



Institut für Informatik  
der Technischen Universität München



# **Zeitkalibrierung in Augmented Reality Anwendungen**

Michael Schlegel





# Zeitkalibrierung in Augmented Reality Anwendungen

Michael Schlegel

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. G. Carle

Prüfer der Dissertation: 1. Univ.-Prof. G. J. Klinker, Ph. D.  
2. Univ.-Prof. Dr. G. Reitmayr,  
Technische Universität Graz / Österreich

Die Dissertation wurde am 03.06.2011 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 06.10.2011 angenommen.



Für Mama



# Zusammenfassung

Diese Arbeit untersucht die Kalibrierung des zeitlichen Offsets zweier Sensoren im Gebiet der Augmented Reality, um die Genauigkeit von Sensorfusionsalgorithmen zu erhöhen. Dass zwei Sensoren zeitlich zu einander kalibriert werden müssen, ist in der Forschergemeinschaft und in der Industrie bekannt. Es existieren jedoch bis heute nur spezielle und auf einzelne Fälle zugeschnittene Lösungen.

Diese Arbeit diskutiert verschiedene Methoden der zeitlichen Kalibrierung und berücksichtigt dabei die speziellen Anforderungen, die von Augmented Reality Anwendungen gestellt werden. Dabei sind diese Methoden generisch gehalten, dass eine Anwendung mit jeglichen Sensoren die Positionen oder Orientierungen liefern möglich ist. Zudem können die hier vorgestellten Methoden auch zur Laufzeit angewendet werden, womit eine Anpassung an sich ändernde zeitliche Offsets möglich ist.

Um zu einer zeitlichen Kalibrierung zu gelangen, ist eine Reihe von Schritten notwendig. Zuerst ist zu prüfen, welche Qualität die vorliegenden Sensordaten haben und welche Daten für eine Kalibrierung verwendet werden können. Nach der Auswahl der Daten müssen die mehrdimensionalen Daten auf einen eindimensionalen Strom reduziert werden. Dies kann mit unterschiedlichen Methoden geschehen. Dabei wird gezeigt, welche davon sich für eine zeitliche Kalibrierung eignen. Für die eigentliche Kalibrierung werden verschiedene Verfahren zur Bestimmung des zeitlichen Offsets vorgestellt und miteinander verglichen. Dabei wird die normalisierte Kreuzkorrelation eingehender betrachtet. Anschließend werden Unsicherheiten der Kalibrierungsverfahren untersucht und Methoden aufgezeigt, die es ermöglichen die Qualität einer Zeitkalibrierung zu beurteilen.

Um die gesamte Prozesskette der zeitlichen Kalibrierung genauer untersuchen zu können, wurde eine Monte-Carlo-Simulationsumgebung entwickelt. Damit ist es möglich, bestimmte Störeinflußgrößen auf die zeitliche Kalibrierung getrennt von anderen Störeinflüssen zu untersuchen. Und somit auch einzelne Verfahren gezielt und isoliert zu untersuchen. Gegen Ende wurden die vorgeschlagenen Methoden in mehreren Szenarien angewendet, um die Anwendbarkeit der zeitlichen Kalibrierung zu zeigen. Dabei konnte gezeigt werden, dass die Methode der zeitlichen Kalibrierung auch in der Praxis erfolgreich angewendet werden kann und zu einer signifikanter Verbesserung in den resultierenden Genauigkeiten führt.



## Abstract

This work studies the calibration of the relative latency between two sensors in the field of Augmented Reality. Taking the relative latency into account increases the accuracy of sensor fusion algorithms. The need of temporal calibration is well known to the research community as well to the developers of commercial applications. But the existing solutions are designed for dedicated applications only and are not capable of a calibration during the runtime of the application. Most works on this topic have already carried out temporal calibrations in a dedicated calibration step. As the relative latency may change over time this static approach can lead to imprecision.

This work discusses several methods of calibrating the temporal offset of different sensors. The presented methods are designed for an online usage, which makes it possible to detect changing latencies.

For a temporal calibration several steps are necessary. First the quality of the data has to be examined and that data, which is not fulfilling the criteria, has to be discarded. When the selection the dimension of the data has to be reduced, which can be done in several ways. A subset of these methods is described in detail in this thesis. After these preliminary steps are done the actual actual calibration of the relative latency can be performed. This is done by comparing the two data sets while shifting one data set back and forward in time. The offset which gains a maximum of similarity, is then chosen as the value of the relative latency. For the purpose of comparison the normalized cross correlation is proposed.

For further research on this type of temporal calibration a Monte Carlo simulation environment was implemented. This environment allows to study different methods and various types of influences separated from each other. For a correct and meaningful simulation various sources of error are described and also how to determine the parameters for the simulation. To show the feasibility of the proposed methods applications are described within this thesis.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Augmented Reality . . . . .	1
1.2	Problembeschreibung . . . . .	4
1.3	Lösungsvorschlag . . . . .	6
<b>2</b>	<b>Hintergrund</b>	<b>9</b>
2.1	Tracking . . . . .	9
2.2	Graphen über Räumliche Abhängigkeiten (SRG) . . . . .	10
2.3	Darstellung von Trackingdaten . . . . .	11
2.3.1	Pose Darstellung . . . . .	12
2.3.2	Klassifikation von Tracking Fehlern . . . . .	13
2.3.3	Tracking Daten als stochastische Signale . . . . .	15
2.4	Tracking Sensoren . . . . .	17
2.5	Sensorfusion . . . . .	21
2.5.1	Taxonomie Sensorfusion . . . . .	21
2.5.2	Mathematische Modelle der Sensorfusion . . . . .	24
2.6	Tracking Frameworks . . . . .	26
2.6.1	OpenTracker/Studierstube . . . . .	26
2.6.2	CampAR . . . . .	26
2.6.3	DWARF . . . . .	27
2.6.4	ubitrack . . . . .	27
2.6.5	LocationStack . . . . .	28
2.7	Latenzen von Trackingsystem . . . . .	29
2.8	Zusammenfassung . . . . .	29
<b>3</b>	<b>Prinzipien der Zeitkalibrierung</b>	<b>31</b>
3.1	Verwandte Arbeiten . . . . .	31
3.2	Vorschlag für eine zeitliche Kalibrierung . . . . .	38
3.3	Voraussetzungen . . . . .	39
3.4	Dimensionsreduktion . . . . .	40
3.5	Synchronisierung und Segmentierung . . . . .	42
3.6	Time Delay Estimation . . . . .	43
3.7	Korrektur . . . . .	45
3.8	Zusammenfassung . . . . .	47

<b>4</b>	<b>Eigenschaften von Signalen</b>	<b>49</b>
4.1	Räumliche Registrierung . . . . .	49
4.2	Update-Raten . . . . .	51
4.3	Ausreißer . . . . .	55
4.4	Pathologische Fälle . . . . .	59
4.5	Rauschen . . . . .	62
4.6	Zusammenfassung . . . . .	67
<b>5</b>	<b>Dimensionsreduktion</b>	<b>69</b>
5.1	Einführung . . . . .	69
5.2	Statische Verfahren . . . . .	70
5.2.1	Positionen . . . . .	70
5.2.2	Ableitungen von Positionen . . . . .	71
5.2.3	Orientierung . . . . .	72
5.2.4	Ableitungen von Rotationen . . . . .	73
5.3	Adaptive Verfahren . . . . .	73
5.3.1	Kanonische Korrelation . . . . .	74
5.3.2	Two-Choice Algorithmen . . . . .	76
5.3.3	Hauptkomponentenanalyse . . . . .	77
5.4	Zusammenfassung . . . . .	82
<b>6</b>	<b>Synchronisierung und Segmentierung</b>	<b>83</b>
6.1	Interpolation . . . . .	83
6.1.1	Lineare Interpolation von Positionen . . . . .	84
6.1.2	Lineare Interpolation von Rotationen . . . . .	84
6.2	Synchronisierung . . . . .	84
6.2.1	Generelle Methode . . . . .	85
6.2.2	Wiederholte Synchronisierung . . . . .	85
6.3	Segmentierung . . . . .	88
6.4	Zusammenfassung . . . . .	90
<b>7</b>	<b>Zeitkalibrierung</b>	<b>91</b>
7.1	Einleitung . . . . .	91
7.2	Timeshift-Verfahren . . . . .	91
7.2.1	Grobe Bestimmung . . . . .	92
7.2.2	Feine Bestimmung . . . . .	93
7.3	Ähnlichkeitsmaße . . . . .	94
7.3.1	Korrelation . . . . .	94
	Berechnung . . . . .	95
7.3.2	Geometrisch . . . . .	96
7.4	Unsicherheiten der TDE . . . . .	98
7.4.1	Cramér-Rao Schranke . . . . .	99
7.4.2	Mögliche Maße für die Sicherheit der Schätzung . . . . .	100
7.5	Aggregation . . . . .	102

7.6	Zusammenfassung . . . . .	104
<b>8</b>	<b>Monte-Carlo-Simulation für Zeitkalibrierung</b>	<b>105</b>
8.1	Einführung und Related Work . . . . .	105
8.2	Monte-Carlo-Modell . . . . .	106
8.3	Framework . . . . .	110
8.3.1	Ablauf der Simulation . . . . .	111
8.3.2	Bestimmung der Verteilungen . . . . .	111
	Rauschen . . . . .	112
	Update Raten und Jitter . . . . .	114
	Bewegungen . . . . .	114
	Registrierungsfehler . . . . .	116
8.3.3	Beispiel . . . . .	117
8.4	Zusammenfassung . . . . .	119
<b>9</b>	<b>Anwendungen</b>	<b>121</b>
9.1	Ringschluss . . . . .	121
9.2	Optisches Odometer . . . . .	122
9.3	Genauigkeitsevaluation . . . . .	123
<b>10</b>	<b>Zusammenfassung</b>	<b>129</b>
10.1	Diskussion . . . . .	129
10.2	Weiteres Vorgehen . . . . .	130
	<b>Literaturverzeichnis</b>	<b>142</b>



# 1 Einleitung

Die Kalibrierung der relativen Latenz zwischen Sensoren ist Grundlage für eine präzise und zuverlässige Sensorfusion. Dieses Kapitel soll dem Leser eine kurze Einführung in die Augmented Reality bieten und das Problem der relativen Latenzen greifbar machen.

## 1.1 Augmented Reality

Augmented Reality ist ein Mensch-Maschine-Kommunikationsmittel, welches die Realität, die uns umgibt, mit virtuellen Inhalten anreichert. Mit Hilfe der Augmented Reality ist es möglich dem Benutzer eines solchen Systems Informationen in einer ihm vertrauten visuellen Art und Weise zu präsentieren. Die Informationen werden Teil der visuellen oder auch haptischen Wahrnehmung des Benutzers und können somit leichter erfasst und interpretiert werden. Man kann auch von virtuellen Annotationen der Realität sprechen. Diese Annotationen müssen dazu ortsbezogen sein und sich somit so verhalten wie reale Objekte in der Umgebung.

Die Anwendungen von Augmented Reality sind zahlreich und sind mittlerweile in sehr vielen Bereichen des Lebens angekommen. Der Fußballinteressierte erfreut sich an dem augmentierten Freistoßkreis [2] oder die Entwickler von Fahrzeugsensoren an einer Umgebung, die es ihm erlaubt die Messwerte visuell zu überprüfen. In Abbildung 1.1 ist ein solches System zu sehen, welches dem Entwickler erlaubt, die Messdaten der Sensoren in der Realität und in Echtzeit zu überprüfen [103].

Für die Darstellung der erweiterten Realität werden Displaytechnologien benötigt. Dies können handelsübliche Computerdisplays sein, wie bei der eben gezeigten Anwendung, oder auch komplizierte Spezialhardware wie *Head-Mounted-Displays*. Dies sind Brillen, die mit Hilfe von in einer Brille integrierten Displays dem Benutzer die Erweiterung der Realität präsentieren.

Die Abbildung 1.2 zeigt zwei Generationen von Head-Mounted-Displays. Während das Bild 1.2(b) einen Typ der neueren Generation zeigt, zeigt die Abbildung 1.2(a) das „The Sword Of Damocles“ [101] von Ivan Sutherland aus dem Jahr 1968, welches das erste Head-Mounted-Display war. Die Idee mit Hilfe von Displays Realität und Virtualität zu verbinden wurden von Sutherland schon drei Jahre vorher in der Arbeit über „The Ultimate Display“ formuliert [100]. Von der Zeit des Damoklesschwertes bis in das Jahr 1994 wurden zahlreiche neue Displaytypen entwickelt und auch auf den Markt gebracht. Milgram hat ebenfalls im Jahr 1994 in seinem Artikel [70] eine Einteilung von Displays in verschiedene Klassen vorgestellt und dabei auch den Begriff der *Mixed Reality* geprägt.



Abbildung 1.1: Beispiel einer Überlagerung der Realität mit virtuellen Inhalten: (a) zeigt das unveränderte Bild aufgenommen mit einer Video Kamera; (b) zeigt das gleiche Bild, aber zusätzlich mit augmentierten Sensordaten des Fahrzeugs

Eine Darstellung einer vereinfachten Version des Virtualitäts-Kontinuum (engl.: *Virtuality-Continuum*) ist in Abbildung 1.3 zu sehen.

Etwa in der selben Zeit hat Azuma die heute weit akzeptierte Definition von Augmented Reality formuliert [3]. Dieser Definition zufolge kombiniert ein AR-System die Realität mit der Virtualität und präsentiert dem Benutzer dies in Echtzeit. Dabei muss der Nutzer mit dem System interagieren können und die virtuellen Objekte mit realen Objekten im dreidimensionalen Raum in Beziehung stehen. Diese Beziehung wird im englischen *registered* genannt. Um die realen Objekte virtuell annotieren zu können, muss ein AR-System zunächst Informationen über die Position und Orientierung dieser Objekte haben. Und diese Information muss nach der Definition von Azuma in Echtzeit vorliegen. Dieser Vorgang des Verfolgens von Position und Orientierung von Objekten wird Tracking genannt. Greg Welch schreibt in [113], dass es nicht eine perfekte Trackingtechnologie für alle Situationen gibt, wohl aber ausgewählte kombinierte Methoden für bestimmte Anwendungen. Werden aber die Anforderungen einer Augmented Reality Anwendung nicht mit einem einzelnen Sensor abgedeckt, so kann die Sensorfusion eine Lösung bieten.

Eine Sensorfusion verknüpft Informationen von verschiedenen Sensoren und erzeugt daraus ein Signal mit hochwertigeren Informationen, welches dann die Anforderungen der Anwendung besser erfüllt.

Nun ist die Aufgabe zwei Ströme von Sensordaten miteinander zu fusionieren, um daraus einen einzelnen Strom an Daten zu generieren. Durch technische Eigenheiten der Sensoren oder auch durch unterschiedliche Übertragungswege hinkt ein Signal stets dem anderen hinterher. Diese Arbeit stellt Methoden und Verfahren vor, wie der Zeitversatz zwischen diesen Sensoren kalibriert und korrigiert werden kann.

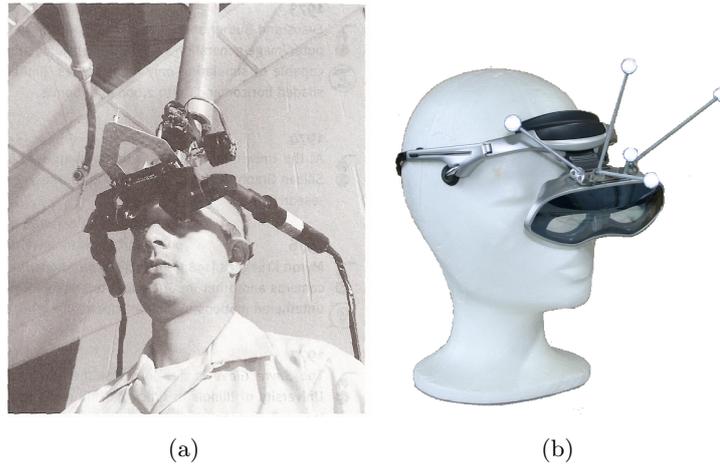


Abbildung 1.2: Darstellung zweier Generationen von Head-Mounted-Displays (HMD); (a) zeigt *The Sword of Damocles*, als eines der ersten HMDs entwickelt von Ivan Sutherland im Jahre 1968 [101]; (b) zeigt ein HMD der neuesten Generation (mit freundlicher Genehmigung von Martin Bauer)

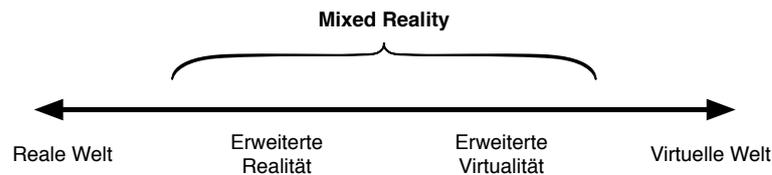


Abbildung 1.3: Vereinfachtes Virtualitäts-Kontinuum nach Milgram [70]

Ein besonderes Gebiet der Augmented Reality beschäftigt sich mit der Durchdringung des alltäglichen Lebens mit Augmented Reality. Dieser Teilbereich ist eine Zusammenführung der Forschungsgebiete des *ubiquitous computing* und Augmented Reality und heißt demnach *Ubiquitous Augmented Reality* (UAR). In das Deutsche kann dies mit *Allgegenwärtige Erweiterte Realität* übersetzt werden. Diese Vision setzt hochdynamischen Sensornetze voraus, um das oben besprochene Trackingproblem zu lösen. Und nachdem es keine einzelne Trackingtechnologie für alle Anforderungen gibt, ist auch die UAR auf Sensorfusion angewiesen. Doch im Gegensatz zur klassischen AR steht bei UAR zum Zeitpunkt der Entwick-



Abbildung 1.4: Spatial Relationship Graph eines Trackingsystems.

lung noch nicht fest, welche Sensoren später benutzt werden oder miteinander fusioniert werden. Dies bedeutet, dass eine zeitliche Kalibrierung nicht schon zum Zeitpunkt der Entwicklung eines Systems vorgenommen werden kann, sondern Methoden benötigt werden, welche eine zeitliche Kalibrierung der Sensoren während der Laufzeit erlauben.

## 1.2 Problembeschreibung

Mit dynamischen Fehlern werden in der AR Fehler bezeichnet, die nur auftreten, wenn Teile des Systems in Bewegung sind. Betrachtet man das augmentierte Bild in Abbildung 1.1(b), dann kann man eine virtuelle Box erkennen, welche um eine Person herum gezeichnet wurde. Liefert nun der Sensor die Positionsdaten der zu kennzeichnenden Person später als das Videosystem das zugehörige Bild und die Person im Bild bewegt sich, dann wird die augmentierte Box der realen Person immer ein Stück hinterherlaufen. Azuma hat dieses Phänomen in seiner Dissertation auch „swimming around“ genannt [5]. Je größer und schneller die Dynamik in einem System desto größer sind die Auswirkungen eines solchen Fehlers.

Besonders drastisch wirken sich dynamische Fehler während einer räumlichen Registrierungsprozedur aus. Räumliche Registrierungen werden benötigt um die räumliche Beziehung zwischen zwei Objekten zu bestimmen, zum Beispiel zwischen einem Sensor und einem Display. Treten während einer Registrierung Dynamiken auf, werden also die Sensoren bewegt, so bewirken die dynamischen Fehler einen Fehler in der Registrierung. Dieser hat dann wiederum Auswirkungen in den Anwendungen, selbst zu Zeitpunkten, wenn dort gerade keine Dynamiken mehr auftreten. Aus einem dynamischen Fehler während des Registrierungs Vorgangs wird somit ein systematischer statischer Fehler.

Dynamische Fehler können eine Vielzahl an störenden Effekten auslösen und die Störung des visuellen Eindrucks des Benutzers ist nur eine mögliche Auswirkung. Die bereits angesprochene relative Latenz zwischen zwei Sensoren führt auch bei einer Fusion der beiden Sensoren zu Fehlern im Ergebnis.

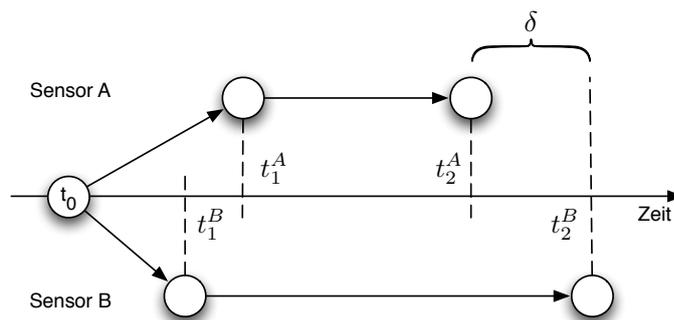


Abbildung 1.5: Zeitlicher Ablauf zweier Messungen des gleichen Ereignisses.

Die Abbildung 1.5 zeigt den schematischen Ablauf zweier Messungen des glei-

chen Ereignisses. Zum Zeitpunkt  $t_0$  findet das reale physikalischen Ereignis statt. Dieses wird von zwei Sensoren A und B vermessen, wobei A und B zwei verschiedene Sensoren sind. Ein Sensor muss ein gemessenes physikalisches Ereignis in der Regel erst verarbeiten. Diese Verarbeitung ist in jedem Sensor anders implementiert, weswegen unterschiedliche Latenzen auftreten. Dies ist in Abbildung 1.5 dargestellt. Die Daten werden nun an das Trackingsystem auf unterschiedlichen Wegen weitergereicht. Die Möglichkeiten der Übertragung sind vielfältig. Neben Netzwerkübertragungstechniken und seriellen Busse existieren noch viele weitere Techniken. Jedes Übertragungstechnik bringt weitere verschiedene Latenzen in das System, ein Umstand der auch in der Abbildung 1.5 zu sehen ist. Mit  $t_2^A$  und  $t_2^B$  seien nun die Zeitpunkte benannt, zu welchen die Daten im System mit einem Zeitstempel versehen werden. Die Differenz der Zeitpunkte  $t_2^A$  und  $t_2^B$ , welche den absoluten Sensorlatenzen entsprechen, wird die *relative Latenz*  $\delta$  genannt. Diese Arbeit beschäftigt sich nur mit der Bestimmung dieser relativen Latenz  $\delta$  und nicht mit der Bestimmung der absoluten Latenzen der einzelnen Sensoren. Die Latenzen der Sensoren selber sind für eine korrekte Sensorfusion nicht von Bedeutung. Ein Beispiel, wie sich die relative Latenz auswirkt ist in Abbildung 1.6 zu sehen. Dort sind zwei Ströme normierter Positionsmessungen dargestellt. Es ist klar zu erkennen, dass die blauen und die roten Messpunkte nahezu identisch sind, aber die blau dargestellten Messergebnisse ein wenig verzögert sind. In Abbildung 1.7 ist der Fehler ausgelöst von einer relativen Latenz und einer Bewegung der beiden Sensoren quantitativ dargestellt. Klar zu erkennen ist die Abhängigkeit des dynamischen Fehlers von der Geschwindigkeit.

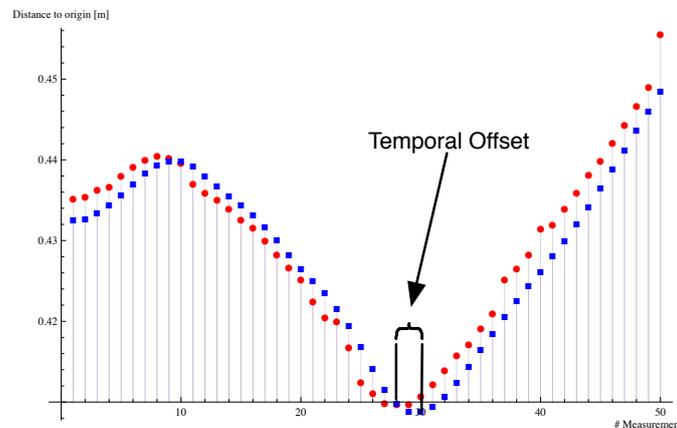


Abbildung 1.6: Auswirkung der relativen Latenz; Der Plot zeigt zwei Ströme normierter Positionsmessungen; Die blau dargestellten Messergebnisse (Rechtecke) sind um  $\delta$  verzögert.

Nun soll das Problem der relativen Latenz mathematisch modelliert werden. Dazu betrachtet man zwei Sensoren A und B, welche das identische physikalische

Ereignis messen. Das reale Ereignis sei nun mit  $S(t)$  bezeichnet und die Messungen der Sensoren A und B mit  $S^A(t)$  und  $S^B(t)$ . Die Latenzen der Sensoren seien mit  $\delta^A$  und  $\delta^B$  bezeichnet. Es sei weiterhin angenommen, dass die beiden Sensoren zu einander registriert sind und somit alle Messungen bezüglich eines Koordinatensystems dargestellt werden können. Damit kann man die relative Latenz wie folgt modellieren

$$S^A(t) = S(t + \delta^A) + e^A(t) \quad (1.1)$$

$$S^B(t) = S(t + \delta^B) + e^B(t) \quad (1.2)$$

Wobei  $S(t)$  das kontinuierliche Signal und  $e^A(t)$  und  $e^B(t)$  ein Fehlerterm ist, der Störeinflüsse wie Rauschen, fehlerhafte Registrierung und andere Sensormessfehler beinhaltet. Nachdem aber die absoluten Latenzen in dieser Arbeit nicht behandelt werden, sondern nur relative Latenz, ist eine einfachere Modellierung von Vorteil. Sei nun o. B. d. A. angenommen, dass Sensor A die Daten schneller an das Messsystem liefert als Sensor B, dann kann das Problem mit  $\tilde{S}(t) = S(t + \delta^A)$  wie folgt geschrieben werden

$$S^A(t) = \tilde{S}(t) \quad (1.3)$$

$$S^B(t) = \tilde{S}(t + \delta). \quad (1.4)$$

Der Parameter  $\delta = \delta^B - \delta^A$  bezeichnet die gesuchte relative Latenz. Um diese Latenz zu bestimmen wird nun ein Signal, beispielsweise das Signal B, systematisch verschoben und dabei kontinuierlich die Signale A und B verglichen. Bei einer bestimmten Verschiebung wird eine maximale Übereinstimmung auftreten, welche dann als die gesuchte relative Latenz gewertet wird.

Die relative Latenz zwischen Sensoren ist kein statischer Wert. So kann beispielsweise eine höhere Last des Messrechners zu einer Veränderung der relativen Latenz führen. Bei einer hohen Genauigkeitsanforderung ist somit eine fortlaufende Rekalibrierung der relativen Latenz nötig. Ein Verfahren zur Kalibrierung der relativen Latenz muss also echtzeitfähig sein.

### 1.3 Lösungsvorschlag

Nach der Formulierung des Problems sollen nun kurz mögliche Lösungen vorgestellt werden. Eine Möglichkeit die relativen Latenzen erst gar nicht entstehen zu lassen ist die Verwendung von identischen Sensoren und von identischen Übertragungskanälen. Dieses Verfahren wird von einigen optischen Sensorsystemen benutzt. Aber durch die technischen Voraussetzungen ist dieses Verfahren nicht für alle Sensoren geeignet.

Ein etwas allgemeinerer Ansatz beruht auf dem Vergleichen zweier Messsignale, während ein Signal zeitlich gegen das andere verschoben wird [65, 93]. Wenn eine maximale Ähnlichkeit zwischen den verschobenen Signalen auftritt, wird dies als

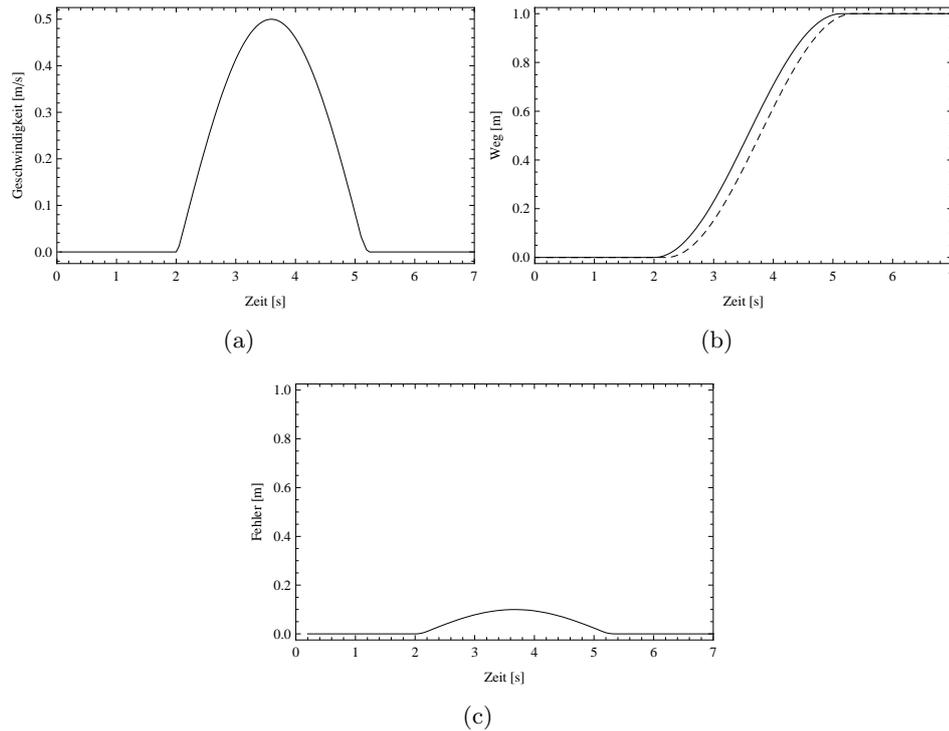


Abbildung 1.7: Auswirkungen der relativen Latenz zwischen zwei Sensoren ab einem theoretischen Beispiel. Die zwei Sensoren sind fest miteinander verbunden und messen die zurückgelegte Strecke; (a) zeigt die Geschwindigkeit beider Sensoren; (b) zeigt die von den Sensoren *gemessene* Distanz. Ein Sensor, angedeutet durch die gestrichelte Linie im Plot, hängt 200 ms hinterher; (c) zeigt die Auswirkung der relativen Latenz als geometrischen Abstand zweier Messwerte zum gleichen Zeitpunkt

relative Latenz gewertet. Beide Autoren von [65, 93] verwenden ein geometrisches Maß um die beiden Signale zu vergleichen. Dieses Vorgehen bietet sich an, da die Kalibrierung der relativen Latenz zur gleichen Zeit wie die räumliche Registrierung durchgeführt wird.

In dieser Arbeit wird ein Kalibrierverfahren vorgestellt, welches ebenfalls auf einer Zeitverschiebung basiert, aber als Vergleichsmaß die normierte Kreuzkorrelation propagiert. Diese Methode ist in der Signalverarbeitungstheorie weit verbreitet. Es wird beispielsweise in Systemen mit Radar oder Sonar [17, 107] benutzt, um die Laufzeit eines Signals zu bestimmen. Das Problem wird in dem Gebiet der Signalverarbeitung Zeitversatzschätzung (engl.: *Time Delay Estimation*, *TDE*) genannt. Dieses Verfahren muss aber auf die speziellen Anforderungen von Augmented Reality Anwendungen angepasst werden. Da die Kreuzkorrelation nur eindimensionale Signale verarbeiten kann, müssen die mehrdimensionalen räumlichen Messergeb-

nisse erst in der Dimension reduziert werden. Zwei verschiedene Sensoren liefern die Messergebnisse in der Regel mit verschiedenen Frequenzen, dies macht ein direktes Vergleichen der Signale unmöglich. Die Sensoren müssen also zusätzlich zu einander synchronisiert werden, um zwei vergleichbare Signale zu erhalten. Diese zwei Schritte seien nur als Beispiel aufgeführt, um zu zeigen, dass für eine erfolgreiche und präzise Kalibrierung der relativen Latenz Vor- und Nacharbeiten durchgeführt werden müssen. Als zentrales Element bleibt jedoch die normierte Kreuzkorrelation als Vergleichsfunktion der beiden Signale. Neben den Methoden für eine korrekte Kalibrierung werden in dieser Arbeit auch Einflüsse betrachtet, die eine Kalibrierung stören können. Hier sei nur kurz das Rauschen der Sensoren und Fehler in der räumlichen Registrierung der Sensoren genannt. Diese Fehler treten aber in der Realität nie getrennt von einander auf, was ein direktes Messen der Auswirkungen unmöglich macht. Daher wurde ein Framework für Monte Carlo Simulationen aufgebaut, welches ein direktes Messen der Auswirkungen erlaubt. Somit ist es möglich, die benutzten Verfahren so anzupassen, dass sie möglichst robust gegen Störeinflüsse werden. In den Evaluationen wird gezeigt, dass die während des Trackings mit zwei zueinander registrierten Sensoren auftretenden Fehler mit einer Korrektur des Zeitversatzes stark verringert werden können.

Teile dieser Arbeit sind im Rahmen des Forschungsprojekts *trackframe* der Bayerischen Forschungstiftung entstanden. Insbesondere ist die Idee zur Zeitkalibrierung und einige der Grundlagen für hier aufgeführte Verfahren und Teile der aufgeführten Versuche in enger Zusammenarbeit mit Manuel Huber entstanden. Einige der hier vorgestellten Ergebnisse sind in der Dissertation von Huber [49] nachzulesen.

## 2 Hintergrund

In diesem Kapitel werden die nötigen Grundlagen und Hintergründe vermittelt, die für ein Verständnis der zeitlichen Kalibrierung nötig sind. Dabei wird zu Anfang auf die Grundzüge von Tracking in Augmented Reality Anwendungen eingegangen und auch die weiterführenden Tracking-Themen wie Sensorfusion und Tracking-Frameworks beschrieben. Am Ende des Kapitels wird dann eine grundsätzliche Darstellung der in einer AR Anwendung auftretenden Latenzen aufgezeigt, die für die späteren Teile der Arbeit nötig sind.

### 2.1 Tracking

Um eine genau Überlagerung der Realität zu ermöglichen, müssen dem AR-System die Lage und Position aller beteiligten Objekte und Subjekte bekannt sein. Mit Hilfe dieser Orts- und Lageinformation und vorher durchgeführten Registrierungen können die Überlagerungen des realen Bildes mit virtuellen Inhalten registriert in Echtzeit generiert werden. Diese Bestimmung von Position und Orientierung wird Lokalisierung oder im Bereich der Erweiterten Realität auch *Tracking* genannt. Dabei unterscheidet man nicht auf welche Weise die Position und Orientierung bestimmt wird, sondern jedes Verfahren, welches eine Position oder Orientierung liefert wird als Tracking bezeichnet. Eine Möglichkeit ist es die Parameter direkt zu messen. Das satellitengestützte GPS beispielsweise liefert eine absolute eindeutige Position auf der Erdoberfläche. Eine weitere Möglichkeit ist es, die Veränderungen der Position und Lage zu messen. Ein Beispiel hierfür ist die Koppelnavigation in der Seefahrt. Ausgehend von einer bekannten Position kann unter Berücksichtigung von Fahrtrichtung, Geschwindigkeit und Zeit eine aktuelle Position berechnet werden. Außerhalb der Forschungsrichtungen der Augmented- und Virtual-Reality wird der Begriff des Tracking nur für die Bestimmung durch Veränderungen gebraucht, da die Objekte dabei verfolgt, also getrackt, werden. In dieser Arbeit wird unter dem Begriff Tracking für beide prinzipiellen Methoden äquivalent gebraucht.

Wie bereits ausgeführt versteht man unter dem Begriff des Tracking die Bestimmung von Position und Orientierung von Objekte oder Subjekten. Diese Kombination der räumlichen Parameter wird *Pose* genannt. Eine Pose hat sechs Freiheitsgrade, denn sowohl die Position als auch die Orientierung benötigen jeweils drei Freiheitsgrade. Um die Pose eines Objekts messen zu können, stehen dem Entwickler eines Augmented Reality Systems verschiedenste technische Möglichkeiten zur Verfügung. Eine Auswahl hiervon sind in Abschnitt 2.4 oder auch in [113, 89] aufgeführt. Welches Trackingsystem oder auch welche Kombination von Tracking-

systemen verwendet wird ist von Faktoren wie Genauigkeit, Kosten, Unterhalt, Arbeitsvolumen oder auch Zuverlässigkeit abhängig.

Bei einer Anwendung, welche das Kommissionieren von Waren in einem Lagerhaus unterstützt [95], sind die Anforderung an das Volumen besonders groß, denn die Anwendung soll in der gesamten Lagerhalle funktionieren. Bezüglich der Genauigkeit hat diese Anwendung unterschiedliche Anforderungen. So reicht für die Navigation auf den Gängen eine Genauigkeit im Meterbereich. Soll aber beispielsweise eine Schraube in einer Kiste abgelegt werden, so ist eine Genauigkeit im Bereich von Zentimetern gefordert. Dieses einfache Beispiel zeigt, dass die Anforderungen an ein Trackingsystem selbst innerhalb einer Anwendung sehr unterschiedlich sein können. Wie Welch in dem Artikel [113] beschreibt, gibt es keinen einzelnen Trackingsensor, der alle diese heterogenen Anforderungen erfüllt, sondern es werden meist Kombinationen von solchen Trackingsensoren benötigt. Solche Kombinationen bestehen in der Regel aus unterschiedlichen Sensoren mit verschiedenen Charakteristika und einem Softwaresystem, welches die verschiedenen Trackingsensoren zu einem einzelnen Trackingsystem zusammenfasst.

## 2.2 Graphen über Räumliche Abhängigkeiten (SRG)

In dem einfachsten Trackingszenario wird die Pose eines Gegenstandes von einem einzelnen Trackingsensor bestimmt. Wird die Pose des selben Objekts von einem weiteren Trackingsensor bestimmt, so liefern die beiden Sensoren verschiedene Posen, denn jeder Sensor misst die Position relativ zu einem anderen Bezugssystem. Es zählt also beim Tracking nicht die absolute Pose, sondern immer nur die Pose im Abhängigkeit zu einem Bezugssystem. Solche Bezugssysteme heißen auch Koordinatensysteme. Beim Tracking werden also nicht Posen absolut gemessen, sondern immer nur räumliche Abhängigkeiten (engl.: *Spatial Relationships*). In der Literatur werden solche räumlichen Beziehungen als Graphen dargestellt [105]. Diese Graphen heißen räumliche Beziehungsgraphen (engl.: *Spatial Relationship Graph*, *SRG*). In der Abbildung 2.1(a) ist das obige Beispiel durch einen SRG beschrieben.

Ein Graph besteht aus Knoten und Kanten. Im Falle eines SRGs stehen die Knoten für Koordinatensysteme und die Kanten für räumliche Beziehungen oder auch die Transformationen zwischen diesen Koordinatensystemen. In dem Beispiel in Abbildung 2.1(a) sind Kanten zwischen den Bezugssystemen der Sensoren und dem zu trackenden Objekt eingezeichnet. Neben den eingezeichneten räumlichen Beziehungen ist auch die Beziehung zwischen den beiden Bezugssystemen von Interesse. Diese Beziehung ist in der Abbildung 2.1(a) als gestrichelte Kante dargestellt. Mit Kenntnis über diese Beziehung lassen sich die Messungen eines Sensors in das Bezugssystem des anderen Sensors umrechnen und somit sind dann beide Messungen direkt miteinander vergleichbar.

Vorgänge, wie das Umrechnen von Messungen in ein anderes Bezugssystem (Koordinatentransformationen), sind Vorgänge, die in der Praxis sehr oft auftreten. In ubiquitären AR Systemen können die Anforderungen für Koordinatentransfor-

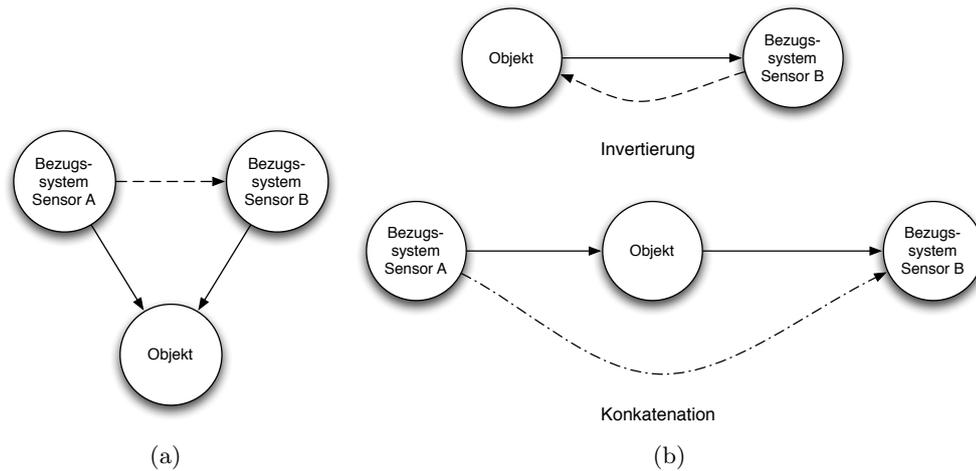


Abbildung 2.1: Beispiel für SRG: (a) zeigt ein Objekt getrackt von zwei Sensoren; (b) zeigt die Patterns Invertierung und Konkatenation

mationen auch zur Laufzeit eines Systems auftreten, so dass diese Abhängigkeiten nicht a priori bestimmt werden können. Neben diesem relativ einfachen Problem, der Transformation von Koordinatensystemen, existieren kompliziertere Vorgänge wie beispielsweise Registrierungen. Um dieses Problem zu lösen wurden von Pustka u. a. in [82] Patterns entwickelt, die eine wohl definierte Struktur haben und mit getesteten Algorithmen hinterlegt sind. In Abbildung 2.1(b) sind zwei solche Patterns abgebildet. Diese beiden Patterns können verwendet werden, um die Messdaten des Sensors A in das Bezugssystem des Sensors B zu transformieren. Für die Details dieses Prinzips siehe die Arbeiten von Newman [75], Huber [50], Echtler [25] und Pustka [82]. Auf den eben vorgestellten Prinzipien von SRGs und Patterns baut das ubitrack Trackingframework auf. Im Abschnitt 2.6.4 wird dieses Framework näher beschrieben.

Neben der formellen Notation und Visualisierung müssen die Trackingdaten auch mathematisch beschrieben werden. Dazu soll der folgende Abschnitt einen kurzen Überblick geben.

## 2.3 Darstellung von Trackingdaten

In Augmented Reality Anwendungen werden reale Objekte mit virtuellen Objekten überlagert. Um diese Überlagerung herzustellen muss eine Anwendung die Position und Orientierung der beteiligten Objekte im Raum kennen. Die Position und Orientierung beschreiben den absoluten geometrischen Zustand eines Objekts im Raum, während Translation und Rotation eine geometrische Veränderung beschreiben. Sowohl die Beschreibung des absoluten Zustands als auch die Beschreibung der Veränderung des Zustands können auf die gleiche Art dargestellt werden. Im Folgenden wird auf verschiedene mathematische Repräsentation von geometri-

schen Trackingdaten eingegangen.

### 2.3.1 Pose Darstellung

Eine Pose ist definiert durch die Information über Position und Orientierung eines Objekts im drei dimensionalen Raum. Die Definition macht keinerlei Vorschriften auf welche Weise diese Informationen repräsentiert werden sollen. Daher gibt es in der Literatur auch ein Vielzahl an verschiedenen Varianten.

Zunächst für die Position ist die Darstellung bis auf die Wahl des Bezugssystems eindeutig

$$\mathbf{x} = (x_1, \dots, x_n)^T \in \mathbb{R}^n. \quad (2.1)$$

Bei Augmented Reality Anwendungen ist die Dimension üblicherweise  $n = 3$ . Dies erlaubt es eine Position als einen Vektor mit drei Elementen zu beschreiben. Daneben existieren noch weitere Repräsentationen, die hier nicht genutzt werden.

Auch für die Darstellung von Orientierungen existieren mehrere Möglichkeiten. Eine übliche Art der Darstellung sind Euler Winkel, welche eine Orientierung mit drei Winkeln beschreibt. Doch diese Art der Darstellung benötigt eine weitere Definition, denn durch die Angabe der Winkel alleine ist noch nicht festgelegt, um welche Achsen gedreht werden soll. Für die Repräsentation von Euler Winkeln existieren nach Craig [21] 24 verschiedenen Varianten. Während die Variantenvielfalt noch durch eindeutige Definitionen und klare Vereinbarungen bewältigt werden kann, stellen Mehrdeutigkeiten eine größere Herausforderung dar. Eine andere Art der Darstellung sind Rotationsmatrizen ( $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ ), die jedoch in einem erheblichen Maße überbestimmt sind. Denn es werden insgesamt neun Matrixeinträge benötigt, um die drei Freiheitsgrade einer Rotation darzustellen. Weitere Darstellungen von Rotationen sind in zahlreichen Arbeiten erschienen, wobei hier besonders [21] hervorzuheben ist. In diesem Buch sind neben den üblichen Darstellungen auch Konvertierungen zwischen diesen beschrieben.

In dieser Arbeit sind Quaternionen als Repräsentationsform gewählt worden. Quaternionen kommen mit einer geringeren Redundanz gegenüber den Rotationsmatrizen aus. Es gibt einfache Verfahren für die lineare Interpolation zwischen zwei Orientierungen und es ist eine in der AR übliche Form der Darstellung. Hier werden nun die für die weitere Arbeit nötigen mathematischen Hintergründe aufgeführt. Für mehr Informationen sei auf Literatur wie [64] verwiesen oder für historisch Interessierte auf [37].

Ein Hamilton Quaternion  $q$  kann man sich als komplexe Zahl mit drei verschiedenen imaginären Teilen und einem Realteil vorstellen. Ein Quaternion kann damit wie folgt geschrieben werden

$$q = iq_x + jq_y + kq_z + q_w, \quad (2.2)$$

wobei  $q_x$ ,  $q_y$  und  $q_z$  die imaginären Teile sind und  $q_0$  der Realteil. Diese vier Werte kann man auch als Vektor schreiben  $\tilde{q} = (q_w, q_x, q_y, q_z)^t \in \mathbb{R}^4$ .

Die Norm von Quaternionen ist über die Euklidische Norm definiert

$$\|q\|^2 = q_x^2 + q_y^2 + q_z^2 + q_w^2. \quad (2.3)$$

Ein Quaternion mit der Länge  $\|q\| = 1$  nennt man ein *Einheitsquaternion*.

Um eine Skalierung durch eine Rotationsoperation zu vermeiden, müssen Quaternionen normiert werden. Durch eine Normierung ist es möglich auf einen Parameter der Beschreibung eines Quaternionen zu verzichten, ohne Information zu verlieren. In der Regel wird auf diese Reduzierung verzichtet und die Quaternionen werden als ein Viertupel beschrieben. Für weitere Details über die Nutzung von Quaternionen für die Darstellung einer Orientierung sei auf [64, 21, 33] verwiesen.

### 2.3.2 Klassifikation von Tracking Fehlern

Jeder Messvorgang ist mit Ungenauigkeiten verbunden. Diese Messungenauigkeiten beeinträchtigen zum einen die Genauigkeit der Kalibrierung, zum anderen können die Differenzen zwischen zwei Sensoren benutzt werden, um die relative Latenz zu bestimmen. Daher ist eine Beschreibung von Messfehlern bzw. der Differenz von Sensormessdaten erforderlich.

Prinzipiell kann man dynamische und statische Fehler unterscheiden. Während dynamische Fehler sich nur bei Bewegungen im System auswirken und in statischen Situationen verschwinden, sind statische Fehler auch in Ruhelage vorhanden. Die relative Latenz zwischen zwei Sensoren führt bei einer Fusion der beiden Sensoren zu einem dynamischen Fehler. Diese Art von Fehlern ist für einen Großteil des Gesamtfehlers in einem AR System verantwortlich [44, 3]. In den folgenden Absätzen wird auf die statischen Fehler eingegangen, die einem Sensor zu eigen sind. Diese Fehler lassen sich wieder in zwei Klassen aufteilen, in Rauschfehler und in systematische Fehler.

Rauschfehler sind zufällige sich nicht wiederholende Abweichungen von einem korrekten Wert. Das heißt bei wiederholenden Messungen wird das Ergebnis immer nicht reproduzierbar um den wahren Wert streuen, wobei der Mittelwert gleich dem wahren Wert ist. Diese Tatsache lässt zum einen die Beschreibung dieses Fehlers als Zufallsprozess mit gaußscher Verteilung zu und zum anderen erlaubt es bei statischen Messungen die Reduktion dieses Fehlers durch Mittelung. Die Mittelung ist in der Regel nicht möglich, da die Sensoren bewegt werden. Aber durch eine Filterung, die später beschrieben wird, kann das Rauschen zumindest reduziert werden. Wie bereits angesprochen, kann das Rauschen als gaußsche Zufallsvariable beschrieben werden, die als Mittelwert die Null hat. Somit kann das Rauschen im eindimensionalen Fall mit der Standardabweichung oder im mehrdimensionalen Fall mit einer Kovarianzmatrix beschrieben werden [7, 91]. Das Rauschen eines Sensors hat negative Auswirkungen auf die Genauigkeit der zeitlichen Kalibrierung und wird im Abschnitt 8.3.2 für eine Simulation benötigt.

Neben den Fehlern, welche durch Rauschen verursacht werden, existieren noch systematische Fehler. Diese Fehler sind reproduzierbar, das heißt unter gleichen

Voraussetzungen wird ein Sensor immer den gleichen systematischen Fehler produzieren. Es ist daher nicht sinnvoll, diesen Fehler als Zufallsvariable zu modellieren, sondern es muss ein Maß für den Fehler gefunden werden, um ihn zu quantifizieren. Im Folgenden werden sowohl für Fehler in der Position als auch für Fehler in der Orientierung jeweils Maße präsentiert, die eine solche Quantifizierung erlauben.

**Fehler in der Position** Um den Fehler einer Positionsschätzung anzugeben, wird in dieser Arbeit die euklidische Norm

$$\|\mathbf{x}\|_2 = \sqrt{x^2 + y^2 + z^2}$$

als Fehlermaß benutzt. Statt  $\|\mathbf{x}\|_2$  kann auch kurz  $\|\mathbf{x}\|$  geschrieben werden.

Seien nun  $\mathbf{t}$  die wirkliche Position und  $\hat{\mathbf{t}}$  eine Messung der wirklichen Position, dann ist der relative Messfehler definiert als  $e_{rel} = \|\hat{\mathbf{t}} - \mathbf{t}\|/\|\mathbf{t}\|$ . Der absolute Fehler wird mit  $e_{abs} = \|\hat{\mathbf{t}} - \mathbf{t}\|$  definiert.

**Fehler in der Orientierung** Während für Positionen das Maß für eine Abweichung noch intuitiv begreifbar ist, ist das für Fehler in der Orientierung nicht mehr so leicht möglich. Ein Fehler bei der Messung einer Orientierung kann aufgefasst werden als Differenz zwischen der korrekten Orientierung und der gemessenen Orientierung. Diese Differenz wird in Form einer Rotation angegeben, welche die korrekte Orientierung auf die gemessene abbildet. Daher wird im Folgenden von Rotationsdifferenzen gesprochen. Für Rotationsdifferenzen existiert eine Vielzahl von Maßen, die in den verschiedenen Darstellungen der Rotation begründet sind.

Eine Möglichkeit ist es, die Differenz der Drehwinkel in der Repräsentation der Rotation mit Drehachse und Drehwinkel zu nutzen. Diese Methode ist zwar leicht zu berechnen, berücksichtigt aber nur die Drehwinkel, während die Drehachse unberücksichtigt bleibt.

Im Gegensatz dazu benutzt Daniilidis [23] ein Methode für die Quantifizierung des Fehlers, welche auf Quaternionen operiert. Das Maß des Fehlers wird durch den RMS der Differenz des Referenzquaternions  $q$  und der Messung  $\hat{q}$  ausgedrückt. Hier ist zu beachten, dass beide Quaternionen normiert sein müssen, also  $\|q\| = \|\hat{q}\| = 1$  gilt. Zusätzlich müssen beide Quaternionen gleich ausgerichtet sein, damit eine Mehrdeutigkeit ausgeschlossen werden kann. Der Fehler kann dann wie folgt dargestellt werden

$$e = \|q - \hat{q}\|.$$

Eine weitere Metrik wird von Horaud in [45] eingeführt. Diese Metrik arbeitet auf Rotationsmatrizen  $\mathbf{R}$  und  $\hat{\mathbf{R}}$ . Dazu wird die Differenz der Matrizen gebildet und die Frobenius Norm dieser Differenz berechnet. Die Frobenius Norm ist definiert als die Summe aller Quadrate der Matrixeinträge

$$\|\mathbf{R}\|_2 = \sqrt{\sum_{i,j} r_{i,j}^2}.$$

Dadurch ergibt sich für Differenz der Orientierungen  $e_{\mathbf{R},\hat{\mathbf{R}}} = \|\mathbf{R} - \hat{\mathbf{R}}\|_2$ . Die Frobenius Norm und die von Daniilidis genutzte Norm stehen in einem engen Zusammenhang und sind somit von der Aussage vergleichbar [23].

Neben diesen beiden Normen werden in der Literatur auch die Repräsentation der Euler Winkel für eine Fehlernorm genutzt. Seien  $r = (r_x, r_y, r_z)$  und  $\hat{r} = (\hat{r}_x, \hat{r}_y, \hat{r}_z)$  zwei Orientierungen dargestellt mit Eulerwinkeln. Die Differenz dieser Eulerwinkel ist dann durch die Distanz im  $\mathbb{R}^3$  Vektorraum definiert

$$e = \|r - \hat{r}\| = \sqrt{(r_x - \hat{r}_x)^2 + (r_y - \hat{r}_y)^2 + (r_z - \hat{r}_z)^2}. \quad (2.4)$$

Pentrieder benutzt diese Fehlernorm in [78] für die Auswertung von Monte-Carlo-Simulationen die jeweiligen Mittelwerte der drei Euler Winkel. Für den Gesamtfehler wird dann der Durchschnitt der drei Mittelwerte gebildet. Diese Fehlerdarstellung ist jedoch von der jeweiligen Definition der Euler Winkel abhängig.

### 2.3.3 Tracking Daten als stochastische Signale

Da, wie schon in der Einleitung erwähnt, in dieser Arbeit Trackingdaten als Signale interpretiert und mit Methoden der Signalverarbeitung untersucht werden, sollen an dieser Stelle die Grundzüge der Darstellung von Trackingdaten als Signale beschrieben werden.

Dazu müssen zuerst die entsprechenden Begriffe definiert werden. Nach [106] versteht man unter einem Signal die Repräsentation einer Information. Die Information kann beispielsweise ein Spannungsverlauf sein, der Verlauf einer Aktie oder auch die Bewegung eines Sensors. Prinzipiell kann man Signale unterscheiden in kontinuierliche und zeitdiskrete Signale. Kontinuierliche Signale können durch eine Funktion  $s(t) \in \mathbb{R}^n$  mit  $t \in \mathbb{R}$  beschrieben werden. Die Signale sind nicht auf das Eindimensionale beschränkt, so kann auch die Bewegung eines Sensors in allen sechs Freiheitsgraden als Signal betrachtet werden. In der Praxis liegen Signale zwar in kontinuierlichen Form vor, sie können jedoch nur in zeitdiskreter Form gemessen werden. Die Abtastfrequenz der Sensoren wird durch mehr oder weniger präzise Zeitgeber, auch Uhren genannt, gesteuert. Das Ziel ist dabei die Messungen mit möglichst äquidistanten Zeitabständen durchzuführen. Jedoch sind diese Zeitgeber oft sehr unpräzise, weil beispielsweise instabile Quarze benutzt werden. Damit ergeben sich Ungenauigkeiten und Schwankungen in den Wiederholraten. Dieser Effekt wird in der Literatur mit „crystal time shift“ bezeichnet.

Eine Messung eines physikalischen Ereignisses kann als Diskretisierung eines kontinuierlichen Signals interpretiert werden. Sei nun  $\Delta t$  der Zeitraum zwischen zwei Messungen und  $s(t)$  das kontinuierliche Signal, dann kann das diskrete Signal  $\hat{s}_i$  wie folgt beschrieben werden

$$\hat{s}_i = s(i\Delta t) \quad , i \in \mathbb{Z}.$$

Es ist auch möglich, aus einem zeitdiskreten Signal  $\hat{s}_i$ , welches beispielsweise eine Messung darstellt, wieder ein kontinuierliches Signal zu erzeugen. Dies kann zum

Beispiel durch Interpolation zwischen den Messungen geschehen. In Abschnitt 6.1 wird auf die Interpolation von räumlichen Messungen näher eingegangen. Aber dieser Interpolation sind Grenzen durch das Nyquist-Shannon-Abtasttheorem [56] gesetzt.

Das Nyquist-Shannonsche Abtasttheorem definiert eine untere Schranke für die Abtastfrequenz eines kontinuierlichen Signals. Liegt die Abtastfrequenz unter dieser Schranke, so kann aus dem zeitdiskreten Signal das Ursprungssignal nicht eindeutig rekonstruiert werden. Abbildung 2.2 zeigt die Frequenzambiguität, die durch eine zu niedrige Abtastfrequenz ausgelöst wird. Die Abtastfrequenz wurde gerade so gewählt, dass sie mit der Frequenz des höherfrequenten Signals übereinstimmt. Durch die so gewonnenen Stützstellen kann aber auch ein Signal mit halber Frequenz gelegt werden. Daher muss die Abtastfrequenz doppelt so groß sein, wie die größte vorkommende Frequenz. Formal wird vom Nyquist-Shannon-Abtasttheorem für kontinuierliche Signale, deren Frequenzen zwischen  $f_{\min}$  und  $f_{\max}$  liegen, eine Abtastfrequenz  $f_{\text{abtast}} > 2(f_{\max} - f_{\min})$  gefordert [56]. Wenn das Abtasttheorem nicht beachtet wird, sind Mehrfachdeutungen in der Signalinterpretation und damit auch in der Signalrekonstruktion möglich. In Bild 2.2 ist beispielhaft eine solch mögliche Mehrfachinterpretation zu sehen.

In dem Gebiet der AR werden neben der Bewegung von Objekten auch die Bewegungen einzelner Gliedmaßen des Benutzers gemessen. Welch hat in [114] gezeigt, dass ein Benutzer eines AR Systems seine Arme und seinen Kopf mit einer Frequenzen zwischen 2 Hz und 20 Hz bewegt. Um das gesamte Bewegungsspektrum mit Messungen abzudecken, müssen nach dem Nyquist-Shannon-Abtasttheorem die Abtastfrequenz doppelt so groß sein wie die schnellste zu messende Frequenz. Mit den von Welch bestimmten Frequenz und dem Abtasttheorem ergibt sich eine Mindestfrequenz von 40 Hz. Alle in dieser Arbeit verwendeten Sensoren genügen dieser Anforderung mit Abtastraten  $f_{\text{abtast}} \geq 40$  Hz. Das verwendete kamerabasierte Trackingsystem liefert hingegen nur rund 15 Hz. Dies ist bedingt durch die niedrige Qualität der handelsüblichen WebCam Kamera. Aber schnelle Bewegungen und damit auch hohe Bewegungsfrequenzen sind mit dieser Kamera auch aus einem anderen Grund nicht möglich. Die Bildunschärfe (engl.: *motion blur*) macht ein Tracking bei schnellen Bewegungen der Kamera unmöglich. Dass solche Systeme, trotz dieser Einschränkung, trotzdem gut für AR geeignet sind, zeigen zahlreiche Applikationen wie von Klinker [63] oder Pentenrieder [77].

Ein Signal setzt sich aus einem deterministischen Signal  $s(t)$  und einem statistischen Signal  $n(t)$  zusammen

$$\mathbf{x}(t) = s(t) + \mathbf{n}(t) \quad [106].$$

Die Summe  $\mathbf{x}(t)$  wird dann als stochastisches Signal oder stochastischer Prozess genannt. Der deterministische Teil enthält die eigentliche Information, wie die Bewegung eines Sensors, der stochastische Teil enthält die Störanteile. Unter die Kategorie der Störanteile fällt nicht nur das Rauschen eines Sensors, sondern auch systematischen Fehler der Trackingsysteme. Auch Fehler, die durch unsaubere Registrierung verursacht werden, fallen in diese Kategorie. In Abbildung 4.10 auf

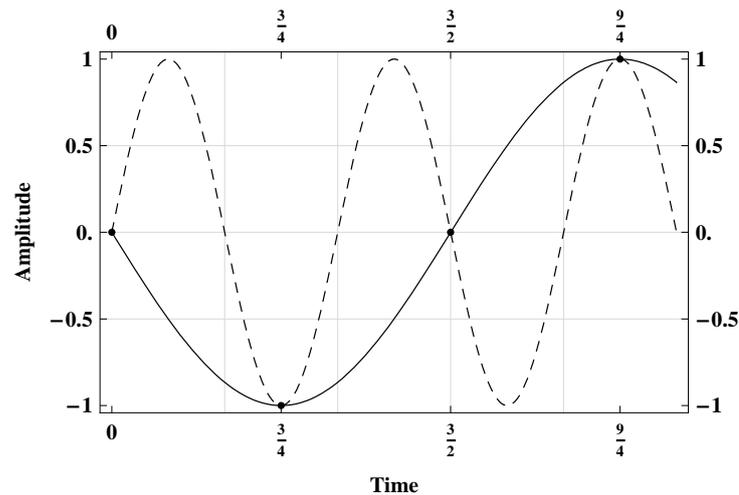


Abbildung 2.2: Sinuswellen als Beispiel für das Nyquist-Shannonsche Abtasttheorem [32]; Die Abtastfrequenz liegt bei 1,33 Hz und die zwei Sinuswellen haben Frequenzen von 1,0 Hz und 0,333 Hz

Seite 64 ist eine beispielhafte Abbildung des deterministischen, des statistischen und des kombinierten Signals zu sehen.

## 2.4 Tracking Sensoren

In dieser Arbeit werden die Prinzipien der Zeitkalibrierung anhand einiger Sensoren demonstriert und nachgewiesen. Diese Sensoren seien im Nachfolgenden kurz vorgestellt.

**Markertracker** Optische Flachmarkertracker werden in dem Gebiet der Augmented Reality schon seit langem erfolgreich eingesetzt. Die Technik ist ausgereift und günstig in der Anschaffung. Die Flachmarker können auf einem handelsüblichen Drucker ausgedruckt werden, und als Kamera reicht eine einfache handelsübliche Kamera. Sechs zu einem Würfel zusammengebauten Flachmarker kann man in Abbildung 2.3(d) sehen, und eine Kamera in Abbildung 2.3(c). Bei höheren Ansprüchen an Genauigkeit, Auflösung und Abtastrate können auch hochwertigere Kameras eingesetzt werden.

Kato und Billinghurst machten mit dem ARToolkit [58] das Prinzip des Markertrackers einer breiten Öffentlichkeit kostenlos zugänglich. Auch heute noch sind Markertracker weitverbreitet. So findet sich diese Technologie in industriellen Anwendungen [77] mit hohen Anforderungen an die Genauigkeit.

Das Funktionsprinzip ist bei allen erhältlichen Markertrackern ähnlich. Es werden geometrische Muster mit hohem Kontrast mit einer optischen Kamera erfasst. Im Falle von ARToolkit sind dies schwarze Quadrate auf weißem Grund. Sind die Maße des Musters bekannt, kann sowohl die Entfernung als auch die Lage der Mar-

ke bezüglich der Kamera bestimmt werden. Damit sind alle sechs Freiheitsgrade zur Angabe einer Pose bestimmt. Für weitere Details bezüglich Funktionsweise und erreichbaren Genauigkeiten sei auf Kato und Billinghurst [58] und auf Pentrieder [78] verwiesen.

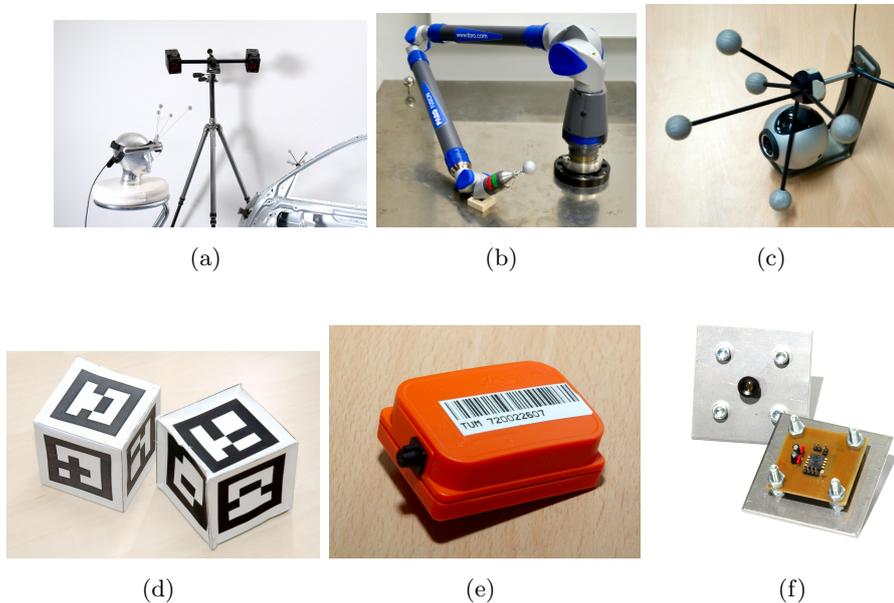


Abbildung 2.3: Übersicht über die verwendeten Sensoren; (a) Optisches auf Infrarotlicht basierendes System SmartTrack der Firma A.R.T. GmbH; (b) FaroArm Fusion Koordinaten Messmaschine der Firma FARO Technologies Inc.; (c) Videokamera (WebCam) mit einem für optisches Infrarot Tracking angebrachten Target; (d) Zwei Papier basierte Marker für optisches Tracking mit einer Kamera ; (e) Gyroskop der Firma Xsens Technologies B.V. ; (f) Sensor für optisches Odometer

**Auf Infrarotlicht basierende optische Trackingsysteme** Auf Infrarotlicht basierende optische Trackingsysteme nutzen aus, dass im Tageslicht nur relative geringe Anteile an Infrarotlicht vorhanden sind und verwenden diese Eigenschaft, um die benutzten Marker besser von der Umgebung segmentieren zu können. In der Abbildung 2.4 ist das Prinzip eines solchen Systems zu sehen. Das Infrarotlicht wird von einer speziellen Lichtquelle emittiert und an Marken mit spezieller Oberflächenbeschaffenheit reflektiert. Da vor die Kamera ein optischer Filter montiert ist, der nur Infrarotlicht passieren lässt, werden die Bilder der Marken in der Kamera mit einem hohen Kontrast gegenüber dem Restbild erfasst. Dadurch lassen sich die Marken schneller und sicherer von der Umgebung trennen. Andere Systeme verwenden keine Beleuchtungseinheit und reflektierende Marken, sondern im Infrarotlicht leuchtende Marken. Man spricht dann von aktivem optischen IR

Tracking im Gegensatz zu einem passivem.

Sind die Marken in den Kamerabildern gefunden worden, kann über Triangulation und ähnliche geometrische Verfahren ihre drei dimensionale Position berechnet werden. Werden mindestens drei solche Marken zu einem Target montiert, wie in Abbildung 2.3(c) zu sehen, kann das System auch die Orientierung messen und somit eine komplette Pose erkennen.

Die Positionsfehler sind meist kleiner als 0.1 mm und die Winkelfehler kleiner als  $0,1^\circ$ . Die Updateraten solcher Systeme liegen im Allgemeinen bei 50 Hz und das Arbeitsvolumen meist kleiner als 10 m x 10 m x 10 m. Der Arbeitsbereich kann aber auch vergrößert werden, wodurch aber der Aufwand bei der Installation erheblich steigt.

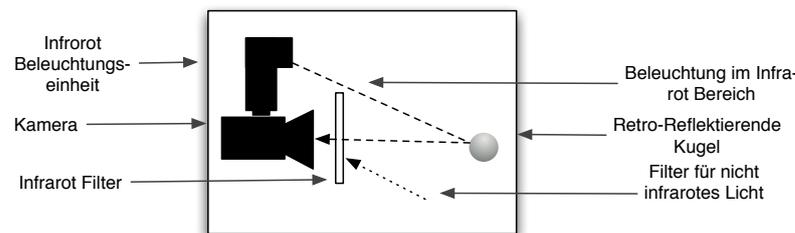


Abbildung 2.4: Prinzip eines optischen auf infrarotem Licht basierendem Trackings.

**Inertiale Sensoren** Inertiale Sensoren sind Sensoren, die die Masseträgheit nutzen, um Änderungen in der Orientierung bzw. Position zu bestimmen. Für Orientierungsänderungen heißen die Sensoren Gyroskope und für Positionsänderungen Beschleunigungssensoren.

Da inertielle Sensoren nur Änderungen eines Systems relativ zur vorherigen Pose messen können, kann man sie in AR Anwendungen nicht ohne Sensoren mit absolutem Messprinzip einsetzen. Aber ausgehend von einer bekannten Position und Orientierung kann die Information der inertialen Sensoren integriert werden und somit die aktuelle Position und Orientierung des Sensors bestimmt werden.

In AR Anwendungen werden inertielle Sensoren hauptsächlich zu Überbrückung kurzer Zeiten benutzt, wenn beispielsweise ein primäres Trackingsystem auf Grund von Verdeckungen nicht verfügbar ist. Aber auch zur Stabilisierung des Trackings bei schnellen Bewegungen werden inertielle Sensoren benutzt [83]. Dazu werden die Daten des primären Trackings mit den Daten der inertialen Sensoren fusioniert. In Abbildung 2.3(e) ist ein Gyroskop der Firma Xsens Technologies B.V. zu sehen, wie es für AR System häufig eingesetzt wird.

**Optisches Odometer** Ein Odometer misst zurückgelegte Strecken. Dies kann entweder durch mechanische Zähler geschehen, die beispielsweise die Anzahl der Um-

drehungen eines Rades messen. Eine andere Methode benutzt optische Kameras, die auf den Untergrund gerichtet sind. Durch die Bewertung der aufgenommenen Bilder kann dann die zurückgelegte Distanz ermittelt werden. In Abbildung 2.3(f) ist ein solcher Sensor zu sehen und in der Abbildung 2.5 ist eine Darstellung eines Kamerabildes dieses Sensors zu sehen.

Dazu wird der *optical flow* [47] berechnet, und mit einer Registrierung, die die Änderungen des Bildes mit Positionsänderungen in Relation bringt, kann die Positionsänderung eines solchen Sensors berechnet werden. Nutzt man mehr als zwei solche Sensoren, kann neben einer Positionsänderung auch die Änderung der Orientierung berechnet werden. Nachdem diese Sensoren nur Änderungen, also relative Informationen, messen können, kann mit solch einem System alleine noch kein Tracking vollzogen werden. Dazu wird wie bei inertialen Sensoren mindestens einmal eine Initialisierung mit einem anderen Sensorsystem benötigt.

Sensoren, wie ein optisches Odometer, sind zwar nicht sehr genau, bestehen aber durch ihre hohe Verfügbarkeit. Sie können die Lücken zwischen genauen Systemen füllen und somit ein unterbrechungsfreies Tracking gewährleisten. Ein mögliches Einsatzszenario ist in [109, 49] beschrieben. Der in dieser Arbeit benutzte odometrische Sensor wurde in Zusammenarbeit mit Manuel Huber im Rahmen des Forschungsprojekts *trackframe* der Bayerischen Forschungstiftung entwickelt.

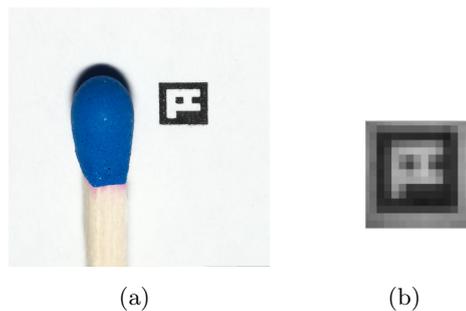


Abbildung 2.5: Darstellung des Maussensors: (a) zeigt einen Marker im Vergleich mit einem Streichholz (mit freundlicher Genehmigung von Manuel Huber); (b) zeigt einen gleich großen Marker aufgenommen mit einem Sensor des Odometers

**FaroArm** Die Firma FARO Technologies Inc. stellt verschiedene mechanische hochpräzise Messarme her. Diese werden häufig in der Industrie als kleine und mobile Messsysteme verwendet. In Abbildung 2.3(b) ist ein Bild eines FaroArm Fusion Geräts zusehen. Ein Arm des Typs Fusion liefert eine garantierte Genauigkeit von unter 0.1 mm. Aufgrund der hohen garantierten Genauigkeit wird dieses System häufig als Referenzsystem benutzt, um andere Trackingsystem bezüglich der Genauigkeit zu evaluieren [88, 65]. Dabei erweist es sich als hilfreich, dass dieses Messsystem Daten mit hoher Wiederholrate liefern kann.

## 2.5 Sensorfusion

Jeder der gerade beschriebenen Sensoren hat Vor- und Nachteile. So liefert ein optisches kamerabasierendes System eine sehr hohe Genauigkeit, aber dafür ist das Arbeitsvolumen mit ein paar Kubikmetern klein. Das optische Odometer hingegen bietet ein beliebiges Arbeitsvolumen, dafür ist es ungenau. Würde man beide Sensoren kombinieren, hätte man ein Sensorsystem mit einem großen Arbeitsvolumen und hoher Genauigkeit. Eine solche Kombination von Sensoren nennt man eine Sensorfusion. Unter Sensorfusion versteht man also das Zusammenführen von Daten von mehreren Sensoren in einen einzigen Datenstrom, der dann mehr Informationen beinhaltet, als die einzelnen Datenströme alleine.

Man kann bei der Sensorfusion verschiedene Grundtypen von einander unterscheiden. So ergibt die einfache Kombination der Daten eines Sensors für die Position und der eines Sensors für die Orientierung eine vollständige Pose. Dieser Vorgang lässt sich zweifellos als Sensorfusion beschreiben. Andere Sensorfusionen kombinieren die Daten gleichen Typs und benötigen dafür aufwendigere mathematische Verfahren, um eine Kombination zu gewährleisten, welche höherwertige Informationen liefern als die zwei einzelnen Sensoren.

Die Klassifizierung in verschiedene Sensorfusionstypen und eine kurze Beschreibung zweier Fusionsalgorithmen liefern die folgenden Abschnitte.

### 2.5.1 Taxonomie Sensorfusion

Die Sensorfusion für Orts- und Lagebestimmung kann nach [24, 16, 110] in drei verschiedene Klassen eingeteilt werden.

**Komplementäre Sensorfusion** Eine Sensorfusion heißt komplementär, wenn sie Sensoren miteinander verbindet, die nicht voneinander abhängen. Die Kombination beider Sensoren ergeben eine vollständigere Messung als beide Messungen für sich. Als Beispiel seien Sensoren gleichen Typs genannt, die aber voneinander unterschiedliche Arbeitsbereiche haben. Ein weiteres Beispiel ist die Kombination eines GPS Sensors, der eine Position mit drei Freiheitsgraden detektiert, und eines Kompasses, welcher die Orientierung misst.

Es kann zwischen einer zeitlich komplementären, räumlich komplementären und funktional komplementären Sensorfusion unterschieden werden. Die oben beschriebenen Beispiele fallen in die Kategorie der räumlich und funktional komplementären Fusion, da die Sensoren verschiedene Arbeitsbereiche haben bzw. unterschiedliche geometrische Eigenschaften messen. Bei einer zeitlich komplementären Fusion können verschiedene Sensoren zwar die gleichen geometrischen Eigenschaften messen, es wird aber zu einem Zeitpunkt nur immer nur einer der Sensoren verwendet. Ein Beispiel hierfür ist der in Abbildung 2.6 dargestellte Prototyp der in [109, 49] vorgestellten Fusionsplattform. Der Wagen ist mit einer Kamera für das Tracken von optischen Flachmarkern und mit drei optischen Odometern ausgestattet. Der Markertracker liefert eine Pose des Wagens, solange sich mindestens

eine Marke im Blickfeld der Kamera befindet. Die optischen Odometer sind in Abschnitt 2.4 beschrieben worden und liefern nach einer Initialisierung kontinuierlich Position und Orientierung der Plattform. Wenn beide Sensoren eine Pose der Plattform liefern, wählt die Sensorfusion die genaueren Daten des Markertrackers und verwirft die Informationen des Odometers. Wenn aber der Markertracker keine Messdaten liefert, weil beispielsweise die Sicht zu einem Marker verdeckt wird, nutzt die zeitlich komplementäre Sensorfusion die Pose des Odometers.

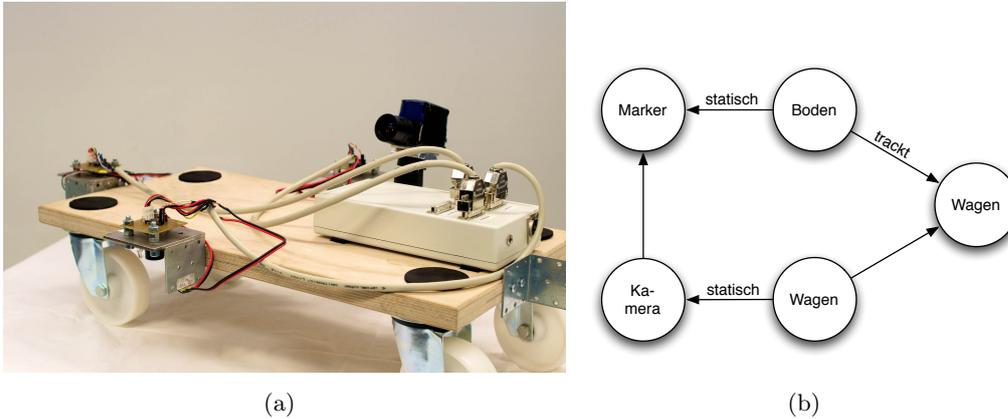


Abbildung 2.6: Komplementäre Fusion: (a) Beispiel einer Anwendung mit komplementärer Fusion; (b) zugehöriger Spatial Relationship Graph

**Kompetitive Sensorfusion** Bei einer kompetitiven Sensorfusion werden Sensoren miteinander fusioniert, die ganz oder teilweise die gleichen physikalischen Effekte messen. Durch die Fusion dieser redundanten Informationen wird die Genauigkeit der Messung erhöht. Die Art der Fusion ist aufwendiger als die komplementäre Sensorfusion. Es bedarf hierfür Sensorfusionsalgorithmen wie in Abschnitt 2.5.2 beschrieben.

Ein Beispielanwendung für die kompetitive Sensorfusion ist in Abbildung 2.7 dargestellt. Diese Kombination einer Kamera und eines Gyroskops ist Teil des in Abbildung 2.6 gezeigten Prototypen der Fusionsplattform. Die Sensorfusion kombiniert in diesem Fall die vollständigen Posen des Markertrackers mit den Orientierungsdaten des Gyroskops. Dadurch wird eine größere Genauigkeit der Messung der Orientierung erreicht.

**Kooperative Sensorfusion** Als kooperative Sensorfusion kann eine Kombination von Sensoren aufgefasst werden, in welcher die Sensoren aufeinander aufbauen. Dies kann die Initialisierung eines zweiten Trackingsystems sein oder auch die Hintereinanderschaltung von Trackingsystemen. In dem letzteren Szenario trackt das System A einen Target B, welches mit einem Tracker C fest verbunden ist. Wenn nun der Tracker C ein Target D trackt und die räumliche Beziehung zwischen

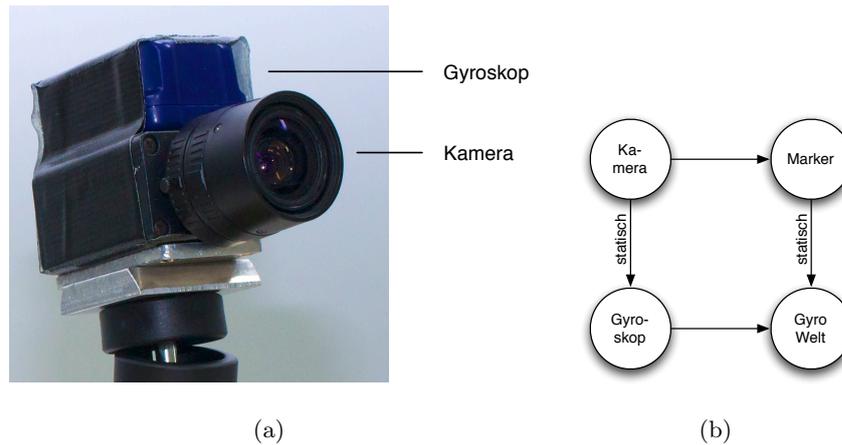


Abbildung 2.7: Kompetitive Fusion: (a) Beispiel einer Anwendung mit kompetitiven Fusion; (b) zugehöriger Spatial Relationship Graph

B und C bekannt ist, dann kann auch unter Ausnutzung der Transitivität die räumliche Beziehung zwischen A und D berechnet werden.

Als Beispiel hierfür kann die in [61] beschriebene Erweiterung der Bolzenschweißanwendung von Echtler et al. [26] gesehen werden. Die Bolzenschweißanwendung war bis 2010 im produktiven Einsatz im Versuchsfahrzeugbau eines namhaften Automobilherstellers. Ziel dieser Anwendung war es, den Werker beim Schweißen von Bolzen zu unterstützen. Gerade das manuelle Suche der Positionen ist ein sehr aufwendiger Prozess, weswegen in [26] eine Anwendung entwickelt wurde, mit dem Ziel diese Aufgabe zu optimieren. Dazu werden aus einer Datenbank die Positionen der zu setzenden Bolzen in die Anwendung geladen. Die Bodengruppe des Fahrzeugs wird dann in einen sogenannten Wender eingespannt und mit Hilfe eines optischen Infrarotlicht basierendem Trackingsystem (A.R.T.) getrackt. Auch die Bolzenpistole wird von diesem Trackingsystem erfasst. Um eine Navigation zu den einzelnen Schweisspositionen zu ermöglichen, wurde an der Bolzenpistole ein Display angebracht, auf welchem virtuelle Geometrien aus der Konstruktions-Datenbank dargestellt werden. Wenn alle Komponenten zu einander registriert sind, ist es möglich dem Werker bei der Navigation von Bolzenposition zu Bolzenposition mit Hilfe von Visualisierungen auf dem Display zu unterstützen. Dafür wird auf dem Display lagerichtig die VR Darstellung der gleichen Bodengruppe, wie der Werker vor sich sieht, dargestellt. Um die nächste Position schneller zu finden, wird eine Linie von der aktuellen Pistolenposition zur der gesuchten Position visualisiert. Befindet sich der Werker in der Nähe der Zielposition und auf der richtigen Seite der Karosse, wird die Visualisierung auf eine Darstellung von Kimme und Korn umgestellt. Dies erlaubt eine schnelle und präzise Zielführung.

Durch den Einsatz eines auf Infrarotlicht basierendem Trackingsystems ist man auf eine Sichtverbindung zwischen Kameras und den Markern angewiesen. Nachdem die Kameras an der Decke der Werkhalle hängen, kann nicht in einer geschlos-

senen Rohkarosse gearbeitet werden. Auch der Kofferraum oder der Moterraum sind von einem an der Decke montiertem Trackingsystem nicht leicht einzusehen. Nachdem die fest installierten Systeme nicht verlegt werden können, wurde der folgende Ansatz verfolgt. Ein weiteres optisches Trackingsystem wird dem Arbeitsbereich hinzugefügt, welches vom statischen an der Decke montieren Trackingsystem getrackt wird. Damit ergibt sich der oben beschriebene transitive Situation. In Abbildung 2.8(b) ist der SRG dieser Erweiterung zu sehen. Das statische System trackt ein Target. Dieses Target ist fest auf einem mobilen System montiert, welches dann die Bolzenpistole im Inneren des Rohkarosse tracken kann. Dies ist in Abbildung 2.8(a) als Skizze gezeigt. Diese Tracking-Methode kann auch indirektes oder transitives Tracking genannt werden [36].

Diese Methode bietet Vorteile in der flexiblen Gestaltung und Aufteilung des Arbeitsvolumens, birgt aber einen erheblichen Mehraufwand für Registrierung in sich. Die beschriebene Erweiterung wurde in enger Zusammenarbeit mit Peter Keitler im Rahmen des Forschungsprojekt *trackframe* der Bayerischen Forschungstiftung entwickelt.

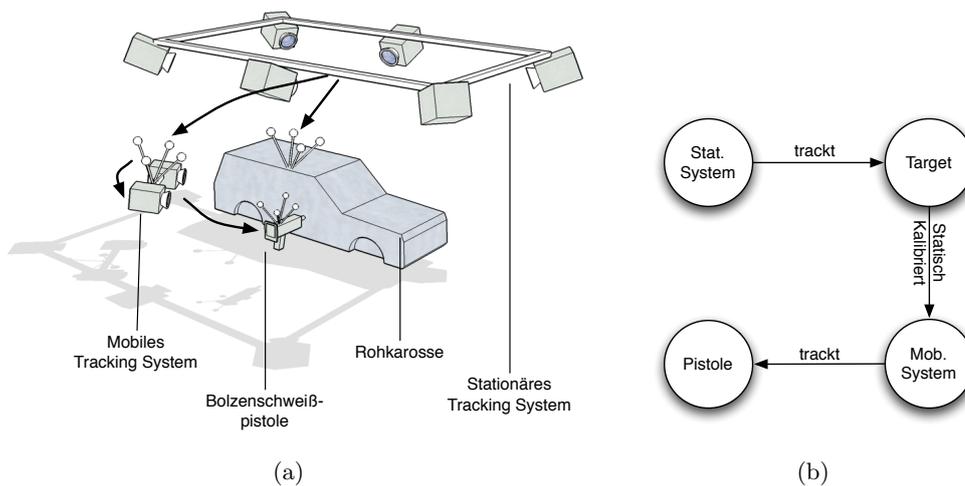


Abbildung 2.8: Kooperative Fusion; (a) Beispiel einer Anwendung mit kooperativen Fusion; (b) zugehöriger Spatial Relationship Graph

Aber auch das zuvor vorgestellte Beispiel der Fusionsplattform nutzt das Konzept der kooperativen Sensorfusion. So wird das optische Odometer mit Hilfe der registrierten Sensordaten des optischen Trackingsystems initialisiert.

## 2.5.2 Mathematische Modelle der Sensorfusion

Im Weiteren werden die zwei in der Augmented Reality Forschungsrichtung meist benutzten mathematischen Modelle der Sensorfusion beschrieben. Zu Anfang wird der Kalman-Filter beschrieben und anschließend die Methode des Partikel-Filters

kurz vorgestellt. Beide beschriebenen Methoden erlauben es, mit Hilfe eines Modells aus Messwerten Zustände eines Systems zu schätzen. Dadurch können diese Filter fehlerhafte Sensormesswerte korrigieren und auf Grund ihrer Struktur auch Sensormesswerte von mehreren Sensoren fusionieren

**Kalman Filter** Im Jahr 1960 hat Rudolf Emil Kálmán in [55] eine rekursive Lösung für das Filtern diskreter linearer Daten beschrieben. Diese Filtertechnik war seitdem Gegenstand zahlreicher Untersuchungen. Im Folgenden werden nur die Grundzüge des Kalman Filters beschrieben. Für tiefere Informationen sei der Leser auf [55, 10, 112] verwiesen.

Ein Kalman-Filter ist ein rekursiver Schätzer. Das bedeutet, dass nur der vorherige Zustand des Systems und ein aktueller Messwert bekannt sein müssen, um den neuen Zustand des Systems schätzen zu können. Beim Tracking besteht dieser Zustand meist aus Position, Orientierung, Geschwindigkeiten und Beschleunigungen eines Objektes oder einer Person.

Der Kalman-Filter arbeitet mit einer Prädiktor-Korrektor-Struktur. Das heißt eine Schätzung wird in die zwei Schritte Prädiktion und Korrektur aufgeteilt. Im Prädiktionsschritt wird der aus dem vorherigen Schätzschritt bekannten und mit einer Störung versehenen Zustand ein neuer Zustand geschätzt. Gleichzeitig wird das Modell der Messungenauigkeit angepasst. Im nächsten Schritt wird der prognostizierte Zustand mit den aktuellen Messwerten verglichen und korrigiert. Das ergibt dann die aktuell gültige Schätzung des Zustands. Die im Korrekturschritt festgestellten Abweichungen werden dann wieder im nächsten Zeitschritt für den Prädiktionsschritt verwendet.

Die rekursive Verfahrensweise und die für jeden Schritt konstant bleibende Rechenzeit sind die wesentlichen Gründe für die weite Verbreitung des Kalman Filters. Das in 1960 von Kálmán entwickelte Verfahren ist jedoch auf lineare Modelle beschränkt und deckt nur zeitdiskrete Reihen ab. Da die meisten Probleme nicht linear beschrieben werden können, wurde eine Erweiterung entwickelt, welche erweiterter Kalman-Filter (engl.: *extended kalman filter*, *EKF*) genannt wird. Der erweiterte Kalman-Filter linearisiert die Modelle lokal, woraufhin das Problem wieder auf den regulären Kalman-Filter reduziert werden kann. Dazu muss das Modell differenzierbar sein. Die Grundzüge des Kalman Filters wie Rekursivität und Diskretheit bleiben dabei durch die Reduktion auf den normalen Kalman-Filter erhalten. Auf Grund der Struktur des Kalman-Filters können auch mehrere Messwerte benutzt werden, um einen neuen Zustand des Systems zu ermitteln. Dies ist vereinfacht gesprochen die eigentliche Sensorfusion.

**Partikel-Filter** Die Partikel-Filter gehören zu der Familie der sequenziellen Monte-Carlo-Methoden und sind in der Lage, ähnlich dem Erweiterten Kalman-Filter, nicht lineare Modelle und nicht gaußverteiltes Mess- und Prozessrauschen zu verarbeiten [87].

Ziel ist es, eine Verteilungsfunktion für den Zustandsraum zu schätzen. Mit Hilfe

dieser Verteilungsfunktion können dann die wahrscheinlichsten Zustände abgeleitet werden. Die Verteilungsfunktion wird nicht direkt berechnet, sondern mit Monte-Carlo Methoden approximiert. Dazu wird eine festgelegte Anzahl an so genannten Partikeln erzeugt, die jeweils einen Zustand repräsentieren, und mit einem Gewicht versehen sind. Das Gewicht steht für die Qualität eines Partikels, also wie gut ein Partikel den momentanen Zustand vorhergesagt hat. Wie im Kalman-Filter wird auch bei dem Partikel-Filter ein Vorhersage- und einen Korrekturschritt durchgeführt. In jedem Schritt werden Partikel mit zu kleinem Gewicht, also die die sich zu weit von dem momentanen Zustand wegbewegt haben, aus dem System entfernt. Details hierzu können in [41, 30, 31] nachgelesen werden.

## 2.6 Tracking Frameworks

Jede AR Anwendung benötigt Tracking und Registrierungen. Aber auch die Probleme der Sensorfusion und die Organisation von Datenströmen muss fast jede AR Anwendung lösen. Daher haben Softwareingenieure Frameworks gebaut, die die häufigsten Funktionalitäten kapseln und somit mehreren Anwendungen zur Verfügung stellen. So muss eine Anwendung keine Kenntnis über die verwendeten Sensoren haben, oder dass zwei Trackingsensoren fusioniert werden. Für die Anwendung ist nur die Verfügbarkeit der Daten von Interesse. Die folgenden Absätze stellen kurz einige Trackingframeworks vor. Darunter ist auch *ubitrack*, welches die Grundlage für die Auswertungen in dieser Arbeit war.

### 2.6.1 OpenTracker/Studierstube

Die OpenTracker wurde 2001 von Reitmayr et al. [85] vorgestellt und stellt eine Tracking Softwareschicht für das Studierstube AR Projekt [92] dar. Es unterstützt verschiedene Sensoren, eine flexible Konfiguration mittels Beschreibung in XML

Die verschiedenen Schritte, die typischerweise in AR Anwendungen für das Tracking notwendig sind, können in OpenTracker als Graph von Objekten modelliert werden. Der Graph beinhaltet Eingangsknoten, welche Daten bereitstellen. Dies können beispielsweise Hardwaredreiber für reale Sensoren sein. Des weiteren gibt es Verarbeitungsknoten, die Berechnungen durchführen können. Sie sind dadurch charakterisiert, dass sie Eingangs- und Ausgangskanten besitzen. Ausgabeknoten hingegen senden die verarbeiteten Trackingdaten schlussendlich an die Anwendung.

### 2.6.2 CampAR

Die von Sielhorst et el. [97] vorgestellte CAMPAR Software ist eine an der Technischen Universität München entwickelte Softwareplattform für Augmented Reality. Der Fokus bei dieser Software ist stärker auf Zuverlässigkeit und Genauigkeit gerichtet, da das Einsatzgebiet der Software primär medizinische Anwendungen sind.

Ähnlich zu anderen Softwareplattformen werden die Datenflüsse vorab spezifiziert, womit dann die benötigten Komponenten gestartet werden.

### 2.6.3 DWARF

Das „Distributed Wearable Augmented Reality Framework“ [8], kurz *DWARF*, ist eine Framework für die Entwicklung von Augmented Reality Anwendungen [67]. Die Plattform hatte zum Ziel mobile Systeme dynamisch miteinander zu verbinden. Da die einzelnen Subsysteme austauschbar sind, ergibt sich ein sehr dynamisches Gesamtsystem. Daher musste die Software in der Lage sein, Komponenten dynamisch miteinander zu verbinden ohne auf einen starren vorher festgelegten Datenfluss angewiesen zu sein.

Aus dem DWARF Framework ist später dann die *Ubitrack* Trackingsoftware entstanden.

### 2.6.4 ubitrack

Die Ubitrack Software ist wurde als Nachfolger der DWARF Software entwickelt, um die Schwächen in der Implementierung zu beheben. Die dynamischen Konzepte, wie sie schon in DWARF vorhanden waren, wurden in Ubitrack noch weiter ausgebaut. Ubitrack wurde stark in Richtung von Ubiquitous Tracking (UT) entwickelt. Unter Ubiquitous Tracking versteht man Tracking, welches auf einer dynamischen Fusion von weit verteilten, unterschiedlichen heterogenen Sensoren basiert [75]. Daher bietet Ubitrack Funktionalitäten wie automatische Deduktion von geometrischen Zusammenhängen und Sensorfusion von verschiedenen Sensoren. Im Gegensatz zu DWARF und OpenTracker werden in Ubitrack eine XML Beschreibung für die Spatial Relationship Graphen und auch für die Datenflußkonfigurationen genutzt.

Die System-Architektur von ubitrack erlaubt eine zentral koordinierte Peer-To-Peer Architektur für ubiquitäres Tracking [50]. Dabei berechnet ein Server automatisch optimale Datenflußkonfigurationen, so dass die einzelnen Knoten die Trackingdaten direkt und mit geringer Latenz austauschen. Die Deduktion von Datenflüssen wird mit Hilfe von Patterns berechnet [81]. Wobei Patterns auf einen bestehenden Spatial Relationship Graphen, kurz *SRG*, gematcht werden und jedes Pattern eine Rechenoperation repräsentiert. Mit Hilfe dieser Techniken, können aus räumlichen Abhängigkeiten Datenflüsse mit Berechnungen abgeleitet werden. Neben der Deduktion von Datenflüssen wurden in Ubitrack auch Fehlerbetrachtungen und -propagierungen implementiert. Somit könnte das System auch zur Laufzeit entscheiden, ob eine Sensorfusion das Gesamtergebnis verbessern kann.

Das Ubitrack Framework wird in der *Fachgruppe für Augmented Reality* von Prof. Klinker an der Technische Universität München im Rahmen der Forschungsrichtung Ubiquitäre Augmented Reality weiter vorangetrieben.

Besonders sei in diesem Zusammenhang auch die Software *trackman* von Peter Keitler erwähnt, die eine Konfiguration der SRGs und der Datenflüsse mit Hilfe

eines graphisches Nutzerinterface erlaubt [60, 62]. Ein Abbildung dieses System ist in Abbildung 2.9 dargestellt.

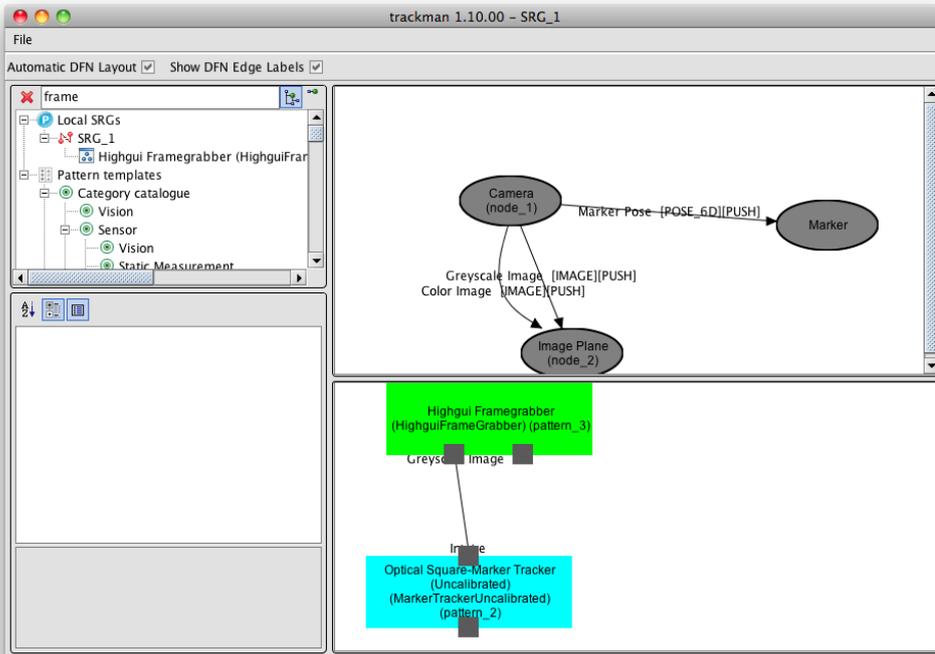


Abbildung 2.9: Das Konfigurationstool *Trackman* [62] für ubitrack erlaubt die Erstellung und Konfiguration von Spatial Relationship Graphen und Datenflüssen

### 2.6.5 LocationStack

Der Location Stack Ansatz von Hightower [42] war der erste Ansatz Richtung einer einheitlichen Abstraktion von Lokalisierung mit mehreren Sensoren.

Das verwendete Modell ist dem Modell mit sieben Schichten vom ISO/OSI Modell für Computer Netzwerke angelehnt. Die einzelnen Schichten sind wie folgt aufgeteilt. Die Basis bilden die Sensoren. Dazu gehören sowohl die Hardware der Sensoren als auch die Softwaretreiber, um die Messdaten auszulesen. Die zweite Schicht ist die Schicht der Messwerte. In dieser Schicht werden Algorithmen beschrieben, die die rohen Messdaten in einheitliche Messwerte umwandeln. Dabei werden auch Messunsicherheiten beschrieben. Die dritte Schicht beinhaltet die Fusion der Daten. In diesem Falle Methoden, um mehrere kontinuierliche Messdatenströme in eine probabilistische mit Zeitstempeln versehene Repräsentation für Ort und Orientierung zu bringen. Die weiteren Schichten beziehen sich auf die Interaktion der so lokalisierten Objekte und deren Einordnung in einen räumlichen Kontext.

Das Projekt Universal Location Framework (ULF) [34] ist eine Umsetzung der

Konzepte des Location Stacks. Dabei werden GPS Sensoren, WiFi Netzwerkkarten und Annäherungssensoren verwendet und die Messwerte anschließend mit einem probabilistischen Ansatz fusioniert [31].

## 2.7 Latenzen von Trackingsystem

In Augmented Reality und auch allgemeiner in Virtual Reality Anwendungen treten verschiedene Arten von Latenzen auf. Für die Akzeptanz eines Systems ist primär die Gesamtlatenz (engl.: *end-to-end latency*) von Bedeutung. Diese gibt die Zeitspanne zwischen der Bewegung eines Objekts in der Realität und deren Visualisierung in der VR/AR Anwendung. Ausgelöst durch die Gesamtlatenz eines Systems reagiert das System mit einer zeitverzögerten Reaktion auf Bewegungen des Nutzers. Eine geringe Gesamtlatenz lässt sich nicht verhindern, aber eine zu große Gesamtlatenz wirkt auf den Nutzer störend. Es kann den Nutzer auch in der Bedienung eines Systems einschränken, wenn die Reaktion des Systems zu spät auf seine Aktion folgt. Die Gesamtlatenz ist nicht nur bei einer visuellen sondern auch bei einer haptischen Rückmeldung von Bedeutung und sollte eine obere Grenze nicht überschreiten [53].

Die Gesamtlatenz ist für die Genauigkeit von Sensorfusion nur indirekt von Bedeutung. Maßgeblich für eine korrekte Sensorfusion zweier Sensoren ist Differenz der Latenzen der beiden Sensoren. Die Latenz eines Sensors ist die Zeit zwischen der Bewegung in der Realität und deren Messung durch die Sensoren. Nachdem Sensoren unterschiedlich angebunden werden können, wird in dieser Arbeit die Latenz eines Sensors durch die Zeitspanne zwischen der eigentlichen Bewegung und dem Eingang in dem Trackingsystem definiert. Für die Sensorfusion ist es entscheidend, dass Daten des gleichen physikalischen Ereignisses fusioniert werden. Mit dem Wissen über die Differenz der Latenzen, also die relative Latenz, ist eine Korrektur der Daten möglich und somit eine korrekte Fusion.

## 2.8 Zusammenfassung

Dieses Kapitel hat die technischen Hintergründe für die Zeitkalibrierung eingeführt. Neben den grundsätzlichen Fragen des Trackings wurde auch auf die mathematische Repräsentation von räumlichen Trackingdaten eingegangen und wie Beziehungen zwischen den Sensoren und anderen beteiligten Objekten, die mit Hilfe eines Spatial Relationship Graphen repräsentiert werden können. Weiterhin wurden sowohl für Positionen als auch für Orientierungen Fehlernormen vorgestellt, die für die Beurteilung der Trackingdaten und für die eigentliche Kalibrierung verwendet werden können. Neben Interpretation von Trackingdaten als Signale wurden auch einige wichtige Vertreter von Trackingsensoren präsentiert. Gegen Ende wurde eine Einteilung der Sensorfusion und zwei Algorithmen für die Sensorfusion vorgestellt.



## 3 Prinzipien der Zeitkalibrierung

Dieses Kapitel vermittelt dem Leser einen Überblick über die einzelnen Schritte und Methoden der Zeitkalibrierung. Dabei werden die Techniken nicht im Detail beschrieben, sondern der Fokus auf das Zusammenwirken der einzelnen Komponenten gesetzt. Somit bekommt der Leser einen Überblick über die in dieser Arbeit vorgestellten Methoden und Prozesse zur Kalibrierung der relativen Latenz.

Zwei gleiche Sensoren sind nicht gleich. Zumindest aus der Sicht eines Tracking-frameworks oder aus der Sicht eines Fusionsalgorithmus. Denn jeder der beiden Sensoren liefert seine Daten zu einem anderen Zeitpunkt im System ab. Das mögen nur Millisekunden sein, aber bei einer Fusion dieser beiden Sensoren ergeben sich bei einem dynamischen System Fehler, die es zu vermeiden gilt. Gravierender ist die Situation bei Einsatz verschiedener Sensoren. Hier wird eine korrekte zeitliche Kalibrierung unabdingbar für ein genaues Messergebnis.

### 3.1 Verwandte Arbeiten

Da die Kalibrierung der relativen Latenz für dynamische Systeme mit Sensorfusion einen entscheidenden Einfluss auf die zu erwartenden Genauigkeiten hat, wird dieses Thema von einer Vielzahl von Forschern aufgegriffen. Dabei wird jedoch meist eine ganz spezielle Lösung für genau ein Problem gesucht und entwickelt. Die folgenden Abschnitte zeigen Arbeiten, welche detailliert auf die Zeitkalibrierung eingehen.

#### Visuelle Bestimmung der relativen Latenz

Jacobs u. a. greifen in [51] das Problem der unterschiedlichen Latenz-Zeiten in einem Augmented Reality System auf. Dazu geben sie einen ausführlichen Überblick über die in einem AR System vorkommenden Latenzen und geben Methoden an, wie diese vermessen und korrigiert werden können. Dabei betrachten die Autoren alle in einem AR Systeme vorkommenden Komponenten, wie Sensoren, Algorithmen und auch die Visualisierung.

Dazu wird zu Beginn eine formelle Beschreibung der unterschiedlichen in einem AR System auftretenden Latenzen vorgenommen und auch eine Darstellung dieser Latenzen eingeführt (siehe Abbildung 3.1). Neben der in dieser Arbeit betrachteten relativen Latenzen, untersuchen Jacobs u. a. auch die Gesamtlatenz, also die Zeit von einem realen physikalischen Ereignis bis zur Darstellung eines augmentierten Videobilds.

Die Gesamtlatenz (engl.: *end-to-end latency*) wird in dem Artikel in einzelne Teillatenzen aufgeteilt. So wird mit *off-host delay* die Zeitdauer zwischen dem Zeitpunkt des realen physikalischen Ereignisses und dem Zeitpunkt des Ankommens des Messergebnisses im Rechensystem bezeichnet. Daneben sind auch die Verzögerungen, welche durch Rendering und Darstellung erzeugt werden, dargestellt. Der Artikel benennt auch die wichtigsten Zeitpunkte in dem gesamten Vorgang. Zum Beispiel der Zeitpunkt, zu dem die Messpakete dem Tracking-Framework das erste mal zugänglich sind. Zu diesem Zeitpunkt werden die Messpakete mit einem Zeitstempel versehen. Diese Zeitstempel erlauben es dann die Messpakete im Tracking-Framework so zu verarbeiten, dass keine weiteren relative Latenzen mehr entstehen. Die verschiedenen Arten von Verzögerungen und die einzelnen Zeitpunkte sind in der schematischen Abbildung 3.1 zu sehen. In der Abbildung ist neben der Gesamtlatenz auch die relative Latenz zwischen zwei Sensoren dargestellt, welche der relativen Latenz entspricht, wie sie in dieser Arbeit behandelt wird.

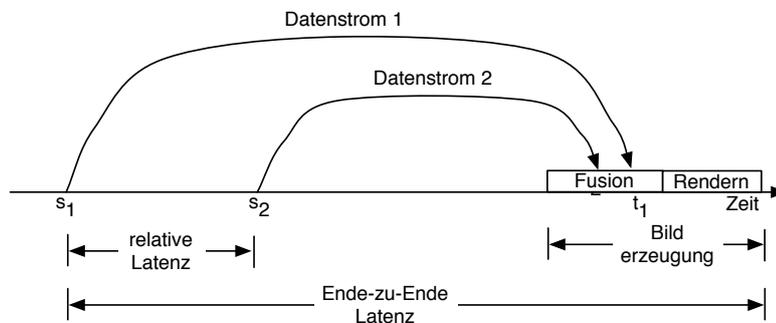


Abbildung 3.1: Darstellung der, in einem AR System auftretenden, Latenzzeiten (nach Jacobs [51]).

Das Wissen über die relativen Latenzen soll natürlich zu einer besseren visuellen Darstellung führen, die geringere Registrierungsfehler aufweist. Dazu geben die Autoren zwei mögliche Korrekturverfahren an. So kann ein Datenstrom kurz aufgehalten werden oder der andere Datenstrom mit Hilfe Prädiktion in die Zukunft geschoben werden. Der Prädiktion ist im Allgemeinen Vorrang zu gewähren, da sie die Gesamtlatenz des Systems nicht negativ beeinflusst. Diese ist jedoch nicht immer möglich, z. B. in einem System das eine Kamera und ein 6-DoF Tracker benutzt. Die Bilder der Kamera sollen in diesem Szenario augmentiert werden unter Zuhilfenahme der Daten des 6-DoF Trackers. Sei weiterhin angenommen die Kamera hat eine größere *off-host* Verzögerung als der benutzte Tracker. Das Kamerabild kann aber nicht einem Prädiktionsalgorithmus unterzogen werden und somit bleibt nur die Verzögerung des Trackersignals.

In der weiteren Arbeit beschreiben die Autoren eine Methode, um die relative Latenz zwischen der Kamera und einem Tracker zu bestimmen. Dabei nutzen sie

ein AR System und die Prädiktion, um die negativen Auswirkungen der relativen Latenz zu kompensieren. Der Prädiktionszeitraum kann durch den Nutzer frei variiert werden, womit auch die relative Latenz variiert wird. Der durch die variierende relative Latenz ausgelöste Fehler kann vom Nutzer in der Visualisierung beobachtet werden. Wenn in der Visualisierung ein minimaler Fehler beobachtet wird, ist der korrekte Wert für die Prädiktion gefunden worden und somit auch der Wert der relativen Latenz.

Weiterhin geben die Autoren ein Verfahren an, um die Gesamtlatenz des Systems zu bestimmen. Auch wenn durch ein Wissen der Gesamtlatenz ein System nicht verbessert werden kann, so hilft ein Messen der Gesamtlatenz dennoch ein System zu beurteilen. Um die Gesamtlatenz eines kamerabasierten Systems zu bestimmen, können Methoden von Jacobs in [51] und Sielhost [98] verwendet werden (vgl. Seite 37).

### **Synchronisation in Hardware**

Das einfachste aber auch das genaueste Verfahren ist, beim Design von Tracking-Systemen schon auf deren relativen Latenzen zu achten und somit auf eine perfekte Synchronisierung. Bei Kamerasystemen ist dies gängige Praxis. Die Firma A.R.T. GmbH aus Weilheim stellt kamerabasierte auf Infrarotlicht operierende Tracking-Systeme her. Diese sind mit einer Synchronisierungsbox ausgestattet, welche über ein separates Kabel die Synchronisierungsinformationen zu den beteiligten Kameras schickt. Auch andere optische Mehrkamerasysteme nutzen eine solche Synchronisierung in Hardware. Als weiteres Beispiel sei hier das System OptiTrack der Firma NaturalPoint Inc. zu erwähnen.

Diese Systeme wurden aber schon vom Hersteller als Mehrkamerasystem konzipiert, und somit wurde auch auf eine korrekte Synchronisierung geachtet. Schwieriger wird es bei Systemen mit Sensoren, die auf unterschiedlichen physikalischen Messprinzipien beruhen. Hier muss der Hersteller entweder die Sensoren in Hardware synchronisieren oder darauf achten, dass sich die im Werk bestimmte relative Latenz nicht mehr ändert. Ein Beispiel für kombinierte Sensoren sind inertielle Sensoren. Diese bestehen oft aus einem Beschleunigungssensor, einem Gyroskop und einem Kompass (vgl. Abschnitt 2.4).

Der in der Ubiquitous Augmented Reality übliche Fall ist die Kombination von verschiedenen Sensoren, die zudem noch von unterschiedlichen Herstellern produziert wurden. Doch auch in dieser Situation, setzen Forscher auf die Synchronisierung durch Hardware. So haben Bleser und Stricker in [11] die Daten einer Kamera und einer Trägheitsmesseinheit fusioniert und beide Systeme mit Hardware synchronisiert. Doch dieses Verfahren ist aufwendig zu installieren, vor allem wenn die Hardware nicht dafür vorbereitet ist. Eine dynamische Fusion, wie es in der Ubiquitous AR gefordert, ist mit diesen Verfahren nicht möglich.

### **Kalibrierung stark schwankender Latenzen**

Aron u. a. haben in [1] einen Zeitkalibrieralgorithmus für eine komplementäre Sensorfusion entwickelt. Ziel war es, ein optisches Kamera basiertes Tracking mit einem Gyroskop zu fusionieren. Dabei werden die Gyroskopdaten nur verwendet, wenn das optische Tracking durch schnelle Rotationen keine Messdaten mehr liefert. Das von den Autoren verwendete Gyroskop wies starke Schwankungen in der Latenz auf, so dass eine a priori Kalibrierung des zeitlichen Abstands zwischen Kamera und Gyroskop nicht möglich war.

Die Autoren haben ein Verfahren entwickelt, dass keiner zeitlichen Kalibrierung vorab bedarf, sondern immer genau dann vorgenommen wird, wenn es nötig ist. Dabei wird angenommen, dass die Gyroskopdaten vor den Kameradaten im System ankommen. Die Rotationsdaten des Gyroskops sowie die Posen und die Referenzpunkte des optischen Trackings werden für eine gewisse Zeitspanne gespeichert. Das Verfahren geht von der Annahme aus, dass zu einem Zeitpunkt  $i$  eine großen Rotation aufgetreten ist, welche ein optisches Tracking verhindert. Sei nun  $t_i$  der Zeitpunkt zu dem das Tracking versagt und  $s_{i-1}$  und  $s_{i-2}$  die zwei Referenzpunktmenge des bildbasierten Trackings bevor das Tracking versagt hat. Nun wird versucht aus der Menge der gespeicherten Rotationsdaten des Gyroskops diejenige Rotation zu finden, die Referenzpunkte  $s_{i-1}$  und  $s_{i-2}$  mit den geringsten Fehlern aufeinander abbildet. Mit dem Zeitstempel, der auf diese Weise gefundenen Rotationsmessung, und den Zeitstempeln der gespeicherten Referenzpunkte lassen sich die zwei Zeitstempelmenge in Relation bringen. Somit wird eine Kalibrierung der relativen Latenz durchgeführt. Anschließend können dann die korrekten Rotationsmessungen ausgewählt werden, um die nicht vorhandene Messung des optischen Trackings auszugleichen.

Dieses Verfahren kann zumindest in der vorgestellten Art nur mit schnellen bzw. großen Rotationen arbeiten. Es könnte jedoch durch Nutzung eines positiongebenden Sensors auch auf translatorische Bewegungen erweitert werden. In der beschriebenen Art sind aber nur Rotationen berücksichtigt, und dies hat zur Folge, dass bei einer gleichzeitigen Rotation und Translation große Ungenauigkeiten zu erwarten sind. In welchem Maße diese Ungenauigkeiten auftreten, wird von den Autoren offen gelassen.

### **Kombinierte Raum-Zeit-Kalibrierung**

Eine weitere Methode für die Bestimmung des temporalen Offsets zweier Sensoren kann mit Hilfe von geometrischen Methoden erreicht werden. Bei dieser Methode werden die Daten beider Sensoren aufgezeichnet und eine räumliche Registrierung durchgeführt. Dabei wird ein Residuum entstehen. Verschiebt man nun einen Datensatz in der Zeit, kann die räumliche Registrierung erneut durchgeführt werden. Da der zu erwartende Zeitabstand in der Regel nicht mit den Wiederholraten der Sensoren übereinstimmt, ist eine Interpolation der Daten erforderlich. Dazu werden Positionsdaten linear und Orientierungsdaten, welche in der Quater-

nionendarstellung vorzuliegen haben, mit Hilfe des Slerp Verfahrens interpoliert. Ähnlich den zuvor vorgestellten Verfahren, lässt sich so ein Minimum des Residuums finden. Somit liefert dieses Verfahren neben einer Schätzung der räumlichen Beziehung zwischen den verwendeten Sensoren, auch eine Schätzung des temporalen Abstands der Sensoren. Angewendet wurde dieses Verfahren von Lieberknecht u. a. [65].

Auch Schwald hat in [93] ein Verfahren beschrieben, mit welchem sich Sensoren zu einander zeitlich kalibrieren lassen. Das Verfahren wird wie bei Lieberknecht während der Registrierung durchgeführt, jedoch wird hier auf den Einsatz von Interpolation verzichtet. Auch bei diesem Verfahren, werden die Messreihen gegeneinander verschoben und dabei über die Zeitstempel nach Korrespondenzen gesucht. Mit den Messwerten, welche zu der bestimmten Menge von Zeitstempel gehören, wird dann erneut eine örtliche Registrierung durchgeführt. Der Offset, welcher dabei das kleinste Residuum während Registrierung liefert, wird dann als relative Latenz zwischen diesen Sensoren gewertet.

Auf Grund der gleichzeitigen Ausführung von räumlicher Registrierung und zeitlicher Kalibrierung ist das Verfahren sehr rechenintensiv und kann nur während der Vorbereitungsphase einer Anwendung genutzt werden. Somit erscheint dieses Verfahren für ubiquitäre Einsatzszenarien unbrauchbar, denn diese Szenarien benötigen eine flexible und dynamische Zusammenschaltung von Sensoren. Des weiteren können Sensoren schwankende temporale Eigenschaften besitzen. Aron u. a. dokumentieren in [1] solche zeitlichen Schwankungen bei dem Gyroskopsensor Xsens MT-9B. Solch starke Schwankungen wie bei diesem Sensor sind sicher die Ausnahme, aber selbst kleinere Schwankungen können negative Auswirkungen haben.

Das von Lieberknecht u. a. [65] beschriebene Verfahren gestattet keine Kalibrierung des zeitlichen Abstands während der Laufzeit der Anwendung. Es ist zwar durchaus möglich, das Verfahren während der Laufzeit zu nutzen und die simultane Registrierung und zeitliche Kalibrierung durchzuführen, aber dadurch steigt die Möglichkeit für pathologische Fälle. Hält der Nutzer zum Beispiel die beiden Sensoren still in seiner Hand, dann wird die räumliche Registrierung verzerrt und auch die räumliche Registrierung wird dann vom korrekten Wert divergieren. Daher ist bei der gleichzeitigen Bestimmung von sieben Freiheitsgraden ein hoher Anspruch an die Daten zu stellen. Somit wird der Nutzungsbereich dieses Verfahrens eingeschränkt.

Aber es existieren einige Szenarien, in welchen sich dieses Verfahren gut eignet. Wenn beispielsweise auf Grund der Konstruktionsweise eines Systems von einer konstanten relativen Latenz und räumlichen Registrierung ausgegangen werden kann, hat die kombinierte zeitliche Kalibrierung und räumliche Registrierung und keinen Nachteil. Als Beispiel ist die Anwendung, die in [65] beschrieben wird, anzuführen. Es gibt auch Hinweise darauf, dass die relative Latenz mit diesem Verfahren mit einer größeren Sicherheit als mit der Korrelation als Vergleichsmethode bestimmt werden kann (vgl. Abschnitt 7.3.2).

### **Kalibrierung Video basierter Sensoren**

Prager u. a. haben 1998 in [79] ein System zur zeitlichen und räumlichen Registrierung eines bildgebenden Ultraschall Sensors und eines 6DoF Pose Sensors vorgestellt. Diese Kombination von Sensoren ist insoweit besonders, weil einer der verwendeten Sensoren keine Position oder Orientierung sondern ein Videobild liefert. Daher können in diesem Falle die bisher vorgestellten Methoden nicht angewendet werden, da diese alle entweder Positionen oder Orientierungen voraussetzen. Das Verfahren soll aber trotzdem kurz präsentiert werden, da es durchaus auch Potential für den Einsatz für zeitliche Kalibrierung im Gebiet der Augmented Reality bietet.

Die Grundidee für diese Kalibrieremethode ist es, den Benutzer eine ruckartige Bewegung durchführen zu lassen. Diese plötzliche Bewegung ist in den Messdaten des räumlichen Sensors relativ leicht zu ermitteln. Um den Zeitpunkt der Bewegung auch im Videobild zu finden, vergleicht diese Methode jedes Bild mit dem Bild davor und berechnet hieraus ein Differenzmaß, welches dem optischen Fluss [47] ähnlich ist. Die Autoren geben für ihr Verfahren eine Genauigkeit von  $\pm(T + t)/2$  an, wobei  $T$  und  $t$  die Aktualisierungsabstände der Sensoren sind. Um die Genauigkeit zu erhöhen wird empfohlen, das Verfahren öfters durchzuführen.

Das Verfahren eignet sich sehr gut, um eine zeitliche Kalibrierung mit einem Videobild und einem räumlichen Sensor durchzuführen. Die Genauigkeit könnte jedoch durch Einsatz von Interpolation erhöht werden. Dies müsste aber in einem weiteren Schritt näher untersucht werden. Nachteilig an dieser Lösung ist jedoch, die spezielle ruckartige Bewegung, die der Benutzer durchführen muss. Dadurch ist diese Methode nicht für den Einsatz während des laufenden Betriebes einer AR Applikation geeignet.

### **Kalibrierung mit kanonischer Korrelation**

Die kanonische Korrelation, die von Johansson in seiner Diplomarbeit [54] umfangreich beschrieben wurde, kann benutzt werden, um die relative Latenz zwischen zwei Sensoren zu bestimmen. Das Bemerkenswerte an diesem Verfahren ist die Invarianz der kanonischen Korrelation gegenüber affinen Transformationen. Nachdem Koordinatentransformationen lineare Abbildungen sind, welche eine Untergruppe der affinen Abbildungen sind, ist die kanonische Korrelation auch invariant gegenüber Koordinatentransformationen. Es ist also unerheblich wie genau die räumliche Registrierung ist oder ob überhaupt eine Registrierung durchgeführt wurde.

Diese Unabhängigkeit wird durch die Art der Dimensionsreduktion erreicht, in diesem Falle eine Projektion. Die zwei Projektionsvektoren werden so gewählt, dass die Korrelation zwischen den beiden mit diesen Vektoren projizierten Signalen maximal wird. Dieses Vorgehen liefert nachgewiesen [54] die Invarianz gegenüber affinen Abbildungen. Nach der Bestimmung der optimalen Projektionsvektoren wird ein zweites Optimierungsproblem gelöst. Ähnlich zu anderen hier vorgestell-

ten Methoden wird dabei ein, nun eindimensionales, Signal in der Zeit verschoben und dabei kontinuierlich mit dem anderen Signal verglichen. Der Zeitpunkt mit maximaler Übereinstimmung wird dann als zu bestimmender Zeitversatz, in diesem Falle *time delay*, gewertet. Neben dieser prinzipiellen Vorgehensweise geht Johansson in seiner Arbeit [54] noch auf verschiedenen Filtertechniken ein, die er mit Hilfe von Simulationen evaluiert hat.

Auf das hier nur grob vorgestellte Vorgehen von Johansson wird in Kapitel 5.3.1 näher eingegangen. Vor allem wird dabei der Teil der Dimensionsreduktion näher betrachtet.

### **Gesamt Latenz**

Geringe Gesamtlatenzen von AR Systemen sind wichtig für die Akzeptanz der Nutzer. Für die Genauigkeiten der Sensorfusion ist die Gesamtlatenz nur von indirekter Bedeutung (vgl. Abschnitt 2.7). Dennoch sollen einige Arbeiten, die eine Bestimmung der Gesamtlatenzen beschreiben, kurz vorgestellt werden.

In [98] bestimmen Sielhorst u. a. die Gesamtlatenz eines Video-AR-Systems. Damit definieren die Autoren die Zeitspanne zwischen dem Zeitpunkt des realen physikalischen Ereignisses und dem Zeitpunkt, zu dem das Bild im Rechnersystem ankommt. Das Verfahren ähnelt dem Verfahren von Jacobs u. a. [51], in welchem auch eine Methode vorgestellt wurde, welche es erlaubt die Gesamtlatenz eines visuellen Systems zu vermessen. Sielhorst definiert die Gesamtlatenz eines AR Systems als den zeitlichen Abstand einer Visualisierung eines augmentierten Bildes und der Realität. Um diese Zeitdifferenz zu bestimmen, wird in ein computergeneriertes Bild ein Zeitstempel enkodiert und auf einem Monitor dargestellt. Mit der für das AR-System benutzten Videokamera wird dieser Monitor erfasst und in den Bildern der Zeitstempel dekodiert. Der Unterschied zwischen der im Bild kodierten Zeit und der aktuellen Zeit ist dann die gesuchte Gesamtlatenz. Dies von Sielhorst u. a. vorgestellte Methode erlaubt es nur die Latenz von video basierten Subsystemen zu messen, andere Sensoren können mit diesem Verfahren nicht vermessen werden.

Es existieren noch weitere Methoden für die Bestimmung der Gesamtlatenz von AR/VR Systemen. So beschreiben He u. a. die Bestimmung der Latenz zwischen einem Sensor und der Visualisierung mit Hilfe einer zusätzlichen Videokamera [40]. Die Kamera zeichnet sowohl die Bewegung des Sensors als auch die generierte Visualisierung auf. In einem eigenen Schritt bewegt der Nutzer den Sensor bei moderater Geschwindigkeit hin und her. In dem aufgezeichneten Videobild werden anschließend die Bewegungen des Sensors mit den Bewegungen der Visualisierung verglichen und somit die Zeitdifferenz bestimmt. Diese Zeitdifferenz entspricht genau der Gesamtlatenz des Systems, da alle beteiligten Komponenten von Anfang bis zum Ende berücksichtigt werden. Auch die Komplexität der Visualisierung und damit eine längere Rechenzeit werden mit dieser Methode berücksichtigt. Ein Vorteil dieser Lösung ist, dass die Gesamtlatenz des Systems nicht durch die Messung beeinflusst wird, da mit einem zweiten unabhängigen System gearbeitet wird.

Dieses Verfahren bedarf aber eines hohen Arbeitsaufwand, da der Vergleich der Bewegung des realen Sensors und die Visualisierung im Video manuell durchgeführt werden muss.

Auf ein ähnliches Prinzip wie He setzen Miller u. a. in dem System „Lateny Meter“ [71]. Hier werden zwei lineare CCD Sensoren benutzt, wobei ein Sensor auf den Nutzer gerichtet ist und der andere auf die Visualisierung. Die Verwendung dieser CCD Sensoren erlaubt eine einfache Synchronisierung der beiden Sensoren. Für die Bestimmung der Latenz muss sich der Benutzer wie bei der Methode von He hin und her bewegen. Diese Bewegungen haben Änderungen in den Messungen beider Sensoren zur Folge, welche dann eine automatisierte Bestimmung der Gesamtlatenz erlauben. Auch diese Methode verändert durch die Messung nicht die Gesamtlatenz des Systems, aber im Gegensatz zur Methode von He ist die Bestimmung der Gesamtlatenz automatisiert.

### 3.2 Vorschlag für eine zeitliche Kalibrierung

Wie in den vorigen Absätzen zu sehen ist, ist das Problem der zeitlichen Kalibrierung nicht neu, und es wurde schon von mehreren Forschern und Autoren adressiert. Grundlegendes Prinzip der meisten Methoden ist das zeitliche Verschieben eines Datenstroms gegen den anderen. Dabei wird fortlaufend überprüft, wann beide Datenströme am besten zu einander passen. Die Überprüfung, wann die zwei Datenströme zeitlich optimal zu einander passen, wird bei den verschiedenen Methoden jedoch unterschiedlich gelöst. Ein Prinzip beruht auf gleichzeitiger räumlicher und zeitlicher Kalibrierung zweier Sensoren. Angenommen zwei fest miteinander verbundene Positionssensoren sollen zueinander registriert werden. Dazu wird ein Datensatz mit dem Paar von Sensoren aufgenommen und beispielsweise mit dem Algorithmus zur Bestimmung der absoluten Orientierung von Horn [46] eine Abbildung zwischen den beiden Punktwolken berechnet. Dabei wird ein Residuum entstehen, welches unter anderem auch von der Größe der relativen Latenz abhängt. Wird nun eine dieser aufgezeichneten Punktwolken zeitlich verschoben, so wird sich auch das Residuum ändern. Bei einer Verschiebung, welche genau den relativen Latenzen entgegen wirkt, wird das Residuum in der Regel ein Minimum haben. Dieses Verfahren wurde von mehreren Forschern wie etwa [65, 93] benutzt, um die relative Latenz zu kalibrieren. Dieses Vorgehen hat aber den Nachteil, dass es nur während einer räumliche Registrierung durchgeführt werden kann. Es müssen also Bewegungen gemacht werden, die sowohl dem Algorithmus für die räumliche Registrierung genügen als auch dem Algorithmus für die zeitliche Kalibrierung. Ein Einsatz dieses Vorgehens während der Laufzeit ist somit ohne weitere Untersuchungen nicht anzuraten. Falls die relative Latenz sich ändern sollte, ist eine Unterbrechung der Anwendung nötig. Und erst dann kann eine erneute Kalibrierung durchgeführt werden. Eine Änderung der relativen Latenz kann auftreten, da die beteiligten Sensoren verschiedene Uhren verbaut haben, die unterschiedliche clock-drifts aufweisen. Oder auch eine vermehrte Last

auf dem Messrechner, falls kein Echtzeitsystem verwendet wird.

In dieser Arbeit wird detailliert auf die einzelnen Schritte eingegangen, die für eine fortlaufende Rekalibrierung der relativen Latenz benötigt werden. Dabei werden unterschiedliche Verfahren für die einzelnen Schritte gegenübergestellt und miteinander verglichen. Der Kern der gesamten Zeitkalibrierung ist die Vergleichsoperation, welche es erlaubt die relative Latenz durch Vergleichen der beiden Sensorsignale zu bestimmen. In dieser Arbeit wird dabei die normalisierte Kreuzkorrelation als Vergleichsmethode vorgeschlagen und untersucht.

Jedoch bedarf es für eine Anwendung der Kreuzkorrelation einer Reihe von Vorarbeiten. So kann die Kreuzkorrelation nur eindimensionale Signale verarbeiten, aber Trackingdaten liegen in der Regel in mindestens drei Freiheitsgraden vor. Auch die Unterschiede der einzelnen Sensoren müssen berücksichtigt werden. So haben zwei verschiedene Sensoren in der Regel auch verschiedene Update-Raten. Dieses muss auch ausgeglichen werden, da für den Vergleich die Daten für den gleichen Zeitpunkt benutzt werden müssen. Dabei müssen die Daten auch interpoliert werden. All diese Schritte sind in Abbildung 3.2 schematisch dargestellt und werden in den folgenden Abschnitten kurz besprochen, damit der Leser einen Überblick über die benutzten Methoden bekommt und auch jede Methode grob einordnen kann.

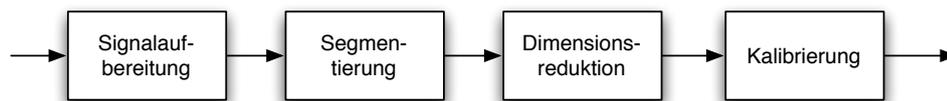


Abbildung 3.2: Vorgeschlagener Ablauf einer zeitlichen Kalibrierung

### 3.3 Voraussetzungen

Um die folgenden Schritte für eine Kalibrierung durchführen zu können, müssen ein paar Voraussetzungen geschaffen werden.

Dieses Verfahren ist nur für eine paarweise Kalibrierung ausgelegt. Es können also nur jeweils zwei Sensoren zueinander zeitlich kalibriert werden. Dies ist keine prinzipielle Einschränkung, denn durch eine paarweise transitive zeitliche Kalibrierung können beliebig viele Sensoren miteinander kalibriert werden. In einem späteren Abschnitt wird an einem Beispiel eine solche transitive zeitliche Kalibrierung durchgeführt (siehe Abschnitt 9.1).

Eine weitere Voraussetzung für die hier vorgestellten Verfahren ist eine korrekte räumliche Registrierung der Sensoren. Nur mit einer korrekten räumlichen Registrierung können die beiden Sensoren miteinander verglichen werden. Es werden zwar auch Methoden in dieser Arbeit beschrieben, die keiner räumlichen Registrierung bedürfen, jedoch wurden diese Verfahren nur im Rahmen einer Voruntersuchung evaluiert. Eine räumliche Registrierung ist aber in den meisten Fällen keine

Einschränkung, da diese auch die Voraussetzung für ein AR System ist. Denn ohne eine solche Registrierung können die virtuellen Überblendungen nicht genau auf die realen Objekte ausgerichtet werden.

Die korrekte Vergabe von Zeitstempeln ist eine weitere Voraussetzung für eine genaue Fusion. Selbst für eine korrekte Darstellung und für die Aufrechterhaltung der AR Illusion müssen die Zeitstempel korrekt vergeben sein. Eine präzise Vergabe der Zeitstempel in einem herkömmlichen Computersystem ist nicht leicht zu gewährleisten, weswegen Sensoren mit eigenem Zeitstempel anderen vorzuziehen sind. Oder es werden zumindest für die Vergabe der Zeitstempel Echtzeitsysteme benutzt [3]. Wenn diese Methoden aus technischen oder anderen Gründen nicht angewendet werden können, kann zumindest die Wiederholrate der Sensoren mit einem Kalman-Filter geschätzt werden. Dieses Verfahren wird in der Ubitrack Software angewendet.

### 3.4 Dimensionsreduktion

Um die Daten effizient interpolieren und später mit der Kreuzkorrelation vergleichen zu können, müssen mehrdimensionale Daten auf eine Dimension reduziert werden. Für die Reduktion der Dimensionen kann auf verschiedene Methoden zurückgegriffen werden. Es ist aber darauf zu achten, welche Typen von Daten vorliegen, damit dann entsprechende Reduktionen ausgewählt werden. So macht es keinen Sinn ein normiertes Quaternion, als Repräsentant einer Orientierung, mit der Norm auf eine Dimension zu reduzieren, da der Wert konstant eins sein wird. Bei Positionen ist dieses Verfahren jedoch anwendbar und wird im Laufe dieser Arbeit erfolgreich eingesetzt.

Wie gerade beschrieben, kann die Situation auftreten, dass durch ein Dimensionsreduktionsverfahren konstante Werte erzeugt werden. Diese Situationen werden auch pathologische Fälle genannt. Durch die Reduktion mehrerer Dimensionen auf eine einzige gehen auch Teile der ursprünglichen Information verloren. Durch eine unglückliche Kombination von Bewegungen und Dimensionsreduktionsverfahren, werden bei der Reduktion genau die Teile aus den Daten eliminiert, die für eine Kalibrierung erforderlich sind. Bei dem oben erwähnten Beispiel, als Dimensionsreduktion die Norm eines normierten Quaternion zu wählen, wären beide Signale konstant gleich eins und somit ein Vergleich zwischen diesen Signalen nicht mehr sinnvoll möglich. Für jedes Dimensionsreduktionsverfahren lassen sich ein oder mehrere pathologische Fälle konstruieren. Zum Beispiel funktioniert die Norm für Positionsdaten im Allgemeinen sehr gut, versagt aber, wenn die Sensoren genau auf einer Kreisbahn um den Ursprung des Bezugssystem bewegt werden. Dann wird das reduzierte Signal konstant dem Radius dieses Kreises sein.

Ein späteres Kapitel in dieser Arbeit wird sich genauer mit diesem Phänomen beschäftigen und auch zeigen, dass die pathologischen Fälle zwar theoretisch existieren, jedoch in der Praxis eine untergeordnete Rolle spielen. Das liegt vor allem daran, dass ein menschliche Benutzer einen Sensor nicht so präzise führen kann,

dass über einen längeren Zeitraum eine pathologische Situation eintritt. Es wird gezeigt werden, dass alleine das Zittern der Hand ausreicht, um die relative Latenz zu bestimmen.

Formal gesehen muss eine dimensionsreduzierende Funktion Werte aus einem Raum mit beliebigen Dimensionen in den eindimensionalen Raum abbilden

$$P : \mathbb{R}^n \rightarrow \mathbb{R}. \quad (3.1)$$

Neben der schon beschriebenen Norm kann auch eine Projektion eine Methode sein, um die Dimensionen von Trackingsignalen zu reduzieren. Im Falle von positiongebenden Sensoren kann dies etwa die Projektion auf eine der Koordinatenachsen sein. So projiziert der Vektor  $\mathbf{w} = (1, 0, 0)^T$  durch  $x^T w$  die Daten auf die  $x$ -Achse. Für diese Dimensionsreduktion wäre der pathologische Fall ein Bewegung, die auf einer Ebene senkrecht zu  $x$ -Achse stattfindet. Die Projektion zweier solcher Datensätze liefert dann konstante Signale, mit welchen keine zeitliche Kalibrierung vorgenommen werden kann. Die Kalibrierung beruht auf einem Vergleich der Signale. Wenn aber beide Signale identisch sind, ist dies nicht möglich. Neben Positionssensoren müssen auch die Daten von Lagesensoren in der Dimension reduziert werden. Eine umfangreichere Darstellung an Dimensionsreduktionsmethoden zeigt das Kapitel 5.

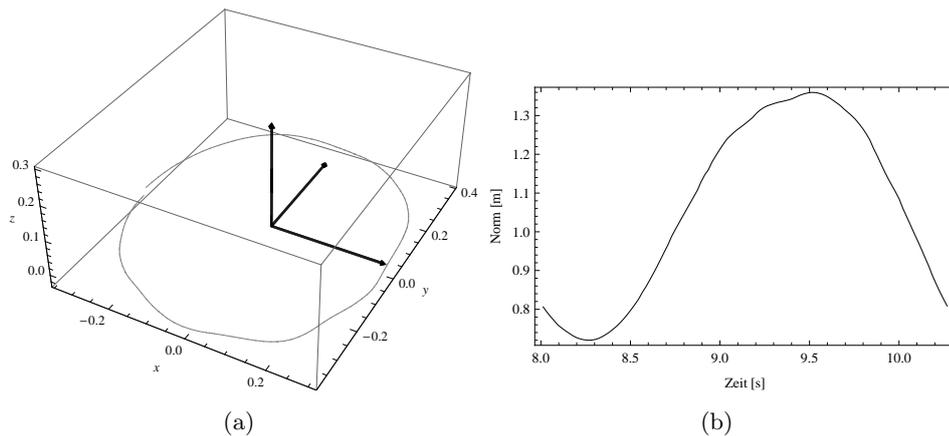


Abbildung 3.3: Dimensionsreduktion einer dreidimensionalen Kurve auf ein eindimensionales Signal mit Hilfe der Norm als Dimensionsreduktionsfunktion; (a) zeigt eine Kreisbewegung; (b) zeigt das dimensionsreduzierte Signal

Bis jetzt wurde nur der Umgang mit Sensoren beschrieben, die absolute Messungen liefern. In AR Umgebungen wird aber auch eine Vielzahl von inertialen Sensoren eingesetzt, die Positionänderungen und Änderungen der Orientierung liefern. Diese Messung sind Ableitungen von Positions- oder Rotationsdaten und können ebenfalls als Daten für eine Kalibrierung verwendet werden.

Meist werden inertiale Sensoren eingesetzt, um die Robustheit des Trackings zu erhöhen [115]. So liefern manche Trackingsysteme ungenaue Sensordaten bei

schnellen oder ruckartigen Bewegungen. Da inertielle Systeme wie Gyroskope oder Beschleunigungssensoren gerade diese Änderungen messen, liefern sie in diesen Situationen genauere Daten. Oft werden inertielle Sensoren auch verwendet, um kurze Zeitspannen ohne ein absolutes Tracking zu überbrücken [109]. Bei einer Kalibrierung von inertialen Sensoren ist darauf zu achten, dass die Daten vergleichbar sind. So sind bei der Verwendung eines Positionssensors und eines Beschleunigungssensors entweder die Daten des Beschleunigungssensors zweimal zu integrieren, oder die Daten des Positionssensors zweimal abzuleiten. Das Ableiten von absoluten Messwerten birgt das Risiko, Fehler zu erzeugen, da das Rauschen des Sensors das Ergebnis negativ beeinflusst. Neben dem Ableiten können die inertialen Sensordaten auch integriert werden, damit sie mit den absoluten Sensormesswerten vergleichbar sind. Dieses Vorgehen ist in einigen inertialen Sensoren implementiert, wie beispielsweise der Sensor MT-9 von Xsens.

In Kapitel 5 werden eine Vielzahl von Dimensionsreduktionsmethoden vorgestellt. Dort wird auch auf adaptive Methoden näher eingegangen, die es erlauben das Auftreten von pathologischen Fällen weitgehend zu vermeiden oder auch den negativen Einfluss der räumlichen Registrierung zu verringern.

### 3.5 Synchronisierung und Segmentierung

Die Daten liegen nun in einer Form vor wie, sie von der Korrelation verarbeitet werden können. Doch die Sensoren liefern die Daten mit unterschiedlichen Frequenzen, so dass zu dem Zeitpunkt einer Messung des Sensors A keine aktuelle Messung des Sensors B vorliegt. Ein direkter Vergleich der beiden diskreten Datensätze ist so nicht möglich. Daher müssen die Daten beider Sensoren zu einander synchronisiert werden.

Eine solche Synchronisierung lässt sich mit Hilfe von Interpolation realisieren. Durch eine lineare Interpolation [65, 81] lässt sich jeder Zeitpunkt sowohl der mehrdimensionalen geometrischen Messdaten als auch der eindimensionalen Daten rekonstruieren.

Durch diese Art der Synchronisierung ist es möglich die Signale um beliebig kleine Schrittwesen in der Zeit zu verschieben ( $x(t) \rightarrow x(t + \delta)$ ). Dies ermöglicht dann in den weiteren Schritten eine Kalibrierung in einer Auflösung, die kleiner ist als die Abstände zwischen zwei Messwerten.

Das Kapitel 6 geht auch auf die Einordnung der Interpolation in den Gesamtprozess ein. Es stellt sich die Frage, ob die Synchronisation, bzw. die Interpolation, vor der Reduktion der Dimension oder nach der Reduktion stattfinden soll. Eine Interpolation der mehrdimensionalen Daten ist nur in seltenen Fällen identisch zur Interpolation mit eindimensionalen Daten. Im Allgemeinen ist die Interpolation von mehrdimensionalen Daten zwar genauer aber auch rechenintensiver. Eine Berechnung der eindimensionalen Daten ist dagegen schnell und für jeden Datentyp identisch. Die Ausführungen in Kapitel 6 zeigen, dass die Fehler einer späteren Interpolation nicht von Relevanz sind.

Der Schritt der Segmentierung ist sehr einfach, aber für ein adaptives Verfahren unabdingbar. Die Signale werden in Stücke von konstanter Zeit aufgeteilt. Da es nicht sicher ist, dass die relative Latenz zwischen zwei Sensoren konstant bleibt, muss das Verfahren sich auf solche Änderungen einstellen können. Würde man die Kalibrierung immer mit allen vorliegenden Daten vornehmen, so würde das Verfahren nicht auf schnelle Schwankungen in der Latenz eingehen können. Auch die Rechenzeit wird bei einer solchen Vorgehensweise zu einem Problem. Die Frage ist also nicht, ob das Signal aufgeteilt werden muss, sondern wie lange die einzelnen Segmente sein müssen. Es muss ein Kompromiss gefunden werden zwischen einer möglichst hohen Anzahl an Messwerten für eine möglichst genaue Schätzung des Offsets und einer möglichst kleinen Anzahl von Messwerten für eine möglichst große Adaptionfähigkeit des Verfahrens. Weiterhin muss berücksichtigt werden, dass mit länger werden Segmenten das Vorhandensein von Ausreißern oder das Ausbleiben von Daten wahrscheinlicher wird. Bei manchen Messverfahren, insbesondere bei optischen Verfahren, kann die Situation eintreten, dass keine Messwerte vorliegen, sei es durch optische Verdeckungen oder durch schnelle Bewegungen. Auf diesen Segmenten kann keine Schätzung der relativen Latenz vorgenommen werden. In den Versuchen in dieser Arbeit hat sich eine Länge von 200 Messwerten als optimal erwiesen.

### 3.6 Time Delay Estimation

Nun liegen alle Voraussetzungen vor, um den zeitlichen Versatz zweier Sensoren zu bestimmen. Die Daten sind eindimensional, sie liegen in kurzen Stücken und für die gleichen Zeiträume vor. Mit Hilfe der Interpolation aus dem vorherigen Schritt ist es möglich den Wert eines Signals zu beliebigen Zeitpunkten zu bestimmen. Dies wird von der folgenden Methode für die Zeitkalibrierung benötigt. Der Schlüssel zu diesem Verfahren ist es, ein Signal Schritt für Schritt in der Zeit zu verschieben und dabei zu beobachten, wann die Ähnlichkeit zwischen diesen Signalen am größten wird. Wichtig sind dabei zwei Punkte: Verschieben und Vergleichen. Das Verschieben der Signale muss feingranular geschehen und dafür kann die eben erwähnte Interpolation dienen. Auf die Interpolation soll im Folgenden nicht mehr eingegangen werden. Sie ist im Prinzip identisch zu der aus dem vorherigen Abschnitt, es sollte aber vermieden werden, dass nicht schon interpolierte Daten nochmals interpoliert werden.

Beim Vergleichen der Signale wird für jede Verschiebung der Grad der Ähnlichkeit zwischen den Signalen berechnet. In dieser Arbeit wird die normalisierte Kreuzkorrelation als Ähnlichkeitsmaß zwischen zwei Signalen vorgeschlagen. Neben der normalisierten Kreuzkorrelation kann aber zum Beispiel auch der geometrische Abstand als Ähnlichkeitsmaß benutzt werden. Nachteil bei der Nutzung eines geometrischen Maßes ist die starke Abhängigkeit von der Güte der Registrierung der räumlichen Beziehung zwischen den beiden Sensoren. Die Korrelation ist hingegen auch ein Maß für den Grad des linearen Zusammenhangs zweier Signale.

Das heißt die Korrelation zweier Signale ist maximal, wenn diese eindimensionalen Signale sich mit einer linearen Abbildung aufeinander abbilden lassen. Die Korrelation zwischen den eindimensionalen Signalen  $s^A(t)$  und  $s^B(t)$  ist also gleich der Korrelation zwischen den Signalen  $s^A(t)$  und  $\alpha + \gamma s^B(t)$ . Mit anderen Worten korrelieren zwei Signale maximal, wenn ein Signal durch Skalierung und durch Verschiebung auf das andere Signal abgebildet werden kann. Für einen Beweis dieser Eigenschaft sei auf Abschnitt 7.5 auf Seite 95 verwiesen.

Angenommen zwei Sensoren A und B liefern exakt die gleichen Daten nur um eine Latenz  $\delta$  verschoben, so lässt sich das ausdrücken durch

$$S^A(t) = S(t) \tag{3.2}$$

$$S^B(t) = S(t + \delta) \tag{3.3}$$

Die normalisierte Kreuzkorrelation, auch Pearsons'sche Korrelationskoeffizient genannt, ist dann für die Signale  $S^A$  und  $S^B$  definiert durch der Kovarianz der beiden Signal, normalisiert mit dem Produkt der beiden Standardabweichungen. Formal lautet die Definition wie folgt

$$\rho_{S^A, S^B} = \text{Corr}(S^A, S^B) := \frac{\text{Cov}(S^A, S^B)}{\sigma_{S^A} \sigma_{S^B}},$$

wobei  $\text{Cov}(S^A, S^B)$  die Kovarianz der beiden Signale ist und  $\sigma_{S^A}$  bzw.  $\sigma_{S^B}$  die Standardabweichungen der Signale  $S^A$  bzw.  $S^B$  sind. Die Kreuzkorrelation zwischen  $S^A$  bzw.  $S^B$  lässt sich dann schreiben als

$$\text{Corr}(S^A(t), S^B(t)) = \text{Corr}(S(t), S(t + \delta)). \tag{3.4}$$

Eines der beiden Signale wird nun um den Parameter  $\tilde{\delta}$  in der Zeit verschoben und die Signale dann mit der Korrelation verglichen. Dies drückt der Term  $\text{Corr}(S(t + \tilde{\delta}), S(t + \delta))$  aus, welcher maximal wird wenn  $\tilde{\delta} = \delta$ , also der Parameter  $\tilde{\delta}$  gleich der relativen Latenz ist. Somit besteht die Lösung für die Bestimmung der relativen Latenz in der Maximierung der Korrelation der beiden Signale  $S(t + \tilde{\delta})$  und  $S(t + \delta)$

$$\tilde{\delta} = \text{argmax}_{\tilde{\delta}} \{ \rho_{S^A_{(\tilde{\delta}t)}, A^B_{(\delta t)}} \}. \tag{3.5}$$

In Abbildung 3.4 ist ein Beispiel für einen solchen Maximiervorgang zu sehen. An der x-Achse sind die getesteten  $\delta$ , also die verschiedenen zu testenden relativen Latenzen, angetragen und an der y-Achse die zugehörigen normalisierten Korrelationskoeffizienten. Der Suchraum für die relative Latenz war in diesem Beispiel  $\delta \in [-0,1 \text{ s}; 0,1 \text{ s}]$ . Gut zu erkennen ist, dass das Maximum dieser Funktion mit einem Wert von knapp unter Eins an der Stelle  $\delta = 0,033 \text{ s}$  erreicht wird. An dieser Stelle ist eine maximale Ähnlichkeit der Signale hergestellt worden und somit ist dieser Wert die Schätzung der relative Latenz.

Der letzte Absatz beschrieb die Bestimmung der relativen Latenz unter Verwendung der Korrelation als Ähnlichkeitsmaß. Dabei wurden einige Einschränkungen

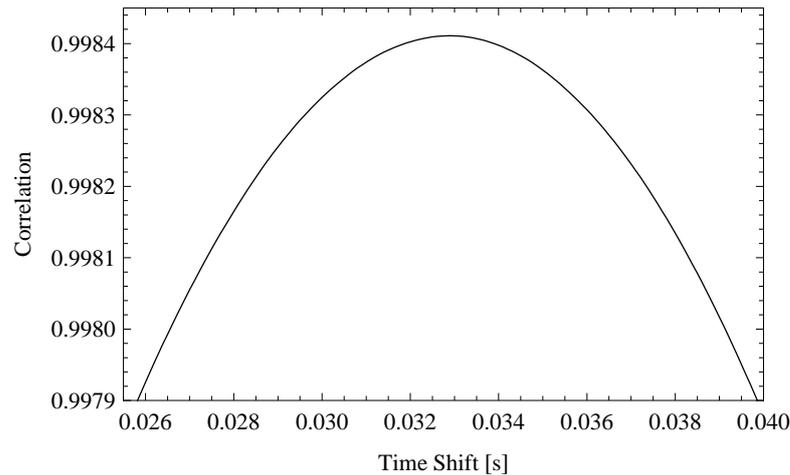


Abbildung 3.4: Zeitverschiebungsplot zeigt die Korrelation zwischen einem Signal  $x(t)$  und einem Signal  $y(t + \delta)$  mit  $0,026 \text{ s} \leq \delta \leq 0,040 \text{ s}$

gemacht. So wurde angenommen, dass beide Sensoren kein Eigenrauschen aufweisen und beide Sensoren perfekt zueinander registriert sind. Diese Annahmen treffen in der Realität nicht zu. Dort treten neben den eben erwähnten Fehlern auch Ausreißer auf oder einer der Sensoren liefert keine Daten. All diese Einflüsse verschlechtern das Ergebnis der zeitlichen Kalibrierung.

In der Literatur gibt es eine theoretische untere Schranke für die Genauigkeit einer solchen Kalibrierung. Die Cramér-Rao Ungleichung oder auch Cramér-Rao-Bound (CRB) liefert eine untere Schranke für die Varianz eines Schätzers. Diese Schranke hängt nicht von verwendeten Algorithmen oder benutzten Methoden ab, sondern alleine auf von den Sensordaten und deren Unsicherheit, wie beispielsweise Rauschen. Die CRB liefert somit eine untere Schranke für jegliches Verfahren. In der Praxis ist eine solche Schranke jedoch nur insoweit relevant, als dass die einzelnen Verfahren sich mit dieser unteren Schranke messen und vergleichen lassen.

### 3.7 Korrektur

Mit der soeben bestimmten relativen Latenz sollen nun die Ergebnisse des Tracking in einer dynamischen AR Anwendung verbessert werden.

Für eine Korrektur der relativen Latenz gibt es zwei Möglichkeiten. Entweder wird das vorauseilende Signal um den bestimmten Zeitraum  $\delta$  aufgehalten oder das hinterherhinkende Signal wird um den Zeitraum  $\delta$  in die Zukunft prognostiziert.

Beide Möglichkeiten haben Vor- und Nachteile. So ist das Aufhalten eines Signals per se sehr genau, denn damit werden keine zusätzlichen Fehler in das System eingebracht. AR Anwendungen müssen registriert und in Echtzeit arbeiten, sonst werden sie von den Benutzern nicht akzeptiert. Ein Tracking Framework, welches Signale aufhält, um Zeitsynchronizität zu schaffen, wird die Trackingdaten erst

zur Verfügung stellen können, wenn der langsamste aller Sensoren die Daten im Framework abgeliefert hat. Damit erhält man zwar in sich stimmige und genaue Trackingdaten, man muss aber auf den langsamsten Sensor warten. In der Literatur werden für typische AR Anwendungen Gesamtlatenzzeiten von ca. 100 ms angegeben [3]. In den späteren Auswertungen werden alleine für die relative Latenz zwischen zwei Sensoren Werte über dieser Schranke ermittelt. Die Methode alle Trackingdaten auf den langsamsten Sensor zu synchronisieren bietet in mancher Situation großes Potential. Gerade in Anwendungen, die eine hohe Präzision aber keine Echtzeitdaten erfordern, ist diese Methode anzuwenden. Erwähnt seien hier spezielle Messanwendungen, Kalibrierungen [65] oder auch die Aufnahme von *ground truth* Daten [35]. Für die Korrektur der Daten in der Arbeit von Gruber u. a. [35] wurden die Daten mit dem ubitrack Framework und dem in Abbildung 3.5 dargestellten ubitrack Pattern korrigiert. Manche AR-Systeme können von dieser leicht zu implementierenden genauen Methode profitieren [6].

Die Methode, ein Signal zu extrapolieren und somit zu prognostizieren, erlaubt es die relative Latenz zu korrigieren, ohne die Gesamtlatenz des Systems zu vergrößern. Eine Vorhersage der Daten ist nicht ohne Fehler möglich, und mit steigender relativer Latenz wird diese Vorhersage immer ungenauer. Diese Fehler müssen jedoch akzeptiert werden, wenn die Gesamtlatenz mit dem reinen Aufhalten der Daten zu groß wird. Neben den unvermeidbaren Fehlern, die bei einer Vorhersage von Sensordaten entstehen, ist die Methode Daten vorausszusagen nicht immer möglich. Benutzt beispielsweise ein AR System eine Kamera, dann müssen neben den relativen Latenzen zwischen den Sensoren auch die relativen Latenzen zwischen Kamera und Sensoren bestimmt und korrigiert werden. Die Videobilder der Kamera können aber nicht in die Zukunft vorhergesagt werden. Somit müssen die Daten von Sensoren, die schneller sind als die Bilddaten der Kamera, verzögert werden [51].

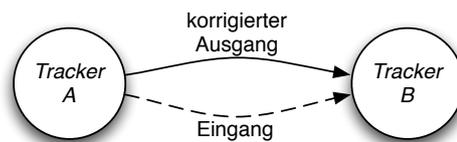


Abbildung 3.5: SRG Pattern zur Korrektur eines zeitlichen Versatzes.

Somit gibt es zwei Methoden, um die Synchronizität der verwendeten Sensoren und Kameras zu gewährleisten. Die Daten können entweder verzögert oder prognostiziert werden. Dabei ist die Prognose von Trackingdaten komplexer als das simple Verzögern von Daten. In der Literatur geht die Bandbreite von einfachen linearen Modellen für die Extrapolation [51] bis hin zu aufwändigeren Kalman-Filtern, die ein Bewegungsmodell enthalten [5]. Mit den so korrigierten Trackingdaten kann dann eine genaue Sensorfusion durchgeführt werden.

## 3.8 Zusammenfassung

Dieses Kapitel beleuchtete die Vorarbeiten im Bereich der Zeitkalibrierung, damit die Beiträge dieser Arbeit besser in den aktuellen Stand der Forschung eingeordnet werden können. Weiterhin wurden eine erste Skizze des Lösungsvorschlags für die Kalibrierung der relativen Latenz gegeben und die Voraussetzungen für das Verfahren bestimmt. Anschließend wurde ein Überblick über alle für eine Zeitkalibrierung nötigen Schritte wie Dimensionsreduktion, Synchronisierung und Segmentierung gegeben und zuletzt die Kalibrierung der relativen Latenz und deren Korrektur vorgestellt.

Die in diesem Kapitel aufgezeigten Methoden werden in den nun folgenden Kapiteln im Detail dargestellt und evaluiert.



## 4 Eigenschaften von Signalen

In diesem Kapitel werden Fehlerquellen betrachtet, die eine zeitliche Kalibrierung zweier Sensoren negativ beeinflussen. Solche negativen Einflüsse sind zum Beispiel schlechte räumliche Registrierungen der verwendeten Sensoren oder schwankende Wiederholraten. Diese Fehlerquellen werden dabei erst beschrieben, dann wird untersucht, wie sie sich auf die Genauigkeit der zeitlichen Kalibrierung auswirken. Und zuletzt werden Verfahren präsentiert, die eine Reduktion dieser Einflüsse zulassen.

### 4.1 Räumliche Registrierung

Eine gute räumliche Registrierung der verwendeten Sensoren ist in der Regel eine notwendige Grundbedingung für eine zeitliche Kalibrierung. Nachdem die Kalibrierung der relativen Latenz auf dem Prinzip Verschieben und Vergleichen beruht, müssen die Sensoren die Messwerte bezüglich des gleichen Koordinatensystems liefern. Sonst ist ein Vergleich nicht möglich. Jedoch ist diese Vorbedingung keine zusätzliche Anforderung an das System, denn für ein Augmented Reality System, welches mindestens zwei Sensoren fusioniert, müssen die räumliche Beziehungen zwischen diesen Sensoren vorab schon registriert worden sein.

Formal wird bei einer Registrierung eine Koordinatentransformation bestimmt, die es erlaubt die Messwerte, die in dem Koordinatensystem eines Sensors gemessen wurden, in das Koordinatensystem eines zweiten Sensors zu transformieren

$$S^A = \mathbf{H}S^B.$$

Eine solche Koordinatentransformation existiert für Orientierungen, Positionen oder auch für komplette Posen mit sechs Freiheitsgraden. Die Bestimmung der Transformation wird auch Registrierung genannt und mit Hilfe von Algorithmen wie Hand-Eye-Calibration [104, 23], Absolute-Orientierung [46] oder ähnlichen Verfahren berechnet. Auf die einzelnen Methoden und welche konkreten Fehler diese erzeugen, soll hier nicht weiter eingegangen werden.

Wenn die Beziehung zwischen den Sensoren mit diesen Verfahren hergestellt wurde, muss das Bezugssystem eines der Sensoren geändert werden, indem die Daten mit der gewonnenen Transformation transformiert werden. In der Regel wird die Registrierung nicht perfekt sein und mit Fehlern behaftet sein. Die Ursache hierfür sind das Rauschverhalten der Sensoren, systematische Fehler und dynamische Fehler der Trackingsysteme. Diese Fehler sind nicht zu vermeiden, aber die dynamischen Fehler können mit einer korrekten Kalibrierung der relativen Latenz

verringert werden. Die dynamischen Fehler haben einen großen Anteil am Gesamtfehler eines AR Systems [2]. Da zu dem Zeitpunkt der räumlichen Registrierung noch keine Korrektur der relativen Latenz vorliegt, muss bei der Registrierung darauf geachtet werden, dass die Sensoren nicht in Bewegung sind. Falls eine Registrierung im Stillstand nicht möglich ist, können die räumliche Registrierung und zeitliche Kalibrierung in iterativen Schritten wiederholt werden.

Wird die kanonische Korrelation als Dimensionsreduktionsverfahren verwendet, ist keine vorherige Registrierung der absoluten Messdaten nötig. Denn dieses Verfahren bietet eine Invarianz gegenüber affinen Abbildungen zwischen den mehrdimensionalen Signalen. Erste Voruntersuchungen haben jedoch keine vielversprechenden Ergebnisse gezeigt. Hierzu sollten weitere Untersuchungen angestrengt werden, da das Potential einer registrierungsunabhängigen Kalibrierung so hoch ist. Eine Möglichkeit ist, diese Methode mit Hilfe synthetischer Daten und der Monte-Carlo-Methode (s. Kapitel 8) zu untersuchen.

Um die Abhängigkeit von der Registrierung zu demonstrieren, wurde ein kurzes Experiment durchgeführt. Dazu wurde ein 5 s langes Segment einer Aufnahme dupliziert, so dass zwei identische Segmente vorlagen. Mit diesen zwei identischen Segmenten wurde dann eine zeitliche Kalibrierung durchgeführt. Wie zu erwarten lieferte die Kalibrierung eine relative Latenz von  $\delta = 0$  ms. Für die Simulation einer fehlerhaften Registrierung, kann man eines der Signale stückweise gegen das andere identische Signal im Raum verschieben oder verdrehen. Nach jeder Translation oder Rotation wurde dann die zeitliche Kalibrierung wiederholt. Je signifikanter die Transformation eines der Datenströme ist, desto größer wird der Fehler der Schätzung der relativen Latenz. Die Ergebnisse dieser Versuche sind in Abbildung 4.1 zu sehen. In Abbildung 4.1(a) ist ein Signal im Bereich  $\pm 21^\circ$  rotiert und dabei an diskreten Stellen die relative Latenz bestimmt. Für diese sehr großen Orientierungsfehler ergeben sich Fehler in der relativen Latenz von mehr als 30 ms. Mit dem Auge sind in den Zeitversatzplots keine Unterschiede zu erkennen (vergl. 4.1(b)). In der Praxis treten solch große Registrierfehler nicht auf, weshalb für die Abbildung 4.1(c) der Orientierungsfehler in einem kleineren Intervall ( $\pm 0,2^\circ$ ) untersucht wurde. Für dieses Intervall wurde eine Schwankung der relativen Latenz in einem Bereich von  $\pm 0,3$  ms errechnet. Auch ein Fehler im translationalen Teil der Registrierung verursacht Fehler in der Kalibrierung der relativen Latenz. Um dies zu demonstrieren wurde eine Kurve  $\pm 2$  m gegen die zweite Kurve verschoben. Wie in Abbildung 4.1(d) dargestellt, trat dabei ein Kalibrierfehler von  $\pm 1$  ms auf.

Es bleibt somit festzuhalten, dass die Genauigkeit der Registrierung beider Sensoren einen Einfluss auf die Genauigkeit der zeitlichen Kalibrierung hat. Wie stark der Einfluss auf die Kalibrierung ist, hängt vom speziellen Einzelfall der Daten ab. In diesem Zusammenhang sei auch die kanonische Korrelation erwähnt. Diese Methode verspricht eine Kalibrierung der relativen Latenz unabhängig von einer Registrierung (vgl. Abschnitt 5.3.1 auf Seite 74).

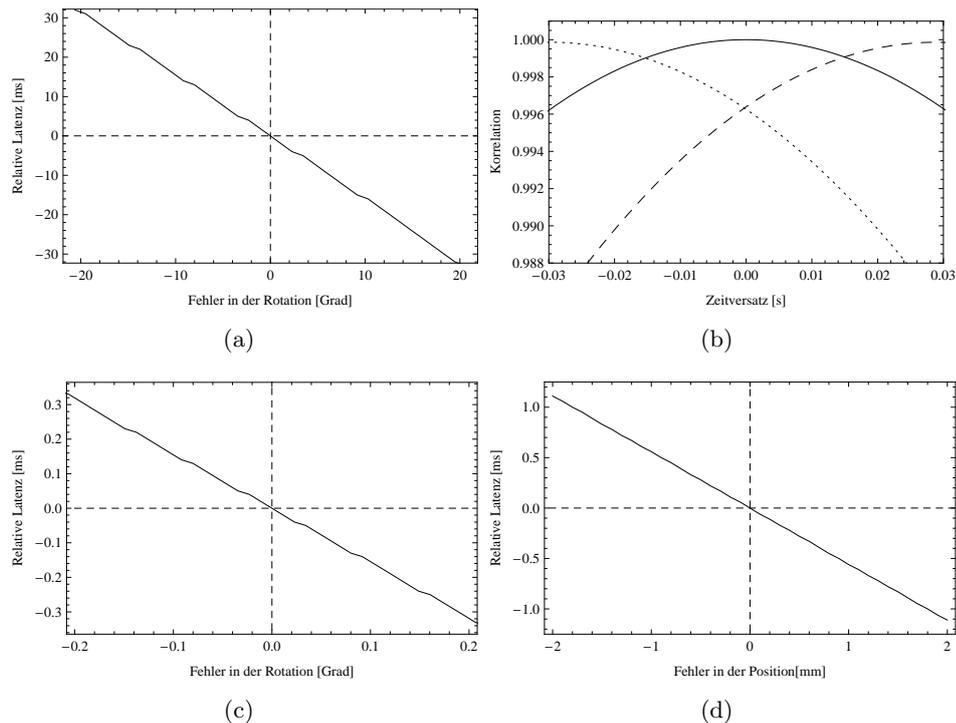


Abbildung 4.1: Fehler ausgelöst durch eine falsche räumliche Registrierung demonstriert durch eine Autokalibrierung; (a) zeigt beispielhaft die Abhängigkeit der Kalibrierung der relativen Latenz vom Fehler in der Registrierung der Orientierung; (b) zeigt die Korrelationskurven bei einem Orientierungsfehler von ca.  $-20,6^\circ$ ,  $0^\circ$  und  $+20,6^\circ$ ; (c) zeigt die Auswirkung eines Registrierungsfehlers in der Rotation im Bereich von  $-0,2^\circ$  bis  $0,2^\circ$ ; (d) zeigt die Auswirkungen eines Registrierungsfehlers in der Position auf die Kalibrierung der relativen Latenz.

## 4.2 Update-Raten

Die Anzahl an Messungen, die ein Sensor pro Sekunde vornimmt und im System abliefern, wird Update-Rate oder Wiederholrate genannt. Diese werden angegeben als Frequenz. Die Höhe der Frequenz und deren Stabilität haben Einfluss auf die Genauigkeit der zeitlichen Kalibrierung. Die folgenden Absätze gehen näher auf diese Eigenschaften von Messsignalen ein. Dabei werden die Ursachen und Auswirkungen der Update-Rate, der Frequenzstabilität und des Schwankung der relativen Latenz näher beschrieben.

Damit Sensoren die aufgenommen Messwerte in regelmäßiger Weise an andere Systeme weitergeben können, müssen sie eine Uhr und damit einen Taktgeber verbaut haben. Als Taktgeber kann beispielsweise eine einfache elektronische Schaltung dienen oder eine auf einem Schwingquarz beruhende Schaltung. Es gibt

dabei unterschiedliche Qualitätsstufen hinsichtlich der Stabilität des Schwingverhaltens [99].

Neben den Update-Raten des Sensors zählt vor allem die Art und Weise wie die Tracking Software die eingehenden Daten behandelt zu den kritischen Faktoren. Bei der eingesetzten ubitrack Software werden eingehende Daten zuerst mit einem Zeitstempel versehen. Diese Aufgabe wird mit höchster Priorität durchgeführt, denn sie hat entscheidenden Einfluss auf die erzielbaren Genauigkeiten. Da aber kein Echtzeitsystem verwendet wurde, kann eine optimale Antwortzeit des Systems nicht garantiert werden. Ist das System unter Last kann es somit vorkommen, dass Messdaten zu spät mit einem Zeitstempel versehen wurden. Sind die Messungen aber schon mit einem Zeitstempel versehen, ist die größte Gefahr gebannt. Denn solange das System nicht so unter Last ist, dass es die Gesamtheit an Messungen nicht mehr verarbeiten kann, kann es anhand der Zeitstempel die Messungen immer in der richtigen Reihenfolge und mit den anderen Messungen synchron halten. Dies sind aber Aufgaben des Trackingframeworks. Für eine detaillierte Abhandlung siehe die Arbeiten von Huber [50, 49] für den Einsatz von Echtzeitsystemen für AR Anwendungen.

**Update-Rate** Die Höhe der Update-Rate, auch Wiederholrate genannt, hat einen Einfluss die Genauigkeit der Kalibrierung.

Bei einer niedrigen Update-Rate liegen weniger Daten vor, die bei einer in der Zeit konstanten Segmentlänge für eine Kalibrierung verwendet werden können. Eine Vergrößerung der Segmente für die Kalibrierung kann diesen Malus ausgleichen, aber zugleich wird die Adaptivität des Kalibrierung verringert. Als Beispiel sollen hier der Wiederholraten eines auf Kameradaten arbeitenden Marker Trackers und des A.R.T. Trackingsystems verglichen werden. Das A.R.T. Trackingsystem liefert eine Update-Rate von ca. 60 Hz, aber der Marker Tracker nur 14 Hz. Soll die Kalibrierung mit 240 Messungen durchgeführt werden, sind vom A.R.T. nur 4s Daten nötig. Für die gleiche Anzahl von Messungen müssen mit dem Markertracker mehr als 17s Daten aufgenommen werden. Daher muss die Segmentlänge in Zeiteinheiten entsprechend der Update-Rate angepasst werden, damit die gewünschte Anzahl von Messwerte vorliegt. Bei einem langen Segment steigt die Gefahr, dass Schwankungen in der relativen Latenz nicht mehr zuverlässig erkannt werden. Schwankungen die relativen Latenz innerhalb der Dauer eines Segments sind dann nicht mehr zu ermitteln.

Ebenfalls durch eine niedrige Update-Rate bedingt steigen die Interpolationsfehler bei der Synchronisation und bei der Zeitkalibrierung.

**Jitter** Unregelmäßigkeiten in der Zeitdimension ergeben sich sowohl durch Unregelmäßigkeiten des Zeitgebers in der Sensor Hardware als auch durch die Tracking Systeme auf den Messrechnern.

Im der Abbildung 4.2 ist die Update-Rate des verwendeten FaroArms abgebildet. Der FaroArm schickt die Daten nicht autonom in den Rechner, sondern

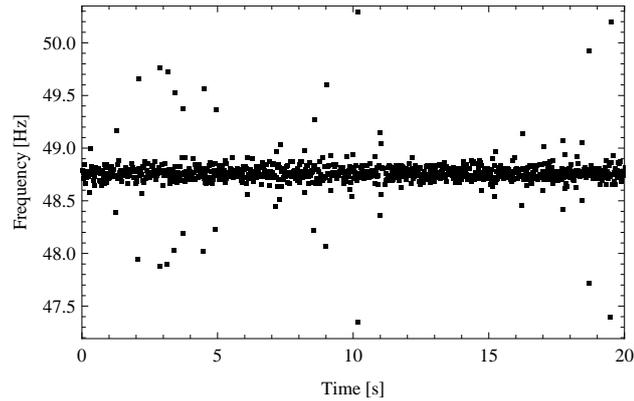


Abbildung 4.2: Update-Rate des FaroArm ( $\mu_f = 48,8$  Hz,  $\sigma_f = 0,53$  Hz)

der Rechner frägt, gesteuert über die Software Uhr, die Daten vom FaroArm ab. Die ermittelte Standardabweichung  $\sigma_f = 0,53$  Hz von der durchschnittlichen Frequenz  $\mu_f = 48,8$  Hz wird Jitter genannt. Betrachtet man die durchschnittliche Zeit zwischen zwei Messungen und deren Standardabweichungen, kann man eine untere Schranke für die mögliche Zeitkalibrierung angeben. In diesem Beispiel ist die durchschnittliche Zeit zwischen zwei Dateneinheiten ( $\mu_t = 20,51$  ms) und eine Standardabweichung von  $\sigma_t = 0,37$  ms.

Bei dem schon erwähnten in ubitrack integrierten Markertracker wurde ebenfalls die Wiederholraten überprüft. In Abbildung 4.3 sind die ermittelten Wiederholraten über einen Zeitraum von 40 s dargestellt. Abbildung 4.3(a) zeigt eine ermittelte Verteilung der Frequenzen. Es sind drei verschiedene klar von einander zu unterscheidende Frequenzen zu erkennen ( $\mu_1 = 15,6$  Hz,  $\mu_2 = 10,4$  Hz,  $\mu_3 = 7,81$  Hz). Die Frequenz  $\mu_1$  tritt mit einer Häufigkeit von 85 % auf und ist somit die Frequenz, die am häufigsten auftritt. Die Frequenz  $\mu_3$  ist bis auf eine kleine Abweichung, die halbe Frequenz von  $\mu_1$  und somit durch Ausbleiben eines Bildes (engl.: *framedrop*) der Kamera zu erklären. Diese Frequenz tritt mit einer Häufigkeit von 6,4 % auf. Für die dritte Frequenz  $\mu_2$  kann man vermuten, dass diese durch Irregularitäten im Rechner, in dem verwendeten Betriebssystem oder in der Kamera selbst ausgelöst wird.

Welche Auswirkungen die eben aufgeführten Schwankungen auf die Genauigkeiten haben, bedarf jedoch noch weiterer Untersuchungen. Man kann vermuten, dass der Ausfall einzelner Bilder zu keinen großen Ungenauigkeiten führen wird. Denn wenn die Videobilder beim Eingang in die Tracking Software ordnungsgemäß mit einem Zeitstempel versehen wurden, sind die Bilder und Zeitstempel auch bei dem Ausfall von Einzelbildern weiterhin konsistent. Ausfälle von Bildern führen aber auch zu ausbleibenden Messungen, was zu Fehlern in der Interpolation und somit auch in der zeitlichen Kalibrierung führen kann.

**Varianz der relativen Latenz** Die Varianz der relativen Latenz beschreibt den Grad der Schwankungen der relativen Latenz über die Zeit während der Nutzung

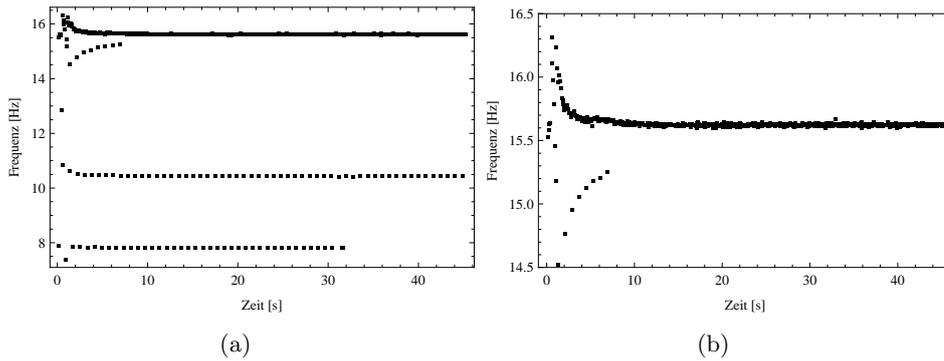


Abbildung 4.3: Update-Rate des Marker Trackers: (a) zeigt alle auftretenden Frequenzen (15,6 Hz, 10,4 Hz, 7,81 Hz); (b) zeigt die Hauptfrequenz (15,6 Hz)

eines Systems. Nicht gemeint ist die Varianz der Latenz unter Einsatz anderer Software oder Rekonfiguration des Aufbaus eines Systems, sondern die Schwankungen ausgelöst durch den Sensor selbst. Als Beispiel für die Varianz der relativen Latenz sei die Arbeit von Aron u. a. zitiert [1]. Die Autoren haben eine zeitliche Kalibrierung zwischen einem Xsens Gyroskop (Modell MT9-B) und einer Kamera vorgenommen. Dazu haben sie erst versucht mit der Methode von Prager [79] (siehe Abschnitt 3.1) eine Kalibrierung vorzunehmen. Dabei haben sie festgestellt, dass die relative Latenz bei dieser Kombination von Sensoren nicht konstant ist. Nach 75 Bildern hat sich die relative Latenz von einem Bild auf fünf Bilder gesteigert und nach weiteren 55 Bildern wieder zurück auf zwei Bilder. Um diese Schwankungen auszugleichen haben sie ein eigenes adaptives Verfahren entwickelt, welches in Abschnitt 3.1 genauer beschrieben wird. In der Arbeit von Aron wird leider keine Update-Rate der Kamera angegeben. Somit ist eine absolute Angabe in Sekunden nicht möglich. Geht man aber von einer Wiederholfrequenz von 30 Hz aus, bedeutet das eine Halbierung der relativen Latenz in weniger als 2 s. Solche großen Schwankungen können durch das in dieser Arbeit propagierte Verfahren nicht aufgefangen werden. Denn sie finden in einem Zeitraum statt, den die Methode der Korrelation für eine zuverlässige Schätzung braucht. Was die Ursachen für solche starken Schwankungen der relativen Latenz sind, konnte nicht festgestellt werden und auch die Literatur gibt keine Hinweise darauf. Eine mögliche Ursache für diese Art von Schwankungen kann aber ein schlechter Taktgeber (engl.: *clocks*) der Sensoren sein. Azuma beschreibt in [4] einen Sensor, dessen Uhr in drei Minuten um 8,3 ms also in einer Sekunde umgerechnet 0,046 ms von einer Referenzzeit abweicht. Diese Abweichungen können jedoch nicht diese doch sehr großen Schwankungen des Xsens Sensors erklären. Es bleibt also ungewiss, ob die Instabilität der verwendeten Taktgeber tatsächlich für die oben genannten Schwankungen ursächlich ist.

**Verbesserung** Es gibt verschiedene Möglichkeiten, um die Genauigkeit der Wiederholrate der Sensoren zu verbessern. Das sind zum einen Verfahren, die von den Herstellern der Sensoren umzusetzen sind, oder Verfahren, die Wiederholrate mit Hilfe eines Kalmanfilters schätzen. Auf beide Arten dieser Verfahren soll im Folgenden kurz eingegangen werden.

Von den Herstellern der Sensoren könnte eine Möglichkeit für eine externe Taktung geschaffen werden. Somit ist es möglich, einen einzigen Taktgeber für alle verwendeten Sensoren zu benutzen. Zum anderen ist es mit einer zentralen Uhr als Taktgeber leichter möglich eine hochpräzise Uhr einzusetzen und somit die Genauigkeit des Taktsignals zu erhöhen. Eine weitere Methode ist die Messwerte schon im Sensor mit Zeitstempeln zu versehen. Dadurch wirken sich zeitliche Schwankungen in den Übertragungskanälen nicht mehr negativ auf die relative Latenz aus. Werden Messwerte beispielsweise erst in einem Computersystem mit einem Zeitstempel versehen, so können sich unterschiedliche Auslastungen der Rechner auf die Vergabe der Zeitstempel auswirken. Diese beiden Möglichkeiten sind auch für sich genommen vorteilhaft und können die Genauigkeit und Zuverlässigkeit erhöhen.

Die eben aufgeführten Möglichkeiten verhindern, dass Ungenauigkeiten die durch ungenaue Zeitstempel verursacht werden. Doch diese Änderungen in den Sensoren kosten Geld und werden somit von den Herstellern in den meisten Fällen nicht implementiert. Es ist jedoch auch möglich ohne Eingriffe in die Geräte die Genauigkeit der Zeitstempel zu erhöhen. Auf der Seite der Tracking Software kann beispielsweise ein Echtzeitsystem verwendet werden [4] und darauf geachtet werden, dass die Zeitstempel so früh wie möglich und mit höchster Priorität vergeben werden [51, 81]. Im verwendeten ubitrack framework sind Methoden implementiert, die die Wiederholrate eines Sensors mit Hilfe eines Kalmanfilters schätzen. Dadurch kann selbst bei Schwankungen innerhalb des Übertragungskanals den Messungen ein korrigierter Zeitstempel aufgedrückt werden [57].

### 4.3 Ausreißer

Als Ausreißer werden Messwerte bezeichnet, die nicht in eine Messreihe passen oder nicht den Erwartungen entsprechen. In Augmented Reality Anwendungen sind die Erwartungen an die Messungen meist durch die Möglichkeiten der menschlichen Bewegungen definiert oder durch die physikalischen Modelle beschränkt. Ein Beispiel für einen Ausreißer ist in der Abbildung 4.4(b) zu sehen. Ausreißer sind für die Kalibrierungen im Allgemeinen ein Problem, da durch große Messfehler, also Ausreißer, die Optimierung in eine falsche Richtung gedrängt wird. Auch die Kalibrierung der relativen Latenz ist ohne besondere Maßnahmen anfällig für Ausreißer. Das Ziel ist natürlich den Einfluss von Ausreißern zu minimieren, das Verfahren also robust zu gestalten. In dem folgenden Abschnitt, werden die Auswirkungen von Ausreißern auf die zeitliche Kalibrierung demonstriert und Möglichkeiten aufgezeigt diese zu minimieren.

Um die Auswirkungen von Ausreißern zu demonstrieren, sei zu Beginn ein kleines Beispiel gezeigt. Dazu ist in Abbildung 4.4(a) ein mit der Norm dimensionsreduziertes 4 s langes Segment zu sehen. Das selbe Signal ist in Abbildung 4.4(b) mit Ausreißer sehen. Für dieses Beispiel ist der Ausreißer manuell hinzugefügt worden. Mit diesen zwei kurzen Segmenten wird jeweils eine Zeitkalibrierung mit der Kreuzkorrelationsmethode durchgeführt. Die resultierenden Ergebnisse sind in den Abbildungen 4.5 zu sehen.

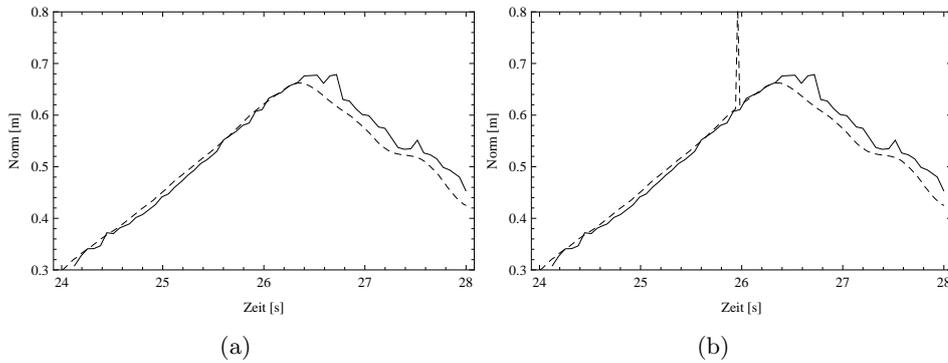


Abbildung 4.4: Ausreißer in Messdaten. Die beiden Abbildungen zeigen durch die Norm dimensionsreduzierte Messwerte des A.R.T. Systems (a) ohne Ausreißer; (b) mit einem Ausreißer

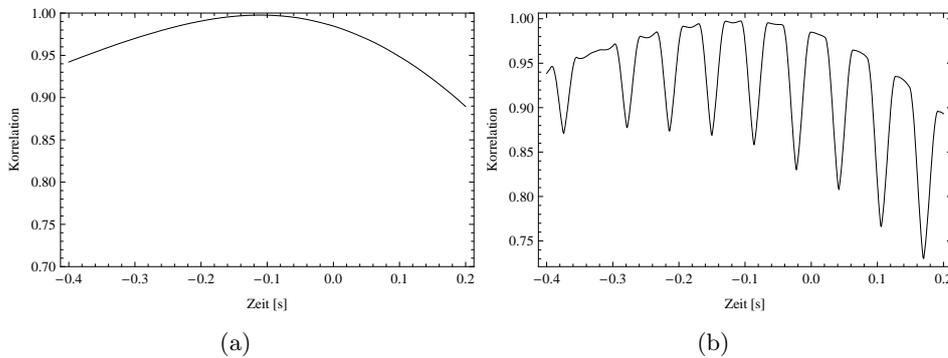


Abbildung 4.5: Ergebnis der zeitlichen Kalibrierung: Zeitverschiebungsgraph (a) ohne Ausreißer und (b) mit einem Ausreißer

Die in Abbildung 4.5(a) abgebildete Zeitverschiebungskurve hat nur ein einziges Maximum bei  $\delta t = -111$  ms und bei einem Korrelationskoeffizienten von  $\rho = 0,998$ . Somit lässt sich leicht die Latenz bestimmen, die eine bestmögliche Übereinstimmung der Signale erzeugt. In Abb. 4.5(b) ist die Zeitverschiebungskurve der in Abb. 4.4(b) gezeigten Signale zu sehen. Es ist immer noch eine ähnliche Kurve zu sehen wie ohne Ausreißer, jedoch tritt jetzt eine Art Schwingung in der Zeitverschiebungskurve auf. Das Maximum dieser Kurve liegt bei

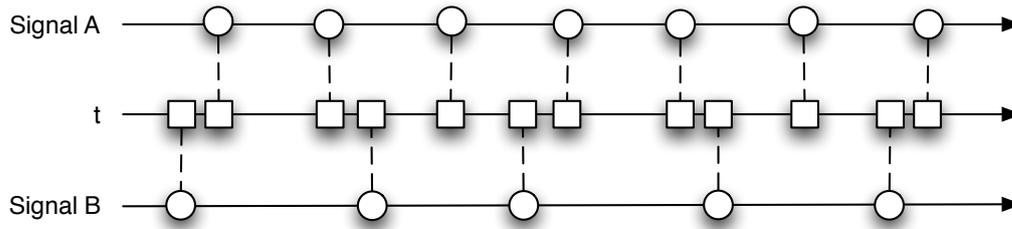
( $\delta = -107$  ms,  $\rho = 0,998$ ) und somit fast identisch wie ohne Ausreißer.

Die Ursache für das Schwingen liegt in der Synchronisierung der beiden Signale. In diesem Beispiel gibt das Signal  $A$  ohne Ausreißer die Frequenz  $f^A = 14,9$  Hz für die Synchronisierung vor und somit wird das Signal  $B$  ohne Ausreißer interpoliert und gegen das Signal  $A$  verschoben. Das Signal  $B$  hat eine Taktfrequenz von  $f^B = 48,6$  Hz. Ein Ausreißer in Signal  $B$  ist von extrem kurzer Dauer gegenüber dem Signal  $A$ . Nachdem aber Signal  $B$  nun Stück für Stück gegen Signal  $A$  verschoben wird, wird der Ausreißer regelmäßig für die Interpolation benutzt und liefert dann schlechtere Werte für die Kreuzkorrelation. Wird ein solcher Ausreißer nicht entfernt, so führt dies bei der Zeitschätzung zu einem Fehler. Wie schon angedeutet wurde, existieren Methoden, die es erlauben fehlerhafte Zeitschätzungen zu erkennen und entsprechend weniger oder gar nicht in die Kalibrierungen einfließen zu lassen. Während die negativen Auswirkungen eines einzelnen Ausreißers mit dieser Methode noch ausgeglichen werden können, ist dies bei einem gehäuften Auftreten von Ausreißern nicht mehr möglich. Eine solche Häufung von Ausreißern, kann auftreten, wenn beispielsweise ein Markertracker einen Marker gerade noch erkennen kann. Ohne eine Entfernung dieser Ausreißer ist dann keine sinnvolle Zeitschätzung möglich.

Im Folgenden wird anhand der RANSAC Methode beschrieben, wie Ausreißer effizient gefiltert werden können. Dazu wird erst das Prinzip der RANSAC Methode beschrieben und dann eine Methode vorgestellt, wie diese effizient zur Filterung benutzt werden kann, damit das gefilterte Signal zur Laufzeit in einem AR System eingesetzt werden kann. Das RANSAC Verfahren ist ein einfaches, schnelles und einfach zu implementierendes Verfahren, das von Fischler und Bolles 1981 entwickelt wurde [28].

In [38] wird von Hartley und Zisserman eine kurze Einführung in dieses Thema gegeben. Die Abkürzung RANSAC ist ein Akronym für „RANdom SAMple Consensus“ was auf Deutsch mit „Konsens von zufälligen Stichproben“ übersetzt werden kann. Für das Verfahren muss zuerst ein Modell aufgestellt werden, das die gemessenen Daten beschreiben kann. Anschließend werden zufällig so viele Messwerten gewählt wie nötig sind, um das Modell zu bestimmen. Mit diesen Messwerten werden die Modellparameter berechnet und dann bestimmt, welche der restlichen Messwerte abgesehen von einer gewissen Toleranz in dieses Modell passen. Dabei wird gezählt wie viele Messwerte dieses Modell unterstützen, also Unterstützer (engl.: *supporter*), dieser These sind. Die Schritte ab dem zufälligen Wählen der Messwerte werden noch mehrmals wiederholt. Das Verfahren endet nach einer vorher festgelegten Anzahl von Iterationen oder wenn eine ausreichend große Unterstützermenge gefunden wurde. Die Menge, welche die meisten Unterstützer enthält, wird in der Literatur auch größte Konsensmenge (engl.: *consensus set*) genannt. Alle Messwerte, die nicht Element dieser Konsensmenge sind, gelten als Ausreißer.

Für den RANSAC Algorithmus ist es notwendig, dass die Daten einem Modell folgen. Dies ist aber bei beliebigen Bewegungen der Benutzer nicht bekannt, wes-


 Abbildung 4.6: Synchronisierung an gemeinsamen Stützstellen  $T = T^A \cup T^B$ 

wegen die Differenz der beiden Sensorsignale betrachtet wird. Möglich macht dies die Tatsache, dass beide Sensoren, wie bereits beschrieben, registriert sind, also Messwerte bezüglich des gleichen Bezugssystems liefern. Seien nun  $T^A$  und  $T^B$  die Zeitstempel der Sensoren A und B und  $T = T^A \cup T^B$  die Vereinigung dieser Zeitstempel. Weiterhin seien  $s^A(t)$  und  $s^B(t)$  die linearen Interpolationsfunktionen von den eindimensionalen Signalen an den Stellen  $T$  (vgl. Abbildung 4.6).

Die Ausreißer beider Sensoren können dann mit Hilfe der Differenz

$$e(t) = \|s^A(t) - s^B(t)\| \quad \forall t \in T = T^A \cup T^B \quad (4.1)$$

bestimmt werden. Denn sind beide Sensoren perfekt zueinander registriert, werden beide Sensoren bis auf die Differenz die selben Messwerte liefern. Die Differenz  $e(t)$  wird dann dominiert sein vom Rauschen der beiden Sensoren und der Erwartungswert von  $e(t)$  wird gleich null sein. Als Modell für den RANSAC Algorithmus kann daher eine einfache Gerade gewählt werden. Die Ausreißer werden dann durch Abweichungen von diesem Modell zu erkennen sein. Für das oben eingeführte Beispiel ist diese Differenz  $e(t)$  in Abbildung 4.7 zu sehen. Es ist zu erkennen wie die Mehrzahl der Daten eine relativ kleine Standardabweichung haben und nur die Ausreißer sich nicht an diesen Konsens halten. Ohne die Ausreißer beträgt die Standardabweichung nur  $\sigma = 13,27$  mm und der Mittelwert  $\mu = 16$  mm. Auch bei dieser Methode verschwinden die Ausreißer, während ein Signal gegen das andere verschoben wird. Die Ursache ist hier ebenfalls die Synchronisierung und dadurch bedingte Interpolation an diskreten Stützstellen.

Dieses Verfahren ist auch anwendbar, wenn bei beiden Sensoren jeweils ein Ausreißer zur gleichen Zeit auftritt. Denn entweder sind diese verschieden, dann ist dies in der Differenz zu beobachten oder beide verhalten sich identisch, dann haben diese aber auch keine Auswirkung auf die Zeitkalibrierung. Die Berechnung des Fehlerterms in Formel 4.1 muss in einem gesonderten Schritt einmal für jedes Segment berechnet werden. Nachdem der Fehler  $e(t)$  für die Zeitstempel beider Sensoren berechnet wird, können Ausreißer von beiden Signalen erkannt werden und bestimmt werden.

Die vorgestellten Methoden zeigen, dass Ausreißer effizient gefiltert werden können. Dies ist auch zur Laufzeit möglich, da die Methoden auch auf den relativ kurzen Segmenten operieren können. Bei verrauschten Signalen und für die Fusion

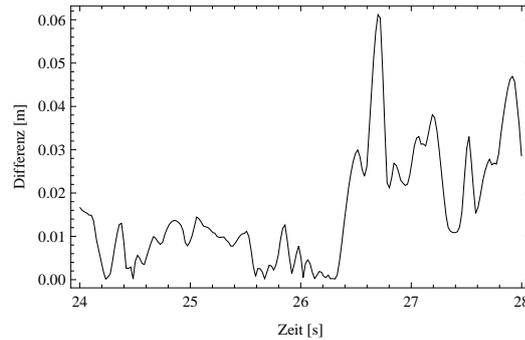


Abbildung 4.7: Differenz der Trackingsignale für die Bestimmung von Ausreißern mit RANSAC.

zweier Sensoren wird oft ein Kalman-Filter genutzt. Der Einsatz eines Kalman-Filters alleine kann keine Ausreißer aus den Datenstrom filtern. Denn besteht zwischen der Prädiktion und der Messung eine durch einen Ausreißer bedingte große Abweichung, dann wird der Kalman Filter die Vorhersage mit dem Ausreißer korrigieren. Diese Korrektur führt dann zu einem unrealistischen Kompromiss zwischen der ursprünglichen Annahme und den beobachteten Messergebnissen [68, 11].

Wie bereits erwähnt, müssen für die Ausreißererkenkung die Sensorsignale schon synchronisiert sein. Nun ist noch zu klären, ob die Filterung vor oder nach der Dimensionsreduktion statt finden soll. Findet die Dimensionsreduktion nach der Ausreißererkenkung statt, bleibt für die Filterung der geometrische Kontext erhalten, aber im Gegensatz dazu steigt der Rechenaufwand für die von der Synchronisierung benötigte Interpolation für mehrdimensionale Daten. Es ist jedoch anzunehmen, dass die Ausreißer ebenso in dem dimensionsreduzierten Signal als solche zu erkennen sind und es somit ausreicht, die Ausreißerfilterung nach der Dimensionsreduktion durchzuführen.

In der Abbildung 4.8 ist der Ablauf einer Filterung mit RANSAC dargestellt. Die Filterung erfolgt im dem RANSAC Block. Dort wird erst die Synchronisierung auf die Zeitstempel beider Sensoren durchgeführt und die Norm der Differenz der beiden eindimensionalen synchronisierten Signale berechnet. Auf dieser Differenz wird dann die RANSAC Methode ausgeführt.

## 4.4 Pathologische Fälle

Die relative Latenz zwischen zwei Sensoren verursacht nur in dynamischen Szenarien Fehler. Daher liegt es nahe, dass die Kalibrierung der relativen Latenz nur dann funktionieren kann, wenn ein dynamisches System vorliegt. Bewegen sich also die Sensoren nicht, dann ist auch keine Kalibrierung möglich. Dieser Stillstand wird im Folgenden als der *triviale pathologische Fall* bezeichnet. Auch bei linearen Bewegungen mit konstanter Geschwindigkeit kann keine Zeitkalibrierung erfolgen, da in diesem Fall die durch die Dynamik ausgelösten Fehler nicht von den Fehler

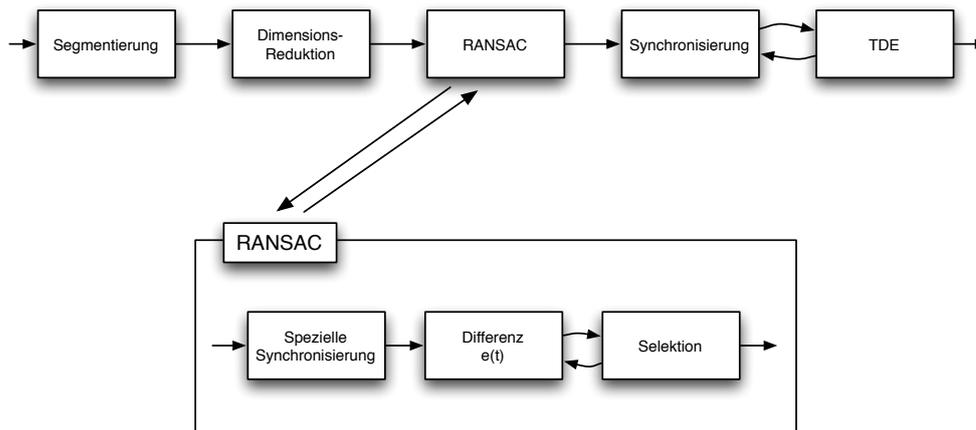


Abbildung 4.8: Vorgeschlagener Ablauf für eine Filterung mit RANSAC.

ausgelöst von einer schlechten Kalibrierung unterschieden werden können. Jedoch ist es unwahrscheinlich, dass der Nutzer eine perfekte lineare Bewegung mit konstanter Geschwindigkeit durchführen kann. Denn wenn in einem Segment entweder eine Abweichung von der linearen Bewegung auftritt oder die Geschwindigkeit variiert wird, ist eine Zeitkalibrierung wieder möglich.

Von diesen Fällen abgesehen gibt es noch weitere pathologische Fälle, die aber in ihrer Art und Weise von den genutzten Dimensionsreduktionsverfahren abhängen. Auf diese Ursachen wird im Folgenden kurz eingegangen. Später wird am Beispiel eines trivialen pathologischen Falls gezeigt, dass während einer pathologischen Bewegung sehr wohl eine Kalibrierung der relativen Latenz möglich ist.

Im Rahmen der Aufbereitung der Trackingdaten für die Zeitkalibrierung, müssen die Daten in der Dimension reduziert werden. Verschiedene Varianten der Dimensionsreduktion sind im Kapitel 5 ausführlich beschrieben. Durch die Dimensionsreduktion entstehen jedoch neben den trivialen noch weitere pathologische Situationen. Wird beispielsweise eine Projektion der Position auf die  $x$ -Achse gewählt und die beiden Sensoren anschließend nur in der  $y - z$  Ebene bewegt, wird das Bild dieser Projektion konstant bleiben. Wie für das eben beschriebene Beispiel der Dimensionsreduktion kann für jedes andere Verfahren mindestens ein weiterer pathologischer Fall gefunden werden. Diese hängen neben der Art des Verfahrens auch von den konkret gewählten Parametern des Verfahrens ab.

Werden diese pathologischen Bewegungen von einem Menschen ausgeführt, so ist bei allen Bewegungen ein gewisses Maß an Ungenauigkeit enthalten. Diese Imperfektion des Menschen verhindert die pathologischen Eigenschaften der Bewegungen. Dies soll am Beispiel des trivialen pathologischen Falls gezeigt werden, dem Stillstand. Um einen Gegenstand, wie beispielsweise zwei Sensoren, stillzuhalten, arbeiten die Muskeln wie ein Regelkreis [66]. Muskeln sind kontraktile Organe, welche durch Kontraktion und Erschlaffen Bewegungen erzeugen. Um beispielsweise eine Hand stillzuhalten, müssen die Muskeln wiederholt kontrahieren und erschlaffen. Da dieser Regelkreis nicht perfekt ist, ergibt sich ein leichtes Zittern,

das bei gesunden Menschen aber nicht sichtbar ist. Von Lippold [66] und anderen Forschern wurde für dieses Zittern eine Frequenz von 8 bis 10 Hz gemessen. Dieses Zittern wird in der Literatur auch Haltetremor genannt. Um zu zeigen, dass auch in pathologischen Fällen eine korrekte Kalibrierung der relativen Latenz möglich ist, wurden zwei Versuchspersonen gebeten eine Kombination von Sensoren möglichst ruhig zu halten.

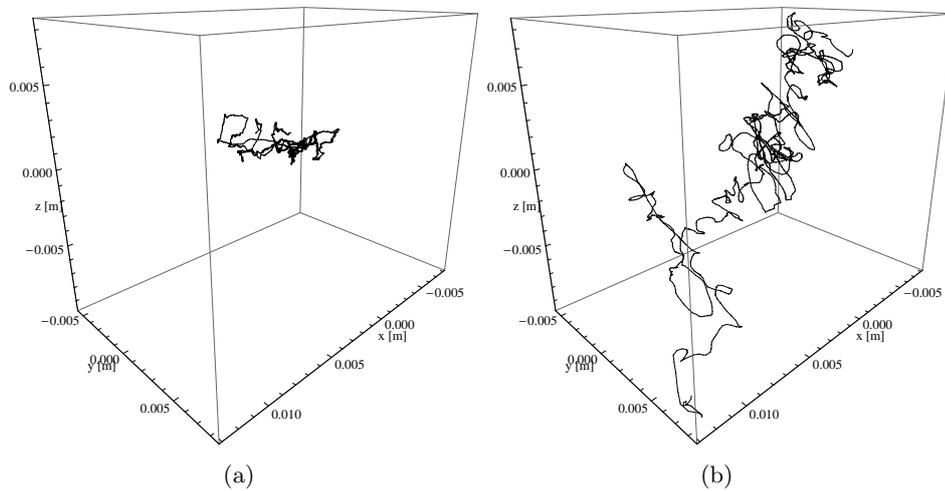


Abbildung 4.9: Darstellung des Haltetremors der Hand; Aufgezeichnet mit einem FaroArm mit montierter A.R.T. Kugel; Beide Pfade wurden in den Ursprung verschoben; (a) Handtremor Proband A; (b) Handtremor Proband B

In Abbildung 4.9 sind die Pfade der beiden Probanden zu sehen. Es ist in der Abbildung deutlich zu erkennen, dass Proband A und Proband B sehr unterschiedliche Haltetremores haben. Für die Versuche wurden 23s Daten bei Proband A und 27s bei Proband B mit einem A.R.T. Trackingsystem und einem FaroArm aufgezeichnet. Auf die Stärke des Tremores deutet die Geschwindigkeit der Bewegung hin. Bei Proband A war dies  $v_A = 3,61 \cdot 10^{-3} \text{ms}^{-1}$  und bei Proband B  $v_B = 10,7 \cdot 10^{-3} \text{ms}^{-1}$ . Für die in Abbildung 4.9 zu sehenden Pfade wurde jeweils eine Zeitkalibrierung durchgeführt mit einer Fensterbreite von 4 s, einer Überlappung von 1 s, der Norm als Dimensionsreduktion und der Korrelation als Vergleichsfunktion. Dabei wurde mit der Bewegung von Proband A im Durchschnitt ein zeitlicher Versatz von  $\Delta t_A = 32,1 \text{ms}$  mit einer Standardabweichung  $\sigma_A = 3,5 \text{ms}$  bestimmt. Die Kalibrierung mit der Bewegung des Probanden B ergab einen durchschnittlichen Versatz von  $\Delta t_B = 31,5 \text{ms}$  mit einer Standardabweichung von  $\sigma_B = 1,9 \text{ms}$ .

Diese Werte zeigen, dass sich die beiden Schätzungen der relativen Latenz nicht signifikant unterscheiden. Bei Proband B wurde der zeitliche Versatz mit einer kleineren Standardabweichung geschätzt. Es ist anzunehmen, dass dies an der höheren Geschwindigkeit von Proband B liegt. Diese Hypothese wird von einem weiteren Versuch unterstützt. Bei diesem Versuch wurde das Sensorpaar auf einer Kreisbahn

24 s lang mit einem ungefähren Durchmesser von 0,8 m und mit einer Geschwindigkeit von  $0,91 \text{ ms}^{-1}$  bewegt. Die relative Latenz wurde anschließend mit den gleichen Parametern wie bei den oben aufgeführten Tremorversuchen bestimmt. Dabei ergab sich für die relative Latenz ein Wert von  $\Delta t_K = 32,0 \text{ ms}$  mit einer Standardabweichung von  $\sigma_K = 0,10 \text{ ms}$ .

Werden nun die beiden Sensoren in ein Stativ eingespannt, dann sind sämtliche Bewegungen ausgeschlossen und es bleibt nur noch das unkorrelierte Sensorrauschen übrig. Ein unkorreliertes Rauschen erlaubt keine Kalibrierung der beiden Sensoren. Dies wird von einem weiteren Versuch bestätigt. Bei einer Versuchsdauer von 47 s wurde die Kombination mit einer Durchschnittsgeschwindigkeit von  $v_S = 0,16 * 10^{-3} \text{ ms}^{-1}$  bewegt. Bei dieser geringen aber nur scheinbaren Bewegung von rund 7,52 mm war keine Kalibrierung möglich.

Somit bleibt festzuhalten, dass selbst wenn der Benutzer versucht die Sensoren absolut still zu halten, dennoch eine Kalibrierung möglich ist. Es darf angenommen werden, dass bei anderen vermeintlichen pathologischen Fällen das Ergebnis sogar noch besser sein wird. Wenn beispielsweise die oben beschriebene Projektion auf die  $x$  Achse verwendet wird. Dann ist es unwahrscheinlich, dass er Benutzer sich exakt in der  $y - z$  Ebene bewegen kann oder wird. Es ist dem menschlichen Benutzer eines AR Systems somit nicht möglich, einen solchen pathologischen Fall herbei zuführen, in welchem keine Kalibrierung mehr möglich ist. Wohl wird die Qualität der Kalibrierung gemindert, da weniger Information, oder wie in diesem Kapitel auch beschrieben, weniger Signal im Verhältnis zu Rauschen vorhanden ist. Dies gilt natürlich auch wenn die Sensoren am Körper fixiert sind. Sind die Sensoren hingegen technisch an einem Tisch oder anderen statischen Objekten fixiert ist keine Kalibrierung möglich. In einem automatisch arbeitendem System kann dieser Fall einfach erkannt werden. So kann auf eine Kalibrierung der relativen Latenz verzichtet werden, wenn die Geschwindigkeit unter einem Schwellwert bleibt. Ändert sich also während einer solchen Ruhephase die relative Latenz, so kann dies nicht detektiert werden. In diesem Fall ist der durch die geänderte relative Latenz verursachte Fehler zu vernachlässigen, da diese Fehler nur in dynamischen Systemen auftreten. Somit ist eine Kalibrierung der relativen Latenz zweier Sensoren auch während des Betriebs einer Anwendung möglich, ohne dass pathologische Fälle die Kalibrierung beeinträchtigen. Eine Kalibrierung ist nicht möglich ist, wenn das System sich im Zustand des absoluten Stillstands befinden.

## 4.5 Rauschen

Neben den schon beschriebenen Qualitätsmerkmalen wie Registrierungsfehler oder Ausreißer beeinflusst auch Rauschen die Qualität einer Zeitkalibrierung negativ. Das Rauschen eines Signal muss aber im Verhältnis zum Signal selbst gesehen werden. Genau diese Verhältnis von Rauschen zu einem Signal selber wird als Signal-Rausch-Verhältnis (engl.: *signal-to-noise ratio*, *SNR*) bezeichnet. Es wird vornehmlich in der Nachrichtentechnik als Maßzahl für die Qualität eines Signals

verwendet, findet aber auch in anderen Bereichen Anwendung.

Das Rauschen von Sensoren kann als unkorreliert betrachtet werden. Tritt bei einem Sensor das Signal in den Hintergrund und das Rauschen in den Vordergrund, dann kann das Signal dieses Sensors nicht mehr klar mit dem Signal des anderen Sensor verglichen werden. Es ist leicht einzusehen, dass die Qualität der Kalibrierung unter starkem Rauschen in einem oder beiden Sensoren gemindert wird. Neben einem großen Rauschen kann auch eine niedrige Signalleistung der SNR Wert verschlechtern. Diese Situation tritt ein, wenn die Sensoren wenig oder gar nicht bewegt werden. Damit wird das Nutzsinal klein im Verhältnis zum Rauschen der Sensoren und führt somit auch zu einem niedrigeren SNR Wert. In diese Klasse fallen auch die in Abschnitt 4.4 beschriebenen pathologischen Fälle.

Um den Einfluss des Rauschens auf die zeitliche Kalibrierung zu untersuchen, wird in den folgenden Abschnitten das Signal-Rausch-Verhältnis eingeführt. Anschließend werden verschiedene Methoden aufgeführt, um das Rauschen zu verringern und somit zu einem besseren Verhältnis von Nutzsinal zum Rauschen zu kommen. Zuletzt wird das Signal-Rausch-Verhältnis mit Hilfe eines Tiefpassfilters verbessert und die Auswirkungen auf die Kalibrierung untersucht.

**Definition Signal-Rausch-Verhältnis** In der Literatur gibt es verschiedene Definitionen des Signal-Rausch-Verhältnisses. Eine gängige Definition legt den SNR Wert durch das Verhältnis der mittleren Signalleistung und der mittleren Rauschleistung fest

$$\text{SNR} = \frac{P_S}{P_N} = \left( \frac{A_S}{A_N} \right)^2. \quad (4.2)$$

Dabei kann die Leistung auch durch das Quadrat der RMS Amplitude ausgedrückt werden kann, wobei der RMS als die Wurzel der normierten Summe der Quadrate definiert ist

$$\text{RMS} = \sqrt{\frac{1}{N}(x_1^2 + x_n^2 + \dots + x_N^2)} \quad [111, 91].$$

In der Regel weisen die Signale einen sehr großen Dynamikumfang auf. Die Signalleistung ist also um Größenordnungen größer als die des Rauschens. Deswegen wird auch häufig eine logarithmische Definition des SNR angewandt

$$\text{SNR}_{dB} = 10 \lg \left( \frac{P_S}{P_N} \right) = 10 \lg \left( \frac{A_S}{A_N} \right)^2. \quad (4.3)$$

Der SNR Wert kann als Qualitätszahl für Signale für die zeitliche Kalibrierung benutzt werden. Dabei sind ein paar Besonderheiten zu beachten. Wird beispielsweise die normalisierte Kreuzkorrelation als Vergleichsmaß benutzt ist der Mittelwert der Signale nicht von Bedeutung. Denn die Korrelation ist invariant gegenüber einer Translation und Skalierung der Signale (siehe Abschnitt 7.3.1). Somit ist bei Nutzung der Korrelation nur die Standardabweichung des Signals und die Standardabweichung des Rauschens ausschlag gebend [13].

Als Beispiel für ein Signal-Rausch-Verhältnis ist in Abbildung 4.10 zu sehen. Sei  $s(t) = \sin(t)$  ein Sinussignal mit  $t \in \{0, 5\pi\}$  (vgl. Abb. 4.10(a)) und  $n(t)$  beschreibt ein gaußsches Rauschen mit  $\sigma = 0.1$  wie in Abb. 4.10(b) zu sehen. Addiert man beide Signale zu  $x(t) = s(t) + n(t)$  so ergibt sich ein Signal wie in Abbildung 4.10(c) zu sehen. Mit der Definition 4.3 kann dann das Signal-Rausch-Verhältnis bestimmt werden. Wird der RMS Wert für  $s(t)$  mit  $\text{RMS}_s = 0.707061$  berechnet und der RMS Wert für  $n(t)$  mit  $\text{RMS}_n = 0.0999655$  bestimmt, dann ergibt sich ein SNR von 16.9921 für  $x(t)$ .

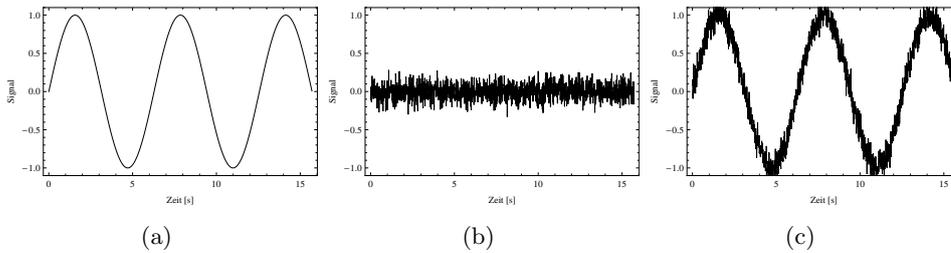


Abbildung 4.10: (a) Signal  $s(t) = \sin(t)$  ohne Rauschen; (b) Rauschen  $n(t)$  mit  $\mu = 0$  und  $\sigma = 0.1$ ; (c) Verrauschtes Signal  $x(t) = s(t) + n(t)$  und ein berechneter SNR Wert von 17 dB

Diese Bestimmung des Signal-Rausch-Verhältnisses ist wichtig für die später eingeführte Monte-Carlo-Simulation. Denn in einer Simulationsumgebung sind alle Parameter bekannt, um die genauen SNR Werte der Signale zu kontrollieren und mit den gewonnen Zeitkalibrierungen in Bezug zu setzen.

In [18] wird ein Maximum-Likelihood Schätzer für den Zeitversatz zwischen zwei Signalen bei unkorreliertem Rauschen entwickelt. Dabei wird, wie in dieser Arbeit vorgeschlagen, ein Signal gegen das andere verschoben und dabei fortlaufend die Korrelation zwischen den Signalen bestimmt. Bei einem maximalen Wert der Korrelation ist der wahrscheinlichste Wert des Zeitversatz gefunden. Vor der Kreuzkorrelation werden die beiden Signale gefiltert, in dem die Signale mit Hilfe des Wertes der Kohärenz gewichtet werden. Diese Methode, also eine Präfilterung der Signale mit anschließender Kreuzkorrelation, wird auch Generalisierte Kreuzkorrelationsmethode (engl.: *generalized cross correlation method, GCC*) genannt. Die Arbeiten, die in [18] entwickelt wurden, wurden von Carter weiterentwickelt und in [19] präsentiert.

**Evaluation** Es existiert eine Vielzahl von Methoden, um das Rauschen eines Signals zu reduzieren. Eine Möglichkeit ist die eben erwähnte GCC Methode oder der Einsatz eines Kalman Filters. Ein Ansatz aus der Signalverarbeitung ist die Verstärkung eines Signals vor einer Übertragung. So gewinnt man beim Empfänger ein besseres Signal zu Rauschen Verhältnis. Dieses Prinzip würde in einer AR Anwendung schnellen und ruckartigen Bewegungen der Sensoren entsprechen. Auf der anderen Seite wird dies weitere Fehler auslösen. So reagieren optische kameraba-

sierte Systeme mit größeren Ungenauigkeiten oder auch Ausreißern auf schnelle Bewegungen. Die Methode ist also nur in Grenzen anwendbar. Von Signalverarbeitung inspiriert ist das Filtern eines Signals zu optimieren. Wenn man mit geeigneten Filtern das Rauschen filtern kann, dann würde dies auch das Verhältnis von Signal und Rauschen verbessern. Eine Filterung ist möglich, da die Rauschanteile meist höher-frequente Anteile sind, während die typischen Bewegungen des Nutzers eines AR-Systems zwischen 2 Hz und 20 Hz liegen [114]. Mit einem Tiefpassfilter lässt sich dann das Rauschen reduzieren. Im Bereich der Augmented Reality haben sich Kalman-Filter und Partikel-Filter etabliert, da beide Methoden auf Grund ihrer mathematischen Struktur Signale in Echtzeit filtern können (siehe Abschnitt 2.5.2). Und Augmented Reality Anwendungen müssen die Realität in Echtzeit erweitern können [3, 2, 102].

Für den Zweck der Voruntersuchung wurde das gleitende Mittel genutzt, um das Signal-Rausch-Verhältnis zu erhöhen. Die Methode des gleitenden Mittels zu Erhöhung des Signal-Rausch-Verhältnisses einzusetzen ist ein übliches Verfahren, wenn keine Echtzeitfähigkeit gefordert wird. Im Gegensatz zu den in der Literatur erzielten Ergebnissen [18, 19] wurden in dieser Voruntersuchung aber keine signifikanten Verbesserungen durch das gleitende Mittel festgestellt.

Um Trackingdaten mit Hilfe des gleitenden Mittels zu filtern, bedarf es einiger Vorbereitungen. Zunächst sei das gleitende Mittel mit einem Fenster von ungerader Länge  $n$  definiert durch

$$\bar{x}_{\frac{n+1}{2}} = \frac{x_1 + x_2 + \dots + x_{\frac{n+1}{2}} + \dots + x_{n-1} + x_n}{n} \quad (4.4)$$

$$\bar{x}_{\frac{n+1}{2}+1} = \frac{x_2 + x_3 + \dots + x_{\frac{n+1}{2}+1} + \dots + x_n + x_{n+1}}{n}. \quad (4.5)$$

In diesem Falle wurden nur Fenster mit ungerader Länge benutzt, damit die Zuordnung der Zeitstempel weiterhin gewährleistet ist. Am linken und rechten Rand wurden jeweils  $\lfloor n/2 \rfloor$  Werte verworfen, damit das Ergebnis nicht durch das Auffüllen von Werten verfälscht wird.

Die Daten können im Allgemeinen nicht direkt gefiltert werden, sondern müssen erst aufbereitet werden. Dazu werden die Signale zunächst mit einer Dimensionsreduktion auf eine Dimension reduziert. Dies führt zu einem geringeren Rechenaufwand, als die  $n$ -dimensionalen Daten zu filtern. Die zwei Signale liegen nun diskret in der Form  $s_i^A \in \mathbb{R}$  und  $s_i^B \in \mathbb{R}$  mit unterschiedlichen Update-Raten vor. Sie müssen daher vor einer Filterung mit dem gleitenden Mittel noch synchronisiert werden. Durch die unterschiedlichen Update-Raten würde eine konstante Fensterlänge bezüglich der Messwerte zu unterschiedlich langen Fenstern bezüglich der Zeit resultieren. Für die Synchronisierung wird eines der beiden Signale linear interpoliert, und somit kann zu den Zeitstempeln des anderen Signals ein Wert bestimmt werden.

Die beiden synchronisierten Signale können nun mit dem gleitenden Mittel Tiefpass gefiltert werden. Die gefilterten Daten durchlaufen dann die weiteren Schritte

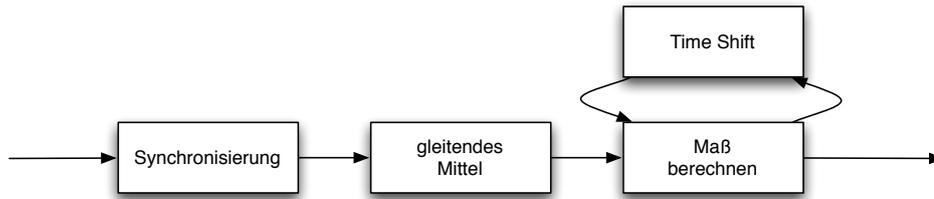


Abbildung 4.11: Blockschaubild des Versuchs die Genauigkeit der Schätzung durch Filterung zu steigern

der zeitlichen Kalibrierung, wobei darauf hingewiesen sei, dass bei diesem Verfahren ein Signal zweimal interpoliert wird. Wie groß die hieraus zu erwartenden Ungenauigkeiten sind wurde nicht untersucht. In Abbildung 4.11 ist ein Ablaufdiagramm dieses Vorgangs dargestellt.

Für die folgenden Versuche wurde eine zeitliche Kalibrierung zwischen einem Markertracker und einem A.R.T. System vorgenommen. Beide Systeme sind physikalisch fest miteinander verbunden und registriert worden, liefern also Positionsdaten bezüglich des selben Referenzsystems. Für den Versuch wurden 32 s Daten aufgezeichnet. Eine Darstellung der aufgezeichneten Bewegung kann in Abbildung 4.12 betrachtet werden. Die relativen Latenzen wurden mit Hilfe der linearen Interpolation, der Norm als Dimensionsreduktionsfunktion und der normierten Kreuzkorrelation als Maß bestimmt. Die Kreuzkorrelation wurde jeweils über einen Zeitraum von 4 s berechnet und mit einem Überlappungsabstand von 1 s. Die Registrierung der Daten ist in diesem Falle relativ ungenau. Der Fehler, bestimmt durch den geometrischen Abstand der Messwerte, liegt gemittelt über den gesamten Versuchszeitraum bei 26,0 mm.

In der Tabelle 4.1 sind die Durchschnittswerte, der Median und die Standardabweichungen der Schätzungen aufgeführt. Es ist deutlich zu sehen, dass mit wachsender Fenstergröße auch die Standardabweichung wächst. Die Schätzung der relativen Latenz wird also mit wachsender Fenstergröße ungenauer. Mit wachsender Fenstergröße wuchs dabei auch die Standardabweichung. Dieser Effekt konnte auch bei anderen Messreihen mit anderen Sensoren beobachtet werden. Dies zeigt, dass eine Filterung mit dem gleitenden Mittel nicht ausreicht, um die Genauigkeit der Kalibrierung zu erhöhen.

Fensterbreite	1	3	5	7	9	11
Mittelwert	-60 ms	-60 ms	-61 ms	-62 ms	-62 ms	-63 ms
Standardabw.	11,8 ms	12,5 ms	13,5 ms	14,9 ms	15,3 ms	15,5 ms

Tabelle 4.1: Abhängigkeit des Mittelwerts und der Standardabweichung der Zeitkalibrierung von der Fenstergröße des gleitenden Mittels. Die Fenstergröße ist dabei in Anzahl der Messwerten angegeben.

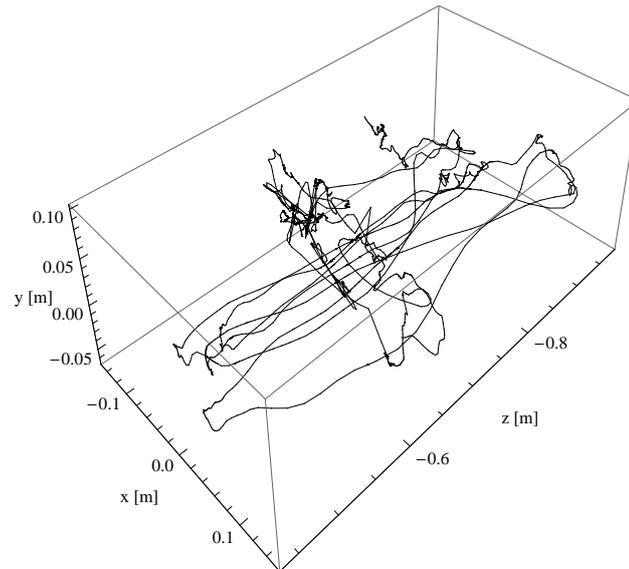


Abbildung 4.12: Aufgezeichneter Pfad für die Untersuchung des Filterns

Als erster weiterer Schritt sollte der Einfluss der Kalman Filters auf die Resultate der zeitlichen Kalibrierung untersucht werden. Aber auch die, in den Arbeiten von Carter und Knapp in [18, 19] vorgestellten, Methoden zur Bestimmung einer Maximum-Likelihood Filterung sollten auf das Gebiet des Trackings für Augmented Reality adaptiert und untersucht werden.

## 4.6 Zusammenfassung

In diesem Kapitel wurden eine Reihe von Einflüssen beschrieben, die die Genauigkeit einer zeitlichen Kalibrierung maßgeblich beeinflussen. Es wurden nicht alle möglichen Fehlerquellen aufgezeigt, wobei die größten Einflüsse beschrieben wurden. Andere mögliche Quellen wie Quarzungenauigkeiten in den Sensoren oder in den verwendeten Rechnern wurden nicht untersucht und können Gegenstand weiterer Untersuchungen sein. Dabei wurden die genauen Ursachen dieser Fehlerquellen aufgezeigt und anhand von Messdaten die resultierenden Ungenauigkeiten demonstriert. Für manche Einflüsse wurde weiterer Forschungsbedarf aufgezeigt, während für andere konkrete Handlungsanweisungen gegeben wurden, wie negative Einflüsse auf die Kalibrierung verhindert werden konnten. Für die Gruppe der pathologischen Fälle konnte gezeigt werden, dass diese in der Praxis nicht von Bedeutung sind, außer wenn die Sensoren physikalisch fixiert werden.



## 5 Dimensionsreduktion

Die in dieser Arbeit vorgeschlagene Methode zur Kalibrierung der zeitlichen Abstände, fordert als Eingabeparameter eindimensionale Daten. Da aber bei Augmented Reality in der Regel drei- oder sechsdimensionale Daten in Form von Positions- und Orientierungsdaten anfallen, müssen diese auf eine Dimension reduziert werden. Nachdem eine solche Reduktion auch mit Verlust an Information verbunden ist, kann nur versucht werden, diesen Verlust so gering wie möglich zu halten. In diesem Kapitel werden Methoden aufgezeigt, mit welchen sich die mehrdimensionalen geometrischen Signale sinnvoll auf eine einzelne Dimension reduzieren lassen. Es wird dabei zwischen statischen und adaptiven Verfahren unterschieden. Bei statischen werden die Parameter des Verfahren zu Beginn festgelegt und dann konstant gehalten. Dem entgegen wird bei den adaptiven Verfahren versucht, durch definierte Regelkreise die Verfahren so zu steuern, dass möglichst viel Information in den Signalen verbleibt.

### 5.1 Einführung

Die Kreuzkorrelation, welche in dieser Arbeit als Mittel für die Kalibrierung der relativen Latenz vorgeschlagen wird, benötigt als Eingabeparameter eindimensionale Daten. Also müssen die Messdaten bestehend aus Positionen oder Orientierungen auf eine einzelne Dimension reduziert werden. Im Folgenden werden verschiedene Verfahren vorgestellt, wie die Anzahl der Dimensionen reduzieren werden kann.

Für die Reduktion der Dimensionen kann eine Vielzahl von Verfahren konstruiert werden, die das Ergebnis der späteren Kalibrierung nicht negativ beeinflussen dürfen. Da mit der Reduktion der Dimension unweigerlich auch ein Informationsverlust eintritt, tritt dabei stets auch ein Fehler auf. Aber man kann das Verfahren so wählen, dass der Fehler minimal bleibt.

Prinzipiell wird zwischen adaptiven Verfahren und statischen Verfahren unterschieden. Bei statischen Verfahren werden während der gesamten Laufzeit die Parameter nicht verändert und können somit auch nicht auf sich verändernde Rahmenbedingungen reagieren. Adaptive Systeme passen hingegen die Parameter der jeweiligen Dimensionsreduktion an die gerade vom Benutzer durchgeführten Bewegungen an. Dadurch generieren sie eine bessere Datengrundlage für die Schätzung der relativen Latenz. Beispiele für adaptive Verfahren sind die kanonische Korrelation oder auch eine randomisierte Variante der Projektionsmethode.

Neben der grundsätzlichen Art der Dimensionsreduktion ist natürlich auf die Eigenschaften der Daten einzugehen. Position und Orientierung unterscheiden sich

sowohl in der Struktur als auch in der geometrischen Interpretation grundsätzlich von einander. Während bei Positionen die Norm als Reduktionsverfahren durchaus einen geometrischen Sinn hat, fällt eine Interpretation einer normierten Orientierung schwer. Zudem gibt es, wie in Abschnitt 2.3.1 gezeigt, eine Vielzahl an Darstellungen für die Orientierung, beispielsweise die in der Computergrafik und Augmented Reality übliche Darstellung durch Quaternionen. Da diese Quaternionen meist normiert sind, macht die Norm als Reduktionsverfahren gar keinen Sinn. Es müssen also für die Position und die Orientierung verschiedene Reduktionsverfahren entwickelt werden.

Neben absoluter Orientierung  $r$  und Position  $p$  müssen auch noch die Ableitungen dieser Datentypen betrachtet werden, denn diese Datentypen werden von inertialen Sensortypen geliefert. Im Falle von Orientierungen sind das Rotationsgeschwindigkeiten  $\dot{r}$ , die Geschwindigkeiten  $\dot{p}$  oder Beschleunigungen  $\ddot{p}$ . Diese physikalischen Messgrößen können von absoluten Typen durch Differenzierung gewonnen werden, wobei dieser Vorgang Signale mit sehr hohem Rauschanteil produziert. Aber manche Sensoren, etwa inertielle Sensoren messen diese physikalischen Größe direkt. Beispielsweise messen Gyroskope die Rotationsgeschwindigkeit  $\dot{r}$  und Beschleunigungssensoren (engl.: *accelerometer*), wie der Name andeutet, die Beschleunigungen  $\ddot{p}$ . Auch auf diese physikalischen Messgrößen muss gesondert eingegangen werden, um eine optimale zeitliche Kalibrierung zu gewähren.

## 5.2 Statische Verfahren

Bei statischen Dimensionsreduktionsverfahren werden die Parameter der Reduktion a priori festgelegt, und sie werden während der Laufzeit der Anwendung nicht mehr geändert. Um die Parameter für ein Verfahren sinnvoll wählen zu können, müssen ein paar Grundkenntnisse über die zukünftige Anwendung und die verwendeten Trackingtechnologien vorliegen. Angenommen in einer Anwendung wird ein Fahrzeug getrackt, welches sich nur in einer Ebene bewegen wird und es wird beispielsweise die Höhe über Grund für eine Kalibrierung verwendet. Diese Signale von beiden Sensoren werden einen niedrigen SNR Wert aufweisen, da sich die Höhe über Grund nur geringfügig ändert. Eine Kalibrierung des zeitlichen Offsets mit diesen Signalen wird dadurch fehleranfälliger. Im Folgenden werden einige statische Verfahren vorgestellt, wobei diese Aufzählung keinen Anspruch auf Vollständigkeit erhebt.

### 5.2.1 Positionen

Für Positionen kann aus einer Reihe von Verfahren gewählt werden. Von diesen zahlreichen werden nun einige wichtige Repräsentanten vorgestellt. Im Folgenden sei  $s_i$  eine Reihe von Sensormessdaten und  $S_i = (x_i, y_i, z_i)$  ein Messwert, der eine Position im dreidimensionalen Raum beschreibt.

**Projektion** Eine Projektion ist eine lineare Abbildung eines Vektorraums in den gleichen Vektorraum mit gleicher oder niedrigerer Dimension. Mathematisch kann diese Abbildung wie folgt ausgedrückt werden

$$P : \mathbb{R}^n \rightarrow \mathbb{R}^m, \quad m \leq n, \quad (5.1)$$

wenn  $P$  eine lineare Abbildung ist. In dem Fall der Dimensionsreduktion einer Position ist  $n = 3$  und  $m = 1$ . Die Projektion kann mit einem Skalarprodukt dargestellt werden. Sei  $S_i \in \mathbb{R}^n$  eine Reihe von Sensormessdaten beliebigen Ursprungs und beliebiger Art und  $p \in \mathbb{R}^n$  ein Projektionsvektor, dann ist die Projektion der  $S_i$  durch  $p$  definiert durch

$$S_i^T p \in \mathbb{R}. \quad (5.2)$$

Neben der Definition einer Projektion ist noch zu entscheiden, welcher Projektionsvektor für eine stabile und korrekte Kalibrierung gewählt werden kann. In einigen Vorversuchen hat sich gezeigt, dass selbst die willkürliche Wahl einer Achse eine stabile Kalibrierung zulässt. Dies steht in einem engen Zusammenhang mit den in Abschnitt 4.4 beschriebenen pathologischen Fällen.

**Norm** Eine Norm ist definiert als eine sublineare Abbildung von einem Vektorraum in den zugrunde liegenden Körper und ist die Verallgemeinerung des geometrischen Begriffs der Länge

$$N : \mathbb{R}^n \rightarrow \mathbb{R}. \quad (5.3)$$

Den intuitiven Begriff der Länge stellt die Euklidische Norm dar und ist wie folgt definiert

$$\|S_i\| = \sqrt{x_i^2 + y_i^2 + z_i^2}. \quad (5.4)$$

Damit gibt diese Dimensionsreduktion den Abstand zum Ursprung des Bezugssystems an. Die pathologischen Bewegungen sind in diesem Fall, wenn sich mindestens einer der beiden Sensoren auf einer Kreisbahn um den Ursprung bewegt. Darin enthalten ist auch der Fall, dass ein Sensor konstant im Ursprung verharrt. Die Methode der Norm ist einfach und berücksichtigt alle drei Freiheitsgrade der Position gleichermaßen.

### 5.2.2 Ableitungen von Positionen

Neben den Positionen  $S_i$  liefern manche Sensoren auch davon abgeleitete Messwerte, wie Geschwindigkeiten  $\dot{S}_i$  oder auch Beschleunigungen  $\ddot{S}_i$ . Für beide abgeleiteten Datentypen lassen sich dieselben Methoden wie für die Position verwenden. So bezeichnet  $\|\dot{S}_i\|$  die aktuelle Geschwindigkeit mit der ein Sensor bewegt wird.

Ebenso wie für die Positionen müssen hier beide Sensoren zueinander registriert werden.

Auch bei der Nutzung von Geschwindigkeit und Beschleunigung für die zeitliche Kalibrierung kann es zu pathologischen Fällen kommen, die auch unabhängig von der gewählten Dimensionsreduktionsmethode sind. Wird die Kombination der beiden Sensoren um das Zentrum des Koordinatensystems eines Sensors gedreht, so liefert dieser Sensor eine konstante Position und dadurch auch keine Geschwindigkeit und keine Beschleunigung. Ein weiterer pathologischer Fall entsteht, wenn beide Sensoren mit konstanter Geschwindigkeit entlang einer Geraden bewegt werden.

### 5.2.3 Orientierung

Neben den Positionen können auch die Orientierungen für die zeitliche Kalibrierung verwendet werden. Im Folgenden werden Methoden beschrieben, die eine sinnvolle Reduktion der Dimensionen liefern, damit ein Vergleich zweier Signale durchgeführt werden kann.

**Drehwinkel** Eine einfache Methode, die drei Freiheitsgrade einer Orientierung auf einen Freiheitsgrad zu reduzieren, ist es, den Winkel der Achsen-Winkel-Darstellung zu benutzen. Die Achsen-Winkel-Kodierung beschreibt eine Orientierung mit Hilfe einer Rotationsachse und eines Drehwinkels [21].

Auf das gleiche Prinzip lassen sich auch Orientierungen in der Quaternionendarstellung in den Freiheitsgraden reduzieren. Sei  $q = (q_w, q_x, q_y, q_z)$  ein normiertes Quaternion, welches eine Orientierung repräsentiert, mit den imaginären Anteilen  $(q_x, q_y, q_z)$  und dem realen Anteil  $q_w$ . Dann repräsentiert das Tripel  $(q_x, q_y, q_z)$  die Rotationsachse und  $q_w$  steht mit dem Rotationswinkel in direktem Zusammenhang. Der Rotationswinkel kann mit  $\alpha = 2 \cos^{-1} q_w$  berechnet werden [64].

Die Darstellungen von Orientierungen in der Achsen-Winkel-Kodierung und in Quaternionendarstellung sind mehrdeutig. Wird die Rotationsachse negiert, so wird auch der Winkel negiert.  $q_1 = (q_w, q_x, q_y, q_z)$  und  $q_2 = (-q_w, -q_x, -q_y, -q_z)$  beschreiben dann die gleiche Orientierung.

Bei einem Vergleich muss also sicher gestellt werden, dass die Drehachse nicht springt, oder es muss die Norm des Winkels verwendet werden.

**Projektion** Analog zu Positionsdaten können auch Orientierungen projiziert werden. Das Prinzip ist identisch, egal ob es sich um eine Projektion von Quaternionen oder von einer Darstellung mit Euler Winkeln handelt. Es sollte bei beiden Messwerten natürlich die gleiche Darstellung benutzt werden. Im Folgenden wird die Projektion nur für Euler Winkel beschrieben, nachdem die Projektion für die anderen Darstellungen analog durchgeführt werden kann. Sei nun  $r = (\alpha, \beta, \gamma)$  ein Vektor, der eine Orientierung in Euler-Winkel Repräsentation beschreibt. Es ist hier nicht relevant welche der zahlreichen Versionen der Repräsentationen von Euler Winkeln genutzt wird, solange beide Sensoren die gleiche Repräsentationen nutzen. Sei weiterhin  $\mathbf{p} \in \mathbb{R}^3$  ein Vektor, der für die Projektion verwendet wird.

Dann liefert das Skalarprodukt von  $\mathbf{r}^T \mathbf{p}$  eine Projektion in den  $\mathbb{R}$ . Diese Reduktion ist wie bei Positionen nur dann anwendbar, wenn beide Sensoren präzise zu einander registriert sind.

Auch in diesem Fall existieren pathologische Situationen. Diese sind analog zu den Fällen bei der Projektion von Positionen.

**Exponentialabbildung** Neben den schon beschriebenen Quaternionen, Euler Winkeln und Rotationsmatrizen können Orientierungen auch mit Hilfe der Exponentialabbildung dargestellt werden. Es soll hier nicht auf die mathematischen Einzelheiten dieser Darstellung eingegangen werden. Diese sind in den Arbeiten [33, 80] aufgeführt. Demnach kann eine Orientierung mit Hilfe eines Triplets  $v = (r_x, r_x, r_x)$  dargestellt werden. Dabei ist der Vektor  $v$  die Drehachse, und die Länge  $\|v\|$  des Vektors  $v$  gibt den Drehwinkel an. Die Berechnung dieses Triplets kann durch den Logarithmus der Quaternionen Darstellung erfolgen

$$v = \log(q) = \frac{2 \cos^{-1} q_w}{\|q\|} (qx, qy, qz).$$

Die Dimensionsreduktion dieser Darstellung einer Orientierung kann durch die oben beschriebenen Verfahren durchgeführt werden, da sowohl der Drehwinkel als auch die Drehachse in dem Triplet enthalten sind. Wird jedoch die Norm als Dimensionsreduktionsmethode für diese Repräsentation gewählt, ist dies identisch zur Verwendung des Drehwinkels.

#### 5.2.4 Ableitungen von Rotationen

Die Ableitungen von Rotationen  $r$  sind Rotationsgeschwindigkeiten  $\dot{r}$  und Rotationsbeschleunigungen  $\ddot{r}$ . Diese Daten lassen sich analog zu Orientierungen repräsentieren. Pustka gibt in [80] eine ausführliche Übersicht über die möglichen Repräsentationen von Rotationsgeschwindigkeiten und -beschleunigungen. Mit Hilfe dieser Darstellungen können die Dimensionsreduktionsverfahren für Orientierungen analog benutzt werden.

### 5.3 Adaptive Verfahren

Die eben vorgestellten Verfahren waren statisch. Das heißt das Verfahren und die Parameter der Dimensionsreduktion wurden a priori festgelegt und werden zur Laufzeit nicht mehr verändert. Damit können sie sich schlecht auf die momentanen Zustände einer AR- bzw. Tracking-Anwendung einstellen und können dadurch zu schlechteren Ergebnissen bei der Kalibrierung führen. Jedes der vorgestellten Verfahren kann durch zusätzliche Methoden zu einem adaptiven Verfahren erweitert werden. In Abbildung 5.1 ist ein möglicher Ablauf für eine adaptive Steuerung aufgezeigt. Das Verfahren benutzt die Ergebnisse aus der Registrierung, um die Verfahren der Dimensionsreduktion zu optimieren. Dieser Regelungsschritt kann beliebig angepasst werden.

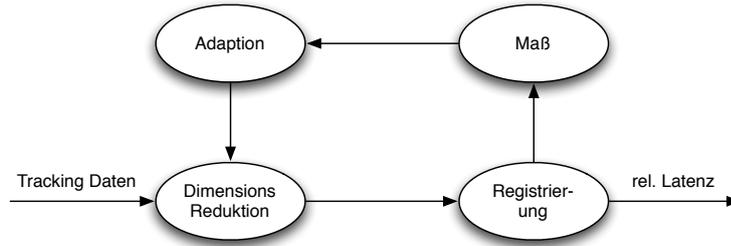


Abbildung 5.1: Prozessvorschlag einer adaptiven Steuerung eines Verfahrens der Dimensionsreduktion.

In den folgenden Abschnitten werden Methoden vorgestellt, die teilweise der eben vorgestellten Funktionsweise folgen.

### 5.3.1 Kanonische Korrelation

Die kanonische Korrelation ist nach [12] ein Ähnlichkeitsmaß zwischen zwei mehrdimensionalen Signalen und wurde erstmalig von Hotelling im Jahre 1936 in [48] vorgestellt. Sie definiert im Gegensatz zur auf eindimensionalen Daten beruhenden Korrelation ein Maß über den Grad der linearen Beziehung zwischen zwei mehrdimensionalen Signalen. Das von Johansson [54] beschriebene Verfahren beruht im Wesentlichen auf der Bestimmung von Projektionsvektoren, mit welchen Signale mit verschiedener Anzahl an Dimensionen projiziert werden, und auf einer Berechnung des zeitlichen Versatzes zwischen diesen Signalen. Die Kreuzkorrelation ist für dieses Verfahren ein zentrales Werkzeug, da sie sowohl für die Bestimmung der optimalen Projektionsvektoren als auch für die Bestimmung der zeitlichen Versatzes der Signale benutzt wird.

Für die Dimensionsreduktion, welche in diesem Kapitel beschrieben ist, ist nur die Bestimmung der optimalen Projektionsvektoren von Bedeutung. Im Folgenden wird kurz die Berechnung der Vektoren vorgestellt. Zu Beginn des Verfahrens liegen zwei zeitdiskrete Signale  $S_i^A \in \mathbb{R}^n$  und  $S_i^B \in \mathbb{R}^m$  vor. Es wird angenommen, dass beide Signale zu den gleichen Zeitpunkten abgetastet wurden. Dies ist in der Regel nicht der Fall, doch mit den in dieser Arbeit vorgestellten Möglichkeiten ist dies zu erreichen.

Die Aufgabe ist es nun, zwei Projektionen  $P_a : \mathbb{R}^n \rightarrow \mathbb{R}$  und  $P_b : \mathbb{R}^m \rightarrow \mathbb{R}$  zu finden, so dass die Bilder der Projektionen mit Hilfe der regulären normalisierten Kreuzkorrelation verglichen werden können. Für die Suche nach den optimalen Projektionsvektoren wird dabei die Korrelation der Bilder der Projektionen maximiert. Seien  $w_a$  und  $w_b$  die Projektionsvektoren und  $s_i^A = w_a^T S_i^A \in \mathbb{R}$  und  $s_i^B = w_b^T S_i^B \in \mathbb{R}$  die Bilder der Projektionen. Dann lässt sich die Suche nach den Projektionsvektoren schreiben als

$$(w_a, w_b) = \operatorname{argmax}_{(w_a, w_b)} (\operatorname{Corr}(s_i^A, s_i^B)). \quad (5.5)$$

Das Verfahren zeichnet sich vor allem durch die Invarianz gegenüber affinen Abbildungen aus [54], wobei eine affine Abbildung  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$  nach [111] wie folgt definiert ist

$$F(\mathbf{p}) = \mathbf{A}\mathbf{p} + \mathbf{q} \quad \forall \mathbf{p} \in \mathbb{R}^n. \quad (5.6)$$

$\mathbf{A}$  bezeichnet hier eine lineare Abbildung und  $\mathbf{q}$  bezeichnet eine Translation.

Durch die Invarianz gegenüber affinen Abbildungen ist die kanonische Korrelation von zwei Signalen  $S_i^A$  und  $S_i^B$  gleich der Korrelation von  $S_i^A$  und  $F(S_i^B)$ . Somit ist es irrelevant, bezüglich welchen Koordinatensystems die Signale repräsentiert sind. Dies hat zur Folge, dass auch keine Registrierung der Sensoren zueinander nötig ist. Denn Transformationen zwischen den Koordinatensystemen sind auch lineare Abbildungen [38].

In [54] beschreibt Johansson eine Adaption der kanonischen Korrelation zur Bestimmung des temporalen Abstands zweier beliebig dimensionaler Datenströme. Das Problem der Zeitkalibrierung von Sensoren in AR Anwendungen ist damit vergleichbar. In ersten Voruntersuchungen hat sich jedoch gezeigt, dass die Bestimmung der Projektionsvektoren unter Maximierung der Korrelation negative Auswirkungen auf das Ergebnis der Kalibrierung hat. Denn der Verlauf der Korrelationskurve wird sehr flach und somit die Maximumsfindung sehr ungenau. Die Streuung der Ergebnisse nahm somit zu. Der Grund hierfür liegt wahrscheinlich an der zweimaligen Anwendung der Korrelation sowohl für die Bestimmung der optimalen Projektionsvektoren als auch für die Bestimmung des zeitlichen Versatzes.

In Abbildung 5.2 ist eine Abbildung der kanonischen Korrelation kurz dargestellt. Es bleibt aber für zukünftigen Arbeiten zu klären, wie der viel versprechende Einsatz der kanonischen Korrelation aussehen kann. So kann untersucht werden, ob die Bestimmung der Projektionsvektoren auf Grundlage aller bis her vorliegender Sensordaten durchgeführt wird, oder ob die Berechnung für jedes Segment erneut durchgeführt wird. Es ist sogar denkbar, bei jedem Verschiebungsschritt der eigentlichen Kalibrierung diese Bestimmung durchzuführen. Dabei muss aber auch auf die effiziente Berechnung geachtet werden.

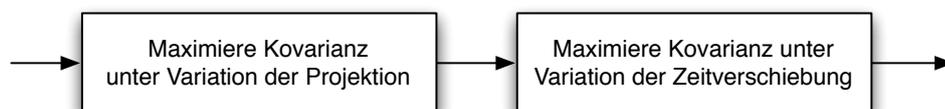


Abbildung 5.2: Schematische Darstellung der kanonischen Korrelationsmethode nach Johansson [54].

### 5.3.2 Two-Choice Algorithmen

Das Verfahren der kanonischen Korrelation muss bei jedem neuen Segment einen neuen Optimierungsvorgang durchführen, um eine für den aktuellen Kontext optimale Projektion zu bestimmen. Ein radikal anderer Ansatz ist durch das Two-Choice Verfahren gegeben. In diesem Verfahren werden zwei zufällig gewählte Parametersätze für das Dimensionsreduktionsverfahren ausgewählt und das Ergebnis gewählt, welches das bessere Resultat zeigt. Dieses Verfahren versucht erst gar nicht, ein optimales Ergebnis zu finden. Es versucht nur die pathologischen Fälle zu verhindern, indem zwei zufällig gewählte Dimensionsreduktionen angewendet werden. Mit diesen zwei Ergebnissen der Dimensionsreduktion wird dann jeweils eine Kalibrierung der relativen Latenz durchgeführt. Mit einem geeigneten Maß, welches die Güte einer Registrierung bestimmen kann, wird dann das bessere Ergebnis gewählt. Das Verfahren verringert durch die zufällige Auswahl der Dimensionsreduktion die Wahrscheinlichkeit des Eintritts pathologischer oder ungünstiger Parameter. Es kann zwar nicht garantieren, dass keine für die gewählte Dimensionsreduktion pathologische Situation auftritt, aber es kann die Eintrittswahrscheinlichkeit deutlich zu verringern.

Andere Anwendungen für Two-Choice Verfahren sind zum Beispiel Hashing-Funktionen oder Algorithmen zu Lastverteilung von Servern [72]. Mitzenmacher beschreibt in [72] die Grundlagen, verschiedene Methoden und Einsatzgebiete der Two-Choice Methode. Auch für den Zweck der Dimensionsreduktion wurde dieses Verfahren benutzt [29]. Die von Mitzenmacher beschriebenen Anwendungen lassen sich vergleichen mit dem Versuch  $n$  Bälle in  $n$  Urnen zu werfen. Ziel dabei ist es, die Bälle möglichst gleich zu verteilen und die maximale Anzahl von Bällen in einer Urne zu minimieren. Es wird in [72] gezeigt, dass selbst eine minimale Wahlmöglichkeit des Algorithmus eine erhebliche Verbesserung der Ergebnisse erzielt.

In dem Fall der Dimensionsreduktion kann der Parameterraum für die Reduktion in disjunkte Unterräume aufgeteilt werden. Es gilt nun einen solchen Unterraum zu finden, so dass die Elemente dieses Unterraums eine korrekte Reduktion der Eingangsdaten erlauben. Diese Unterräume entsprechen den Urnen bei Mitzenmacher, nur dass in diesem Fall nicht eine Häufung von Bällen verhindert werden soll, sondern die schlechten Urnen vermieden werden sollen.

Eine mögliche Methode, um dieses Prinzip für die Dimensionsreduktion umzusetzen, wird im Folgenden beschrieben. Es werden zwei zufällige Parametersätze für ein beliebiges aber festes Dimensionsreduktionsverfahren gewählt. Für diese beiden Parameter werden jeweils die Positions- oder Orientierungsdaten der beiden Sensoren in der Dimension reduziert und zwei Zeitkalibrierungen durchgeführt. Mit Hilfe eines Maßes, welches die Güte der Bestimmung bestimmt (vgl. Abschnitt 7.4.2), kann nun das Ergebnis gewählt werden, welche die bessere Güte besitzt.

Das Verfahren mag einfach erscheinen, es bietet den Vorteil, dass nicht wie bei anderen Verfahren systematisch nach den optimalen Parametern für eine Reduk-

tion gesucht werden muss. Im Vergleich zu einem statischen Verfahren verdoppelt sich der Rechenaufwand hier nur um den Faktor Zwei. Weiterhin kann mit Hilfe der Stochastik die Fehlerwahrscheinlichkeit dieses Verfahrens theoretisch bestimmt werden und somit a priori bestimmt werden, ob dieses Verfahren für einen konkreten Fall ausreicht.

### 5.3.3 Hauptkomponentenanalyse

Die Hauptkomponentenanalyse ist ein statistisches Verfahren, um große und mehrdimensionale Datenmengen zu vereinfachen und zu analysieren. Mit diesem Verfahren werden sogenannte Hauptkomponenten bestimmt, die die Daten annähern. Die Hauptkomponenten sind Vektoren mit der gleichen Dimensionalität wie die zugrundeliegenden Daten. Unter allen möglichen normierten Projektionsvektoren liefert die Projektion mit der ersten Hauptkomponente die größte Varianz im Bild der Projektion [29]. Diese Eigenschaft kann man benutzen, einen großen SNR Wert nach der Dimensionsreduktion zu erreichen.

Der primäre Bestandteil einer Hauptkomponentenanalyse (engl.: *principle components analysis, PCA*) ist eine orthogonale Transformation von multidimensionalen korrelierten Variablen in eine Menge neuer und unkorrelierter Variablen. Diese werden auch Hauptkomponenten genannt. Mit Hilfe dieses Verfahrens ist es möglich, die Struktur umfangreicher Datensätze zu analysieren, indem es eine Anzahl von korrelierten Variablen in eine geringere Anzahl an nicht korrelierenden Variablen überführt [14]. Diese Eigenschaft kann genutzt werden, um einen Projektionsvektor zu finden, der zum einen keinen pathologischen Fall erzeugt und zum anderen möglichst wenig Information eliminiert. Die PCA wurde auch schon in Arbeiten eingesetzt um die Dimensionalität eines Problems zu reduzieren [29].

Zunächst soll das Prinzip anhand eines einfachen Beispiels vorgestellt werden. Angenommen der Benutzer bewegt während der zeitlichen Kalibrierung das Sensorpaar in einer Ebene. Obwohl diese Bewegung im drei dimensionalen Raum stattfindet, kann sie mit zwei Freiheitsgraden beschrieben werden, da eine Ebene im dreidimensionalen Raum nur zwei Dimensionen hat. Mit Hilfe der Hauptkomponentenanalyse ist es möglich, diese Ebene, in welcher die Bewegung stattfindet, zu bestimmen. Und damit ist es möglich, Aussagen über einen möglichen Projektionsvektor zu treffen. Ein Projektionsvektor, der senkrecht zu dieser Ebene ist, erzeugt einen pathologischen Fall, da die Projektion auf diesen Vektor immer Null ergeben wird. Für die Zeitkalibrierung ist aber, wie in Abschnitt 4.5 beschrieben, ein großes Signal-Rausch-Verhältnis von Vorteil. Nachfolgend wird gezeigt, dass dies mit der Wahl der ersten Hauptkomponente als Projektionsvektor eine großer SNR Wert erzielt werden kann.

Rein formal werden bei einer Hauptkomponentenanalyse eines  $n$  dimensionalen Datensatzes  $n$  orthogonale Vektoren bestimmt, die die Richtungen der größten Variationen angeben. Diese Vektoren entsprechen dann den Hauptkomponenten. Wenn die Variablen  $x_i$  die  $n$  dimensionalen Daten repräsentieren, dann kann erste Hauptkomponente  $v_1$  kann wie folgt bestimmt werden

$$v_1 = \operatorname{argmax}_{\|v\|=1} \operatorname{Var}\{x_i^T v\} \quad \text{für } x \in \{1, \dots, N\}. \quad (5.7)$$

Die zweite Hauptkomponente  $v_2$  gibt die Richtung der stärksten Variation senkrecht zur ersten Hauptkomponente an. Diesem Prinzip folgend können dann die weiteren Hauptkomponenten bestimmt werden [29].

Die Hauptkomponenten können auch mit Hilfe der Kovarianzmatrix bestimmt werden [29]. Dazu bestimmt man die Kovarianzmatrix  $\Sigma = \operatorname{Cov}(X)$  der Messwerte  $X = (X_i, \dots, X_{i+N})$  eines Segments mit  $N$  Messdaten. Von dieser Kovarianzmatrix wird die orthogonale Transformation bestimmt, welche es erlaubt die Daten zu dekorrelieren. Dies geschieht mit Hilfe der Bestimmung der Eigenwerte und Eigenvektoren [86]. Seien nun  $\lambda_i$  die Eigenwerte der Kovarianzmatrix  $\Sigma$  der Größe nach sortiert, also  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_M$ . Die zu  $\lambda_i$  gehörigen Eigenvektoren bilden dann die Spalten der orthogonalen Transformationsmatrix

$$\mathbf{R} = (v_1, \dots, v_M). \quad (5.8)$$

Werden die Eigenwerte  $\lambda_i$  als Diagonalmatrix

$$\mathbf{D} = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_N \end{pmatrix} \quad (5.9)$$

geschrieben, dann lässt sich für die Kovarianzmatrix die folgende Zerlegung angeben

$$\Sigma = \mathbf{RDR}^t. \quad (5.10)$$

Die Details für diese Zerlegung sind in [14, 86] nachzulesen. Nun liegen die Hauptkomponenten der Daten vor, aber wie können diese verwendet werden, um eine möglichst verlustarme Dimensionsreduktion zu erstellen? Sieht man sich die Hauptkomponenten einer einfachen geradlinigen Bewegung an, so ist die Frage nach einem optimalen Projektionsvektor leicht zu beantworten. Es ist ein Vektor der parallel zur Bewegungsrichtung liegt und genau die gleiche Ausrichtung hat wie die größte Hauptkomponente (Abb. 5.3(a)). Die anderen Hauptkomponenten sind bei einer geradlinigen Bewegung gleich Null, denn es findet keine Bewegung orthogonal zur Bewegungsrichtung statt. Sieht man sich eine elliptische Bewegung im dreidimensionalen Raum an, so entsprechen den Hauptachsen der Ellipse die ersten zwei Hauptkomponenten (Abb. 5.3(b)). Die dritte Komponente ist klein gegenüber den ersten beiden Komponenten (Abb. 5.3(b)).

Mit Hilfe der Formel 5.7 ist leicht einzusehen, dass die erste Hauptkomponente gerade der Projektionsvektor ist, der die größte Variation im Bild der Projektion liefert. Somit liefert auch dieser Projektionsvektor den größten SNR Wert in den Daten nach der Dimensionsreduktion. Wenn  $\lambda_1$  der größte Eigenwert ist und  $v_1$  der zugehörige Eigenvektor, dann lautet der gesuchte Projektionsvektor wie folgt

$$p = \lambda_1 v_1. \quad (5.11)$$

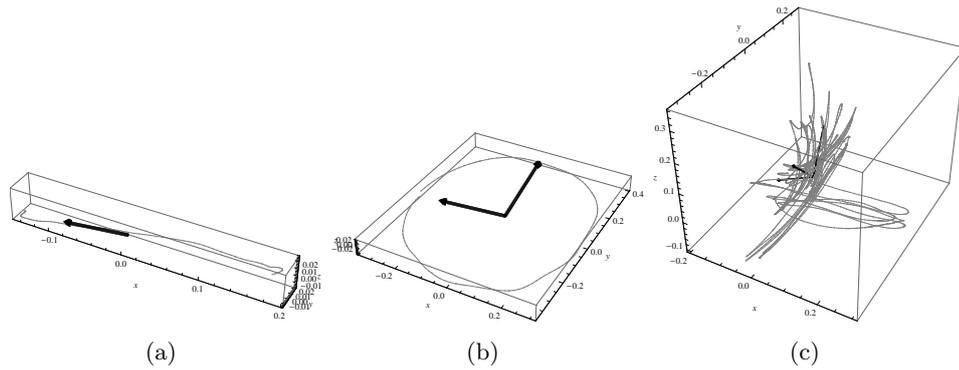


Abbildung 5.3: Visualisierung von Raumkurven und deren Hauptkomponenten; für eine bessere Visualisierung wurden die Hauptvektoren skaliert; (a) zeigt eine geradlinige Bewegung; Die erste Hauptkomponente ist parallel zur Bewegungsrichtung; (b) zeigt eine Ellipsenförmige Bewegung; Die erste Hauptkomponente ist parallel zur längeren Hauptachse der Ellipse (c) zeigt eine wirre Bewegung im dreidimensionalen Raum ohne erkennbare Tendenz; Die erste Hauptkomponente unterscheidet sich kaum von den anderen beiden Hauptkomponenten

Die Methode der Hauptkomponentenanalyse für die Bestimmung optimaler Projektionsvektoren kann sowohl global also auch lokal durchgeführt werden. Bei einer globalen Bestimmung der Projektionsvektoren werden alle vorhandenen Daten berücksichtigt. Dadurch stehen dem Verfahren zwar sehr viele Daten zur Verfügung, aber das Verfahren ist dafür auch sehr unflexibel bezüglich sich ändernder Latenzen. Zudem schwindet bei einer großen Anzahl von Daten die Wahrscheinlichkeit, dass der Nutzer eines AR Systems Bewegungen in einer Ebene oder entlang einer Geraden durchführt. Und damit würde die erste Hauptkomponente sich nicht wesentlich von den restlichen unterscheiden (vgl. Abbildung 5.3(c)). Eine lokale Bestimmung führt man hingegen auf Segmentebene durch. Dieses Verfahren ist deutlich adaptiver und kann daher sehr gut in einem Verfahren, welches zur Laufzeit arbeitet, eingesetzt werden.

Um die globale PCA Methode zu demonstrieren wurden mit dem FaroArm über 20s Daten aufgenommen. Dabei wurde der Sensor absichtlich mehrfach auf einer Kreisbahn bewegt. Wie in Abbildung 5.4(a) zu sehen ist, wird der Sensor hauptsächlich in der  $x-y$  Ebene bewegt. Anschließend wurden mit der PCA die Hauptkomponenten berechnet. Erwartungsgemäß waren zwei Komponenten sehr groß und eine sehr klein. Als Projektionsvektor wurde der normierte Vektor der ersten Hauptkomponente gewählt. Zum Vergleich wurde ein Projektionsvektor  $w_z = (0, 0, 1)^t$  senkrecht zur  $x - y$  Ebene gewählt, welcher in diesem Fall einen pathologische Fall erzeugt. In Abbildung 5.4(b) sind die Bilder beider Projektionen zu sehen. Eindeutig ist zu erkennen, dass der Projektionsvektor welcher durch

die PCA bestimmt wurde einen größeren SNR Wert liefert, als der pathologische Projektionsvektor entlang der z-Achse.

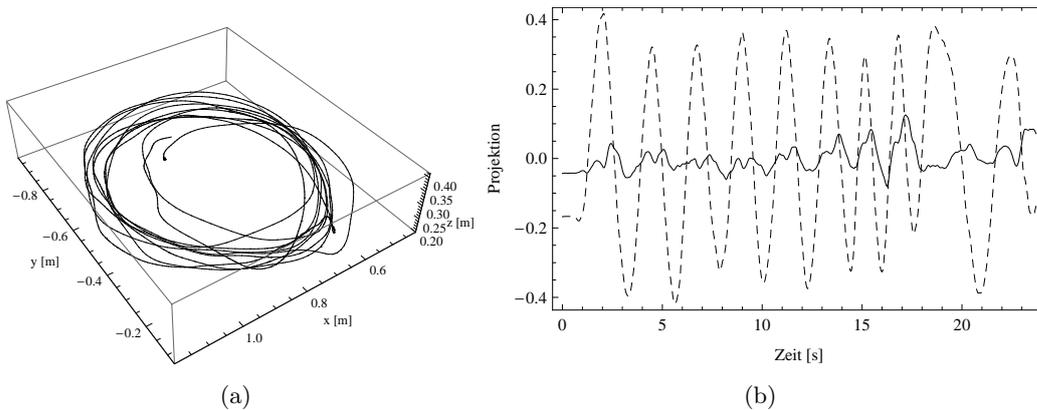


Abbildung 5.4: Globale Bestimmung des Projektionsvektors: (a) zeigt den Pfad; (b) zeigt die Ergebnisse zweier Projektionen; Die gestrichelte Kurve wurde mit einem Projektionsvektor bestimmt durch die PCA Methode berechnet; Die andere Kurve wurde mit dem pathologischen Projektionsvektor berechnet

In diesem Fall funktioniert die globale Methode sehr gut, denn die Daten liegen in einer Ebene, und das Verfahren kann somit leicht einen optimalen Projektionsvektor in dieser Ebene bestimmen. Im Allgemeinen sehen Bewegungen in einem Augmented Reality Szenario nicht so gleichmäßiger aus, doch auf lokale Ebene können sie gleichmäßige Anteile aufweisen. Daher wird im Folgenden die Hauptkomponentenanalyse auf Segmentebene untersucht.

In den Abbildungen 5.5(a) und 5.5(b) sind jeweils zwei mit einer Kombination von A.R.T. und FaroArm aufgenommen Pfade dargestellt. Für den Pfad A wurden schnelle Bewegungen in alle drei Raumdimensionen durchgeführt, um einen möglichst gut verteilten Pfad für eine zeitliche Kalibrierung zu erhalten. Pfad B hingegen ist eine Bewegung, die weitgehend planar ist, aber viele ruckartige Bewegungen aufweist. Auf den Daten von Pfad A und Pfad B wurden die Projektionsvektoren global auf dem gesamten Datensatz bestimmt und anschließend mit Hilfe dieses Vektors alle Daten beider Sensoren auf eine Dimension reduziert. Auf diesen Daten wurde anschließend eine Bestimmung der relativen Latenz durchgeführt. Die Ergebnisse dieser Kalibrierung sind in Tabelle 5.1 aufgeführt. Beide Methoden liefern vergleichbare Ergebnisse.

Dann wurden die Bestimmung der Projektionsvektoren zur Dimensionsreduktion auf der Ebene von Segmenten durchgeführt, also für jedes Segment der beiden Pfade ein eigener Projektionsvektor bestimmt. Die Ergebnisse dieser Kalibrierungen sind ebenfalls in Tabelle 5.1 aufgeführt. Auch hier liefert die Methode vergleichbare Ergebnisse. Eine signifikanter Unterschied der bestimmten relativen Latenzen unter Verwendung der lokalen und der globalen Bestimmung der Projektionsvek-

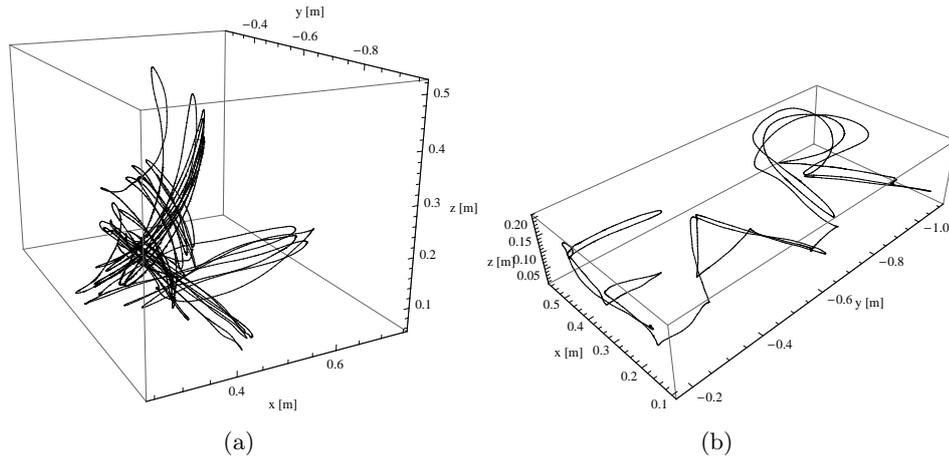


Abbildung 5.5: Bespielpfade für die Evaluation der globalen und lokalen Bestimmung der Projektionsvektoren mit der Hauptkomponentenanalyse; 5.5(a) zeigt Pfad A; 5.5(b) zeigt Pfad B

toren ist nicht zu erkennen.

	Global		Lokal	
	$\mu_{\delta_1}$	$\sigma_{\delta_1}$	$\mu_{\delta_2}$	$\sigma_{\delta_2}$
Pfad A	31,9 ms	0,67 ms	32,0 ms	0,86 ms
Pfad B	31,7 ms	0,39 ms	32,1 ms	1,2 ms

Tabelle 5.1: Vergleich der lokalen und der globalen Bestimmung der Projektionsvektoren mit Hilfe der Hauptkomponentenanalyse anhand der Kalibrierergebnisse und deren Standardabweichung

Es ist jedoch eine Erhöhung der Standardabweichung der Schätzungen unter Verwendung der lokalen Methode festzustellen. So ist bei der Kalibrierung mit den Daten von Pfad B die Standardabweichung um den Faktor Drei gestiegen. Auch die Kalibrierung über die Daten von Pfad A bei Verwendung der lokalen Methode weist eine erhöhte Standardabweichung auf, jedoch ist hier der Faktor nur rund 1,3. Warum die Standardabweichung bei Verwendung der lokalen Bestimmung der Projektionsvektoren steigt, muss in weiteren Untersuchungen geklärt werden. Man kann jedoch vermuten, dass einzelne Segmente ungünstige Daten für eine Bestimmung der Projektionsvektoren enthalten. Somit liefert eine segmentweise Bestimmung der Vektoren unterschiedliche Ergebnisse und daher weisen auch die Ergebnisse der Kalibrierung stärkere Sprünge auf. Bei Verwendung der globalen Methode treten diese ungünstigen Segmente in den Hintergrund. Welche Segmente sich nicht für eine Bestimmung des Projektionsvektors mit Hilfe PCA Methode eignen, muss in weiteren Untersuchungen geklärt werden. Eine Möglichkeit solche Segmente zu erkennen ist die Verwendung der in Kapitel 7 vorgestellten Methoden.

Mit der PCA steht ein effektives Werkzeug zur Dimensionsreduktion unter Vermeidung von pathologischen Fällen zur Verfügung. Und weiterhin kann man vermuten, dass die PCA Methode den Einfluss schlechter Kalibrierungen vermindern kann. Angenommen zwei identische Bahnen wurden bezüglich verschiedener Koordinatensysteme gemessen und es wurde keine oder eine fehlerhafte räumliche Registrierung vorgenommen. Nachdem die erste Hauptkomponente genau der Vektor ist, unter dessen Projektion die Varianz der Daten maximal wird, wird die Projektion der beiden Datensätze durch ihre ersten Hauptkomponenten die gleichen eindimensionalen Signale liefern.

### 5.4 Zusammenfassung

Dieses Kapitel hat mehrere Verfahren für eine Dimensionsreduktion vorgestellt. Es wurden sehr einfache statische Methoden gezeigt, bis hin zu aufwendigen adaptiven Verfahren, die eine Optimierung in jedem Schritt benötigen. Die Bestimmung der Projektionsvektoren bei der kanonischen Korrelation ist sicher die attraktivste Lösung, da sie eine Unabhängigkeit von der vorliegenden räumlichen Kalibrierung bietet. Dieser Vorteil wird jedoch durch eine hohe Varianz und einen hohen Rechenaufwand erkauft. Dieser Rechenaufwand lässt sich zwar durch einige Annahmen reduzieren, aber im Grunde wird in jedem Segment im übertragenen Sinne eine räumliche Kalibrierung vorgenommen. Die *two-choice* Methode ist sehr einfach in der Anwendung und benötigt wenig Rechenzeit, jedoch muss diese Methode in weiteren Untersuchungen getestet werden. Die Verwendung der PCA Methode kann pathologische Fälle verhindern, indem sie die Varianz des dimensionsreduzierten Signals maximiert. Zudem steht zu vermuten, dass diese Methode auch eine gewisse Unabhängigkeit von der räumlichen Registrierung bietet. Diese Hypothese muss aber in weiteren Versuchen validiert werden.

## 6 Synchronisierung und Segmentierung

In diesem Kapitel wird die Synchronisierung der Daten beschrieben. Die Daten zweier Sensoren müssen für eine zeitliche Kalibrierung direkt miteinander verglichen werden. Daher ist es notwendig, dass beide Sensoren die Daten mit der selben Frequenz messen bzw. die Messergebnisse zu den gleichen Zeitpunkten dem Messsystem übergeben. Dies ist aber im Allgemeinen nicht der Fall. Dieses Kapitel befasst sich mit Methoden, wie eine Synchronizität der Sensoren hergestellt werden kann, wobei die dabei auftretenden Fehler möglichst klein gehalten werden sollen. Daneben wird auch diskutiert, wie die Aufteilung der Daten in einzelne Segmente geschehen kann, und wie diese Segmentierung sich auf den Gesamtprozess der Kalibrierung auswirkt.

### 6.1 Interpolation

Um ein Signal auf ein Anderes zu synchronisieren, ist die Interpolation ein gängiges Prinzip [65, 81]. In Abbildung 6.1 ist das Prinzip der Synchronisierung durch Interpolation schematisch dargestellt.

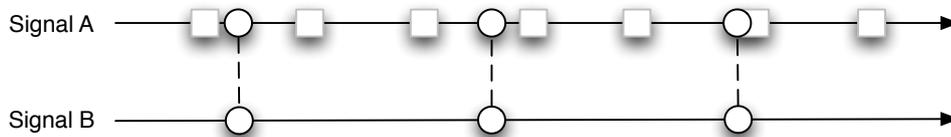


Abbildung 6.1: Prinzip der Synchronisierung durch Interpolation. Das Signal (A) mit der größeren Wiederholrate wird zu den Messzeitpunkten des Signals (B) mit der geringeren Wiederholrate interpoliert.

Die zwei Sensoren haben verschiedene Wiederholraten und somit liefern sie die Messungen nicht zu den gleichen Zeitpunkten und nicht in den gleichen Intervallen. Nun kann mit Hilfe von numerischen Methoden eine Funktion zwischen zwei Stützstellen interpoliert werden und somit ein Wert der Funktion an jeder beliebigen Stelle berechnet werden. Dieses Prinzip kann auch für die mehrdimensionalen Signale oder dimensionsreduzierten Signale verwendet werden. Aus Gründen, die später in diesem Kapitel erläutert werden, wird das Signal (A) mit der größeren Wiederholrate auf das Signal (B) mit der geringeren Wiederholrate synchronisiert. Damit wird das Signal A zu den Zeitstempeln des Signals B interpoliert. Im Folgenden wird die Methode der Interpolation genauer betrachtet, während die

Synchronisation in einem späteren Teil dieses Kapitels behandelt wird.

Für die Interpolation wurden in der numerischen Mathematik eine Vielzahl von Verfahren entwickelt. In dem Bereich der Augmented Reality wird jedoch meist nur eine einfache lineare Interpolation verwendet [5, 50, 65], denn bei den relativ hohen Wiederholraten im Vergleich zu den Bewegungsgeschwindigkeiten ist eine lineare Interpolation ausreichend. Daher wird auch in dieser Arbeit auf die lineare Interpolation zurückgegriffen, da andere Verfahren einen höheren Rechenaufwand bedeuten. Eine grundsätzliche Untersuchung über die Auswirkungen verschiedener Interpolationsmethoden [94] ist jedoch zu empfehlen.

### 6.1.1 Lineare Interpolation von Positionen

Sei  $p_0 \in \mathbb{R}^n$  die Startposition und  $p_1 \in \mathbb{R}^n$  die Endposition, dann wird die lineare Interpolation zwischen diesen Positionen durch

$$p_t = p_0 + t(p_1 - p_0), \quad t \in [0, 1] \quad (6.1)$$

definiert. Diese lineare Interpolation kann, wie in der Formel zu sehen, für beliebige Vektorräume benutzt werden, also auch für eindimensionale Daten.

### 6.1.2 Lineare Interpolation von Rotationen

Auch für Rotationen ist eine lineare Interpolation definiert. In der Literatur wird meist das Verfahren der sphärischen Interpolation (spherical linear interpolation, *Slerp*) für Quaternionen [65] benutzt. Dieses Verfahren wurde von Ken Shoemake in [96] beschrieben und garantiert eine konstante Rotationsgeschwindigkeit zwischen einer Startorientierung  $q_1$  und einer Endorientierung  $q_2$ . Wobei hier  $q_1$  und  $q_2$  als Quaternion dargestellt sind. Für die Definition und Darstellung von Rotationen siehe Abschnitt 2.3.1.

Die Interpolation zwischen  $q_1$  und  $q_2$  kann dann wie folgt berechnet werden

$$q_t = \text{Slerp}(q_0, q_1, t) = q_0(q_0^{-1}q_1)^t, \quad t \in [0, 1], \quad (6.2)$$

wobei die  $\text{Slerp}(q_0, q_1, t)$  Funktion für  $t = 0$  das Ausgangs-Quaternion  $q_0$  liefert und für  $t = 1$  das  $q_1$  Ziel-Quaternion liefert. Weitere Informationen zu der Interpolation von Rotationen und Informationen zu anderen Verfahren sind in [64, 22, 33] zu finden.

## 6.2 Synchronisierung

Mit einer Synchronisierung wird erreicht, dass für jeden Messwert des einen Sensors ein Messwert des anderen Sensors für den gleichen Zeitpunkt vorliegt. Ein Mittel, um die Synchronisierung zu erreichen, ist die eben beschriebene Interpolation. Es werden dazu die Zeitstempel eines der Sensoren benutzt, um die Daten des zweiten Sensors zu interpolieren.

### 6.2.1 Generelle Methode

Seien nun  $S_i^A$  für  $i \in \{1, \dots, N\}$  und  $S_i^B$  für  $i \in \{1, \dots, M\}$  die Messwerte der Sensoren an den diskreten Zeitpunkten  $T^A = \{t_1^A, \dots, t_N^A\}$  und  $T^B = \{t_1^B, \dots, t_M^B\}$ . Die angegebenen Zeitpunkte geben jedoch nicht die Zeitpunkte an, zu welchen die physikalischen Ereignisse aufgetreten sind, sondern die Zeitpunkte zu welchen die Trackingsoftware die Messwerte in Empfang genommen hat. Man spricht daher auch eher von Zeitstempeln als von Zeitpunkten. Abhängig von der Wahl der Interpolationsmethode, kann für einige Zeitpunkte keine Interpolation durchgeführt werden. Für die lineare Interpolation kann ein beliebiger Zeitpunkt in dem Intervall  $[t_1^A, t_N^A]$  gewählt werden, da die lineare Interpolation nur zwei Stützstellen benötigt. Andere Verfahren benötigen mehr als zwei Stützstellen, womit sich der Bereich, in welchem eine Interpolation möglich ist, weiter einschränkt. Dies ist jedoch keine Einschränkung, da diese Interpolation nicht zur Echtzeit durchgeführt wird, sondern auf vorher aufgezeichneten kurzen Segmenten erfolgen kann. Die formale Beschreibung einer solchen Interpolation für eindimensionale Daten ist in dem vorigen Abschnitt aufgeführt. Um zwei Signale miteinander zu synchronisieren, können zwei Wege verfolgt werden. Zum einen können beide Signale zu Zeitpunkten interpoliert werden, die weder der Menge  $T^A$  noch der Menge  $T^B$  entsprechen. Oder ein Signal wird zu den Zeitpunkten des anderen interpoliert. Die erste der beiden Möglichkeiten ist nicht empfehlenswert, denn durch die Interpolation beider Signale entstehen zweimal Interpolationsfehler und auch der Rechenaufwand steigt. Bei der Nutzung einer der beiden Zeitstempelmengen als Taktgeber muss festgelegt werden, welche der beiden Zeitstempelmengen  $T^A$  und  $T^B$  als Taktgeber benutzt wird. Für die Auswahl eines Signals sind viele Merkmale denkbar. So ist es sicherlich besser, die Interpolation an einem Signal mit möglichst geringem Rauschen und kleinen systematischen Fehlern vorzunehmen. In dieser Arbeit wird vorgeschlagen, die Auswahl des Taktes anhand der Wiederholrate zu treffen. Denn eine Auswahl anhand von Qualitätsmerkmalen ist zur Laufzeit schwerlich möglich und könnte sich auch zur Laufzeit mehrmals ändern. Die Wiederholrate eines Sensors ist dagegen leicht zu prüfen. Zudem müssen weniger Interpolationen durchgeführt werden, wenn das Signal mit der geringeren Wiederholrate als Takt gewählt wird und dies spart Rechenzeit. Auch die Fehler, bedingt durch die Interpolation, verringern sich, denn bei dem interpolierten höher frequenten Signal liegen die Stützstellen dichter zusammen und somit sinkt der Interpolationsfehler.

### 6.2.2 Wiederholte Synchronisierung

Die Synchronisierung ist kein einmaliger Vorgang, sondern wird bei jeder Teilverschiebung während der Kalibrierung wiederholt. Soll bei der Kalibrierung die relative Latenz eine Genauigkeit von 1 ms bestimmt werden, so muss das Signal A maximal in Schritten von 1 ms gegen das Signal B verschoben werden. Und da die relative Latenz nicht konstant ist, muss diese Suche in einem Fenster mit einer gewissen Breite durchgeführt werden. Daher ist bei der Wahl des Interpolations-

verfahrens, umso mehr auf die Effizienz des Verfahrens aber auch auf die Fehler zu achten.

Die Wahl des Zeitpunkts in der Prozesskette, zu dem die Synchronisierung durchzuführen ist, ist ebenfalls eine Abwägung zwischen Genauigkeit und Performanz. Die Signale können entweder vor der Reduktion der Dimensionen interpoliert werden oder danach. Wird die Interpolation auf den dimensionsreduzierten Signalen durchgeführt, kann die Interpolation mit einer einfachen und performanten eindimensionalen linearen Interpolation geschehen. Eine Interpolation kann auch aber auf den mehrdimensionalen Positionsdaten bzw. den Rotationsdaten durchgeführt werden. Dadurch wird die Genauigkeit erhöht, jedoch leidet die Performanz unter der komplizierten Interpolation. Nachdem sich die Interpolationsmethoden von Positionen und Orientierungen stark unterscheiden, werden nachfolgend die Interpolationen beider Datentypen genauer untersucht und auch der Fehler bei einer eindimensionalen Interpolation betrachtet.

**Position** Um zu zeigen, dass durch eine Interpolation auf eindimensionalen Daten keinen wesentlichen größeren Ungenauigkeiten als bei einer mehrdimensionalen Interpolation erzeugt wird, wurde ein kurzer Versuch durchgeführt. Bei diesem Versuch wurde zuerst das dreidimensionale Signal interpoliert und das interpolierte Signal anschließend auf eine Dimension reduziert. Zum Vergleich wurde das dreidimensionale Signal erst auf eine Dimension reduziert und danach interpoliert. Das Resultat sind zwei eindimensionale Signale, von welchen dann die Differenz gebildet wurde. Je kleiner diese Differenz ist, desto kleiner ist der Fehler, der durch die Interpolation der eindimensionalen Daten entsteht.

Für diesen Versuch wurden mit einem FaroArm  $N = 4342$  Positionsdaten einer Bewegung aufgezeichnet. Die Bewegung ist in der Abbildung 6.2(a) dargestellt. Jede Messung  $(S_i, t_i)$  besteht dabei aus einer Position  $S_i \in \mathbb{R}^3$  mit  $1 \leq i \leq N$  und einem Zeitstempel  $t_i$ . Der Sensor wurde mit einer durchschnittlichen Geschwindigkeit von  $0,13 \text{ ms}^{-1}$  und auf einer Strecke von 11 m bewegt. Das Geschwindigkeitsdiagramm der Bewegung ist in Abbildung 6.2(b) zu sehen.

Die Interpolation wurde immer in der Mitte zwischen zwei Zeitstempeln durchgeführt  $\tilde{t}_i = \frac{1}{2}(t_{i+1} - t_i)$ , da hier die größten Fehler zu erwarten sind. Für diesen Versuch wurde die Norm als Dimensionsreduktion verwendet, also  $s_i = P(S_i) = \|S_i\|$ . Weiterhin bezeichne  $\tilde{s}_i \in \mathbb{R}$  und  $\tilde{S}_i \in \mathbb{R}^3$  die Interpolationen der eindimensionalen bzw. dreidimensionalen Daten zum Zeitpunkt  $\tilde{t}_i$ . Die Differenz zwischen den zwei Methoden kann dann wie folgt ausgedrückt werden

$$e_i = \|P(\tilde{S}_i) - \tilde{s}_i\| = \| \|\tilde{S}_i\| - \tilde{s}_i \|.$$

Der RMS der Differenz betrug nur  $2,1 \times 10^{-3}$  mm. Dieser Fehler ist vernachlässigbar klein, denn die Messungenauigkeit des benutzten FaroArm Fusion Geräts liegt stets über  $3,6 \times 10^{-2}$  mm und somit um ein Magnitude schlechter als der ermittelte RMS der Differenz. In Abbildung 6.2(c) ist diese Differenz als ein Fehler-Zeit-Diagramm dargestellt.

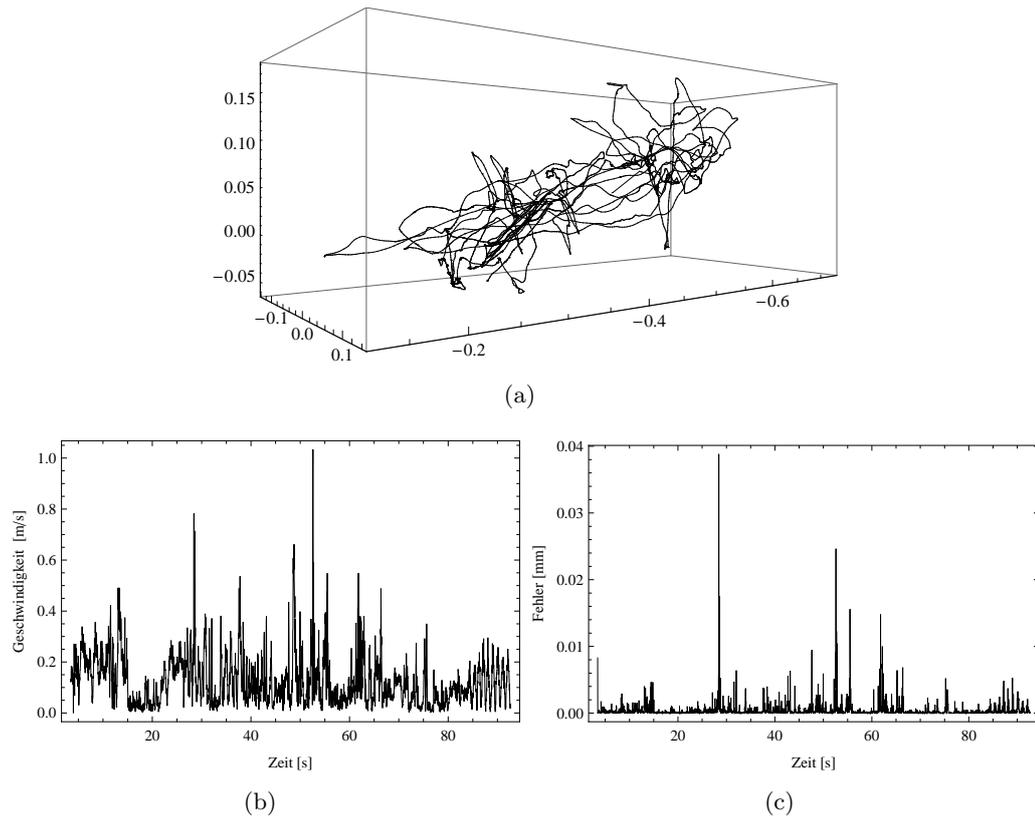


Abbildung 6.2: Verschiedene Zeitpunkte der Interpolation von Positionen in der Prozesskette; Plot (a) zeigt den für die Evaluation aufgenommenen Pfad des Sensors, Plot (b) zeigt die Geschwindigkeit des Sensors und Plot (c) zeigt den Unterschied zwischen dreidimensionaler Interpolation mit anschließender Dimensionsreduktion und Dimensionsreduktion mit anschließender eindimensionaler Interpolation.

**Orientierung** Ebenso wie bei Positionen, ist die Interpolation zwischen Orientierungen sowohl auf den mehrdimensionalen Daten als auch auf den dimensionsreduzierten Daten möglich. Dabei ist aus Gründen der Effizienz die Interpolation auf den eindimensionalen Daten vorzuziehen. Um zu zeigen, dass beide Methode annähernd die gleichen Fehler liefern, wurde der Versuch analog zu dem obigen durchgeführt. Die Orientierungen werden mit  $S_i \in \mathbb{R}^4$  und  $s_i \in \mathbb{R}$  bezeichnet. Als Repräsentation der  $S_i$  wurden Quaternionen gewählt. Die Zeitstempel sind identisch wie im obigen Versuch. Die Interpolation der  $S_i$  wurde mit der Slerp-Methode (s. Abschnitt 6.1.2) durchgeführt und die Interpolation der eindimensionalen Daten erfolgte linear. Als Dimensionsreduktion wurde der Rotationswinkel des Quaternionen gewählt  $s_i = P(S_w) = 2 \cos^{-1}(S_w)$ . Die Zeitpunkte für die Interpolationen wurden genauso wie im Versuch mit Positionen gewählt.

Die Differenz wurde dann wie folgt berechnet

$$e_i = \|P(\tilde{S}_i) - \tilde{s}_i\| = \|\tilde{S}_{w,i} - \tilde{s}_i\|.$$

Der RMS der Differenzen betrug  $0,030^\circ$ . Diagramme der Winkelgeschwindigkeiten und der Differenzen sind in der Abbildung 6.3 zu sehen. Somit produziert im Falle von Orientierungen die Interpolation der eindimensionalen Daten relativ zu den Bewegungen nur kleine Fehler und ist somit der aufwändigeren Slerp Interpolation vorzuziehen.

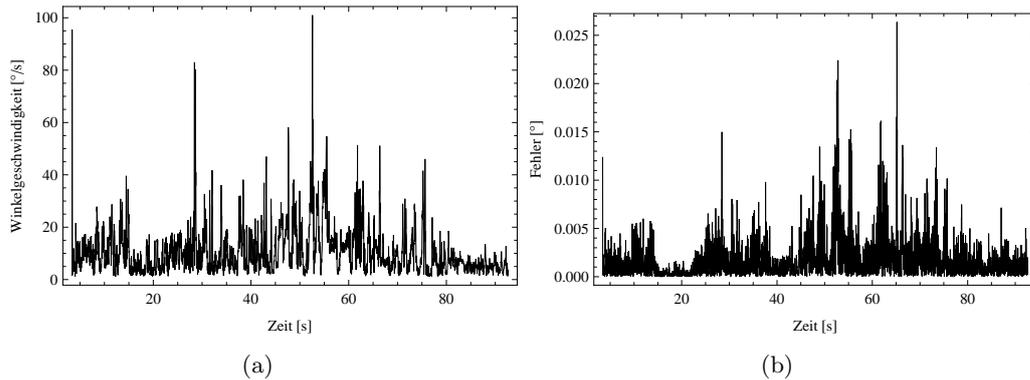


Abbildung 6.3: Verschiedene Zeitpunkte der Interpolation von Orientierungen in der Prozesskette; Plot (a) zeigt die Rotationsgeschwindigkeit des Sensors und Plot (b) zeigt den Unterschied zwischen Slerp Interpolation und anschließender Dimensionsreduktion und der Dimensionsreduktion mit angeschlossener linearer eindimensionaler Interpolation.

### 6.3 Segmentierung

Die Segmentierung teilt die Daten der Sensoren in kurze Bereiche auf. Die Kalibrierung bzw. nahezu die gesamte Prozesskette wurde auf diesen Segmenten durchgeführt. Eine Aufteilung des fortlaufenden Datenstroms nutzt vor allem in Hinblick auf die Adaptivität und die effiziente Berechenbarkeit einer Kalibrierung während der Laufzeit eines Systems.

Die Aufteilung an sich ist sehr einfach. Der Messstrom beider Sensoren wird in Segmente mit einer vorher festgelegter Anzahl  $N$  an Datenpunkten aufgeteilt. Neben der Anzahl an Messpunkten  $N$  ist noch zu entscheiden, um wie viele Messungen  $n$  sich die Segmente überlappen. Eine schematische Darstellung dieser Segmentierung ist in Abbildung 6.4 zu sehen.

Es gibt zahlreiche Argumente für eine Aufteilung in kurze Datensegmente. Die beiden wichtigsten sind Robustheit und Adaptivität. Treten bei den Messungen Ausreißer auf oder sind sie aus diversen anderen Gründen korrumpiert, so lassen

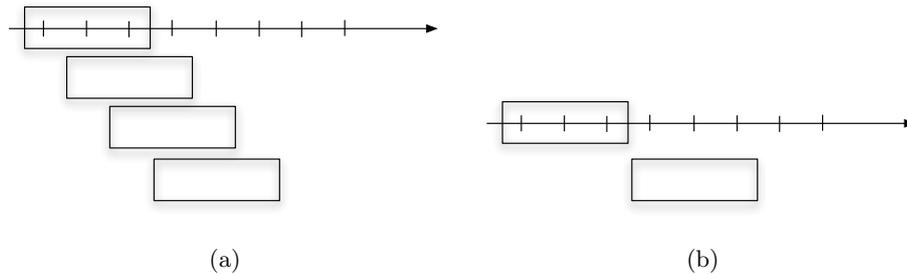


Abbildung 6.4: Segmentierung von Messdaten: (a) zeigt eine überlapende Segmentierung und (b) zeigt eine diskunkte Segmentierung der Messdaten

sich ganze Segmente einfach verwerfen. Ein Beispiel für eine Kalibrierung, welche aufgrund minderwertiger Daten misslungen ist, ist in Abbildung 4.4 auf Seite 56 zu sehen. Neben der Möglichkeit, ganze Segmente zu verwerfen, kann mit Gütemaßen die Qualität der Schätzung beurteilt werden und damit die einzelnen Schätzungen zu einer einzelnen genaueren Messung aggregiert werden.

Weiterhin ist die relative Latenz zwischen den Sensoren nicht konstant. Als Beispiel lässt sich wieder das Gyroskop MT-9B zitieren, welches innerhalb von Minuten die relative Latenz ändert, so dass auch mit den hier aufgeführten Methoden keine statische Kalibrierung durchgeführt werden kann. Vergleiche hierzu auch Abschnitt 4.2 und [1]. Neben den Schwankungen kann auch eine veränderte Last des Messsystems zu einer Veränderung der Latenz führen, wenn kein Echtzeitsystem genutzt wird. Solche Schwankungen können mit der Nutzung von Segmenten detektiert werden.

Bei der Wahl der Parameter für die Länge der Segmente und die Länge der Überlappung muss ein Kompromiss zwischen der Genauigkeit, der Robustheit, der Adaptivität und der Rechenkomplexität getroffen werden. Die Länge der Segmente betrifft, wie schon erwähnt, die Adaptivität und die Güte der Kalibrierung. Um präzise zu sein, hängt die Genauigkeit der Kalibrierung von der Anzahl der Messungen ab. Mit einer geringen Anzahl von Daten lässt sich keine exakte Kalibrierung durchführen. Zu lange Segmente verringern jedoch die Adaptivität gegenüber schwankenden Latenzen der Sensoren. Durch lange Segmente steigt außerdem auch die Wahrscheinlichkeit für negative Einflüsse wie beispielsweise Ausreißer, jedoch können diese mit der größeren Anzahl an Daten aber auch besser detektiert werden. Zum Beispiel stoßen bei sehr schnelle Bewegungen oder Drehungen, kamerabasierter Sensoren an ihre Grenzen und liefern qualitativ schlechtere Daten. Inertiale Sensoren liefern hingegen gerade bei schnellen Bewegungsänderungen genaue Daten, weswegen diese beiden Sensortypen oft miteinander fusioniert werden [84, 83]. Die im Rahmen dieser Arbeit durchgeführten Messungen haben gezeigt, dass mit  $N = 200$  gute Resultate erzielt werden konnten.

Neben der Wahl der Länge der Segmente  $N$  muss auch die Länge der Überlappung  $n$  gewählt werden. Hier muss ein Kompromiss zwischen Rechen-

aufwand und Anzahl der Schätzungen, welche später aggregiert werden, getroffen werden. Eine große Überlappung führt sehr schnell zu einem erhöhtem Rechenaufwand. Für die Ergebnisse in dieser Arbeit wurde eine Überlappung von  $n = 50$  Messungen gewählt. Für eine Implementierung für ein Echtzeitverfahren ist eine genauere Untersuchung dieser Parameter sinnvoll.

### **6.4 Zusammenfassung**

In diesem Kapitel wurde näher auf die Synchronisierung und Segmentierung eingegangen. Es wurden Interpolationsmethoden für den eindimensionalen Fall, für Positions- und Orientierungsdaten vorgestellt. Weiterhin wurde gezeigt, dass die Interpolation auf eindimensionalen Daten keine signifikanten Fehler erzeugt und somit der Interpolation mehrdimensionaler Daten vorzuziehen ist. Anschließend wurden verschiedene Parameter einer Segmentierung vorgestellt und wie diese zu wählen sind.

# 7 Zeitkalibrierung

In diesem Kapitel werden die Techniken vorgestellt, die für eine genaue zeitliche Kalibrierung zweier Sensoren benötigt werden. Dabei wird davon ausgegangen, dass die Daten entsprechend der vorherigen Schritte aufbereitet wurden und jetzt den Methoden zugeführt werden, die die eigentliche zeitliche Kalibrierung vornehmen.

Dazu wird zuerst auf den Stand der Technik eingegangen und dann auf die einzelnen Verfahren für die Bestimmung des zeitlichen Versatzes. Es wird im Anschluss auch eine Methode vorgestellt, die es erlaubt die Ergebnisse der Kreuzkorrelationsmethode zu bewerten und somit eine Aussage über die Güte der Kalibrierung zu treffen. Des Weiteren wird eine Möglichkeit beschrieben, wie die Ergebnisse zahlreicher Kalibrierungen auf den einzelnen Datenteilstücken zusammengefasst werden können, so dass sich einerseits ein stabiles und robustes Ergebnis ergibt, aber andererseits auch auf sich ändernde Latenzzeiten reagiert werden kann.

## 7.1 Einleitung

Es wurden bis jetzt zumeist Vorarbeiten präsentiert, die dazu dienen die Daten so aufzubereiten, dass eine Bestimmung der relativen Latenz vorgenommen werden kann. Die hier vorgestellte Technik trägt im Englischen den Namen „Time Delay Estimation“, also Schätzung des zeitlichen Versatzes. Dieses Verfahren ist in der Signalverarbeitung schon lange bekannt. Es stellt die Basis von Produkten wie Sonar oder Radar [17, 107] dar. Im Falle von Sonar wird etwa eine bestimmte Wellenform ausgesendet und dann mit einem Unterwassermikrofon (Hydrophon) auf das Echo gewartet. Dabei wird kontinuierlich das ausgesendete Signal mit dem empfangenen Signal verglichen. Sobald die beiden Signale übereinstimmen, wird gemessen, wie lange das Signal unterwegs war und mit dieser Zeit und der bekannten Schallgeschwindigkeit im Wasser kann berechnet werden, in welcher Entfernung der Schall im Wasser reflektiert wurde.

## 7.2 Timeshift-Verfahren

Zwei Signale in der Zeit gegeneinander zu verschieben wird im Weiteren Zeitschiebeverfahren oder auch Timeshift Verfahren genannt. Während die Signale gegeneinander verschoben werden, werden sie fortlaufend miteinander verglichen und die Zeitverschiebung, bei der eine maximale Übereinstimmung der beiden Signale auftritt, wird als relative Latenz gewertet. Das Zeitschiebeverfahren wird in zahl-

reichen Quellen beschrieben und darf somit als üblich betrachtet werden [18, 65]. Diese Arbeiten unterscheiden sich in Details. So wird die Ähnlichkeit der zu vergleichenden Signale bei einigen mit Korrelation bestimmt [54], bei anderen wird der visuelle Eindruck des Benutzers genutzt [51]. Timeshift Verfahren können sowohl zur Laufzeit des Systems, als auch offline durchgeführt werden. So gehen Lieberknecht u. a. in [65] von einer konstanten Latenz aus und kalibrieren die relative Latenz ausserhalb der Anwendung. Die in dieser Arbeit aufgeführten Methoden, sollen aber zu Echtzeit eingesetzt werden, weswegen auch der nötige Rechenbedarf so gering wie möglich gehalten werden soll. Im Folgenden wird das Timeshift Verfahren näher beleuchtet und dabei ein besonderer Fokus auf die Korrelation als Vergleichsfunktion gesetzt.

In allen Arbeiten, die sich mit der relativen Latenz beschäftigen, wird stets konstanten Latenzen der Sensoren berichtet. Eine Ausnahme ist der von Aron [1] untersuchte Sensor. Die Beobachtung konstanter relativer Latenzen konnten auch die Untersuchungen, die im Rahmen dieser Arbeit durchgeführt wurden, bestätigt werden. Bei identischen Sensoren und identischer Trackingsoftware konnten konstante Latenzen bestimmt werden. Wird jedoch bei einem Sensor auch ein Rechner genutzt um die Orientierungs- oder Positionsdaten zu bestimmen, wie beispielsweise bei einem Flackmarkertracker, kann sich die Latenz schon dadurch ändern, dass auf diesen Rechner weitere rechenintensive Prozesse hinzukommen oder andere beendet werden. Wenn in einer Anwendung alle Software- und Hardwarekomponenten konstant gehalten werden, sind keine signifikanten Abweichungen in der relativen Latenz zu erwarten.

Es wird vorgeschlagen die Kalibrierung in zwei Phasen aufzuteilen. In der ersten Phase wird eine grobe Bestimmung der relativen Latenz auf einem großen Suchfenster bestimmt, um den Wert der relativen Latenz eingrenzen zu können. Dies ähnelt dem Verfahren in der Arbeit von Prager [79], in welcher die relative Latenz mit einer maximalen Genauigkeit der Wiederholrate der Sensoren bestimmt wird. In der zweiten Phase wird das Suchfenster verkleinert bei gleichzeitiger Erhöhung der Auflösung. Die erste Phase wird meist nur in der Anfangsphase benötigt, wenn die relative Latenz noch unbekannt ist. Es ist aber durchaus möglich, dass während der Laufzeit eines AR-System kurzzeitig in Phase eins zurückgesprungen wird, falls sich die relative Latenz verändert. Mit der Aufteilung in zwei Phasen mit verschiedenen Präzisionen und Suchräumen ist gewährleistet, dass mit möglichst geringem Rechenaufwand eine hohe Genauigkeit der Kalibrierung erzielt werden kann.

### 7.2.1 Grobe Bestimmung

In der ersten Phase wird nur eine grobe Bestimmung der Latenz durchgeführt. Um möglichst wenig Rechenaufwand zu erzeugen, wird die Synchronisierung auf dem Takt des Sensors mit der niedrigeren Frequenz  $f_1$  durchgeführt. Das andere Signal wird dann in Schritten von  $T = \frac{1}{f_1}$  verschoben. Dabei werden beide Signale kontinuierlich miteinander verglichen. Die Größe des Suchfensters wird durch die visuellen Anforderungen von Augmented Reality Anwendungen nach oben be-

schränkt. Der Benutzer eines AR Systems würde eine Latenz von beispielsweise 1 s nur schwerlich akzeptieren. Angenommen das Suchfenster ist insgesamt 1 s breit dann müssen  $N = f_1$  Verschiebungen überprüft werden. Die Verschiebung mit der maximalen Übereinstimmung der Signale wird dann als Ausgangspunkt für die Bestimmung der Latenz mit einer Genauigkeit, die unterhalb der Wiederholrate beider Sensorsysteme liegt.

### 7.2.2 Feine Bestimmung

Diese Phase des Verfahrens operiert auf einzelnen Segmenten, aber mit einem kürzeren Suchfenster und einer feineren Suchschrittweite als in der Phase der groben Bestimmung. Dabei wird das Signal mit der größeren Frequenz auf das Signal mit der kleineren Frequenz synchronisiert und anschließend das Signal mit der größeren Wiederholrate in der Zeit verschoben. Mit diesem Vorgehen wird sichergestellt, dass eine hohe Auflösung bei der Kalibrierung der relativen Latenz möglich ist, aber die Interpolationsfehler so gering wie möglich bleiben. Dies liegt an den kleineren Abständen der Stützpunkte der Interpolation.

Seien nun  $S_i^A$  und  $S_i^B$  die zeitlich zu kalibrierenden diskreten Signale,  $s_i^A$  und  $s_i^B$  die auf eine Dimension reduzierten Signale und  $f^A$  und  $f^B$  die zugehörigen Wiederholfrequenzen. Weiterhin sei o. B. d. A.  $f^A < f^B$  angenommen. Mit  $\text{TS}^A = \{t_1^A, t_2^A, \dots, t_N^A\}$  und  $\text{TS}^B = \{t_1^B, t_2^B, \dots, t_M^B\}$  werden die Zeitstempelmengen der Sensoren A und B für das eben betrachtete Segment bezeichnet und  $N$  steht für die Länge der Segmente gemessen in Anzahl der Messung von Sensor A. Somit ergibt sich eine zeitliche Segmentlänge  $\Delta t = \frac{N}{f^A}$ . Nachdem das Signal B interpoliert wird, müssen für das Signal B sowohl Messungen früher als die erste Messung von Sensor A als auch Messungen später als die letzte Messung von Sensor B vorliegen. Daraus folgt, dass  $M > (\Delta t f^B + 2)$  und  $t_1^B < t_1^A$  und  $t_M^B > t_N^A$  gelten muss. Mit zwei zeitkontinuierlichen Signalen lässt sich das Problem der TDE wie folgt formulieren

$$\delta = \operatorname{argmax}_{\delta} (f(s^A(t), s^B(t + \delta))), \quad (7.1)$$

wobei  $f$  eine geeignete Vergleichsoperator ist.

In Falle von zeitdiskreten Signalen muss ein Signal interpoliert werden. Angenommen es soll das Signal B zum Zeitpunkt  $t_j^A$  interpoliert werden. Dann müssen zunächst zwei Zeitstempel  $t_i^B$  und  $t_{i+1}^B$  mit  $t_i^B \leq t_j^A \leq t_{i+1}^B$  gefunden werden und damit dann eine Interpolation des Signals berechnet werden

$$\tilde{s}_j^B(t) := s_i^B + \frac{s_{i+1}^B - s_i^B}{t_{i+1}^B - t_i^B} (t - t_i^B) \quad \text{mit } t_i^B \leq t \leq t_{i+1}^B. \quad (7.2)$$

Somit ist  $\tilde{s}_j^B(t)$  die lineare Interpolation von  $s_i^B$  zum Zeitstempel  $t_j^A$ . Definiert ist diese Interpolante für den Zeitraum  $t_i^B \leq t \leq t_{i+1}^B$ . Das Optimierungsproblem lässt sich dann schreiben als:

$$\delta = \operatorname{argmax}_{\delta} \left( f_{j \in \text{TS}^A} (s_j^A, \tilde{s}_j^B(t_j + \delta)) \right). \quad (7.3)$$

Diese Methode erlaubt die relative Latenz mit einer beliebigen Auflösung zu bestimmen, und somit ist es auch möglich, das Verfahren noch präziser zu steuern, um weitere unnötige Berechnungen zu verhindern. Andere Arbeiten wie etwa [52, 73] setzen nicht auf eine Interpolation der Daten, sondern bestimmen die relative Latenz wie in Abschnitt 7.2.1 beschrieben. Doch statt wie eben beschrieben die Signale zu interpolieren, wird die so gewonnene Korrelationskurve interpoliert. Das Maximum wird dann auf der angenäherten Parabel gesucht. Andere Arbeiten ersetzen die Parabel durch eine Gaußsche Kurve wie Zhang in [116]. Diese Art der Bestimmung der relativen Latenz hat sich bei Vorversuchen als nicht belastbar gezeigt, da die vorherrschenden Fehler systematischer Natur sind. Die Methode der Näherung mit einer Parabel eignet sich um die Güte der Latenzbestimmung abzuschätzen. Vergleiche hierzu den Paragraph 7.4.2 auf Seite 100.

## 7.3 Ähnlichkeitsmaße

Wie bereits erwähnt, müssen die Signale während sie gegeneinander verschoben werden, fortlaufend miteinander verglichen werden. Bei Eintreten der größten Ähnlichkeit der Signale, wird die gewählte Verschiebung als eine Schätzung für die relative Latenzen wertet. Im Folgenden werden verschiedene Maße vorgestellt, wobei der Fokus auf der Korrelationsmethode liegt. Der Vollständigkeit halber seien die anderen Methoden auch vorgestellt, da diese in der Literatur ebenfalls Verwendung finden.

### 7.3.1 Korrelation

Die Korrelation beschreibt den Zusammenhang zwischen zwei oder mehreren statistischen Variablen. Werden zwei verschiedene statistische Variablen miteinander verglichen, spricht man von einer Kreuzkorrelation und wenn eine Variable mit sich selber verglichen wird von einer Autokorrelation. In dem Falle einer Zeitkalibrierung können auch Messwerte als Zufallsvariablen interpretiert werden.

**Definition** Die Quantifizierung dieses Zusammenhangs, also wie stark zwei Messreihen zusammenhängen, wird mit dem Korrelationskoeffizienten beschrieben, der wie folgt definiert ist

$$\rho_{X,Y} = \text{Corr}(X, Y) := \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{\text{E}((X - \mu_x)(Y - \mu_y))}{\sigma_X \sigma_Y}. \quad (7.4)$$

Der Korrelationskoeffizient wird auch Pearsons Korrelation oder Produkt-Moment-Korrelationskoeffizient genannt [111]. Die Korrelationsfunktion  $\text{Corr}(X, Y)$  zweier eindimensionaler Signale  $X$  und  $Y$  ist invariant gegenüber linearen Abbildungen der Signale  $X$  und  $Y$ . Mit anderen Worten ändert sich die Korrelation zweier Signale nicht, wenn diese Signale skaliert werden oder mit einem Summanden versehen werden. Seien  $\tilde{X} = \lambda X + \alpha$  und  $\tilde{Y} = \gamma Y + \beta$  die Bilder von  $X$  und  $Y$  unter zwei linearen Funktionen, dann gilt  $\text{Corr}(X, Y) = \text{Corr}(\tilde{X}, \tilde{Y})$ . Der Beweis kann

unter Nutzung der Linearität des Erwartungswertes ( $E[\lambda X + \alpha] = \lambda E[X] + \alpha$ ) [90] erbracht werden.

$$\begin{aligned}
\text{Corr}(\tilde{X}, \tilde{Y}) &= \frac{\text{Cov}(\tilde{X}, \tilde{Y})}{\sigma_{\tilde{X}}\sigma_{\tilde{Y}}} = \frac{E[(\tilde{X} - E[\tilde{X}])(\tilde{Y} - E[\tilde{Y}])]}{\sqrt{E[(\tilde{X} - E[\tilde{X}])^2]E[(\tilde{Y} - E[\tilde{Y}])^2]}} \\
&= \frac{E[(\lambda X + \alpha - E[\lambda X + \alpha])(\gamma Y + \beta - E[\gamma Y + \beta])]}{\sqrt{E[(\lambda X + \alpha - E[\lambda X + \alpha])^2]E[(\gamma Y + \beta - E[\gamma Y + \beta])^2]}} \\
&= \frac{E[(\lambda X + \alpha - \lambda E[X] - \alpha)(\gamma Y + \beta - \gamma E[Y] - \beta)]}{\sqrt{E[(\lambda X + \alpha - \lambda E[X] - \alpha)^2]E[(\gamma Y + \beta - \gamma E[Y] - \beta)^2]}} \quad (7.5) \\
&= \frac{\lambda\gamma E[(X - E[X])(Y - E[Y])]}{\lambda\gamma\sqrt{E[(X - E[X])^2]E[(Y - E[Y])^2]}} \\
&= \text{Corr}(X, Y)
\end{aligned}$$

Die lineare Unabhängigkeit macht die Korrelationsmethode robuster gegen Registrierungsfehler der Sensoren.

### Berechnung

Die Berechnung der Korrelation zweier diskreter Signale ist rechenaufwendig. Wird die Suche nach der relativen Latenz auf einer Signalreihe mit  $M$  Datenpunkten berechnet und insgesamt  $N$  verschiedene Zeitverschiebungen ausgewertet, so muss  $N$  mal eine Korrelation zweier Signale mit  $M$  Daten berechnet werden. Bei dem Zeitschiebverfahren ist es in erster Hinsicht wichtig, ein Maximum zu finden und nicht den korrekten Wert der Korrelation zu ermitteln. Dies kann am Ende der Optimierung erfolgen. Wenn es also nur auf die Stelle des Maximums ankommt, kann die Normierung der Korrelation ausgelassen werden. Der Normierungsfaktor ist  $\sigma_{\tilde{X}}\sigma_{\tilde{Y}}$  und daher positiv. In einem Optimierungsvorgang werden die beiden Signale in der Zeit gegeneinander verschoben, aber nicht selbst verändert, daher bleibt der Faktor  $\sigma_{\tilde{X}}\sigma_{\tilde{Y}}$  innerhalb eines Vorganges konstant. Daher ist es nicht nötig die Korrelation während des gesamten Optimierungsvorgang zu normieren. Darüber hinaus kann Rechenzeit gespart werden wenn die Signale so verschoben werden, dass der Erwartungswert gleich Null wird. Diese Verschiebung wird im Folgenden als gegeben betrachtet.

Damit kann die Bestimmung der relativen Latenz geschrieben werden als

$$\delta_{\text{DC}} = \text{argmax}_{\delta} \left( \frac{1}{N} \sum_{i=1}^N (s_i^A(iT + \delta)s_i^B(iT)) \right). \quad (7.6)$$

Diese Berechnung heißt in der Literatur auch der direkte Korrelator (engl.: *direct correlator*, *DC*) [52].

Neben dieser direkten Berechnungsmethode gibt es weitere Berechnungsmethoden. Es seien hier nur zwei weitere Methoden kurz vorgestellt. Die Funktion ASDF

(engl.: *average square difference function*) berechnet den durchschnittlichen Abstand über die quadrierte Differenz. Daher leitet sich auch der Name ab.

$$\delta_{\text{ASDF}} = \operatorname{argmin}_{\delta} \left( \frac{1}{N} \sum_{i=1}^N [s_i^A(iT + \delta) - s_i^B(iT)]^2 \right). \quad (7.7)$$

Die Funktion AMDF (engl.: *average magnitude difference function*) ist eine weitere Ähnlichkeitsfunktion, welche die durchschnittliche Größe der Differenz berechnet. Diese Funktion kommt ohne jegliche Multiplikation oder Quadratur aus und kann somit sehr effizient berechnet werden.

$$\delta_{\text{AMDF}} = \operatorname{argmin}_{\delta} \left( \frac{1}{N} \sum_{i=1}^N |s_i^A(iT + \delta) - s_i^B(iT)| \right). \quad (7.8)$$

Betrachtet man die beiden letzten Berechnungsmethoden so fällt auf, dass diese ähnlich sind zur Minimierung des geometrischen Abstands zweier Signale. Auch der geometrische Abstand zweier Signale lässt sich als Vergleichsmaß von Signalen verwenden. Vergleiche hierzu den folgenden Abschnitt 7.3.2. Die Berechnungsfunktionen ASDF und AMDF operieren auf Signalen mit nur einer Dimension, während geometrische Vergleichsfunktionen auf den räumlichen und dadurch mehrdimensionalen Signalen operieren.

Die verschiedenen Korrelationsfunktionen wurden in der Literatur schon mehrfach auf das Verhalten unter verschiedenen SNR Werten und Anzahl der zugrunde liegenden Datenpunkte untersucht. Vergleiche hierzu beispielsweise die Ergebnisse von Fertner in [27]. In diesem Artikel wird gezeigt, dass die sehr effizient zu berechnende AMDF Funktion nicht signifikant schlechtere Ergebnisse liefert als die DC Funktion.

### 7.3.2 Geometrisch

Das geometrische Ähnlichkeitsmaß stellt in diesem Rahmen eine Besonderheit dar, denn es operiert nicht wie die anderen Verfahren auf den dimensionsreduzierten Signalen, sondern direkt auf den geometrischen Messgrößen. Dadurch fällt der Prozessschritt der Dimensionsreduktion weg. Dies führt zum einen zu einer kürzeren Rechenzeit und schließt zum anderen eine Fehlerquelle aus. Durch die speziellen statistischen Vergleichsoperatoren, wie Korrelation, haben aber geometrische Fehler, verursacht durch ungenaue Kalibrierung oder Messfehler, geringeren Einfluss.

In der Literatur werden geometrische Vergleichsoperatoren für die Zeitkalibrierung schon verwendet. Lieberknecht u. a. beschreiben in [65], wie sie Referenzdaten für markerloses Tracking aufgenommen haben. Nachdem zwei verschiedene Sensoren benutzt wurden und es auf höchste Genauigkeit ankommt, ist von Lieberknecht auch eine zeitliche Kalibrierung zwischen diesen Sensoren vorgenommen worden. Er benutzte aber nicht den in dieser Arbeit vorgeschlagenen Prozess, sondern kombinierte bei seinem Ansatz die räumliche und zeitliche Kalibrierung in einem einzigen Schritt. Es wurde also neben den sechs Freiheitsgraden der räumlichen

Kalibrierung auch noch ein siebter zeitlicher Parameter für die relative Latenz bestimmt. Lieberknecht hat für die Kalibrierung der relativen Latenz ebenfalls den Prozessschritt der Synchronisierung mit Hilfe der Interpolation und das Timeshift Verfahren genutzt. Andere Schritte wie Segmentierung waren bei seinem Ansatz nicht nötig.

Schwald u. a. beschreiben in [93] ebenfalls eine geometrische Methode um die relative Latenz zwischen zwei Sensoren zu bestimmen. In der konkreten Anwendung wurde ein elektromagnetisches Trackingsystem und ein optisches auf Infrarotlicht basierendes System sowohl räumlich als auch zeitlich kalibriert. Ähnlich zu dem Ansatz von Lieberknecht wurde die Kalibrierung der relativen Latenz zum Zeitpunkt der räumlichen Kalibrierung durchgeführt. Um den optimalen zeitlichen Versatz zu finden, wurde dabei versucht das Residuum der räumlichen Kalibrierung zu minimieren. Die Messreihen wurden nicht mit Hilfe von Interpolation in der Zeit verschoben und neu auf einander synchronisiert, sondern es wurden Punktkorrespondenzen für die Kalibrierung ausgewählt. Die Autoren schreiben, dass durch diese Methode die Anzahl der Punkte für eine Kalibrierung dramatisch zurückgehen, und trotzdem liefert das Verfahren reproduzierbare Ergebnisse ( $\mu = 207$  ms;  $\sigma = 8,6$  ms).

Beide hier vorgestellten Verfahren benutzen geometrische Normen, um die relative Latenz zu ermitteln. Auch wird bei beiden Verfahren die Kalibrierung der räumlichen und zeitlichen Parameter zusammengefasst. Mit dem dynamischen online Einsatz während der Laufzeit hat sich keine der beiden Arbeiten befasst. Um zu zeigen, dass die geometrische Methode auch auf kleinen Datensätzen zuverlässige Ergebnisse liefert, wird diese Methode hier kurz vorgestellt. Anschließend wird das Ergebnis eines Versuchs vorgestellt, der die zeitliche Kalibrierung erst mit der Korrelation als Maß und dann mit der geometrischen Norm ermittelt.

Sei nun  $d(x, y) = \|x - y\|_2$  der Euklidische Abstand zweier Punkte im  $\mathbb{R}^3$  und  $d(q_x, q_y) = \|q_x - q_y\|_2$  eine Norm für Differenzen von Orientierungen (vgl. Abschnitt 2.3.2). Seien nun also allgemein  $s^A(t)$  und  $s^B(t)$  zwei mehrdimensionale Signale, welche zu dem Zeitpunkt  $iT$  mit  $s_i^A(iT)$  und  $s_i^B(iT)$  vorliegen bzw. interpoliert wurden. Die Differenz der Signale kann dann mit  $d_i(s_i^A(iT), s_i^B(iT))$  für einen Zeitpunkt berechnet werden. Um einen skalaren Wert für ein gesamtes Segment zu erhalten, wird der *RMS* über die einzelnen Werte berechnet. Damit kann die Bestimmung der relativen Latenz als Minimierungsproblem formuliert werden

$$\delta = \operatorname{argmin}_\delta \left( \sqrt{\frac{\sum_{i=1}^N (d(s_i^A(iT + \delta), s_i^B(iT)))^2}{N}} \right). \quad (7.9)$$

Die hierzu nötigen Synchronisierungs- und Interpolationsmethoden können dem Kapitel 6 entnommen werden.

Um zu zeigen, dass beide Vergleichsfunktionen analoge Resultate liefern, wurde eine Kalibrierung sowohl mit der Korrelation durchgeführt, als auch mit dem geometrischen Abstand als Vergleichsfunktion. Der Versuch besteht aus zehn Segmenten zu je 240 Messungen mit einer Überlappung von jeweils 60 Messungen.

Als Sensoren wurden ein FaroArm und ein A.R.T. verwendet, wobei für die Registrierung nur die Position genutzt wurden. Die räumliche Registrierung zwischen FaroArm und A.R.T. wurde hochgenau mit der Messsoftware des FaroArms durchgeführt. Bei beiden Versuchen wurde eine relative Latenz von  $\delta = 32,0$  ms ermittelt, wobei die Standardabweichung bei Verwendung der geometrischen Vergleichsmethode  $\sigma = 0,08$  ms marginal größer ist als die Standardabweichung  $\sigma = 0,07$  ms bei dem Vergleich mit der Korrelation.

In Abbildung 7.1 sind die entsprechenden Zeitversatzkurven dargestellt. Bemerkenswert sind die unterschiedlichen Formen der Kurven. Während bei Verwendung der geometrischen Methode die Kurve nahezu linear bis zum Minimum verläuft, folgt die Form der zeitlichen Korrelationskurve eher einer Parabel. Das Optimum ist in diesem Beispiel bei einem Vergleich mit der geometrischen Methode klarer zu erkennen als mit der Korrelationsmethode. Es bleibt noch zu untersuchen, ob diese positive Eigenschaft des geometrischen Vergleichs auch in weiteren Fällen zu beobachten ist, wie beispielsweise bei einer fehlerhaften Registrierung der Sensoren.

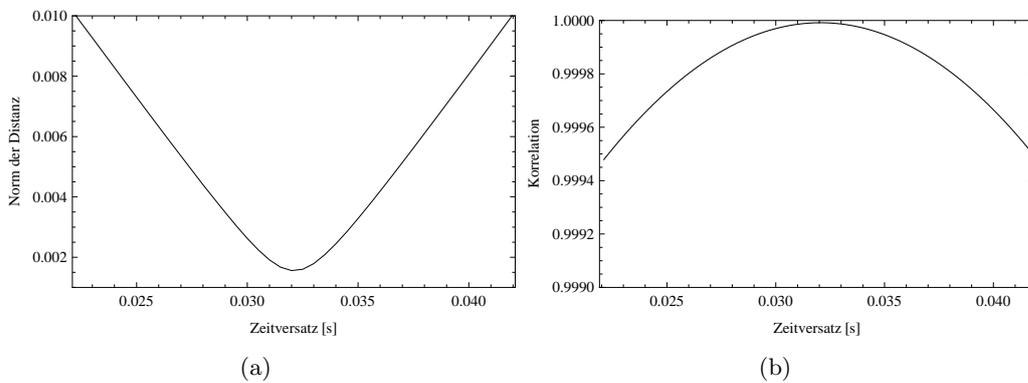


Abbildung 7.1: Darstellung der Zeitversatzkurve (a) unter Verwendung des geometrischen Abstands als Vergleichsfunktion und (b) unter Verwendung der Korrelation als Vergleichsfunktion. Der optimale Zeitversatz ist gefunden, wenn der Abstand der beiden Signale minimal bzw. die Korrelation maximal wird.

## 7.4 Unsicherheiten der TDE

Wenn eine Kalibrierung vorgenommen wird, ist es für den Anwender von großem Interesse, wie genau diese Kalibrierung war. Dies ist bei einer zeitlichen Kalibrierung nicht anders, denn auch hier gibt es eine Reihe von Störeinflüssen, die eine Kalibrierung beeinträchtigen können. Neben dem Anwender ist es auch für manche Teilprozesse des Verfahrens wichtig, dass ein Maß für die Güte der Schätzungen existiert. Ein Beispiel hierfür ist das „Power of Two-Choice“-Verfahren für die Dimensionsreduktion (siehe Abschnitt 5.3.2).

In den folgenden Abschnitten wird kurz gezeigt, dass eine vom angewendeten Verfahren unabhängige theoretische untere Schranke für die Genauigkeit existiert. Danach werden zwei Vorschläge für ein Maß vorgestellt. Dazu werden im ersten Fall die Parameter einer Parabel gesucht, so dass diese Parabel mit der Zeitverschiebungskurve übereinstimmt. Als weiteres Maß wird der normalisierte Kreuzkorrelationskoeffizient vorgeschlagen, und es wird anschließend gezeigt, dass diese zwei Maße dieselbe Information liefern.

### 7.4.1 Cramér-Rao Schranke

Die Cramér-Rao Schranke (engl.: *Cramér-Rao-Bound*; *CRB*) liefert eine untere Schranke für die Varianz eines Schätzers eines deterministischen Parameters. Benannt wurde diese Schranke nach den beiden Mathematikern Harald Cramér und C. R. Rao. Die Cramér-Rao Schranke ermöglicht es, eine untere Schranke für die Bestimmung der relativen Latenz anzugeben. Diese Schranke ist eine rein theoretische Schranke, die nicht von den verwendeten Algorithmen oder gar deren Implementierung abhängt, sondern alleine von den Eingangsdaten, deren Unsicherheit und den verwendeten Sensoren. Somit kann die Genauigkeit der Bestimmung der relativen Latenz in Abhängigkeit der Sensoren bestimmt werden, und es können anschließend die Algorithmen gegen diese untere Schranke getestet werden.

Die Cramér-Rao Schranke kann für viele Situationen hergeleitet werden. Im Folgenden ist sie für die Bestimmung der relativen Latenz von Patwari [76] dargestellt. Diese Arbeit beschäftigt sich mit der kooperativen Lokalisierung von drahtlosen Sensoren. Die Lokalisierung der verschiedenen Knoten geschieht mit Hilfe von Laufzeitunterschieden der Funksignale und damit im Prinzip auch für die Bestimmung von Latenzen.

Die Formel 7.10 zeigt die von Patwari hergeleitete Cramér-Rao Schranke

$$\sigma_{\text{TDE}}^2 \geq \frac{1}{8\pi^2 B F_c^2 T \text{SNR}}. \quad (7.10)$$

$B$  steht für die Signalbandbreite des Signals in Hz,  $F_c$  beschreibt die mittlere Frequenz in Hz, der Parameter  $T$  steht für die Beobachtungsdauer in Sekunden und  $\text{SNR}$  stellt das Signal-Rausch-Verhältnis dar. Mit Hilfe dieser Formel 7.10 lassen sich die Zusammenhänge der einzelnen Faktoren studieren und eine Intuition dafür gewinnen, wie diese zusammenhängen. So kann beispielsweise eine Halbierung des Signal-Rausch-Verhältnisses durch eine Verdoppelung der Beobachtungszeit kompensiert werden.

Doch die Cramér-Rao Schranke, wie sie in der Formel 7.10 dargestellt ist, erfasst nicht die Einflüsse von Kalibrierfehlern oder von ungenauen Sensoruhren. Es bleibt zu prüfen, wie solche Parameter wie Kalibrierfehler oder andere in diese Betrachtung mit aufgenommen werden können, um diese Abschätzung noch realistischer zu gestalten. Aber auch ohne diese Einflüsse bietet die von [76] präsentierte Abschätzung einen guten Ausgangspunkt für die Abschätzung der zu erwartenden Ungenauigkeiten der Schätzung der relativen Latenz.

### 7.4.2 Mögliche Maße für die Sicherheit der Schätzung

Die Cramér-Rao Schranke gibt eine untere Schranke für die Varianz der Schätzung an. In der Praxis wird die Genauigkeit der Schätzung der relativen Latenzen von weit mehr Faktoren beeinflusst als in der Formel 7.10 zu sehen. Für eine realistische Schätzung müssten Parameter, wie in Kapitel 4 aufgeführt, berücksichtigt werden. Die Cramér-Rao Schranke spiegelt auch nicht die Ungenauigkeit eines speziellen Verfahrens wider, sondern nur die theoretische untere Schranke eines jeden Verfahrens. Somit ist diese Schranke für den Vergleich einzelner Verfahren von Interesse, kann aber nicht genutzt werden, um die Sicherheit einer einzelnen Schätzung zu bestimmen.

Um aber bei einem adaptiven Verfahren, wie dem Two-Choice-Verfahren (vgl. Abschnitt 5.3.2), die Steuerung des Verfahrens zu ermöglichen, wird ein Maß für die Güte der aktuellen Schätzung benötigt. Es ist möglich mit einer Reihe von Schätzungen der relativen Latenz eine Varianz der Schätzungen berechnen, welche dann Rückschlüsse auf die Genauigkeit zulässt. Dies ist auch sinnvoll, um dem Benutzer des Systems eine Rückmeldung über die Sicherheit der zeitlichen Kalibrierung zu geben. Für Verfahren, wie dem Two-Choice-Verfahren, ist es aber nötig, für jede Schätzung der relativen Latenz eine Aussage über die Güte treffen zu können.

Im Folgenden wird ein Vorschlag für ein solches Gütemaß entwickelt. Das Maß besteht dabei aus zwei Teilen. Zum einen werden über die Form der Korrelationskurve Informationen gewonnen und zum anderen kann auch der Korrelationskoeffizient Auskunft über die Güte der Schätzung geben.

**Parabelmethode** In Arbeiten von [52, 116] werden Parabeln genutzt, um bei zeitlich diskreten Signalen eine bessere zeitliche Auflösung der Zeitversatzschätzung als die Samplingrate  $T$  zu erhalten. Die Parabel ist demnach in der Nähe der korrekten relativen Latenz  $\delta$  eine Näherung der Korrelationskurve. In [18] wird die Ableitung der Korrelationskurve mit der Sicherheit der Schätzung in Verbindung gebracht. Es liegt somit nahe, eine Parabel zu nutzen, um die Breite der Korrelationskurve nahe des Maximums zu untersuchen und somit auch Rückschlüsse auf die Sicherheit der Schätzung zu ziehen.

Wie stark die Krümmung des Graphs einer Parabel ist, wird durch den Parameter  $a$  der folgenden quadratischen Funktion bestimmt

$$y = ax^2 + bx + c. \quad (7.11)$$

Für die Näherung der Parabel an die Daten der Korrelationskurve müssen die Parameter der quadratischen Form bestimmt werden. Diese Bestimmung kann mit einem beliebigen Optimierungsverfahren bestimmt werden [94].

Die Parabel ist stärker gekrümmt, je größer der Wert des Parameters  $|a|$  wird. Der Parameter  $a$  lässt aber neben der Breite der Parabel und somit der Unsicherheit der Schätzung noch weitere Rückschlüsse über die aktuelle Schätzung zu. So muss bei Nutzung der normierten Korrelationsfunktion als Vergleichsfunktion der

Parameter  $a$  negativ sein, damit die Parabel ein Maximum und kein Minimum hat. Ein Wert  $a = 0$  bedeutet, dass die Ausgangsfunktion eine Gerade ist und sich somit kein Maximum finden lässt.

Neben dieser Form der Klassifizierung kann auch die Abweichung der Korrelationskurve von der Parabelform genutzt werden, um die Güte der Schätzung zu beurteilen. Eine mögliches Vorgehen hierfür ist die Berechnung des Residuums zwischen diesen beiden Kurven. Das Residuum wird in der Regel schon bei der Optimierung der Parameter der angenäherten Parabel berechnet. Es stellt somit keinen zusätzlichen Rechenbedarf dar.

Treten, wie in Abschnitt 4.3 beschrieben, Ausreißer auf, so kann die Korrelationskurve eine Gestalt wie in Abbildung 4.5 haben. Bei diesem Beispiel ist noch eine Parabelform zu erkennen von welcher auch ein Maximum bestimmt werden kann. Treten jedoch noch weitere Störquellen wie starkes Rauschen, große Registrierungsfehler oder andere Einflüsse auf, dann kann die Form der Korrelationskurve von der idealen Parabelform stark abweichen. Eine Darstellung eines entarteten Falls ist in Abbildung 7.2 zu sehen. Dies zeigt, dass eine Schätzung der relativen Latenz nicht unbesehen übernommen werden darf. Es muss geprüft werden, ob beispielsweise innerhalb des Suchfensters ein Maximum der Korrelation auftritt oder ob das Maximum außerhalb des Suchbereichs liegt und somit der vorliegende maximale Wert kein Maximum der Korrelation ist.

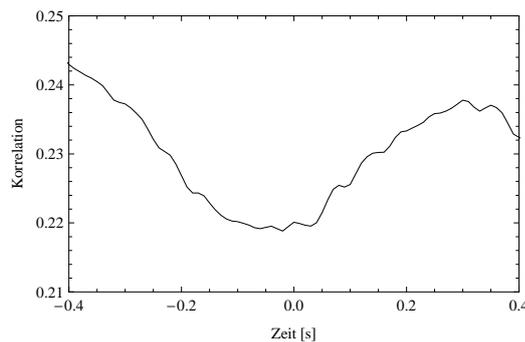


Abbildung 7.2: Korrelationskurve, die durch eine ungenaue Kalibrierung gestört wurde. Die korrekte relative Latenz liegt bei diesem Beispiel bei  $\delta = -0,13$  ms.

Durch die Näherung der Korrelationskurve als Parabel können somit mehrere Hinweise für die Güte einer Schätzung gewonnen werden. Auch die Krümmung der Korrelationskurve, also der Parameter  $a$  in der quadratischen Form, liefert einen Hinweis auf die Güte der Schätzung. Ist eine Korrelationskurve flach, dann rührt die wahrscheinlich von zu wenig Bewegung der Sensoren her. Anhand des Vorzeichens der Parameters  $a$  kann zusätzlich erkannt werden, ob die Parabel nach unten oder oben geöffnet ist. Schätzungen von Segmenten, die eine nach oben geöffnete Parabel haben, können damit verworfen werden. Weiterhin gibt das Residuum die Abweichung der Korrelationskurve von einer quadratischen Form

an. Damit können entartete Korrelationskurven, wie in Abbildung 7.2 zu sehen, erkannt werden und dann verworfen werden.

Aber der berechnete normierte Korrelationskoeffizient, welcher dem Scheitelpunkt der Parabel entspricht, kann ebenfalls als ein Maß für die Güte des Schätzung interpretiert werden. Der Korrelationskoeffizient beschreibt den Grad des linearen Zusammenhangs zweier Signale. Je stärker der Korrelationskoeffizient gegen Eins geht, desto stärker ist dieser der Zusammenhang. Ein Wert gleich Eins heißt somit, dass die beiden Signale durch Skalierung und Verschieben zur genau Deckung gebracht werden können. Durch Rauschen wird in der Praxis verhindert, dass der Korrelationskoeffizient genau den Wert Eins annimmt. Aber ein Wert, der stark von dem Wert Eins abweicht, bedeutet, dass die beiden Signale sich signifikant unterscheiden. Grund für einen kleinen Korrelationskoeffizienten kann beispielsweise eine fehlerhafte räumliche Registrierung sein. Der Zusammenhang von Korrelationskoeffizienten und räumlicher Registrierung nutzt Johansson in [54], um für eine Dimensionsreduktion mehrdimensionaler Signale die Projektionsvektoren mit einem Gradienten-Suchalgorithmus zu bestimmen. Zur Steuerung des Gradienten-Suchalgorithmus nutzt Johansson den normierten Korrelationskoeffizient.

Für einen empirischen Nachweis der Korrektheit dieser beiden Verfahren, bedarf es einer Monte-Carlo-Simulation wie in Kapitel 8 beschrieben. Denn ein Fehleruntersuchung lässt sich nur dann korrekt durchführen, wenn die exakte relative Latenz zwischen den Sensoren bekannt ist.

## 7.5 Aggregation

In den vorangegangenen Abschnitten wurden die Methoden dargestellt, um von einem Segment mit definierter Länge eine Schätzung für die relative Latenz zu erhalten. Wird dies für mehrere sequentielle Segmente durchgeführt, liefert dies eine Reihe von Schätzungen der relativen Latenz  $\delta_i$ . Bewertet man die Ergebnisse mit einem der vorgestellten Maße, dann erhält man neben den geschätzten Latenzen auch ein Gewicht  $w_i$ . Um nun eine Aussage über die relative Latenz zwischen den beiden Sensoren treffen zu können, müssen diese gewonnenen Ergebnisse zusammengefasst werden. Verschiedene Möglichkeiten für diese Aggregation sollen im folgenden Abschnitt aufgezeigt werden. Neben der Aggregation der Schätzungen der Latenz, soll zudem die Varianz des Ergebnisses angegeben werden, damit der Benutzer des Systems über die Sicherheit der Schätzung informiert werden kann. Um die Adaptivität der Bestimmung der relativen Latenz nicht zu gefährden, dürfen nur eine bestimmte Anzahl an Ergebnissen zusammengefasst werden. Die konkrete Anzahl an zu bewerteten Segmenten  $N$  festzulegen ist eine Herausforderung, da ein Kompromiss zwischen Adaptivität und Stabilität getroffen werden muss. Die aktuelle Schätzung der relativen Latenz beruht somit immer auf den letzten  $N$  Segmenten.

**Arithmetisches Mittel** Die wohl einfachste Methode ist das Mittel der letzten  $N$  Schätzungen zu nehmen. Dabei wird dann das bestimmte Maß vernachlässigt und nur das gewichtete Mittel der Schätzungen  $\delta_i$  und deren Standardabweichung bestimmt [15].

$$\mu_\delta = \frac{1}{N} \sum_{i=1}^N \delta_i \quad (7.12)$$

$$\sigma_\delta^2 = \frac{1}{N-1} \sum_{i=1}^N (\delta_i - \mu_\delta)^2. \quad (7.13)$$

Der Einsatz des ungewichteten Mittels verwirft jedoch die Informationen über die Genauigkeit der einzelnen Schätzungen. Anstatt des einfachen arithmetischen Mittels kann auch das gewichtete Mittel eingesetzt werden, um die Informationen über die einzelnen Messungen nicht ignorieren.

**Gleitendes Mittel** Das gewichtete arithmetische Mittel ist eine einfache Methode, um die Schätzungen der einzelnen Segmente zu mitteln. Nach Rinne [86] ist das gewichtete arithmetische Mittel und die gewichtete Standardabweichung definiert durch

$$\mu_\delta = \frac{1}{\sum_{i=1}^N w_i} \sum_{i=1}^N w_i \delta_i \quad (7.14)$$

$$\sigma_\delta^2 = \frac{1}{\sum_{i=1}^N w_i} \sum_{i=1}^N w_i (\delta_i - \mu_\delta)^2. \quad (7.15)$$

Dabei steht  $\delta_i$  für die Schätzungen und  $w_i$  für die Gewichte, welche die Güte der Schätzung repräsentieren.

**Median** Der Median ist eine weitere Methode, um die Vielzahl der Schätzungen zusammenzufassen. Dabei werden die Sicherheiten der einzelnen Schätzungen, wie im Falle des arithmetischen Mittels, nicht berücksichtigt. Angenommen die geschätzten Latenzen  $\delta_i$  sind sortiert, es gelte somit  $\delta_i < \delta_{i+1}$  für  $1 \leq i < N$ , dann ist der Median wie folgt definiert

$$\tilde{\delta} = \begin{cases} \delta_{\frac{n+1}{2}} & n \text{ ungerade} \\ \frac{1}{2} (\delta_{\frac{n}{2}} + \delta_{\frac{n}{2}+1}) & n \text{ gerade.} \end{cases}$$

Der Median ist robuster gegenüber Ausreißern als das einfache arithmetische Mittel [90]. Einzelne Ausreißer können somit das Ergebnis nicht verfälschen. Es ist jedoch darauf zu achten, dass ein einzelner Ausreißer in den Messwerten zu mehreren Ausreißern in den Schätzungen für die Latenzbestimmung führen kann. Dies gilt insbesondere für überlappende Segmente (vgl. Kapitel 6.3), da bei überlappenden Segmenten ein Ausreißer in mehreren Segmenten enthalten sein kann.

## 7.6 Zusammenfassung

Dieses Kapitel hat eine umfassende Darstellung der in der Literatur vorhandenen Methoden für die zeitliche Kalibrierung zweier Sensoren gegeben. Dabei wurde Methode der Kreuzkorrelation aus dem Gebiet der Signalverarbeitung für Anforderungen von Augmented Reality Anwendungen adaptiert. Diese Methode wird von anderen Forschergemeinschaften schon seit langem für das Bestimmen von „time delays“ benutzt. Neben der Kreuzkorrelation als Vergleichsfunktion wurde auch ein geometrisches Verfahren beleuchtet.

Für eine abschließende Beurteilung der vorgenommenen Schätzungen bedarf es eines Maßes für die Qualität bzw. die Unsicherheit der Schätzung. In dem letzten Teil des Kapitels wurde ein pragmatischer Vorschlag für die Lösung dieses Problems formuliert. Diese Lösung ermöglicht es Schätzungen während des Trackingvorgangs, also während die AR Anwendung läuft, einzuordnen und den Zeitversatz somit robust zu schätzen.

# 8 Monte-Carlo-Simulation für Zeitkalibrierung

In diesem Kapitel wird die Monte-Carlo-Simulation (MCS) vorgestellt, und wie diese für die Evaluation von Zeitkalibrierungs-Verfahren benutzt werden kann. Nach einem kurzen Überblick werden Arbeiten beschrieben, welche die Monte-Carlo-Simulation benutzen. Anschließend wird ein Modell vorgeschlagen und anhand eines Beispiels erläutert. Dabei wird auch auf die Sicherheit der Simulationsergebnisse eingegangen und angegeben wie Konfidenzintervalle für die Sicherheit einer Schätzung bestimmt werden können.

Diese Ergebnisse fließen dann in die Beschreibung eines Softwarepaketes ein, welches es erlaubt die Zeitkalibrierung mit einem Monte-Carlo-Ansatz zu simulieren. Dabei können verschiedene Störeinflüsse wie Rauschen der Sensoren oder auch Fehler in der Kalibrierung hinsichtlich der zu erwartenden Kalibrierergebnisse untersucht werden. Aber auch verschiedene Techniken für die Kalibrierung der relativen Latenz können mit Hilfe dieser Software schnell evaluiert werden.

## 8.1 Einführung und Related Work

Monte Carlo Simulationen sind eine Familie von Berechnungsalgorithmen, die wiederholte randomisierte Werte benutzen, um ein Ergebnis zu berechnen. Das Verfahren wird häufig zur Lösung physikalischer und mathematische Probleme benutzt, für die keine oder nur schwer berechenbare direkte Lösungen existieren. Das Verfahren geht auf eine Idee von Enrico Fermi im Jahre 1930 zurück und wurde später maßgeblich von Stan Ulam und John von Neumann 1945 ausgearbeitet. Erstmals wurde die Methode benutzt, um die Streuung von Neutronen in spaltbaren Material zu simulieren [69].

In vielen Bereichen wird diese Methode auch verwendet, um zu untersuchende Modelle vorab zu simulieren und damit die Anzahl benötigter realer physikalischer Experimente zu reduzieren. Oft wird die Monte Carlo Simulation aber auch zum Lösen von Problemen eingesetzt, für die es keine oder nur sehr komplizierte explizite Lösungen gibt – oder auch um Teilaspekte eines großen Gesamtsystems zu untersuchen.

Im Bereich der Time Delay Estimation sind Simulationen üblich um Methoden auf ihre Wirksamkeit und Genauigkeit zu untersuchen. So haben Fertner und Sjölund in [27] verschiedene Korrelationsfunktionen evaluiert. Dazu wurde ein Signal generiert und dieses mit zwei unabhängigen Rauschsignalen überlagert und um einen festen zeitlichen Offset  $\delta$  verschoben. Mit den in Kapitel 7 vorgestellten

Methoden wurden für diese zwei gewonnen Signale dann mit fünf verschiedenen Berechnungsmethoden der Korrelation die vorher festgelegte Offset bestimmt. Die gewonnenen Ergebnisse wurden anhand der Varianz der Ergebnisse bewertet. Das Resultat dieser Arbeit war, dass die sehr effizient zu berechnende Vergleichsfunktion *AMDF* keine signifikant schlechteren Ergebnisse liefert, als die aufwendig zu berechnende direkte Korrelation *DC*. Für eine Erklärung der verschiedenen Funktionen siehe Abschnitt 7.3.1 oder die Arbeiten von Fertner [27] und Jacovitti [52].

Auch im Bereich der Optik wird die Methode der Monte-Carlo-Simulation benutzt um beispielsweise ein photogrammetrisches Kamerasystem hinsichtlich unterschiedlicher Kameramodelle, Objektraum und Beleuchtung zu untersuchen [39]. Gerade die Simulation komplexer Systeme ist die Stärke von Monte-Carlo-Simulationen, denn sie bieten die Möglichkeit die einzelnen Subsysteme getrennt voneinander zu betrachten und jeweils optimale Lösungen zu finden und deren Auswirkung auf das Gesamtsystem zu bestimmen [74]. Diese Möglichkeit wird als „one-factor-at-a-time“ Ansatz bezeichnet. Die Methode vernachlässigt jedoch jegliche Abhängigkeiten zwischen den einzelnen Faktoren. Werden hingegen alle Faktoren eines Systems in einer Simulation variiert, können alle Aspekte und Abhängigkeiten eines Systems dargestellt werden. Jedoch ist diese Methode in den meisten Fällen zu komplex zu berechnen. Sollen in einer Simulation  $N$  Parameter abgebildet werden und für jeden Parameter werden  $M$  Stichproben benötigt, dann werden insgesamt  $M^N$  Kombinationen von Stichproben nötig sein. Die Komplexität der Simulation wächst exponentiell mit der Anzahl der zu simulierenden Parameter. Bei dem „one-factor-at-a-time“ Ansatz sind unter den gleichen Randbedingungen nur  $NM$  Kombinationen nötig. In der Praxis ist ein Kompromiss zwischen diesen Methoden zu wählen, so dass genügend Abhängigkeiten zwischen den Parametern abgebildet werden können, das Modell jedoch noch in der Komplexität beherrschbar bleibt. Für die Zeitkalibrierung können die einzelnen Verfahren auf einzelne Störeinflüsse hin untersucht werden.

## 8.2 Monte-Carlo-Modell

In dieser Arbeit wird ein Modell benutzt, welches unter anderem von Cox in [20] beschrieben wird. Cox et al. beschreiben und adaptieren darin das Verfahren der Monte-Carlo-Simulation für die Messtechnik. Die Autoren teilen das Verfahren in drei Phasen auf.

1. In eine Vorbereitungsphase, in welcher das Berechnungsmodell und die Wahrscheinlichkeitsverteilungen festgelegt werden.
2. In eine Durchführungsphase, in welcher die eigentliche Simulation durchgeführt wird.
3. In eine Auswertungsphase, in der die Ergebnisse berechnet und statistisch ausgewertet werden.

In Abbildung 8.1 ist eine Übersicht des von Cox vorgeschlagenen Modells zu sehen.

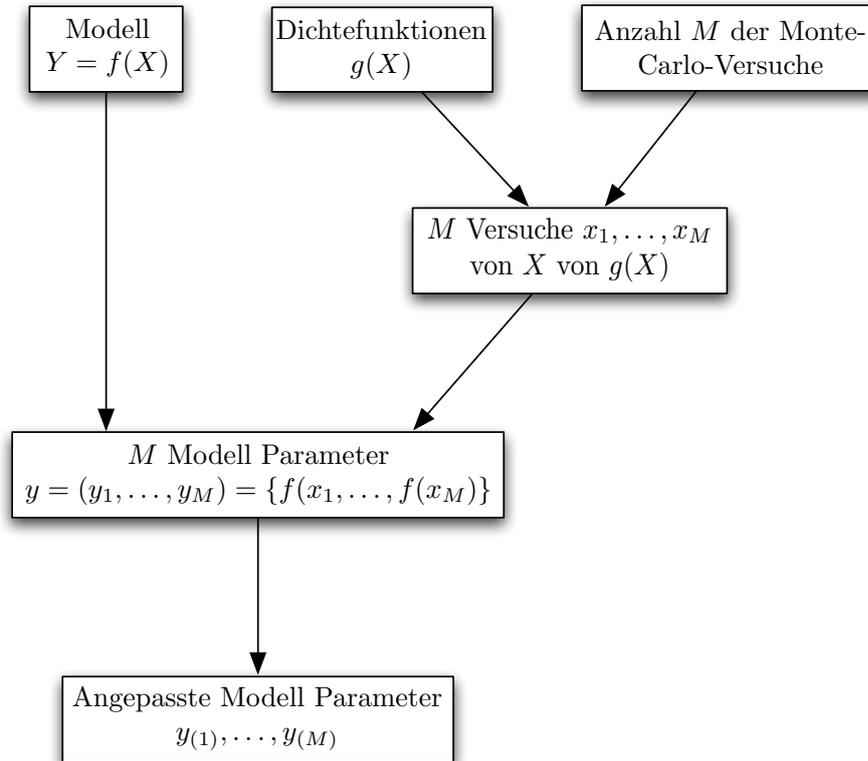


Abbildung 8.1: Monte-Carlo Simulationsprozess nach Cox [20].

Die folgenden Absätze illustrieren das Prinzip der Monte-Carlo-Simulation anhand eines kleinen Beispiels. Die Kreiszahl  $\pi$  wird mit Hilfe einer MCS berechnet. Dabei werden die wichtigsten Teile der Methode erläutert. Für die Simulation der Zeitkalibrierung werden zwar etwas andere Methoden benötigt, doch unterscheiden sich diese nur in Details von den hier vorgestellten.

Die erste Phase konzentriert sich auf die Vorbereitungen. Darunter fällt auch die Wahl der Wahrscheinlichkeitsfunktion der Zufallsvariablen  $X$ , die Formulierung des Berechnungsmodells und die Bestimmung der Eingabeparameter. Da vor der Durchführungsphase sollte auch festgelegt sein sollte wie viele Versuche durchgeführt werden, geschieht auch das in der Vorbereitungsphase. In dem Beispiel soll nun die Kreiszahl  $\pi$  berechnet werden. Dabei wird jeder der oben aufgeführten Schritte durchlaufen und ein Konfidenzintervall für diese Berechnung angegeben.

Die Grundidee hinter der Kreiszahlberechnung mit Monte-Carlo ist nicht neu und lässt sich kurz beschreiben. Es werden gleichverteilt sehr viele Punkte auf einem Quadrat mit Kantenlänge 1 ausgewählt. Zeichnet man jetzt einen Viertelkreis mit Radius 1 in das Quadrat ein, kann man Punkte daran unterscheiden, ob sie innerhalb des Kreises liegen oder außerhalb. Mit einer zunehmenden Anzahl von

Punkten nähert sich das Verhältnis der Anzahl der Punkte innerhalb des Viertelkreises zu der Anzahl aller Punkte im Quadrat dem Verhältnis der zugehörigen Flächen an. Das Verhältnis dieser Flächen entspricht  $\frac{1}{4}\pi$ .

**Phase Eins** Um die Punkte zufällig in dem Quadrat zu verteilen, muss erst eine Wahrscheinlichkeitsverteilung gewählt werden. Da die Punkte gleichmäßig auf einer Fläche verteilt werden sollen, werden für jede Dimension ein Zufallsvariablen  $X_1$  und  $X_2$  benötigt. Für diese beiden Zufallsvariablen wird die Gleichverteilung gewählt. Nach [86] ist eine Rechteckverteilung von  $X$  in dem hier vorliegenden Fall genau dann gegeben, wenn die Dichte gegeben ist durch

$$g(X_i) = \begin{cases} 1 & \|x\| \leq 1 \\ 0 & \|x\| > 1 \end{cases} \quad i \in \{1, 2\}.$$

Nach der Wahl der Dichte muss auch festgelegt werden wie viele Stichproben für die Berechnung erzeugt werden sollen. Die Anzahl der Versuche steht in direktem Zusammenhang mit der Genauigkeit der Simulation. Das heißt je mehr Versuche gemacht werden, desto sicherer wird das Ergebnis. Die Genauigkeit der Berechnung kann aus der Anzahl der Stichproben  $M$  berechnet werden.

Als letzter Schritt in der ersten Phase ist das Modell für die Berechnung zu wählen. Es werden zwei einfache Funktionen dazu benötigt. Die erste Funktion entscheidet, ob eine Stichprobe inner- oder außerhalb des Kreises liegt. Die zweite Funktion zählt wie viele innerhalb und wie viele außerhalb des Kreises liegen.

Sei also  $X_j = (X_{1,j}, X_{2,j})$  eine zufällige Position in dem Quadrat und  $h(X)$  eine Funktion, die entscheidet ob ein Punkt innerhalb oder außerhalb des Kreises liegt. Da das Quadrat eine Kantenlänge von 1 hat, kann die Funktion damit wie folgt geschrieben werden

$$h(x) = \begin{cases} 1 & \|X\| \leq 1 \\ 0 & \|X\| > 1 \end{cases}.$$

Die Funktion liefert Eins, wenn der Punkt  $X$  innerhalb des Kreises liegt und Null wenn er außerhalb liegt.

**Phase Zwei** In der zweiten Phase werden nun die vorher festgelegten  $M$  Stichproben gezogen. In dem Beispiel wurde das mit der Algebra Software Mathematica durchgeführt. In den weiteren Schritten wird davon ausgegangen, dass diese Software perfekte Zufallszahlen erzeugt. Für jede erzeugte Zufallszahl wird sogleich mit Hilfe der Funktion  $h(X_i) = I_i$  entschieden, ob sie im Kreis liegt oder nicht. Die Ergebnisse werden für den nächsten Schritt gespeichert.

**Phase Drei** In der letzten Phase werden die Ergebnisse zusammengefasst. Um die Kreiszahl zu berechnen muss zunächst gezählt werden, wie viele Stichproben sich im Kreis befinden

$$I = \sum_{i=1}^M I_i.$$

Somit ergibt sich für das Flächenverhältnis  $Q = \frac{I}{M}$ . Zusammen mit dem schon gezeigten  $Q \approx \frac{1}{4}\pi$  kann die Kreiszahl in Verhältnis zu den Punkten innerhalb des Kreises und der gesamten Punktezahl gebracht werden

$$\frac{I}{M} \approx \frac{1}{4}\pi$$

und damit kann die Kreiszahl  $\pi$  geschätzt werden durch

$$\pi \approx 4IM^{-1}.$$

Der Leser wird leicht einsehen, dass dies nur eine Schätzung für  $\pi$  sein kann. Und zudem wird bei einer Wiederholung des Experiments nicht wieder die selbe Schätzung geliefert, da das Verfahren nicht deterministisch ist. Im Folgenden wurde beispielsweise  $M = 5000$  gewählt und das Verfahren viermal angewandt. Dabei haben wurden für  $\pi$  die Werte  $\{3,1528, 3,1664, 3,1192, 3,1224\}$  geschätzt. Selbst mit der vermeintlich großen Anzahl von 5000 Versuchen gelingt es nicht  $\pi$  auf die zweite Nachkommastelle zu schätzen. Aber nach dem Gesetz der Großen Zahl kann die Genauigkeit beliebig groß sein, wenn die Anzahl der Versuche  $M$  nur groß genug gewählt wird.

Bernoullis Gesetz der großen Zahl besagt, wenn  $\Pi_M(A)$  die relative Häufigkeit von  $A$  bei  $M$  unabhängigen Versuchen sei, bei welchen das Ereignis  $A$  mit der Wahrscheinlichkeit  $P(A)$  eintritt, dann konvergiert die relative Häufigkeit gegen die Wahrscheinlichkeit [86, p. 334]. Formal lässt sich diese Tatsache wie folgt beschreiben:

$$\lim_{n \rightarrow \infty} P(|\Pi_n(A) - P(A)| < \epsilon) = 1. \quad (8.1)$$

In dem konkreten Fall ist die relative Häufigkeit das Verhältnis der Punkte innerhalb des Kreises  $I$  zu der Gesamtanzahl der Punkte  $M$  und die Wahrscheinlichkeit entspricht dem Verhältnis der Flächen. Somit kann die Kreiszahl  $\pi$  mit dieser Methode beliebig genau berechnet werden. Das Gesetz der großen Zahl ist somit auch die formale Begründung für die Korrektheit der MCS.

Damit kann zumindest theoretisch die Genauigkeit einer MCS beliebig groß sein. In der Praxis muss aber eine Abwägung zwischen Genauigkeit und Zeitaufwand getroffen werden. Eine Möglichkeit diese Abwägung zu treffen ist es die Genauigkeit der Schätzung in Abhängigkeit der Versuchsanzahl zu berechnen. Das Prinzip der  $\pi$  Berechnung mit MCS entspricht dem Werfen einer zweiseitigen Münze und somit einem Bernoulli-Prozess. In [86] wird für Bernoulli Experimente eine Berechnungsmethode für das Konfidenzintervall angegeben, welches ein Maß für die Sicherheit dieser Bestimmung liefert. Sei  $\hat{p}$  die zu schätzende Parameter, so liegt der wirkliche Wert des Parameters mit einer bestimmten Wahrscheinlichkeit in folgendem Intervall

$$p \in \left[ \hat{p} - c\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}; \hat{p} + c\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \right].$$

Wobei  $n$  die Stichprobenanzahl ist und der Parameter  $c = \Phi^{-1}(1 - \frac{\alpha}{2})$  von dem Konfidenzniveau  $(1 - \alpha)$  abhängt. Wählt man für das Konfidenzintervall  $(1 - \alpha) = 95\%$  ergibt sich für  $c = 1,96$  und mit  $M = 5000$  damit ein Konfidenzintervall für die Schätzung  $3.08793 \leq \tilde{\pi} \leq 3.17927$ .

Für das Beispiel der  $\pi$  Berechnung ist eine solche Verteilung der Punkte in Abbildung 8.2(a) für  $M = 5000$  Versuchen zu sehen. In Abbildung 8.2(b) sind die Konfidenzintervalle für  $M \in \{1000, 10^4, 10^5, 10^6, 10^7\}$  Versuchen zu sehen. Die Abhängigkeit des Konfidenzintervalls von der Anzahl an Stichproben ist klar zu erkennen.

Im Fall der Zeitkalibrierung werden die Eingangsdaten empirisch erhoben. So werden für die verschiedenen Sensoren die Rauschparameter bestimmt oder die Regelmäßigkeit der Wiederholrate. Mit Hilfe der experimentell gewonnenen Daten lassen sich statistische Verteilungen ermitteln und somit auch Stichproben generieren. Die erzeugten Stichproben werden dann in die zu untersuchenden Algorithmen gesteckt und die Ausgabe berechnet. Prinzipiell lassen sich somit alle Parameter, welche eine Kalibrierung beeinflussen, modellieren und damit auch simulieren.

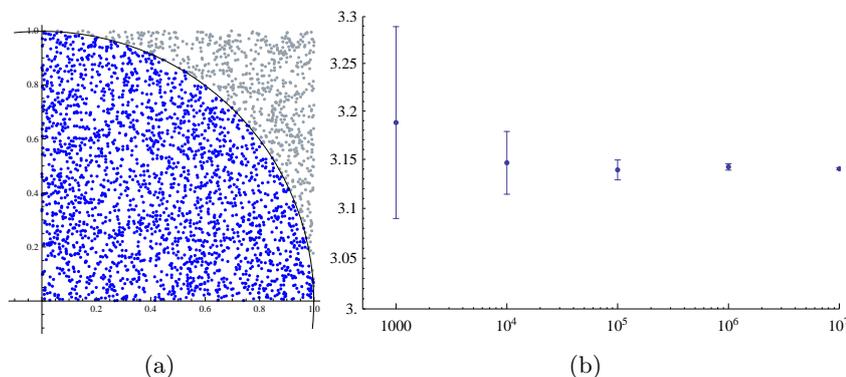


Abbildung 8.2: Konfidenzintervalle einer Monte-Carlo-Simulation; (a) Berechnung von  $\pi$  mit  $M = 3000$  Versuchen; (b) Konfidenzintervalle in Abhängigkeit von der Anzahl an Versuchen

### 8.3 Framework

Im Folgenden wird ein Softwarepaket beschrieben, welches erlaubt verschiedenste Eingabeparameter wie Rauschen und Registrierungsfehler zu simulieren und dadurch einzelne Fehlerquellen isoliert von anderen betrachten zu können. Durch Verwendung einer Simulation ist zudem eine verlässliche *ground truth* vorhanden, denn der wahre Wert der relativen Latenz ist schon bekannt.

Das Ziel dieses Frameworks ist es ein Werkzeug zu liefern, mit dem die Genauigkeit, Robustheit und die Effizienz der einzelnen Schritte für eine Kalibrierung genau untersucht werden können und somit die Schätzung der relativen Latenz

qualitativ verbessert werden kann.

### 8.3.1 Ablauf der Simulation

Die Simulation beruht im Grunde auf den in Kapitel 8.2 beschriebenen Prinzipien. Dazu müssen erst die statistischen Parameter der zu simulierenden Eingabeparameter und eine zu untersuchende Bewegungssequenz der zwei Sensoren bestimmt werden. In Abbildung 8.3 ist der prinzipielle Ablauf einer Simulation dargestellt.

Diese startet mit der Aufnahme einer Bewegung. Dazu wird die Bewegung eines Sensors aufgezeichnet und mit Hilfe von Filtern von Sensor-Rauschen befreit. Diese Bewegung ist die Grundlage für die weiteren Simulationen. Die weiteren Schritte in der Simulation werden so oft wie gewünscht wiederholt.

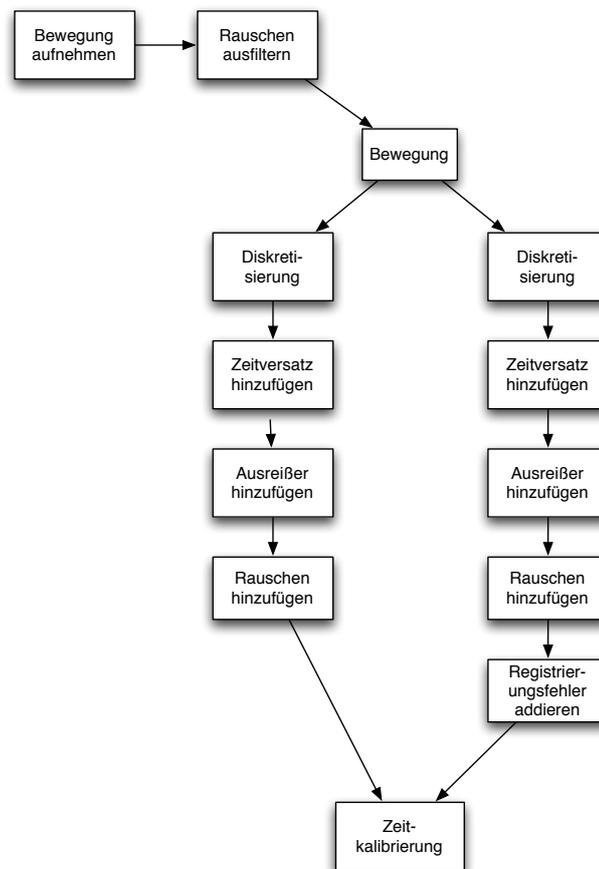


Abbildung 8.3: Monte-Carlo Simulation für die Zeitkalibrierung

### 8.3.2 Bestimmung der Verteilungen

Für die Durchführung einer Monte-Carlo-Simulation ist wie beschrieben ein statistisches Modell der simulierenden Parameter nötig. Diese können unter Umständen dem Datenblatt des Herstellers entnommen werden oder falls diese Werte nicht

vorliegen experimentell bestimmt werden. Um die oben beschriebene Methode des „one-factor-at-a-time“ Ansatzes nutzen zu können, müssen die zu bestimmenden statistischen Parameter unabhängig von anderen Parametern sein. Dies ist in manchen Fällen leicht zu gewährleisten. Folgende Absätze beschreiben, wie für die wichtigsten Einflüsse die statistischen Parameter bestimmt werden können.

### Rauschen

Das Rauschen eines Sensors ist zu unterscheiden von systematischen Fehlern eines Sensors [7]. Systematische Fehler können reproduziert werden. Das heißt, dass ein Fehler bei wiederholter Messung unter gleichen Bedingungen wieder in dem gleichen Maße auftritt. Davon abweichend werden Rauschfehler als nicht deterministisch bezeichnet. Es ist möglich das Rauschen eines Sensors mit Hilfe eines statistischen Modells zu beschreiben und dann dieses Modell des Rauschens als Eingangsparameter für eine Monte-Carlo-Simulation zu nutzen.

Die  $n$ -dimensionale Gauß-Verteilung  $X \sim N_n(\mu, \Sigma)$  ist ein gängiges Modell zur Beschreibung von Sensorrauschen [43, 86]. Wobei  $\mu$  der Mittelwert der Verteilung ist und  $\Sigma$  die Kovarianzmatrix repräsentiert. Der Mittelwert  $\mu$  darf als Null angenommen werden. Denn ein von der Null abweichender Mittelwert wäre ein systematischer Fehler und somit kein Rauschen. Somit reicht eine Kovarianzmatrix  $\Sigma$  für die vollständige Beschreibung des Modells für das Rauschen eines Sensors aus.

Für die Monte-Carlo-Simulation muss also die Kovarianzmatrix ermittelt werden. Dazu müssen die Sensoren mit technischen Hilfsmitteln fixiert werden, so dass sich der Sensor nicht bewegen kann. Auch auf eine ruhige Umgebung muss geachtet werden, denn auch sich bewegende Personen in einem Raum können die Messung beeinflussen. Um eine verlässliche Kovarianz zu ermitteln müssen über einen längeren Zeitraum Daten aufgenommen werden. Nach der Aufnahme der Daten müssen diese so verschoben werden, dass der Mittelpunkt der Daten im Nullpunkt liegt. Anschließend kann die Kovarianzmatrix berechnet werden.

Manche Sensoren liefern Daten, die über die Zeit gesehen, driften. Als Beispiel hierfür seien inertielle Sensoren wie Gyroskope genannt. Das Driften der Sensoren ist ein systematischer Fehler, der auf keinen Fall in die Bestimmung der Kovarianz einfließen darf. Wird das Driften nicht korrigiert, führt dies zu einer zu großen Einschätzung des Rauschens und alle darauf aufbauenden Simulationen würden auf zu großen Rauschparametern beruhen. Bei einigen Sensoren ist das Rauschen im Arbeitsvolumen nicht konstant. So ist zum Beispiel das Rauschenverhalten bei einem A.R.T. System abhängig von der Entfernung zwischen Sensoreinheit und Markerkugel. Je weiter die Markerkugel vom Sensor entfernt wird desto größer wird das Rauschen. Dieses ortsabhängige Verhalten des Rauschens wurde in [9, 91, 62, 59] untersucht und soll hier nicht weiter berücksichtigt werden.

Beispielhaft wurden die Kovarianzmatrizen für die Messung einer Position mit einem A.R.T. System und für die Messung mit einem FaroArm bestimmt. In Abbildung 8.4 ist der technische Aufbau für diese Messung zu sehen. Das A.R.T.

System war dabei ca. 2 m von dem Aufbau entfernt. Die folgenden Kovarianzmatrizen  $\Sigma_{\text{FARO}}$  und  $\Sigma_{\text{ART}}$  wurden aus einem 42 s langen Messstrom bestimmt. Insgesamt lagen 2098 Messungen des FaroArms und 2581 Messungen des A.R.T. System vor.

$$\Sigma_{\text{FARO}} = \begin{pmatrix} 7,9 \times 10^{-11} & -6,0 \times 10^{-11} & 1,4 \times 10^{-11} \\ -6,0 \times 10^{-11} & 3,0 \times 10^{-10} & -6,5 \times 10^{-11} \\ 1,4 \times 10^{-11} & -6,5 \times 10^{-11} & 2,4 \times 10^{-10} \end{pmatrix} \quad (8.2)$$

$$\Sigma_{\text{ART}} = \begin{pmatrix} 6,3 \times 10^{-10} & -3,0 \times 10^{-10} & -2,8 \times 10^{-10} \\ -3,0 \times 10^{-10} & 2,0 \times 10^{-9} & -2,8 \times 10^{-9} \\ -2,8 \times 10^{-10} & -2,8 \times 10^{-9} & 6,1 \times 10^{-9} \end{pmatrix} \quad (8.3)$$

Aus den Kovarianzmatrizen können mit Hilfe der Hauptkomponentenanalyse die Hauptkomponenten des Rauschens ermittelt werden. Liegen die Hauptkomponenten vor, erlaubt dies die Bestimmung eines skalaren Wertes (RMS). Der RMS schreibt das Rauschen eines Sensors zwar nur qualitativ, jedoch erlaubt es der RMS einem Anwender das Rauschenverhalten eines Sensors intuitiver und schneller zu beurteilen als durch Betrachten der Kovarianzmatrix. Für die beiden Kovarianzmatrizen  $\Sigma_{\text{ART}}$  und  $\Sigma_{\text{FARO}}$  lässt sich jeweils ein RMS mit Methoden aus [91] berechnen. Für den FaroArm ergibt sich  $\text{RMS}_{\text{FARO}} = 0,025 \text{ mm}$  und für das A.R.T. System  $\text{RMS}_{\text{ART}} = 0,094 \text{ mm}$ .



Abbildung 8.4: Befestigung der Kombination aus einem FaroArm und einer Markerkugel mit Hilfe eines Stativs

Das Rauschen von Sensoren war auch schon Gegenstand von theoretischen Untersuchungen [7, 91, 62], so dass die Eingabeparameter auch von diesen Untersuchungen oder aus anderen Quellen benutzt werden können.

### Update Raten und Jitter

Um die Wiederholraten der Sensoren zu bestimmen wurden 40 s Daten mit dem Ubitrack Framework aufgenommen. Das ubitrack System versieht jedes eingehende Datenpaket mit einem Zeitstempel. Dies geschieht als allererste Maßnahme im Framework und mit höchster Priorität. Für weitere Details zu ubitrack sei auf Abschnitt 2.6.4 verwiesen. Die so mit Ubitrack gewonnenen Zeitstempel wurden dann hinsichtlich der durchschnittlichen Updaterate und eventueller Jitter untersucht.

In Abbildung 8.5 ist eine Darstellung der Update-Zeiten  $\Delta t_i$  dreier Sensoren (A.R.T., FaroArm und Marker Tracker) abgebildet. In den ersten zehn Sekunden kann eine starke Veränderung der Update-Zeiten beobachtet werden. Ab der zehnten Sekunde ist kein Trend bei den Update-Zeiten mehr zu erkennen. Für die Simulation wird das Einpendeln der Update-Raten ignoriert und nur die Update-Rate benutzt, auf die sich die Systeme einpendeln. Der Grund für dieses Einpendeln wurde nicht weiter untersucht. Es wird aber vermutet, dass es von der internen Verarbeitung innerhalb des ubitrack frameworks ausgelöst wird.

	Mittelwert $\mu$	mittlere Frequenz	Standardabweichung $\sigma$
A.R.T.	16,7 ms	59,9 Hz	0,0804 ms
FaroArm	20,5 ms	48,8 Hz	0,0596 ms
Marker	70,9 ms	14,1 Hz	17,5 ms

Tabelle 8.1: Mittlere Aktualisierungs-Raten und deren Standardabweichungen

### Bewegungen

Für eine vollständige Simulation sind neben den schon beschriebenen Parametern auch Bewegungsdaten nötig. Diese Bewegungsdaten müssen ein breites Spektrum an Bewegungstypen abdecken. So sind zum einen generierte Daten nötig, um das Verhalten der einzelnen Verfahrensschritte in Grenzsituationen untersuchen zu können. Pathologische Fälle sind Beispiele für solche Grenzsituationen. Und zum anderen sind natürliche Bewegungen von menschlichen Probanden nötig, damit die Verfahren in typischen AR Situationen untersucht werden können. Als Beispiel hierfür kann das Vorgehen aus dem Artikel *City of Sights* dienen [35]. Dort wurden zwölf Probanden gebeten, mit einer Kamera eine Kamerafahrt durch eine Modelllandschaft durchzuführen. Gleichzeitig wurde die Daten auch mit einem FaroArm aufgenommen und eine präzise Kalibrierung zwischen den Systemen vorgenommen. In diesen Datensätzen wurden dann die relative Latenz mit den hier vorgestellten Verfahren korrigiert und sie dienen nun als Referenz für kamerabasierte Trackingmethoden.

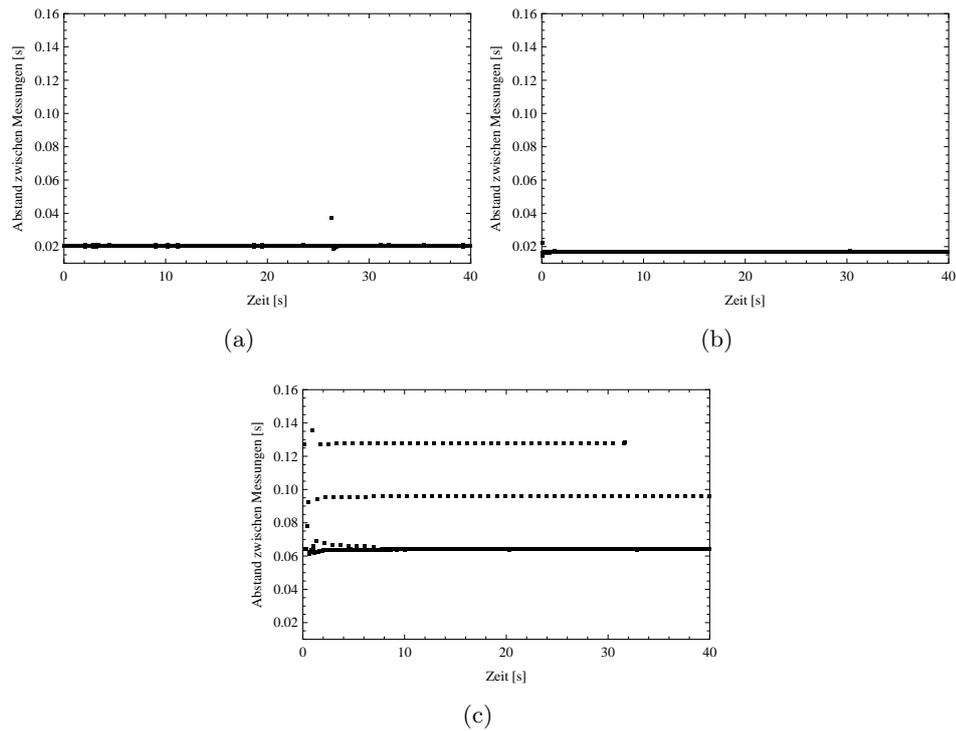


Abbildung 8.5: Update-Raten dreier Sensoren; (a) zeigt die Aktualisierungs-Rate eines FaroArms, (b) zeigt die eines A.R.T. Systems und (c) zeigt die unregelmäßigen Aktualisierungs-Raten eines Marker Trackers

Für die Erzeugung von Datensätzen für die Monte-Carlo-Simulation der zeitlichen Kalibrierung wurde eine exemplarische Bewegung mit einem FaroArm durchgeführt. Der FaroArm weist ein sehr niedriges Sensorrauschen, eine sehr große Messgenauigkeit und eine sehr geringe Anzahl an Ausreißern auf. Daher eignet sich dieses System ideal für die Erzeugung typischer Bewegungen für Augmented Reality, denn die Daten müssen nicht anschließend aufwendig aufbereitet werden. Bei anderen Sensoren, wie dem Markertracker, ist eine Filterung nötig, da das Signal stark verrauscht ist und eine vergleichbare hohe Quote an Ausreißern aufweist. Nachdem dieses Signal Ausgangsbasis für die Simulation von genau diesen Störquellen sein soll, dürfen solche Störungen nicht vorhanden sein.

Bei der Wahl der Bewegungen sollten neben typischen Bewegungen von AR Nutzern auch weitere Bewegungen betrachtet werden. So kann es vorkommen, dass der Nutzer einen Sensor auf einem Tisch ablegt und somit keine sinnvolle Kalibrierung durchgeführt werden kann. Das System muss Fälle wie diesen zuverlässig erkennen und darf eine Kalibrierung unter diesen Umständen nicht durchführen. Auch andere Ereignisse wie sehr schnelle und sehr langsame Bewegungen sollten betrachtet werden, damit ein Kalibrieralgorithmus auf diese Bewegungsmuster hin untersucht werden kann. Neben solchen Bewegungen sollten auch typische

Kalibrier-Bewegungen aufgezeichnet werden. Das sind Bewegungen mit großen und verschiedenen Beschleunigungen sowohl der Position als auch der Orientierung.

### Registrierungsfehler

In nahezu allen AR Anwendungen müssen Koordinatensysteme zueinander registriert werden. Dies kann je nach Problemstellung beispielsweise eine Hand-Zu-Auge-Kalibrierung (engl.: *hand-eye-calibration*) oder auch eine 3D-3D Kalibrierung (engl.: *absolute orientation*) sein. Diese Kalibrierungen sind in der Regel nicht ohne Fehler, da die Sensormesswerte schon mit Fehlern behaftet sind. Die Güte der Kalibrierung kann aber durch geschickte Auswahl der Sensordaten, durch eine hohe Anzahl an Kalibrierpunkte verbessert werden und durch weitere Maßnahmen optimiert werden. Durch eine Steigerung der Anzahl der Kalibrierpunkte wird der negative Einfluss des Sensorrauschens reduziert [38, 62]. Diese Maßnahme ist zwar leicht umzusetzen, doch leider kann die Auswirkung in der Praxis nur abgeschätzt werden. Ein direktes Messen von Registrierungsfehlern kann nur unter sehr großem Aufwand durchgeführt werden. Daher werden Registrierungsfehler meist vom Konvergenzverhalten während der Registrierung abgeleitet [62]. Soll jedoch das Verhalten eines Kalibrieralgorithmus unter dem Einfluss bestimmter Störgrößen beurteilt werden, bietet sich auch hier eine Monte-Carlo-Simulation an. Denn so kann die Streuung der Kalibrierung unter klar definierten Bedingungen angegeben werden. Diese Streuung kann dann wiederum als Eingangsparameter für die Monte-Carlo-Simulation der Kalibrierung der relativen Latenz dienen.

Im Rahmen dieser Arbeit wurde keine Untersuchung einzelner Kalibrieralgorithmen durchgeführt. Es sollen hier nur kurz die prinzipiellen Verfahrensschritte und Einflussgrößen beschrieben werden. Für die detaillierte Beschreibung sei auf die Arbeiten von Keitler [59, 62] verwiesen.

Wie schon beschrieben ist eine Kalibrierung von verschiedenen Einflussgrößen abhängig. Als erster Faktor sei das Rauschen der einzelnen Sensoren benannt. Das Rauschen kann mit Hilfe einer mehrdimensionalen Gaußverteilung um den Nullpunkt beschrieben werden. Somit reicht eine Kovarianzmatrix  $\Sigma$  aus, um das Rauschen zu beschreiben. In Abschnitt 8.3.2 auf der Seite 112 sind hierzu weitere Details zu entnehmen. Wie schon erwähnt kann der negative Einfluss des Rauschens durch eine Erhöhung der Anzahl  $N$  der Kalibrierpunkte ausgeglichen werden. Daher ist die Anzahl der Kalibrierpunkte auch in der Simulation zu berücksichtigen.

Das Verfahren ist analog wie in Abschnitt 8.2 beschrieben. Als ein erster Eingabeparameter kann das Rauschen der einzelnen Sensoren dienen. Wie schon beschrieben ist aber eine Kalibrierung auch abhängig von der Anzahl der verwendeten Kalibrierpunkte, so dass dieser Parameter auch berücksichtigt werden muss. Weiterhin ist auch die geometrische Position und Lage, bzw. die Kalibrierbewegung, von Relevanz für die Genauigkeit. Dies ist dann ein weiterer Parameter für die Simulation. Natürlich spielt bei einer Kalibrierung die räumliche Verteilung der Messwerte eine Rolle. Dies lässt sich bei einer Simulation nur schwer berücksichtigen, denn jeder Nutzer eines Kalibrierungsalgorithmus verhält sich

während einer Kalibrierung verschieden. In [62] hat Keitler dafür die Punkte mit einer Gleichverteilung in einem typischen Arbeitsvolumen verteilt und somit verschiedene Eingabemuster simuliert.

In Abbildung 8.6 ist ein einfaches Beispiel für eine Monte-Carlo-Simulation abgebildet. Dort wird ein Satz von  $N$  Punkten  $n$ -mal mit einer durch eine Kovarianzmatrix gegebenen Gaußschen-Verteilung mit Rauschen belegt. Anschließend wird dann eine Kalibrierung der Absoluten Orientierung durchgeführt und aus den Ergebnissen dann eine Kovarianzmatrix errechnet. Diese Kovarianzmatrix beschreibt die Ungenauigkeit der Kalibrierung, die durch Sensorrauschen ausgelöst wurde.

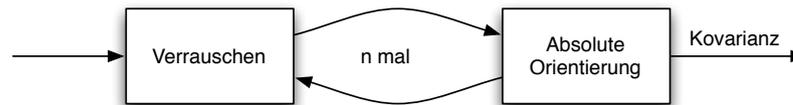


Abbildung 8.6: Ablauf der Montecarlo Simulation für Bestimmung des Registrierungsfehlers.

### 8.3.3 Beispiel

Um die eben vorgestellten Methoden der Monte-Carlo Simulation zu demonstrieren, sollen im Folgenden als Beispiel mit der Norm und der PCA-Methode zwei Dimensionsreduktionsmethoden miteinander verglichen werden.

Der Vergleich soll an zwei verschiedenen Pfaden, die Daten mit einem hohen SNR Wert und Daten mit einem geringen SNR Wert repräsentieren, durchgeführt werden. In Abbildung 8.7 sind die beiden Pfade A und B dargestellt. Während der Pfad A (vgl. Abb. 8.7(a)) große Bewegungen im Dezimeterbereich mit vielen Bewegungsänderungen beinhaltet, besteht Pfad B (vgl. Abb. 8.7(b)) zwar aus vielen Richtungsänderungen, die sich aber nur im Millimeterbereich abspielen. Die beiden Datensätze sind 4 s lang und umfassen 195 Datensätze. Neben den zwei verschiedenen Signalrauschverhältnissen soll auch die Auswirkung von Rauschen auf die Ergebnisse der Zeitkalibrierung unter der Anwendung der zwei Dimensionsreduktionen untersucht werden.

Das Vorgehen für diese Simulation ist in Abbildung 8.3 dargestellt. Es wurden also die Daten der Pfade dupliziert und um einen Zeitversatz von  $\delta = 32,12$  ms gegeneinander verschoben. Anschließend wurden die Positionsdaten des verschobenen Datensatzes mit einem normalverteilten Rauschen  $X \sim N_n(\vec{0}, \Sigma_{\text{ART}})$  belegt. Die zugehörige Kovarianzmatrix wurde in einem vorherigen Versuch ermittelt (vgl. Formel 8.3, S. 113). Mit diesen beiden Datensätzen wird nun eine Bestimmung der relativen Latenz vorgenommen. Wobei die Dimension einmal mit Hilfe der PCA Methode reduziert wurde und einmal mit der Norm. Dieser Versuch wurde  $M = 10.000$  wiederholt, um die nötige Sicherheit der Simulation zu gewinnen. In

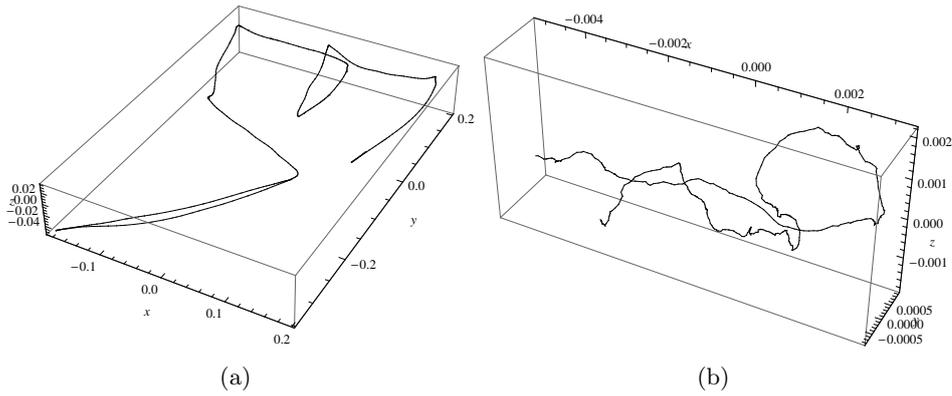


Abbildung 8.7: Die zwei für die Monte-Carlo Simulation benutzten Positionspfade: (a) zeigt Pfad A mit viel Bewegung und Bewegungsänderung; (b) zeigt Pfad B eines Handtremors und dadurch relativ wenig Bewegung

der Abbildung 8.8 ist der Verlauf des Durchschnitts der Schätzungen über die Anzahl  $M$  der Versuche dargestellt. Man kann erkennen, wie sich der Durchschnitt an die im Versuch vorgegebene Latenz  $\delta$  annähert. In Tabelle 8.2 sind die Ergebnisse der vier Durchläufe aufgeführt.

	Norm		PCA	
	$\bar{e}_\delta$	$\sigma_\delta$	$\bar{e}_\delta$	$\sigma_\delta$
Pfad A	$1,9 \times 10^{-5}$ ms	0,016 ms	$4,8 \times 10^{-5}$ ms	0,017 ms
Pfad B	0,029 ms	1,0 ms	0,014 ms	0,96 ms

Tabelle 8.2: Vergleich der Norm und der PCA als Dimensionsreduktion an zwei verschiedenen Pfaden und unter dem Einfluss von Rauschen. Die Schrittweite der Suche in Pfad A betrug 0,01 ms und in Pfad B 0,1 ms

Die hier präsentierte Genauigkeit der Kalibrierung wird in der Praxis nicht erreicht. Das liegt an der Separation der einzelnen Fehler. Aber als erstes Ergebnis kann man festhalten, dass das Sensorrauschen die Genauigkeit der Kalibrierungen nicht wesentlich beeinflusst. So liegt der RMS des Fehlers der Kalibrierung unter den Daten von Pfad B bei rund 1,0 ms.

Weiterhin zeigen die Ergebnisse, dass die PCA als Dimensionsreduktionswerkzeug keinen Vorteil bei verrauschten Signalen bringt. Die Fehler sind sogar ein wenig größer. Bei Pfad A, welcher den größeren SNR Wert aufweist, kann kein Unterschied zwischen den beiden Dimensionsreduktionsmethoden festgestellt werden. Hingegen ist der durchschnittliche Fehler bei Pfad B mit der Norm als Dimensionsreduktionsfunktion doppelt so groß wie mit der Projektion an der Hauptkomponente. Doch muss erwähnt werden, dass diese Halbierung nur von theoretischer Natur ist. Die Effekte von falscher Kalibrierung und Ausreißern sind um Magni-

tuden größer als diese hier errechneten Fehler. In der Abbildung 8.8 ist für jeden Durchlauf der Monte-Carlo-Simulation der Verlauf des Durchschnitts der Kalibrierungen zu sehen. In dieser Darstellung ist die Konvergenz der Simulation zu erkennen.

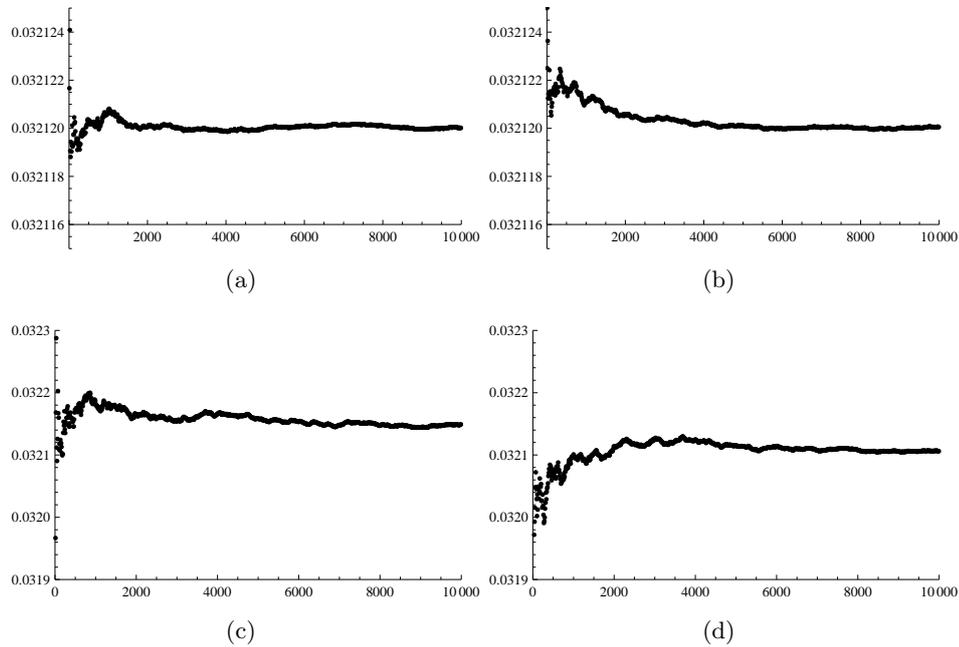


Abbildung 8.8: Konvergenz der Simulationen als Verlauf des Durchschnitts der Kalibrierungen: (a) Pfad A mit Norm als Dimensionsreduktion; (b) Pfad A mit PCA als Dimensionsreduktion; (c) Pfad B mit Norm als Dimensionsreduktion; (d) Pfad B mit PCA als Dimensionsreduktion;

## 8.4 Zusammenfassung

Dieses Kapitel hat die Monte-Carlo-Simulation im Allgemeinen gezeigt und demonstriert wie diese Methode für die Untersuchung der Zeitkalibrierung verwendet werden kann. Das vorgestellte Modell der Monte-Carlo-Simulation erlaubt es nahezu jeden Faktor zu berücksichtigen, der eine Zeitkalibrierung beeinflussen kann. Für einen dieser Faktoren wurde gezeigt, wie eine Modellierung aussehen kann, und wie diese dann in einer Simulation verwendet werden kann.

Es ist nicht möglich alle möglichen Faktoren mit einer MCS zu untersuchen, ohne Abhängigkeiten zwischen diesen zu vernachlässigen. Ohne diese Einschränkung wird die Komplexität durch das exponentielle Wachstum der Variationen zu groß. Dieser Umstand wird zusätzlich dadurch verschärft, dass für eine sichere Schätzung eine Vielzahl an Versuchen nötig ist. Diese Komplexität kann durch Variation weniger Parameter beherrschbar gehalten werden. Und durch gezielte Untersuchungen

einzelner Abhängigkeiten können bestimmte Eigenschaften näher untersucht werden. Beispielsweise kann durch Variierung des Rauschens und des Registrierungsfehlers untersucht werden, welches Verfahren zur Berechnung der Korrelation die zuverlässigsten Resultate liefert. Durch die Tatsache, dass bei der Simulation die relative Latenz als Parameter selbst gewählt wurde, kann die Simulation auch die Fehler eines Verfahrens exakt bestimmen.

Gegen Ende dieses Kapitels wurde mit einem exemplarischen Durchgang der Monte-Carlo-Simulation die Anwendbarkeit des Verfahrens demonstriert. Wobei die Implementierung in Mathematica keine performante Ausführung zulässt, da dieses System mehr auf algebraische denn auf numerische Operationen optimiert wurde. Eine Portierung der MCS in eine andere Programmiersprache oder in Mat-Lab ist daher zu empfehlen.

## 9 Anwendungen

Für eine komplette Zeitkalibrierung sind nun alle Werkzeuge vorhanden. In diesem Kapitel wird in zwei Versuchen demonstriert, wie mit den vorgestellten Methoden eine Zeitkalibrierung durchgeführt werden kann. Der erste Versuch zeigt eine Kalibrierung der relativen Latenz zwischen drei Sensoren. Dabei wird mit einem Ringschluss gezeigt, dass diese Kalibrierungen korrekt und in sich stimmig sind. Anschließend wird eine Anwendung gezeigt, die eine Evaluierung eines auf Infrarotlicht basierendem Trackingsensors mit dem FaroArm erlaubt. Für eine solche Anwendung ist größte Präzision erforderlich und daher auch eine Kalibrierung der relativen Latenz.

### 9.1 Ringschluss

Die Anwendung Ringschluss soll die Richtigkeit der Schätzung des Zeitversatzes zeigen. Dazu wurden die relativen Latenzen zwischen drei Sensoren kalibriert. Diese Sensoren waren ein FaroArm ( $S_F$ ), ein A.R.T. ( $S_A$ ) und ein Markertracker ( $S_M$ ). Der Aufbau dieses Versuches entspricht dem Aufbau der auch in [35] verwendet wurde. Nun ist es theoretisch mit nur zwei Kalibrierungen möglich alle Latenzen zu bestimmen, denn sind die Latenzen zwischen zwei Sensoren schon bekannt, kann die dritte Latenz durch die anderen zwei berechnet werden.

Bezeichne nun  $\delta(S_A, S_F)$  die relative Latenz zwischen dem Sensor A.R.T. und dem Sensor FaroArm in Sekunden. Wobei  $\delta(S_A, S_F) > 0$  ms bedeutet, dass die Werte des Sensors  $S_A$  das System vor den Werten des Sensors  $S_F$  erreichen. Wenn alle Kalibrierungen korrekt sind, dann muss folgende Gleichung gelten

$$\delta(S_A, S_F) + \delta(S_F, S_M) = \delta(S_A, S_M). \quad (9.1)$$

Nun wurden die drei Sensorpaare jeweils mechanisch miteinander fixiert. Eine Illustration einer solchen Montage ist in Abbildung 9.1 zu sehen. Anschließend wurden die Paare jeweils erst einer Raumkalibrierung und mit Hilfe dieser Registrierung dann einer Zeitkalibrierung unterzogen. In der Tabelle 9.1 sind die kalibrierten Werte aufgeführt. Neben den Mittelwerten der einzelnen Werte sind auch die Mediane und die Standardabweichungen dort aufgeführt. Wie die Messungen zeigen, erreichen erst die Daten des A.R.T. Systems das System, gefolgt von den Daten des FaroArms und des optischen Markertrackers. Rechnet man die Ergebnisse der Kalibrierung zusammen also

$$e = \delta(S_A, S_F) + \delta(S_F, S_M) - \delta(S_A, S_M) = -6,6 \text{ ms} \quad (9.2)$$

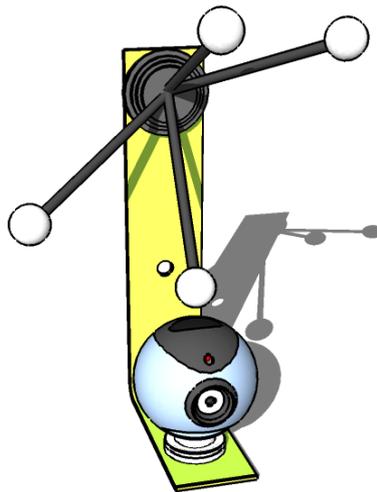


Abbildung 9.1: Illustration der Kombination einer WebCam und eines A.R.T. Targets montiert auf einem Metallwinkel.

ergibt sich ein Fehler von  $e = -6,6$  ms. Im Verhältnis zu den bestimmten Latenzen ist dieser Fehler klein. Eine Darstellung der kalibrierten relativen Latenzen mit der Visualisierungsmethode von [51] ist in Abbildung 9.2 zu sehen.

	A.R.T. zu FaroArm	FaroArm zu Marker	Art zu Marker
Mittelwert	31,87 ms	86,32 ms	119,5 ms
Median	32,4 ms	84 ms	123 ms
Std.abweichung	2,7 ms	30,24 ms	65,71 ms

Tabelle 9.1: Bestimmte relative Latenzen und deren Standardabweichung

## 9.2 Optisches Odometer

In dem folgenden Abschnitt wird anhand der in Abschnitt 2.5.1 schon kurz vorgestellten Fusionsplattform gezeigt werden, dass die Berücksichtigung der relativen Latenz auch schon während der Kalibrierung berücksichtigt werden muss.

Einer der Sensoren der Fusionsplattform ist ein optisches Odometer, welches im Rahmen des trackframe Forschungsprojekts in enger Zusammenarbeit mit Manuel Huber entwickelt wurde. Der Sensor besteht aus drei einzelnen optischen odometrischen Sensoren, die als eine Einheit funktionieren. Die Funktionsweise der einzelnen Sensoren wurde in Abschnitt 2.4 beschrieben.

Jeder einzelne Sensor misst seine Bewegung relativ zum Boden. Nutzt man mehrere Sensoren, deren räumliche Beziehungen zu einander bekannt sind, dann kann neben der Positionsänderung auch die Orientierungsänderung der gesamten Einheit berechnet werden. Diese Art des Trackings wird in der Fusionsplattform

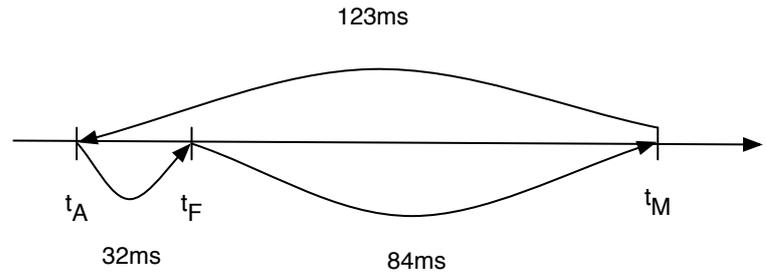


Abbildung 9.2: Drei relativen Latenzen zwischen den drei Sensoren FaroArm, A.R.T und Marker Tracker.

dann genutzt, wenn das optische Tracking nicht zur Verfügung steht.

Für die Kalibrierung des Odometers wird die gesamte Plattform mit einem Trackingsystem erfasst. Mit Hilfe dieser Trackingdaten und eines Modells kann dann die Position der odometrischen Sensoren, deren Orientierung und die Skalierung bestimmt werden. Dabei muss aber auf die relative Latenz zwischen Odometer und dem Trackingsystem geachtet werden. In dem Fall des selbst entwickelten Odometers und dem A.R.T. Trackingsystem beträgt diese relative Latenz ca.  $0,5\text{ s}$ . Die Bestimmung dieser sehr großen relative Latenz wurde nicht mit den in dieser Arbeit beschriebenen Methoden durchgeführt, sondern die Daten wurden mit Hilfe eine Computeralgebrasystems verglichen und somit die Latenz bestimmt. Weitere Untersuchungen über das optische Odometer wurden von Manuel Huber in [49] vorgestellt.

### 9.3 Genauigkeitsevaluation

In dieser Untersuchung sollen die Auswirkungen der zeitlichen Kalibrierung demonstriert werden. Dazu wird ein Szenario benutzt, welches für die Untersuchung der Genauigkeit eines Trackingsystems entwickelt wurde. In diesem Szenario werden Längenmessungen zweier Systeme benutzt, die Genauigkeit zu untersuchen. Dieses Verfahren wird beschrieben in [108, 59]. Dort werden zeitaufwendige statische Messungen verwendet, um dynamische Fehler auszuschließen. In diesem Abschnitt wird gezeigt werden, dass der durch die relative Latenz ausgelöste Fehler auf ein Minimum reduziert werden kann.

Um die Genauigkeit eines Systems zu evaluieren, wird ein Referenzsystem benötigt, welches um vieles genauer ist als das zu untersuchende System. In dem gezeigten Szenario soll ein auf optischem Infrarotlicht basierendes A.R.T. System, vgl. Abschnitt 9.1, untersucht werden. Der Trackingfehler für Positionsmessungen liegt bei diesem System nach Angaben des Herstellers bei  $1\text{ mm}$ . Jedoch hängt diese Genauigkeit von Kalibrierungen und anderen Einflüssen ab. Das System kann zur Zeit dem Nutzer keine Information über die zu erzielende Genauigkeit geben. Als Referenzsystem wurde ein FaroArm gewählt, welches Positionen mit

einem Genauigkeit von unter 0,1 mm liefern kann. Somit kann der FaroArm als Referenzsystem für das A.R.T. System genutzt werden.

**Mechanischer Aufbau** Um beide Trackingsysteme vergleichen zu können, wurde eine retroreflektierende Marker Kugel auf die auswechselbare Spitze eines Faro-Arms montiert. Ein Bild dieser Anordnung ist in Abbildung 9.3(a) zu sehen. Das A.R.T. System besteht aus zwei Kameras mit einem Abstand von 0,5 m mit 3,5 mm Brennweite. Das optische Trackingsystem misst den Schwerpunkt der Markerkugel. Der FaroArm misst den Schwerpunkt einer Messkugel, die an der Spitze des Messarms montiert ist. Die Größe der Kugel ist beliebig, da der Durchmesser und die Position auf dem Messarm mit Hilfe der Messsoftware von FARO kalibriert werden kann. In diesem Fall ist die Messkugel eine Marker Kugel des A.R.T. Systems. Dieses Vorgehen erlaubt es die Kalibriersoftware des FaroArms zu verwenden, die bei der durchgeführten Kalibrierung einen  $2\sigma$  Fehler von weniger als 0,1 mm angeben hat.

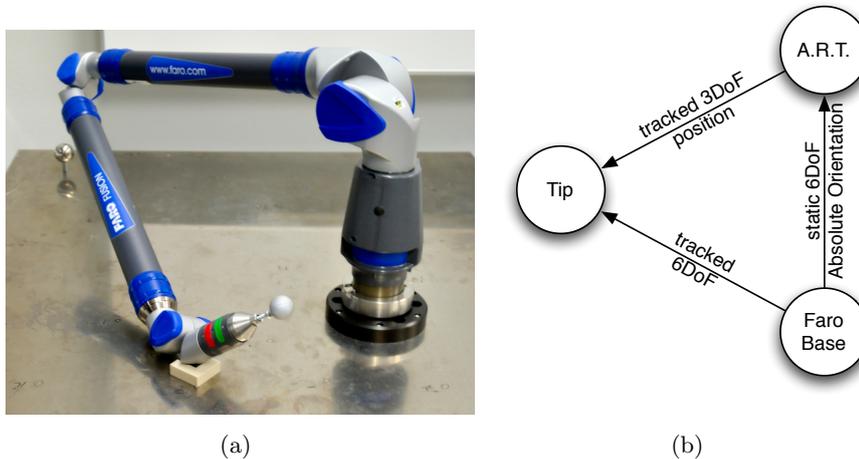


Abbildung 9.3: Anwendung Genauigkeitsmessung A.R.T. mit einem FaroArm; (a) Mechanisches Setup von A.R.T. Markerkugel auf einem FaroArm; (b) SRG der Genauigkeitsmessung

**Räumliche Registrierung** Die beiden Trackingsysteme messen nun die Position desselben Objekts, aber in unterschiedlichen Koordinatensystemen. Diese Situation kann man in dem SRG in Abbildung 9.3(b) sehen. Der Knoten „Tip“ wird im „A.R.T.“ Koordinatensystem und im „FARO Base“ Koordinatensystem gemessen. Die Beziehung zwischen den Knoten „A.R.T.“ und „FARO Base“ kann mit einer Absolute Orientierung Kalibrierung berechnet werden.

Daher muss die räumliche Beziehung der beiden System zueinander bestimmt werden. In diesem Versuch wurde das Verfahren Horn [46] verwendet. Um zu gewährleisten, dass die Kalibrierung nicht durch die noch nicht zeitlich kalibrierten

Sensoren negativ beeinflusst wird, wurde die für die Kalibrierung nötigen Positionen der Kugel mechanisch mit einem Stativ fixiert. Dies stellte sicher, dass die Kugel statisch an der gewünschten Position blieb.

**Zeitliche Kalibrierung** Im nächsten Schritt wurde der zeitliche Offset kalibriert. Dazu wurde die in Abbildung 9.4 zu sehende Bewegung aufgenommen. Für die Kalibrierung des relativen Offsets wurden 18 Segmente mit einer jeweiligen Länge von 4,0s mit einer Überlappung von 1s benutzt. Für die Dimensionsreduktion wurde die Norm benutzt und als Vergleichsfunktion die Korrelation. Die Suche des Offsets wurde mit einer Auflösung von 0,01 ms durchgeführt. Der arithmetische Durchschnitt der 18 bestimmten relativen Latenzen betrug  $\delta = 32,1$  ms mit einer Standardabweichung von  $\sigma = 0,08$  ms. Dabei kommen die Daten des FaroArms später als die A.R.T. Daten bei der Trackingsoftware an.

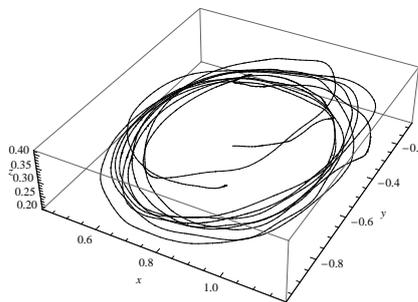


Abbildung 9.4: Darstellung der für die zeitliche Kalibrierung benutzten Bewegung.

**Vergleich** Für diesen Versuch wurden die Daten vorab aufgezeichnet und anschließend zwei Datenflüsse in der Software ubitrack konstruiert. Ein Datenfluss gibt die Daten in dem gleichen Koordinatensystem aus, ohne die relative Latenz zu korrigieren. Der zweite Datenfluss führt auch eine Koordinatentransformation durch, korrigiert aber gleichzeitig die relative Latenz. Dies geschieht durch Verzögerung des Datenstroms des A.R.T. Systems.

Für die Evaluation wurden der FaroArm auf mehrere Kreisbahnen mit annähernd konstanter Geschwindigkeit bewegt. Die Durchschnittsgeschwindigkeit betrug dabei  $1 \text{ ms}^{-1}$  und die Standardabweichung hiervon  $0,13 \text{ ms}^{-1}$ . In Abbildung 9.4 sind diese Kreisbahnen dargestellt.

In dieser Untersuchung wurde als Fehler die Norm der Differenz der von dem FaroArm und der von dem A.R.T. System gemessenen System gewählt. Nachdem beide Systeme unterschiedliche Wiederholraten haben und die Daten nicht zur gleichen Zeit liefern, müssen die beiden Systeme zu einander synchronisiert werden. Dies geschieht durch eine lineare Interpolation der dreidimensionalen Positionsdaten.

Durch die Korrektur der relativen Latenz wurde der Fehler des A.R.T. Systems nahezu um den Faktor 22 gesenkt. Vor der Korrektur betrug der durchschnittliche

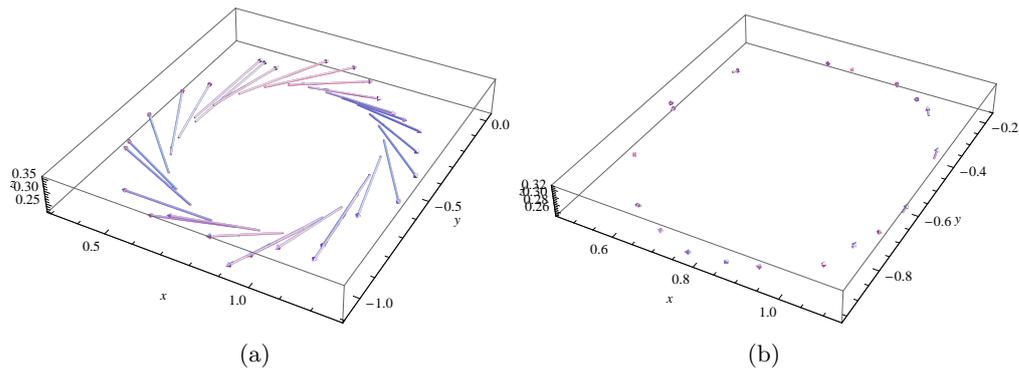


Abbildung 9.5: Darstellung des Trackingfehlers mit Hilfe eines Pfeils von A.R.T. gemessenen Markerposition zu der von dem FaroArm vermessenen Position. Die Länge der Pfeile repräsentiert dabei die um den Faktor zehn vergrößerten Fehlers. (b) zeigt den Fehler ohne Korrektur der relativen Latenz; (a) zeigt den Fehler nach der Korrektur der relativen Latenz von  $\delta = 32,1$  ms

Fehler 31,8 mm mit einer Standardabweichung von 4,5 mm. Nach der Korrektur betrug der durchschnittliche Fehler nur noch 1,46 mm mit einer Standardabweichung von 0,55 mm. Somit konnte auch die Standardabweichung durch die Korrektur reduziert werden. Wobei diese erzielten Ergebnisse stark von der gewählten Geschwindigkeit abhängen. Die Abbildung 9.6 zeigt die Ergebnisse der Korrektur graphisch.

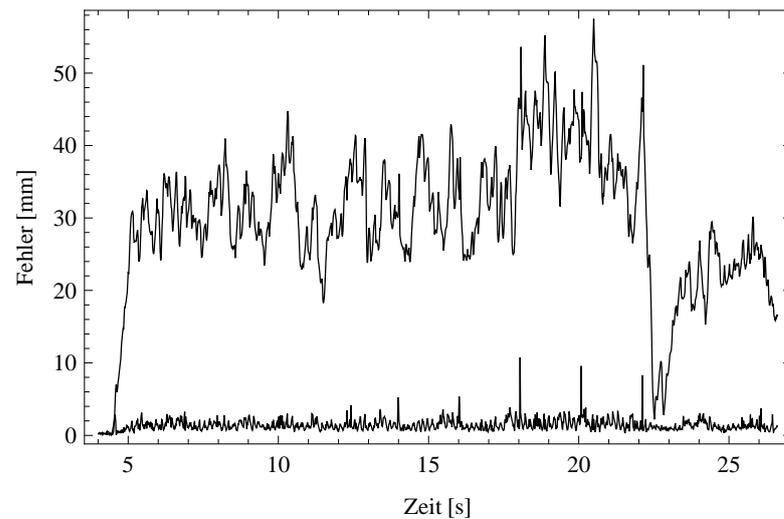


Abbildung 9.6: Auswirkung der Korrektur der relativen Latenz bei einem bewegten Sensorpaar. Die Bewegung ist in Abbildung 9.5 dargestellt. Die beiden Graphen zeigen jeweils den Fehler als geometrischen Abstand zwischen den Messungen des FaroArms und den Messungen des A.R.T. Systems bei der gleichen Bewegung. Der obere Graph repräsentiert den ursprünglichen Fehler, der untere Graph zeigt den Fehler nach der Korrektur der relativen Latenz.



# 10 Zusammenfassung

Das Ziel dieser Arbeit war es die Machbarkeit und Anwendbarkeit von Methoden für eine zeitliche Kalibrierung in Augmented Reality Anwendungen zu zeigen. Dabei wurde neben der Anwendbarkeit der Korrelationsmethode für diesen Zweck der Fokus auch auf die Anwendbarkeit dieser Methodik während der Laufzeit einer AR Anwendung gelegt. Zum Ende dieser Arbeit diskutiert dieses Kapitel noch einmal kurz die wichtigsten Aspekte der vorgestellten Methoden.

## 10.1 Diskussion

Die vorgestellten Methoden und Auswertungen haben gezeigt, dass eine Kalibrierung der relativen Latenz zwischen zwei Sensoren in der Augmented Reality mit der Kreuzkorrelation möglich ist. Doch diese Methode, wie auch andere Methoden, sind anfällig für Einflüsse von räumlicher Registrierung, Rauschen, Ausreißern und anderen Ungenauigkeiten des Trackings. Es ist für eine genaue Kalibrierung nötig, genau diese Einflüsse auf die Kalibrierung zu verringern.

Es wurde beschrieben, wie die Sonderfälle erkannt werden können, die eine Kalibrierung ungenau machen bzw. auch verhindern. Dabei wurde darauf geachtet, dass die Schritte während der Laufzeit durchgeführt werden können und dabei die Erhöhung der Gesamtlatenz der Anwendung möglichst gering zu halten. Dabei wurden die einzelnen Schritte auf ihre Eignung für die Kalibrierung der relativen Latenz untersucht und dabei Bezug auf den aktuellen Stand der Forschung genommen. Es wurde die ganze Prozesskette, von der Aufnahme der Daten über die Eliminierung der Ausreißer, Rauschen, Dimensionsreduktion und die zeitliche Kalibrierung, als zentrale Komponente, betrachtet. Auch auf die Interpretation der somit gewonnen Daten wurde eingegangen. Für die Dimensionsreduktion wurde in dieser Arbeit die Norm als Reduktionsfunktion erfolgreich genutzt. Aber es wurden auch weitere Funktionen in dieser Arbeit vorgestellt und evaluiert, die eine weitaus größere Robustheit gegen störende Einflüsse aufweisen. Auch auf die Schritte der Synchronisierung der Daten wurde vertieft eingegangen und dort untersucht, wie diese effizient und dennoch genau durchgeführt werden kann. Für die zentrale Komponente der zeitlichen Kalibrierung wurde die Kreuzkorrelation gewählt. Diese Methode wird in anderen zahlreichen Feldern erfolgreich benutzt und es wurde gezeigt, dass diese Methode auch für die Kalibrierung der relativen Latenz erfolgreich genutzt werden kann. Es wurde gezeigt, dass diese Methode vergleichbare Ergebnisse liefert wie der in dem Bereich der Augmented Reality schon eingesetzte geometrische Vergleich der beiden Messdatenströme. Zudem wur-

den auch zwei Maße angegeben, die es erlauben die geschätzte relative Latenz qualitativ zu beurteilen und somit Rückschlüsse auf die Sicherheit der Schätzung zu ziehen. Neben der reinen Fähigkeit die relative Latenz zwischen zwei Sensoren zu bestimmen, werden die Methoden in dieser Arbeit auch auf die Anwendbarkeit während eines Trackingvorgangs hin untersucht. Dazu muss zum einen auf die Rechenzeiten der einzelnen Verfahren geachtet werden und zum anderen muss sich das Verfahren dynamisch der aktuellen Trackingsituation anpassen können. Neben der Erkennung von Situationen in denen eine Kalibrierung nicht möglich ist, wird gezeigt wie die Verfahren auch mit anderen nicht trivialen pathologischen Fällen umgehen können.

Nach der Beschreibung der einzelnen Schritte für eine zeitliche Kalibrierung wird ein Ansatz mit Hilfe der Monte-Carlo Simulation vorgestellt und dabei ausführlich auf die einzelnen zu berücksichtigenden Eingangsparameter, wie Rauschen, Latenz, ungenaue Registrierung etc. eingegangen. Der Ansatz der Monte-Carlo Simulation erlaubt es jeden Fehler getrennt von anderen Störeinflüssen zu untersuchen und die Ergebnisse unter Kenntnis der genauen relativen Latenz zu untersuchen. Diese theoretische Simulation erlaubt es verschiedene Methoden auf die erreichbare Genauigkeit hin zu untersuchen und diese Simulation unter genau kontrollierbaren Umgebungssituation durchführen zu können. Es soll jedoch nicht verschwiegen werden, dass die Möglichkeit Fehlerquellen getrennt voneinander zu untersuchen neben den Vorteilen auch nicht freiwillig geschieht. Denn würden alle Fehlerquellen gleichzeitig in einer Simulation betrachtet, würde die Komplexität für eine signifikante Simulation zu groß werden. Dabei können die Ergebnisse der Simulation nur Hinweise geben, die mit realen Messungen nachgewiesen werden müssen. Jedoch erleichtert die vorgestellte Simulation die Untersuchung der Auswirkung einzelner Einflussgrößen.

Am Ende dieser Arbeit werden ein einige Anwendungen für die zeitliche Kalibrierung beschrieben. Diese Anwendungen zeigen, die Anwendbarkeit der hier vorgestellten Methoden und die positiven Auswirkungen auf die Genauigkeit.

## 10.2 Weiteres Vorgehen

Es gibt zahlreiche Ansatzpunkte die hier vorgestellten Methoden noch weiter zu verbessern. Die Simulation kann beschleunigt werden, wenn sie hardwarenäher implementiert wird. Zur Zeit ist die Simulation in Mathematica implementiert und arbeitet daher nicht sehr performant, das diese Software nicht schnelle numerische Operationen ausgelegt wurde.

Um das Verfahren robuster gegen die ungenaue Registrierungen zu gestalten, können die Methoden der Dimensionsreduktion weiterentwickelt werden. So kann die Kanonische Korrelation genauer untersucht werden, welche eine Invarianz gegenüber affinen Transformationen verspricht. Dies beinhaltet auch die Rotationen und Translationen im dreidimensionalen Raum. Zu klären ist hier, welche Genauigkeiten die gelieferten Ergebnisse ausweise und welche Varianz sie haben.

Des Weiteren verspricht die Hauptkomponentenanalyse zur Bestimmung optimaler Projektionsvektoren einen gewissen Grad an Unabhängigkeit in Bezug auf die räumlichen Registrierungen. Neben einer tieferen Untersuchung der Hauptkomponentenanalyse für Positionen, sollte diese auch für eine Kalibrierung anhand von Rotationen evaluiert werden.

Bei der eigentlichen Bestimmung der relativen Latenz sollten die Auswirkungen der Anwendung der Methoden von Knapp und Carter [18] und anderer Methoden auf die Genauigkeit untersucht werden. Durch diese Methoden lässt sich die SNR Werte der Signale steigern und somit eine genauere Kalibrierung durchführen. Auch andere Arbeiten in dem Gebiet der Signalverarbeitung befassen sich mit einer optimalen Filterung.

Neben den zahlreichen Optimierungen der vorliegenden Signale ist aber auch eine valide Schätzung der zu erwartenden Genauigkeiten vonnöten. Die präsentierten Vorschläge können zwar die Qualität einer Schätzung beurteilen. Sie liefern aber kein absolutes Maß für die Genauigkeit. Auch hierfür bedarf es weiterer Arbeiten.



# Literaturverzeichnis

- [1] ARON, M. ; SIMON, G. ; BERGER, M. ; LORIA, I. ; NANCY, F. : Handling Uncertain Sensor Data in Vision-Based Camera Tracking. In: *Third IEEE and ACM International Symposium on Mixed and Augmented Reality, 2004. ISMAR 2004*, 2004, S. 58–67 (3.1, 3.1, 4.2, 6.3, 7.2)
- [2] AZUMA, R. ; BAILLOT, Y. ; BEHRINGER, R. ; FEINER, S. ; JULIER, S. ; MACINTYRE, B. : Recent Advances in Augmented Reality. In: *Computer Graphics and Applications, IEEE* 21 (2001), Nr. 6, S. 34–47 (1.1, 4.1, 4.5)
- [3] AZUMA, R. : A Survey of Augmented Reality. In: *Presence-Teleoperators and Virtual Environments* 6 (1997), Nr. 4, S. 355–385 (1.1, 2.3.2, 3.3, 3.7, 4.5)
- [4] AZUMA, R. ; BISHOP, G. : Improving Static and Dynamic Registration in an Optical See-through HMD. In: *SIGGRAPH*, 1994, S. 197–204 (4.2, 4.2)
- [5] AZUMA, R. T.: *Predictive Tracking for Augmented Reality*. Chapel Hill, University of North Carolina, Diss., 1995. – 1–262 S. (1.2, 3.7, 6.1)
- [6] BAJURA, M. ; NEUMANN, U. : Dynamic Registration Correction in Augmented-Reality Systems. In: *Virtual Reality Annual International Symposium* Published by the IEEE Computer Society, 1995, S. 189–196 (3.7)
- [7] BAUER, M. A.: *Tracking Errors in Augmented Reality*, Institut für Informatik der Technischen Universität München, Diss., April 2007 (2.3.2, 8.3.2, 8.3.2)
- [8] BAUER, M. A. ; BRUEGGE, B. ; KLINKER, G. ; MACWILLIAMS, A. ; REICHER, T. ; RISS, S. ; SANDOR, C. ; WAGNER, M. : Design of a Component-Based Augmented Reality Framework. In: *Proceedings of the International Symposium on Augmented Reality (ISAR)* (2.6.3)
- [9] BAUER, M. A. ; SCHLEGEL, M. ; PUSTKA, D. ; NAVAB, N. ; KLINKER, G. : Predicting and Estimating the Accuracy of Vision-Based Optical Tracking Systems. In: *Proc. IEEE International Symposium on Mixed and Augmented Reality (ISMAR'06)*. Santa Barbara (CA), USA, October 2006 (8.3.2)
- [10] BISHOP, G. ; WELCH, G. ; ALLEN, B. D.: Tracking: Beyond 15 Minutes of Thought. In: *SIGGRAPH Course Pack* (2001) (2.5.2)

- [11] BLESER, G. ; STRICKER, D. : Advanced tracking through efficient image processing and visual-inertial sensor fusion. In: *Computers & Graphics* (2008) (3.1, 4.3)
- [12] BORGA, M. : *Learning Multidimensional Signal Processing*. Linköping, Linköping University, Diss., 1998 (5.3.1)
- [13] BORGA, M. : *Canonical Correlation: A Tutorial*. Online, 2001 (4.5)
- [14] BORTZ, J. : *Statistik: für Human-und Sozialwissenschaftler*. Springer-Verlag, 2004 (5.3.3, 5.3.3)
- [15] BRONSTEIN, I. ; SEMENDJAJEW, K. ; MUSIOL, G. ; MÜHLIG, H. : *Taschenbuch der Mathematik*. Verlag Harri Deutsch-Thun, Frankfurt/M, 1993 (7.5)
- [16] BROOKS, R. R. ; IYENGAR, S. S.: Real-Time Distributed Sensor Fusion for Time-Critical Sensor Readings. In: *Optical Engineering* 36 (1997), Nr. 3, 767-779. <http://dx.doi.org/10.1117/1.601274>. – DOI 10.1117/1.601274 (2.5.1)
- [17] CARTER, G. : Time Delay Estimation for Passive Sonar Signal Processing. In: *Acoustics, Speech and Signal Processing, IEEE Transactions on* 29 (1981), jun., Nr. 3, S. 463 – 470. <http://dx.doi.org/10.1109/TASSP.1981.1163560>. – DOI 10.1109/TASSP.1981.1163560. – ISSN 0096–3518 (1.3, 7.1)
- [18] CARTER, G. ; KNAPP, C. : Time Delay Estimation. In: *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'76*, 1976 (4.5, 4.5, 4.5, 7.2, 7.4.2, 10.2)
- [19] CARTER, G. : Coherence and Time Delay Estimation. In: *Proceedings of the IEEE* 75 (1987), Nr. 2, S. 236–255 (4.5, 4.5, 4.5)
- [20] COX, M. ; HARRIS, P. : Software Support for Metrology Best Practice Guide No. 6 Uncertainty Evaluation / National Physical Laboratory. Teddington, UK, 2006 (DEM-ES-011). – Forschungsbericht (8.2, 8.1)
- [21] CRAIG, J. J.: *Introduction to Robotics Mechanics & Control*. Addison-Wesley, 1986 (2.3.1, 2.3.1, 5.2.3)
- [22] DAM, E. ; KOCH, M. ; LILLHOLM, M. : Quaternions, Interpolation and Animation / Department of Computer Science, University of Copenhagen, Denmark. 1998. – Forschungsbericht (6.1.2)
- [23] DANILIDIS, K. : Hand-Eye Calibration Using Dual Quaternions. In: *The International Journal of Robotics Research* 18 (1999), Nr. 3, S. 286 (2.3.2, 4.1)

- [24] DURRANT-WHYTE, H. : Sensor Models and Multisensor Integration. In: *The International Journal of Robotics Research* 7 (1988), Nr. 6, S. 97 (2.5.1)
- [25] ECHTLER, F. ; HUBER, M. ; PUSTKA, D. ; KEITLER, P. ; KLINKER, G. : Splitting the Scene Graph - Using Spatial Relationship Graphs Instead of Scene Graphs in Augmented Reality. In: *Proceedings of the 3rd International Conference on Computer Graphics Theory and Applications (GRAPP)*, 2008 (2.2)
- [26] ECHTLER, F. ; STURM, F. ; KINDERMANN, K. ; KLINKER, G. ; STILLA, J. ; TRILK, J. ; NAJAFI, H. : The Intelligent Welding Gun: Augmented Reality for Experimental Vehicle Construction. In: ONG, S. (Hrsg.) ; NEE, A. (Hrsg.): *Virtual and Augmented Reality Applications in Manufacturing, Chapter 17*. Springer Verlag, 2003 (2.5.1)
- [27] FERTNER, A. ; SJOLUND, A. : Comparison of Various Time Delay Estimation Methods by Computer Simulation. In: *IEEE Transactions on Acoustics, Speech and Signal Processing* 34 (1986), Nr. 5, S. 1329–1330 (7.3.1, 8.1)
- [28] FISCHLER, M. ; BOLLES, R. : Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. In: *Communications of the ACM* 24 (1981), Nr. 6, S. 381–395 (4.3)
- [29] FODOR, I. : A Survey of Dimension Reduction Techniques / US DOE Office of Scientific and Technical Information. 2002 (UCRL-ID-148494). – Forschungsbericht (5.3.2, 5.3.3, 5.3.3)
- [30] FOX, D. ; HIGHTOWER, J. ; KAUZ, H. ; LIAO, L. ; PATTERSON, D. : Bayesian Techniques for Location Estimation. In: *Proceedings of The 2003 Workshop on Location-Aware Computing*, 2003, S. 16 (2.5.2)
- [31] FOX, D. ; HIGHTOWER, J. ; LIAO, L. ; SCHULZ, D. ; BORRIELLO, G. : Bayesian Filtering for Location Estimation. In: *IEEE Pervasive Computing* 2 (2003), Nr. 3, S. 24–33 (2.5.2, 2.6.5)
- [32] FRERKING, M. : *Digital Signal Processing in Communication Systems*. Van Nostrand Reinhold, 1994 (2.2)
- [33] GRASSIA, F. S.: Practical Parameterization of Rotations Using the Exponential Map. In: *Journal of Graphics Tools: JGT* 3 (1998), Nr. 3, S. 29–48. – ISSN 1086–7651 (2.3.1, 5.2.3, 6.1.2)
- [34] GRAUMANN, D. ; LARA, W. ; HIGHTOWER, J. ; BORRIELLO, G. : Real-World Implementation of the Location Stack: The Universal Location Framework. In: *Fifth IEEE Workshop on Mobile Computing Systems and Applications*, 2003, S. 122–128 (2.6.5)

- [35] GRUBER, L. ; GAUGLITZ, S. ; VENTURA, J. ; ZOLLMANN, S. ; HUBER, M. ; SCHLEGEL, M. ; KLINKERM, G. ; SCHMALSTIEG, D. ; HÖLLERER, T. : The City of Sights: Designing an Augmented Reality Stage Set. In: *Proc. IEEE International Symposium on Mixed and Augmented Reality (ISMAR'10)*, 2010 (3.7, 8.3.2, 9.1)
- [36] HALLAWAY, D. ; FEINER, S. ; HÖLLERER, T. : Bridging the Gaps: Hybrid Tracking for Adaptive Mobile Augmented Reality. In: *Applied Artificial Intelligence* 18 (2004), Nr. 6, S. 477–500 (2.5.1)
- [37] HAMILTON, W. R.: On Quaternions. In: WILKINS, D. R. (Hrsg.): *Proceedings of the Royal Irish Academy* Bd. 3, 1847, S. 1–16 (2.3.1)
- [38] HARTLEY, R. ; ZISSERMAN, A. : *Multiple View Geometry in Computer Vision*. Cambridge Univ Pr, 2003 (4.3, 5.3.1, 8.3.2)
- [39] HASTEDT, H. : Monte-Carlo-Simulation in Close-Range Photogrammetry. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science* 35 (2004), S. 18–23 (8.1)
- [40] HE, D. ; LIU, F. ; PAPE, D. ; DAWE, G. ; SANDIN, D. : Video-Based Measurement of System Latency. In: *International Immersive Projection Technology Workshop* Citeseer, 2000 (3.1)
- [41] HIGHTOWER, J. ; BORRIELLO, G. : Particle Filters for Location Estimation in Ubiquitous Computing: A Case Study. In: *Lecture notes in computer science* (2004), S. 88–106 (2.5.2)
- [42] HIGHTOWER, J. ; BRUMITT, B. ; BORRIELLO, G. : The Location Stack: A Layered Model for Location in Ubiquitous Computing. In: *Proceedings of the 4th IEEE Workshop on Mobile Computing Systems & Applications (WMCSA 2002)*, 2002, S. 22–28 (2.6.5)
- [43] HOFF, W. ; VINCENT, T. : Analysis of Head Pose Accuracy in Augmented Reality. In: *IEEE Transactions on Visualization and Computer Graphics* Bd. 6(4). IEEE Computer Society, 2000, S. 319–334 (8.3.2)
- [44] HOLLOWAY, R. : *Registration Errors in Augmented Reality*, UNC Chapel Hill Department of Computer Science, Diss., August 1995 (2.3.2)
- [45] HORAUD, R. ; DORNAIKA, F. : Hand-Eye Calibration. In: *The International Journal of Robotics Research* 14 (1995), Nr. 3, S. 195 (2.3.2)
- [46] HORN, B. K. P.: Closed-Form Solution of Absolute Orientation Using Unit Quaternions. In: *Journal of the Optical Society of America* 4 (1987), Apr., Nr. 4 (3.2, 4.1, 9.3)
- [47] HORN, B. ; SCHUNCK, B. : Determining Optical Flow. In: *Artificial Intelligence* 17 (1981), Nr. 1-3, S. 185–203 (2.4, 3.1)

- 
- [48] HOTELLING, H. : Relations between Two Sets of Variates. In: *Biometrika* 28 (1936), Nr. 3-4, S. 321 (5.3.1)
- [49] HUBER, M. : *Parasitic Tracking for Ubiquitous Augmented Reality*. München, Technische Universität München, Diss., 2011 (1.3, 2.4, 2.5.1, 4.2, 9.2)
- [50] HUBER, M. ; PUSTKA, D. ; KEITLER, P. ; FLORIAN, E. ; KLINKER, G. : A System Architecture for Ubiquitous Tracking Environments. In: *Proceedings of the 6th International Symposium on Mixed and Augmented Reality (ISMAR)*, 2007 (2.2, 2.6.4, 4.2, 6.1)
- [51] JACOBS, M. ; LIVINGSTON, M. ; STATE, A. : Managing Latency in Complex Augmented Reality Systems. In: *Proceedings of the Symposium on Interactive 3D Graphics* ACM, 1997 (3.1, 3.1, 3.1, 3.7, 3.7, 4.2, 7.2, 9.1)
- [52] JACOVITTI, G. ; SCARANO, G. : Discrete Time Techniques for Time Delay Estimation. In: *IEEE Transactions on Signal Processing* 41 (1993), Nr. 2, S. 525–533 (7.2.2, 7.3.1, 7.4.2, 8.1)
- [53] JAY, C. ; HUBBOLD, R. : Delayed Visual and Haptic Feedback in a Reciprocal Tapping Task. In: *Eurohaptics Conference, 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. World Haptics 2005. First Joint IEEE*, 2005, S. 655–656 (2.7)
- [54] JOHANSSON, B. : *Multidimensional Signal Recognition, Invariant to Affine Transformation and Time-Shift, using Canonical Correlation*. SE-581 83 Linköping, Sweden, Linköping University, Diplomarbeit, October 1997 (3.1, 5.3.1, 5.3.1, 5.3.1, 5.2, 7.2, 7.4.2)
- [55] KALMAN, R. : A New Approach to Linear Filtering and Prediction Problems. In: *Journal of Basic Engineering* 82 (1960), Nr. 1, S. 35–45 (2.5.2)
- [56] KAMMEYER, K. ; KROSCHER, K. : *Digitale Signalverarbeitung: Filterung und Spektralanalyse; mit Matlab-Übungen*. Teubner, 1998 (2.3.3)
- [57] KANTONEN, T. : Sensor Synchronization for AR Applications. In: *IEEE/ACM International Symposium on Mixed and Augmented Reality, Poster Presentation, ISMAR 2010*, 2010 (4.2)
- [58] KATO, H. ; BILLINGHURST, M. : Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System. In: *2nd IEEE and ACM International Workshop on Augmented Reality*, 1999, S. 85 (2.4)
- [59] KEITLER, P. ; BECKER, B. ; KLINKER, G. : Management of Tracking for Industrial AR Setups. In: *Proceedings of the 9th International Symposium on Mixed and Augmented Reality (ISMAR)* IEEE, 2010, S. 73–82 (8.3.2, 8.3.2, 9.3)
-

- [60] *Kapitel Management of Tracking for Mixed and Augmented Reality Systems.* In: KEITLER, P. ; PUSTKA, D. ; HUBER, M. ; ECHTLER, F. ; KLINKER, G. : *Engineering of Mixed Reality Systems.* Dubois, E. and Gray, P. and Nigay, L., 2009 (2.6.4)
- [61] KEITLER, P. ; SCHLEGEL, M. ; KLINKER, G. : Indirect Tracking to Reduce Occlusion Problems. In: *Advances in Visual Computing, Fourth International Symposium, ISVC 2008 Las Vegas, USA, December 1-3* Bd. 5359(2). Berlin : Springer, 2008, S. 224–235 (2.5.1)
- [62] KEITLER, P. : *Management of Tracking and Tracking Accuracy in Industrial Augmented Reality Environments,* Technische Universität München, Diss., 2011 (2.6.4, 2.9, 8.3.2, 8.3.2, 8.3.2)
- [63] KLINKER, G. ; DUTOIT, A. ; BAUER, M. A. ; BAYER, J. ; NOVAK, V. ; MATZKE, D. : Fata Morgana – A Presentation System for Product Design. In: *Proceedings of the 1st International Symposium on Mixed and Augmented Reality* IEEE Computer Society, 2002, S. 76 (2.3.3)
- [64] KUIPERS, J. B.: *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality.* Princeton Univ Pr, 2002 (2.3.1, 2.3.1, 5.2.3, 6.1.2)
- [65] LIEBERKNECHT, S. ; BENHIMANE, S. ; MEIER, P. ; NAVAB, N. : A Dataset and Evaluation Methodology for Template-Based Tracking Algorithms. In: *Proceedings of the 2009 8th IEEE International Symposium on Mixed and Augmented Reality* IEEE Computer Society, 2009, S. 145–151 (1.3, 2.4, 3.1, 3.2, 3.5, 3.7, 6.1, 6.1, 6.1.2, 7.2, 7.3.2)
- [66] LIPPOLD, O. : Physiological Tremor. In: *Scientific American* 224 (1971), Nr. 3, S. 65–73 (4.4)
- [67] MACWILLIAMS, A. : *A Decentralized Adaptive Architecture for Ubiquitous Augmented Reality Systems.* München, Technischen Universität München, Diss., 2005 (2.6.3)
- [68] MEINHOLD, R. ; SINGPURWALLA, N. : Robustification of Kalman Filter Models. In: *Journal of the American Statistical Association* 84 (1989), Nr. 406, S. 479–486 (4.3)
- [69] METROPOLIS, N. : The Beginning of the Monte Carlo Method. In: *Los Alamos Science* 15 (1987), S. 125–130 (8.1)
- [70] MILGRAM, P. ; KISHINO, F. : A Taxonomy of Mixed Reality Visual Displays. In: *IEICE Transactions on Information and Systems* E77-D (1994), Nr. 12, S. 1321–1329 (1.1, 1.3)

- 
- [71] MILLER, D. ; BISHOP, G. : Latency Meter: A Device to Measure End-to-End Latency of Ve Systems. In: *Proceedings of SPIE-Stereoscopic Displays and Virtual Reality Systems*. San Jose, CA, USA, January 2002 (3.1)
- [72] MITZENMACHER, M. ; RICHA, A. ; SITARAMAN, R. : The Power of Two Random Choices: A Survey of Techniques and Results. In: *Handbook of Randomized Computing 1* (2001), S. 255–312 (5.3.2)
- [73] MODDEMEIJER, R. : On the Determination of the Position of Extrema of Sampled Correlators. In: *IEEE Transactions on Signal Processing* 39 (1991), Nr. 1, S. 216–219 (7.2.2)
- [74] MONTGOMERY, D. C.: *Design and Analysis of Experiments*. Wiley New York, 1991 (8.1)
- [75] NEWMAN, J. ; WAGNER, M. ; BAUER, M. A. ; MACWILLIAMS, A. ; PINTARIC, T. ; BEYER, D. ; PUSTKA, D. ; STRASSER, F. ; SCHMALSTIEG, D. ; KLINKER, G. : Ubiquitous Tracking for Augmented Reality. In: *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2004, S. 192–201 (2.2, 2.6.4)
- [76] PATWARI, N. ; ASH, J. ; KYPEROUNTAS, S. ; HERO, A. ; MOSES, R. ; CORREAL, N. : Locating the Nodes. In: *IEEE Signal processing magazine* 22 (2005), Nr. 4, S. 54 (7.4.1, 7.4.1)
- [77] PENTENRIEDER, K. ; BADE, C. ; DOIL, F. ; MEIER, P. : Augmented Reality-Based Factory Planning - An Application Tailored to Industrial Needs. In: *6th IEEE and ACM International Symposium on Mixed and Augmented Reality, 2007. ISMAR 2007*, 2007, S. 31–42 (2.3.3, 2.4)
- [78] PENTENRIEDER, K. ; MEIER, P. ; KLINKER, G. : Analysis of Tracking Accuracy for Single-Camera Square-Marker-Based Tracking. In: *Dritter Workshop Virtuelle und Erweiterte Realitt der GI-Fachgruppe VR/AR, Koblenz, Germany*, 2006 (2.3.2, 2.4)
- [79] PRAGER, R. ; GEE, A. ; BERMAN, L. : STRADX: Real-Time Acquisition and Visualisation of Freehand 3D Ultrasound / University of Cambridge. 1998 (CUED / F-INFENG / TR 319). – Forschungsbericht. – 129–140 S. (3.1, 4.2, 7.2)
- [80] PUSTKA, D. : *Handling Error in Ubiquitous Tracking Setups*, Technische Universität München, Diplomarbeit, 2004 (5.2.3, 5.2.4)
- [81] PUSTKA, D. : Construction of Data Flow Networks for Tracking in Augmented Reality Applications. In: *Proc. Dritter Workshop Virtuelle und Erweiterte Realität der GI-Fachgruppe VR/AR*. Koblenz, Germany, September 2006 (2.6.4, 3.5, 4.2, 6.1)
-

- [82] PUSTKA, D. ; HUBER, M. ; BAUER, M. A. ; KLINKER, G. : Spatial Relationship Patterns: Elements of Reusable Tracking and Calibration Systems. In: *Proc. IEEE International Symposium on Mixed and Augmented Reality (ISMAR'06)*, 2006 (2.2)
- [83] PUSTKA, D. ; HUBER, M. ; KLINKER, G. : Integrating Gyroscopes into Ubiquitous Tracking Environments. In: *Proceedings of IEEE Virtual Reality 2008*. Reno, Nevada, USA, März 2008 (2.4, 6.3)
- [84] REITMAYR, G. ; DRUMMOND, T. : Going out: Robust Model-based Tracking for Outdoor Augmented Reality. In: *IEEE/ACM International Symposium on Mixed and Augmented Reality, 2006. ISMAR 2006*, 2006, S. 109–118 (6.3)
- [85] REITMAYR, G. ; SCHMALSTIEG, D. : Opentracker – An Open Software Architecture for Reconfigurable Tracking Based on XML. In: *Proceedings of the ACM Symposium on Virtual Reality Software & Technology (VRST)*. Banff, Alberta, Canada, 2001, S. 47–54 (2.6.1)
- [86] RINNE, H. : *Taschenbuch der Statistik*. Harri Deutsch Verlag, 2008 (5.3.3, 5.3.3, 7.5, 8.2, 8.2, 8.2, 8.3.2)
- [87] RISTIC, B. ; ARULAMPALAM, S. ; GORDON, N. : *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House Publishers, 2004 (2.5.2)
- [88] ROHLING, R. ; MUNGER, P. ; HOLLERBACH, J. ; PETERS, T. : Comparison of Relative Accuracy Between a Mechanical and an Optical Position Tracker for Image-Guided Neurosurgery. In: *Journal of image guided surgery* 1 (1995), Nr. 1, S. 30–34 (2.4)
- [89] ROLLAND, J. ; DAVIS, L. ; BAILLOT, Y. : A Survey of Tracking Technology for Virtual Environments. In: *Fundamentals of Wearable Computers and Augmented Reality* (2001), S. 67–112 (2.1)
- [90] SCHICKINGER, T. ; STEGER, A. : *Diskrete Strukturen: Band 2: Wahrscheinlichkeitstheorie und Statistik*. Springer, 2001. – ISBN 354067599X (7.3.1, 7.5)
- [91] SCHLEGEL, M. : *Predicting the Accuracy of Optical Tracking Systems*, Technische Universität München, Diplomarbeit, 2006 (2.3.2, 4.5, 8.3.2, 8.3.2, 8.3.2)
- [92] SCHMALSTIEG, D. ; FUHRMANN, A. ; HESINA, G. ; SZALAVÁRI, Z. ; ENCARNACAO, L. ; GERVAUTZ, M. ; PURGATHOFER, W. : The Studierstube Augmented Reality Project. In: *PRESENCE – Teleoperators and Virtual Environments* 11 (2002), Nr. 1, S. 32–45 (2.6.1)

- [93] SCHWALD, B. ; SEIBERT, H. : Registration Tasks for a Hybrid Tracking System for Medical Augmented Reality. In: *Journal of WSCG* 12 (2004), S. 411–418 (1.3, 3.1, 3.2, 7.3.2)
- [94] SCHWARZ, H. R.: *Numerische Mathematik*. Stuttgart : B. G. Teubner, 1993 (6.1, 7.4.2)
- [95] SCHWERDTFEGER, B. : *Pick-by-Vision: Bringing HMD-Based Augmented Reality into the Warehouse*. München, Technische Universität München, Dissertation, 2010 (2.1)
- [96] SHOEMAKE, K. : Animating Rotation with Quaternion Curves. In: *Proceedings of the International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '85)*, 1985 (6.1.2)
- [97] SIELHORST, T. ; FEUERSTEIN, M. ; TRAUB, J. ; KUTTER, O. ; NAVAB, N. : CAMPAR: A Software Framework Guaranteeing Quality for Medical Augmented Reality. In: *International Journal of Computer Assisted Radiology and Surgery* 1 (2006), S. 29 (2.6.2)
- [98] SIELHORST, T. ; SA, W. ; KHAMENE, A. ; SAUER, F. ; NAVAB, N. : Measurement of Absolute Latency for Video See-Through Augmented Reality. In: *Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, 2007 (3.1, 3.1)
- [99] SIVRIKAYA, F. ; YENER, B. : Time Synchronization in Sensor Networks: A Survey. In: *IEEE Network* 18 (2004), Nr. 4, S. 45–50 (4.2)
- [100] SUTHERLAND, I. : The Ultimate Display. In: *Proceedings of the IFIP Congress Bd. 2*, 1965 (1.1)
- [101] SUTHERLAND, I. : A Head-Mounted Three Dimensional Display. In: *Proceedings of the Fall Joint Computer Conference, Part I ACM*, 1968, S. 757–764 (1.1, 1.2)
- [102] TÖNNIS, M. : *Augmented Reality: Einblicke in die Erweiterte Realität*. Berlin : Springer Verlag, 2010 (4.5)
- [103] TÖNNIS, M. ; LINDL, R. ; WALCHSHAEUSL, L. ; KLINKER, G. : Visualization of Spatial Sensor Data in the Context of Automotive Environment Perception Systems. In: *Proceedings of the 6th International Symposium on Mixed and Augmented Reality (ISMAR)*, 2007 (1.1)
- [104] TSAI, R. Y. ; LENZ, R. K.: A New Technique for Fully Autonomous and Efficient 3D Robotics Hand/Eye Calibration. In: *IEEE Journal of Robotics and Automation* 5 (1989), Jun., Nr. 3, S. 345–358 (4.1)

- [105] TUCERYAN, M. ; GREER, D. S. ; WHITAKER, R. T. ; BREEN, D. E. ; ROSE, E. ; AHLERS, K. H. ; CRAMPTON, C. : Calibration Requirements and Procedures for a Monitor-Based Augmented Reality System. In: *IEEE Trans. Vis. and Comp. Graph.* 1 (1995), Nr. 3 (2.2)
- [106] UNBEHAUEN, R. : *Systemtheorie 1*. München : R. Oldenbourg Verlag, 1997 (2.3.3, 2.3.3)
- [107] VACCARO, R. : The past, Present, and the Future of Underwater Acoustic Signal Processing. In: *Signal Processing Magazine, IEEE* 15 (1998), Jul., Nr. 4, S. 21 –51. <http://dx.doi.org/10.1109/79.689583>. – DOI 10.1109/79.689583. – ISSN 1053–5888 (1.3, 7.1)
- [108] *Optical 3D Measuring System - Imaging Systems with Point-by-Point Probing*. VDI/VDE guideline 2634/1, 2002 (9.3)
- [109] WAECHTER, C. ; HUBER, M. ; KEITLER, P. ; SCHLEGEL, M. ; PUSTKA, D. ; KLINKER, G. : A Multi-Sensor Platform for Wide-Area Tracking. In: *9th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2010)*, 2010 (2.4, 2.5.1, 3.4)
- [110] WAGNER, M. : *Tracking with Multiple Sensors*, Technische Universität München, Diss., 2005 (2.5.1)
- [111] WEISSTEIN, E. W.: *CRC Concise Encyclopedia of Mathematics*. Chapman Hall CRC, 1998 (4.5, 5.3.1, 7.3.1)
- [112] WELCH, G. ; BISHOP, G. : An Introduction to the Kalman Filter / Department of Computer Science, University of North Carolina at Chapel Hill. Technical Report TR 95-041, University of North Carolina, Department of Computer Science, 1995, 2001 (TR 95-041). – Forschungsbericht (2.5.2)
- [113] WELCH, G. ; FOXLIN, E. : Motion Tracking: No Silver Bullet, but a Respectable Arsenal. In: *Computer Graphics and Applications, IEEE* 22 (2002), Nov.-Dec., Nr. 6, S. 24 –38. <http://dx.doi.org/10.1109/MCG.2002.1046626>. – DOI 10.1109/MCG.2002.1046626. – ISSN 0272–1716 (1.1, 2.1)
- [114] WELCH, G. ; BISHOP, G. : SCAAT: Incremental Tracking with Incomplete Information. In: WHITTED, T. (Hrsg.): *SIGGRAPH 97 Conference Proceedings*, 1997, S. 333–344 (2.3.3, 4.5)
- [115] YOU, S. ; NEUMANN, U. ; AZUMA, R. : Hybrid Inertial and Vision Tracking for Augmented Reality Registration. In: *Proceedings of IEEE Virtual Reality Conference*, 1999. – ISSN 1087–8270, S. 260 –267 (3.4)
- [116] ZHANG, L. ; WU, X. : On the Application of Cross Correlation Function to Subsample Discrete Time Delay Estimation. In: *Digital Signal Processing* 16 (2006), Nr. 6, S. 682–694 (7.2.2, 7.4.2)