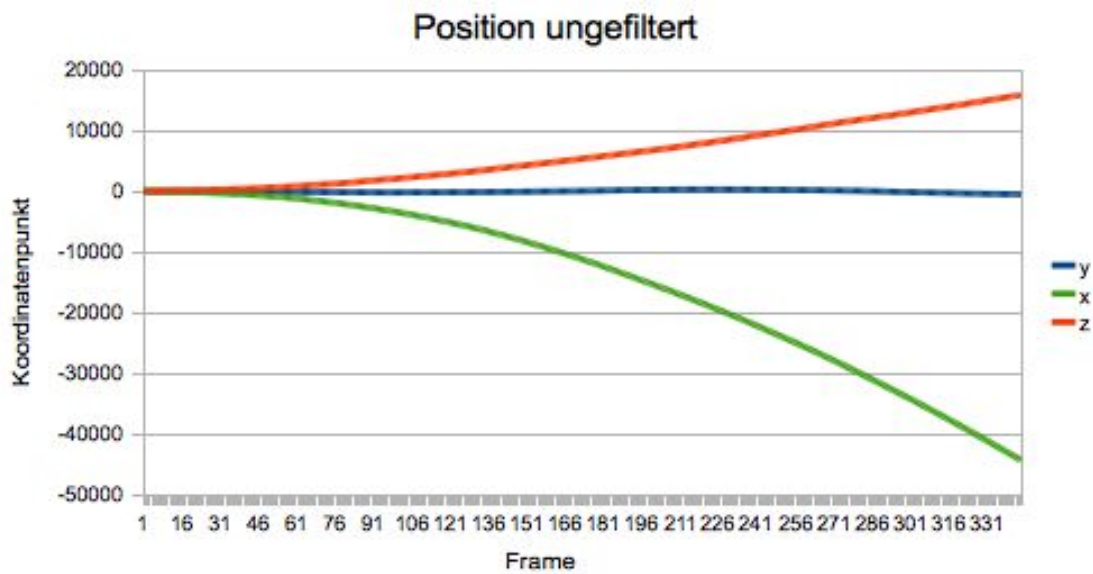


Abb. 5.2: Ruhestellung 2 – ungefilterte Position

Als Gegenmaßnahme bietet sich hier eine Filterung der originalen Sensordaten an. Abb. 5.3 zeigt die korrigierten Positionskoordinaten nachdem Werte, die größer als 7 LSB oder kleiner als -7 LSB mit 0 LSB ersetzt werden.

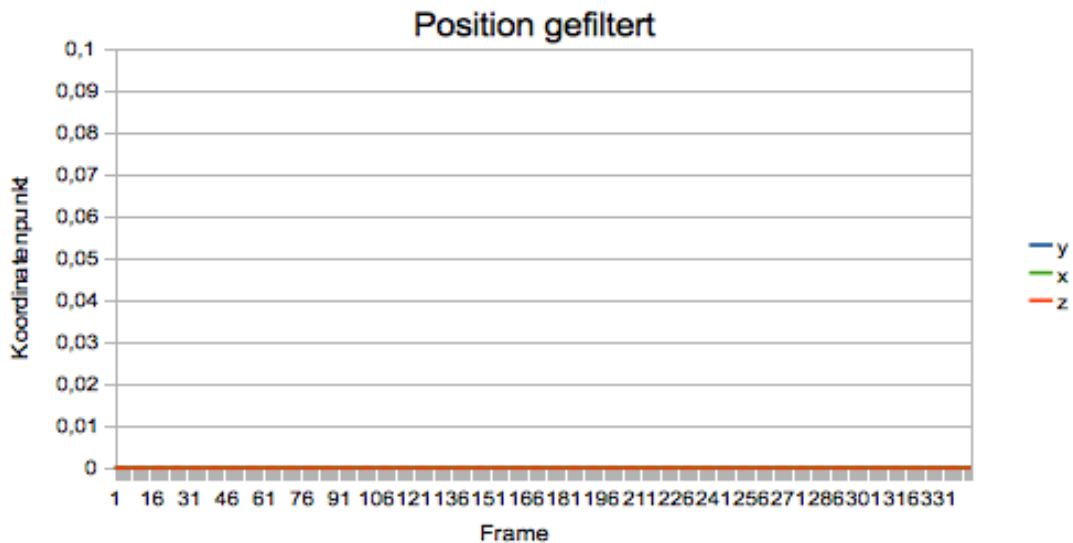


Abb. 5.3: Ruhestellung 2 – gefilterte Position

Der Langzeittest über fünf Minuten zeigt kaum andere Ergebnisse als nach wenigen Sekunden. Die Werte pendeln konstant zwischen ± 3 LSB. Erst mit einer Filterung ist eine Positionsbestimmung im Ursprung möglich. Andernfalls bewegen sich Geschwindigkeit und Position in utopische Werte wie z.B. -24.771.405 in der x-Achse.

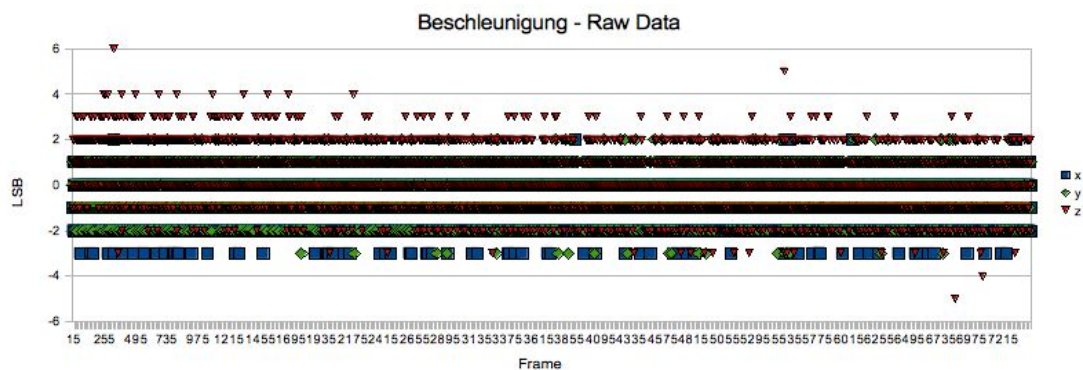


Abb. 5.4.: Ruhestellung 3 – Beschleunigung (Laufzeit 5 min)

Abb. 5.5 zeigt die Evaluationsergebnisse der **Neigungswinkel** über fünf Minuten. Bei der Betrachtung fällt bereits nach wenigen Sekunden eine kontinuierlich steigende Abweichung vom Idealwert auf. Die Werte pendeln sich erst nach ca. 40 Sekunden auf einen Durchschnittswert ein. Der Yawwinkel weist neben dem anfänglichen Drift ein weniger kontinuierliches Rauschen als Roll- und Pitchwinkel auf.

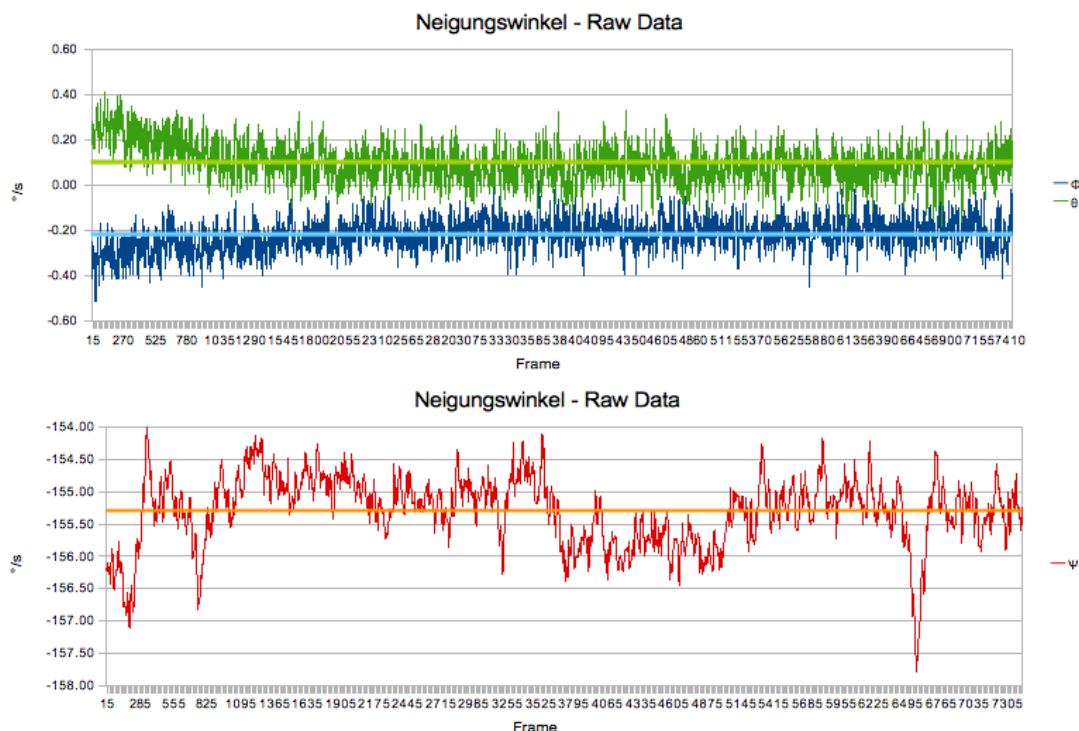


Abb. 5.5.: Ruhestellung 3 – Neigungswinkel (Laufzeit 5 min)

Rauschverhalten:

- $\pm 0.1^\circ/\text{s}$ Φ -Abweichung vom Durchschnittswert
- $\pm 0.1^\circ/\text{s}$ θ -Abweichung vom Durchschnittswert
- $\pm 0.5^\circ/\text{s}$ Ψ -Abweichung vom Durchschnittswert

5.1.2. Sensor in Bewegung

Der Sensor wird für die Evaluation in einer Achsenrichtung bewegt. Im Rahmen einer freihändigen Bewegung wurde versucht x- und z-Achse zu fixieren. Nach der Initialisierung des Sensors wird der Sensor auf der y-Achse 30 cm verschoben und wieder in Ruhestellung gebracht, danach auf gleichem Wege an seine Ursprungsposition zurück versetzt.

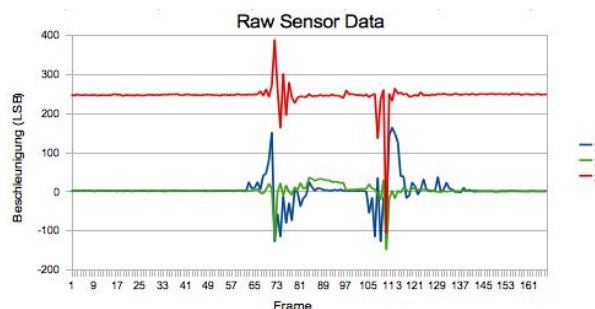


Abb. 5.6: y-move: Raw Sensor Data in LSB

Im Ergebnis sollten die Positionskoordinaten optimalerweise am Ende der Zeit wie zu Anfang auf Null kommen.

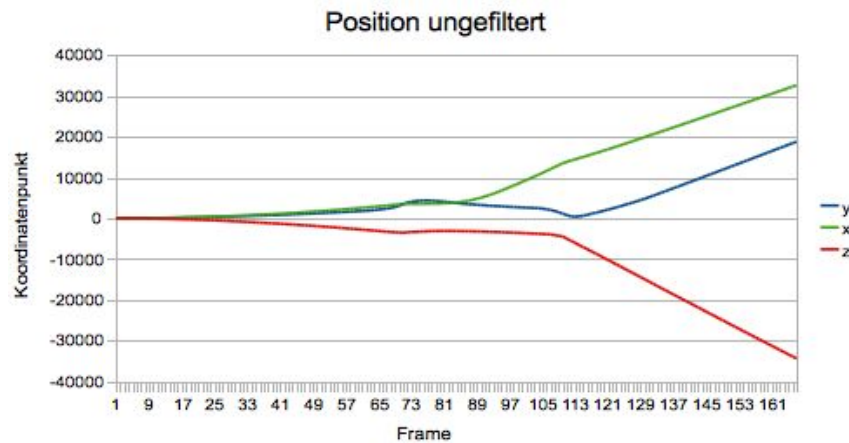


Abb. 5.7: y-move 4: Position ungefiltert

Abb. 5.7 zeigt jedoch den deutlichen Drift der berechneten Positionswerte, die aus dem hohen Rauschverhalten der Sensoren resultieren.

In Abb. 5.8 wird der Drift bereits in der Geschwindigkeit deutlich.

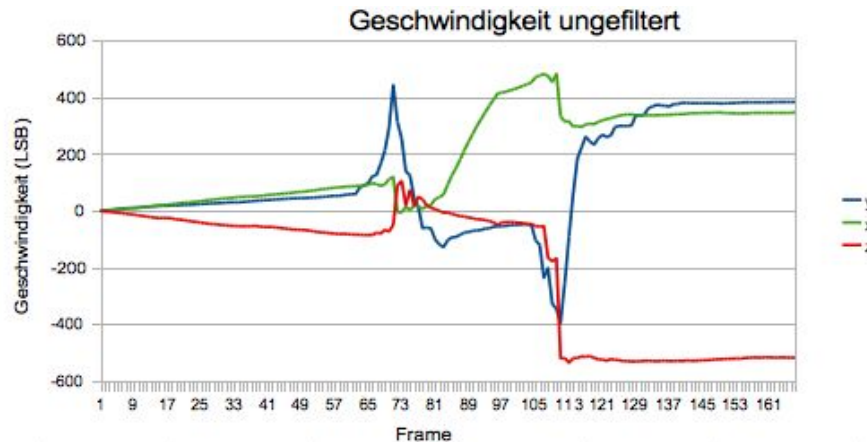


Abb. 5.8: y-move 4: Geschwindigkeit ungefiltert

Nur nach einer Filterung der rohen Beschleunigungsdaten, bei der alle Werte die zwischen -7 LSB und +7 LSB liegen auf 0 LSB gemittelt werden, nimmt der Weg annäherungsweise einen realistischen Verlauf an, vgl. Abb. 5.10.

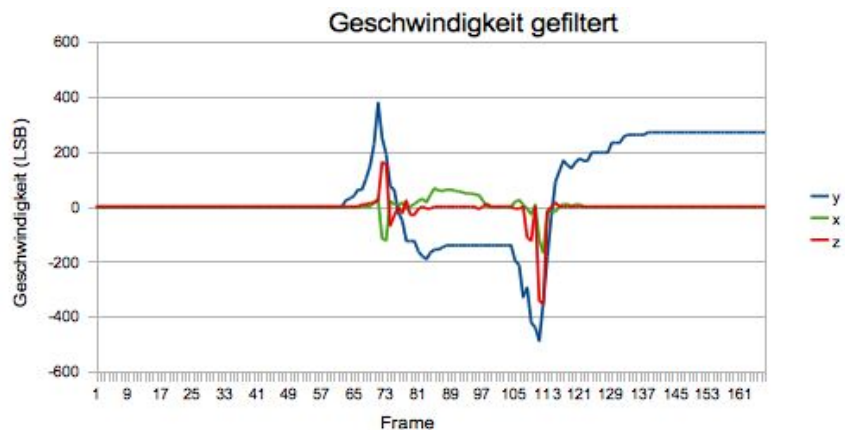


Abb. 5.9: y-move 4: Geschwindigkeit gefiltert

Die Bewegung in x- und z-Achse kommt durch die Natürlichkeit der Bewegung zustande, da diese Achsen in dieser Evaluation nicht komplett ausgeschlossen werden konnten. Dennoch müssten hier die Koordinaten ihren Weg wieder auf Null finden. Stattdessen driftet der y-Wert auf über 7000 ab.

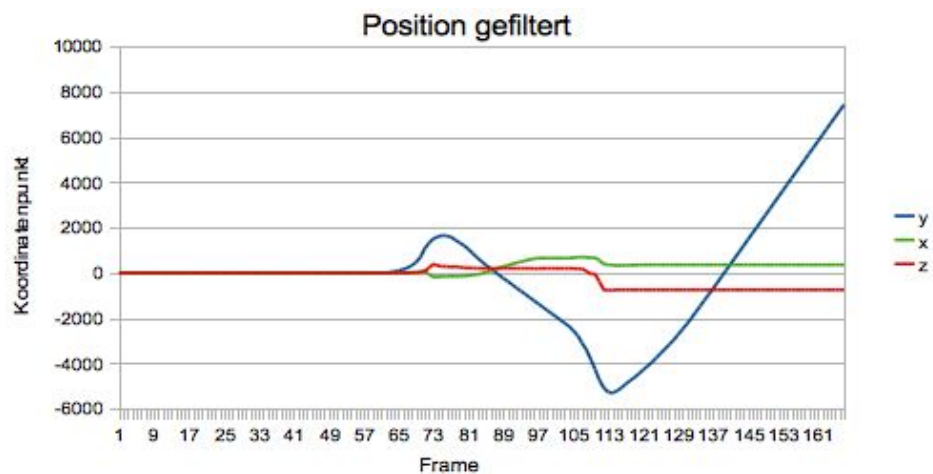


Abb. 5.10: y-move 4: Position gefiltert

Die y-Bewegung zwischen Frame 64 und 89 ist auf eine kurze Rückwärtsbewegung beim Anschlag zurückzuführen. Die eigentliche y-Bewegung findet ab Frame 90 statt und findet ihren Wendepunkt bei Frame 115.

Neigungswinkelanalyse:

Der Sensor wurde initialisiert und nach ca. 1000 Frames¹⁹ in Bewegung gesetzt. Zunächst wird der Sensor über die y-Achse bis zu 90° gedreht, danach wieder zurück auf den Ursprung gebracht. Direkt darauf folgt die gleiche Bewegung schneller. Nach kurzer Ruhephase wird der Sensor in die negative y-Achse 90° hin und zurück gedreht. Damit ist die Rollwinkel-Bewegung abgeschlossen und es folgt eine 90° Drehung über die negative x-Achse vor und zurück ebenso über die positive x-Achse.

Als letztes wird eine 180° Drehung über die z-Achse durchgeführt, mit kurzem Einhalt auf 90°. Nach 180° wird der Sensor wieder zurück in seine Ausgangslage gebracht.

Diese Evaluation ist in Abb. 5.11 Move Gyros Evaluationsergebnis visualisiert.

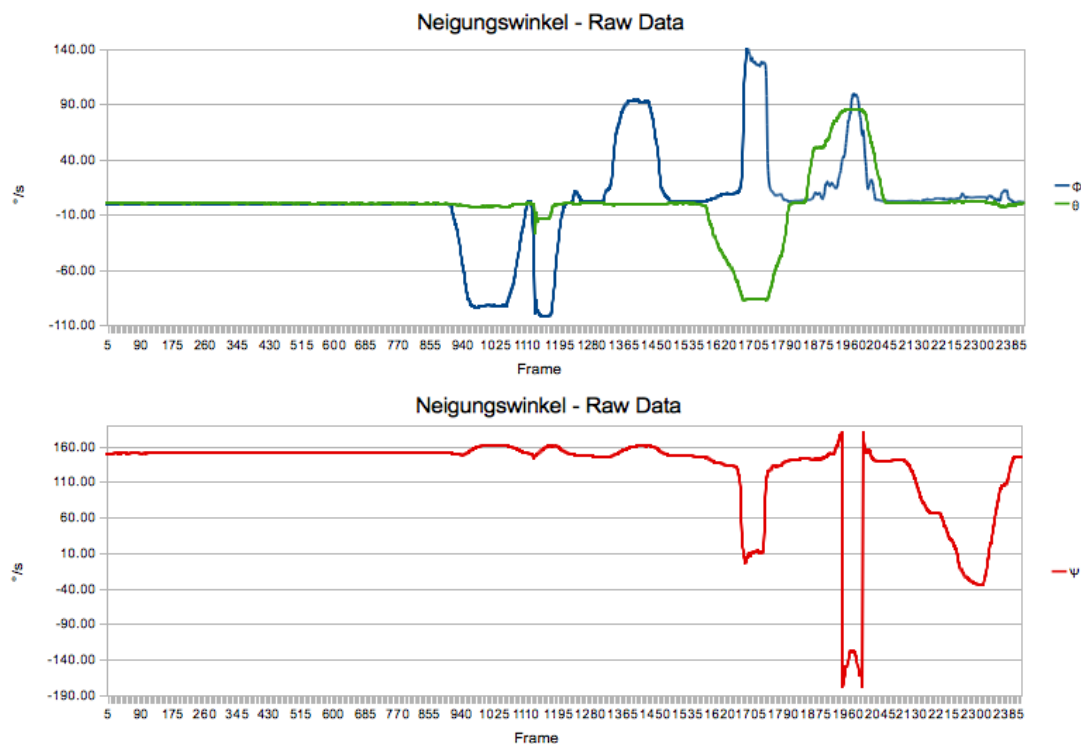


Abb. 5.11: Move Gyros Evaluationsergebnis

Die einzelnen Drehbewegungen sind deutlich von einander zu unterscheiden. Nach der Ruhephase schlägt der Rollwinkel zweimal in die negative, dann einmal in die positive Achsenrichtung aus. Auffällig sind jedoch die Ausschläge danach zu den

¹⁹ In dieser Zeit pendelt sich der Sensor in den Neigungswinkeln noch ein, wie in der Evaluation „Ruhestellung“ festgestellt wurde.

Zeiten der eigentlich reinen Pitchbewegung. Der Pitchwinkel hält sich ansonsten stets am Ursprungswinkel von 0° . Der Yawwinkel schlägt aufgrund der unfixierten Handbewegung ebenfalls bei anderen Achsenrichtungen aus. Entscheidend ist die letzte Drehung auf 180° , wobei auch die Pause nach 90° bei zwischen Frame 2539 und 2579 sehr gut zu erkennen ist.

5.1.3. Zwischenfazit

Nimmt man nun die Ergebnisse der Anforderungsanalyse in den Vergleich mit der Prototypevaluation, so können bereits einige Rückschlüsse erfolgen.

Die Neigungswinkelerfassung arbeitet bereits ordentlich. Dennoch ist sie für sehr diffizile Bewegungen zu ungenau. In der Anforderungsanalyse auf Seite 19 „Ergebnisse der Winkelempfindlichkeiten aus den Fallbeispielen,“ wird eine Genauigkeit von weniger als 0.01° erforderlich. In der Prototypevaluation „Ruhestellung“ ergab sich ein Rauschverhalten, besonders im Yawwinkel von bis zu 0.5° . Wenn man über den feinen Stativschwenk hinweg sieht und die Handkamerabewegungen betrachtet ist die Anforderung weniger hoch und das Rauschen kann durch Filterung der Rohdaten überbrückt werden.

In den Bewegungsanalysen lässt sich ein noch größeres Grundrauschen evaluieren, das sich mit der Zeit in der Positionsermittlung aufaddiert und eine eindeutige Positionsbestimmung nicht ermöglicht.

Dennoch soll im Zuge dieser Arbeit noch eine Drehevaluation stattfinden, um einen Vergleich zum Optischen Matchmovingprozess zu veranschaulichen und die Filteroptionen in der Praxis zu beurteilen.

5.2. Einsatz am Drehort

Im Studio wird eine Panasonic AG-HVX 200E inkl. P2 Capture Card mit dem Sensor bestückt. Der Sensor wird unterhalb der Linse mit einem 62 mm Versatz zum Nodalpunkt mit Hilfe von doppelseitigem Klebeband fest fixiert.



Abb. 5.12: Kamera mit Sensor am Set

Für das Matchmoving in der Postproduktion wird das Set vermessen und Kamera-spezifische Daten notiert.

Kameraspezifikationen:

Panasonic AG-HVX 200E mit P2 Capture Card

Gewicht (ohne Batterie): 2,5kg

Sensor: 1/3-inch 3-CCD

Sensorgroße: 4.6809 x 2.633 mm

Aspect Ratio: 16:9 (1.778)

Leica Dicomar Objektiv F1.6

Brennweite: $f=4.2\text{mm} - 55\text{mm}$ (35mm equivalent: 32.5mm - 423mm)

Codec: DVCPRO HD 1080p/25

Auflösung: 1440x1080px

Datenrate: 118.24 mbits/s

Initialisierung:

Um die Kamera später in der Postproduktion an die exakte Position im Verhältnis zum Raum zu bringen, bedarf es einer Setvermessung zu Beginn des Drehs. Ebenso muss das Positionsverhältnis von Sensor und Objektiv (Nodalpunkt) notiert werden. Die praktische Vermessung wird auf zwei Punkte, die zu Beginn der Aufnahmen im Bild sichtbar sind, von Sensor bis zu je zwei Raumpunkten (x/y) reduziert. Die z-Position ist auf Null fixiert, da sich die Startposition der Kamera auf dem Boden befindet, ansonsten wurde bei der Verwendung des Stativs zusätzlich die Kamerahöhe (Sensorhöhe) gemessen. Voraussetzung für die spätere Genauigkeit der Ursprungskordinate ist eine exakte Abmessung der Bezugspunkte mit einem Maßband bzw. in diesem Fall mit einem Lasermessgerät.

Laut vorgesehener Fallunterscheidung aus Kapitel 3.4. werden zwei Aufnahmen gespeichert. Der Sensor ist über das USB-Kabel mit dem Laptop (Apple Macbook Pro) verbunden. Die Daten werden über den bereits eingeführten Prozess mit Arduino und Processing in einer Textdatei gespeichert.

Shot	Stativschwenk 008KI	Handkamera 027G0
Brennweite	22mm (170mm äquivalent zu 35mm)	4,2mm (32,5mm äquivalent zu 35mm)
Sensorhöhe	0,99m	0m
Entfernung zu Punkt 1	1,71m	2,62m
Entfernung zu Punkt 2	1,88m	2,84m
Sensor Vorlaufzeit	~3 sec	~3 sec

Tabelle 5.1: Setsurvey zur Initialisierung

5.2.1. Drehergebnisanalyse

Die Daten der Kamerabewegung werden in Excel ausgewertet. Die Eliminierung des Grundrauschens wurde hier nach den Erkenntnissen des Selbsttests durch Filterung angewendet.

Zum Vergleich stehen die 3D Kamera Ergebnisse aus dem Matchmoving, das in der Postproduktion mit SynthEyes™ durchgeführt wurde. Die optischen Trackingdaten wurden ebenso in Excel visualisiert und legen den Qualitätsmaßstab vor.

Da die Achsen von Sensor und Matchmoving vertauscht sind müssen diese neu geordnet werden, um einen Vergleich zu ermöglichen.

Matchmoving-Achse	Sensor-Achse
y	z (3. Wert)
z	y (2. Wert)
x	x (1. Wert)
x	θ (2. Wert)
y	Ψ (3. Wert)
z	Φ (1. Wert)

Tabelle 5.2: Achsenzuordnung Matchmoving und Sensorreihenfolge

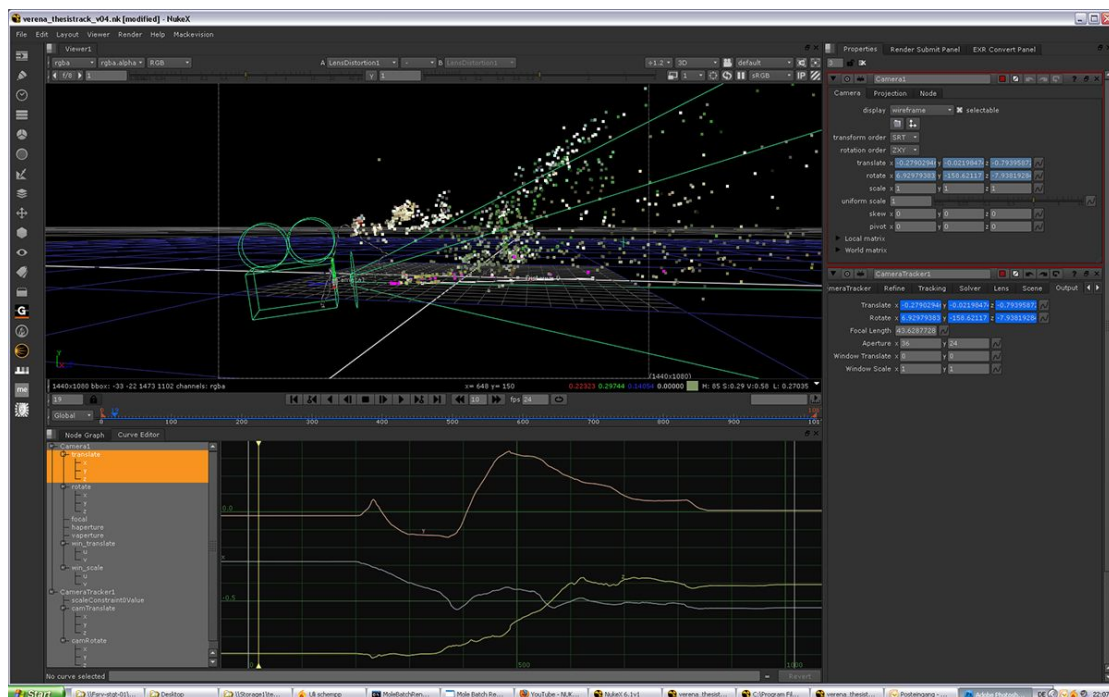


Abb. 5.13: 3D Kamera (027G0) mit Punktwolke in Nuke nach dem Tracking

Fall 1 - Stativschwenk 00KI8:

Die Kamera ist auf einem Stativ befestigt und wird langsam im entgegengesetzten Uhrzeigersinn geschwenkt. Die Translationsachsen bleiben in diesem Fall von der Bewegung unberührt und optimalerweise auf dem Koordinatenpunkt Null.

Die Beschleunigungsdaten im Matchmoving halten sich konstant auf ihrer Ursprungsordinate. Ebenso bleiben die Sensordaten bis auf das Rauschen (± 3 LSB) konstant im Ursprungsbereich, vgl. Abb. 5.15.

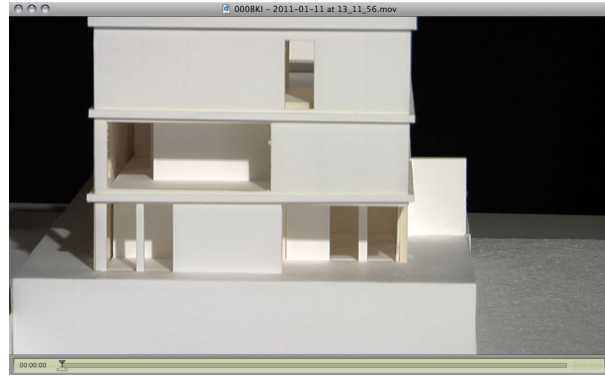


Abb. 5.14.: Startframe der Stativschwenkaufnahme

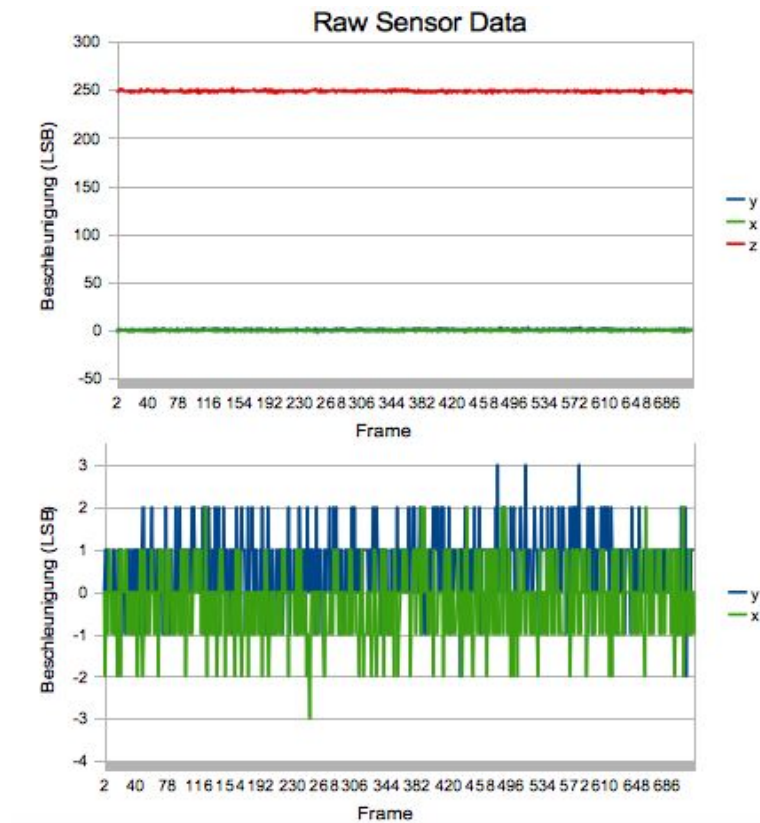


Abb. 5.15.: Stativschwenk: Raw Sesor Data der Translationsachsen

Nach Eliminierung des Rauschens kann die Position auch auf die Nullkoordinaten transferiert werden.

Der graphische Vergleich des Matchmovingtracks und des Sensortracks bezüglich der Neigungswinkel wird erst durch ein Angleichen der Winkel aneinander möglich.

Die Werte aus dem optischen Tracking müssen invertiert werden, um die gleiche Drehrichtung wie die der Sensoren anzunehmen. Darüber hinaus wird hier zunächst jeweils ein Offset geschätzt: Offset Ψ : -68° ; Offset Φ : $+90^\circ$; Offset θ : $+90^\circ$. Ebenso wird der Startframe manuell um die ca. drei Sekündliche Vorlaufzeit des Sensors verschoben.

Die Visualisierung 5.16 zeigt die Matchmoving Kamera in den helleren Farben im Vergleich zu den dunkleren Sensordaten. Im Yawwinkel unterscheidet sich der Sensorwert am Ende um $3^\circ/s$. Der Rollwinkel sollte laut physikalischer Fixierung des Stativs auf dieser Achse kaum eine Änderung aufzeigen, das Ergebnis pendelt zwischen einer Ungenauigkeit von $0.5^\circ/s$. Der Pitchwinkel scheint am weitesten von seinem Qualitätsmaßstab fern zu liegen, wobei es sich hier um Differenzen von weniger als $1^\circ/s$ handelt.

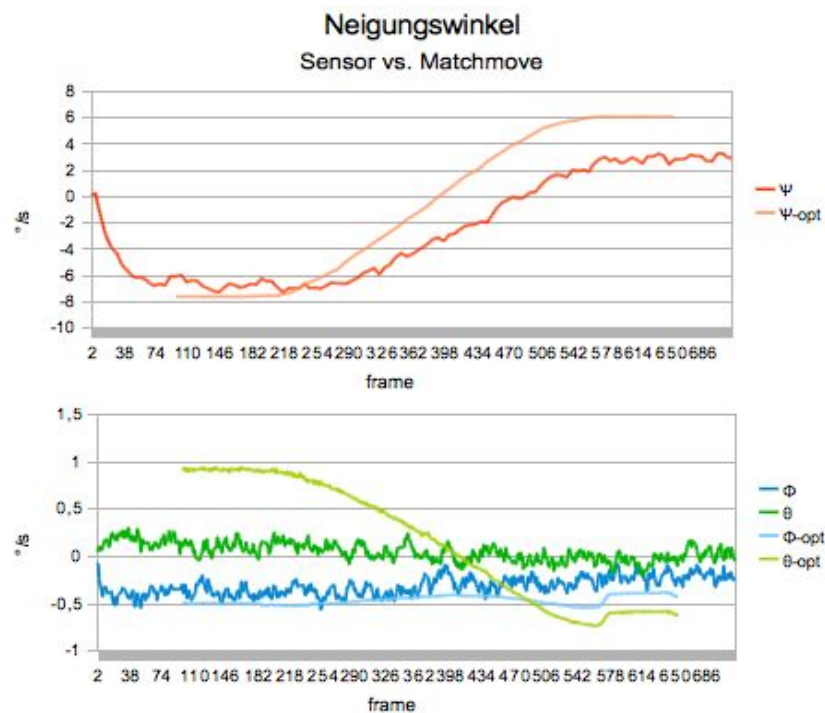


Abb. 5.16.: Stativschwenk Neigungswinkel: Matchmove- vs. Sensordaten

Fall 2 - 027G0 Handkamera:



Abb. 5.17.: Startframe der Handkameraaufnahme mit Trackingpunkten

Die Handkamerabewegung beginnt aus dem Stillstand am Boden und wird ca. 30 Sekunden im Radius von 150 cm bewegt bevor sie wieder am Boden abgestellt wird.



Abb. 5.18: 027G0-Handkamera Matchmoving Positionskoordinaten

Der Bewegungspfad finden im Vergleich zum Qualitätsmaßstab keine Ähnlichkeit; vgl. Abb. 5.18 und Abb. 5.19/5.20. Die Werte driften weit vom Idealwert ab. Ebenso kommen die Neigungswinkel nicht an verwertbare Werte heran, vgl. Abb. 5.21.

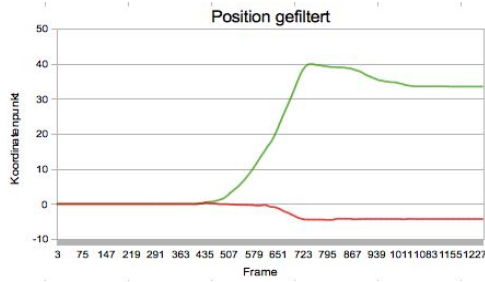


Abb. 5.19: 027G0-Handkamera: Sensor
Positionskoordinaten

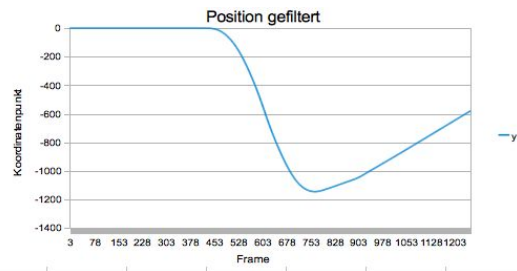


Abb. 5.20: 027G0-Handkamera: Sensor
Positionskoordinaten

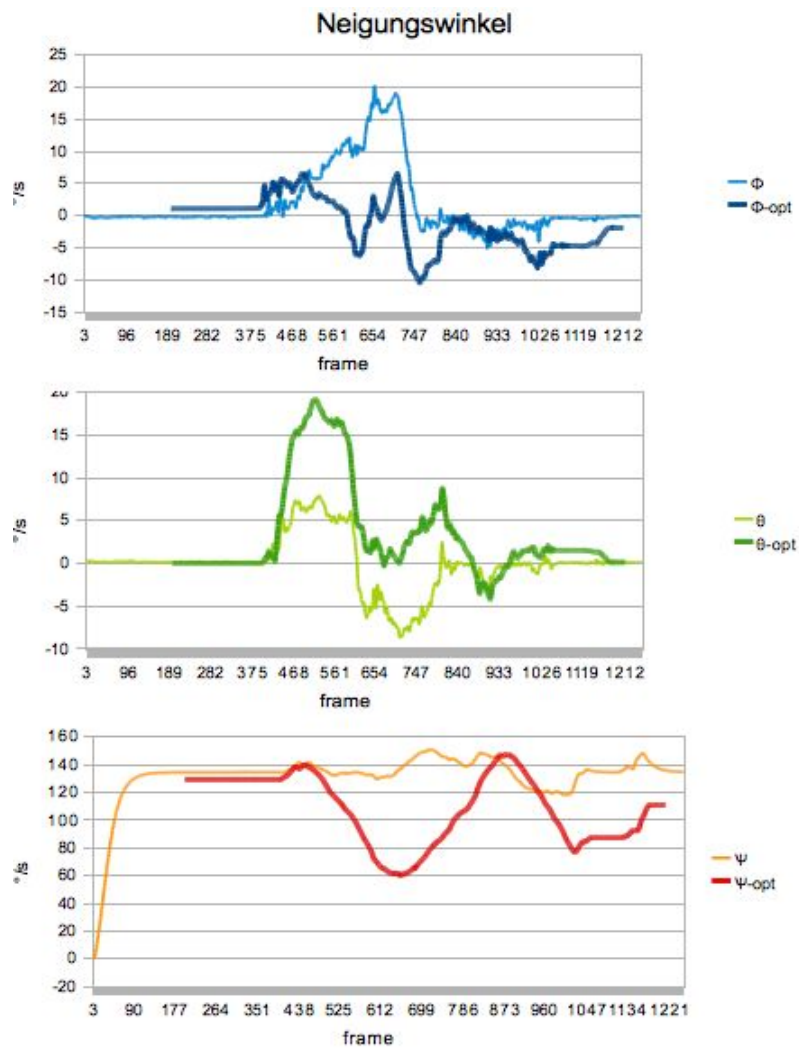


Abb. 5.21: 027G0-Handkamera Vergleich der Neigungswinkel vom Matchmoving und
Sensordaten

6. Fazit

Ziel dieser Arbeit war es herauszufinden, inwieweit sich ein Inertialsystem dazu eignet, die Bewegung einer Kamera am Set zu verfolgen und in ein Koordinatensystem zu speichern. Eine Marktanalyse der Sensoren brachte zunächst Einblick in die verschiedenen Sensorsysteme. Dabei kann zwischen teuren Systemen für Luftfahrt- und Verteidigungstechnik und dem günstigen Consumer Markt unterschieden werden. Für einen wirtschaftlichen Nutzen setzt der Einsatz eines Sensors in der Filmproduktion einen Kompromiss von Kosten, Größe und Güte voraus. Diese Faktoren erfüllen MEMS Sensoren mit Einschränkungen in ihrer Güte. Die Kombination einzelner Sensoren nennt sich IMU (Inertial Measurement Unit) und wird zur Voraussetzung für die Speicherung in sechs Freiheitsgraden.

Die Analyse von vorhandenen Kamerabewegungen aus dem Matchmoving zeigte die Anforderung an die Sensoren auf. Die Mindestauflösung für Beschleunigungssensoren liegt bei ± 1 mg und für Gyroskope bei $0.001^\circ/s$, wenn es um typische Studio- bzw. Handkamerabewegungen geht.

Für eine hohe Empfindlichkeit der Sensoren ist u.a. ein geringes Rauschverhalten notwendig. Es gibt bereits Sensoren auf dem Markt (wie z.B. der ADXL103 von Analog Devices), die ein Rauschverhalten von weniger als $110 \mu g/\sqrt{Hz}$ aufweisen, solange die Bandbreite unter 60 Hz liegt und somit Signale unter 1 mg gemessen werden können. Dies stellt allerdings den Grenzbereich des MEMS-Marktes dar und es gilt diese Empfindlichkeit zukünftig in einem IMU zu kombinieren.

Für eine praktische Analyse wurde in dieser Arbeit auf ein IMU zurückgegriffen, das eine möglichst einfache Programmierung für den Einsatz als Prototyp zulässt, was allerdings auf Kosten der Empfindlichkeit gehen musste und mit 4 mg für die Beschleunigungssensoren und bei $0.035^\circ/s$ für die Gyroskope über den ermittelten Anforderungen liegt. Dennoch galt im nächsten Schritt herauszufinden, ob sich dieser Kompromiss noch auszahlen kann, wenn die Daten bei schwierigen Kamerabewegungen das Matchmoving unterstützen können.

Im Selbsttest „Ruhestellung“ wurde das effektive Rauschverhalten von ± 3 LSB \rightarrow 12 mg ermittelt. Das über den Anforderungen liegende Rauschen macht sich dann

insbesondere in der Positionsermittlung bemerkbar, die durch den Strapdown-Algorithmus berechnet wird. Da sich durch Propagation der Beschleunigung und Geschwindigkeit das Rauschen stets addiert, driften die Positionskoordinaten nach kurzer Zeit linear ab. Durch Filterung der Messdaten kann dieser Bereich ausgeblendet werden, was jedoch wieder auf Kosten der Genauigkeit geht.

Für die Neigungswinkel konnten Ergebnisse mit einem Rauschverhalten von $0.1^\circ/\text{s}$ bis $0.5^\circ/\text{s}$ gemessen werden.

Die Messdaten aus den einfachen Bewegungstests konnten nur die grobe Richtung der Bewegungen wiedergeben und sind im Sinne des Kameratrackings nicht verwendbar.

Nur die Laufzeitlänge für den Kameraeinsatz stellte kein Problem dar, die Werte pendelten auch nach fünf Minuten noch konstant im Ursprung.

Der abschließende Studiotest brachte noch Erkenntnisse im Umgang mit dem Sensor am Set. Sehr einfach war die Befestigung des Sensors an der Kamera. Je näher sich der Sensor am Nodalpunkt der Linse befindet, desto größer sind die Messausschläge für die Gyroskope, als wenn der Sensor am hinteren Teil der Kamera, nahe der Drehachsen befestigt wäre. Der Offset zum Nodalpunkt ist in jedem Fall genau zu notieren, um die Ursprungskoordinate im Verhältnis zum Raum zu berechnen.

Die Ergebnisse vom Dreh driften weit vom Idealwert ab. Die Bewegungen sind unter Verwendung dieses Sensors zu subtil für eine Trajektorie, die noch im Matchmoving hilfreich sein könnte. Dennoch ist im Ausblick S. 55 noch das Potential des Systems beschrieben.

Probleme, die über die Ungenauigkeit hinaus auftauchten:

Zu beliebigen Zeiten lies sich das Sensorboard nicht mehr updaten, d.h. neuer Programmcode konnte nicht mehr auf das Board geladen werden. Dieser Missstand lies sich auch nach intensiver Fehlersuche nicht beheben und reduzierte so die Testläufe auf wenige Programmiersuche.

Während der Prototypentwicklung lies sich die Zeitsynchronisation mit dem Timecode nicht nach angedachtem Konzept umsetzen. Die Programmierung über die Firewireschnittstelle war komplizierter als zunächst angenommen und lies sich dann

aus Mangel an einer Fachkraft nicht realisieren. Diese Option soll für zukünftige Entwicklungen jedoch kein Problem darstellen.

Eine alternative Zeitsynchronisation lies sich über die Uhrzeit finden. Da die Kamera anstelle des Timecodes auch die Uhrzeit in die Metadaten speichert musste die Kamera und der Computer synchronisiert werden und die Messdatei mit einem Zeitstempel pro Frame versehen werden. Der manuelle Abgleich lies dennoch Spielraum von ca. einer Sekunde.

6.1. Ausblick

Das Potential der Sensoren für den Trackingeinsatz am Set ist trotz der geringen Güte des hier verwendeten Prototyps hoch. Der Studiotest hat gezeigt, dass die Verwendung eines Sensors an einer Kamera nicht abwegig ist und im Drehalltag funktioniert. Um die Bewegungsfreiheit einer Kamera nicht einzuschränken kann ein Sensor mit Antenne ausgestattet werden, so dass die Daten per Funk an den Hostrechner (mit einem Empfängermodul und USB-Schnittstelle) übertragen werden.

Im Außeneinsatz der Kamera bzw. Sensors, bei dem womöglich größere Strecken abgefahren werden, kann ein GPS zur Positionsermittlung eine zusätzliche Optimierung bringen. Der Schwerpunkt liegt hier in der Programmierung der Implementierung im Zusammenspiel mit den Sensormessdaten. Das Funktionsprinzip wird bereits bei unbemannten Flugkörpern angewandt.

Die Integration eines Kalman-Filters würde die Propagation der Wegstrecke noch einmal verbessern. Dieses mathematische Verfahren eliminiert die durch die Sensoren entstehenden Messfehler durch Beobachtung vergangener Messfehler.²⁰

Den größten Zeitaufwand am Set verbringt man mit der Vermessung des Sets zur Initialisierung der Position des Kameraursprungs, der zu jeder Aufnahme bekannt sein muss. Wenn die Kamera auf einem Stativ fixiert ist, kann diese Position leicht vermessen werden. Bei einer freien Kamera erzielt man den besten Kompromiss wenn die Kamera zum Aufnahmestart auf dem Boden in einem vordefinierten Muster platziert und ausgerichtet wird. Nachdem die Aufnahme gestartet wurde kann sich die Kamera sofort an den Drehstartpunkt bewegen. Der Weg vom Initialisierungspunkt ab

²⁰ Weiterführende Informationen zum Kalman-Filter und Sensoren finden sich in [INT07].

wird mitgetrackt. Das Initialisierungsmuster ermöglicht die Verwendung von stets gleichen Raumkoordinaten.

Die Ermittlung von Raumkoordinaten kann zudem von einem Laserscan der Umgebung unterstützt werden.

Die beschriebene Initialisierung setzt allerdings den Dreh mit Digitalkameras voraus, da sich die Kosten für Filmmaterial nicht für den Weg zwischen Initialisierungspunkt und dem eigentlichen Drehstart rechtfertigen.

Für eine weiterführende Forschung im Bereich dieser Masterarbeit empfiehlt sich die Evaluation von Sensorpaketen mit höheren Empfindlichkeitsgraden, ohne auf eine kleine IMU-Größe, die für eine einfache Handhabung am Set spricht, zu achten.

Darüber hinaus ist die Verwendung von zwei IMUs interessant, um die Ergebnisse zu mitteln und Fehlmessungen gering zu halten.

Der Industriezweig für die MEMS Produktion boomt insbesondere durch den Consumer Absatz. Der Trend liegt nun vor allem in der Kombination mehrerer Sensoren, den IMUs. Doch anstelle dem Voranbringen einer hohen Empfindlichkeit wird erst einmal auf Reduktion von Gewicht und Größe der IMUs gesetzt.

Für den Einsatz in der Filmproduktion zum Kameratracking wird man bis auf Weiteres auf die Kombination mit einer optischen Infrastruktur angewiesen sein. Die Entwicklungen in den nächsten Jahren sind jedoch zu verfolgen, da Technologieerfolge immer schneller verzeichnet werden.

Meiner Meinung nach wird der Einsatz von Inertial Measurement Units durchaus nach weiterer Forschung und Entwicklung einen Platz zur Optimierung im Matchmoving finden. Da der Aufwand am Set gering ist wird sich die Investition nachhaltig in der Postproduktion lohnen – mit der Einschränkung, dass dies nur für ausgewählte Kameraeinstellungen gilt, die im Matchmoving schwierig sind. Ein kompletter Ersatz des Matchmovings wird wohl auf längere Sicht nicht der Fall sein, zumal auch die Verbesserungen der Matchmovingprogramme voranschreiten werden.

Abbildungsverzeichnis

Abb. 2.1. : Beschleunigungssensoren; Quelle: http://www.sparkfun.com/products/692 bzw. / 8791.....	6
Abb. 2.2: Gyroskop Bohnenberger; Quelle: [INT07].....	9
Abb. 2.3: Freiheitsgrade.....	10
Abb. 2.4.: relative Positionsbestimmung aus Vektoren.....	11
Abb. 2.5: IMU Marktsegmentierung; Quelle: IMU Market 2007-2012 Report Flyer.....	12
Abb. 3.1: Visualisierung der Vorgehensweise.....	14
Abb. 3.2: virtueller Kamerapfad im 3D Raum.....	16
Abb. 3.3: Channel File <.chan>.....	16
Abb. 3.4: Neigungswinkeldelta der Handkamera.....	18
Abb. 3.5: Neigungswinkeldelta des Stativschwenks.....	19
Abb. 3.6: Zusammenhang zwischen Objektgröße, Brennweite der Linse, Objektdistanz und Sensorgröße einer Kamera. Quelle: [RYA93].....	22
Abb. 4.1: IMU von Sparkfun; Quelle: http://dlnmh9ip6v2uc.cloudfront.net/images/products/09623-01b.jpg	27
Abb. 4.2: Prototyp Sparkfun Sensor mit USB-Breakoutboard.....	29
Abb. 4.3: Arduino Codeauszug Registerbelegung.....	30
Abb. 4.4: Main Loop – Schleife mit 25 Hz Abtastfrequenz.....	31
Abb. 4.5: MEMS-Achsenzuordnung; Quelle: http://www.st.com/internet/com/MULTIMEDIA_RESOURCES/IMAGE/WEB_BANNER/ov_prod_l2_motion_sensors_mems.jpg	32
Abb. 4.6: Processing Code zur Speicherung der Sensordaten in einer Textdatei.....	33
Abb. 4.7: Textdatei mit Rohdaten aus dem Sensor.....	35
Abb. 5.1: Ruhstellung 2 – Beschleunigungsdaten (Laufzeit 13 Sekunden).....	37
Abb. 5.2: Ruhstellung 2 – ungefilterte Position.....	38
Abb. 5.3: Ruhstellung 2 – gefilterte Position.....	39
Abb. 5.4: Ruhstellung 3 – Beschleunigung (Laufzeit 5 min).....	39
Abb. 5.5: Ruhstellung 3 – Neigungswinkel (Laufzeit 5 min).....	40
Abb. 5.6: y-move: Raw Sensor Data in LSB.....	40
Abb. 5.7: y-move 4: Position ungefiltert.....	41
Abb. 5.8: y-move 4: Geschwindigkeit ungefiltert.....	41
Abb. 5.9: y-move 4: Geschwindigkeit gefiltert.....	42
Abb. 5.10: y-move 4: Position gefiltert.....	42
Abb. 5.11: Move Gyros Evaluationsergebnis.....	43

Abbildungsverzeichnis

Abb. 5.12: Kamera mit Sensor am Set.....	45
Abb. 5.13: 3D Kamera (027G0) mit Punktwolke in Nuke nach dem Tracking.....	47
Abb. 5.14: Startframe der Stativschwenkaufnahme.....	48
Abb. 5.15: Stativschwenk: Raw Sensor Data der Translationsachsen.....	48
Abb. 5.16: Stativschwenk Neigungswinkel: Matchmove- vs. Sensordaten.....	49
Abb. 5.17: Startframe der Handkameraaufnahme mit Trackingpunkten.....	50
Abb. 5.18: 027G0-Handkamera Matchmoving Positionskoordinaten.....	50
Abb. 5.19: 027G0-Handkamera: Sensor Positionskoordinaten.....	51
Abb. 5.20: 027G0-Handkamera: Sensor Positionskoordinaten.....	51
Abb. 5.21: 027G0-Handkamera Vergleich der Neigungswinkel vom Matchmoving und Sensordaten.....	51

Tabellenverzeichnis

Tabelle 2.1: Beispielhaftes Datenblatt eines Beschleunigungssensors.....	8
Tabelle 2.2: Beispielhaftes Datenblatt eines Drehratensensors.....	9
Tabelle 2.3: Überblick IMUs.....	13
Tabelle 3.1: Channel File Formatierung.....	16
Tabelle 3.2: Beschleunigungsergebnisse aus den Daten der drei Fallbeispiele.....	18
Tabelle 3.3: Ergebnisse der Winkelempfindlichkeiten aus den Fallbeispielen.....	19
Tabelle 3.4: Anforderungen an die Sensoren im Einsatz mit einer Kamera.....	19
Tabelle 4.1: Spezifikationen 9DoF Razor IMU.....	28
Tabelle 5.1: Setsurvey zur Initialisierung.....	46
Tabelle 5.2: Achsenzuordnung Matchmoving und Sensorreihenfolge.....	47

Literaturverzeichnis

- [DES10] **Charles J. Murray**, *DESIGN NEWS Inertial Sensors Go Hollywood*, http://www.xsens.com/images/stories/PDF/2010_sept_designnews.pdf, letzter Zugriff am: September 2010
- [LAN11] **Langenscheidt**, *Langenscheidt Fremdwörterbuch Online*, <http://services.langenscheidt.de/fremdwb/fremdwb.html>, letzter Zugriff am: 19.02.2011
- [DUD74] **Wissenschaftlicher Rat der Dudenredaktion**, *Duden Fremdwörterbuch Band 5*, Bibliographisches Institut Mannheim, 1974
- [BRO02] **Brockhaus in der Wissenmedia**, *Der Brockhaus Naturwissenschaft und Technik*, Brockhaus in der Wissenmedia, 2002
- [INT07] **Jan Wendel**, *Integrierte Navigationssysteme*, Oldenburg Wissenschaftsverlag GmbH, 2007
- [STR04] **Titterton, David H. ; Weston, John L.**, *Strapdown inertial navigation technology*, Stevenage : Institution of Electrical Engineers, 2004
- [OPE11] **OpenPilot**, *AHRS im Einsatz beim OpenPilot Projekt*, 16.02.2011, <http://www.openpilot.org>, letzter Zugriff am:
- [AHR11] **Wikipedia**, *Attitude and Heading Reference Systems*, http://en.wikipedia.org/wiki/Attitude_and_Heading_Reference_Systems, letzter Zugriff am: 16.02.2011
- [IMA11] **iMAR GmbH**, *iMAR iIMU-FCAI Senor*, <http://www.imar-navigation.de/index.php/en/products/imu-of-all-classes-of-performance/iimu-fcai-imu-of-class-1-degh-001-degsqrth-1-mg-5-g--10-g.html>, letzter Zugriff am: 16.02.2011
- [DOB05] **Tim Dobbert**, *Matchmoving: The Invisible Art of Camera Tracking*, Sybex, 2005
- [RYA93] **Dr. Rod Ryan**, *American Cinematographer Manual Seventh Edition*, ASC Press, 1993
- [SPA11] **Sparkfun**, *9DoF Razor IMU*, <http://www.sparkfun.com/products/9623>, letzter Zugriff am: 02.01.2011
- [MUN11] **Bill Premerlani, Jordi Munoz, Jose Julio, Doug Weibel**, *AHRS Code für den 9DoF Razor IMU von Sparkfun*, <http://code.google.com/p/sf9domahrs/>, letzter Zugriff am: 19.02.2011
- [AOW11] **Arduino**, *Arduino offizielle Webrpräsenz*, <http://www.arduino.cc/>, letzter Zugriff am: 19.02.2011
- [ARD11] **Wikipedia**, *Arduino Definition*, <http://de.wikipedia.org/wiki/Arduino-Plattform>, letzter Zugriff am: 19.02.2011

Anhang: IMU Arduino Code

```
// Sparkfveun 9DOF Razor IMU AHRS
// 9 Degree of Measurement Attitude and Heading Reference System
// Firmware v1.0
//
// Released under Creative Commons License
// Code by Doug Weibel and Jose Julio
// Based on ArduIMU v1.5 by Jordi Munoz and William Premerlani, Jose
Julio and Doug Weibel

// Axis definition:
// X axis pointing forward (to the FTDI connector)
// Y axis pointing to the right
// and Z axis pointing down.
// Positive pitch : nose up
// Positive roll : right wing down
// Positive yaw : clockwise

/* Hardware version - v13

ATMega328@3.3V w/ external 8MHz resonator
High Fuse DA
    Low Fuse FF

ADXL345: Accelerometer
HMC5843: Magnetometer
LY530: Yaw Gyro
LPR530:      Pitch and Roll Gyro

Programmer : 3.3v FTDI
Arduino IDE : Select board "Arduino Duemilanove w/ATmega328"
*/
// This code works also on ATmega168 Hardware

#include <Wire.h>

// ADXL345 Sensitivity(from datasheet) => 4mg/LSB    1G => 1000mg/4.9mg =
256 steps
// Tested value : 248
#define GRAVITY 248 //this equivalent to 1G in the raw data coming from
the accelerometer
#define Accel_Scale(x) x*(GRAVITY/9.81)//Scaling the raw data of the
accel to actual acceleration in meters for seconds square
```

Anhang: IMU Arduino Code

```
#define ToRad(x) (x*0.01745329252) // *pi/180
#define ToDeg(x) (x*57.2957795131) // *180/pi

// LPR530 & LY530 Sensitivity (from datasheet) => (3.3mv at 3v)at 3.3v:
3mV/°/s, 3.22mV/ADC step => 0.93
// Tested values : 0.92
#define Gyro_Gain_X 0.92 //X axis Gyro gain
#define Gyro_Gain_Y 0.92 //Y axis Gyro gain
#define Gyro_Gain_Z 0.92 //Z axis Gyro gain
#define Gyro_Scaled_X(x) x*ToRad(Gyro_Gain_X) //Return the scaled ADC raw
data of the gyro in radians for second
#define Gyro_Scaled_Y(x) x*ToRad(Gyro_Gain_Y) //Return the scaled ADC raw
data of the gyro in radians for second
#define Gyro_Scaled_Z(x) x*ToRad(Gyro_Gain_Z) //Return the scaled ADC raw
data of the gyro in radians for second

#define Kp_ROLLPITCH 0.02
#define Ki_ROLLPITCH 0.00002
#define Kp_YAW 1.2
#define Ki_YAW 0.00002

/*For debugging purposes*/
//OUTPUTMODE=1 will print the corrected data,
//OUTPUTMODE=0 will print uncorrected data of the gyros (with drift)
#define OUTPUTMODE 1

#define PRINT_DCM 1 //Will print the whole direction cosine matrix
#define PRINT_ANALOGS 1 //Will print the analog raw data
#define PRINT_EULER 1 //Will print the Euler angles Roll, Pitch and Yaw

#define ADC_WARM_CYCLES 50
#define STATUS_LED 13

int8_t sensors[3] = {1,2,0}; // Map the ADC channels gyro_x, gyro_y,
gyro_z
int SENSOR_SIGN[9] = {-1,1,-1,1,1,1,-1,-1,-1}; //Correct directions
x,y,z - gyros, accels, magnetormeter

float G_Dt=0.02; // Integration time (DCM algorithm) We will run the
integration loop at 50Hz if possible

long timer=0; //general purpuse timer
long timer_old;
long timer24=0; //Second timer used to print values
int AN[6]; //array that store the 3 ADC filtered data (gyros)
```

```
int AN_OFFSET[6]={0,0,0,0,0,0}; //Array that stores the Offset of the
sensors
int ACC[3];           //array that store the accelerometers data

int accel_x;
int accel_y;
int accel_z;
int magnetom_x;
int magnetom_y;
int magnetom_z;
float MAG_Heading;

float Accel_Vector[3]= {0,0,0}; //Store the acceleration in a vector
float Gyro_Vector[3]= {0,0,0}; //Store the gyros turn rate in a vector
float Omega_Vector[3]= {0,0,0}; //Corrected Gyro_Vector data
float Omega_P[3]= {0,0,0}; //Omega Proportional correction
float Omega_I[3]= {0,0,0}; //Omega Integrator
float Omega[3]= {0,0,0};

// Euler angles
float roll;
float pitch;
float yaw;

float errorRollPitch[3]= {0,0,0};
float errorYaw[3]= {0,0,0};

unsigned int counter=0;
byte gyro_sat=0;

float DCM_Matrix[3][3]= {
  {
    1,0,0  }
  ,{
    0,1,0  }
  ,{
    0,0,1  }
};
float Update_Matrix[3][3]={{0,1,2},{3,4,5},{6,7,8}}; //Gyros here

float Temporary_Matrix[3][3]={
  {
    0,0,0  }
  ,{
```

```
    0,0,0 }
  ,{
    0,0,0 }
};

//ADC variables
volatile uint8_t MuxSel=0;
volatile uint8_t analog_reference;
volatile uint16_t analog_buffer[8];
volatile uint8_t analog_count[8];

void setup()
{
  Serial.begin(57600);
  pinMode (STATUS_LED,OUTPUT); // Status LED

  Analog_Reference(DEFAULT);
  Analog_Init();
  I2C_Init();
  Accel_Init();
  Read_Accel();

  Serial.println("Sparkfun 9DOF Razor AHRS");

  digitalWrite(STATUS_LED,LOW);
  delay(1500);

  // Magnetometer initialization
  Compass_Init();

  // Initialize ADC readings and buffers
  Read_adc_raw();
  delay(20);

  for(int i=0;i<32;i++) // We take some readings...
  {
    Read_adc_raw();
    Read_Accel();
    for(int y=0; y<6; y++) // Cumulate values
      AN_OFFSET[y] += AN[y];
    delay(20);
  }

  for(int y=0; y<6; y++)
    AN_OFFSET[y] = AN_OFFSET[y]/32;
```

```
AN_OFFSET[5]-=GRAVITY*SENSOR_SIGN[5];

Serial.print("Offset:");
for(int y=0; y<6; y++)
  Serial.println(AN_OFFSET[y]);
  Serial.println("roll, pitch, yaw; accel_y/x/z; DCM");

delay(2000);
digitalWrite(STATUS_LED,HIGH);

Read_adc_raw();      // ADC initialization
timer=millis();
delay(20);
counter=0;
}

void loop() //Main Loop
{

  if((millis()-timer)>=40) // Main loop runs at 50Hz 1000ms/50fps =20
  {
    counter++;
    timer_old = timer;
    timer=millis();
    if (timer>timer_old)
      G_Dt = (timer-timer_old)/1000.0;    // Real time of loop run. We
use this on the DCM algorithm (gyro integration time)
    else
      G_Dt = 0;

    // *** DCM algorithm
    // Data adquisition
    Read_adc_raw(); // This read gyro data
    Read_Accel();   // Read I2C accelerometer

    if (counter > 5) // Read compass data at 10Hz... (5 loop runs)
    {
      counter=0;
      Read_Compass(); // Read I2C magnetometer
      Compass_Heading(); // Calculate magnetic heading
    }

    // Calculations...
    Matrix_update();
```

```
    Normalize();
    Drift_correction();
    Euler_angles();
    // ***

    printdata();

    //Turn off the LED when you saturate any of the gyros.
    if((abs(Gyro_Vector[0])>=ToRad(300))||
(abs(Gyro_Vector[1])>=ToRad(300))|| (abs(Gyro_Vector[2])>=ToRad(300)))
    {
        if (gyro_sat<50)
            gyro_sat+=10;
        }
    else
        {
            if (gyro_sat>0)
                gyro_sat--;
            }

    if (gyro_sat>0)
        digitalWrite(STATUS_LED,LOW);
    else
        digitalWrite(STATUS_LED,HIGH);

    }

}
```

Anhang: Processing Code

```
import processing.serial.*;
PrintWriter output;

Serial myPort; // The serial port
int h = hour();
int m = minute();
int s = second();
float mil = millis();
int go = 10;

void setup() {
  // List all the available serial ports:
  println(Serial.list());

  // I know that the first port in the serial list on my mac
  // is always my Keyspan adaptor, so I open Serial.list()[0].
  // Open whatever port is the one you're using.
  myPort = new Serial(this, Serial.list()[0], 57600);
  output = createWriter("9doftracking_"+ h+m+s +".txt");
}

void draw() {
  while (myPort.available() > 0) {
    String inBuffer = myPort.readString();
    if (inBuffer != null) {
      // print(h + ":" + m + ":" + s + ":" + (mil/40) + "; ");
      print(inBuffer);
      //output.print(h + ":" + m + ":" + s + "? ");
      output.print(inBuffer);
    }
  }
}

void keyPressed() {
  output.flush(); // Writes the remaining data to the file
  output.close(); // Finishes the file
  exit(); // Stops the program
}
```

Anhang: Sensor Datenblatt LY530ALH (Auszug)

LY530ALH

Mechanical and electrical specifications

2 Mechanical and electrical specifications

2.1 Mechanical characteristics

Table 3. Mechanical characteristics @ Vdd = 3 V, T = 25 °C unless otherwise noted⁽¹⁾

Symbol	Parameter	Test condition	Min.	Typ. ⁽²⁾	Max.	Unit
FSA	Measurement range	4x OUT (amplified)		±300		°/s
FS		OUT (not amplified)		±1200		°/s
SoA	Sensitivity ⁽³⁾	4x OUT (amplified)		3.33		mV/°/s
So		OUT (not amplified)		0.83		mV/°/s
SoDr	Sensitivity change vs temperature	Delta from 25°C		0.05		%/°C
Voff	Zero-rate level ⁽³⁾			1.23		V
Vref	Reference voltage			1.23		V
OffDr	Zero-rate level change Vs temperature	Delta from 25°C		0.05		°/s/°C
NL	Non linearity	Best fit straight line		±1		% FS
BW	Bandwidth ⁽⁴⁾			140		Hz
Rn	Rate noise density			0.035		°/s / √Hz
Top	Operating temperature range		-40		+85	°C

1. The product is factory calibrated at 3 V. The operational power supply range is specified in [Table 4](#).

2. Typical specifications are not guaranteed

3. Sensitivity and zero-rate level are not ratiometric to supply voltage

4. The product is capable of measuring angular rates extending from DC to the selected BW.

Anhang: Sensor Datenblatt ADXL345 (Auszug)

ADXL345

SPECIFICATIONS

T_A = 25°C, V_S = 2.5 V, V_{DD I/O} = 1.8 V, acceleration = 0 g, C_S = 1 μF tantalum, C_{IO} = 0.1 μF, unless otherwise noted.Table 1. Specifications¹

Parameter	Test Conditions	Min	Typ	Max	Unit
SENSOR INPUT					
Measurement Range	Each axis User selectable		±2, ±4, ±8, ±16		g
Nonlinearity	Percentage of full scale		±0.5		%
Inter-Axis Alignment Error			±0.1		Degrees
Cross-Axis Sensitivity ²			±1		%
OUTPUT RESOLUTION					
All g Ranges	Each axis 10-bit resolution		10		Bits
±2 g Range	Full resolution		10		Bits
±4 g Range	Full resolution		11		Bits
±8 g Range	Full resolution		12		Bits
±16 g Range	Full resolution		13		Bits
SENSITIVITY					
Sensitivity at X _{OUT} , Y _{OUT} , Z _{OUT}	±2 g, 10-bit or full resolution	232	256	286	LSB/g
Scale Factor at X _{OUT} , Y _{OUT} , Z _{OUT}	±2 g, 10-bit or full resolution	3.5	3.9	4.3	mg/LSB
Sensitivity at X _{OUT} , Y _{OUT} , Z _{OUT}	±4 g, 10-bit resolution	116	128	143	LSB/g
Scale Factor at X _{OUT} , Y _{OUT} , Z _{OUT}	±4 g, 10-bit resolution	7.0	7.8	8.6	mg/LSB
Sensitivity at X _{OUT} , Y _{OUT} , Z _{OUT}	±8 g, 10-bit resolution	58	64	71	LSB/g
Scale Factor at X _{OUT} , Y _{OUT} , Z _{OUT}	±8 g, 10-bit resolution	14.0	15.6	17.2	mg/LSB
Sensitivity at X _{OUT} , Y _{OUT} , Z _{OUT}	±16 g, 10-bit resolution	29	32	36	LSB/g
Scale Factor at X _{OUT} , Y _{OUT} , Z _{OUT}	±16 g, 10-bit resolution	28.1	31.2	34.3	mg/LSB
Sensitivity Change Due to Temperature			±0.01		%/°C
0 g BIAS LEVEL					
0 g Output for X _{OUT} , Y _{OUT}	Each axis	-150	±40	+150	mg
0 g Output for Z _{OUT}		-250	±80	+250	mg
0 g Offset vs. Temperature for x-, y-Axes			±0.8		mg/°C
0 g Offset vs. Temperature for z-Axis			±4.5		mg/°C
NOISE PERFORMANCE					
Noise (x-, y-Axes)	Data rate = 100 Hz for ±2 g, 10-bit or full resolution		<1.0		LSB rms
Noise (z-Axis)	Data rate = 100 Hz for ±2 g, 10-bit or full resolution		<1.5		LSB rms
OUTPUT DATA RATE AND BANDWIDTH					
Measurement Rate ³	User selectable	6.25		3200	Hz
SELF-TEST⁴					
Output Change In x-Axis	Data rate ≥ 100 Hz, 2.0 V ≤ V _S ≤ 3.6 V	0.20		2.10	g
Output Change In y-Axis		-2.10		-0.20	g
Output Change In z-Axis		0.30		3.40	g
POWER SUPPLY					
Operating Voltage Range (V _S)		2.0	2.5	3.6	V
Interface Voltage Range (V _{DD I/O})	V _S ≤ 2.5 V	1.7	1.8	V _S	V
	V _S ≥ 2.5 V	2.0	2.5	V _S	V
Supply Current	Data rate > 100 Hz		145		μA
	Data rate < 10 Hz		40		μA
Standby Mode Leakage Current			0.1	2	μA
Turn-On Time ⁵	Data rate = 3200 Hz		1.4		ms
TEMPERATURE					
Operating Temperature Range		-40		+85	°C
WEIGHT					
Device Weight			20		mg

¹ All minimum and maximum specifications are guaranteed. Typical specifications are not guaranteed.² Cross-axis sensitivity is defined as coupling between any two axes.³ Bandwidth is half the output data rate.⁴ Self-test change is defined as the output (g) when the SELF_TEST bit = 1 (in the DATA_FORMAT register) minus the output (g) when the SELF_TEST bit = 0 (in the DATA_FORMAT register). Due to device filtering, the output reaches its final value after 4 × τ when enabling or disabling self-test, where τ = 1/(data rate).⁵ Turn-on and wake-up times are determined by the user-defined bandwidth. At a 100 Hz data rate, the turn-on and wake-up times are each approximately 11.1 ms. For other data rates, the turn-on and wake-up times are each approximately τ + 1.1 in milliseconds, where τ = 1/(data rate).

Anhang: Daten-DVD

Die DVD enthält folgende Dateien:

- Anforderungsanalysedateien (Text- und Exceldateien)
- Sensorevaluationsdateien zum Selbsttest (Text- und Exceldateien)
- Drehevaluationsdateien (Excel- und Textdateien)
- Arduino Programmdatei
- Processing Programmdatei
- Drehaufnahmen (Stativschwenk und Handkamera) als Quicktime
- 9 DOF Razor Datenblätter