

Eberhard Karls Universität Tübingen  
Mathematisch-Naturwissenschaftliche Fakultät  
Wilhelm-Schickard-Institut für Informatik

## Studienarbeit Informatik

### **Orientierungsberechnung mittels Multisensordatenfusion auf iOS-Endgeräten**

Sebastian Engel

26.02.2012

**Betreuer**

Jürgen Sommer  
Wilhelm-Schickard-Institut für Informatik  
Universität Tübingen

**Engel, Sebastian:**

*Orientierung von Objekten bei inertialer Navigation*

Studienarbeit Informatik

Eberhard Karls Universität Tübingen

Bearbeitungszeitraum: 10/2011 - 01/2012

## **Zusammenfassung**

Im Rahmen eines Navigations-Programms für Bibliotheken auf mobilen Geräten der neusten Generation beschäftigt sich diese Studienarbeit mit der Orientierung (Winkellage) des Geräts. Zur genauen Navigation in Räumen genügt die Bestimmung der Position nicht. Es ist zusätzlich relevant wie das Gerät orientiert ist. Durch diese Information ist es möglich den Benutzer direkt zu einem bestimmten Ort, im Falle einer Bibliothek ein Buch, zu führen. Im Wesentlichen befasst sich die Studienarbeit mit der Beschaffung und Berechnung der Orientierungsdaten.

## **Danksagung**

Vielen Dank an Jürgen Sommer, Alex Decker, Achim Fritz, Stephan Doerr, Martin Lahl, Philipp Wolter, Markus ... und Sabrina Pfeffer.

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>v</b>
<b>Tabellenverzeichnis</b>	<b>vii</b>
<b>Abkürzungsverzeichnis</b>	<b>ix</b>
<b>1 Einleitung</b>	<b>1</b>
<b>2 Stand der Technik</b>	<b>3</b>
2.1 Beschreibung der Orientierung von Objekten im dreidimensionalen Raum . . . . .	3
2.1.1 Euler-Winkel . . . . .	3
2.1.2 Rotationsmatrizen . . . . .	5
2.1.3 Quaternionen . . . . .	6
2.2 Positionsbestimmung . . . . .	6
<b>3 Eigener Ansatz</b>	<b>9</b>
3.1 Kompassstabilisierung mit Gyroskop . . . . .	9
<b>4 Auswahl geeigneter Hardware</b>	<b>11</b>
4.1 Überblick am Markt befindlicher Geräte hinsichtlich IMU-Ausstattung . . . . .	11
4.2 Wahl iPad 2 . . . . .	11
4.3 Verfügbare APIs . . . . .	12

<b>5</b>	<b>Umsetzung</b>	<b>13</b>
5.1	Unity-Umgebung . . . . .	13
<b>6</b>	<b>Ergebnis</b>	<b>15</b>
<b>7</b>	<b>Ausblick</b>	<b>17</b>
	<b>Literaturverzeichnis</b>	<b>21</b>

# Abbildungsverzeichnis

2.1	Roll-Pitch-Yaw [ <a href="#">App11</a> ] . . . . .	4
2.2	Gimbal Lock [ <a href="#">Wik09</a> ] . . . . .	5





# Tabellenverzeichnis



# Abkürzungsverzeichnis

<b>IMU</b>	Inertial Measuring Unit
<b>API</b>	Application Programming Interface
<b>GPS</b>	Global Positioning System
<b>MEMS</b>	Microelectromechanical Systems
<b>RSSI</b>	Received Signal Strength Indication



# Kapitel 1

## Einleitung

Die Arbeit gliedert sich dazu wie folgt: Zu Beginn wird in Kapitel 2 die Entwicklung der benötigten Hard- und Software betrachtet. In Kapitel 3 besprochen wird die Wahl des eigenen Ansatzes. Bevor wir uns der genauen Implementierung in Kapitel 5 widmen können wählen wir in Kapitel 4 die für unsere Zwecke geeignete Hardware aus. Es folgt eine Auswertung der Ergebnisse in Kapitel 6 mit einer Diskussion. Ein kurzer Ausblick im Kapitel 7 beschließen diese Arbeit.



# Kapitel 2

## Stand der Technik

Bisher findet Navigation hauptsächlich im Freien statt. Zum Beispiel schon seit langem bei Navigationssystemen für Autos. Dabei wird ausschließlich GPS verwendet. Für Navigation innerhalb von Gebäuden ist GPS nicht brauchbar. Es ist zu ungenau und wird durch die Wände des Gebäudes noch zusätzlich ungenauer. Bei der genauen Positionsbestimmung wird es zunehmend wichtiger auch die Orientierung zu bestimmen. Denn innerhalb von Gebäuden und bei Positionsunterschieden von wenigen Metern ist die Information in welche Richtung man schaut ebenfalls interessant und liefert zusätzliche Information. Gerade bei einer Navigations-App für Bibliotheken ist es wichtig zu wissen welches Regal gerade angeschaut wird. Außerdem darf die Position eine Ungenauigkeit von einigen Zentimetern nicht überschreiten, denn sonst ist die Führung zum Buch so ungenau, dass die App mehr Umstand als Nutzen bringt.

### 2.1 Beschreibung der Orientierung von Objekten im dreidimensionalen Raum

Zur Beschreibung der Orientierung von Objekten im dreidimensionalen Raum in kartesischen Koordinatensystemen gibt es mehrere Möglichkeiten. Die drei am häufigsten verwendeten und für uns relevanten werden im Folgenden vorgestellt.

#### 2.1.1 Euler-Winkel

Bei Euler-Winkeln handelt es sich um drei Winkel die jeweils die Rotation um eine bestimmte Achse des Koordinatensystems angeben. So kann eine Transformation zwischen zwei Koordinatensystemen, dem Labor- und dem Körperfesten-System definiert werden.

Es existieren mehrere Definitionen von Euler-Winkeln, was die Reihenfolge der Drehungen um die Achsen anbelangt. Für unsere Zwecke beschäftigen wir uns mit Yaw-Pitch-Roll - zu deutsch: Roll-Nick-Gier-Winkel. Dies entspricht auch der Luftfahrtnorm (DIN 9300).

- Roll (Roll-Winkel) beschreibt die Querneigung, also die Drehung um die X-Achse.
- Pitch (Nick-Winkel) beschreibt die Längsneigung, also die Drehung um die Y-Achse.
- Yaw (Gier-Winkel) beschreibt Orientierung, also die Drehung um die Z-Achse.

Bei mobilen Geräten wie dem Apple iPhone gibt es anders als bei Fahrzeugen keine fest definierte Ausrichtung. Beim iPhone und iPad sind die Winkel darum so verteilt wie auf Bild 2.1 zu sehen.

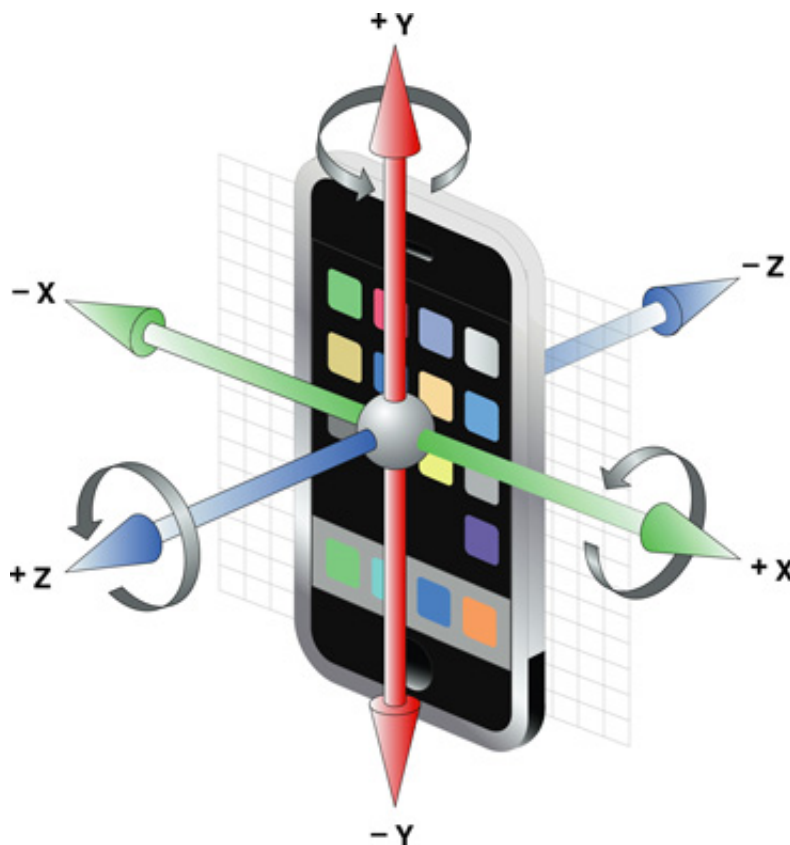


Abbildung 2.1: Roll-Pitch-Yaw [App11]

Euler-Winkel haben den Vorteil, dass sie jeder intuitiv verstehen kann. Jeder lernt Euler-Winkel in der Schule kennen. Somit kann man mit ihnen auch einfach rechnen.



## 2.1. BESCHREIBUNG DER ORIENTIERUNG VON OBJEKTEN IM DREIDIMENSIONALEN

Eine Drehung mit Euler-Winkeln setzt sich immer aus einer Kombinationen von Rotation der drei Achsen zusammen. Das heißt eine Drehung findet nie direkt statt, sondern über mehrere nacheinander ausgeführte Rotationen der einzelnen Achsen. Das kann bei manchen Anwendungen ein Problem sein. Zum Beispiel wenn die Orientierung schneller abgefragt wird als die Teilschritte einer Drehung berechnet werden, kann es vorkommen, dass in einzelnen Key-Frames der Anwendung falsche Orientierungsdaten einfließen.

### Gimbal Lock

Der große Nachteil von Euler-Winkeln ist der Gimbal Lock (engl. f. Blockade der Kardanischen Aufhängung). So nennt man es wenn zwei Achsen die selbe Drehung bestimmen. Dadurch fehlt ein Freiheitsgrad. Man kann eine bestimmte Drehung erst dann wieder durchführen wenn man eine der beiden zusammengefallenen Achsen zurück dreht. In Abbildung 2.2 ist leicht zu erkennen, dass die innere (blau) und äußere (grün) Achse die selbe Drehung bestimmen. Es ist daher momentan nicht möglich das Flugzeug nach vorne oder hinten zu Kippen. Erst müsste das Flugzeug entlang der mittleren Achse um  $90^\circ$  gedreht werden. [Wik12b] [Koc08]

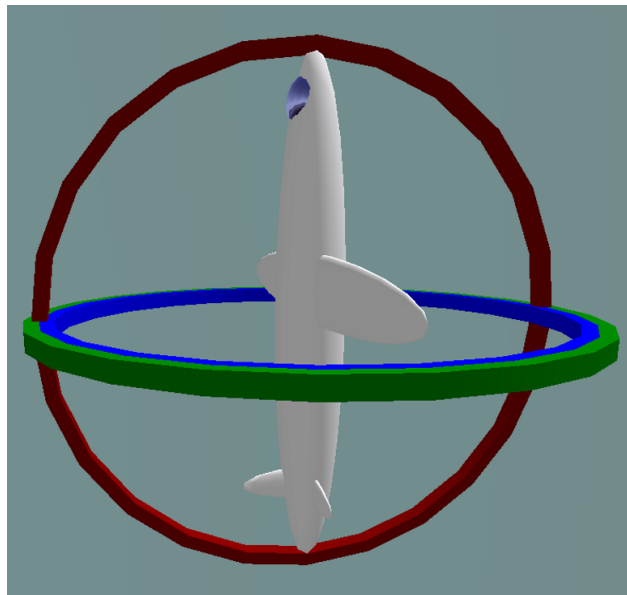


Abbildung 2.2: Gimbal Lock [Wik09]

### 2.1.2 Rotationsmatrizen

Eine Rotationsmatrix ist eine orthogonale Matrix, die ebenfalls die Drehung im Raum beschreibt. Sie ist als eine Hintereinanderausführung einer oder mehrerer

Rotationen um beliebige Drehachsen im dreidimensionalen Raum definiert. Darum ist klar, dass Rotationsmatrizen quasi nur eine Zusammenfassung von einzelnen Rotationen mit Euler-Winkeln in ein einziges Konstrukt sind. Gimbal Lock kann auch mit Rotationsmatrizen auftreten. [Wik12a]

### 2.1.3 Quaternionen

Quaternionen sind ein mathematisches Konstrukt um Orientierung von Objekten im dreidimensionalen Raum zu beschreiben. Sie setzen sich aus einem skalaren und einem vektoriellen Teil zusammen. Der vektorielle Teil ist allein dazu da um die Achse der durchzuführenden Drehung zu beschreiben. Der Skalaranteil gibt den Winkel der Drehung an. Es wird also für jede Rotation eine eigene Achse konstruiert entlang der gedreht wird. Dadurch gibt es keine Zwischenschritte, sondern nur eine einzige Rotation. So kann auch das Problem des Gimbal Locks gar nicht erst entstehen.

## 2.2 Positionsbestimmung

Die Positionsbestimmung erfolgt in unserem Fall über Bluetooth. Dazu werden in dem Raum in dem man Navigieren möchte Bluetooth-Sender (Beacons) ausgelegt. Diese sollten möglichst gleichmäßig verteilt sein, damit eine gleichmäßige Bluetooth-Abdeckung gewährleistet ist. Die App weiß wo sich die einzelnen Beacons im Raum befinden und empfängt die RSSI-Werte der ausgelegten Beacons und kann so die Position des Geräts bestimmen.

Dabei können mehrere Probleme auftreten. Das größte ist, die Senderate der Beacons. Herkömmliche Bluetooth-Sticks sind dafür gemacht eine ständige Verbindung zu einem Gerät aufrechtzuerhalten um Daten zu übertragen. Bei unserem Anwendungsfall wollen wir keine Daten übertragen, sondern nur möglichst oft den RSSI-Wert der einzelnen Beacons erfahren. Normale Bluetooth-Sticks schaffen meistens nur eine Rate von ca. drei Sekunden. Das ist zu wenig um mit wenigen Sticks eine zuverlässige Navigation zu realisieren. Man muss entweder in relativ teure (über 100) Bluetooth-Sender investieren die schnell sind, oder viele von den langsameren auslegen.

Ein weiteres Problem ist, dass Bluetooth leicht gestört werden kann. Personen, Wände, Bücherregale sind ein Problem bei Bluetooth. Darum kann man nicht sicher sein, dass die RSSI-Werte die beim Gerät ankommen die entsprechende Entfernung repräsentieren.

Um diesen beiden Hauptproblemen entgegenzuwirken wird ein Partikelfilter eingesetzt. Mit dem partikelfilter wird eine Wolke gewichteter Partikel erzeugt die den aktuellen Aufenthaltsort schätzen. Anhand der aktuellsten Position die aus den Bluetooth-RSSI-Werten berechnet wurde werden die einzelnen Partikel

gewichtet. So kann die Ungenauigkeit der Bluetooth-Werte etwas korrigiert werden. [\[Wik12c\]](#)



# Kapitel 3

## Eigener Ansatz

Die Orientierungs-Angabe, die man aus den Gyroskop-Daten gewinnt ist relativ zur Position des Geräts bei Beginn der Messung. Ohne zusätzliche absolute Angaben kann man nicht die absolute Orientierung im Raum bestimmen. Darum ist es von Nutzen zusätzlich den Kompass (und das Accelerometer) zu nutzen.

### 3.1 Kompassstabilisierung mit Gyroskop

Eine funktionierende Orientierung lässt sich bereits mit Kompass und dem Pitch-Wert den das Gyroskop liefert realisieren.

Eine Orientierungs-Angabe als Euler-Winkel würde beispielsweise wie folgt aussehen:

$$\begin{pmatrix} 0.016134 \\ -0.000284 \\ 1.618407 \end{pmatrix}$$

Die Werte sind in Radiant angeben. Negative Werte können zustande kommen, da die Skala von  $-\pi$  bis  $+\pi$  geht.



# Kapitel 4

## Auswahl geeigneter Hardware

Bei der Suche nach einem passenden Gerät kamen mehrere Kriterien zum Tragen. Es sollte der Visualisierung größer als ein Smartphone sein aber trotzdem portabler als ein herkömmliches Notebook. Es steht also fest, dass ein Tablet am besten geeignet ist für diese Art Anwendung.

### 4.1 Überblick am Markt befindlicher Geräte hinsichtlich IMU-Ausstattung

Ernsthaft am Markt vertreten waren zum Zeitpunkt der Hardware-Entscheidung (Anfang 2011) nur das Apple iPad 1 und das Motorola Xoom. Das iPad war mit Accelerometer und Kompass ausgestattet, jedoch nicht mit einem Gyroskop. Das Motorola Xoom hatte alle drei IMUs verbaut. Allerdings war das Android-Betriebssystem anfangs Berichten zufolge instabil. Diese Android Version war die erste, die für Tablets optimiert war.

### 4.2 Wahl iPad 2

Das Apple iPad 2 war das erste ernstzunehmende Tablet das alle drei, für unsere Zwecke, wichtigen IMUs mitbrachte. Zum Erscheinungszeitpunkt war Apple auch der Hersteller mit der meisten Erfahrung. Das iPad der ersten Generation war bereits ein Jahr auf dem Markt und hatte mit iOS ein ausgereiftes Betriebssystem. iOS wird auf dem Apple iPhone schon seit 2007 verwendet. Man konnte also sicher gehen, dass ein ausgereiftes Betriebssystem vorhanden ist.

Abgesehen von der Ausstattung hatte das Apple iPad Anfang 2011 den größten Marktanteil mit sehr großem Abstand zu allen anderen Geräten. Auch das Betriebssystem iOS war mit großem Abstand marktführend.

### 4.3 Verfügbare APIs

Bei iOS sind für unsere Zwecke vor allem zwei Frameworks wichtig. CoreLocation im den Kompass und das Accelerometer auszulesen und CoreMotion um das Gyroskop oder auch das Accelerometer auszulesen.

CoreMotion liefert einerseits rohe Daten wie zum Beispiel die Drehraten des Gyroskops, andererseits aber auch bereits bereinigte Daten. Zum Beispiel lassen sich Beschleunigung und Gravitation getrennt auslesen.



# Kapitel 5

## Umsetzung

### 5.1 Unity-Umgebung



# Kapitel 6

## Ergebnis



# Kapitel 7

## Ausblick

Mit das Wichtigste natürlich!

Hier gilt es beides, die Info-Seite der Arbeit sowie die Bio-Seite zu diskutieren!!

Take your time for writing the discussion, it is the most important chapter of your thesis.

Mindestens 5 Seiten lang.

Ausblick kann auch ein extra Kapitel werden, wenn man das will.





# Literaturverzeichnis

- [App11] Apple. Datei:acceleration\_axes.jpg, 2011.  
<http://developer.apple.com/library/ios/#documentation/EventHandling/Conceptual/EventHandlingiPhoneOS/MotionEvents/MotionEvents.html> [Online; Stand 4. Januar 2012].
- [Koc08] Thomas Koch. Rotationen mit quaternionen in der computergrafik, 2008. Diplomarbeit, Fachhochschule Gelsenkirchen.
- [Wik09] Wikipedia. File:gimbal lock.png, 2009.  
[http://en.wikipedia.org/wiki/File:Gimbal\\_lock.png](http://en.wikipedia.org/wiki/File:Gimbal_lock.png) [Online; Stand 24. Februar 2012] This file is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported license.
- [Wik12a] Wikipedia. Drehmatrix — wikipedia, die freie enzyklopädie, 2012.  
<http://de.wikipedia.org/w/index.php?title=Drehmatrix> [Online; Stand 24. Februar 2012].
- [Wik12b] Wikipedia. Gimbal lock — wikipedia, the free encyclopedia, 2012.  
[http://en.wikipedia.org/w/index.php?title=Gimbal\\_lock](http://en.wikipedia.org/w/index.php?title=Gimbal_lock) [Online; Stand 24 Februar 2012].
- [Wik12c] Wikipedia. Particle filter — wikipedia, the free encyclopedia, 2012.  
[http://en.wikipedia.org/w/index.php?title=Particle\\_filter](http://en.wikipedia.org/w/index.php?title=Particle_filter) [Online; Stand 24 Februar 2012].

