



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

# **Bachelorarbeit**

**Sascha Möhlmann**

## **Visuelle Positionsbestimmung auf mobilen Endgeräten**

*Fakultät Technik und Informatik  
Studiendepartment Informatik*

*Faculty of Engineering and Computer Science  
Department of Computer Science*

Sascha Möhlmann

**Visuelle Positionsbestimmung auf mobilen  
Endgeräten**

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Angewandte Informatik  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Birgit Wendholt  
Zweitgutachter: Prof. Dr. Gunter Klemke

Eingereicht am: 29. Juni 2012

**Sascha Möhlmann**

**Thema der Arbeit**

Visuelle Positionsbestimmung auf mobilen Endgeräten

**Stichworte**

Android, Smartphone, ARToolKit, NyArtoolkit, visuelle Positionsbestimmung, räumliches Referenzsystem

**Kurzzusammenfassung**

Dieses Dokument beschreibt die Entwicklung und die Evaluierung einer mobilen Anwendung zur visuellen Positionsbestimmung innerhalb von Gebäuden. Dabei wird mit der mobilen Plattform Android und einer Bibliothek zur Tagerkennung gezeigt, wie anhand von Richtungs- und Abstandsinformationen Positionen innerhalb eines räumlichen Referenzsystems sehr genau berechnet werden können.

**Sascha Möhlmann**

**Title of the paper**

Visual positioning on mobile devices in german

**Keywords**

Android, Smartphone, ARToolKit, NyArtoolkit, visual positioning, spatial reference system

**Abstract**

This document describes the development and evaluation of a mobile application for visual position determination inside buildings. It is shown with the Android mobile platform and a library for tag recognition, how a positions within a spatial reference system can be calculated very accurate using direction and distance information.

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
1.1	Motivation	1
1.2	Ziel dieser Arbeit	2
1.3	Gliederung	3
<b>2</b>	<b>Vergleichbare Arbeiten</b>	<b>5</b>
2.1	Global Navigation Satellite System (GNSS)	5
2.2	GSM	5
2.3	Infrarot	6
2.4	Funk	6
2.4.1	Radio-frequency identification (RFID)	7
2.4.2	WLAN	7
2.5	Ultraschall	8
2.6	Trägheitsnavigationssystem - Inertial Navigation System	9
2.7	Visuelle Positionsbestimmung	9
2.8	Bewertung	13
2.9	Zusammenfassung	15
<b>3</b>	<b>Analyse</b>	<b>17</b>
3.1	Systemidee	17
3.2	Zielgruppen	18
3.3	Funktionale Anforderungen	18
3.3.1	Anwendungsfälle aus der Konfigurationsphase	19
3.3.2	Anwendungsfälle aus der Betriebsphase	22
3.4	Nichtfunktionale Anforderungen	23
3.4.1	Zuverlässigkeit	23
3.4.2	Effizienz	23
3.4.3	Verfügbarkeit	23
3.4.4	Reaktionszeiten	24
3.4.5	Portierbarkeit	24
3.5	Zusammenfassung	24

<b>4</b>	<b>Design</b>	<b>26</b>
4.1	Bestehende Systemumgebung . . . . .	26
4.1.1	Indoor Spatial Information Service (ISIS) . . . . .	26
4.1.2	Blackboard Architektur im Living Place . . . . .	27
4.2	Wahl der Clientumgebung . . . . .	29
4.3	Wahl des visuellen Positionsbestimmungsverfahrens . . . . .	30
4.4	Designentscheidung . . . . .	32
4.4.1	Architekturmuster Pipes and Filters . . . . .	32
4.4.2	Observable Results . . . . .	33
4.4.3	MarkerBuilder . . . . .	33
4.5	Klassendiagramm . . . . .	34
4.6	Zusammenfassung . . . . .	39
<b>5</b>	<b>Evaluierung</b>	<b>40</b>
5.1	Kalibrierung . . . . .	40
5.2	Testbetrieb . . . . .	40
5.3	Ermittlung der tatsächlichen Abweichung . . . . .	41
5.3.1	Versuchsaufbau . . . . .	41
5.3.2	Auswertung der Messergebnisse . . . . .	42
5.4	Zusammenfassung . . . . .	45
<b>6</b>	<b>Schluss</b>	<b>46</b>
6.1	Zusammenfassung . . . . .	46
6.2	Ausblick . . . . .	46

# 1 Einführung

## 1.1 Motivation

Die Positionsbestimmung und die darauf aufbauenden Dienste, wie zum Beispiel die Navigation, haben in den letzten zehn Jahren im Outdoorbereich eine riesige Erfolgsgeschichte hingelegt. Digitale Karten, eine automatische Routenberechnung und die Anzeige der eigenen Position hat das Zurechtfinden vieler Autofahrer in fremden Umgebungen stark vereinfacht. In der Schifffahrt und im Flugbetrieb können Fahrzeuge mittlerweile den größten Teil ihrer Strecken alleine durch den Autopiloten gesteuert werden. Unternehmen mit vielen „Außendienstmitarbeitern“ wie Speditionen, Taxibetriebe oder solche mit vielen Service-Technikern<sup>1</sup> überwachen und steuern ihre Flotten anhand der genauen Positionsdaten der einzelnen Fahrzeuge, so dass die Disposition neuer Aufträge optimiert werden kann. Auch im Freizeitbereich entstand mit dem Geocaching eine neue Art der Schnitzeljagd.

Dieses ist allein durch die günstige Verfügbarkeit der Positionsdaten, welche das satellitengestützte „Global Positioning System“ kostenlos zur Verfügung stellt, entstanden. Im Indoorbereich existieren zurzeit nur Spezialanwendungen, zum Beispiel für die Lagerverwaltung. Allerdings gibt es noch keine vergleichbaren Lösungen für die Indoorlokalisierung, die die Erschließung des Massenmarktes für Consumer Produkte ermöglicht. Aufgrund der mittlerweile hohen Verfügbarkeit von leistungsfähigen Mobilgeräten wie Smartphones, die eine Fülle an Sensoren, wie zum Beispiel:

- Bluetooth
- Wlan

---

<sup>1</sup>z. B. ADAC, BMW

- GPRS
- NFC
- Bewegungssensoren
- Kamera
- Kompass

bieten und der Vielzahl an Anwendungsgebieten, ist aber mit einer Erschließung in diesem Bereich zu rechnen. So fangen zum Beispiel große Anbieter für „location based services“ im Outdoorbereich, wie Google, mit der Kartografierung des Innenraumes und dessen Nutzung an (McClendon (2011)).

Für die Positionsbestimmung stehen dabei viele voneinander sehr unterschiedliche Verfahren zur Verfügung, wie zum Beispiel Funk oder Ultraschallortung, die aber allesamt aufwendige Installationen in den Räumlichkeiten voraussetzen, spezielle Hardware benötigen oder aber nicht die gewünschte Genauigkeit bei der Positionierung erreichen. Die Bestimmung der eigenen Position anhand von Bildmaterial stellt hingegen einen vielversprechenden Ansatz dar. Zum Einen besitzen so gut wie alle Smartphones eine eingebaute Kamera und zum Anderen ist das Ausdrucken und Platzieren der Marker im Vergleich zu allen anderen Lösungen kostengünstig und einfach handzuhaben. Mit steigender Rechenleistung der mobilen Geräte und vor allem mit Verbesserungen bei den Grafikalgorithmen, wie „SIFT“ und „SURF“, könnte zukünftig sogar auf die Unterstützung solcher Marker komplett verzichtet werden.

## 1.2 Ziel dieser Arbeit

In dieser Arbeit soll eine mobile Anwendung auf Basis eines Android-Smatphones zur visuellen Positionsbestimmung innerhalb von Gebäuden entwickelt und dabei möglichst schon bestehende Bibliotheken verwendet werden. Des Weiteren soll diese Anwendung in eine bestehende Systemumgebung, dem „Living Place“ der Hochschule für Angewandte Wissenschaften Hamburg, integriert und die Genauigkeit anhand eines Prototypen evaluiert werden.

Dabei soll die visuelle Positionsbestimmung mittels eines markerbasierten Verfahrens zuverlässig auf dem mobilen Endgerät laufen und nur einen geringen zeitlichen Verzug bei der Berechnung bieten, um auch bei zeitkritischen Anwendungen die Genauigkeit der Positionsangabe nicht durch zeitlichen Versatz negativ zu beeinträchtigen.

Die verwendeten Marker sind innerhalb des im Living Place genutzten räumlichen Referenzsystems, welches durch ISIS bereitgestellt wird, zu verorten. Dabei ist eine Entkopplung zwischen der visuellen Darstellung der Marker und deren logischen Repräsentation zu gewährleisten um diese auch in anderen räumlichen Kontexten wieder verwenden zu können. Anhand von Entfernung und Orientierung zu einem erkannten 2D-Marker soll dann eine möglichst exakte Position innerhalb des räumlichen Referenzsystems der Umgebung berechnet werden. Um die Positionsinformationen allen anderen Softwarekomponenten im Living Place zur Verfügung zu stellen, soll eine Anbindung als „Publisher“ für das Living Place Blackboard erfolgen.

### 1.3 Gliederung

Das Kapitel 2 „Vergleichbare Arbeiten“ befasst sich mit einer Auswertung der bisher vorhandenen Produkte und Verfahren zur Positionsbestimmung. Dabei wird vor allem der Schwerpunkt auf die in Gebäuden nutzbaren und im Besonderen auf optische Verfahren gelegt.

Im anschließenden Kapitel 3 „Analyse“ werden die Systemidee und Zielgruppen definiert und die daraus resultierenden Anwendungsfälle aus der Konfigurations- und Betriebsphase, die das System abdecken soll, in Form von Use-Cases beschrieben.

In Kapitel 4 „Design“ wird zunächst die bestehende Systemlandschaft, in die sich das zu entwickelnde Produkt integrieren soll, vorgestellt. Aus den im vorherigen Kapitel definierten Anforderungen werden dann Entscheidungen wie die Wahl der Clientumgebung und die des zu verwendeten Positionierungsverfahrens für den Entwurf des Systems aufgeführt. Dabei wird gezeigt, wie mit der Wahl des Architekturmusters die einzelnen Verarbeitungsschritte der Positionsbestimmung

miteinander in Verbindung stehen und wie eine Parallelisierung dieser genutzt werden kann.

Das Kapitel 5 „Evaluierung“ befasst sich mit der Bewertung der Qualität der Positionsangaben die das erstellte System liefert. Dabei wird neben den Messungen unter Laborbedingungen auch auf den testweisen Einsatz unter realen Bedingungen im „Living Place“ eingegangen.

Zum Abschluss fasst das Kapitel 6 „Schluss“ die Ergebnisse dieser Arbeit kurz zusammen und gibt einen Ausblick über mögliche Erweiterungen und Verbesserungen des entstandenen Systems sowie ein abschließendes Fazit.

## 2 Vergleichbare Arbeiten

In diesem Kapitel werden verschiedene Kategorien teilweise exemplarisch an bestehenden Lösungen zur Lokalisierung vorgestellt und auf ihre Anwendbarkeit für den Indoorbereich untersucht, sowie deren Vor- und Nachteile dargestellt. Dabei wird im Besonderen auf die visuellen Verfahren zur Indoorpositionierung eingegangen.

### 2.1 Global Navigation Satellite System (GNSS)

Bei den satellitengestützten Positionierungssystemen werden von den mindestens 24 Satelliten die SatellitenID, die genaue Position sowie die Uhrzeit per Funk ausgestrahlt. Daraus berechnet ein Empfänger über die Ermittlung der Signallaufzeiten aus mindestens 4 Signalen per Multilateration seine aktuelle Position. Unter optimalen Bedingungen ist eine Positionsgenauigkeit von 4 Metern im *GPS* "Standard Positioning Service" (ziviler Dienst)<sup>1</sup> möglich. Die großen Vorteile dieser Systeme liegen vor allem in der globalen Verfügbarkeit, des günstigen Empfängers sowie der Selbstpositionierung. Aufgrund des gestörten Empfanges im Indoorbereich ist die dortige Verwendung nicht möglich und mit mindestens 4 Metern Abweichung auch zu ungenau.

### 2.2 GSM

Bei der Ortung von Mobiltelefonen werden je nach verbauter Hardware an den Funkmasten die Verfahren „cell of origin“, „time of arrival“ und „angle of arrival“ sowie deren Kombination verwendet. Dabei ist die Cell of origin das einzige, welches ohne Änderung der Hardware selbstständig von dem Telefon durchgeführt werden kann.

---

<sup>1</sup><http://www.gps.gov/technical/ps/2008-SPS-performance-standard.pdf>

Da die Ausdehnung einer GSM-Zelle in dünn besiedelten Gebieten bis zu 35 km beträgt, ist die Ortungsgenauigkeit dort entsprechend schlecht und nur als schnelle erste Näherung nutzbar. Wird die Ortung hingegen von mindestens 4 GSM-Stationen in Reichweite des Mobiltelefons per Multilateration (UL-ToA) durchgeführt, ist eine Genauigkeit von 50 bis 150 Metern erreichbar (Roth (2005) Seite 303). Allerdings ist die damit erzielte Genauigkeit für viele Anwendungen insbesondere im Indoorbereich einfach immer noch viel zu gering und ohne Unterstützung der Netzbetreiber nicht nutzbar.

### 2.3 Infrarot

Das *Active Badge System* dient dem Tracking von Personen oder Objekten innerhalb von Gebäuden. Dabei ist jeder Raum mit einem IR-Sensor bestückt, welcher die aufgezeichneten Signale an einen zentralen Location Server zur Auswertung weiterleitet. Um das Tracking zu ermöglichen, muss jede Person den "Active Badge", einen kleinen IR-Sender, sichtbar bei sich tragen.

*WIPS*<sup>2</sup> hingegen nutzt statische IR-Sender (IR-Bake) im Raum und einen "WIPS Badge" zum Positioning als Empfänger, der per WLAN Kontakt zum Location Server hält.

Da das IR-Signal den Raum nicht verlassen kann, ist die Genauigkeit der beiden Systeme hierbei auf die Zelle / den Raum beschränkt und ein fehlerhaftes Verrutschen der gefundenen Position in den Nachbarraum, ein "False Positive", unmöglich. Nachteilig ist vor allem die geringe Genauigkeit in größeren Räumen und die Verwendung spezieller Hardware, die jede Person mit sich führen muss, sowie die aufwändige Installation und Wartung des Sender- bzw. Sensornetzwerkes.

### 2.4 Funk

*SpotOn* (Hightower u. a. (2000)) nutzt den gleichen Aufbau wie das *Active Badge System*, nur verwendet es Funk für das Tracking. Da die Ausbreitung der Funkwellen

---

<sup>2</sup>WIPS

nicht durch die Raumwände aufgehalten wird, ist die Positionsberechnung anhand der Empfangsstärke deutlich komplexer. Die Auflösung von 3 Metern lässt zwar in größeren Räumen eine genauere Positionsbestimmung zu als bei infrarotbasierten Systemen. Zusätzlich ist das Auftreten von “False Positives” in der Raumgenauigkeit signifikant vorhanden.

### 2.4.1 Radio-frequency identification (RFID)

RFID-Tags besitzen keine eigene Stromquelle, jedoch Antenne, Speicher und einen Prozessor. Über die Funksignale werden sie mit Energie versorgt und angesprochen. Aufgrund der sehr geringen Reichweite von ca. einem Meter wird diese Technik vor allem in der Logistik eingesetzt wo die RFID-Tags an genau definierten Orten gelesen werden. Eine flächendeckende Positionsermittlung ist aufgrund der nötigen Anzahl an Lesegeräten hingegen nicht realisierbar.

### 2.4.2 WLAN

Die Positionsberechnung aufgrund der Sichtbarkeit COO<sup>3</sup> von WLAN-Accesspoints stellt außerhalb von Gebäuden in urbanen Gegenden eine gute Alternative zum GNSS<sup>4</sup>, da dort typischerweise ausreichend Access Points installiert sind und aufgrund der geringen WLAN-Zellgröße eine Genauigkeit von ca. 40 Metern erreicht werden kann.

Um innerhalb von Gebäuden eine ausreichende Genauigkeit zu erreichen ist es jedoch erforderlich die Empfangsstärken mehrerer Access Points mit einer zuvor erstellten/kalibrierten Karte abzugleichen. Leider sind die guten Ergebnisse von unter einem Meter Genauigkeit aus den Laborversuchen häufig nicht direkt übertragbar, da WLAN Access Points ihre Sendeleistung z.B. aufgrund der schwankenden Benutzeranzahl dynamisch anpassen oder die Empfangsstärke durch eine sich ändernde Umgebung wie z. B. Menschen oder Türen sich stetig ändert. Auf dem Markt befinden

---

<sup>3</sup>k-nearest neighbor

<sup>4</sup>Beispielhafte Verwendung: Android Smartphone oder Firefox Webbrowser

sich sowohl kommerzielle Systeme wie Eckahau<sup>5</sup> als auch wissenschaftliche Arbeiten wie z. B. MagicMap<sup>6</sup>.

Während eines Tests des letztgenannten Systems konnten auch mit zusätzlich speziell für diesen Anwendungszweck platzierten und konfigurierten Access-Points innerhalb des Versuchsaufbaus keine ausreichende Genauigkeit erzielt werden. So lag zum Beispiel die ermittelte Position teilweise mehrere Räume von dem realen Standort entfernt.

### 2.5 Ultraschall

*Activ Bat* verwendet ein Sensornetzwerk zur Erkennung von Ultraschallsignalen. Die Sensoren werden typischerweise in einem Raster von 1,2 Metern an der Decken befestigt. Zur Bestimmung der Position sendet der Location Server ein Funksignal an genau einen mobilen "Bat" der daraufhin ein kurzes Ultraschallsignal erzeugt. Jeder Deckensensor der das Signal dabei empfangen hat, leitet diese Nachricht wiederum an den Location Server, der mit Hilfe eines nicht linearen Gleichungssystems daraus die Positionierung bis auf 10 Zentimeter genau berechnen kann.

Das *Cricket* (Priyantha u. a. (2000)) System kehrt dieses Funktionsprinzip um. Die im Raum installierten Baken senden jeweils ihre Position per Funk sowie das Ultraschallsignal aus. Die Positionsberechnung kann nun komplett per Multilateration vom Mobilgerät erledigt werden. Um Raumgenauigkeit zu erreichen ist die Installation eines einzelnen Beacons pro Raum ausreichend. Um eine bessere Positionsbestimmung zu erhalten, werden mindestens zwei weitere Beacons benötigt, die dabei erreichbare Genauigkeit liegt dann bei ca. 30 cm.

Beide Systeme wären für die Indoorortung gut geeignet. Allerdings setzen sie spezielle Hardware voraus und benötigen zudem wiederum ein Sender- bzw. Sensornetzwerk, was Aufwand und Kosten vor allem für größere Installationen deutlich steigert.

---

<sup>5</sup><http://www.ekahau.com/>

<sup>6</sup><http://www.magicmap.de/>

## 2.6 Trägheitsnavigationssystem - Inertial Navigation System

Hierbei handelt es sich um Systeme die eine neue Position relativ zu einer alten bekannten Position berechnen. Dazu werden vor allem Gyroskop und Beschleunigungssensor, aber auch der Kompass genutzt um aus Geschwindigkeit und Orientierung die neue Position zu bestimmen.

Aufgrund von kleinen Messungenauigkeiten die sich mit der Zeit jedoch summieren, sind solche Systeme vor allem zur Bestimmung von Positionen innerhalb des Messintervalls eines anderen absolut positionierenden Systems oder zu dessen Verbesserung geeignet (Serra u. a. (2010)).

## 2.7 Visuelle Positionsbestimmung

Bei der visuellen Positionsbestimmung trägt der Benutzer eine Kamera, dabei wird durch Bildanalyse meist versucht, die aufgenommene Szene mit einer zuvor bekannten Referenz zu vergleichen. Da die Position dieser dann bekannt ist, ist auch automatisch die Position der Kamera näherungsweise bekannt. Zum Beispiel lässt sich durch ein aufgenommenes Bild der Mona Lisa auf einen bestimmten Raum im Louvre schließen. Ist die reale Lage bestimmter Bildpunkte bekannt, lässt sich weiterhin sogar noch die relative Position der Kamera zu dem aufgenommenen Bild berechnen und somit die Position im Raum noch genauer bestimmen.

Die Auswertung des Bildmaterials ist sehr rechenaufwändig, so erreicht das weiter unten aufgeführte SURF-Verfahren auf einem 3GHz schnellen 8 Core-Xeon-Prozessor lediglich ein bis zwei "frames per second" (Gossow u. a. (2011)). Deshalb nutzen viele Systeme visuelle 2D Marker, da diese mit deutlich geringerem Aufwand erkannt und ausgewertet werden können. Aufgrund der bekannten Größe dieser Marker lässt sich dabei auch Entfernung sowie Winkel zur Kamera berechnen.

*Sky-Trax* (SkyTrax) (siehe Abbildung 2.1) verwendet ein solches Verfahren mit an der Hallendecke angebrachten 2D-Markern um die Position von Gabelstaplern und

## 2 Vergleichbare Arbeiten

---

anderen Fahrzeugen zu überwachen und erreicht damit eine Positionsgenauigkeit von ca. 2 bis 30 cm.



Abbildung 2.1: Hallendach mit Sky-Trax-Markern

*ProCam*<sup>7</sup> ist ein Produkt der Firma AICON 3d.systems und dient zur Vermessung von Gegenständen. Der hauptsächliche Anwendungsfall dieses Systems ist das Feststellen von sichtbaren und unsichtbaren Verformungen bei Fahrzeugen an denen Crashtests durchgeführt wurden.

Dazu wird das Auto innerhalb eines Messlabors jeweils vor und nach einem Crashtest von einem Techniker mit einem Messtaster vermessen. Der Messtaster lässt sich am ehesten mit einem überdimensionierten Stift vergleichen, an dessen oberem Ende eine Kamera integriert ist. Diese nimmt während des Messvorganges die an den Wänden und Decken flächendeckend befindlichen 2D-Marker auf und versendet das Bildmaterial zur weiteren Auswertung an einen Computer. Die Messgenauigkeit dieses Systemes ist mit 0,1 mm angegeben.

Dieses System zeigt zwar wie Präzise eine visuelle Ortung unter besten Voraussetzungen sein kann, lässt sich aber Aufgrund seines Anwendungsfalles nicht allgemein einsetzen.

---

<sup>7</sup><http://www.aicon3d.de/produkte/moveinspect-technology/procam/>

Am „Institut Teknologi Bandung“ (Reza u. a. (2011)) wurde ein 2D-Lokalisierungs-System für mobile Roboter entwickelt, welches den von einer Funkkamera übermittelten Videostrom auf einem Laptop mittels ARToolKit auswertet. Werden dabei mehrere Landmarken erkannt, wird die Positionsbestimmung von einer Fuzzy Inferenzmaschine unterstützt. Die dabei erreichbare Genauigkeit wird bei einem 12x12 cm Marker mit -6,8 bis 2,4% Abweichung innerhalb des 2 qm großen Messfeldes angegeben und beträgt somit ca. 13 cm innerhalb dieser Grenzen.

Das „Institute for Computer Graphics and Vision“ der TU Graz entwickelte einen ortsbasierten Konferenzführer *SignPost* (Mulloni u. a. (2009)). Dieser arbeitet auf Windows CE-Geräten und verwendet das Studierstube ES AR-Framework, welches sich auch auf die Auswertung von 2D-Markern stützt. In einer größeren Installation während der Microsoft Mobile and Embedded Developers Conference 2007 konnte das System seine Tauglichkeit und Akzeptanz unter Beweis stellen. Das Studierstube ES Framework ist nicht verfügbar und die Anwendung nicht mit aktueller Hardware kompatibel.

An der „University of Tokyo“ (Saito u. a. (2007)) wurde ein Marker basiertes Indoorpositionierungssystem entwickelt, welches auf den Einsatz klassischer 2D-Marker verzichtet und stattdessen nahtlos integrierte Muster auf Wänden und Böden verwendet. Jeder einzelne Marker wird dabei von mindestens vier Bildelementen repräsentiert, die wie in Abbildung 2.2 erkennbar über die Auswertung des Schwerpunktes sowie die Berechnung der Rotation der einzelnen Bildelemente zu einer Grundlinie eine eindeutige Identifikationsnummer ergeben. Die verwendete USB-Kamera und der moderne Laptop mit einem 2.16 GHz DualCore-Prozessor konnte zu jedem Zeitpunkt eine Verarbeitungsgeschwindigkeit von über 20 frames per second erreichen, welches somit auch gut für Navigationsaufgaben oder ähnliche zeitkritische Anwendungen verwendbar ist.

Der erhöhte Rechenaufwand, die Schwierigkeit eine Mustertapete, mit den entsprechend steigenden Kosten in allen Räumlichkeiten an Wänden oder Fußböden zu verwenden, lässt einen breiten Einsatz eher unwahrscheinlich erscheinen.

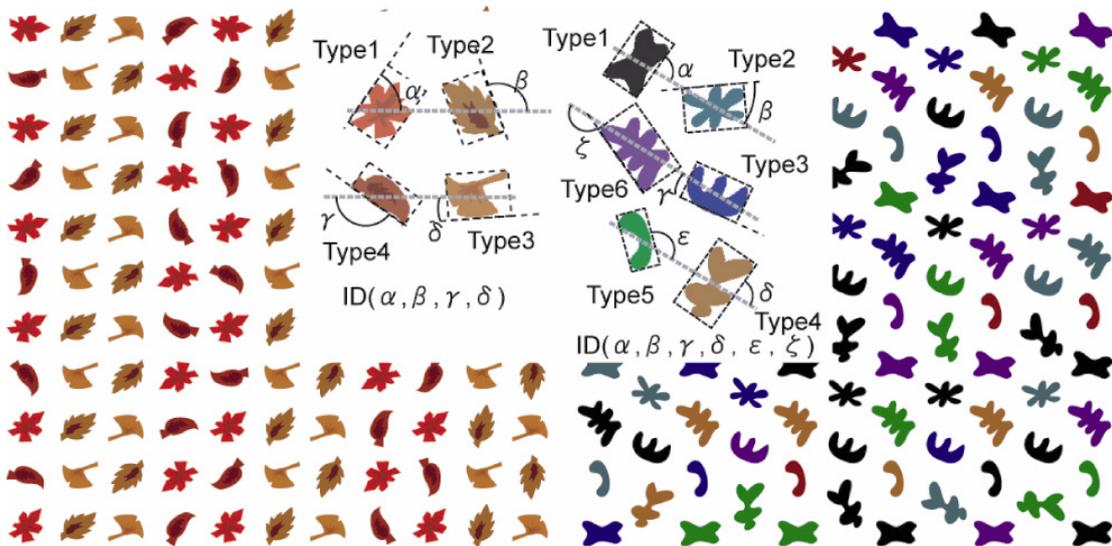


Abbildung 2.2: Beispiel Pattern der Universität Tokyo

Eine weitere Möglichkeit für die visuelle Positionsbestimmung ist es Bildmaterial mit denen aus einer Datenbank zu vergleichen und so auf die aktuelle Position zu schließen. Dazu werden z. B. mit dem SIFT-<sup>8</sup> (Lowe (2004)) oder SURF-Algorithmus<sup>9</sup> (Bay u. a. (2008)) lokale Bildmerkmale<sup>10</sup> extrahiert und mit einer zuvor hinterlegten Referenz verglichen.

Solch ein Verfahren ist zwar prinzipiell gut für die Indoorlokalisierung geeignet. Mit einfachen dekorativen Maßnahmen ließen sich dabei auch nicht zu unterscheidende Räumlichkeiten gut individualisieren. Allerdings ist ein solches Verfahren vor allem aufgrund der genannten Nachteile wie des hohen Rechenaufwands und des größeren Speicherverbrauchs auf mobilen Geräten leider zur Zeit noch nicht geeignet (Gossow u. a. (2011)).

V. Schmude (2011) stellt für den Roboterfußball ein Positionierungsverfahren vor, bei dem über die Auswertung von Kantenfiltern sowie von farbigen Flächen eines Bildes Rückschlüsse auf die eigene Position, die der Mitspieler, der Tore und des

<sup>8</sup>Scale Invariant Feature Transform

<sup>9</sup>Speeded Up Robust Features

<sup>10</sup>Ähnlich der Analyse von Fingerabdrücken

Balles gezogen werden. Auch innerhalb von Gebäuden wäre durch Abgleich dieser Auswertung mit baulichen Strukturen wie Türen, Fenster, Wände oder Fußböden eine Positionierung möglich. Da Wände, Türen, Fenster, etc. aber in der realen Umgebung fast immer ein gleiches Erscheinungsbild haben, dürfte es schwierig werden dieses Verfahren außerhalb einer künstlichen Umgebung zu verwenden.

**Roth (2005)** stellt ein weiteres Verfahren vor, das Tiefeninformationen für die Lokalisierung verwendet. Per stereoskopischer Kamera oder dem "Structure from motion"-Verfahren, werden dabei die Tiefeninformationen aus dem Bildmaterial ermittelt. Über diese Informationen ließen sich Teile des 3D-Raummodells erfassen und mit einem bekannten Modell abgleichen um somit die eigene Position ermitteln zu können.

Dieses Verfahren ist zwar theoretisch vorstellbar, da aber um die eigene Position zu berechnen erst das umgebene Raummodell erfasst und dann abgeglichen werden muss, sehr rechenaufwändig. Zudem kommt dabei noch hinzu, dass bei großen oder gleichartigen Räumlichkeiten das Erfassen des Modells bzw. die eindeutige Identifizierung nicht möglich ist.

## 2.8 Bewertung

Für eine Lokalisierung innerhalb von Gebäuden sind die Outdoorverfahren GNSS- und GSM-Ortung vollkommen ungeeignet, da diese wenn überhaupt anwendbar, mit Konsumergeräten derzeit viel zu ungenau sind (siehe Tabelle 2.1). Dabei ist es im besten Fall noch möglich das Gebäude zu identifizieren, aber Stockwerke oder gar Räume lassen sich damit nicht mehr verorten.

Die Lösungen die hingegen auf eine eigene Infrastruktur in Form eines Sensornetzwerkes oder verteilten Beacons setzen, sind je nach nötiger Positionsgenauigkeit brauchbar. Wobei die mit Ultraschall arbeitenden Systeme vor allem ihre Stärken in der guten Positionierung von 10-30 cm und der geringen Anzahl an Raumfehlern haben. Infrarot kommt über Raumbgenauigkeit nicht hinaus und Funk alleine liefert oft zu viele Raumfehler. Da alle diese Systeme spezielle Hardware für den Client benötigen und die Installation und Wartung des Sensor- / Sendernetzwerk aufwändig

## 2 Vergleichbare Arbeiten

Name	Kategorie	Basistechnik	Medium	Genauigkeit
GPS	Satelliten	TOA	Funk	3m
SpotON	Indoor	Signalstärke	Funk	3m
Active Badge	Indoor	COO	Infrarot	Zelle
Cricket	Indoor	TOA	Ultraschall/Funk	0,3m
RFID	Indoor	COO	Funk	Zelle
Visual Tags	Indoor	Video	Optisch	Abhängig von der Kameraauflö- sung

Tabelle 2.1: Vergleichstabelle - Roth (2005)

ist, sind diese Verfahren oft nur in speziellen Anwendungsfällen zu gebrauchen.

Die WLAN-Ortung punktet zwar mit der in der Regel bereits vorhandenen Infrastruktur an Access Points und vorhandenen mobilen Endgeräten und erspart somit den zusätzlichen Aufwand zur Installation von weiterer Hard- und Software. Dabei setzt es aber großen Aufwand in der Erstellung von eingemessenen Empfangsstärkearten voraus und schafft dennoch meist keine hinreichend genaue Ortung, da bei vielen modernen Geräten eine automatische Anpassung der Sendeleistung genutzt wird und diese zudem noch von vielen Quellen wie Mikrowellengeräte, Türen oder Menschen gestört werden.

Die Auswertung von Bildmaterial hingegen ist bis auf die Marker gestützten Verfahren nur mit hohem Rechenaufwand möglich und lässt sich so nur schwer auf mobilen Geräten oder in großen Installationen realisieren.

Wie in der Vergleichstabelle 2.2 zu erkennen ist, liefern die optischen Verfahren welche die Positionierung anhand von 2D-Markern ermitteln eine Genauigkeit im Zentimeter- bis Dezimeterbereich. Dabei erreichen sie auch in der Regel mit mehr als 10 Positionsbestimmungen pro Sekunde eine gute zeitliche Abdeckung, so dass auch zeitkritischere Anwendung wie Navigation ohne Verzögerungseffekt betrieben werden können. Bis auf die Lösung von Mulloni (SignPost) (Mulloni u. a. (2009)) arbeiten dabei aber alle anderen Systeme mit spezieller Hardware und / oder leistungsfähiger

geren Rechnern und nicht auf gängigen und verbreiteten mobilen Endgeräten wie Smartphones, PDAs oder Tabletcomputer.

Im Vergleich zu den bestehenden Lösungen die hier vorgestellt wurden, unterscheidet sich die zu erstellende Lösung durch die Kombination der nachfolgenden Eigenschaften:

- Bereitstellung von 3D-Positions- und Orientierungsangaben
- der Plattform der mobilen Clientanwendung, die mit mobilen Smartphones ein für den Anwender vorhandenes und vertrautes Gerät darstellt
- die ressourcenschonende, preislich günstige und flexible Verwendung von 2D-Marker

## 2.9 Zusammenfassung

In diesem Kapitel wurden verschiedene Ansätze zur Lokalisierung vorgestellt und bewertet. Dabei zeigte sich, dass markerbasierte visuelle Verfahren, vor allem für die Anwendung innerhalb von Gebäuden, viele wünschenswerte Eigenschaften auf sich vereinen.

Name	Coordinate Reference	Reported Accuracy	Frame Rate
Kohoutek	CityGML	dm	54 Hz
Hile	floor plan	30 cm	0.1 Hz
Kitanov	vector model	dm	10 Hz
Schlaile	edges segments	1 dm/min	50 Hz
Ido	images	30 cm	30 Hz
Sjö	images / scans	sub m	30 Hz
Muffert	images	0.15gon/min	15 Hz
Maye	images / landm.	1 %	2300 Hz
Sky-Trax	coded markers	2 – 30 cm	
Mulloni	coded markers	cm – dm	15 Hz
AICON ProCam	coded markers	0.1 mm	7 Hz
StarGazer	coded markers	cm – dm	20 Hz
Lee	coded markers	dm	30 Hz
naviSCAN3D	projection	50 µm	1 Hz
TrackSense	projection	4 cm	15 Hz
CLIPS	projection	0.5 mm	30 Hz
Habbecke	projection	mm 25	
Popescu	projection	cm 25	15 Hz
NorthStar	projection	cm - dm	10 Hz
DEADALUS	none	0.04 mm	30 Hz
Boochs	none	0.05 mm	
Tappero	none	dm – m	3 Hz
Soloviev	GNSS	cm	
Aufderheide	image features		5-30 Hz
Liu T.	scanner, image	1 %	10 Hz
Liu W.	magnetic field	1 mm	25 Hz

Tabelle 2.2: Vergleichstabelle Optical Indoor Positioning Systems - nach **Mautz und Tilch (2011)** gekürzt

## 3 Analyse

Dieses Kapitel befasst sich in erster Linie mit der Analyse der Anforderungen aus Nutzersicht und legt so die Systemspezifikation fest, die die Basis des späteren Softwareentwurfs sowie der Implementierung ist.

### 3.1 Systemidee

Realisiert werden soll ein System, welches es ermöglicht mobile Endgeräte durch Nutzung der eingebauten Kamera unter Verwendung visueller 2D-Marker innerhalb eines Gebäudes zu verorten und dessen Positions- und Orientierungsangaben für weitere Verarbeitung bereitzustellen. In diesem Rahmen wird eine Evaluation dieser Anwendung in der Umgebung des Living Places der Hochschule für Angewandte Wissenschaften (HAW) Hamburg erfolgen und dabei die Positionen innerhalb des vorhandenen räumlichen Referenzsystems auf einer Anwendung, dem Indoor Spatial Information Service (ISIS)<sup>1</sup>, zu visualisieren.

Dabei ist es in einer Konfigurationsphase vor Inbetriebnahme des Systems nötig, für jeden verwendeten Marker die genaue Position, dessen Ausrichtung, Größe und zu einem Muster vorverarbeitetes Bildmaterial auf einem Patternserver zu hinterlegen. Auf der Seite des mobilen Clients ist hingegen zuvor die Kamera zu kalibrieren.

Für die Betriebsphase müssen vor allem die durch das Living Place benötigten Dienste der Kommunikationsinfrastruktur mit dem Messagingservice ActiveMQ und das Indoor Spatial Information Service (ISIS) bereitgestellt und erreichbar sein, damit der mobile Client die nötigen Markerdaten laden und die ermittelte Positionsangaben publizieren kann.

---

<sup>1</sup>Karstaedt (2012)

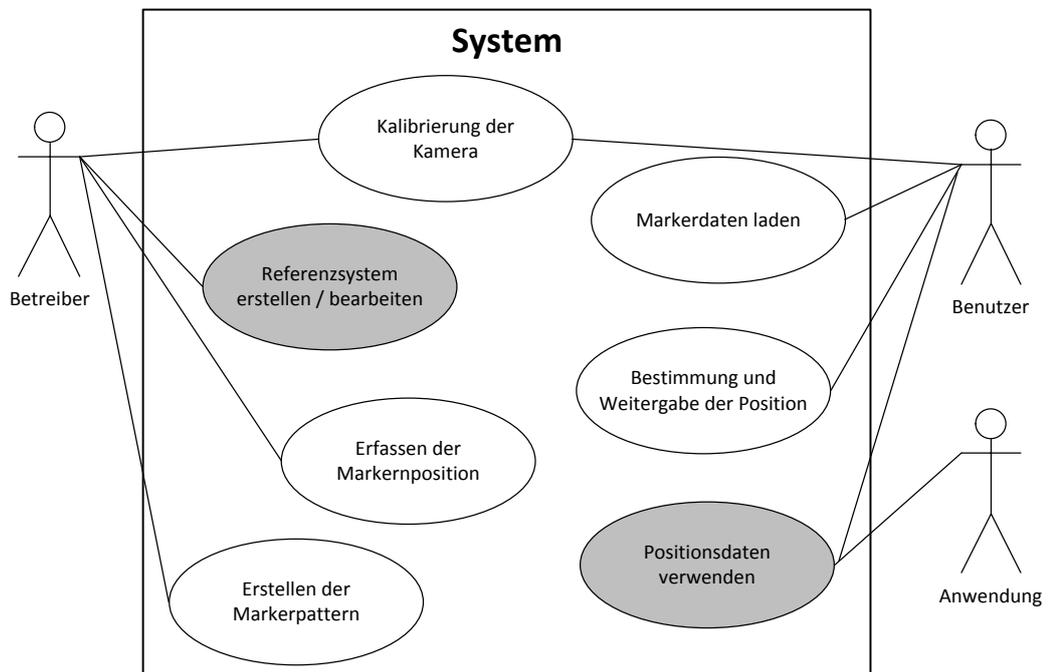
## 3.2 Zielgruppen

Zielgruppen sind vor allem Betreiber und Nutzer großer Gebäude in denen viel Publikumsverkehr besteht wie zum Beispiel Museen, Messen, Behörden, Flughäfen, Universitäten und Krankenhäuser in denen Anwendungen wie Navigation, Augmented-Reality sowie weitere Location Based Services angeboten werden sollen. Des Weiteren ist auch insbesondere die Integration in Smart Homes vorstellbar.

## 3.3 Funktionale Anforderungen

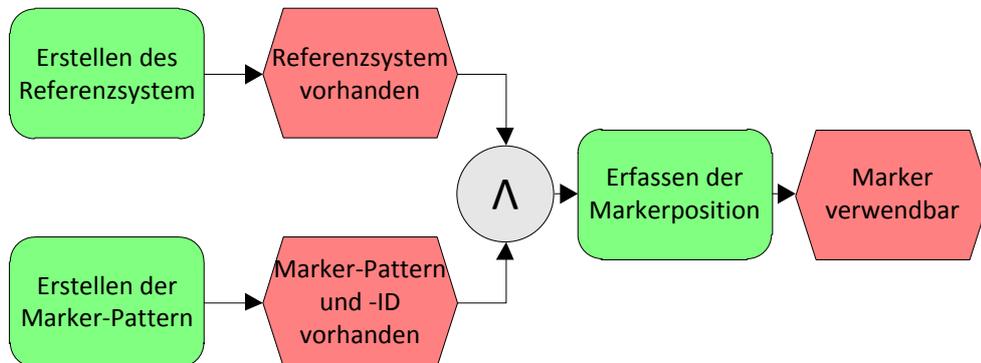
Die funktionalen Anforderungen beschreiben die jeweiligen Anwendungsfälle aus Nutzersicht und werden dabei in Konfigurations- und Betriebsphase, welche im Abschnitt Systemidee beschrieben wurden, unterschieden.

Die einzelnen Akteure nehmen dabei jeweils eine der drei Rollen Betreiber, Nutzer oder Anwendung ein. Der Betreiber ist dabei für die Funktionsfähigkeit der Dienste, für die korrekte Durchführung der Konfigurationsphase und das Bereitstellen der Clientanwendung verantwortlich. Die Nutzer hingegen installieren die Clientanwendung auf ihrem mobilen Endgerät und verwenden diese dann in der Betriebsphase. Die Rolle "Anwendung" beschreibt lediglich eine beliebige externe Anwendung die während der Betriebsphase die publizierten Positions- und Orientierungsdaten verwendet, wie zum Beispiel die Visualisierung der Positionsdaten auf dem ISIS.



### 3.3.1 Anwendungsfälle aus der Konfigurationsphase

Stehen die Dienste der Kommunikationsinfrastruktur sowie des Indoor Spatial Information Service bereit, ist es in einer Konfigurationsphase erforderlich, die verwendeten visuellen 2D-Marker dem System bekannt zu machen und deren Position und Orientierung in einem zuvor festgelegten räumlichen Referenzsystem festzulegen.



Des Weiteren ist die Clientanwendung vor Inbetriebnahme zu kalibrieren.

### Erstellen der Markerpattern

Sollen noch nicht bekannte Marker dem System hinzugefügt werden, muss zum Einen das Bildmaterial und zum Anderen die genauen Abmaße des 2D-Markers an einer zentralen Stelle hinterlegt werden. Dabei wird das Bildmaterial zu einem Muster vorverarbeitet und zusammen mit den Abmaßen einer eindeutigen ID zugeordnet um diese dann später abfragen zu können.

<b>Name des Usecase:</b>	Erstellen der Markerpattern
<b>Aufgabe:</b>	Erzeugt die Referenzdaten zum späteren Erkennen des Markers und weist diesem eine Identifikationsnummer zu
<b>Beteiligter Akteur</b>	Betreiber
<b>Vorbedingung:</b>	
<b>Nachbedingung:</b>	Markerpattern, Größe und -ID persistiert und abrufbar

### Erfassen der Markerposition

Die Positions- und Orientierungsangaben werden im räumlichen Referenzsystem abgelegt, indem dort ein neues Objekt „Marker“ positioniert wird, welches über die Identifikationsnummer einem Pattern zugeordnet ist.

<b>Name des Usecase:</b>	Erfassen der Markerposition
<b>Aufgabe:</b>	Position und Orientierung des Markers werden zusammen mit der PatternID hinterlegt
<b>Beteiligter Akteur</b>	Betreiber
<b>Vorbedingung:</b>	Marker zuvor erfasst und Referenzsystem vorhanden
<b>Nachbedingung:</b>	Position und Orientierung persistiert und mittels ID abfragbar

#### **Kalibrierung der Kamera**

Aufgrund der verschiedenen Kombinationen von Kamera und Objektiv ist eine Kalibrierung unbedingt nötig um später gute Entfernungsabschätzungen zu bekommen. Die in der Clientanwendung integrierte Kalibrierung muss vor allem die Brennweite berücksichtigen, was zu einer linearen Korrektur des Messwertes führt. Um eine bestmögliche Erkennungsleistung der Marker in den Randbereichen des Bildes zu erhalten, sollten zusätzlich noch der Bildmittelpunkt bestimmt sowie die optische Verzerrung der Linse herausgerechnet werden.

<b>Name des Usecase:</b>	Kalibrierung der Kamera
<b>Aufgabe:</b>	Die Kamera wird in einem Winkel von 90° auf den bei der Kalibrierung verwendeten Marker gerichtet. Dabei werden solange mehrere Bilder aus unterschiedlichen Entfernungen aufgenommen und diesen zugeordnet bis die Genauigkeit der berechneten Entfernungsabschätzung ausreichend ist.
<b>Beteiligter Akteur</b>	Benutzer / Betreiber
<b>Vorbedingung:</b>	keine
<b>Nachbedingung:</b>	Entfernungsberechnung zwischen Kamera und Marker nun innerhalb systembedingter Varianz.

### 3.3.2 Anwendungsfälle aus der Betriebsphase

Während der Betriebsphase werden die Positionen der mobilen Endgeräte ermittelt und für weitere Verwendungen über die Kommunikationsplattform den interessierten Komponenten bereitgestellt.

#### Markerdaten laden

Da die 2D-Marker nur mittels vorher berechneter Muster erkannt werden können, ist es nötig diese auf dem Mobilteil zu hinterlegen. Dazu wählt der Benutzer den räumlichen Kontext aus in dem er sich zurzeit bewegt wie zum Beispiel das Living Place. Daraufhin werden dann vom Client über eine Suche im räumlichen Referenzsystem alle Marker ermittelt und deren Pattern anschließend von dem dazugehörigen Server geladen.

<b>Name des Usecase:</b>	Markerdaten laden
<b>Aufgabe:</b>	Der Benutzer wählt beim Start der Clientanwendung den von ihm gewünschten räumlichen Kontext aus, was das Laden der nötigen Markerdaten veranlasst
<b>Beteiligter Akteur</b>	Benutzer
<b>Vorbedingung:</b>	Backendsystem einsatzbereit
<b>Nachbedingung:</b>	Client einsatzbereit

#### Bestimmung und Weitergabe der Position

Das mobile Endgerät ermittelt über die Auswertung des Videostreams die 2D-Marker im Bildmaterial und wählt dabei den Marker, welcher am sichersten erkannt wurde, aus. Von diesem Marker wird dann die Kameraposition innerhalb des Markerkoordinatensystems berechnet und mit Hilfe der bekannten Markerposition in das vom System verwendete Referenzsystem transformiert. Anschließend wird die ermittelte Position automatisch über die Kommunikationsinfrastruktur publiziert.

<b>Name des Usecase:</b>	Bestimmung der Position
<b>Aufgabe:</b>	Der Benutzer bewegt sich mit laufender Clientanwendung durch den Raum und richtet die Kamera dabei auf einen oder mehrere Marker, so dass der Marker erkannt und daraus die 3D-Positionsangaben berechnet werden kann.
<b>Beteiligter Akteur</b>	Benutzer
<b>Vorbedingung:</b>	Markerdaten geladen sowie Marker in Sichtweite
<b>Nachbedingung:</b>	Position und Orientierung innerhalb des Referenzsystems bekannt

## 3.4 Nichtfunktionale Anforderungen

### 3.4.1 Zuverlässigkeit

Client und Backendsysteme sollten sich jeweils stets in einem konsistenten Zustand befinden und im Falle eines kompletten Systemversagens durch einen einfachen Neustart wieder betriebsbereit sein. Bei Fehlbedienung des Clients sowie auch bei Kommunikationsproblemen, ist der Benutzer umgehend zu informieren. Auf der Serverseite werden alle relevanten Ausnahmen behandelt und für spätere Auswertung in eine Logdatei geschrieben.

### 3.4.2 Effizienz

Aufgrund der sehr beschränkten Ressourcen der Clientseite sollte dort sparsam damit umgegangen werden, so dass auch ältere Geräte die alle entsprechenden Voraussetzungen erfüllen noch nutzbar sind. Des Weiteren ist vor allem auf den Energieverbrauch zu achten, so dass ein möglichst langer Betrieb ohne Aufladen möglich ist.

### 3.4.3 Verfügbarkeit

Client und Backendsysteme sollten stets bereit und mindestens innerhalb der Gebäubetriebszeiten verfügbar sein. Innerhalb vieler Szenarien dürfte dieses einem 24/7 Betrieb entsprechen.

### 3.4.4 Reaktionszeiten

Die Reaktionszeiten der Backendanfragen sollten stets unter einer Sekunde liegen, da die Usability der Anwendung darunter sonst zu stark leiden würde.

### 3.4.5 Portierbarkeit

Die Clientanwendung sollte soweit möglich plattformunabhängig entwickelt und ohne große Änderung auf möglichst vielen Versionen der mobilen Endgeräte lauffähig sein. Der Server muss auf den gängigen Serverbetriebssystemen ausführbar und im besten Fall über seine Laufzeitumgebung komplett plattformunabhängig sein.

## 3.5 Zusammenfassung

In diesem Kapitel wurden die Anforderungen definiert, die für das Design im nächsten Kapitel erforderlich sind und bei der Realisierung der entstehenden Anwendung erfüllt werden sollten. Dabei ergab sich aus der Systemanalyse die zu lösenden Teilaufgaben für das Backend sowie des mobilen Clients:

#### **Backend:**

**Bereitstellen der Markerreferenzdaten:** Diese bestehen aus Markerpattern und dessen Größe.

**Bereitstellen der Markerposition:** Bestehend aus der Positionierung im Referenzsystem, als auch aus der Orientierung im Raum.

#### **mobiler Client:**

**Kalibrierung der Kamera:** Dabei werden die nötigen Parameter wie Brennweite, etc. bestimmt um später eine möglichst gute Entfernungsabschätzung zu erhalten.

**Laden der Markerdaten:** Lädt die von dem Backend bereitgestellten Markerdaten auf das mobile Endgerät.

**Bestimmung der Position und Orientierung:** Aus dem Bildmaterial werden die Marker ermittelt und die Position sowie die Orientierung innerhalb des räumlichen Referenzsystems berechnet.

**Publizieren der Positionsangaben:** Die Positionsangaben werden allen interessierten Backendanwendungen bereitgestellt.

## 4 Design

Im Kapitel 3 „Analyse“ wurde das System aus einer funktionalen Benutzersicht beschrieben. Da diese für eine Implementierung noch viel zu abstrakt ist, wird in diesem Kapitel eine weitere Abstraktionsebene, die der technischen Komponenten, eingeführt. Sie legen hauptsächlich die Struktur und die Interaktionen innerhalb der Anwendung fest, verbirgt aber weiterhin die meisten Implementierungsdetails.

Da diese Arbeit im Rahmen der Smart-Home-Umgebung „Living Place<sup>1</sup>“ der Hochschule für Angewandte Wissenschaften (HAW) Hamburg demonstriert wird, wird zuerst die bestehende Systemumgebung des Living Places vorgestellt und die daraus resultierenden Randbedingungen erörtert. Des Weiteren werden daraufhin die zentralen Designentscheidungen sowie die daraus folgende Softwarearchitektur dargestellt.

### 4.1 Bestehende Systemumgebung

#### 4.1.1 Indoor Spatial Information Service (ISIS)

Der Indoor Spatial Information Service entstand während der Masterarbeit von Bastian Karstaedt (Karstaedt (2012); Karstaedt und Wendholt (2011); Ellenberg u. a. (2011)) und stellt räumliches Kontextwissen für weitere Anwendungen sowie eine 2D- /3D-Visualisierung des Gebäudemodells inklusive Mobiliar und Personen bereit. Zum Beispiel sind dabei Anfragen wie „Welches Objekt befindet sich an Position (X,Y,Z)?“ oder „Welche Position hat das Objekt mit der ID <id>?“ vorhanden. Aber auch Anfragen mit räumlichen Relationen, wie zum Beispiel ob sich ein bestimmtes Objekt oder welche Objekte sich innerhalb, auf oder in der Nähe eines anderen befindet, sind

---

<sup>1</sup><http://livingplace.informatik.haw-hamburg.de/blog/>

teilweise vorhanden. So können zum Beispiel alle 2D-Marker innerhalb eines Gebäudes bestimmt und deren Position abgefragt werden.

### 4.1.2 Blackboard Architektur im Living Place

Bei der Blackboardarchitektur handelt es sich um ein Architekturmuster, welches die Lösung von Problemen unterstützt, für die keine feste Lösungsstrategie vorhanden sein muss. Es reicht hingegen aus, dass das Problem sich in spezifische Teilprobleme zerlegen lässt, die dann jeweils von spezialisierten Programmen gelöst werden können. Diese Spezialisten stehen in keiner direkten Verbindung miteinander, sondern beobachten nur das „Blackboard“ und untersuchen die darauf hinterlegten Daten, ob die für ihre Arbeit nötigen Vorbedingungen vorhanden sind. Ist dieses der Fall, wird das von dem Spezialisten berechnete Ergebnis wiederum auf dem „Blackboard“ hinterlegt und befähigt somit hoffentlich weitere Spezialisten ihre Arbeit zu ermöglichen.

Im Living Place wurde das Blackboard als Publish/Subscribe-Modell entworfen und durch ActiveMQ<sup>2</sup> realisiert. Jeder Spezialist, der dort an irgendeiner Problemlösung beteiligt ist, abonniert die für ihn nötigen Informationen und veröffentlicht auch seine Ergebnisse dort. Die über dieses System ausgetauschten Informationen werden als JSON-Nachrichten<sup>3</sup> versandt. Dabei werden zwei unterschiedliche Verfahren mit „Topic“ und „Queue“ angeboten.

Die Queue-Methode funktioniert ähnlich der gleichnamigen Datenstruktur und verbindet mehrere Publisher mit einem Subscriber und stellt somit eine „N zu 1“-Beziehung (siehe Abbildung 4.1) zwischen diesen her, wobei die eintreffenden Nachrichten erst einmal in einem FiFo-Puffer zwischengespeichert werden und dort auf ihre Verarbeitung warten. Sollten mehrere Verbraucher eine Queue abonniert haben, werden diese zum Lastausgleich verwendet, so dass auch in diesem Fall jede Nachricht nur einmal ausgeliefert würde.

Im Gegensatz zur Queue verkettet die „Topic“-Methode (siehe Abbildung ref:fig:ActiveMQTopic) die Publisher und Subscriber über eine „N zu M“-Beziehung, so

---

<sup>2</sup><http://activemq.apache.org/>

<sup>3</sup><http://json.org/>

dass jede eingehende Nachricht sofort an alle Abonnenten ausgeliefert wird.

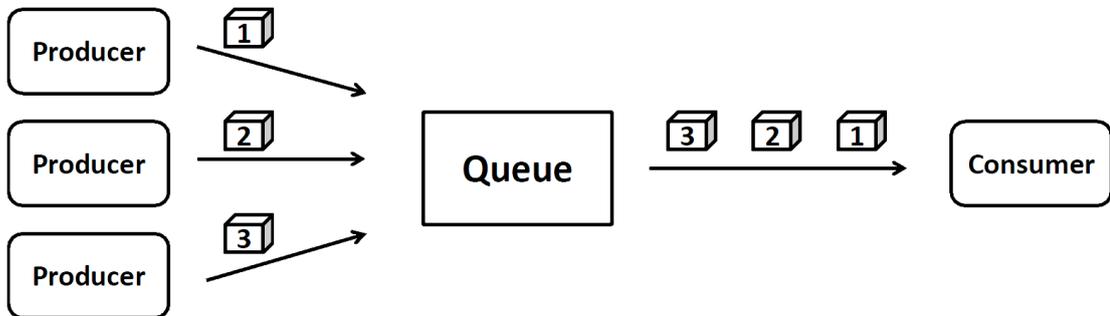


Abbildung 4.1: ActiveMQ Queue

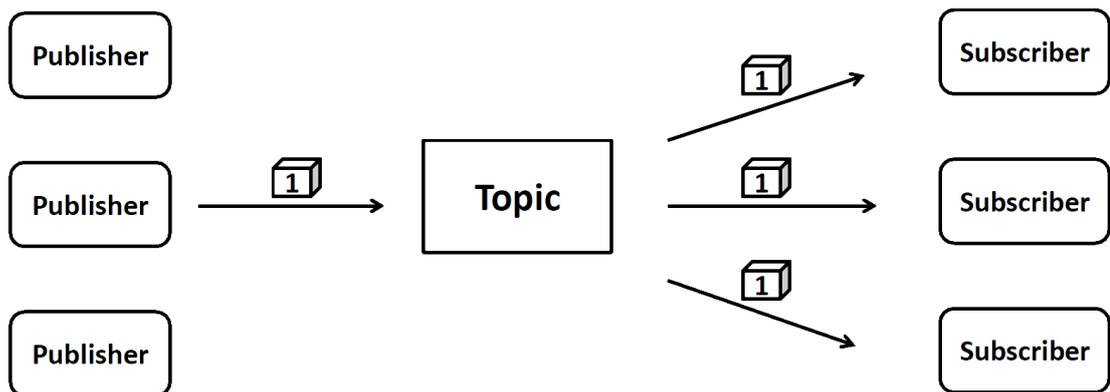


Abbildung 4.2: ActiveMQ Topic

Die Kommunikation findet wie in [Karstaedt \(2012\)](#) beschrieben über das Blackboard statt. Im Falle der Positionsübermittlung (siehe [Abbildung 4.3 \(2a\)](#)) werden die Positionsdaten in dem vorgegebenen Nachrichtenformat an das entsprechende Topic des ActiveMQ-Dienstes übergeben, welcher dann allen Subscriber (unter anderem auch dem ISIS) diese Daten ausliefert. Bei der Abfrage der räumlichen Dienste, wie zum Beispiel welche Marker sich wo im Gebäudemodell befinden, handelt es sich im eigentlichen Sinne eher um ein Request-Response-Verfahren, welches sich nicht be-

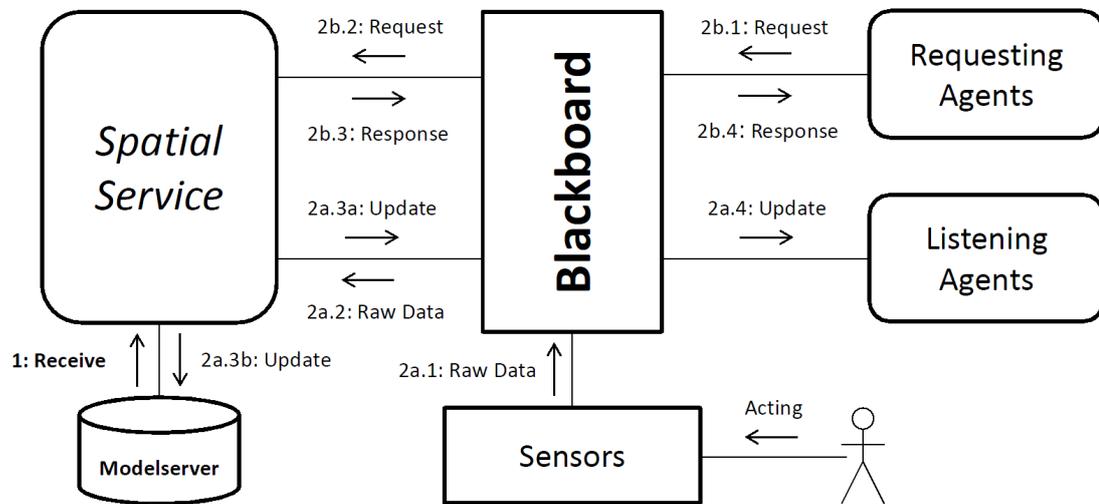


Abbildung 4.3: Blackboard

sonders gut in das Konzept des Blackboards eingliedert. Um aber Medienbrüche zu verhindern, werden die Anfragen eines „Requesting Agents“ (Karstaedt (2012)), siehe Abbildung 4.3 (2b), hier an eine Queue des Nachrichtendienstes gesandt, dann vom ISIS ausgewertet und schließlich über ein vom Client erstelltes Topic beantwortet.

Da unter Android unter anderem das Paket „`javax.naming`“ nicht zur Verfügung steht, werden die Nachrichten mittels des von Sven Boris Bornemann für das LivingPlace der HAW entwickelten „AndroidPublisher“ und „AndroidProxy“ übertragen und um die Funktionalität des „Requesting Agents“, der über das oben genannte Publisher und Subscriber Muster ein Request-Response-Verfahren emuliert, erweitert.

## 4.2 Wahl der Clientumgebung

Nach Auswertung der im Kapitel 3 aufgestellten Anforderungen, wurde als Clientplattform ein Android-Smartphone gewählt. Es ist in der gewählten Benutzergruppe weit verbreitet, mit einem Straßenpreis ab 100 € relativ günstig und besitzt alle nötigen Hardwareanforderungen wie die Netzwerkkommunikation über GPRS oder WLAN,

eine Kamera und ausreichend Arbeits-, Flashspeicher sowie Rechenleistung für eine einfache Bildverarbeitung. Zudem ist es eine „open source“ Plattform mit einer guten Entwicklungsumgebung.

Mit der Wahl des mobilen Endgerätes ist auch zugleich die Programmiersprache Java, inklusive des Android-SDKs mit seinen, wenn auch wenigen aber dennoch vorhandenen Einschränkungen, festgesetzt. Positiv hingegen ist dabei der große funktionale Umfang des Java-SDKs und die sehr große Anzahl an frei verfügbaren Bibliotheken, sowie die Verwendung und Unterstützung von Java in anderen Projekten des Living Places.

### 4.3 Wahl des visuellen Positionsbestimmungsverfahrens

Bei dem Verfahren zur visuellen Positionsbestimmung wurde nach Auswertung der in Kapitel 2 beschriebenen Eigenschaften und der in Kapitel 3 entsprechenden Anforderungen ein 2D-Marker basierendes Verfahren gewählt. Dieses ist vor allem aufgrund der relativ hohen Verarbeitungsgeschwindigkeit und des geringen Speicherplatzbedarfs bei der Erkennung der Marker für den mobilen Einsatz interessant.

Die Auswahlkriterien für eine passende Bibliothek waren:

- Lauffähigkeit unter Java/Android-Plattform
- freie Verfügbarkeit
- stabile Version, welche möglichst schon von anderen Projekten verwendet wurde

Alle diese Eigenschaften konnte vor allem das in C geschriebene ARToolKit<sup>4</sup> mit seinen beiden Java-Ablegern NyARToolkit<sup>5</sup> und AndAR<sup>6</sup> auf sich vereinigen. Nach verschiedenen Tests mit AndAR, welches hauptsächlich ein Java Native Interface Wrapper mit entsprechenden Schnittstellen für das klassische ARToolKit ist und

---

<sup>4</sup><http://www.hitl.washington.edu/artoolkit/>

<sup>5</sup><http://nyatla.jp/nyartoolkit/wp/>

<sup>6</sup><http://code.google.com/p/andar/>

dem mit NyARToolkit komplett in Java nachgebauten ARToolKit zeigte sich, dass NyARToolkit eine höhere Geschwindigkeit und damit auch eine leicht bessere Erkennungsleistung bietet. Des Weiteren ist die Verwendung der Low-Level-Funktionen der Markererkennung und Auswertung des in Java geschriebenen NyARToolkit sehr viel einfacher.

ARToolKit ist eigentlich ein Framework welches 3D-Modelle oder ganze Szenen über einem Livebild darstellt und so platziert, als würden sich diese nahtlos in die aufgenommene Umgebung einfügen.

Wie in Abbildung 4.4 zu sehen ist, wird zuerst das Bildmaterial in ein Schwarz-Weiß-Abbild überführt und nach den schwarzen Rändern der Marker gesucht. Anschließend wird über die perspektivische Auswertung der Eckpunkte Position und Orientierung der Marker berechnet. Um die einzelnen Marker voneinander unterscheiden zu können, wird das Bild innerhalb der schwarzen Ränder mittels eines Rasters in einzelne Blöcke unterteilt. Jedes dieser Blöcke wird im Anschluss dessen auf das Verhältnis von weißen zu schwarzen Pixeln hin untersucht. Über den Vergleich dieses Graustufenrasters mit den jeweils bekannten Pattern wird dann die Identifikationsnummer bestimmt. Mittels dieser und der ermittelten Positionsangaben wird im nächsten Schritt das entsprechende 3D-Objekt der Markerposition angepasst und anschließend gezeichnet.

Da in dieser Arbeit kein Bedarf an einer dreidimensionalen Darstellung von Objekten besteht, kann bei der Implementierung nur auf die internen Klassen des Markertrackings zurückgegriffen werden. Woraus auch folgt, dass die Handhabung des Videostroms in der Verantwortung der Anwendung liegt.

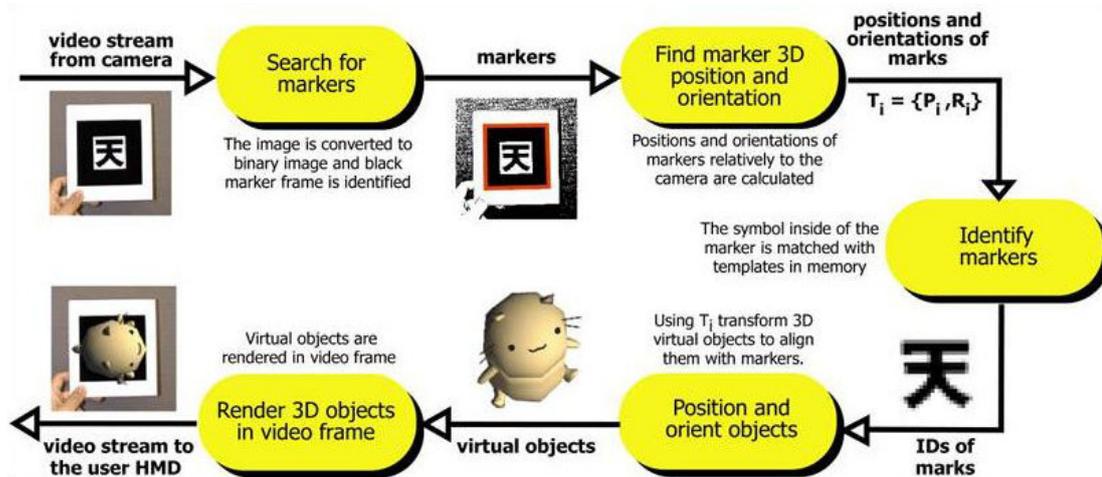


Abbildung 4.4: ARToolKit Workflow

## 4.4 Designentscheidung

### 4.4.1 Architekturmuster Pipes and Filters

Um die Verarbeitungsleistung des Videodatenstroms auf Multiprozessorsystemen zu verbessern, werden vor allem die rechenintensiven Anwendungskomponenten parallel ausgeführt. Dabei wird der Datenstrom der Kamerabilder mit dem Pipes-and-Filters Muster verarbeitet. Im Idealfall wird dabei der Durchsatz (gemessen in frames per second) nur noch von der langsamsten Komponente bestimmt.

Das Pipes-and-Filters-Entwurfsmuster ist gut für die Verarbeitung von Datenströmen geeignet. Dabei werden die einzelnen Verarbeitungsschritte jeweils in Filtern gekapselt. Diese lassen sich so einzeln entwickeln, warten und für die Verwendung beliebig kombinieren. Dabei wird der Verarbeitungsfluss der Daten ausgehend von einer Datenquelle in den einzelnen Filtern transformiert und endet zum Schluss in einer Datensenke.

Sollte es später nötig sein die Verarbeitungsleistung auf Prozessoren mit mehr als zwei Kernen zu erhöhen, ließe sich die ganze Verarbeitungspipeline sehr einfach

durch den Einsatz von Multiplexing parallelisieren. Dabei würde ein DeMultiplexer kurz nach der Datenquelle die einzelnen Frames auf N-Verarbeitungspipelines aufteilen und dessen Ergebnisse am Ende von einem Multiplexer wieder serialisiert werden.

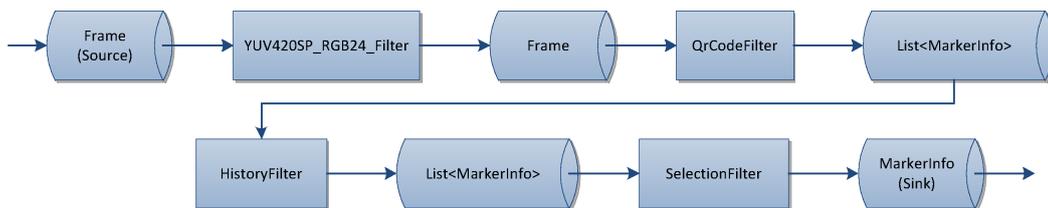


Abbildung 4.5: Pipes and Filters sowie die einzelnen Filterschritte der Anwendung

### 4.4.2 Observable Results

Sobald eine neu berechnete Positionsangabe verfügbar ist, werden alle daran interessierten Komponenten der Clientanwendung über das Observer-Pattern [Gamma u. a. \(1996\)](#), siehe [Abbildung 4.6](#), informiert und stellt somit ein einfaches Publish-Subscribe-Verhalten dar. Vorteilhaft dabei ist die 1-N-Beziehung (Multicast). So lassen sich zum Beispiel die Aktualisierung der Clientbenutzeroberfläche sowie das Publizieren der Positionsangaben zu einem Server leicht umsetzen. Ein weiterer großer Vorteil daran ist die lose Kopplung der Komponenten, welches ein einfaches Austauschen der Beobachter ermöglicht.

### 4.4.3 MarkerBuilder

Für die Erkennung der 2D-Marker sowie zur Berechnung der Kameraposition sind die Angaben wie Position, Orientierung, Breite und Pattern des Markers nötig. Diese Daten werden von der MarkeBuilder bereitgestellt, indem diese über den räumlichen Kontext eine Anfrage nach allen 2D-Markern an den ISIS stellt und anschließend die nötigen Pattern von einem Webserver herunterlädt. [Abbildung 4.7](#) veranschaulicht diese Kommunikation.

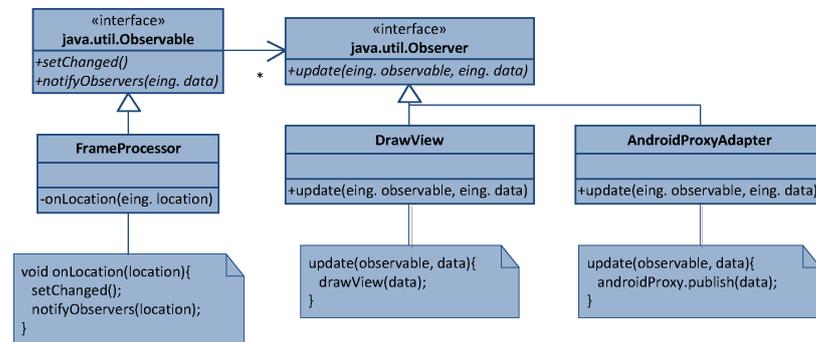


Abbildung 4.6: Observer Pattern

## 4.5 Klassendiagramm

Das Klassendiagramm aus Abbildung 4.8 veranschaulicht die Struktur und das Zusammenwirken der einzelnen Klassen in der Anwendung. Im Folgenden soll noch einmal kurz auf jede einzelne Klasse eingegangen werden:

### CamLocationActivity

Dieses ist die Hauptklasse einer Androidanwendung, sie wird vor allem für das Lifecycle-Management der Anwendung verwendet.

### MarkerBuilder

Die MarkerBuilder stellt eine Liste der nötigen Marker mit den Angaben zur ID, Breite, Pattern sowie der Position und Orientierung innerhalb des Weltkoordinatensystems bereit.

### CamPreview

CamPreview verwalten das Android Camera Objekt und sorgt so dafür, dass der Videostrom als Preview zur Anzeige auf der Benutzeroberfläche im entsprechenden

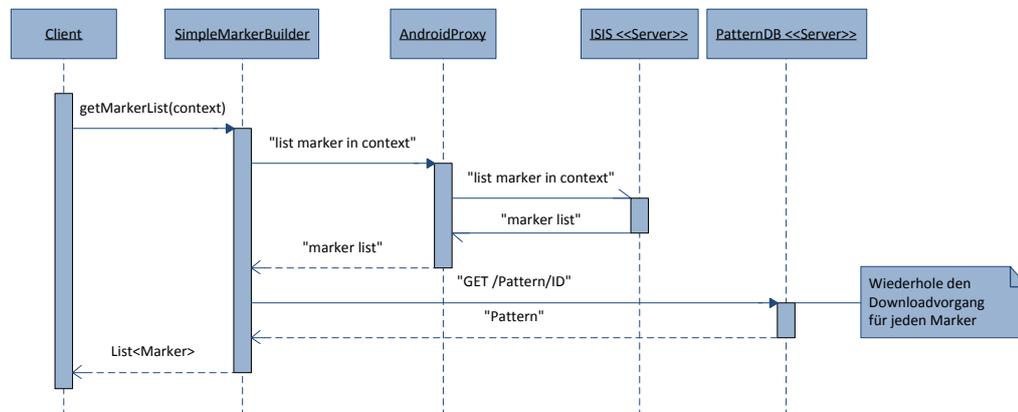


Abbildung 4.7: Sequenzdiagramm MarkerBuilder

SurfaceView landet<sup>7</sup>. Des Weiteren werden die einzelnen Frames dieses Stromes dort an den eigentlichen Verarbeitungsprozess zur Bildanalyse dem FrameProcessor übergeben.

### FrameProcessor

Der FrameProcessor ist die zentrale Komponente zur Bildauswertung. Alle an dem Ergebnis der Bildanalyse interessierten Klassen können sich als Beobachter bei ihm registrieren. Er initialisiert außerdem die benötigten Pipes und Filter, nimmt die Frames des Videostromes entgegen und stellt sie in der Quelle des ersten Filters für die weitere Bearbeitung bereit. Dabei überwacht er gleichzeitig den Füllstand der Senke des letzten Filters (MarkerInfo). Wenn dort Daten vorhanden sind, entnimmt er diese und benachrichtigt alle registrierten Beobachter<sup>8</sup>.

### ThreadFilter

Dieser Filter ist eine abstrakte Klasse die von `java.lang.Thread` abgeleitet

<sup>7</sup>Observer Pattern

<sup>8</sup>Observer Pattern

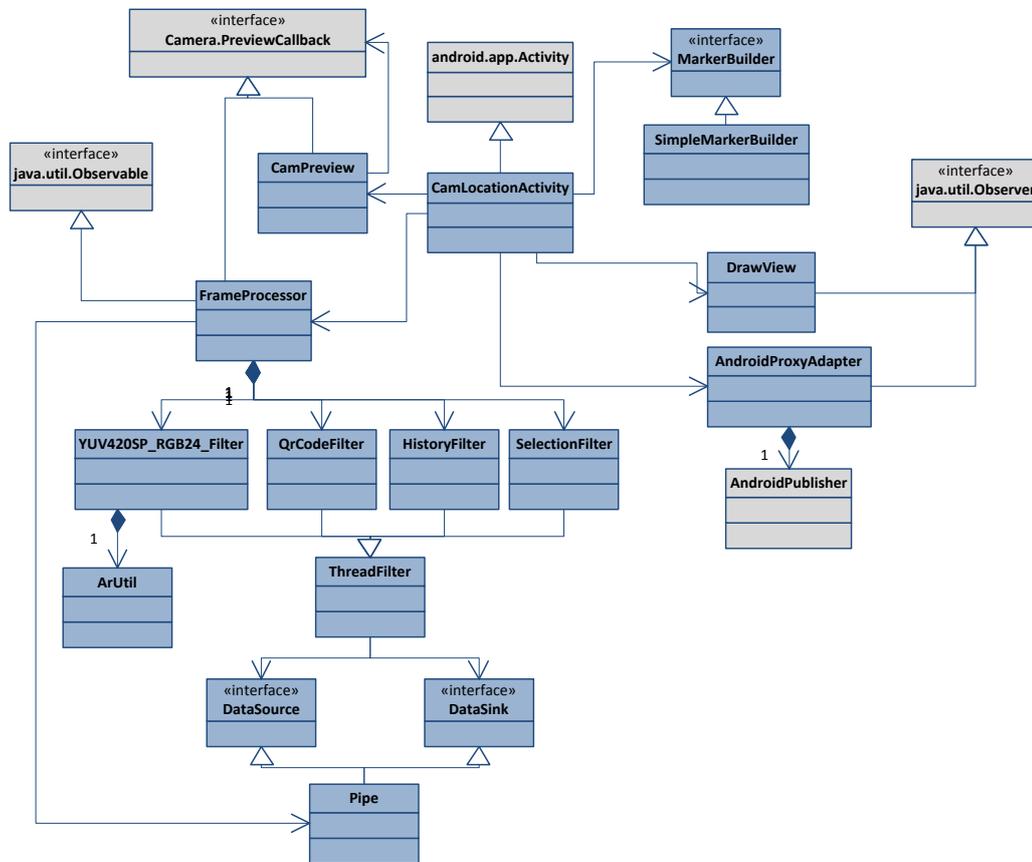


Abbildung 4.8: Klassendiagramm

wurde. Die `run()`-Methode definiert hier als Schablonenmethode (Gamma u. a. (1996)) das nebenläufige Verarbeiten einer Filterstufe, indem sie aus der Quelle die Daten einliest, diese über die abstrakte Methode `process()` von den abgeleiteten Klassen verarbeiten lässt und anschließend das Ergebnis in der Senke für die nächste Filterstufe zur Verfügung stellt.

### YUV420SP\_RGB24\_Filter

`YUV420SP_RGB24.Filter` erbt von `ThreadFilter` und konvertiert in seiner `process()`-Methode jedes Videoframe vom YUV420SP-Format in das RGB24-

Format.

### **ArUtil**

`ArUtil` dient als Hilfsklasse bei der Konvertierung der Bildformate. Es lädt beim Start der Anwendung die native Bibliothek „`libyuv420sp2rgb.so`“ und bietet die entsprechenden statischen Methoden zu dessen Verwendung an.

### **QrCodeFilter**

Dieser `Filter` führt mittels `NyARToolkit` die eigentliche Bildanalyse aus. Werden dabei Marker erkannt, liefert `NyARToolkit` die relative Position des Markers innerhalb des Kamerakoordinatensystems. Diese wird dann in das Markerkoordinatensystem invertiert, so dass man die relative Position der Kamera zu dem erkannten Marker erhält. Anschließend wird das Markerkoordinatensystem in das des verwendeten „Weltkoordinatensystems“ transformiert und per Translation um die Position des Markers verschoben, um die Position und Orientierung der Kamera im Weltkoordinatensystem zu erhalten.

### **HistoryFilter**

Der `HistoryFilter` berechnet über die eingehenden Markerdaten jeweils einen gleitenden Durchschnitt.

### **SelectionFilter**

Aus den eingehenden Markerdaten wird hier der aktuell beste Marker anhand des Konfidenzwertes aus der Bildanalyse ausgewählt und nur dieser den nachfolgenden Filtern bereitgestellt.

### **DrawView**

`DrawView` ist eine GUI-Komponente welche die wichtigsten Markerdaten, die sie als Observer des `FrameProcessors` mitgeteilt bekommt, auf dem Videostrom anzeigt. Dieses umfasst:

- die relative Position der Kamera zum Marker

- die Kameraposition im Weltkoordinatensystem
- den Konfidenzwert des Markers
- die aktuell erreichte Erkennungsrate in fps<sup>9</sup>
- sowie eine grafische Umrandung des konkreten verwendeten Markers



Abbildung 4.9: Screenshot der Anwendung

### **AndroidProxyAdapter**

Der `AndroidProxyAdapter` bekommt als Observer des `FrameProcessors` die jeweiligen aktuellen Markerdaten mitgeteilt, verpackt diese in eine `ActiveMQ`-Nachricht und versendet diese über den `AndroidPublisher`.

---

<sup>9</sup>frames per second

## 4.6 Zusammenfassung

In diesem Kapitel wurde aus den zuvor erarbeiteten Anforderungen ein Softwaredesign für ein System zur visuellen Positionsbestimmung erarbeitet und gezeigt wie sich diese Lösung in den „Living Place“ der Hochschule für Angewandte Wissenschaften einbindet.

Dabei wurde im Besonderen mit dem „Pipes and Filters“-Muster auf eine gute Erweiterbarkeit, Skalierbarkeit bei der Positionsbestimmung geachtet und gezeigt, wie sich die Entkopplung der logischen Marker innerhalb des Referenzsystems von ihrer visuellen Repräsentation ergibt. Des Weiteren wurde über eine lose Kopplung der Positionsinteressenten (Observer Pattern) und den austauschbaren MarkerBuilder auch hier zukünftige Anpassungen berücksichtigt.

# 5 Evaluierung

In diesem Kapitel wird gezeigt, wie das System sich in einer realen Umgebung während eines Testbetriebes verhält. Des Weiteren wird die Genauigkeit der Positionsbestimmung anhand eines Versuchsaufbaus bestimmt und bewertet.

## 5.1 Kalibrierung

Bevor die Position bestimmt werden kann, muss zunächst die Kamera (das mobile Endgerät) kalibriert werden. Dazu wird die Kamera auf einen bekannten Marker gerichtet. Befindet sich der Konfidenzwert des erkannten Markers in einem hinreichend guten Wertebereich, wird der Anwender aufgefordert eine möglichst exakte Entfernungsangabe einzugeben. Dieser Schritt sollte nach Möglichkeit öfters aus verschiedenen Entfernungen wiederholt werden, bevor die Kalibrierung abgeschlossen wird.

Für die nachfolgenden Tests wurde das Gerät nur einmal mit den Entfernungen 0,5, 1, 1,5 und 2,0 Metern direkt von vorne und auf derselben Höhe kalibriert.

## 5.2 Testbetrieb

Die Anwendung wurde im „Living Place“ getestet und vorgeführt. Dabei zeigte sich, dass bei guten Lichtverhältnissen eine stabile Positionsbestimmung möglich ist. Die Daten werden zuverlässig an den ISIS übermittelt und dort, wie in der Abbildung 5.1 zu sehen ist, in der 2D-Ansicht visualisiert. Die Geschwindigkeit der Positionsbestimmung lag dabei im Durchschnitt bei 14 Hz, konnte aber auch bei optimalen Bedingungen knapp über 20 Hz erreichen.

Bei schwierigeren Lichtverhältnissen, wie Schattenschlag über dem Marker oder bei Gegenlicht, war eine Auswertung nur noch in den seltensten Fällen möglich. Wün-

schenswert wäre hier die Verfügbarkeit von unter- bzw. überbelichteten Bildmaterial wie sie in High Dynamic Range Images<sup>1</sup> vorhanden ist um zumindest der Gegenlichtproblematik entgegen zu wirken.

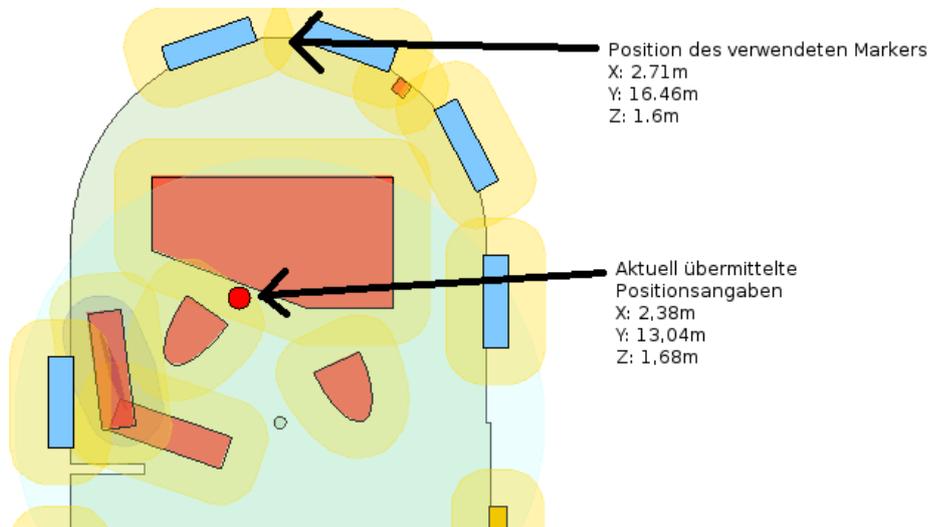


Abbildung 5.1: ISIS-2D Ansicht mit übermittelter Kameraposition im LivingPlace

### 5.3 Ermittlung der tatsächlichen Abweichung

Um die Abweichungen der gemessenen zu der realen Position bestimmen zu können, wurden einige Versuche aus unterschiedlichen Entfernungen mit jeweils verschiedenen Blickwinkeln durchgeführt.

#### 5.3.1 Versuchsaufbau

Das mobile Endgerät wurde an einem Ort fixiert, so dass die Entfernung der Kamera zum Mittelpunkt des Markers genau einer festgelegten Entfernung entspricht. Der Marker ist dabei fest auf einer senkrechten Platte befestigt, welche am Mittelpunkt des Markers um die Z-Achse rotiert werden kann. Dabei wurden die Winkel 90°, 80°, 75°, 45° sowie 25° in den Entfernungen von einem, zwei und drei Metern untersucht.

---

<sup>1</sup>[http://de.wikipedia.org/wiki/High\\_Dynamic\\_Range\\_Image](http://de.wikipedia.org/wiki/High_Dynamic_Range_Image)

Anschließend wurde dieser Versuch noch einmal aus einem Winkel von  $45^\circ$  um die Y- bzw. X-Achse wiederholt.

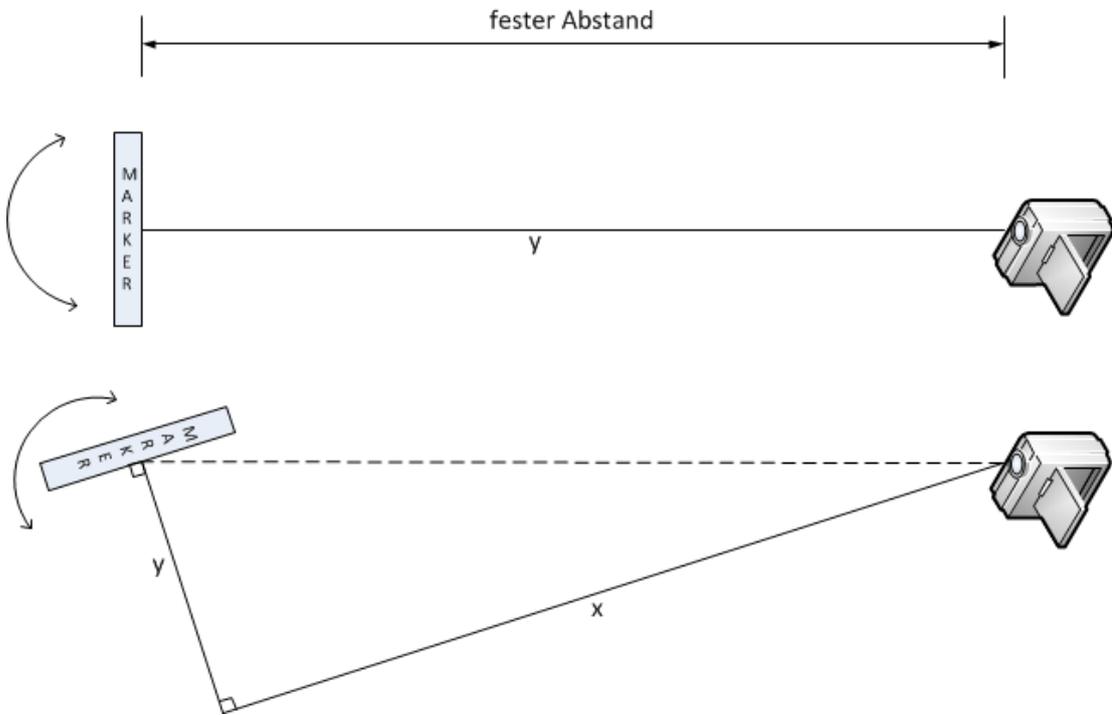


Abbildung 5.2: Versuchsaufbau

### 5.3.2 Auswertung der Messergebnisse

Die jeweiligen Positionsangaben beziehen sich hierbei auf das Markerkoordinatensystem und sind in Metern angegeben. In der Tabelle 5.1 sind die einzelnen Versuche dokumentiert. Die maximale Abweichung betrug ca. 28 cm und ist wie zu erwarten in dem 3-Meter-Messbereich zu finden. Interessanterweise liegt der Mittelwert über alle Abweichungen nur im Millimeterbereich, was sich mit einer guten Kamerakalibrierung erklären lässt.

## 5 Evaluierung

$\sphericalangle$	Entfernung	tatsächliche Pos.	ermittelte Pos.	absolute Abw.
90	1	(0,000; 1,000; 0,000)	( 0,018; 0,990;-0,020)	(-0,018; 0,010; 0,020)
80	1	(0,174; 0,985; 0,000)	( 0,090; 0,980;-0,090)	( 0,084; 0,005; 0,090)
75	1	(0,259; 0,966; 0,000)	( 0,355; 0,931;-0,020)	(-0,096; 0,035; 0,020)
45	1	(0,707; 0,707; 0,000)	( 0,662; 0,759; 0,030)	( 0,045;-0,052;-0,030)
25	1	(0,906; 0,423; 0,000)	( 0,897; 0,451;-0,015)	( 0,009;-0,028; 0,015)
90	2	(0,000; 2,000; 0,000)	(-0,013; 1,961;-0,143)	( 0,013; 0,039; 0,143)
80	2	(0,347; 1,970; 0,000)	( 0,245; 1,947; 0,025)	( 0,102; 0,023;-0,025)
75	2	(0,518; 1,932; 0,000)	( 0,716; 1,886;-0,130)	(-0,198; 0,046; 0,130)
45	2	(1,414; 1,414; 0,000)	( 1,443; 1,470; 0,052)	(-0,029;-0,056;-0,052)
25	2	(1,813; 0,845; 0,000)	( 1,863; 0,875; 0,005)	(-0,050;-0,030;-0,005)
90	3	(0,000; 3,000; 0,000)	( 0,002; 2,987; 0,115)	(-0,002; 0,013;-0,115)
80	3	(0,521; 2,954; 0,000)	( 0,241; 2,885; 0,069)	( 0,280; 0,069;-0,069)
75	3	(0,776; 2,898; 0,000)	( 1,031; 2,900; 0,238)	(-0,255;-0,002;-0,238)
45	3	(2,121; 2,121; 0,000)	( 2,098; 2,323;-0,045)	( 0,023;-0,202; 0,045)
25	3	(2,719; 1,268; 0,000)	( 2,816; 1,397;-0,013)	(-0,097;-0,129; 0,013)

Tabelle 5.1: Auswertung der Messwerte (alle Angaben in Meter)

Die Tabelle 5.2 zeigt die absolute Entfernung der Soll- und Ist-Positionen. Dabei wurde die Strecke zwischen den jeweiligen Punkten sowohl im 2D- als auch im 3D-Koordinatensystem berechnet. Wie zu erwarten war und in Abbildung 5.3 besonders gut zu erkennen ist, liegt der absolute Fehler in der ermittelten 3D-Position leicht über dessen 2D-Pendant, welches natürlich der zusätzlichen Fehlerquelle der Z-Achse geschuldet ist. Der Mittelwert aller 3D-Positionsabweichung liegt bei ca. 14 cm. Der Maximalwert bei 34 cm.

Winkel	Entfernung	2D	3D
90	1	0,021	0,029
80	1	0,084	0,123
75	1	0,102	0,104
45	1	0,069	0,075
25	1	0,030	0,033
90	2	0,041	0,149
80	2	0,105	0,108
75	2	0,204	0,242
45	2	0,063	0,082
25	2	0,059	0,059
90	3	0,013	0,116
80	3	0,288	0,297
75	3	0,255	0,348
45	3	0,203	0,208
25	3	0,162	0,162

Tabelle 5.2: Entfernung zwischen realer und gemessener Position (alle Angaben in Meter)

Die Abbildung 5.3 bereitet die ermittelten Messergebnisse noch einmal grafisch auf.

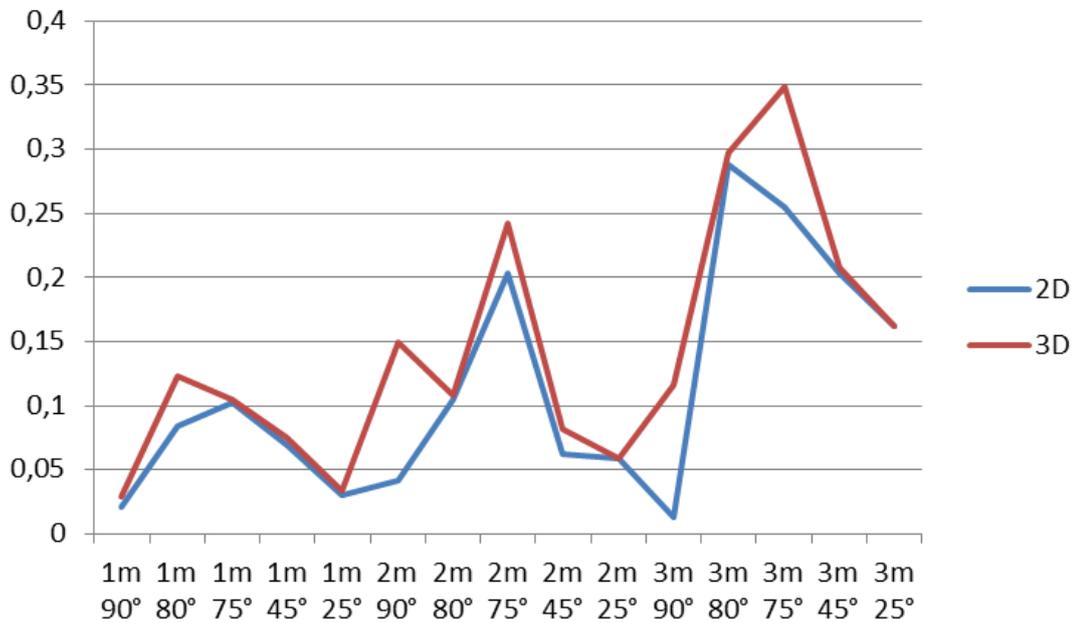


Abbildung 5.3: Darstellung der ermittelten Positionsabweichung

## 5.4 Zusammenfassung

In diesem Kapitel wurde gezeigt, dass das erstellte System sich gut in die bestehende Systemumgebung einbinden ließ. Weiterhin konnte gezeigt werden, dass unter Laborbedingungen eine sehr präzise Positionierung innerhalb des dreidimensionalen Raumes möglich ist und unter realen Bedingungen noch gute Ergebnisse zu erzielen sind, solange kein Gegenlicht die Kamera stört und die 2D-Marker ausreichend beleuchtet sind.

# 6 Schluss

## 6.1 Zusammenfassung

In dieser Arbeit wurde ein Prototyp zur visuellen Indoorlokalisierung entwickelt. Dabei wurden zunächst in Kapitel 2 verschiedene Ansätze zur Lokalisierung erfasst und bewertet. Anschließend wurde während der Analyse in Kapitel 3 die Anforderungen aus Benutzersicht identifiziert und die zu lösenden Teilaufgaben festgelegt. Im nachfolgenden Kapitel 4 wurden daraus ein Softwaredesign zur visuellen Positionsbestimmung erarbeitet und gezeigt, wie sich diese Lösung in eine bestehende Systemumgebung (das „Living Place“ der Hochschule für Angewandte Wissenschaften Hamburg) einbindet. Dabei wurde im Besonderen auf eine gute Erweiterbarkeit, Skalierbarkeit und Wartung der Anwendung geachtet und dargestellt, wie die Wahl des Architekturmusters diese beeinflusst. In Kapitel 5 „Evaluierung“ wurde die Funktionstüchtigkeit innerhalb einer realen Umgebung nachgewiesen und die erreichbare Genauigkeit unter Laborbedingungen ermittelt und ausgewertet.

## 6.2 Ausblick

Zum Schluss gibt es noch einen Ausblick auf mögliche Weiterentwicklungen. Diese lassen sich grob in zwei Kategorien einordnen: Verbesserung der Verarbeitungsgeschwindigkeit und Verbesserungen des Nutzwertes.

Da die Berechnung der Positionsdaten auf einem aktuellen Smartphone mit einer 1,2 GHz DualCore-CPU schon bei über 20 Hz liegt, ist die reine Steigerung der Verarbeitungsgeschwindigkeit nur noch in speziellen Anwendungsgebieten sinnvoll. Damit ist eine weiter wie in Kapitel 4 beschriebene Parallelisierung nur selten

erstrebenswert. Auf der anderen Seite führt eine Verbesserung der Ausführungsgeschwindigkeit der eingesetzten Algorithmen sowie die gleichzeitige Limitierung der einzelnen Positionsberechnung pro Sekunde zu einer erheblich längeren Akkulaufzeit oder ermöglicht die Auswertung größeren Bildmaterials, so dass auch weiter entfernte Marker noch besser erkannt werden können.

Zum Beispiel werden die Kameradaten vom Android-SDK im YUV-Format bereitgestellt. Um diese im (Ny)ARToolkit, welche das RGB-Format verwendet benutzen zu können, sind diese also zuvor zu konvertieren. Da ARToolkit aber nur Schwarz-Weiß-Marker verwendet, wäre es vollkommen ausreichend nur die Luminanz (Y-Kanal) des originalen Bildmaterials zu verwenden. Da diese Konvertierung fast die Hälfte der benötigten Rechenleistung in Anspruch nimmt, ist hierbei eine erhebliche Einsparung möglich.

Eine weitere Möglichkeit Rechenleistung einzusparen könnte darin bestehen, den Bildausschnitt bei der Markersuche einzuschränken. Zum Beispiel könnten die in der Vergangenheit erkannten Markerpositionen im Bildmaterial extrapoliert oder per Gyroskope neu bestimmt werden.

Der Nutzwert der Lokalisierungskomponente könnte durch ein Offline-Modus, in dem alle benötigten Daten vorzeitig heruntergeladen und gespeichert würden, erweitert werden.

Steht mit neuen Geräteversionen mehr Rechenleistung zur Verfügung, ließe sich die Akzeptanz der Benutzer durch den Verzicht der 2D-Marker unter Verwendung eines anderen Bildanalyseverfahrens wie SURF steigern.

Die Positionierungsgenauigkeit könnte durch die Verwendung höher auflösender Bildmaterials gesteigert werden. Eine weitere Möglichkeit dieses zu erreichen wäre bei der Erkennung mehrerer Marker diese zu einem potentiell sehr viel größeren virtuellen Marker zusammen zu fassen, indem die äußeren Ecken der jeweils einzelnen

Marker für die Berechnung der Winkel und Abstände herangezogen werden.

Um der Gegenlichtproblematik entgegen zu wirken, wäre die Verwendung von unter- bzw. überbelichteten Bildmaterial sinnvoll, wie sie auch zur Erzeugung von High Dynamic Range Images verwendet werden.

# Literaturverzeichnis

- [Bay u. a. 2008] BAY, Herbert ; ESS, Andreas ; TUYTELAARS, Tinne ; VAN GOOL, Luc: SURF: Speeded Up Robust Features. In: *Computer Vision and Image Understanding (CVIU)* 110 (2008), Juni, Nr. 3, S. 346–359. – URL <http://dx.doi.org/10.1016/j.cviu.2007.09.014>. – ISSN 10773142
- [Ellenberg u. a. 2011] ELLENBERG, Jens ; KARSTAEDT, Bastian ; VOSKUHL, Sören ; LUCK, Kai von ; WENDHOLT, Birgit: An Environment for Context-Aware Applications in Smart Homes. (2011)
- [Gamma u. a. 1996] GAMMA, Erich ; HELM, Richard ; JOHNSON, Ralph ; VLISSIDES, John: *Entwurfsmuster*. Addison Wesley Verlag, Juli 1996. – ISBN 3827318629
- [Gossow u. a. 2011] GOSSOW, David ; DECKER, Peter ; PAULUS, Dietrich: RoboCup 2010. (2011), S. 169–179. – URL <http://dl.acm.org/citation.cfm?id=1983806.1983822>. ISBN 978-3-642-20216-2
- [Hightower u. a. 2000] HIGHTOWER, Jeffrey ; BORRIELLO, Gaetano ; WANT, Roy: SpotON: An Indoor 3D Location Sensing Technology Based on RF Signal Strength. In: *UW CSE Technical Report 2000-02-02* (2000), Februar
- [Karstaedt 2012] KARSTAEDT, Bastian: *Kontextinterpretation in Smart Homes auf Basis semantischer 3D Gebäudemodelle*, HAW-Hamburg, Diplomarbeit, 2012
- [Karstaedt und Wendholt 2011] KARSTAEDT, Bastian ; WENDHOLT, Birgit: Towards semantic interpretations of spatial information in adaptive smart homes. (2011)
- [Lowe 2004] LOWE, David G.: Distinctive image features from scale-invariant keypoints. In: *International Journal of Computer Vision* (2004). – URL <http://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>

- [Mautz und Tilch 2011] MAUTZ, R. ; TILCH, S.: Survey of optical indoor positioning systems. In: *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, sept. 2011, S. 1 –7
- [McClendon 2011] MCCLENDON, Brian: A new frontier for Google Maps: mapping the indoors. (2011), November. – URL <http://googleblog.blogspot.de/2011/11/new-frontier-for-google-maps-mapping.html>
- [Mulloni u. a. 2009] MULLONI, A. ; WAGNER, D. ; BARAKONYI, I. ; SCHMALSTIEG, D.: Indoor Positioning and Navigation with Camera Phones. In: *Pervasive Computing, IEEE* 8 (2009), april-june, Nr. 2, S. 22 –31. – ISSN 1536-1268
- [Priyantha u. a. 2000] PRIYANTHA, Nissanka B. ; CHAKRABORTY, Anit ; BALAKRISHNAN, Hari: The Cricket location-support system. In: *Proceedings of the 6th annual international conference on Mobile computing and networking*. New York, NY, USA : ACM, 2000 (MobiCom '00), S. 32–43. – URL <http://doi.acm.org/10.1145/345910.345917>. – ISBN 1-58113-197-6
- [Reza u. a. 2011] REZA, D.S.H. ; MUTIJARSA, K. ; ADIPRAWITA, W.: Mobile robot localization using augmented reality landmark and fuzzy inference system. In: *Electrical Engineering and Informatics (ICEEI), 2011 International Conference on*, july 2011, S. 1 –6. – ISSN 2155-6822
- [Roth 2005] ROTH, Jörg: *Mobile Computing*. dpunkt.verlag, 2005
- [Saito u. a. 2007] SAITO, S. ; HIYAMA, A. ; TANIKAWA, T. ; HIROSE, M.: Indoor Marker-based Localization Using Coded Seamless Pattern for Interior Decoration. In: *Virtual Reality Conference, 2007. VR '07. IEEE*, march 2007, S. 67 –74
- [Serra u. a. 2010] SERRA, Alberto ; CARBONI, Davide ; MAROTTO, Valentina: Indoor pedestrian navigation system using a modern smartphone. In: *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*. New York, NY, USA : ACM, 2010 (MobileHCI '10), S. 397–398. – URL <http://doi.acm.org/10.1145/1851600.1851683>. – ISBN 978-1-60558-835-3

- [SkyTrax ] : *The Sky-Trax System™*. – URL <http://totaltraxinc.com/index.php/smart-forklift-solutions/forklift-tracking/sky-trax>
- [V. Schmude 2011] V. SCHMUDE, Naja: Farb- und Kantenbasierte Objekterkennung Humanoider Roboter im RoboCup-Szenario. (2011), November.  
– URL [http://www.fumanoids.de/wp-content/uploads/2011/11/Diplomarbeit\\_Schmude.pdf](http://www.fumanoids.de/wp-content/uploads/2011/11/Diplomarbeit_Schmude.pdf)
- [WIPS ] : *WIPS - Wireless Indoor Positioning System*. – URL <http://www.online.kth.se/csd/projects/0012/>

*Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.*

Hamburg, 29. Juni 2012

---

Sascha Möhlmann