



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# Bachelorthesis

Chris Adolf

Integration und Test einer  
Koordinatensystemregistrierung am Beispiel  
eines optischen Lokalisierungssystems

Chris Adolf

Integration und Test einer  
Koordinatensystemregistrierung am Beispiel eines  
optischen Lokalisierungssystems

Bachelorthesis eingereicht im Rahmen der Bachelorprüfung  
im Studiengang Informations- und Elektrotechnik  
am Department Informations- und Elektrotechnik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Annabella Rauscher-Scheibe  
Zweitgutachter : Dr. Ing. Holger Kunze

Abgegeben am 20.01.2015

**Chris Adolf**

**Thema der Bachelorthesis**

Integration und Test einer Koordinatensystemregistrierung am Beispiel eines optischen Lokalisierungssystems

**Stichworte**

Hand-Auge-Kalibrierung, Lokalisierungssystem, Koordinatensystem, Artis zeego, Optotrak Certus

**Kurzzusammenfassung**

Die vorliegende Arbeit beinhaltet neben aufgabenstellung-bezogenen Erklärungen von technischem Equipment, eine detaillierte und schrittweise Erläuterung der Arbeitsschritte zur Herbeiführung einer automatisierten Lösung einer Koordinatensystemregistrierung

**Chris Adolf**

**Title of the paper**

Integration and test of a coordinate system registry using the example of an optical localization system

**Keywords**

hand-eye calibration, localization system, coordinate system, Artis zeego, Optotrak Certus

**Abstract**

In addition to task-related explanations of technical equipment, the present thesis includes a detailed and step by step explanation of the work stages required to achieve an automated solution to a coordinate system registry

# Inhaltsverzeichnis

|   |            |
|---|------------|
| <b>Kurzzusammenfassung</b>                                    | <b>II</b>  |
| <b>Inhaltsverzeichnis</b>                                     | <b>III</b> |
| <b>Tabellenverzeichnis</b>                                    | <b>V</b>   |
| <b>Abbildungsverzeichnis</b>                                  | <b>VI</b>  |
| <b>Abkürzungsverzeichnis</b>                                  | <b>VII</b> |
| <b>1 Einleitung</b>   | <b>1</b>   |
| 1.1 Einführung . . . . .                                      | 1          |
| 1.2 Motivation . . . . .                                      | 3          |
| 1.3 Aufgabenstellung . . . . .                                | 3          |
| <b>2 Stand der Technik</b>                                    | <b>5</b>   |
| 2.1 Hand-Auge-Kalibrierung . . . . .                          | 5          |
| 2.2 Lokalisierungssysteme . . . . .                           | 7          |
| 2.2.1 Lokalisierung über Elektromagnetische Felder . . . . .  | 7          |
| 2.2.2 Markerbasierte Lokalisierung . . . . .                  | 9          |
| 2.2.3 Lokalisierung durch Tiefeninformationskameras . . . . . | 10         |
| 2.3 Zusammenfassung . . . . .                                 | 12         |
| <b>3 Material und Methoden</b>                                | <b>13</b>  |
| 3.1 Artis zeego . . . . .                                     | 13         |
| 3.2 NDI Optotrak Certus . . . . .                             | 16         |
| 3.3 Software . . . . .  | 20         |
| 3.3.1 Microsoft Visual Studio 2008 . . . . .                  | 20         |
| 3.3.2 IAE . . . . .   | 20         |
| <b>4 Konzept und Umsetzung</b>                                | <b>21</b>  |
| 4.1 Konzept . . . . .   | 21         |
| 4.2 Umsetzung . . . . .                                       | 24         |
| 4.2.1 Aufbau des Lokalisierungssystems . . . . .              | 24         |

---

|          |   |           |
|----------|---|-----------|
| 4.2.2    | Initialisieren . . . . .                                  | 25        |
| 4.2.3    | Markerkoordinaten auslesen . . . . .                      | 26        |
| 4.2.4    | Sichtbarkeits- und Positionsveränderungsprüfung . . . . . | 28        |
| 4.2.5    | Kommunikation beenden . . . . .                           | 30        |
| <b>5</b> | <b>Messaufbau und Testablauf</b>                          | <b>31</b> |
| <b>6</b> | <b>Auswertung</b>   | <b>35</b> |
| <b>7</b> | <b>Diskussion und Ausblick</b>                            | <b>39</b> |
| <b>8</b> | <b>Zusammenfassung</b>                                    | <b>40</b> |
|          | <b>Literaturverzeichnis</b>                               | <b>41</b> |
|          | <b>Versicherung über die Selbstständigkeit</b>            | <b>43</b> |

# Tabellenverzeichnis

|     |  |    |
|-----|--|----|
| 4.1 | Datentypen eines Paketes . . . . .   | 27 |
| 6.1 | Ergebnisse der Koordinatensystemregistrierungen . . . . .                    | 35 |
| 6.2 | Ergebnisse der Koordinatensystemregistrierungen - mit Hilfe von Testpersonen | 37 |

# Abbildungsverzeichnis

|      |  |    |
|------|--|----|
| 1.1  | Auswirkungen von Kontrastmittel bei gleichzeitiger Durchleuchtung . . . . .                  | 1  |
| 1.2  | Aktiver und passiver Marker . . . . .  | 2  |
| 2.1  | Veranschaulichung des Konzeptes einer Hand-Auge-Kalibrierung . . . . .                       | 6  |
| 2.2  | Bestandteile eines RFID-Lokalisierungssystems . . . . .                                      | 8  |
| 2.3  | Funktionsweise Backscatter . . . . .   | 9  |
| 2.4  | Aktiver und passiver Marker . . . . .  | 10 |
| 2.5  | Aufnahme einer Kinect ohne Plastikabdeckung . . . . .  | 11 |
| 2.6  | IR-Gitter einer Kinect . . . . .   | 11 |
| 3.1  | Rotatorische und translatorische Achsen am Artis zeego . . . . .                             | 14 |
| 3.2  | Basis-Koordinatensystem des Artis zeego . . . . .  | 14 |
| 3.3  | Optotrak Certus auf Stativ . . . . .   | 16 |
| 3.4  | Basis-Koordinatensystem Optotrak Certus . . . . .  | 17 |
| 3.5  | Grundsätzlicher System-Aufbau für Messung mit Optotrak Certus . . . . .                      | 18 |
| 4.1  | Innerer Aufbau der IAE . . . . .   | 22 |
| 4.2  | Konzept für Programmstruktur . . . . .   | 22 |
| 4.3  | Überblick über die verwendeten Methoden . . . . .  | 23 |
| 4.4  | Aufbau der Ethernet-Setup-Software von NDI . . . . .   | 25 |
| 4.5  | Client-Server-Model . . . . .  | 26 |
| 4.6  | Aufbau eines Allzweck-Datenpaketes . . . . .   | 27 |
| 4.7  | Struktur zum Speichern von dreidimensionalen Koordinaten . . . . .                           | 28 |
| 4.8  | Sichtbarkeitskonstante . . . . .   | 28 |
| 4.9  | Programmcode-Struktur . . . . .  | 29 |
| 4.10 | Berechnung der Abstandsänderung . . . . .  | 30 |
| 4.11 | Umwandlung der Markerkoordinatenstruktur in Vector3 . . . . .                                | 30 |
| 5.1  | Positionierung der Geräte beim Messaufbau . . . . .  | 31 |
| 5.2  | Position des Lokalisierungssystem-Koordinatensystem zum Weltkoordinaten-<br>system . . . . . | 32 |
| 5.3  | Anfrage der Markerkoordinaten über den MessageBus-Monitor . . . . .                          | 33 |
| 5.4  | Beispiel einer Trajektorie beim Artis zeego . . . . .  | 34 |

# Abkürzungsverzeichnis

|      |   |
|------|---|
| AX   | Angiography and Interventional X-Ray                      |
| AXCS | Angiography and Interventional X-Ray Communication System |
| BU   | Business Unit   |
| CAD  | computer-aided design (rechnerunterstütztes Konstruieren) |
| CT   | Computertomographie                                       |
| ETW  | Eye-to-World Methode (Art der Hand-Auge-Kalibrierung)     |
| HEC  | Hand-Auge-Kalibrierung ( <i>hand-eye-calibration</i> )    |
| IAE  | Interventional Application Engine                         |
| ILED | infrared light emitting diode                             |
| IR-  | infrarot-   |
| LKS  | Lokalisierungssystem-Koordinatensystem                    |
| NDI  | Northern Digital Inc.                                     |
| RFID | Radio-Frequency-Identification                            |
| SCU  | System Control Unit                                       |
| TCP  | Tool Center Point   |
| WKS  | Welt-Koordinatensystem                                    |

# 1 Einleitung

Die Einleitung soll dem Leser einen Überblick über die Aufgabenstellung und den Aufbau der vorliegenden Arbeit verschaffen (siehe Kapitel 1.3). Zunächst wird in Kapitel 1.1 darauf eingegangen, in welches Themengebiet die vorliegende Arbeit einzuordnen ist. Anschließend (vgl. Kapitel 1.2) wird erläutert aus welchem Problem die Aufgabenstellung entstand.

## 1.1 Einführung

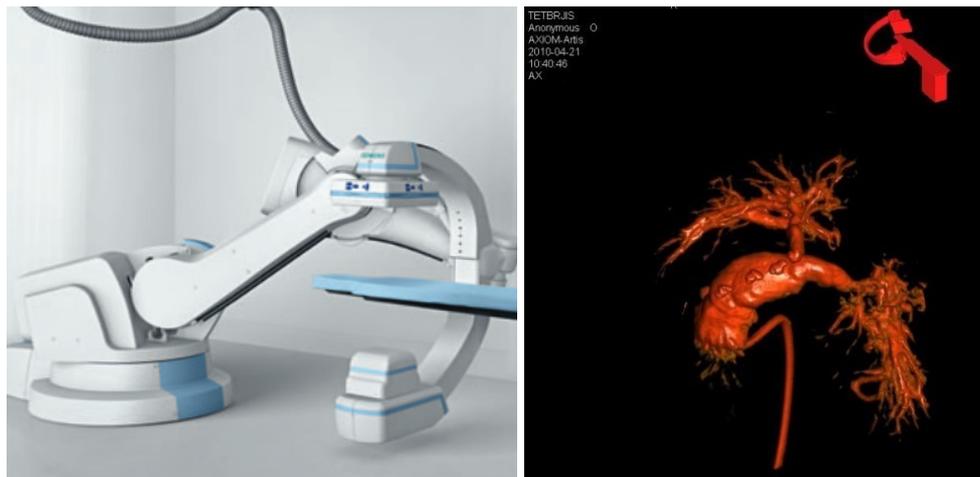
Angiographie ist die röntgenographische Darstellung von Blut- oder Lymphgefäßen nach Injektion eines Kontrastmittels [1, S.1].

Mittels intraarterieller oder intravenöser Zuführung des Kontrastmittels werden die Gefäße für das Angiographie-System sichtbar gemacht und können anschließend aufgenommen werden.



Abbildung 1.1: Die Abbildung verdeutlicht die Auswirkungen von Kontrastmittel im Körper des Menschen bei gleichzeitiger Durchleuchtung mit einem Röntgengerät. Dargestellt sind die rechte, mittlere Hirnarterie und Teile des Gehirns [2].

Um hochauflösende, klare und aussagekräftige Bilder aufnehmen zu können, wird die Forschung und Entwicklung der Röntgengeräte immer weiter intensiviert. Stand der Technik sind bereits dreidimensionale Aufnahmen von Weichteilgewebe mit Hilfe eines C-Bogen-Röntgengerätes. Dabei rotiert das System um den Patienten und nimmt mehrere hundert Bilder auf. Im Anschluss werden die Aufnahmen zu einem hochauflösenden 3D-Bild zusammengerechnet.



(a) Artis zeego

(b) 3D Rekonstruktion

Abbildung 1.2: (a) Zeigt das Artis zeego exemplarisch für ein Angiographie-System. Zu erkennen ist der C-Bogen, welcher um den Patienten herum rotieren kann. Aus den aufgenommenen Bildern ist eine 3D-Rekonstruktion möglich [3]. (b) Dargestellt ist die 3D-Rekonstruktion einer linken Lungenarterie aus mehreren hundert 2D-Bildern [4].

Die Business Unit (BU) Angiography and Interventional X-Ray (AX) der Siemens AG forscht u.a. auf dem Gebiet der medizinischen Bildgebung. Mit Hilfe moderner Robotertechnik und neuen Bildverarbeitungsverfahren soll dabei die Zahl der minimal-invasiven Behandlungen erhöht und die Kosten und Folgen von chirurgischen Eingriffen vermindert werden. Diese Ziele werden neben der Zeit- und Dosisersparung auch mit der Ermöglichung von Bildern aus verschiedenen Winkeln und der multimodaliären Fusion erreicht.

Die Anforderungen an das kontinuierlich zu modernisierende, technische Equipment sind besonders hoch. Hier wird das größte Potential für Innovation und Prozessoptimierung gesehen.

## 1.2 Motivation

In der modernen Medizintechnik steigt das Gesuch nach schnelleren und präziseren Eingriffen. Immer häufiger wird sich deshalb der Assistenz von technischem Equipment bedient. Dieses bietet neue Applikationen und ermöglicht damit effizientere Untersuchungen. Die Flexibilität der technischen Geräte und die Möglichkeit der freien Beweglichkeit im Raum birgt jedoch auch Gefahren für die anwesenden Personen und Objekte. Neben körperlichen Verletzungen können auch kostenintensive Beschädigungen an den Geräten entstehen. Aus diesen Gründen ist es notwendig, die Geräte zueinander bekannt zu machen um damit eine Kollisionsvermeidung zu unterstützen.

Für eine erfolgreiche Bestimmung der Position der eingesetzten Geräte müssen diese zueinander registriert werden. Mit Hilfe der Registrierung werden die durch ein Lokalisierungssystem bestimmten Geräte in ein gemeinsames Koordinatensystem abgebildet. Dadurch kann die Lage der Geräte zueinander eindeutig bestimmt werden.

Aus [5] ist eine Art der Hand-Auge-Kalibrierung (*hand-eye-calibration*, HEC) bekannt, die mittels aktiver optischer Marker eine Registrierung des Lokalisierungssystems zu einem medizinischen System ermöglicht. Die Registrieremethode erfordert eine manuelle Vorgehensweise mit mehreren Einzelschritten und setzt eine gute Kenntnis der mathematischen Grundlagen und des Ablaufs des Algorithmus voraus.

## 1.3 Aufgabenstellung

In der vorliegenden Arbeit wird eine HEC exemplarisch für ein optisches Lokalisierungssystem automatisiert und in ein medizinisches Gerät integriert. Hierfür wird ein Programm zur kontinuierlichen Datenaufnahme, Verarbeitung und Modulierung der von einem Lokalisierungssystem gesendeten Datenpakete erarbeitet und programmiert. Weiterhin wird eine Kommunikation über ein Message-System realisiert, um den kontinuierlichen Datenaustausch zwischen dem Lokalisierungssystem und dem medizinischen System zu gewährleisten.

Anhand eines prototypischen Systemaufbaus wird die Funktionalität der Arbeitsergebnisse untersucht und ausgewertet.

Die vorliegende Arbeit gliedert sich in acht Kapitel. In Kapitel 2 werden neben der verwendeten Hand-Auge-Kalibrieremethode verschiedene, mit der Aufgabenstellung kompatible, Lokalisierungssysteme und deren Funktionsweisen erläutert. In Kapitel 3 werden die Geräte beschrieben, die zur Lösung der Aufgabenstellung und für den prototypischen Systemaufbau ihre Anwendung finden. Kapitel 4 dient der Erläuterung der Erstellung eines Konzeptes und dessen Umsetzung. Bestandteil vom 5. Kapitel ist die Veranschaulichung eines prototypischen Systemaufbaus und die Erläuterung der Arbeitsschritte. Anschließend werden in

Kapitel 6 die Messergebnisse analysiert und ausgewertet. Kapitel 7 diskutiert die Messergebnisse und gibt einen kritischen Ausblick auf die Verbesserungsmöglichkeiten. Abschließend werden in Kapitel 8 die Ergebnisse der vorliegenden Arbeit zusammengefasst.

## 2 Stand der Technik

Im folgenden Kapitel wird der Stand der Technik für die zugrunde liegende Aufgabenstellung erläutert. Abschnitt 2.1 veranschaulicht die Registriermethode nach dem Hand-Auge Prinzip. Dabei wird auf die Funktionsweise und die Bedeutung für die Durchführung einer Registrierung eingegangen. Anschließend werden in Abschnitt 2.2 die verschiedenen Lokalisierungssysteme vorgestellt, welche aufgrund des generischen Aufbaus der Lösung der Aufgabenstellung ihren Einsatz bei einer automatischen Registrierung finden können.

### 2.1 Hand-Auge-Kalibrierung

In der Robotik stellt die Hand-Auge Kalibrierung eine verbreitete Vorgehensweise dar, um ein auf einem Roboter montiertes Lokalisierungssystem (z.B. eine Kamera) zu dem Roboter (z.B. Greifer) zu kalibrieren. Dabei wird das Lokalisierungssystem durch das Auge dargestellt und der Greifer repräsentiert die Hand. Soll die Hand nun ein vom Auge gesehenes Objekt greifen, muss die relative Lage zwischen Kamera und Greifer, durch die Hand-Auge-Kalibrierung, errechnet werden.

Um die Hand-Auge Transformation zu erhalten, müssen mindestens zwei bekannte relative Bewegungen des Roboters mit nicht parallelen Drehachsen ausgeführt werden [5, S.1]. Dadurch werden drei Posen mit verschiedenen Drehachsen angefahren. Durch diese Posen können die Transformations-Matrizen anschließend eindeutig bestimmt werden [6]. Je mehr Bewegungen für die Errechnung der Transformation ausgeführt werden, desto präziser wird die Abschätzung.

Dieser Sachverhalt ist auf die Kollisionsvermeidung zwischen technischen Einrichtungen übertragbar. Mit Hilfe der HEC wird das Koordinatensystem des Messinstrumentes (Lokalisierungssystem-Koordinatensystem, LKS) in das vom medizinischen Gerät genutzte Welt-Koordinatensystem (WKS) transformiert. Wird eine HEC erfolgreich durchgeführt, kann anschließend die Position der Marker des Lokalisierungssystems in das WKS übertragen und somit vom medizinischen Gerät genutzt werden. Ist die Position der Marker im Vorfeld nicht bekannt, müssen diese während der Bewegungen, die zur Errechnung der Transformationsmatrix nötig sind, starr am Objekt befestigt sein. Weiterhin sind mindestens drei Marker notwendig, um die Relativbewegungen des Roboters abbilden zu können. Die drei Marker

erlauben ein eindeutiges Erfassen der sechs Freiheitsgrade (Translationen  $x$ ,  $y$  und  $z$  sowie Rotationen roll, pitch und yaw [7]) bei jeder Bewegung des Roboters im Raum.

Es gibt zwei verschiedene Arten der Hand-Auge-Registrierung: Zum einen die Hand-Auge-Kalibrierung, bei der eine Kamera fest auf einen Roboter montiert wird und zum Anderen die Eye-to-World-Methode (ETW) bei der das LKS zum WKS bekannt gemacht werden soll. Letztere Methode findet Anwendung bei der Lösung der Aufgabenstellung. Abbildung 2.1 veranschaulicht das Konzept der ETW.

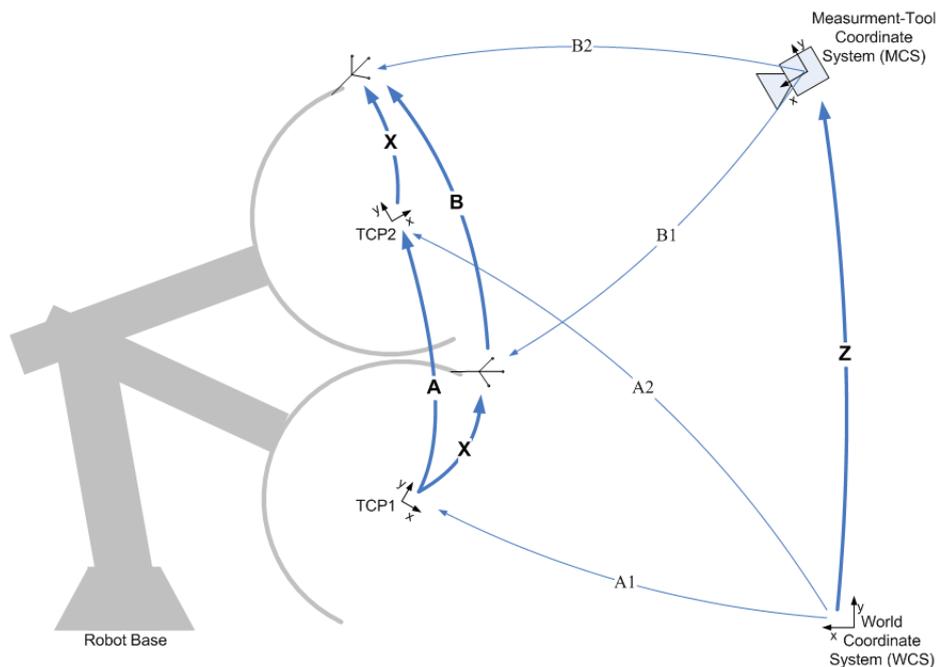


Abbildung 2.1: Veranschaulichung der HEC mit Hilfe eines optischen Lokalisierungssystems und einem Roboter, welcher zwei verschiedene Positionen anfährt. Die Orientierung und die Positionen des Tool Center Point (TCP) sind bezüglich des WKS bekannt (**A1** und **A2**). Weiterhin repräsentieren **B1** und **B2** die Koordinaten der optischen Marker, gemessen durch die Kamera. **A** steht für die relative Bewegung des TCP im WKS und **B** für die relative Bewegung der Marker im LKS. Neben der unbekannt Transformation zwischen WKS und LKS (**Z**) gibt es eine weitere unbekannt Transformation zwischen dem TCP und den Markern des optischen Lokalisierungssystems (**X**) [5].

Die ETW kann über die folgende homogene 4x4 Matrizen-Gleichung ausgedrückt werden:

$$\mathbf{AX} = \mathbf{ZB} \quad (2.1)$$

Jede Matrix aus der Gleichung 2.1 besteht aus einer 3x3 Rotations-Matrix und einem 3x1 Translations-Vektor [7, S.15]. Exemplarisch ist dies in Gleichung 2.2 für die Matrix **A** dargestellt.

$$\mathbf{A} = \begin{bmatrix} R_A & \vec{t}_A \\ 0 & 1 \end{bmatrix} \quad (2.2)$$

**Z** ist die unbekannte Transformationsmatrix zwischen dem LKS und dem WKS. Die Matrix **A** wird mit Hilfe der kinematischen Kette<sup>1</sup> des Roboters errechnet und Transformation **B** wird durch das Lokalisierungssystem erfasst.

In der vorliegenden Gleichung 2.1 sind **A** und **B** bekannt und **X** sowie **Z** unbekannt. Die Berechnung der unbekanntenen Transformationsmatrix **Z** wird mit Hilfe eines bereits vorhandenen Programmcodes realisiert. Inhalt der vorliegenden Arbeit ist es, die kontinuierliche Übermittlung der Markerkoordinaten (siehe Abbildung 2.1, Positionen **B1** und **B2**) an das Programm zu realisieren und den Registriervorgang teil zu automatisieren.

## 2.2 Lokalisierungssysteme

Ein Lokalisierungssystem ist die Gesamtheit von Elementen, die zur Lokalisierung eines Gegenstandes im Raum kooperieren [9]. In der Regel erfolgen die Angaben zur Position des Objektes in drei Dimensionen (3D) mit Bezug auf ein kartesisches Koordinatensystem. Durch Methoden, wie z.B. Ortung über elektromagnetische Felder oder einer optischen Lokalisierung, kann die Position eines sich im Raum befindlichen Objektes, bezogen auf das Lokalisierungssystem, bestimmt werden. Wird das Lokalisieren automatisiert, ist es möglich, Lokalisierungssysteme auch zum sogenannten Tracking<sup>2</sup> zu nutzen.

Im Folgenden werden ausgewählte Lokisierungstechnologien zur Raumüberwachung vorgestellt, die im Rahmen der HEC verwendet werden könnten.

### 2.2.1 Lokalisierung über Elektromagnetische Felder

Zur Identifikation und zum Auslesen von elektronischen Datenträgern, ohne diese direkt zu kontaktieren, kommen technische Verfahren zum Einsatz, die aus der Funk- und Radartechnik bekannt sind [10, S.6]. Radio-Frequency-Identification (RFID) ist eine Technologie, die Daten mit Hilfe von elektromagnetischen Feldern austauscht. Neben zahlreichen Vorteilen, wie z.B. Kopiersicherheit von Daten oder Unempfindlichkeit gegen Verschmutzung oder

---

<sup>1</sup>ist die Beschreibung der Translationen und Rotationen jedes Gelenkes, eines fest mit der Umgebung verbundenen Roboters, vom Basis-Koordinatensystem bis zum Endeffektor [8, S.18]

<sup>2</sup>ist das kontinuierliche Übermitteln und Verarbeiten der Position eines Objektes

Nässe, bietet das RFID-Verfahren auch die Möglichkeit der kontakt- und kabellosen Lokalisierung von Objekten.

Die Reichweite des Systems resultiert dabei aus der Betriebsfrequenz, auf der das Erfassungsgerät<sup>3</sup> sendet. Ein weiteres Unterscheidungskriterium zwischen RFID-Systemen ist die Datenmenge, die ein RFID-Transponder speichern kann. Diese reicht von einem Bit bis zu mehreren kBytes.

Ein schematischer Aufbau einer RFID-Einrichtung ist in Abbildung 2.2 dargestellt.

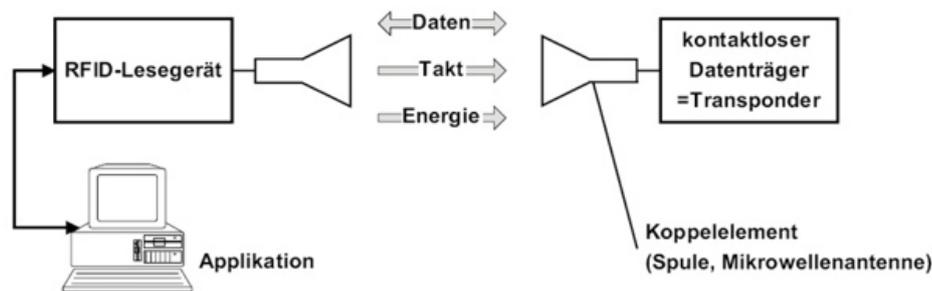


Abbildung 2.2: Ein Transponder und ein Erfassungsgerät sind die zwei Grundbestandteile eines RFID-Systems. Auf dem Transponder sind die objekt-spezifischen Daten gespeichert. Das Erfassungsgerät liefert die Energie und den Takt an den Transponder. Befindet sich der Transponder innerhalb des Sendebereichs des Erfassungsgerätes, wird dieser aktiviert und kann Informationen übertragen [10, S.8].

RFID-Transponder bestehen meist aus einem Mikrochip und einem Koppellement<sup>4</sup>. Handelt es sich um einen passiven Transponder, kommt dieser ohne Batterie aus. Die nötige Energie für seine Versorgung entnimmt der passive Transponder dem elektromagnetischen Feld des Erfassungsgerätes.

Für eine kontaktlose Lokalisierung wird zum größten Teil nach dem Prinzip der Backscatter-Kopplung gearbeitet. Es handelt sich um ein Ultra-Hoch-Frequenz (Ultra-High-Frequency, UHF) Verfahren, das angewendet wird wenn das Erfassungsgerät und der Transponder mehr als einen Meter voneinander entfernt sind. Durch die kurzen Wellenlängen sind die Abmessungen der Antennen kleiner und der Wirkungsgrad ist höher [10, S.56].

Der Transponder wird bei diesem Verfahren von der abgestrahlten Leistung des Erfassungsgerätes gespeist. Um die Reichweite des Systems zu erweitern, verfügen Backscatter-Transponder häufig über eine Stützbatterie zur Energieversorgung des Chips [10, S.57]. Der Vorgang des Backscatterns wird in Abbildung 2.3 erläutert.

<sup>3</sup>oft zum besseren Verständnis auch als *Lesegerät* bezeichnet (vgl. [10])

<sup>4</sup>Antenne, Spule, Leiterschleife

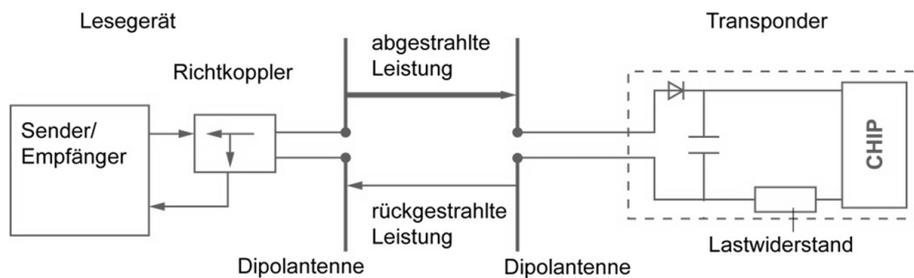


Abbildung 2.3: Die gesendete Leistung des Erfassungsgerätes wird nach Erreichen des Transponders durch Dioden gleichgerichtet und zur Aktivierung des Mikrochips verwendet. Je nach Betrag der empfangenen Leistung kann diese auch zur Energieversorgung für den Chip ausreichen. Ein Teil der empfangenen Leistung wird an der Antenne des Transponders reflektiert und zum Erfassungsgerät zurückgesendet. Die Eigenschaften der Antenne des Transponders können durch den Lastwiderstand beeinflusst werden. Dabei wird der Lastwiderstand im Takt des zu übertragenden Datenstroms ein- und ausgeschaltet wodurch sich die Amplitude des zurückgesendeten Signals modulieren lässt. Das reflektierte Signal läuft in „Rückwärtsrichtung“ beim Erfassungsgerät ein und kann durch den Richtkoppler separiert und verarbeitet werden (vgl. [10, S.58]).

RFID-Transponder können sehr klein und schmal gebaut werden und finden somit auf z.B. Etiketten wie auch in Chipkarten Platz.

## 2.2.2 Markerbasierte Lokalisierung

Die markerbasierte Lokalisierung zählt zu den optischen Lokalisierungsmethoden. Bei dieser Art der Lokalisierung werden Kameras zur Erfassung der Marker verwendet. Um eine Lokalisierung durchführen zu können, muss immer „Sichtkontakt“ zwischen dem zu ortenden Objekt und der Kamera bestehen.

Es gibt zwei Arten von Markern: aktiv oder passiv. Aktive Marker senden nach ihrer Ansteuerung ein Signal aus, auf das die Kamera getriggert ist. So finden z.B. Infrarot-LED-Marker ihre Anwendung. Passive Marker werden nicht extra angesteuert. Sie können z.B. mit einer fluoreszierenden Schicht überzogen sein und somit von einer Infrarot-Kamera gut erkannt werden.

Die messenden Kameras werden auf diese Art der Erfassung ausgerichtet. Somit kann der Rechenaufwand zur Lokalisierung des Objektes erheblich vermindert werden. Ein weiterer Vorteil ist, dass dadurch die Positionsbestimmung zusätzlich unempfindlich gegen irrelevante Fremdbeeinflussung wird.

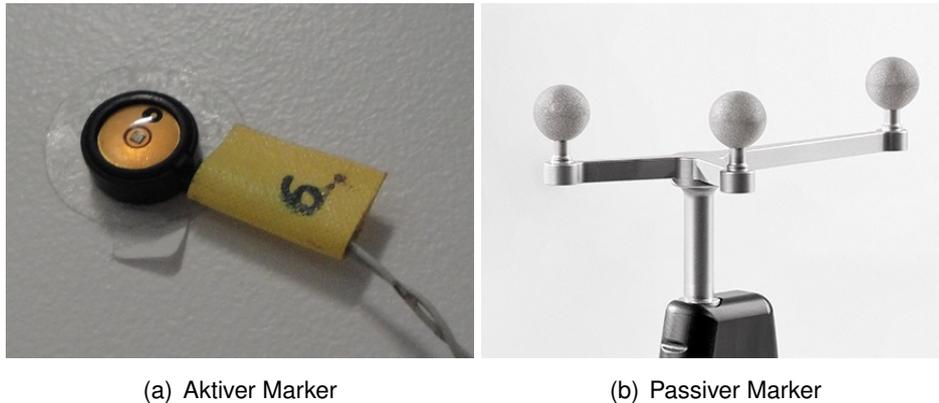


Abbildung 2.4: (a) Aktiver Marker der Firma Northern Digital Inc. Der Marker besteht aus einer Infrarot-LED, die auf einer Polymerbasis (Durchmesser 11mm) angebracht ist. Die Leitung (grau) ist zur Ansteuerung und Spannungsversorgung. (b) Passiver Marker der Firma Brainlab. Die Marker sind mit einer fluoreszierenden Schicht überzogen und nahezu kugelförmig [11].

### 2.2.3 Lokalisierung durch Tiefeninformationskameras

Mit einer einfachen Farbkamera können statische Bilder, die aus Pixeln mit Rot-, Grün- und Blauanteilen bestehen, aufgenommen werden. Es entsteht ein 2D-Farbbild, bei dem es nur begrenzt möglich ist, Tiefeninformationen des aufgenommenen Motivs zu erhalten. Aus Sicht der Lokalisierung hat das 2D-Bild somit kaum einen Nutzen.

Für eine dreidimensionale Lokalisierung werden zusätzliche Informationen benötigt. Diese Informationen können Tiefeninformationskameras liefern. Im Gegensatz zu 2D-Kameras können diese speziellen Kameras Entfernungsinformationen liefern. Um die Funktionsweise und den Nutzen einer Tiefeninformationskamera zu veranschaulichen, wird im Folgenden die „Kinect“ von Microsoft© als Beispiel vorgestellt.

Die Kinect ist eine Kamera, die neben der X- und Y-Koordinaten, auch Informationen über die Entfernung der aufgenommenen Objekte zur Kamera gibt [12]. In Abbildung 2.5 ist der innere Aufbau der Kinect zu sehen.

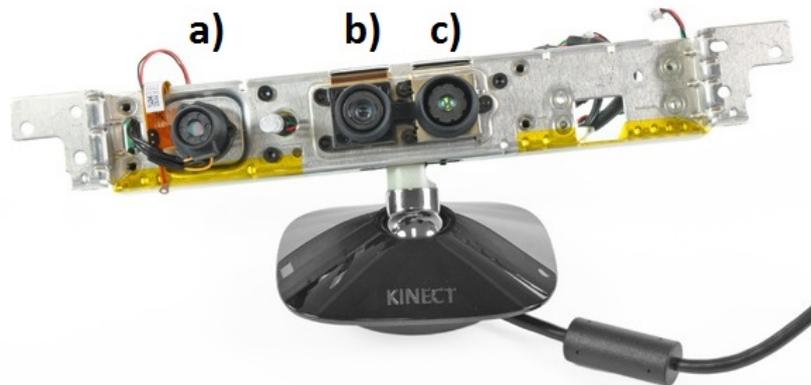


Abbildung 2.5: Die Kinect hat, neben mehreren Mikrofonen und einem Motor zur Höhenverstellung, einen Infrarot(IR)-Projektor (a) und zwei Kameras. Die eine Kamera ist eine Farbkamera (b), die farbige 2D-Bilder aufnimmt. Die andere Kamera ist eine IR-Kamera (c), die die vom IR-Projektor ausgesendeten Punkte, aufzeichnet. Die messbaren Entfernungen gehen dabei von 50,8 cm (20 inches) bis zu 762 cm (25 feet) [12, S.3].

IR-Licht hat eine höhere Wellenlänge als sichtbares Licht und ist vom menschlichen Auge nicht wahrnehmbar. Diese Eigenschaft und die Tatsache, dass es ungefährlich für Menschen ist, erklärt die Anwendung bei Tiefeninformationskameras.

Die Kinect arbeitet mit dem Entsenden eines Gitters aus IR-Licht-Punkten (ausgestrahlt durch den IR-Projektor), das von der IR-Kamera aufgenommen wird. In Abbildung 2.6 ist das IR-Gitter zu erkennen.

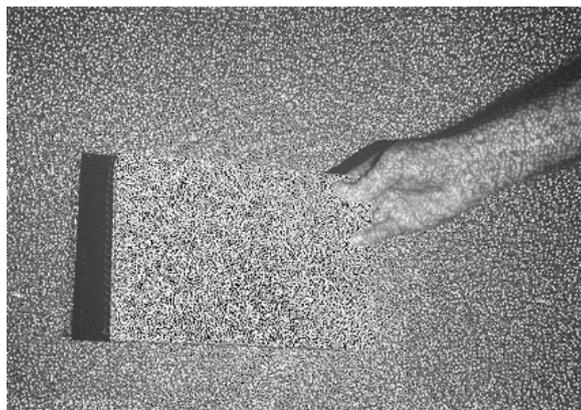


Abbildung 2.6: Wird ein Objekt in das projizierte IR-Gitter gehalten, verändert sich die Gitterstruktur je nach Entfernung des Objektes zur Kinect. Die IR-Kamera nimmt die veränderte Gitterstruktur auf und kann somit die Tiefeninformation ermitteln und in eine Distanz der Objekte zur Kamera umwandeln [12, S.4].

Bei einer werksseitigen Kalibrierung der Kinect werden die IR-Punkte gegen eine flache Wand in einer bekannten Entfernung projiziert. Dadurch ist die Anordnung der Punkte innerhalb der IR-Gitterstruktur für eine bestimmte Entfernung bekannt und kalibriert.

Die entstehenden Bilder unterscheiden sich grundlegend von denen einer Farbkamera. Bei den Tiefeninformationsbildern handelt es sich um schwarz-weiße Bilder bei denen die Helligkeit des Pixels für die Entfernung des Objektes zur Kamera steht. Dabei geht die Skala eines jeden Bildpixels von Null (schwarz, weit weg) bis zu 255 (weiß, dicht an der Kamera). Diese Information ermöglicht es „[...] zweidimensionale Graustufenpixel in dreidimensionale Punkte im Raum [...] [12, S. xii]“ umzurechnen. Dadurch wird eine Lokalisierung von Objekten mit Hilfe von Tiefeninformationskameras möglich.

## 2.3 Zusammenfassung

Für die Demonstration der Automatisierung und Integration einer Systemkoordinatenregistrierung wurde die verfügbare NDI Optotrak Certus als Lokalisierungssystem verwendet. Die dabei zur Anwendung gebrachte Technologie der Lokalisierung ist jedoch für die Realisierung der Integration nicht relevant, da grundsätzlich nur Koordinatendaten weiterverarbeitet werden. Da der Lösungsansatz generisch ausgerichtet ist, können somit alle Lokalisierungssysteme integriert werden, deren Daten Koordinatensystempunkte beinhalten.

## 3 Material und Methoden

Im Folgenden werden die verwendeten Geräte und die Software vorgestellt. Es wird deren Funktionsweise und Bedeutung für die Lösung der Aufgabenstellung herausgestellt. Neben dem Artis zeego (siehe Kapitel 3.1) wird die Optotrak Certus als Beispiel eines optischen Lokalisierungssystem vorgestellt (siehe Kapitel 3.2). Anschließend wird in Kapitel 3.3 die einbezogene Software erläutert.

### 3.1 Artis zeego

Das Artis zeego ist ein Angiographie-System. Es entstand unter dem Aspekt der Optimierung von Zeit, Kosten, Bildqualität und Flexibilität. Das Besondere an dem Artis-System ist die integrierte Industrierobotertechnik. Dabei besteht der Roboterarm aus sechs rotatorischen Gelenken. Ergänzt wird dies durch die Anbindung der Röhre und des Flachdetektors an den C-Bogen über jeweilige rotatorische Achsen sowie einer translatorischen Achse, zwischen diesen beiden bildgebenden Komponenten. Damit kann die Entfernung zum Patienten optimal eingestellt werden (siehe Abbildung 3.1).

Das Artis-System kann durch die sechs Achsen sehr flexibel um den Patienten herum positioniert werden. Dies bietet dem Arzt bei einer Durchleuchtung während einer Intervention mit dem Artis zeego einen besseren Zugang zum Patienten. Zusätzlich kann durch das Verfahren des Artis zeego in eine Ruheposition, ein größerer Freiraum für das behandelnde Personal geschaffen werden.

Durch die fortschrittliche Technik kann der Flachdetektor schnell und präzise um den Patienten rotieren. Das akkurate Abfahren der Trajektorien<sup>1</sup> ermöglicht dem Anwender die Anfertigung von CT-ähnlichen Bildern [13].

Die Kommunikation zwischen den Komponenten des Artis zeego findet über das sogenannte AXCS (Angiography and Interventional X-Ray Communication System) statt. Dieses telegrammbasierte System sorgt u.a. für den Austausch von Datenpaketen entsprechender Komponenten.

---

<sup>1</sup>ist die Kurve im n-dimensionalen Raum eines dynamischen (Roboter-)Systems in einem n-dimensionalen Koordinatensystem [8, S.106]



Abbildung 3.1: Zu sehen sind die verschiedenen Achsen des Artis zeego, die ein hohes Maß an Flexibilität ermöglichen [3].

Über das AXCS werden auch die Daten des Koordinatensystems des Roboters (CarmCS) geschickt. Dieses CarmCS ist zum Basis-Koordinatensystem (siehe Abbildung 3.2) registriert.

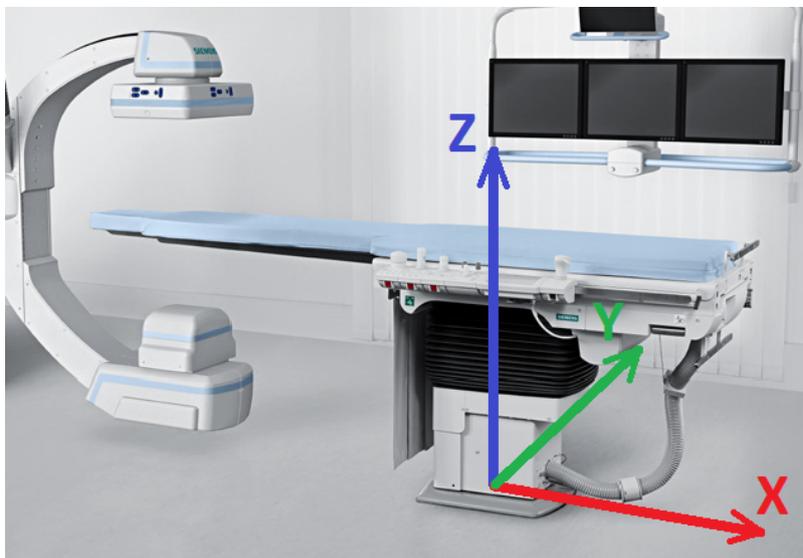


Abbildung 3.2: Das Basis-Koordinatensystem des Artis zeego liegt im Mittelpunkt der Grundplatte des Patiententisches. Das Koordinatensystem des Roboters, welches meist dem TCP (Tool Center Point) entspricht, ist zum Basis-Koordinatensystem registriert.

Somit sind die abgefahrenen Trajektorien bekannt und können in die, für eine Registrierung wichtigen, Transformationen umgerechnet werden (siehe Abbildung 2.1 auf Seite 6 die Transformationen **A**). Damit ist das Artis zeego ein wichtiger Bestandteil der HEC.

Eine Kollisionsvermeidung ist beim Artis zeego ausschließlich für den Roboter mit sich selbst, für den Roboter mit dem Tisch und für den Roboter mit den Raumgrenzen verfügbar. Um Geräte von Drittanbietern in die Kollisionsberechnung mit einbeziehen zu können, muss ein CAD-Model des Gerätes vorhanden sein. Weiterhin müssen die Achsendaten des Gerätes kontinuierlich ausgelesen werden, um die kinematische Kette und damit seine Orientierung im Raum (ausgehend vom gerätespezifischen Basis-Koordinatensystem) berechnen zu können.

Aufgrund der fehlenden Kollisionsberechnung für Geräte von Drittanbietern und aufgrund der vorliegenden Kenntnisse findet das Artis zeego in der vorliegenden Arbeit Anwendung.

## 3.2 NDI Optotrak Certus

Mit Bezug auf das Kapitel 8 wird im Folgenden die Optotrak Certus und deren Anwendung als optisches Lokalisierungssystem vorgestellt.

Die Optotrak Certus von Northern Digital Inc (NDI) ist ein hoch-präzises optisches Messgerät zur Lokalisierung von Gegenständen innerhalb ihres Messvolumens. Durch das kontinuierliche Erfassen der Positionen können auch Bewegungen gemessen werden. Dies ermöglicht dem Anwender zusätzlich das „Tracken“ von Objekten. Dabei hat die Optotrak Certus eine Genauigkeit von bis zu 0,1 mm bei einer maximalen Marker-Frequenz von 4600 Hz.

Das System arbeitet mit der Erfassung von emittiertem, infrarotem Licht. Das IR-Licht wird von Infrarot-LED's abgestrahlt. Damit liefert die Optotrak Certus dauerhafte und echtzeitnahe Daten über die Positionen der Infrarot-LED's.



Abbildung 3.3: In der Abbildung ist der Positionssensor die Optotrak Certus, als ein Teil des Gesamtmesssystems, auf einem Stativ zu sehen. Der Positionssensor besteht aus drei einzelnen Kameras. Die äußeren beiden Kameras sind leicht angewinkelt. Dadurch kann die Optotrak Certus einen größeren aber gleichzeitig auch lückenlosen Messbereich erfassen.

Das Lokalisierungssystem besteht aus den folgenden Komponenten:

1. Position Sensor
2. System Control Unit (SCU)
3. ILED (infrared light emitting diodes)
4. Marker strober
5. Cables (host cable, link cables, power cables, strober extension cables)
6. Data collection software
7. Host computer

Jede Kamera im Positionssensor steht für eine der drei Ebenen X, Y und Z. Die jeweilige Markerposition wird dann über den Schnittpunkt der Ebenen ermittelt. Dazu hat die Optotrak Certus ein werkseitig eingestelltes dreidimensionales Koordinatensystem (mit dem Ursprung Kamera), das dem Bestimmen der Position der Marker dient (siehe Abbildung 3.4).

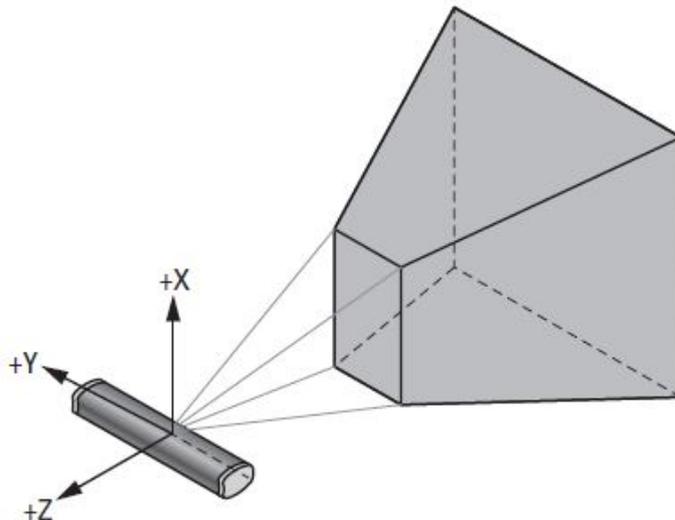


Abbildung 3.4: In der Abbildung ist die Lage des Koordinatensystems relativ zum Messvolumen zu erkennen. Der Ursprung der Ebenen befindet sich in der mittleren Kamera der Optotrak Certus. Weiterhin ist zu erkennen, dass Marker die zu dicht an der Kamera liegen oder solche, die zu weit entfernt sind, nicht erkannt werden können [14].

Um eine Messung mit der Optotrak Certus durchzuführen, muss vorerst einen Systemaufbau vorgenommen werden. Das Wichtigste dabei ist, die Optotrak Certus so aufzustellen, dass sich die Marker während des gesamten Nutzungszeitraumes im Messvolumen des Positionssensors befinden, d.h. dass die Marker sichtbar sind. Ein schematischer Systemaufbau ist in der folgenden Abbildung 3.5 dargestellt.

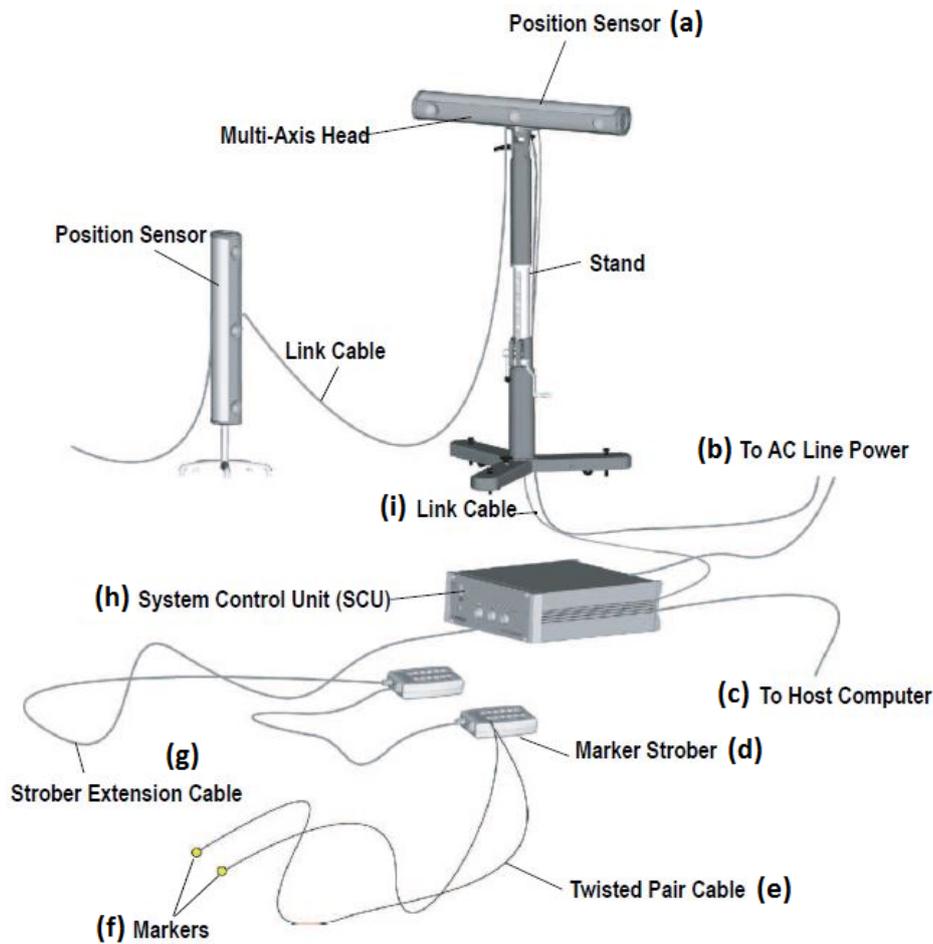


Abbildung 3.5: Zu sehen ist ein Messsystemaufbau zur Durchführung von Positionsbestimmungen mit der NDI Optotrak Certus. Das „Auge“ des Lokalisierungssystems ist der Positionssensor(a). Dieser detektiert das infrarote Licht der Marker innerhalb seines Messvolumens. Dabei ist es auch möglich, mehrere Positionssensoren miteinander zu verbinden um einen größeren Messebereich zu erhalten. Die Marker(f) sind aktiv ausgelegt, d.h. das sie erst nach Ansteuerung infrarotes Licht emittieren. Die zentrale Steuereinheit des Lokalisierungssystems stellt die SCU(h) dar. Sie bereitet die Messdaten auf und sendet sie an den Host-Computer. Weiterhin koordiniert die SCU den Einsatz und die Ansteuerung der Marker Strober und der Positionssensoren. Die Marker Strober(d) sind für das Aktivieren oder Deaktivieren der Marker zuständig. Ergänzend sind in der Abbildung verschiedene Kabel dargestellt. Diese dienen entweder der Verlängerung der Reichweite(g), der Spannungsversorgung der angeschlossenen Komponenten(b, e, g) oder der Kommunikation(c, i) (nach [14, S.1]).

Beim Systemaufbau kann die Verbindung zum Host-Computer über USB, RS 232 oder Ethernet erfolgen. Zum Erstellen der vorliegenden Arbeit wird aus Geschwindigkeitsgründen die Verbindung zur SCU über Ethernet aufgebaut. Am Host-Computer können die von der SCU empfangenen Markerpositionen angezeigt oder zur weiteren Verarbeitung verwendet werden.

Um nun eine Messung durchzuführen und sich die Daten auf dem Host-Computer anzeigen zu lassen, bietet NDI eine Anwender-Software namens „First Principles“ an. Diese eröffnet dem Benutzer die Möglichkeit auf das System zuzugreifen, verschiedene Parameter einzustellen und das System zu initialisieren.

Ein Teil der Aufgabenstellung der vorliegenden Arbeit ist das Umgehen der Installation der, von NDI, bereitgestellten Software „First Principles“. Damit wird die Lösung generischer gestaltet und der Vorbereitungsaufwand, zum Durchführen einer Systemkoordinatenregistrierung, minimiert sich.

Behilflich beim Erstellen eines Programmes zur direkten Kommunikation mit der SCU ist ein, von NDI geschriebenes, application programming interface (*kurz: API*). Es stellt Methoden zur Kommunikation mit dem Lokalisierungssystem bereit.

## 3.3 Software

Im Folgenden wird auf die verwendete Software eingegangen. Es wird dargelegt, warum sie Anwendung findet und was ihre Funktion beim Lösen der vorliegenden Aufgabenstellung darstellt.

### 3.3.1 Microsoft Visual Studio 2008

Visual Studio ist eine, von dem Unternehmen Microsoft© angebotene, integrierte Entwicklungsumgebung für verschiedene Hochsprachen[15]. Dieses Programm findet zum Erstellen der vorliegenden Arbeit Verwendung, da die von NDI erworbene Programmierschnittstelle OAPI (Optotrak Application Programmer's Interface), mit der auf die Optotrak Certus zugegriffen werden kann, mit der Programmiersprache C++ geschrieben ist. Auf die Anwendungsprogrammierschnittstelle wird im folgenden Kapitel 4 genauer eingegangen.

### 3.3.2 IAE

Die Interventional Application Engine (IAE) ist ein Teil des in Kapitel 3.1 beschriebenen AX-CS. Sie ist eine Sammlung von Prozessen, die über den Nachrichten-Bus (MessageBus) kommunizieren. Dazu gehören u.a. sämtliche Befehle zur Steuerung des Artis zeego sowie der Datenaustausch zwischen einzelnen Plug-in's (z.B. 3D Reconstruction, 3D Visualisierung). Um dem Anwender das Anzeigen der Kommunikation zu ermöglichen, hat die IAE einen integrierten MessageBusClient.

Die IAE ist generisch aufgebaut und damit sehr gut zum Einbinden neuer Softwarelösungen geeignet. Unter Beachtung der vorherrschenden Software Engineering Regeln, wie einer gewissen Struktur des Plug-in's, wird das Anbinden an den MessageBus damit schablonisiert. Auch das Plug-in der Hand-Auge-Kalibrierung sowie das Optotrak Plug-in kommunizieren über den MessageBus und sind somit ein Teil der IAE.

Die Anforderung eines kontinuierlichen Datenaustausches der CarmCS-Daten und der Markerdaten der Optotrak Certus für die HEC, befürworteten damit die Einbindung in die IAE.

# 4 Konzept und Umsetzung

Kapitel 4 befasst sich mit der Beschreibung der Vorgehensweise zur Lösungsfindung bezüglich der vorliegenden Aufgabenstellung. Dabei wird auf die Konzeptfindung und iterativ auch auf die Umsetzung der Entwürfe eingegangen. Weiterhin wird die Fusion der einzelnen Arbeitsergebnisse dargelegt.

## 4.1 Konzept

Innerhalb der IAE befinden sich die Programme „AppEngine“ und „LabDaemon“. „LabDaemon“ hat einen integrierten MessageBus-Client und unterstützt Plugins (Dynamic Link Library (DLL)-Dateien). Dadurch kann es für die Komponentenkommunikation verwendet werden. Bei der „AppEngine“ handelt es sich um ein Programm, das über einen integrierten MessageBus-Client mit „LabDaemon“ kommunizieren kann. Das Programm „AppEngine“ übernimmt den auswertenden Teil der Registrierung. Um mit Programmen aus der IAE kommunizieren zu können, gibt es das Programm „MessageBus.Monitor.exe“. Mit dieser Anwendung können Befehle auf den MessageBus innerhalb der IAE gesendet, aber auch gelesen werden. Für das Starten der automatischen Registrierung wird ein Befehl über den „MessageBus.Monitor“ an die „AppEngine“ gesendet. In Abbildung 4.1 ist der Aufbau schematisch dargestellt.

Zum Lösen der vorliegenden Aufgabenstellung ist das Ansprechen der Optotrak Certus und das anschließende Auslesen der Markerkoordinaten von Bedeutung. Dabei soll, wie am Schluss des Kapitels 3.2 bereits erläutert, mit dem Positionssensor ohne Installation zusätzlicher Software kommuniziert werden. Aufgrund der Integration der Markerkoordinaten in eine HEC ist zusätzlich ein Überprüfen der Markerkoordinaten auf ihre Nutzbarkeit von Relevanz.

Abbildung 4.2 zeigt das aus den Anforderungen entstandene Konzept des zu entwickelnden Programms.

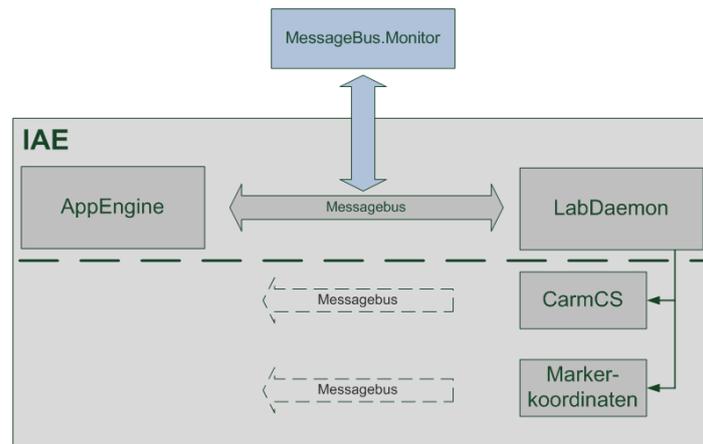


Abbildung 4.1: Um eine Registrierung zu starten, wird eine Nachricht an die „AppEngine“ geschickt. Diese sendet eine Anfrage an das Programm „LabDaemon“. Anschließend wird von „LabDaemon“ das Senden der Koordinaten vom CarmCS und der Marker an die „AppEngine“ veranlasst. Während der gesamten Registrierung ist es möglich, über den MessageBus die gesendeten Nachrichten zu lesen, die Registrierung abzubrechen, oder regulär zu beenden.

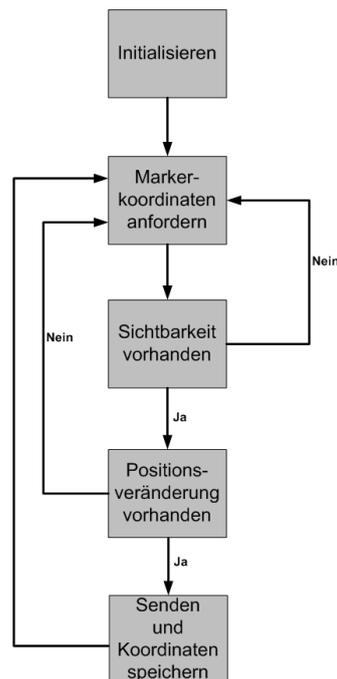


Abbildung 4.2: Das Konzept des erstellten Programmes verdeutlicht den gedachten Ablauf der Kommunikation. Dabei ist das Programm nach dem Initialisieren in einer while-Schleife, die eine kontinuierliche Markerkoordinaten-Abfrage durchführt.

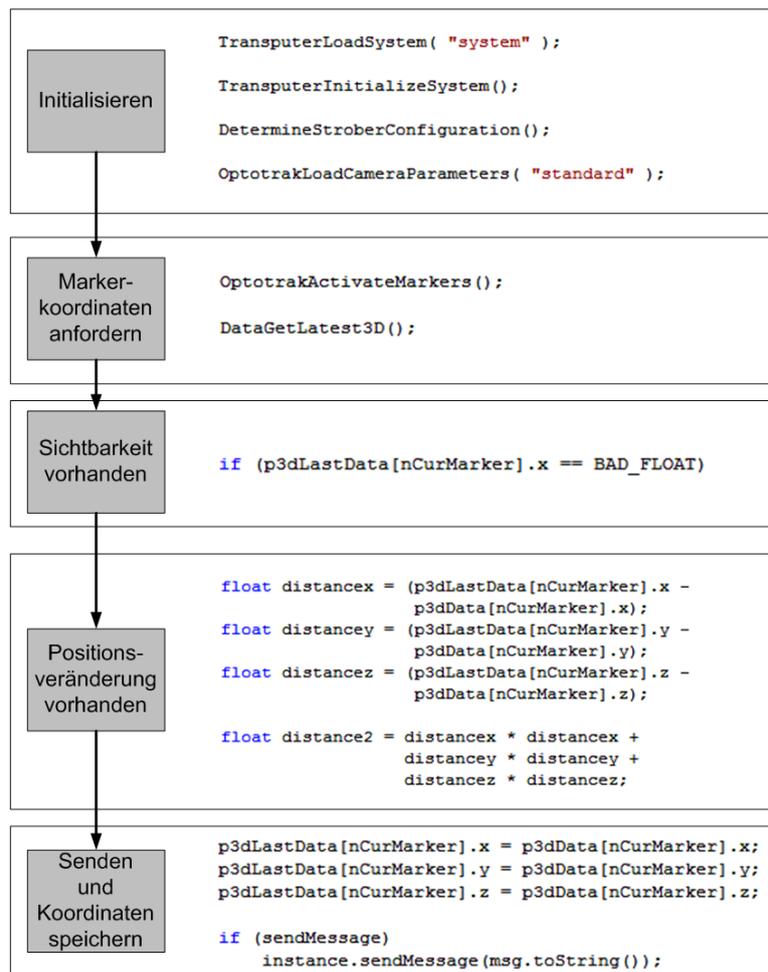


Abbildung 4.3: Die Abbildung gibt einen Überblick über die Methoden, die bei den einzelnen Schritten des Konzeptes ihre Anwendung finden.

Schritt eins zur Kommunikation mit dem Lokalisierungssystem ist das Initialisieren. Ohne eine erfolgreiche Initialisierung findet kein Datenaustausch zwischen dem Host-Computer und der SCU der Optotrak Certus statt.

Nach dem erfolgreichen Aufbau einer Kommunikation mit der SCU wartet das Programm auf eine Eingabe. Diese wird über den MessageBus an die „AppEngine“ geschickt. Ist eine Registrierung gestartet, werden die Markerkoordinaten aller an den Marker Strober angeschlossenen Marker an die „AppEngine“ gesendet. Da auch die Daten nicht-sichtbarer Marker gesendet werden, muss im nächsten Schritt die Sichtbarkeit der Marker geprüft werden. Wie im Kapitel 2.1 bereits erläutert, müssen mindestens drei Marker für eine HEC sichtbar sein. Ist das bei einer Messung nicht der Fall, werden erneut Markerkoordinaten angefordert. Ist die Sichtbarkeit von mindestens drei Markern gewährleistet, werden die Daten auf die Veränderung der Position der Marker untersucht. Haben sich die Koordinaten vom CarmCS und

der Marker zur vorherigen Messung verändert, werden nach einer Sekunde alle Koordinaten an die „AppEngine“ gesendet.

Die Schleife vom Messen bis hin zum Weiterleiten der Markerdaten wird solange ausgeführt, bis ein „Finish“-Befehl an die „AppEngine“ gesendet wird.

## 4.2 Umsetzung

In diesem Abschnitt wird die Struktur des entwickelten Programmcodes erklärt. Dabei werden aufeinander folgend die Schritte aus der Abbildung 4.2 erläutert.

### 4.2.1 Aufbau des Lokalisierungssystems

Der erste Schritt nach Erstellung eines Konzeptes ist der Aufbau des Lokalisierungssystems. Um erfolgreich mit der Optotrak Certus arbeiten zu können, muss ein Systemaufbau nach Abbildung 3.5 vorgenommen werden.

Neben dem Aufbau der Hardware ist es wichtig, dass die Komponenten miteinander kommunizieren können. Wie im vorherigen Kapitel beschrieben, gibt es nur eine Möglichkeit sich die Markerdaten der Optotrak Certus auf dem Host-Computer anzeigen zu lassen: über eine kabelgebundene Verbindung zur SCU. Da diese Verbindung über Ethernet aufgebaut wird, muss ein Ethernet-Netzwerk auf dem Host-Computer eingerichtet werden.

Üblicherweise muss für die Verbindung zwischen der SCU und dem Host-Computer ein Crossover-Kabel<sup>1</sup> verwendet werden. Da die SCU jedoch mit einem Computer verbunden ist, der Windows 7 installiert hat und dieser automatisch die Art des angeschlossenen Kabels erkennt, ist hier ein Patch-Kabel ausreichend.

Nach dem Verbinden der Netzwerk-Teilnehmer über die Hardware muss nun die Software so konfiguriert werden, dass eine Kommunikation ermöglicht wird. Dazu müssen vorausgehend alle von NDI zur Verfügung gestellten Treiber bzw. Updates installiert werden. Dies ist notwendig, da die verschiedenen Komponenten des Lokalisierungssystems nur bei gleichen Versionsnummern miteinander interagieren können.

Um den Ethernet-Anschluss der SCU nutzen zu können, wird diese entweder über eine IP-Adresse oder einen Host-Namen angesprochen [14, S.49]. Da es jedoch nicht möglich ist, die SCU ohne die richtige IP-Adresse anzusprechen, muss diese erst in Erfahrung gebracht werden. NDI bietet hier die Möglichkeit über ein von der NDI Support-Seite herunterladbares Programm (Abbildung 4.4), die SCU über eine serielle Schnittstelle anzusprechen und zu konfigurieren. Dadurch kann z.B. die IP-Adresse wunschgemäß parametrisiert werden.

---

<sup>1</sup>wird benötigt, um zwei gleiche Arten von netzwerkfähigen Geräten miteinander zu verbinden

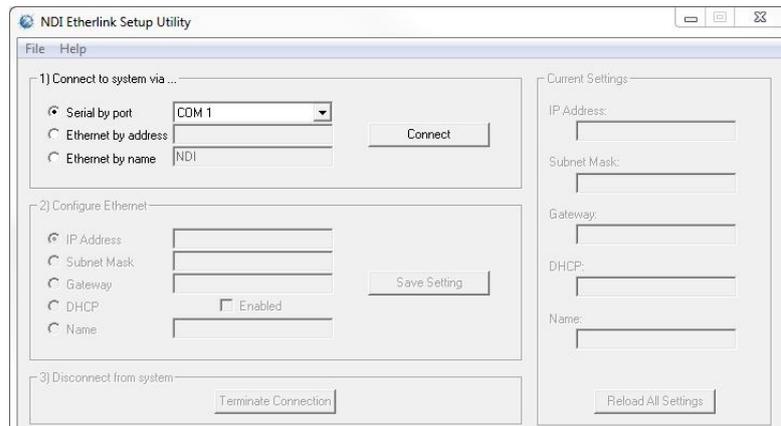


Abbildung 4.4: Das Programm „ethernetSetup“ ermöglicht dem Anwender, über eine serielle Verbindung auf die SCU zuzugreifen und die gewünschten Parameter zu konfigurieren. In der Abbildung sind, unter Punkt 2), die Ethernet Parameter justierbar.

Nachdem die IP-Adresse der SCU festgelegt und abgespeichert ist, kann eine Überprüfung der Verbindung zur SCU über das cmd-Window vom Host-Computer mit dem Befehl „ping 192.168.10.12“ vorgenommen werden. Damit wird die erfolgreiche Einrichtung eines Netzwerkes zwischen dem Host-Computer und der SCU überprüft.

## 4.2.2 Initialisieren

Das Initialisieren muss nach jeder Veränderung der Hardware oder Software vorgenommen werden, um mit dem Lokalisierungssystem kommunizieren zu können. Dafür ist es nötig, als ersten Schritt die Methode „TransputerDetermineSystemCfg()“ aufzurufen. Dadurch werden zwei Dateien generiert, in denen die aktuelle Systemkonfiguration gespeichert ist. Diese Dateien tragen die Namen „standard.cam“ und „system.nif“. Wird das System anschließend nicht mehr verändert, ist es nicht notwendig, die Methode „TransputerDetermineSystemCfg“ erneut aufzurufen.

Im entwickelten Programmcode steht zum Initialisieren die Funktion „OptoTrakLink::init()“ zur Verfügung. Innerhalb dieser Funktion wird zu Beginn einmalig „TransputerLoadSystem()“ ausgeführt. Diese Methode liest aus der Datei „system.nif“ (network information file) und lädt, entsprechend der Konfiguration in dieser Datei, die nötige Software zum Kommunizieren mit der SCU. Dies muss nur einmalig nach dem Anschalten der Hardware geschehen. Das tatsächliche Aufbauen und Trennen der kommunikationsfähigen Verbindung zur SCU findet ausschließlich über die Funktionen „TransputerInitializeSystem()“ und „TransputerShutdownSystem()“ statt [16, S.24, 101 ff].

Ein weiterer Bestandteil dieses Programmabschnittes ist die Methode „OptotrakLoadCameraParameters“, bei deren Aufruf die standardisierten Kameraparameter aus der Datei „standard.cam“ geladen und an das Optotrak System gesendet werden.

### 4.2.3 Markerkoordinaten auslesen

Die vom Positionssensor erfassten Markerdaten sind nahezu in Echtzeit auslesbar. Die Kommunikation zwischen dem Lokalisierungssystem und dem Host-Computer wird über ein Client-Server-Model beschrieben. Dabei fungiert das Optotrak-System als Server und das vom Anwender geschriebene Programm als Client [16, S.14]. Die Abbildung 4.5 verdeutlicht diesen Sachverhalt.

Das Lokalisierungssystem arbeitet ausschließlich mit dem Versenden von Nachrichten in Form von Paketen. Dabei ist jedem Paket eine Kennung angehängt, über die sich der Typ des Paketes bestimmen lässt [16, S.15].

Um mit jedem Host-Computer Nachrichten austauschen zu können, müssen der Server und der Client sich auf die gleiche Art des Packens und Entpackens der Pakete einstellen. Infolgedessen ist der richtige, innere Aufbau jedes Paketes von Bedeutung [17, S.3].

Um die Kommunikation so flexibel wie möglich zu halten und auch bei fehlerhaften oder leeren Nachrichten reagieren zu können, werden Allzweck-Datenpakete als Nachrichten versendet [17, S.4]. Im Header jedes Paketes befindet sich neben dem vier Byte großen Typen-Feld auch ein vier Byte großes Größen-Feld, welches die Größe des Paketes inklusive sich selbst angibt.

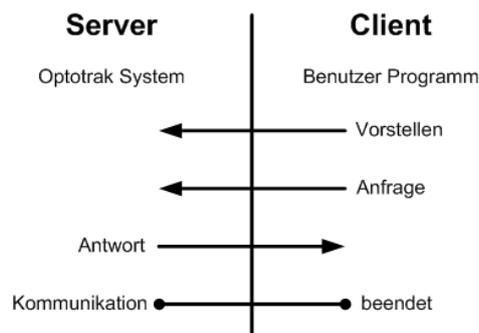


Abbildung 4.5: Einen Überblick über den Ablauf der Verständigung zwischen Server und Client gibt diese Abbildung. Die Verständigung wird mit dem Versenden und Empfangen von Paketen realisiert. Jedes Paket ist von einem bestimmten (Daten-)Typ, der je nach Inhalt des Paketes variiert.

Der innere Aufbau eines Datenpaketes ist in der folgenden Abbildung 4.6 dargestellt.

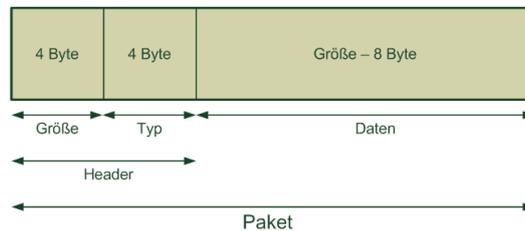


Abbildung 4.6: Aufbau eines Allzweck-Datenpaketes

Im Typen-Feld wird das Paket genauer klassifiziert. Das hat den Vorteil, dass der Empfänger weiß, wie er mit dem Paket weiter zu verfahren hat. Tritt z.B. ein Fehler auf, kann eine Rückmeldung an den Sender gegeben werden. Aber auch für den Anwender ist eine Rückmeldung, gerade beim Verifizieren des eigenen Programmes, von Bedeutung. Für die Kommunikation werden fünf verschiedene Paket-Typen verwendet. Diese sind in Tabelle 4.1 aufgeschlüsselt.

| Typ | Name        | Beschreibung   |
|-----|-------------|--|
| 0   | Fehler      | Ein Fehler wurde durch den letzten Befehl hervorgerufen.   |
| 1   | Befehl      | Eine Rückmeldung, ob der Befehl für oder vom Server erfolgreich war.   |
| 2   | XML         | XML Daten wurden vom, oder zum Server geschickt.   |
| 3   | Daten       | Ein Paket Echtzeit-Daten wurde vom Server geschickt.   |
| 4   | keine Daten | Dieses Paket besteht nur aus dem Header. Es lässt darauf schließen, dass eine Messung fertig ist oder noch nicht begonnen hat. |
| 5   | C3D Datei   | Eine komplette C3D Datei wurde vom Server verschickt.  |

Tabelle 4.1: Hier sind die möglichen Datentypen eines Paketes aufgezeigt [17, S.5].

Nach der Einführung in den Aufbau der Datenpakete wird im Folgenden dargelegt, wie die Markerkoordinaten vom Optotrak-System angefordert werden können.

Im entwickelten Programmcode ist die Funktion „OptoTrakLink::threadFunc(void\* data\_in)“ für das Abrufen der Markerkoordinaten zuständig. Aus dem Namen ist abzulesen, dass es ein extra Thread ist, der gestartet wird. Es gibt einen Thread zur Aufrechterhaltung der Kommunikation und einen zum Senden und Empfangen der Datenpakete.

Die Vorbereitung zum Empfangen der neuesten Markerkoordinaten beginnt mit dem Aktivieren der Marker über die Funktion „OptotrakActivateMarkers()“. Nachfolgend wird die Anzahl der zu messenden Marker festgelegt und zwei Strukturen vom Typ „Position3d“ mit der Größe der Markeranzahl angelegt. Die eine Struktur dient der Speicherung der aktuellen Koordinaten und die andere ist für das Speichern der Koordinaten für den nachfolgenden Messdurchlauf zuständig. Die entsprechenden Code-Zeilen sind in Abbildung 4.7 zu finden.

```

typedef struct _position3d
{
    float   x;           const int markerMax = 12;
    float   y;           Position3d p3dData[markerMax];
    float   z;           Position3d p3dLastData[markerMax];
} Position3d;

```

(a) Struktur definieren                      (b) Strukturen initialisieren

Abbildung 4.7: (a) In der Datei „ndtypes.h“ wird eine Struktur erstellt, in der es möglich ist, alle drei Dimensionen pro Marker zu hinterlegen. (b) Im Programmcode werden die zwei Strukturen vom Typ „Position3d“ mit der Größe der erwünschten Markeranzahl („markerMax“) initialisiert.

Im weiteren Verlauf des Programmcodes werden alle x-Koordinaten der Marker auf „nicht sichtbar“ initialisiert. Das sorgt dafür, dass nach jedem ersten Anfordern von Markerdaten immer eine Nachricht mit den aktuellen Markerkoordinaten gesendet wird. Nachfolgend startet eine while-Schleife, die solange läuft, bis ein Stop-Befehl gesendet wird. Im weiteren Verlauf des Programmes wird über die Funktion „DataGetLatest3D( &uFrameNumber, &uElements, &uFlags, p3dData )“ ein Datenpaket mit allen dreidimensionalen Markerkoordinaten abgefragt.

#### 4.2.4 Sichtbarkeits- und Positionsveränderungsprüfung

Bei optischen Lokalisierungssystemen ist eine Sichtverbindung zwischen dem Erfassungsgerät und den Markern die Grundvoraussetzung für das Lokalisieren. Aus diesem Grund wird im vorliegenden Abschnitt auf die Überprüfung der empfangenen Markerkoordinaten eingegangen.

Wenn ein Marker nicht vom Positionssensor erfasst werden kann, muss dieser Umstand klar definiert sein. Zur Kontrolle der Sichtbarkeit ist es ausreichend, nur eine Koordinate (z.B. x-Koordinate) des Markers zu prüfen. Denn ist eine der drei Koordinaten nicht sichtbar, ist der ganze Marker nicht lokalisierbar. Die Sichtbarkeit der Marker äußert sich in der Belegung der einzelnen Koordinaten mit einer, durch einen Makro definierten, Konstanten (vgl. Abbildung 4.8).

```
#define BAD_FLOAT (float)-3.697314E28
```

Abbildung 4.8: Abgebildet ist ein Ausschnitt aus dem Quellcode. Jedes weitere Vorkommen von „BAD\_FLOAT“ im Quelltext wird automatisch durch die float-Zahl  $-3,697314 \cdot 10^{28}$  ersetzt. Diese Konstante wurde gewählt, da sie nicht als Koordinate eines Markers gemessen werden kann.

Kann ein Marker nicht von der Optotrak erfasst werden, wird all seinen Koordinaten die float-Zahl  $-3,697314 \cdot 10^{28}$  zugeordnet.

In Abbildung 4.9 wird das Prinzip zur Überprüfung der von der SCU empfangenen, dreidimensionalen Markerkoordinaten schematisch dargestellt.

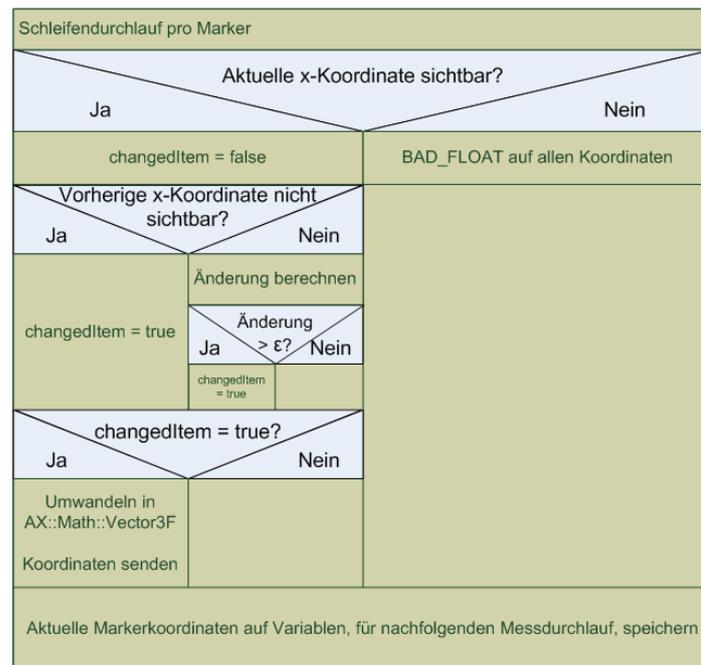


Abbildung 4.9: Dargestellt ist die Programmstruktur zur Prüfung der abgefragten Markerkoordinaten. Die Dreiecke entsprechen den Schleifenbedingungen im Programmcode. Darunterliegend sind die möglichen Antworten zu finden. Im Programm wird neben der Kontrolle der Koordinaten auch die Notwendigkeit des Übersendens eines neuen Datenpaketes, an die IAE, geprüft. Die Änderung, die vorhanden sein muss, bevor die Koordinaten für die HEC verwendet werden, ist entweder eine Abstandsänderung größer  $\varepsilon$  oder ein Wechsel der Sichtbarkeit des Markers. Mit Hilfe der booleschen Variablen „changedItem“ wird im weiteren Verlauf des Programms entschieden, ob die Markerkoordinaten in ein anderes Format für die HEC umgewandelt werden sollen.

Die Sichtbarkeits- und Positionsveränderungsprüfung grenzt an die Funktion „DataGetLa-test3D()“ an. Die Überprüfung der Koordinaten beginnt mit einer for-Schleife, die solange durchlaufen wird, wie es zu aktivierende Marker gibt. Die Marker werden demzufolge nacheinander überprüft.

Ist der Marker nicht sichtbar, wird dies für die nächste Abfrage abgespeichert und der nachfolgende Marker wird untersucht.

Entsprechen die x-, y- und z-Koordinaten nicht  $-3,697314 \cdot 10^{28}$ , ist der Marker sichtbar. Anschließend wird die Variable „changedItem“ angelegt und „false“ gesetzt. Mit Hilfe dieser Variablen wird im weiteren Verlauf des Programmes über das Senden entschieden. Auf das Anlegen der Variablen folgt eine weitere if-Schleife, bei der die vorangegangenen Markerkoordinaten auf Sichtbarkeit überprüft werden. War der Marker in der letzten Messung nicht sichtbar, haben sich seine Koordinaten geändert und es muss auf jeden Fall eine Nachricht mit den neuen Koordinaten gesendet werden. Ist der Marker in der vorherigen Messung jedoch sichtbar gewesen, muss vor dem Senden geprüft werden, ob sich seine Koordinaten ausreichend geändert haben. Das geschieht über die Formel aus Abbildung 4.10.

```
float distancex = (p3dLastData[nCurMarker].x - p3dData[nCurMarker].x);
float distancey = (p3dLastData[nCurMarker].y - p3dData[nCurMarker].y);
float distancexz = (p3dLastData[nCurMarker].z - p3dData[nCurMarker].z);

float distance2 = distancex * distancex + distancey * distancey + distancexz * distancexz;
```

Abbildung 4.10: Nach Berechnung der Änderung pro Koordinate wird der Abstand des Punktes vom vorherigen bestimmt. Das Ergebnis muss, zur Verwendung für die HEC, größer als  $\epsilon$  sein.

Ist der berechnete Abstand groß genug, wird „changedItem“ auf „true“ gesetzt. Im Anschluss an das Überprüfen des Abstandes folgt die Abfrage des Status der Variablen „changedItem“. Damit entscheidet sich ob die aktuellen Markerkoordinaten gesendet werden, oder nicht. Zum Senden müssen die Koordinaten in einen „Vector3“ umgewandelt werden (siehe Abbildung 4.11). Denn in diesem Format erwartet das Programm für die HEC die Daten.

```
AX::Math::Vector3F vector(p3dData[nCurMarker].x, p3dData[nCurMarker].y, p3dData[nCurMarker].z);
```

Abbildung 4.11: Vector3 ist eine Klasse, die der Beschreibung eines dreidimensionalen Punktes dient. Bei der Konstruierung wird der Vector3 mit (0, 0, 0) vorbelegt. Im Fall des vorliegenden Programmcodes wird der Vector3 bei seinem Konstruieren mit den jeweiligen Koordinaten initialisiert.

## 4.2.5 Kommunikation beenden

Um die Verbindung zum Lokalisierungssystem zu trennen, wird im Programmcode die Funktion „OptoTrakLink::onShutdown()“ aufgerufen. Dabei werden die aktiven Marker über das Ausführen von „OptotrakDeActivateMarkers()“ deaktiviert. Nachfolgend wird die Methode „TransputerShutdownSystem()“ bearbeitet. Dadurch wird das Optotrak System informiert, dass die API keine Kommunikation mehr benötigt.

## 5 Messaufbau und Testablauf

Um eine Koordinatensystemregistrierung durchführen zu können, muss vorerst die Hardware installiert und aufeinander abgestimmt werden. Dafür wird das optische Lokalisierungssystem nach Abbildung 3.5 aufgebaut. Im Falle der vorliegenden Arbeit wird nur ein Positionssensor verwendet. Dieser wird möglichst so positioniert, dass die Marker während des Abfahrens der Trajektorien kontinuierlich sichtbar sind.

Das Netzkabel der SCU wird, genauso wie das Netzkabel vom AXCS, in den selben Laptop gesteckt. Dies ist wichtig, da die Koordinatensystemregistrierung die Koordinaten beider Systeme für eine erfolgreiche HEC benötigt. Abbildung 5.1 zeigt die Positionierung der Geräte zueinander.



Abbildung 5.1: Im Vordergrund ist der C-Bogen des Artis zeego zu erkennen. Auf diesem sind drei Marker des Optotrak-Systems befestigt. Um eine kontinuierliche Sichtbarkeit der Marker bei gleichzeitigem großvolumigen Abfahren der Trajektorien zu gewährleisten, wird der Positionssensor (Hintergrund, Mitte) möglichst weit vom C-Bogen positioniert.

Mit der HEC wird das LKS zum WKS registriert. Das heißt, im Falle der vorliegenden Arbeit ist ein Ziel, die X-, Y- und Z-Koordinaten auszugeben, die die Entfernung des LKS vom WKS aufzeigen. Abbildung 5.2 verdeutlicht diesen Aspekt.

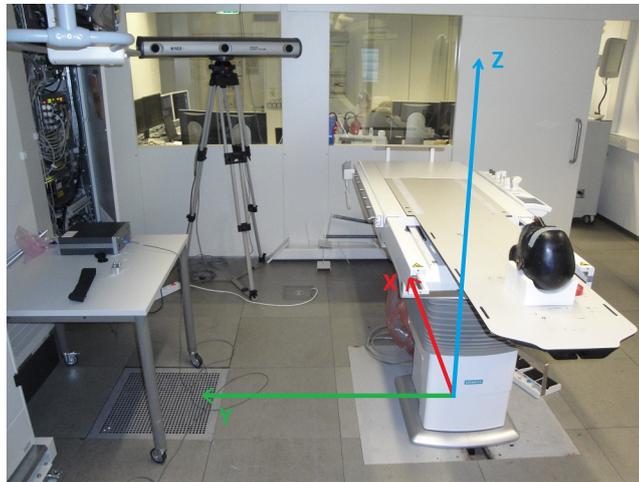


Abbildung 5.2: Im Mittelpunkt der Grundplatte des Tisches des Artis zeego befindet sich das WKS (eingezeichnet). Das LKS befindet sich in der Mitte des Positionssensors. Ziel des Messaufbaus ist die Berechnung der dreidimensionalen Koordinaten, die aufzeigen, wie weit das LKS vom WKS entfernt ist. Durch den Erhalt der Entfernungsdaten kann anschließend eine Transformation des LKS in das WKS erfolgen.

Aufgrund des Messaufbaus (vgl. Abbildung 5.2) werden ausschließlich positive Entfernungen erwartet. Um die Richtigkeit der X-, Y- und Z-Entfernungen abschätzen zu können, werden diese zuvor mit einem Maßband gemessen.

Ist die Hardware aufgebaut und mit dem PC verbunden, müssen die Netzwerke zum Lokalisierungssystem und zum AXCS konfiguriert werden. Anschließend kann die .bat-Datei „Start\_Registration.bat“ ausgeführt werden. Diese Datei startet die .bat-Dateien „runIAE\_LabDaemon“, „runIAE\_AppEngine“ und die Datei „runIAE\_Optotrak\_Adapter“. Letztere ist für die Initialisierung der Kommunikation mit der SCU und dem anschließenden kontinuierlichen Auslesen der Markerkoordinaten zuständig. Mit dem Ausführen dieser Datei wird das Konzept aus Kapitel 4 realisiert.

Auf das Ausführen der verschiedenen .bat-Dateien folgt das Starten der Datei „MessageBus.Monitor.exe“. Der MessageBus-Monitor zeigt zum Einen die Nachrichten, die zwischen den einzelnen Programmen geschickt werden (vgl. Abbildung 4.1), und zum Anderen können über ihn Befehle bezüglich der Registrierung gegeben werden.

Die Befehle sind in .txt-Dateien gespeichert, welche durch Auswahl über den MessageBus-Monitor, gesendet werden können. Mithilfe dieser Befehle werden verschiedene Programm-

Modi aktiviert bzw. deaktiviert. So kann z.B. mit dem Ausführen von „StartOptoTrakSending.txt“ das kontinuierliche Auslesen der Markerkoordinaten auf *aktiv* gesetzt werden. Abbildung 5.3 veranschaulicht diesen Vorgang.

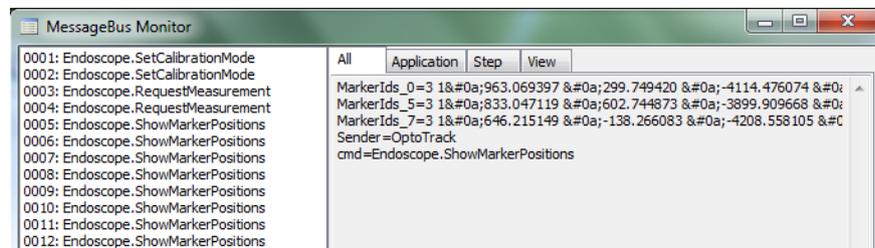


Abbildung 5.3: Dargestellt ist ein Ausschnitt des MessageBus-Monitors. Durch einen Rechtsklick in den linken Teil ist es möglich, Nachrichten auszuwählen. Erst durch einen anschließenden Doppelklick werden diese gesendet. Im vorliegenden Fall werden dauerhaft Markerkoordinaten von der SCU abgefragt und über den MessageBus gesendet. Dazu wurde der Befehl „StartOptoTrakSending.txt“ ausgewählt. Im rechten Teil werden die Inhalte der Nachrichten angezeigt. Zu erkennen ist, dass nur die Marker(-Ids) 0, 5 und 7 für den Positionssensor sichtbar sind.

Dieselben Befehle sind für das kontinuierliche Auslesen der AXCS-Daten vorhanden. Mit dem Befehl „StartRegistration.txt“ wird das kontinuierliche Auslesen der Markerkoordinaten (mindestens einmal pro Sekunde oder bei Veränderung der Koordinaten) und das parallele kontinuierliche Auslesen der AXCS-Koordinaten veranlasst. Das Speichern der Koordinaten im HEC-Plugin und der damit verbundenen Verwendung der Daten zur Berechnung, erfolgt ausschließlich nach einer Bewegung des C-Bogens und dem anschließenden Stillstand von einer Sekunde. Das Abfahren einer Trajektorie wird durch Abbildung 5.4 verdeutlicht.

Nach dem Abfahren der Trajektorie mit dem C-Bogen kann die Messdatenaufnahme beendet und die HEC gestartet werden. Dies geschieht über den Befehl „FinishRegistration.txt“, welcher an die „AppEngine“ gesendet wird. Anschließend kann das Ergebnis der HEC in einem Textdokument, welches als LOG-Datei gehandhabt wird, eingesehen werden. Neben den zur Berechnung herangezogenen Koordinaten der Marker und des AXCS, wird in der LOG-Datei eine 4x4 Matrix gespeichert, die u.a. einen translatorischen Anteil in X-, Y- und Z-Richtung aufzeigt. Dieser Anteil entspricht der Entfernung des LKS zum WKS.

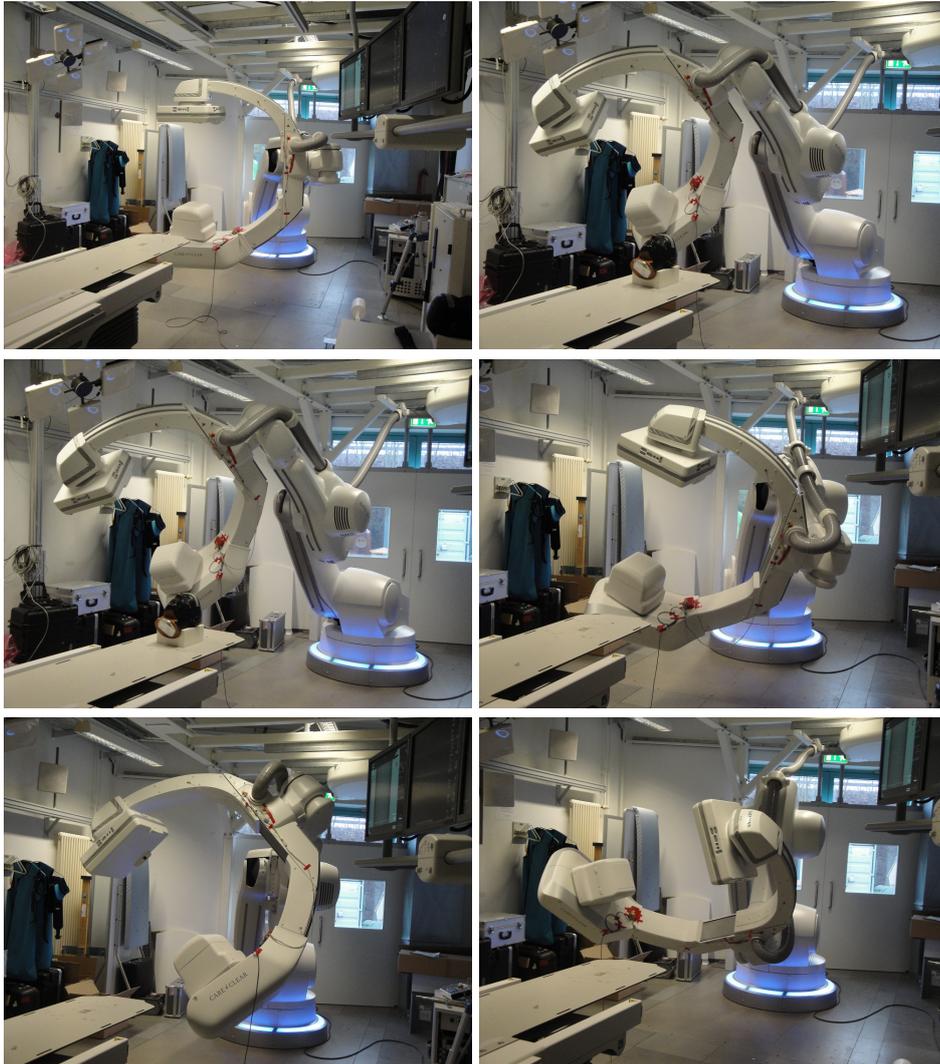


Abbildung 5.4: Die sechs Grafiken verdeutlichen exemplarisch das Abfahren einer Trajektorie im Rahmen einer Koordinatensystemregistrierung. Aufgenommen sind die Motive aus Sicht des optischen Lokalisierungssystems. Beim Anfahren der verschiedenen Positionen ist darauf zu achten, dass ausreichend viele Translationen und Rotationen vorgenommen werden. Das beinhaltet auch das Ausschöpfen des Bewegungsvolumens vom Artis zeego. Nach dem Stillstand des C-Bogens muss eine Sekunde gewartet werden, um die aktuelle Position für die HEC übernehmen zu können. Mit der Anzahl der angefahrenen und gespeicherten Positionen erhöht sich die Genauigkeit der Koordinatensystemregistrierung.

## 6 Auswertung

Nach dem Messaufbau werden mehrere Testdurchläufe absolviert. Damit wird die Genauigkeit und Fehleranfälligkeit der Koordinatensystemregistrierung überprüft. Die Genauigkeit liegt theoretisch bei 0,1 mm.

Die mit dem Maßband gemessenen X-, Y- und Z-Entfernungen lauten wie folgt:

**x [mm]:** 1610      **y [mm]:** 1290      **z [mm]:** 1735

Um ein aussagekräftiges Ergebnis zu erhalten, sind im Anschluss an die Maßband-Messung, 15 Koordinatensystemregistrierungen durchgeführt worden. Die folgende Tabelle 6.1 zeigt die translatorischen X-, Y- und Z-Werte als Ergebnis der HEC.

| Messung | x [mm] | y [mm] | z [mm] |
|---------|--------|--------|--------|
| 1       | 1560,3 | 1230,3 | 1755,2 |
| 2       | 1557,6 | 1264,6 | 1749,0 |
| 3       | 1559,3 | 1268,4 | 1746,1 |
| 4       | 1481,6 | 1211,3 | 1697,3 |
| 5       | 1452,3 | 1179,9 | 1675,2 |
| 6       | 1559,4 | 1273,8 | 1735,5 |
| 7       | 1461,2 | 1260,5 | 1739,7 |
| 8       | 1559,4 | 1266,9 | 1740,6 |
| 9       | 1557,9 | 1262,9 | 1744,5 |
| 10      | 1558,9 | 1271,7 | 1740,5 |
| 11      | 1596,4 | 1271,6 | 1703,2 |
| 12      | 1391,5 | 1154,8 | 1603,5 |
| 13      | 1557,7 | 1270,7 | 1742,1 |
| 14      | 1558,6 | 1265,7 | 1740,9 |
| 15      | 1554,7 | 1266,5 | 1735,6 |

Tabelle 6.1: Ergebnisse der Koordinatensystemregistrierungen

Aus den Messungen aus Tabelle 6.1 folgen die Mittelwerte mit:

$$\bar{x} \text{ [mm]: } 1531,1 \quad \bar{y} \text{ [mm]: } 1247,9 \quad \bar{z} \text{ [mm]: } 1723,3$$

Verglichen mit den ausgemessenen Werten ist eine Abweichung von maximal 78,9 mm (bei x-Translation) zu erkennen. Die Abweichungen liegen darin begründet, dass einerseits das Ausmessen per Hand sehr ungenau ist und somit die Werte keine präzise Angabe liefern sondern lediglich eine grobe Orientierung darstellen. Andererseits werden Messungen in die Mittelwert-Berechnung mit einbezogen, die weit vom Mittelwert abweichen. Also Messungen, bei denen offensichtlich keine ausreichend geeigneten Positionen mit dem C-Bogen angefahren wurden (vgl. Tabelle 6.1 die Messungen 4, 5 und 12).

Geeignete Positionen sind erkennbar, an den nicht parallelen Drehachsen beim Anfahren der Positionen mit dem C-Bogen. Wie im Abschnitt 2.1 erläutert, sind mindestens zwei verschiedene Stellungen des Artis zeego für eine HEC-Berechnung notwendig. Je mehr Positionen angefahren werden, und je weiter diese räumlich voneinander entfernt sind, desto präziser wird das Ergebnis.

Im Falle der vorliegenden Arbeit wurden mindestens 10 Positionen für jede Koordinatensystemregistrierung angefahren. Diese sind anschließend in einer LOG-Datei einsehbar. Dort ist die Kontrolle der, zur Berechnung verwendeten, Koordinaten von Bedeutung, um abschätzen zu können, ob geeignete Positionen angesteuert wurden.

Die Kontrolle der LOG-Datei ergab, dass bei den Messungen 4, 5 und 12 mehrere Positionen für die HEC verwendet wurden, die die gleichen Koordinaten aufwiesen. Aus diesem Grund werden diese verfälschten Messungen aus der Mittelwertberechnung ausgegliedert. Es ergeben sich folgende neue Werte:

$$\bar{x} \text{ [mm]: } 1553,5 \quad \bar{y} \text{ [mm]: } 1264,5 \quad \bar{z} \text{ [mm]: } 1739,4$$

Die Ergebnisse weisen eine deutlich kleinere Abweichung zu den Maßband-Werten auf, da die Koordinatensystemregistrierungen mit ungeeigneten Trajektorien die vorherige Mittelwertberechnung verfälscht haben. Durch die aktuellen Mittelwerte lässt sich, in Verbindung mit den Maßband-Werten, ein präzisere Aussage über die tatsächlichen Translationen treffen.

Die Überlegungen bezüglich der Fehlerquellen und die Anwenderfreundlichkeit der Koordinatensystemregistrierung, werden mit Hilfe von vier Testpersonen überprüft. Jeder dieser Testpersonen hat maximal zwei Versuche um eine Koordinatensystemregistrierung mit den erwarteten Ergebnissen zu erreichen.

Vorerst bekommen die Testpersonen eine kurze Einweisung in den Sachverhalt und werden angeleitet, welche Bewegungen sie mindestens in die selbst gewählte Trajektorie integrieren sollten.

Nach dem Abfahren der Trajektorie wird das Ergebnis der Koordinatensystemregistrierung begutachtet und auf eventuelle Fehler bei der Durchführung untersucht. Tabelle 6.2 zeigt die Ergebnisse der Testdurchläufe.

| Testperson | Versuch | Erfahrung     | Ergebnis [mm] |             |             |
|------------|---------|---------------|---------------|-------------|-------------|
| 1          | 1       | sehr gut      | x: 1557,6     | y: 1264,6   | z: 1749,0   |
| 2          | 1       | unerfahren    | x: 1319,6     | y: 1221,9   | z: 1699,4   |
| 2          | 2       | unerfahren    | x: 1565,4     | y: 1263,6   | z: 1741,9   |
| 3          | 1       | kaum erfahren | x: 1564,6     | y: 1266,6   | z: 1744,7   |
| 4          | 1       | unerfahren    | x: 1521,5     | y: 700,9    | z: 1795,9   |
| 4          | 2       | unerfahren    | x: -49410,2   | y: -56084,5 | z: -20991,7 |

Tabelle 6.2: Ergebnisse der Koordinatensystemregistrierungen - mit Hilfe von Testpersonen

Bei Testperson 1 ist die gute Erfahrung im Umgang mit der HEC von Vorteil. Diese Person weiß, auf welche Bewegungen sie bei der Trajektorie achten muss. Dadurch ergibt sich ein Ergebnis, das den erwarteten Werten entspricht.

Testperson 2 wiederum ist unerfahren hinsichtlich einer Koordinatensystemregistrierung. Die Sichtkontrolle beim Abfahren der Trajektorie ergab zu wenige Translationen und Höhenveränderungen. Weiterhin wirkten sich zu kleine Positionsveränderungen negativ auf die Genauigkeit der HEC aus. Nach einer weiteren, technischen Einweisung und Erklärung des Sachverhaltes, sind deutliche Verbesserungen eingetreten.

Durch gutes Verständnis für den Sachverhalt und aufgrund des Wissens über verwandte Themen konnte Testperson 3 eine gut geeignete Trajektorie abfahren. Eine Kontrolle der LOG-Datei ergab ausreichend viele und unterschiedliche Translationen und Rotationen. Somit wurden bereits beim ersten Versuch Werte erreicht, die den Erwarteten entsprachen.

Bei Testperson 4 sind anfangs nur translatorische Bewegungen mit dem Artis zeego vollzogen worden. Rotationen kamen erst später hinzu und wurden nicht mehr in die HEC mit einbezogen. Ein weiterer Fehler war das zu kurze Warten nach dem Stoppen des Artis zeego. Dadurch wurden viele Koordinaten nicht für die Koordinatensystemregistrierung verwendet. Dieser Fehler setzte sich auch im zweiten Versuch fort, wodurch weniger Werte gespeichert wurden als erwartet. Die Kontrolle der LOG-Datei ergab für den zweiten Versuch nur 6 angefahrte Positionen, wovon 5 Positionen ausschließlich Translationen beinhalteten und nur eine Rotation beim Anfahren der letzten Position ausgeführt wurde. Damit ist eine Koordinatensystemregistrierung nicht möglich. Das erklärt auch die relativ großen und negativen Werte als Ergebnis der HEC.

Zusammenfassend ist zu sagen, dass nach einer Einweisung auch unerfahrene Anwender in der Lage sind, eine Koordinatensystemregistrierung erfolgreich durchzuführen. Bei der Bedienfreundlichkeit und Intuitivität der automatisierten HEC besteht Verbesserungsbedarf. Die Funktionsfähigkeit wurde jedoch unter Testbedingungen belegt.

## 7 Diskussion und Ausblick

Trotz der erwiesenen Funktionsfähigkeit besteht Verbesserungsbedarf für das Automatisieren der Koordinatensystemregistrierung. So sind z.B. die unintuitiven HEC-Positionen des Artis zeego ein Hindernis für eine erfolgreiche HEC. Denn die Nutzer des Artis zeego fahren im Alltag nur selten Positionen an, die für eine HEC gut geeignet sind. Ergänzt wird dieser Umstand dadurch, dass die Anwender nicht immer die optimale Trajektorie für eine HEC finden. Selbst wenn sie bereits Erfahrung mit einer Koordinatensystemregistrierung haben, werden sie wiederholt eine andere Trajektorie als zuvor abfahren und damit eventuell nicht immer das optimale Verhältnis aus Translation und Rotation erreichen.

Diese Umstände empfehlen die Einführung einer automatischen Trajektorie. Das erhöht die Vergleichbarkeit der Messungen und der Ergebnisse. Denn bei unveränderter Hardware sollte auch nahezu dasselbe Ergebnis bei der Koordinatensystemregistrierung herauskommen. Des Weiteren wird durch die Einführung einer automatischen Trajektorie kontinuierlich eine sehr gute Trajektorie zum Registrieren gewährleistet.

Weiterer Verbesserungsbedarf beim Registrieren ist bei der Unwissenheit über die Anzahl und der Sachdienlichkeit der angefahrenen Positionen für eine erfolgreiche HEC zu finden. Da der Anwender nicht wissend über die Anzahl der bereits gespeicherten Koordinatendaten ist, besteht die Möglichkeit, dass zu wenig oder zu viele Stellungen mit dem Artis zeego angesteuert werden. Die Verbesserung dieses Umstandes ergibt eine Zeitoptimierung. Zusätzlich ermöglicht eine Ausgabe der zur Berechnung verwendeten Koordinatendaten, ein frühzeitiges Abschätzen über den Erfolg der aktuellen Koordinatensystemregistrierung und ermöglicht eine weitere Zeitoptimierung.

## 8 Zusammenfassung

Das Thema und Ziel der vorliegenden Arbeit ist das Teil-Automatisieren einer Koordinatensystemregistrierung mit Hilfe eines optischen Lokalisierungssystems. Durch eine Koordinatensystemregistrierung können Basis-Koordinatensysteme von einzelnen Objekten in ein gemeinsames WKS transformiert werden. Das hat den Vorteil, dass die Bewegungen der Objekte in Relation zueinander gesetzt werden und somit in eine globale Kollisionsvermeidung integriert werden können.

Die Medizintechnik ist, durch den steigenden Einsatz modernster Robotertechnik, ein gutes Beispiel für das Anwenden einer Koordinatensystemregistrierung. Als Exempel für die Verwendung von modernen Geräten in der Medizintechnik dient das in der vorliegenden Arbeit vorgestellte Artis zeego. Aufgrund der wachsenden Gefahr für anwesende Personen und Geräte, durch mögliche Zusammenstöße mit technischem Equipment, gewinnt der Aspekt der Kollisionsvermeidung immer mehr an Bedeutung.

Zum Lösen der vorliegenden Aufgabenstellung wurde die Optotrak Certus exemplarisch zum Einsatz gebracht. Neben diesem optischen Lokalisierungssystem sind, aufgrund des generischen Aufbaus der Lösung, weitere (vgl. Kapitel 2) Lokalisierungssysteme mit der Aufgabenstellung vereinbar.

Im Anschluss an die Darlegung des Konzeptes und der Beschreibung der Vorgehensweise (vgl. Kapitel 4) zum Lösen der vorliegenden Problemstellung, wurde die Funktionalität der Arbeitsergebnisse anhand eines Messaufbaus und Tests überprüft. Durch die Auswertung der Messdaten wurde die Funktionalität und Genauigkeit der Koordinatensystemregistrierung bestätigt. Das Testen der Automatisierung der Koordinatensystemregistrierung durch Dritte gibt jedoch Raum für Verbesserungen der Benutzerfreundlichkeit.

# Literaturverzeichnis

- [1] Goldyn. *Praxishandbuch Angiographie: Spektrum der Diagnostik und Interventionen*. Springer Verlag, 2014.
- [2] Siemens AG. figure 1. <http://www.healthcare.siemens.de/angio/clinical-results/acute-stroke-recanalization>, Zugriff 15.01.2014.
- [3] Siemens AG Healthcare sector. *Introducing Artis zee for cardiac procedures*.
- [4] Siemens AG. figure 2b. <http://www.healthcare.siemens.de/angio/clinical-results/tetralogy-of-fallot-supported-by-artis-zeego>, Zugriff 15.01.2014.
- [5] Danilo Briese, Christine Niebler, and Georg Rose. Hand-Eye Calibration using Dual Quaternions in Medical Environment. Technical report, Siemens Healthcare, Germany; Otto-von-Guericke-Universität Magdeburg, Germany.
- [6] Klaus H. Strobl and Gerd Hirzinger. Optimal Hand-Eye Calibration. Technical report, Institute of Robotics and Mechatronics, German Aerospace Center DLR.
- [7] REIMAR K. LENZ and ROGER Y. TSAI. Calibrating a Cartesian Robot with Eye-on-Hand Configuration Independent of Eye-to-Hand Relationship. Technical report, Technische Universität München, Watson Research Center.
- [8] Matthias Haun. *Handbuch Robotik - Programmieren und Einsatz intelligenter Roboter*. Springer Verlag, 2013.
- [9] Lehrstuhl fml an der TUM. Lokalisierungssystem. [http://www.fml.mw.tum.de/fml/index.php?Set\\_ID=945&letter=L&title=Lokalisierungssystem](http://www.fml.mw.tum.de/fml/index.php?Set_ID=945&letter=L&title=Lokalisierungssystem), Zugriff 01.12.2014.
- [10] Klaus Finkenzeller. *RFID HANDBUCH - GRUNDLAGEN UND PRAKTISCHE ANWENDUNGEN VON TRANSPONDERN, KONTAKTLOSEN CHIPKARTEN UND NFC*. Hanser Verlag, 2012.
- [11] Brainlab. Spinal Navigation Application. <http://www.brainlab.com>, Zugriff: 17.12.2014.

- 
- [12] Greg Borenstein. *Making Things See - 3D VISION WITH KINECT, PROCESSING, ADRUINO, AND MAKERBOT*. O'Reilly Verlag, 2012.
- [13] Siemens AG Pressereferat Healthcare. Artis zeego von Siemens bringt Flexibilität in die Angiographie. [http://www.siemens.com/press/de/pressemitteilungen/?press=/de/pressemitteilungen/2008/imaging\\_it/medax200801025.htm&content\[\]=HIM&content\[\]=HCIM](http://www.siemens.com/press/de/pressemitteilungen/?press=/de/pressemitteilungen/2008/imaging_it/medax200801025.htm&content[]=HIM&content[]=HCIM), Zugriff: 14.11.14.
- [14] Northern Digital Inc. *Optotrak Certus User Guide (e-Type)*.
- [15] Wikipedia. Microsoft Visual Studio. [http://de.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](http://de.wikipedia.org/wiki/Microsoft_Visual_Studio), Zugriff: 10.12.2014.
- [16] Northern Digital Inc. *Optotrak Application Programmer's Interface Guide*.
- [17] Northern Digital Inc. *RTC3D Protocol*.

# Versicherung über die Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §16(5) APSO-TI-BM ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen habe ich unter Angabe der Quellen kenntlich gemacht.

Hamburg, 20.01.2015

Ort, Datum

Unterschrift Chris Adolf

# Danksagung

Alle praktischen Arbeiten, die der Lösung der vorliegenden Aufgabenstellung dienen, wurden am Siemens AG Forschungs- und Entwicklungsstandort Forchheim durchgeführt. Stefan Schuster, Holger Kunze und Danilo Briese haben mit Ihrer Unterstützung vor Ort, einen maßgeblichen Teil zum positiven Ergebnis der vorliegenden Arbeit beigetragen. Diesen Personen gilt dafür besonderer Dank.

Ohne die Erlaubnis und die finanzielle Unterstützung wären die praktischen Arbeiten und die damit verbundenen, längeren Aufenthalte in Forchheim nicht möglich gewesen. Aus diesem Grund gilt ein weiterer Dank der Abteilung RC-DE HC NORD CS 1 der Siemens AG.

Ein wichtiges Kriterium zum Erstellen einer guten Bachelorarbeit ist die Betreuung an der zuständigen Hochschule. Für Ihre hilfreiche, kompetente und aufgeschlossene Begleitung beim Erstellen der vorliegenden Arbeit gilt ein besonderer Dank, Frau Annabella Rauscher-Scheibe.

# Anhang

Der Anhang der vorliegenden Arbeit befindet sich auf CD/DVD und ist beim betreuenden Prüfer oder dem Zweitgutachter einzusehen.