



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

Philipp Teske

Ein Multisensor-System
zur Sturzerkennung

Philipp Teske

Ein Multisensor-System zur Sturzerkennung

Masterarbeit eingereicht im Rahmen der Masterprüfung
im Studiengang Master Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. rer. nat. Gunter Klemke
Zweitgutachter: Prof. Dr. rer. nat. Kai von Luck

Abgegeben am 30. Januar 2012

Philipp Teske

Thema der Masterarbeit

Ein Multisensor-System zur Sturzerkennung

Stichworte

Ambient-Assisted-Living, kapazitive Sensoren, Erschütterungssensoren, Bildverarbeitung, Bildanalyse, Multisensor, Living Place, Sturzerkennung

Kurzzusammenfassung

Diese Arbeit befasst sich mit der Entwicklung eines kamera- und sensorbasierten Verfahrens zur kontaktlosen Erkennung von gestürzten und hilflos am Boden liegenden Personen. Für die Detektion wurde ein auf mehreren Rechnern verteiltes System entworfen und in die Software-Infrastruktur des Living Place Hamburg integriert. Im Rahmen dieser Arbeit wurden ebenfalls die hierfür benötigten Sensoren entwickelt. Die Interpretationsergebnisse des Systems werden über den zentralen ActiveMQ-Server des Living Places anderen Projekten verfügbar gemacht.

Philipp Teske

Title of the paper

A multisensor system for fall detection

Keywords

Ambient-Assisted-Living, capacitive sensors, shocksensor, image processing, image analysis, multisensor, living place, fall detection

Abstract

This master thesis deals with the design of a camera and sensor based method for contactless detection of fallen and helpless people. As a part of the Living Place Hamburg infrastructure a distributed software system was developed. The development of the required sensors is also described in this thesis. The interpretation results will be provided to other projects by the central ActiveMQ-Server in the living place.

Inhaltsverzeichnis

1	Einführung	3
1.1	Motivation	3
1.2	Zielsetzung	3
1.3	Überblick	4
2	Grundlagen	5
2.1	Living Place Hamburg	5
2.2	Kapazitive Sensorik	6
2.3	Seismische Messverfahren	7
3	Analyse	9
3.1	Anwendungsszenarien	9
3.2	Anforderungen und Ziele	11
3.3	Existierende Ansätze	12
3.3.1	Kommerzielle Ansätze.....	12
3.3.2	Forschungsprojekte.....	16
3.3.3	Fazit.....	21
3.4	Auswahl geeigneter Messverfahren	23
3.4.1	Erschütterungssensoren.....	24
3.4.2	Kapazitive Sensoren.....	28
3.4.3	Auswertung der Messverfahren.....	36
3.5	Prototyp der Sensorfläche	37
3.6	Software-Bibliotheken zur Bildverarbeitung	38
3.6.1	Resultat.....	40
4	Entwicklung eines kapazitiven Sensors	41
4.1	Technische Umsetzung	41
4.1.1	Stromversorgung.....	41
4.1.2	Mikrocontroller.....	42
4.1.3	RS-485 Bus.....	42
4.1.4	Sensor.....	44
4.1.5	Guard.....	44
4.2	Kommunikationsprotokoll	46
4.3	Software	47
4.3.1	Bootloader.....	47
4.3.2	Firmware.....	48
4.4	Evaluierung	51
5	Konzept und Architektur der Software	54
5.1	Aufbau der Software	54
5.2	Systembeschreibung	54
5.3	Ablaufsteuerung	60
5.4	Verarbeitung der Sensordaten	61
5.5	Bildanalyse	66
5.6	Risiken und Probleme	69
6	Technische Umsetzung und Implementation	71
6.1	Aufbau des Sensorfeldes	71
6.2	Implementation der Software	72

7	Evaluierung	78
8	Fazit	80
8.1	Zusammenfassung	80
8.2	Ausblick	80
	Abbildungsverzeichnis	82
	Tabellenverzeichnis	84
	Listings	85
	Literaturverzeichnis	86
A	Schaltplan des Sensors	90
B	Befehlsübersicht des Sensors	91
C	Klassendiagramme	95
C.1	SettingsFileHandler	95
C.2	ConnectionAdapter	96
C.3	CapacitiveSensors	97
C.4	ActiveMQConnection	98
C.5	GUI	99
C.6	FallDetection	100
C.7	Sensorfeld-Server	101

1 Einführung

1.1 Motivation

Demographische Untersuchungen zeigen, dass die Geburtenrate in Deutschland seit Mitte der 90er Jahre kontinuierlich sinkt [58]. Da sich durch gute medizinische Versorgung die Lebenserwartung jedoch in gleichem Maße erhöht, führt dies zu einem steigenden Altersdurchschnitt in der Bevölkerung. Durch die zunehmende Lebenserwartung bei gleichzeitiger sinkender Kinderzahl gerät die gesamte soziale Versorgungsstruktur, d.h. Kranken-, Pflege-, und Rentenversicherungssystem unter enormen Druck.

Viele alte Menschen wollen ihren Lebensabend nicht in einem Seniorenheim verbringen, sondern sich in ihrem häuslichen Umfeld pflegen lassen. Die ambulante Pflege gewährt ihnen einerseits einen gewissen Grad an Selbstständigkeit, bedeutet aber auch eine erhöhte Unfallgefahr bei Nichtanwesenheit des Pflegepersonals.

In Deutschland stürzen jährlich über 100.000 ältere Menschen. Oftmals kommt die Person mit einem Schrecken davon, nicht selten jedoch führt ein Sturz zu schweren Verletzungen. Häufig bedeutet dies lange Genesungszeiten und Krankenhausaufenthalte. Hinzukommend ist besonders bei älteren Personen festzustellen, dass sie sich aus Angst erneut zu stürzen, vorsichtiger und weniger bewegen als vorher. Dies hat zur Folge, dass die Muskelkraft nachlässt, sie somit unsicherer in ihren Bewegungen werden und es, dadurch bedingt, zu weiteren Stürzen kommt. Da Stürze in den meisten Fällen aus einer Verkettung mehrerer Faktoren bestehen, lassen sie sich nur schwer verhindern. Neben der Bemühung um eine vollständige Genesung und Stärkung der Mobilität, brauchen insbesondere allein lebende Personen die Sicherheit, im Notfall schnellstmöglich Hilfe zu bekommen.

1.2 Zielsetzung

Ziel dieser Masterarbeit ist die Entwicklung eines Systems, welches mit Hilfe verschiedener Sensoren in der Lage ist, Bewegungen zu analysieren um Stürze und Unfälle von Personen zu erkennen und, wenn nötig, Hilfe anzufordern. Dies umfasst sowohl die Implementierung einer entsprechenden Software wie auch gegebenenfalls die Entwicklung spezieller Sensoren. Damit das System nicht als störend empfunden wird, soll es sich nahtlos in bestehende Wohnumgebungen integrieren lassen und die Bewegung eines Menschen kontaktlos erfassen,

erkennen und interpretieren können. Ein erster Aufbau soll im Living Place Hamburg entstehen und in die dort vorhandene Infrastruktur integriert werden. Dies umfasst vor allem die Bereitstellung der durch Interpretation erlangten Informationen für andere am Living Place beteiligten Projekte. Als Anforderungen an das System gelten eine einfache Bedienung, Konfigurierbarkeit sowie eine leichte und kostengünstige Integration in bestehende Wohnbereiche.

1.3 Überblick

Zunächst werden Grundlagen, welche zum tieferen Verständnis der Arbeit beitragen, erläutert. Dies beinhaltet eine Einführung ins Living Place Hamburg in Abschnitt 2.1 sowie Grundlagen in kapazitiver Messtechnik (Kapitel 2.2) und Verfahren zur Messung von seismischen Schwingungen (Kapitel 2.3).

Im 3. Kapitel werden die Anforderungen des Systems festgelegt und die Laborausstattung diskutiert. Dies umfasst neben der Festlegung der Rahmenbedingungen mit Hilfe zweier Anwendungsszenarien in Kapitel 3.1 auch die Vorstellung mehrerer Arbeiten und Projekte aus kommerziellen (Kapitel 3.3.1) und wissenschaftlichen (Kapitel 3.3.2) Bereichen. Im Fokus stehen hierbei Arbeiten, die von der verwendeten Technik und Verfahrensweise her vergleichbar mit dem in dieser Arbeit entwickelten System sind. Weiterhin werden Versuche mit verschiedenen Sensoren durchgeführt (Kapitel 3.4) sowie mehrere Softwarebibliotheken zur Bildverarbeitung und -analyse evaluiert (Kapitel 3.6). Aufbauend auf die in diesem Kapitel erlangten Erkenntnisse wird in Kapitel 4 die Realisierung eines kapazitiven Sensors für das Living Place beschrieben. Dies umfasst sowohl die technische Umsetzung wie auch die Entwicklung passender Software.

Im Anschluss hieran wird in Kapitel 5 das System zur Sturzerkennung näher beschrieben. Kernthema hierbei ist der strukturelle Aufbau und Ablauf der Software sowie die Verarbeitung von Bild- und Sensordaten. Kapitel 6 behandelt den technischen Aufbau des Systems sowie ausgewählte Merkmale der Implementation. Abschließend werden in Kapitel 8 der Verlauf dieser Arbeit noch einmal zusammengefasst sowie verschiedene Optimierungs- und Erweiterungsmöglichkeiten vorgestellt.

2 Grundlagen

2.1 Living Place Hamburg

Am Campus Berliner Tor entsteht seit Anfang 2009 in einem Maschinenbaugebäude der Hochschule für Angewandte Wissenschaften Hamburg der *Living Place Hamburg*. Bei diesem handelt es sich um ein ca. 140qm großes, voll ausgestattetes Apartment in Stil eines Lofts. Ziel des Living Place ist die Realisierung einer intelligenten, dienstorientierten Wohnung für Forschungen im Bereich des *Ambient Assisted Living* und *Smart Living* [53]. Vorarbeiten und Projekte, die im Rahmen des IFlat [59] entstanden sind, sowie zukünftige Technologien können im Living Place unter realitätsnahen Bedingungen erprobt und bewertet werden. Für Usability-Untersuchungen ist der gesamte Wohnbereich mit mehreren Kameras und Mikrofonen ausgestattet. Aus einer Steuerzentrale heraus kann somit jede Bewegung im Living Place überwacht und ausgewertet werden.



Abbildung 1: Modell des Living Place Hamburg
(Quelle: www.livingplace.org)



Abbildung 2: Schlafbereich



Abbildung 4: Wohnbereich



Abbildung 3: Küche mit Multi-Touch-Tresen

2.2 Kapazitive Sensorik

Werden zwei elektrisch leitfähige Platten in geringem Abstand zueinander positioniert, ohne sich jedoch zu berühren, bilden sie einen Kondensator. Sie sind in der Lage, geringe Mengen elektrischer Energie zu speichern und wieder abzugeben. Eine der ältesten Bauformen des Kondensators ist die sogenannte Leidener Flasche (Abbildung 5). Sie bestand aus einem Glasgefäß, auf dessen Innen- und Außenseite Metallbeläge angebracht sind und hatte eine Kapazität von wenigen Nanofarad (nF) [7], [1]. Durch das Anlegen einer positiven sowie einer negativen Spannung an je eine der Platten baut sich ein elektrisches Feld zwischen diesen auf. Dieses bleibt auch nach Abschalten der Spannung erhalten. Wenn anschließend die beiden Elektroden durch ein Kabel

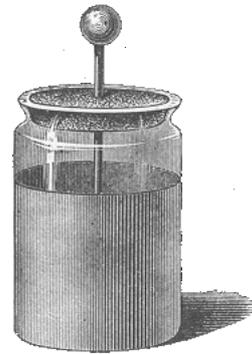


Abbildung 5: Leidener Flasche
(Quelle: wikipedia.de)

miteinander verbunden werden, entlädt sich die gespeicherte Energie in Form eines elektrischer Stroms durch das Kabel, bis das Feld abgebaut ist. Dabei ist die Menge der gespeicherten Energie direkt abhängig von der Größe der Kondensatorplatten [25]. Durch moderne Fertigungstechniken ist es heutzutage möglich, Kondensatoren herzustellen, welche die Kapazität der Leidener Flasche, bei nur einem Bruchteil der Größe, um ein vielfaches übersteigt. Dennoch hat sich an dem Funktionsprinzip bis heute nichts geändert.

Die oben beschriebenen Eigenschaften eines Kondensators lassen sich nutzen, um berührungslos Abstände und Bewegungen von Objekten zu messen. Dazu wird eine der Platten durch das zu messende Objekt ersetzt. Voraussetzung hierbei ist jedoch, dass das Objekt aus einem nicht-isolierendem Material besteht, d.h. es eignen sich Gegenstände aus Metall, ebenso Holz und Lebewesen, da diese größere Mengen leitfähiges Wasser enthalten. Damit das zu messende Objekt nicht unter Spannung gesetzt werden muss, wird dieses üblicherweise mit dem negativ geladenen Erdboden verbunden und die verbliebene Kondensatorplatte mit einer, dem Erdboden gegenüber, positiven Ladung aufgeladen. Durch Verringern oder Vergrößern des Abstandes zwischen Objekt und Elektrode lässt sich nun die Kapazität des gebildeten Kondensators beeinflussen [25].

Dieses Verfahren machen sich kapazitive Messsysteme zu nutzen, bei denen es nicht auf die exakt gemessene Kapazität ankommt, sondern lediglich die Änderung über einen bestimmten Zeitraum von Interesse ist. Da nichtleitende Materialien das elektrische Feld nur minimal beeinflussen, ist es möglich, diese Systeme in Gerätegehäusen oder Wänden zu integrieren. So eignen sich kapazitive Näherungssensoren besonders für Arbeitsbereiche, in denen es auf erhöhte Hygiene ankommt, z.B. als Schalter in Krankenhäusern. Aufgrund ihrer geringen Abmessungen und der Möglichkeit, bereits kleinste kapazitive

Änderungen detektieren zu können, eignen sich die Systeme zudem als Füllstandssensoren in der Industrie, wo u.a. aufgrund von Messgeschwindigkeit oder Verschleiß keine mechanische Messung möglich ist. Kapazitive Messsysteme haben jedoch den Nachteil, dass die Messungen durch das elektrische Feld eines in der Nähe stehenden elektrischen Gerätes gestört werden können. Ebenso beeinflussen Änderungen der Umweltbedingungen wie Luftfeuchtigkeit und Temperatur die Messungen und bewirken ein Verringern der Genauigkeit. In Einsatzbereichen, bei denen es auf eine hohe Genauigkeit der Messwerte ankommt, ist es somit unabdingbar, das System vor einer Messung zu kalibrieren, um Abweichungen durch Umwelteinflüsse zu minimieren.

2.3 Seismische Messverfahren

Die einfachste Art, Erschütterungen zu messen, ist mit Hilfe einer seismischen Masse, welche an einer Feder oder einem Pendel befestigt ist und bei Erschütterungen in Schwingung gerät (Abbildung 6). Diese rein mechanischen Seismographen wurden bis Anfang des 20. Jahrhunderts eingesetzt und werden heutzutage nur noch als Anschauungs- und Forschungsobjekt verwendet. Heutige Erschütterungssensoren verwenden zwar immer noch eine seismische Masse, sind jedoch dank moderner Mikrotechnik wesentlich kleiner und empfindlicher geworden. Eine Art dieser Sensoren sind *Mikro-Vibrationsschalter*. Diese bestehen in ihrem Inneren aus einer frei beweglichen Kugel, welche bei Erschütterungen zwei Kontakte miteinander überbrückt oder unterbricht [12]. Die daraus resultierenden Impulse können durch weitere Elektronik aufgezeichnet und analysiert werden.

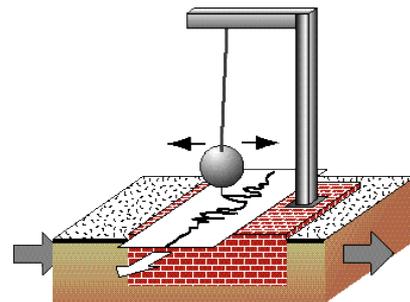


Abbildung 6: Seismograph
(Quelle: slohs.slcsud.org)

Eine andere Technik basiert auf dem piezoelektrischen Effekt. Durch Anlegen einer Spannung lassen sich bestimmte Materialien verformen. Umgekehrt führt dieser Effekt zur Entstehung einer Spannung bei Druck oder Verformung dieser piezoelektrischen Materialien. Das bekannteste Material mit diesen Eigenschaften ist Quarz.

Quarkristalle dienen heutzutage nahezu in jeder Uhr und elektronischer Schaltung als Taktgeber. Jedoch weisen auch keramische Werkstoffe piezoelektrische Eigenschaften auf. Diese findet man häufig in ferroelektrischen Lautsprechern (Abbildung 7). Durch Anlegen



Abbildung 7: Ferroelektrischer Lautsprecher

einer elektrischen Wechselspannung an das piezoelektrisch Material verändert dieses seine Form. Die entstehende Bewegung wird auf eine Membrane übertragen, welche diese in Schallwellen umsetzt. Umgangssprachlich werden diese Lautsprecher daher auch als Piezoschallwandler bezeichnet [48].

Auf umgekehrte Weise kann man sich den oben beschriebenen Effekt zu Nutze machen, um Druckänderungen auf dem Schallwandler zu detektieren.

3 Analyse

Im Rahmen dieser Arbeit soll ein System entwickelt werden, welches mit Hilfe verschiedener Sensoren in der Lage ist, gestürzte und hilflose Personen zu erkennen und eigenständig Hilfe anzufordern. Das System soll sich nahtlos in bestehende Wohnumgebungen integrieren lassen und Bewegungen von Menschen kontaktlos erfassen und interpretieren können. Ein erster Aufbau soll im Living Place Hamburg entstehen und in die dort vorhandene Infrastruktur integriert werden.

In diesem Kapitel wird erarbeitet, wie das System aussehen und was es leisten soll. Hierfür werden zunächst zwei Anwendungsszenarien vorgestellt, anhand derer die Anforderungen und Ziele des Systems festgelegt werden. Den Hauptteil des Kapitels bildet die Vorstellung bereits existierender Systeme und aktuelle Forschungsberichte, sowie die Analyse verschiedener Messverfahren.

3.1 Anwendungsszenarien

Die Anforderungen an ein System zu Erkennung von Stürzen lassen sich am einfachsten mit Hilfe einiger Beispiele verdeutlichen. In den kommenden Abschnitten sollen zwei fiktive Einsatzszenarien beschrieben werden, in denen der Einsatz dieser Systeme deutlich wird.

Beide Szenarien spielen in der Wohnung einer allein lebenden, älteren Dame. Da sie sich auf Grund ihres Alters bereits ein wenig unsicher fühlt, hat sie ein Notfallsystem in ihrer Wohnung installieren lassen. Dieses besteht aus mehreren, in den Räumen verteilten Sensoren sowie einem Notfallknopf, der wie ein Armband getragen werden kann. Das System hat die Aufgabe, in einer Notsituation die Tochter anzurufen und eine Sprachverbindung herzustellen.

Da das Armband die Dame nachts beim Schlafen stört, legt sie es jeden Abend auf ihren Nachttisch und bindet es morgens wieder um. Eines Tages wacht sie früh Morgens mit einem Kratzen im Hals auf. Draußen setzt gerade erst die Morgendämmerung ein. Sie beschließt in die Küche zu gehen, einen Schluck Wasser zu trinken und sich anschließen wieder hinzulegen. Abgelenkt durch die Dunkelheit und die ungewohnte Uhrzeit, vergisst sie, sich das Notfall-Armband umzulegen. Auf dem Weg zur Küche durchquert sie das Wohnzimmer, in dem ihr alter Ohrensessel mit passendem Hocker steht. In dem Dämmerlicht übersieht sie den Hocker und bleibt mit einem Fuß hängen. Sie verliert das Gleichgewicht und stürzt zu Boden. Durch den heftigen Aufprall wird sie sofort bewusstlos. Als sie wieder zu sich kommt, stehen ihre Tochter und zwei Personen in roter Kleidung über ihr.

Was war passiert? Die Sensoren des Notfallsystems erkannten, dass jemand auf dem Boden lag und sich nicht mehr bewegt hat. Nachdem auch keine weiteren Personen in der Wohnung ausgemacht werden konnten, wurde über die Freisprecheinrichtung des Telefons eine Sprachverbindung zum Mobiltelefon der Tochter der Dame aufgebaut, welche gerade auf dem Weg zur Arbeit war. Nachdem diese merkte, dass ihre Mutter am Telefon nicht antwortete, bestellte sie sofort einen Krankenwagen und fuhr zur Wohnung.

Dieses Szenario zeigt, wie ein entsprechendes System in einem Notfall selbstständig Hilfe anfordern kann. In einem weiteren Szenario sollen nun die Problematiken bei der Erkennung von Stürzen verdeutlicht werden.

Einige Wochen nach dem Vorfall hat sich die Dame von der Gehirnerschütterung, welche sie sich bei dem Sturz zugezogen hatte, fast vollständig erholt. Die Tochter ist zu Besuch und hat ihren Hund und ihren kleinen Sohn mitgebracht, da sich niemand gefunden hat, der auf beide aufpassen konnte. Während die beiden Erwachsenen im Esszimmer bei einer Tasse Tee zusammensitzen - der Hund liegt in einer Ecke neben ihnen – spielt das Kind im Wohnzimmer mit seinen Spielsachen. Nach einiger Zeit steht der Hund auf und trottet ins Wohnzimmer zu dem Kind. Die Dame und ihre Tochter bemerken dies nicht. Kurze Zeit später klingelt das Handy der Tochter. Der Anruf, so zeigt ein Blick auf das Telefon, kommt angeblich von ihrer Mutter, welche ihr jedoch gegenüber am Tisch sitzt. Verblüfft stehen beide auf und gehen ins Wohnzimmer, in dem das Telefon der Dame steht, da sie vermuten, dass das Kind damit gespielt hat. Als sie den Raum betreten, sitzt das Kind jedoch auf dem Teppich und streichelt den Hund, welcher sich neben ihm lang ausgestreckt hat .

Anhand dieses Szenarios wird ersichtlich, wie es zu Fehlmeldungen kommen kann. Solange das Kind alleine in dem Raum war, wurde es von den Sensoren des Systems nicht als am Boden liegende Person erkannt. Als jedoch der Hund hinzukam und sich neben das Kind legte, wurde dieser irrtümlicher Weise als Person eingestuft. Da sich niemand sonst in dem Raum befand und auch die beiden am Tisch sitzenden Frauen nicht als Menschen erkannt wurden, ging das System von einem Unfall aus und versuchte, Hilfe zu organisieren.

Dieses Szenario zeigt die Problematik, die durch nicht eindeutige Interpretation der Sensorsignale entstehen kann. Zum Einen kann es, wie im obigen Fall, zu falschen Alarmmeldungen kommen, zum Anderen kann es jedoch auch passieren, dass eine hilflose Person überhaupt nicht erkannt wird. Hier gilt es, die Anforderungen an das System genau abzugrenzen und ggf. mit weiteren Systemen zu kombinieren. So lässt sich beispielsweise ein Sturzerkennungssystem mit einem intelligenten Bett [13] oder Sofa [49],[2] verbinden, um auf diese Weise eine genauere Einschätzung der Situationen zu ermöglichen.

3.2 Anforderungen und Ziele

Mit Hilfe der beiden Anwendungsszenarien lassen sich die Anforderungen an das System konkretisieren. Ausgangspunkt beider Szenarien war, dass eine am Boden liegende Person erkannt wurde. Dies lässt sich am effizientesten durch Sensoren erreichen, die im oder unter dem Boden angebracht sind. Kapitel 3.4 befasst sich hierfür mit der Auswahl eines geeigneten Messverfahrens. Durch den ständigen Ausbau des Living Places, indem das System zum Einsatz kommen wird, ist eine gewisse Mobilität der Installation nötig. Aus diesem Grunde soll ein Teppich zur Aufnahme der Sensoren im Boden dienen (siehe Kapitel 3.5). Dieser lässt sich bei Bedarf entfernen oder an einem anderen Ort verlegen. Somit ist eine Installation des Systems an einem beliebigen Ort gewährleistet.

Gleichzeitig muss jedoch auch auf ein optimales Kosten-Nutzen-Verhältnis geachtet werden. Ein System mit Sensoren, das für Privatpersonen nicht erschwinglich ist, kann zwar in Laborversuchen hervorragende Ergebnisse liefern, wird sich jedoch in einer kommerzieller Installation nicht durchsetzen.

Zusätzlich zu der am Boden liegenden Person muss detektiert werden können, ob sich die Person noch bewegt oder ob weitere Menschen im Raum sind, um der Person zu Hilfe zu kommen. Da sich die Sensoren im Boden nur bedingt für diese Aufgabe eignen werden, sollen Kameras diese Arbeit übernehmen.

Im Living Place sind zwei Kamertypen im Einsatz. Für Nahaufnahmen und Bildausschnitte kommen sogenannte *Pan-Tilt-Zoom-Kameras* (PTZ-Kameras) zum Einsatz. Bei diesen handelt es sich um hochauflösende Kameras, welche zur Fokussierung auf einzelne Objekte über eine Schwenk- und Zoomfunktion verfügen. Zusätzlich sind an der Decke des Living Places drei Dome-Kameras installiert. Mit ihrer speziellen Optik sind diese Kameras in der Lage einen Bereich von 360° zu überwachen (Abbildung 8). Durch eine geschickte Positionierung der Kameras ist somit eine lückenlose Überwachung des Wohnbereichs im Living Places gewährleistet. Für das in dieser



Abbildung 8: Aufnahme einer Dome-Kamera

Arbeit entwickelte System kommen ausschließlich die Dome-Kameras zum Einsatz, da es durch Ihre Optik möglich ist, mit nur einer Kamera einen großen Bereich zu erfassen. Da die PTZ-Kameras über einen eingeschränkten Blickwinkel verfügen, müssten mehrere von ihnen installiert werden, um die gleiche Abdeckung wie eine Dome-Kamera zu erreichen. Durch den Einsatz der Dome-Kameras bleiben die PTZ-Kameras für Anwendungen frei, welche auf die Zoom- und Schwenkfunktionen angewiesen sind, wie beispielsweise die Lokalisierung von einzelnen Objekten (siehe [24]). Um die Arbeit mit den

Kameras zu erleichtern, soll ein bereits existierendes Framework eingesetzt werden, hierfür werden in Kapitel 3.6 mehrere Bildverarbeitungs-Frameworks evaluiert.

Zusammenfassend lassen sich folgende Anforderungen an das zu entwickelnde System festsetzen:

- Einfache Integration in bestehende Umgebungen.
- Kostengünstiger Aufbau.
- Berührungslose Detektion von Stürzen.
- Leichte Konfigurierbarkeit und Wartbarkeit.

3.3 Existierende Ansätze

Anbieter sogenannter Hausnotrufdienste bieten heutzutage ein breites Sortiment zur Überwachung älterer oder kranker Personen. Die verfügbaren Systeme reichen von einfachen Knöpfen oder Kordeln, an denen im Notfall gezogen werden kann, bis zu Wassermeldern und Geräten, welche automatisch den Herd beim Verlassen der Küche abschalten [23]. Die meisten dieser Systeme stellen im Notfall selbstständig eine Sprachverbindung zu einem Service-Center des jeweiligen Anbieters her, über den Angehörige oder das Pflegepersonal informiert werden kann [52] [55].

Die folgenden beiden Kapitel stellen verschiedene Systeme zur Erkennung von gestürzten oder hilflosen Personen vor. Dies umfasst sowohl kommerzielle Produkte verschiedener Hersteller, wie auch aktuelle Forschungsarbeiten.

3.3.1 Kommerzielle Ansätze

QuietCare®

QuietCare®, ein Produkt der Firma *Living Independently*, stellt ein Frühwarnsystem dar und soll älteren Personen ein unabhängigeres und sicheres Leben in ihrer Wohnung ermöglichen. Hierfür werden an bestimmten, besonders gefährdeten Punkten in der Wohnung kleine, unscheinbare Bewegungsmelder installiert. Diese registrieren kontinuierlich die stattfindenden Bewegungen und übertragen die Daten per Funk an eine Basisstation. Von hier aus werden die gesammelten Daten in regelmäßigen Abständen (üblicherweise alle 2 Stunden) über die Telefonleitung an die Server von *QuietCare®* übertragen. Dort werden die aufgezeichneten Bewegungsdaten ausgewertet. Hierbei passt sich das System mit der Zeit automatisch den täglichen Bewegungsgewohnheiten der Person an,

damit auch Veränderungen im Verhalten nicht zu Falschmeldungen führen.

Sollte dennoch eine Abweichung vom normalen Tagesablauf und des Bewegungsrythmus der Person erkannt worden sein, werden Angehörige oder das Pflegepersonal über eine vorher eingespeicherte Telefonnummer informiert [55]. Zudem wird die Art des Ereignisses zusammen mit einem Zeitstempel und dem Ort aufgezeichnet und auf einer Website zur Verfügung gestellt. Dies erlaubt eine Auswertung des Verhaltens der Person auch über mehrere Wochen hinweg. Abbildung 9 verdeutlicht den Aufbau des QuietCare-Systems.

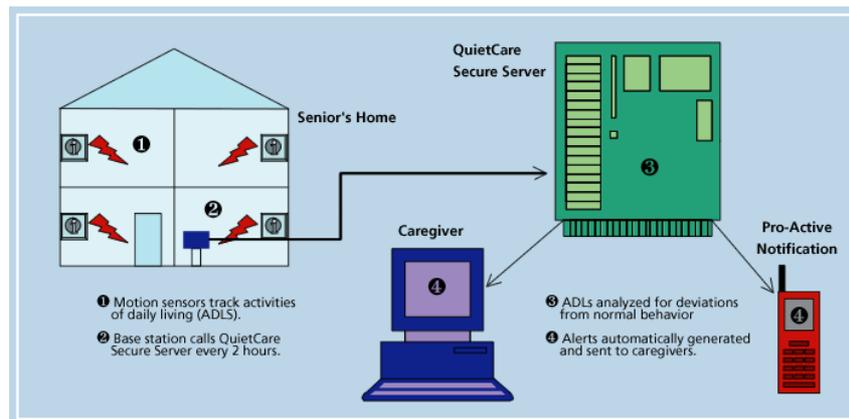


Abbildung 9: QuietCare Aufbau
(Quelle: [55])

MotionSENS®

MotionSENS® ist ein vom Fraunhofer IIS (Institut für Integrierte Schaltungen) entwickeltes System. Es wurde für die Verwendung in Wohnanlagen, Seniorenheimen oder bei alleinlebende Personen entwickelt. Das Gerät besteht aus einem Beschleunigungssensor und einer Einheit zur Auswertung. Durch seine geringe Dicke von nur ca. 4 mm lässt es sich ähnlich wie ein Pflaster am Körper tragen oder kann in Kleidungsstücke eingenäht werden. Das System erlaubt es, die Körperhaltung des Trägers zu erkennen und somit Rückschlüsse auf die Lage (stehend oder liegend) der Person zu ziehen. Zudem ist es mit Hilfe des Sensors möglich, Stürze zu erkennen und automatisch Alarm auszulösen. Für die Kommunikation mit einer stationären Basiseinheit steht eine Funkverbindung wahlweise über ZigBee oder Bluetooth zur Verfügung. Eine Batterieladung reicht je nach verwendetem Funkstandard zwischen max. 36 Std. (Bluetooth) und 72 Std. (ZigBee). Somit ist es eher für einen kurzzeitigen Einsatz geeignet.

Da die Einhaltung des Medizinproduktgesetzes und der europäischen Richtlinie 93/42/EWG bisher noch nicht nachgewiesen wurde, ist *MotionSENS®* noch nicht öffentlich erhältlich [15].

PHILIPS Lifeline

Bei den *Lifeline* Produkten von Philips handelt es sich um verschiedene Notfallknöpfe und deren Empfänger. Der Patient kann wählen, ob er den Knopf wie eine Kette oder wie eine Uhr am Handgelenk tragen möchte. Alle erhältlichen Versionen sind wasserdicht, um sie auch im Badezimmer, wo die meisten Unfälle geschehen, tragen zu können.

Im Falle eines Sturzes oder einer anderen Situation, in der Hilfe benötigt wird, kann durch einen Druck auf den Knopf von der Basiseinheit aus eine Sprachverbindung zum *Lifeline* Notcenter aufgebaut werden. Die Mitarbeiter dort können je nach Bedarf ein Familienmitglied, Pflegedienst oder andere Person informieren. Anfang 2010 wurde die Serie der *Lifeline* Buttons um einen Anhänger mit sogenannter *AutoAlert-Option* erweitert [51]. Dieser kann automatisch, d.h. wenn eine Person nach einem Sturz nicht mehr in der Lage ist, eigenständig den Knopf zu drücken, Alarm auslösen.



Abbildung 10: Lifeline Button mit AutoAlert
(Quelle: [52])

SensFloor®

Der Name *SensFloor*® steht für ein vom Bundesministerium für Bildung und Forschung gefördertes Verbundprojekt mehrerer namenhafter Firmen und hatte zum Ziel, einen intelligenten, universell einsetzbaren Fußboden zu entwickeln [18]. Dieser Bodenbelag besitzt eine Vielzahl von integrierten Sensoren mit denen Position und Bewegung von Personen erkannt und analysiert werden können. Erste Produkte dieser Technologie werden von der Firma *Future Shape GmbH* [17] in



Abbildung 11: SensFloor Matte
(Quelle: [19])

Form eines 2mm dicken, textil basierten Underlays vertrieben. Aufgrund seiner geringen Dicke kann es praktisch völlig unsichtbar unter jeder Art von Bodenbelag verlegt werden. Pro Quadratmeter ist es mit 32 kapazitiven Näherungssensoren ausgestattet, dessen Daten per Funk an die zentrale Steuereinheit übertragen werden. Aufgrund der hohen Sensordichte lassen sich *SensFloor*® Underlays sowohl in Pflegebereichen wie auch für Zutrittskontrollen, Gebäudeautomatisierung oder Entertainmentanwendungen einsetzen. Je nach Bedarf lassen sich *SensFloor*® Underlays auf 100cm breiten Rollen oder in Form

von verschieden großen Matten herstellen. Mögliche Anwendungen für Matten sind als Sturzsensormatte, Trittsensormatte oder Bettsensormatte (SensBed®) [19].

Praktische Einsatzgebiete für Sensfloor® werden aktuell in einem Seniorenzentrum in Höhenkirchen erforscht. Hier werden die Bedürfnisse von älteren Menschen, die alleine zu Hause oder in Pflegeheimen leben, untersucht. Des Weiteren wird ermittelt, wie viel länger eine Person durch SensorFloor® sicher zu Hause leben kann. Ein weiteres Szenario wird in der Meyer Werft GmbH in Papenburg evaluiert. Hier werden Passagierkabinen von Schiffen mit dem Underlay ausgestattet.

3.3.2 Forschungsprojekte

Eine zuverlässige Erkennung und Klassifizierung von Bewegungen und insbesondere von Stürzen war bereits Ziel mehrerer Forschungsprojekte [39]. Die heutzutage erhältlichen Systeme basieren fast ausschließlich auf einem am Körper getragenen Sensor, welcher auch Ausgangspunkt vieler Forschungsberichte ist. Ein einfaches System, welches sich für Messungen den Beschleunigungssensor in Mobiltelefonen zu Nutzen macht, wird in [14] und [64] beschrieben. Ein weiteres Verfahren, welches in dem nächsten Abschnitt näher beschrieben wird, verwendet zur Steigerung der Zuverlässigkeit neben der Beschleunigungsanalyse noch einen Luftdrucksensor [9].

In Bereichen, wo Wohnungen oder ganze Wohnkomplexe überwacht werden müssen, bietet es sich an, auf eine Erkennung mit Hilfe von Videoverarbeitungsalgorithmen zurückzugreifen. Die größte Schwierigkeit hierbei stellt das Erreichen einer zuverlässigen Erkennungsrate da. Ein einfaches System zur Sturzerkennung wird in [36] erläutert. Es basiert auf einer häufig verwendeten Analyse des Verhältnisses von Höhe und Breite sowie die Änderung des Mittelpunktes einer Person. Diese Methode kann, je nach Position der Person und Kamera oder durch eine Verdeckung von Objekten, jedoch äußerst ungenaue Ergebnisse liefern. Letzteres lässt sich durch das Anbringen der Kamera an der Decke lösen [61].

Air Pressure- and Acceleration-Based Fall Detector

Bei diesem Projekt handelt es sich um eine Zusammenarbeit der Universität Rostock und dem Fraunhofer Institut für Graphische Datenverarbeitung (IGD). Ziel dieses Projektes ist die zuverlässige Erkennung von Stürzen mit Hilfe eines 3-Achsen-Beschleunigungssensors, wie er auch in vielen kommerziellen Geräten verbaut ist, sowie einem hochauflösenden Drucksensor. Als Entwicklungssystem dient der vom IGD entwickelte *StairMaster* [10]. Hierbei handelt es sich um eine tragbare, batteriebetriebene Sensorplattform, die bereits beide Sensoren integriert hat. Zusätzlich ist sie mit einer SD/MMC-Karte ausgestattet und erlaubt über eine Bluetooth-Schnittstelle die Kommunikation mit anderen Geräten.

Der *StairMaster* misst kontinuierlich die Beschleunigung des Systems. Die 3 Beschleunigungswerte des Sensors werden zu einem Richtungsvektor (Betragswert) verrechnet und in einem Array abgespeichert. Sobald 20 Betragswerte gespeichert wurden, wird der Mittelwert und die Varianz ermittelt. Die darauffolgende Schwellwertentscheidung gibt Auskunft, ob in dem Intervall ein Sturz stattgefunden hat. Anschließend werden die letzten 10 Werte des Arrays an die ersten 10 Stellen kopiert und die freigewordenen Plätze mit neuen Daten gefüllt. So ist sichergestellt, dass ein Intervall immer aus 10 alten und 10 neuen Werten besteht.

Abbildung 12 zeigt einen simulierten Sturz nach vorne auf eine Gummimatte. Der Sturz beginnt durch ein starkes Absinken der Beschleunigungskräfte auf beinahe 0g und endet mit dem Aufprall, bei dem Kräfte von ca. 4,5g frei werden.

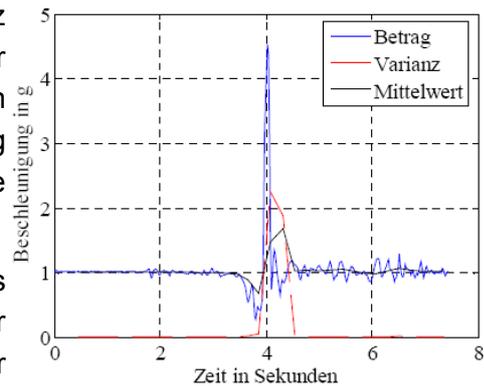


Abbildung 12: Beschleunigungsänderung während eines Sturzes
(Quelle: [9])

Durch die zusätzliche Auswertung des Drucksensors lassen sich Fehler in der Erkennung verringern. Ein möglicher Algorithmus für die Sturzerkennung mit Hilfe von Luftdruckdaten ist, einen dynamischen Mittelwert über die Daten zu berechnen. Der Mittelwert bewirkt hierbei eine Glättung der Werte, filtert mögliches Rauschen heraus und sorgt somit dafür, dass nur relevante Druckänderungen Auswirkungen haben. Anschließend kann über eine Schwellwertentscheidung festgestellt werden, ob ein Sturz stattgefunden hat.

Eine Alternative hierzu stellt das für den Beschleunigungssensor bereits beschriebene Verfahren dar. Aufgrund der deutlich geringeren Datenrate des Drucksensors sowie dem Umstand, dass nur ein Sensorwert analysiert werden braucht, ist jedoch eine Modifizierung des Algorithmus nötig.

Abbildung 13 zeigt die Änderung und den Mittelwert der Intervalle des Luftdrucks während eines simulierten Sturzes. Der Sturz beginnt etwa ab Sekunde 9, was deutlich an dem Anstieg des Luftdrucks von 1006.45hPa auf über 1006.60hPa zu erkennen ist. Anhand des Bildes lässt sich zudem erkennen, dass die Varianz des Mittelwertes vor dem Sturz im Vergleich zu der Varianz während des Sturzes relativ gering ausfällt. Somit lässt sich mit Hilfe einer Schwellwertentscheidung bei der Luftdruckvarianz ein möglicher Sturz ohne großen Rechenaufwand erkennen.

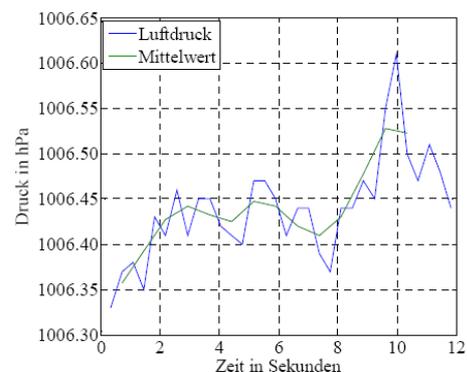


Abbildung 13: Luftdruckänderung während eines Sturzes
(Quelle: [9])

Eine weitere Variante der Luftdruckanalyse ist der Einsatz eines Referenzsystems, welches in einem Raum in fest definierter Höhe angebracht ist. Anhand diesem lassen sich Fehlmeldungen, wie sie durch wetterabhängige Luftdruckänderungen entstehen können, unterdrücken [9].

Praktische Versuche mit dem *StairMaster* und den oben beschriebenen Algorithmen haben gezeigt, dass sich diese Art von Sturzerkennung eignet, um Stürze bereits in ihrer Anfangsphase zu erkennen. Somit können rechtzeitig

Präventionsmaßnahmen, wie beispielsweise das Aufblasen eines Hüftairbags, eingeleitet werden, um Verletzungen so weit wie möglich zu minimieren. Zudem lassen sich die Algorithmen sehr effizient implementieren, was die Verwendung von kleineren und leichteren Geräten erlaubt. Versuche mit dem Luftdrucksensor haben ergeben, dass eine Sturzerkennung ausschließlich auf Basis der Luftdruckanalyse zu Fehlerkennungen führt. Als Gründe hierfür zählen zum einen das bauartbedingte Rauschen des Sensors und zum anderen externe Luftdruckstörungen, wie sie beim Öffnen von Türen oder durch Temperaturschwankungen entstehen.

Video Analytic for Fall Detection from Shape Features and Motion Gradients

Dieses, von Muhammad Jamil Khan und Hafiz Adnan Habib entwickelte und auf dem *World Congress on Engineering and Computer Science 2009* vorgestellte Projekt befasst sich mit der Auswertung von Bewegungsgradienten und der Analyse von Profileigenschaften von Personen zur Erkennung von Stürzen. Das entwickelte Verfahren basiert auf der Erkenntnis, dass die Geschwindigkeit und Richtung eines Objektes sich während eines Sturzes signifikant von der normalen Bewegungsrichtung und -geschwindigkeit unterscheidet. Um Personen von anderen bewegten Objekten zu unterscheiden und die Fehlerrate zu minimieren, wird zusätzlich der Umriss des Objekts analysiert [36].



Abbildung 14: *timed Motion History Image*
(Quelle: [36])

Aussagen über Bewegungsgeschwindigkeit und -Richtung eines Objekts werden mit Hilfe eines *timed Motion History Image (tMHI)* getroffen. Hierfür wird jeder Frame von einem vorher erstellten Hintergrund-Modell abgezogen [29]. Die so erstellte Silhouette wird zusammen mit einem Timestamp und einem Gewichtungsfaktor dem tMHI hinzugefügt (Abbildung 14). Durch Anwenden des Sobel-Operators in X- und Y-Richtung sowie einfacher Winkelberechnungen lässt sich die Steigung, die Richtung und die Stärke eines Punkte bestimmen.

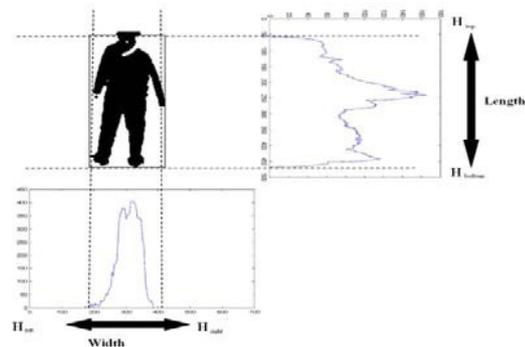


Abbildung 15: *Projektions-Histogramme*
(Quelle: [36])

Um Fehlerkennungen, wie sie z.B. beim schnellen Laufen oder Hinsetzen entstehen können, zu minimieren, wird zusätzlich noch die Höhe und Breite des detektierten Objekts mit einbezogen. Hierfür wird das Binärbild des Objekts zunächst mit Hilfe von zwei Projektions-Histogrammen in die Bereiche H_{top} ,

H_{bottom} , H_{left} und H_{right} unterteilt. Diese Bereiche markieren die 4 Eckpunkte des Objekts (Abbildung 15).

Durch einfache Subtraktion von H_{top} und H_{bottom} , bzw. H_{left} und H_{right} lässt sich die Breite H_{width} und Höhe H_{height} berechnen. Anhand des Verhältnisses dieser Werte zueinander lässt sich erkennen, ob es sich um eine Person handelt und ob diese sich in einer stehenden oder liegenden Körperhaltung befindet. Zusätzlich lässt sich unter der Annahme, dass w und h die Breite und Höhe in Frame n repräsentieren und w' und h' die Breite und Höhe im Frame $n+1$, die Geschwindigkeit einer Änderung β durch folgende Formel bestimmen:

$$\beta = \sqrt{(w - w')^2 + (h - h')^2}$$

Abbildung 17 und 16 verdeutlichen die Änderungen des Breiten/Höhen-Verhältnisses α und der Änderungsgeschwindigkeit β während eines Sturzes.

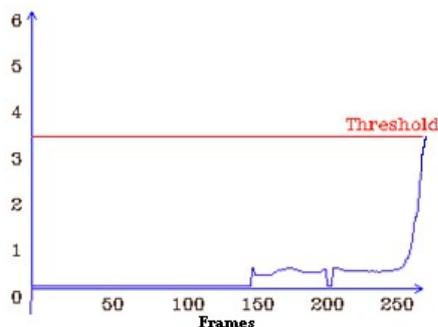


Abbildung 17: Höhen/Breiten-Verhältnis
(Quelle: [36])

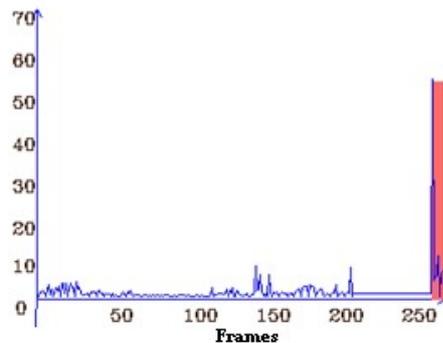


Abbildung 16: Änderung des Höhen/Breiten-Verhältnisses
(Quelle: [36])

Für Experimente wurde eine Implementation in C++ unter Zuhilfenahme der OpenCV Bibliothek erstellt [68]. Als Kamera diente eine handelsübliche USB-Webcam, welche die Probanden aus 4-5 Meter Entfernung aufnahm. Als Bewertungsgrundlage wurden mehrere Arten von Stürzen und sturzähnlichen Bewegungen je 20 mal simuliert. Dabei zeigte sich, dass das System Stürze auf den Rücken, welche aufgrund des hohen Verletzungsrisikos besonders gefährlich sind, zuverlässig erkennt. Dies liegt nicht zuletzt an der eindeutigen vertikalen Bewegungsrichtung und dem signifikanten Änderungen im Höhen/Breiten-Verhältnis. Anders hingegen verhält es sich bei Stürzen auf den Bauch. Da diese in ihrer Bewegung dem normalen Hinlegen sehr ähnlich sind, sind beide nur schwer auseinander zu halten. Dies spiegelte sich auch in den Versuchen wieder, wobei mit 19 richtig erkannten Stürzen dennoch eine hohe Erfolgsquote erreicht wurde.

Intelligent Video Surveillance for Monitoring Fall Detection of Elderly in Home Environments

Ein weiteres Vorgehen wurde an der Universität von Mashhad in Zusammenarbeit mit der Universität von Ramsar von H. Foroughi, B. Shakeri Aski und H. Pourreza entwickelt. In diesem Projekt wird ein Verfahren vorgestellt, welches normale, alltägliche Bewegungen erkennen und von ungewohnten unterscheiden kann, um einen möglichen Sturz zu detektieren [22]. Erreicht wird dies durch die Kombination von Projektions-Histogrammen, der temporalen Analyse der Kopfposition sowie der Untersuchung von Orientierung und Proportion der Person im Raum. Eine abschließende Klassifizierung der Bewegung findet mit Hilfe eines *Multilayer-Perceptron* (MLP)-Netzes statt.

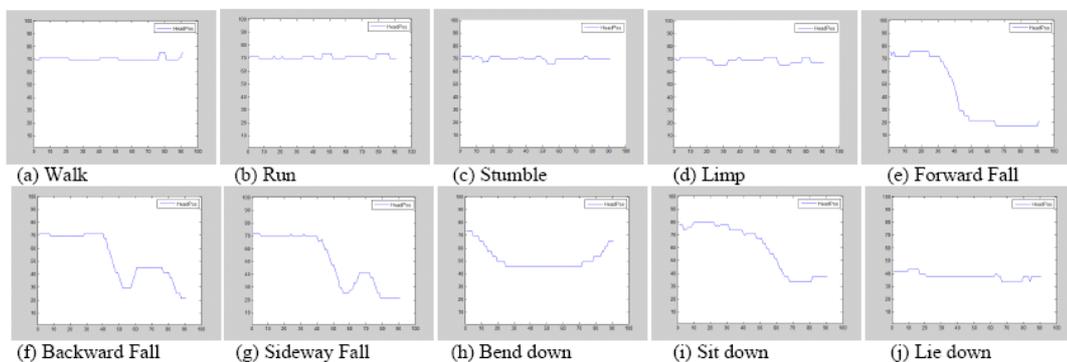


Abbildung 18: Vertikale Änderung der Kopfposition
(Quelle: [22])

Der Kopf einer Person ist einer der markantesten Erkennungspunkte für die grafische Sturzerkennung. Er hat den Vorteil, dass er meistens sichtbar und nur sehr selten von anderen Objekten verdeckt wird und zudem während eines Sturzes eine deutlich andere Bewegung ausführt als bei einer normalen Bewegung. Als Kopfposition wurde in diesem Projekt der höchste Punkt eines vorher erstellten Umrisses der Person gewählt. Dieser wird in jedem Frame neu bestimmt. Aus der Änderung der Position über mehrere Bilder hinweg ergibt sich der erste Inputvektor des MLP-Netzes (Abbildung 18). Einen weiteren Eingabevektor stellen die Eigenschaften einer Ellipse dar, mit deren Hilfe die Orientierung im Raum sowie die Proportion der Person markiert wird. Diese lässt sich über ihre Mittelpunktkoordinaten, die Längen der Major- und Minor-Achse sowie den Winkel zwischen der X-Achse des Bildes und der Major-Achse der Ellipse beschreiben.

Zusammen mit den Merkmalen eines horizontalen und vertikalen Projektions-Histogrammes werden die gesammelten Daten an das neuronale Netz übergeben. Bei diesem handelt es sich um ein 4-lagiges MLP-Netz mit *Backpropagation Learning* Schema.

Als Aktivierungsfunktion wurde

$$a_j = f\left(\sum w_{ji} a_i\right)$$

gewählt, wobei w_{ji} die Gewichtung eines Knotens j einer Schicht zum Knoten i der vorherigen Schicht repräsentiert und a_i den Ausgang des Knoten i . Zum Anlernen des Netzes und für praktische Versuche wurden drei verschiedene Arten von Bewegungen aufgenommen: Normale tägliche Aktivitäten, Stürze und abnormale Bewegungen. Letztere beinhalten Bewegungen wie taumeln oder hinken. Experimente mit insgesamt 1250 unterschiedlichen Bewegungen in 10 Kategorien ergaben eine Erkennungsrate von über 90% in den Bereichen *normale Bewegungen* und *Stürze*. Bei den *abnormalen Bewegungen* liegen die Ergebnisse mit 88% nur minimal darunter.

Im Durchschnitt ließ sich mit dem Verfahren eine Erfolgsquote von 91.12% erreichen. Daraus lässt sich erkennen, dass Verfahren zur Sturzerkennung nicht ausschließlich auf arithmetischen Verfahren basieren müssen, sondern auch mit selbst lernenden neuronale Netze realisiert werden können.

3.3.3 Fazit

In Abschnitt 3.3.1 wurden mehrere kommerzielle Produkte vorgestellt, die sich mit dem Erkennen von hilflosen Personen befassen. Jedes Produkt bietet verschiedenen Vor- und Nachteile, welche sie für ihren jeweiligen Aufgabenbereich qualifizieren. Beispielsweise werden für den Einsatz von *QuietCare*[®] lediglich Bewegungsmelder benötigt, welche sich auch in bestehenden Wohnungen installieren lassen. Die Auswertung der Bewegungsdaten findet jedoch in einem zentralen Servicecenter statt, von wo aus im Notfall auch die Benachrichtigung der zuständigen Personen erfolgt. Dies bedeutet neben monatlichen Kosten auch die Abhängigkeit vom entsprechenden Dienstleister. Ebenfalls mit einem zentralen Servicecenter arbeitet das System von *Philips Lifeline*. Da der Auslöser für Notrufe hier jedoch direkt am Handgelenk oder um den Hals getragen wird, sind keine baulichen Veränderungen in der Wohnung nötig. Zwar vertreibt Philips das Lifeline System aktuell nur in Amerika, jedoch gibt es in Deutschland mehrere Anbieter, welche ein äquivalentes System anbieten. Hierzu zählen beispielsweise das *Deutsche Rote Kreuz* [8], die *Johanniter-Unfall-Hilfe* [28] oder *Malteser* [33]. Das Armband bzw. der Anhänger mit dem Notrufknopf birgt allerdings die Gefahr, dass vergessen wird, es anzulegen oder versehentlich der Auslöser betätigt wird. Im Gegensatz zu diesen Systemen kommt das vom Fraunhofer IFA entwickelte *MotionSENS*[®] vollständig ohne zusätzliche Kosten aus. Durch die Unabhängigkeit von Dienstleistern lässt sich individuell einstellen, welche Personen in einem Notfall alarmiert werden sollen. Ein weiterer Vorteil ist, dass durch die integrierten Sensoren auch Alarm ausgelöst wird, wenn die gestürzte

Person hierzu nicht mehr in der Lage ist. Aufgrund der geringen Akkulaufzeit eignet sich das Gerät jedoch nur für kurzzeitige Einsätze. Da es zudem noch nicht erhältlich ist, ist ein Einsatz im Living Place von vornherein nicht möglich. Ähnlich ist es bei der Sensormatte *SensFloor*[®]. Da sie nur aus einer Matte, Netzteil und einer Empfangseinheit besteht, lässt sie sich problemlos an geeigneten Orten platzieren, ohne weitere Veränderungen vornehmen zu müssen. Die Kommunikation zwischen Sensormatte und Empfangseinheit erfolgt kabellos über eine Funkverbindung. Dies hat den Vorteil, dass die Matte in einem anderem Raum als der Empfänger platziert werden kann. Des Weiteren bietet der Empfänger je nach Ausstattung Anschlussmöglichkeiten für Telefonanlagen und andere Verbraucher. Nachteil von *SensFloor*[®] ist der relativ hohe Preis. Eine Sensormatte von etwa 60 x 110cm kostet aktuell (Stand 12/2011) zwischen 919€ und 954€ [16]. Da eine Sensormatte mit diesen Maßen für den Living Place nicht ausreichend ist, müssten entweder mehrere oder größere Matten beschafft werden. Die hohen Kosten würden jedoch den Anforderungen an die Entwicklung eines kostengünstigen Systems, wie sie in Kapitel 3.2 beschrieben sind, widersprechen.

Tabelle 1 fasst die Vor- und Nachteile der einzelnen Systeme noch einmal zusammen.

	QuietCare[®]	MotionSENS[®]	Philips Lifeline	SensFloor[®]
Pro	-Nachträglich installierbar	-Nachträglich installierbar -Konfigurierbares Verhalten	-Nachträglich installierbar	-Nachträglich installierbar -Konfigurierbares Verhalten
Contra	-Servicecenter nötig, monatliche Kosten	-Kurze Akkulaufzeit -Noch nicht erhältlich -Muss am Körper getragen werden	-Muss am Körper getragen werden -Nicht in Deutschland erhältlich -Servicecenter nötig, monatliche Kosten	-Relativ teuer

Tabelle 1: Vor- und Nachteile mehrerer kommerzieller Produkte

Wie sich zeigt, lassen sich alle vorgestellten Produkte problemlos in bestehende Wohnumgebungen integrieren. Deutliche Unterschiede gibt es jedoch in Bezug auf Verfügbarkeit und Kosten. Die am weitesten verbreitete Art von Systemen setzt auf Dienstleister und zentrale Servicecenter, was für größere Pflegefirmen durchaus von Vorteil sein kann, für Privatpersonen und dem Einsatz im Living Place jedoch nicht unbedingt erwünscht ist. Als einziges vorgestelltes Produkt würde sich der von *FutureShape* vertriebene *SensFloor*[®] für die Verwendung im Living Place eignen. Dies ist jedoch wegen der hohen Anschaffungskosten nicht umsetzbar. Somit bleibt es bei der Entwicklung eines eigenen, kostengünstigen

Systems zur Erkennung von gestürzten Personen.

Die in Abschnitt 3.3.2 vorgestellten Forschungsprojekte geben einen Einblick, welche Methoden und Techniken sich für die Erkennung von Stürzen verwenden lassen. Neben der Auswertung von Beschleunigungs- und Drucksensoren, welche am Körper getragen werden müssen, existieren einige Ansätze für die Analyse von Videodaten. Diese verwenden u.a. Methoden zur Detektion der Kopfposition oder der Lage einer Person, um einen Sturz zu erkennen. Solche Verfahren setzen jedoch voraus, dass die Kamera seitlich zu der entsprechenden Person installiert ist. Während dies in Laboren problemlos möglich ist, kann der Blickbereich von in Wohnungen installierten Kameras durch Gegenstände verdeckt werden, wodurch Personen nicht mehr korrekt erkannt werden. Abhilfe schafft die Positionierung der Kameras an den Decken der Räume. Durch den geänderten Blickwinkel, lassen sich jedoch die beschriebenen Verfahren nicht mehr einsetzen. Da im Bereich der Sturzerkennung bisher kaum Forschungsarbeit mit derart platzierten Kameras betrieben wurde, ist es somit nötig, ein neues Verfahren hierfür zu entwickeln.

3.4 Auswahl geeigneter Messverfahren

Wie aus dem vorherigen Kapitel ersichtlich ist, gibt es eine Vielzahl an unterschiedlichen Sensoren, mit deren Hilfe Stürze oder gestürzte Personen erkannt werden können. Neben Kameras und Geräten, die am Körper getragen werden, beispielsweise zur Messung von Beschleunigung oder Luftdruck, eignen sich auch Installationen, die in die Wohnumgebung integriert werden. Hierzu zählen beispielsweise im Boden integrierten Sensoren wie Druck-, Erschütterungs- oder Annäherungssensoren. Ebenfalls denkbar ist eine Überwachung mittels Mikrofonen, die auf bestimmte Lautstärken oder Geräuschkuster achten.

Für das in dieser Arbeit entwickelte System sollen neben der Kamera ausschließlich Sensoren zum Einsatz kommen, welche sich kostengünstig und leicht installieren lassen.

In den folgenden Abschnitten sollen verschiedene Messverfahren zur Detektion von Stürzen erprobt werden. Dies umfasst neben der in Abschnitt 3.4.1 beschriebenen Erkennung von Erschütterungen durch Beschleunigungssensoren und Seismographen den Aufbau und die Auswertung verschiedener kapazitiver Messtechniken (Abschnitt 3.4.2) zur Detektion von Annäherungen. Den Abschluss des Kapitels bildet eine Zusammenfassung und Auswertung der gesammelten Erkenntnisse.

3.4.1 Erschütterungssensoren

Ein Erschütterungssensor dient zum registrieren von Schwingungen eines Objektes. Aufgrund ihrer Eigenschaft an den zu detektierenden Objekten befestigt zu sein, werden sie auch häufig als Körperschallsensor bezeichnet. Gängige Einsatzgebiete dieser Sensoren sind als Glasbruchmelder in Einbruchmeldeanlagen, als Nachweis für einen sachgemäßen Transport von empfindlichen Dingen oder in Seismographen zur Detektion von seismischen Wellen. In den folgenden Abschnitten sollen zwei Messverfahren evaluiert werden, mit denen es möglich ist, kleinste Schwingungen eines Objekts zu erfassen und auszuwerten.

Piezoelektrischer Seismograph

Ein piezoelektrischer Seismograph bedient sich dem in Kapitel 2.3 beschriebenen Effekt der Spannungs-Erzeugung durch Druck auf ferroelektrischen Materialien.

Für einen Versuchsaufbau wurde der in Abbildung 19 gezeigte Schallwandler mit einem Gewicht von etwa 250 Gramm belastet und auf einer ebenen Fläche befestigt. Da die erzeugte Spannung je nach Druck nur einige Mikro- bis Millivolt betragen kann, wurde diese um den Faktor ~ 100 verstärkt. Als Verstärkerschaltung diente der Eingangverstärker eines einfachen Seismometers [65]. Die Auswertung übernahm ein Mikrocontroller vom Typ ATmega48.



Abbildung 19: Ferroelektrischer Lautsprecher

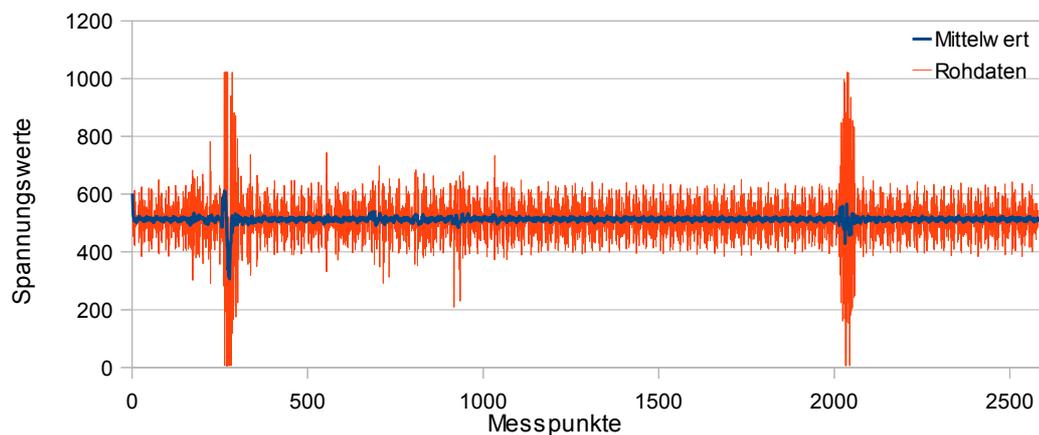


Abbildung 20: Erschütterungssensors Auswertung

Wie in Abbildung 20 erkennbar heben sich trotz starker Schwankungen der Werte die beiden Erschütterungen, die in 3 und 5 Meter Entfernung erzeugt wurden, deutlich ab. Die Schwankungen zwischen den Erschütterungen deuten auf eine unsaubere Versorgungsspannung hin. Ein obendrein in direkter Nähe des Sensors stehender PC könnte zusätzlich zu diesem unsauberen Messverlauf

beigetragen haben.

Mit geringerer Distanz zwischen Sensor und Erschütterung beginnt die Schaltung jedoch zu übersteuern. Dies macht sich durch einen länger andauernden konstanten Spannungspegel am oberen Limit des Verstärkerausgangs bemerkbar. Hierdurch ist es nicht mehr möglich, Stürze und normale Erschütterungen, wie sie z.B. durch Schritte entstehen, in der Nähe des Sensors zu unterscheiden. Um dies zu kompensieren, müsste eine komplexe Elektronik entworfen werden, welche den Verstärkungsfaktor der Schaltung kontinuierlich an die jeweilige Erschütterungsstärke anpasst.

SCA3000

Bei der Serie SCA3000 von VTI Technologies handelt es sich um eine Reihe hochempfindlicher Beschleunigungssensoren. Diese erlauben eine präzise Messung von kleinsten Bewegungen. Als Versuchsobjekt kam ein SCA3000-D01 mit folgenden Eigenschaften zum Einsatz:

- Spannungsversorgung von 2.35V bis 3.6V
- Messbereich $\pm 2g$
- Auflösung 0.75mg
- $\pm 1\%$ Genauigkeit
- SPI-Interface¹

Der Vorteil gegenüber dem im vorherigen Abschnitt beschriebenen Aufbau ist die Integration der kompletten Messkette in den Chip (Abbildung 21). Hierdurch ist keine externe Verstärkung des Signals mehr nötig. Die Kommunikation mit dem Mikrocontroller erfolgt vollständig digital über eine SPI-Schnittstelle.

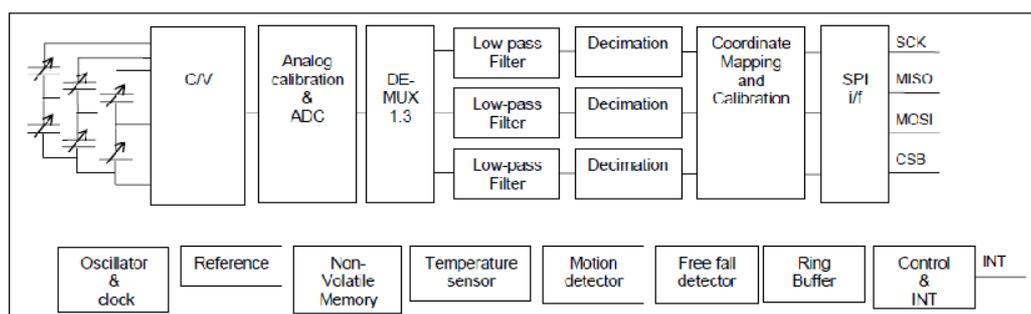


Abbildung 21: Blockschaltbild des SCA3000
(Quelle: SCA3000-D01 Data Sheet)

Um die volle Bandbreite des Sensors ausnutzen zu können, wurde der integrierte Tiefpass-Filter für die Versuche deaktiviert. Dies bewirkt einen Anstieg der internen Messungen etwa um den Faktor 1.6. Durch die Abschaltung des

¹ SPI (Serial Peripheral Interface): Synchroner, serieller Datenbus

Tiefpass-Filters erhöht sich jedoch im gleichen Maße auch das Rauschen des Sensors. Dies hat sich jedoch während der Versuche nicht als hinderlich erwiesen.

Obwohl für Erschütterungen nur die vertikale Beschleunigung von Interesse ist, wurden für Vergleichsmessungen alle 3 Achsen ausgelesen. Abbildung 22 zeigt die gemessene Vertikalbeschleunigung beim Auftreten in einem Abstand von bis zu 2 Metern zum Sensor. Deutlich zu erkennen sind die starken Ausschläge am Anfang der Messreihe.

Diese fanden in nächster Nähe zum Sensor statt. Mit zunehmender Entfernung verringerten sich die Messwerte jedoch sichtbar. Die Ausschläge lassen sich darüber hinaus dennoch weiterhin deutlich erkennen.

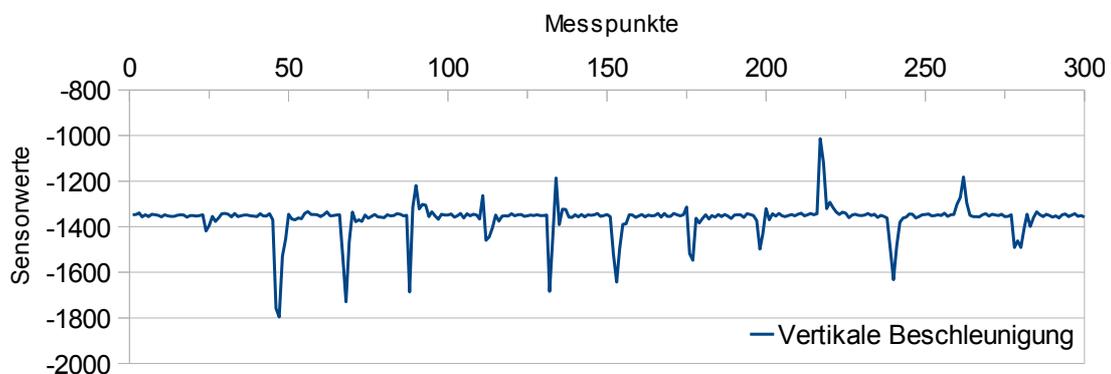


Abbildung 22: SCA3000 - Erschütterungen in geringem Abstand

Anders hingegen sieht es bei Entfernungen von über 3 Metern aus. Wie Abbildung 23 verdeutlicht, dominiert hier das Rauschen des Sensors. Eine Unterscheidung zwischen Rauschen und gemessenen Erschütterungen ist nahezu unmöglich. Beide Messungen wurden mit einer Frequenz von etwa 250Hz durchgeführt, was einen Abstand von 4 Millisekunden zwischen zwei Messpunkten bedeutet.

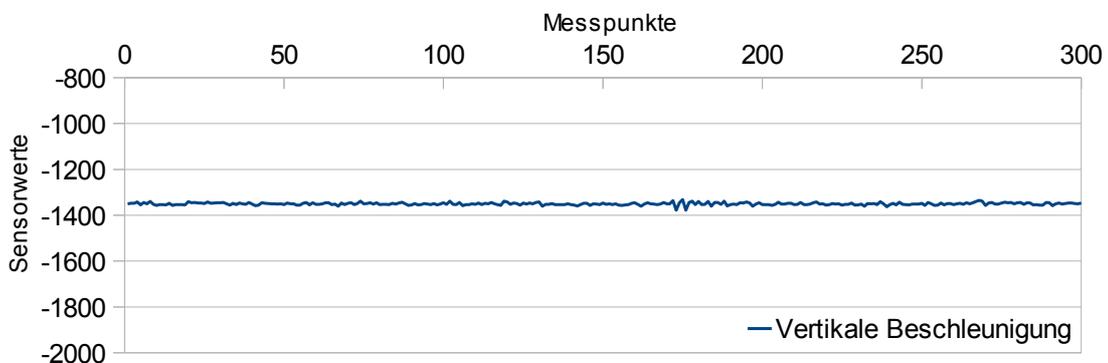


Abbildung 23: SCA3000 - Erschütterungen in großem Abstand

Wie durch die obigen Abbildungen ersichtlich, eignet sich diese Art von Erschütterungssensoren nur für geringe Entfernungen. Sobald der Abstand zwischen Sensor und der Quelle der Erschütterungen weiter als einige Meter auseinanderliegt, ist keine zuverlässig Erkennung mehr möglich. Zudem hängt die maximale Entfernung von dem Dämpfungsverhalten des verwendeten Untergrundes ab. Während Stein- oder gefliester Boden Erschütterungen über einige Meter weiterleiten, absorbiert Teppichboden die Schwingungen schon nach kurzer Distanz vollständig.

3.4.2 Kapazitive Sensoren

Kapazitive Sensorsysteme dienen zur Messung von Änderungen in elektrischen Feldern. Sie bestehen aus einer Steuer- und Auswertungseinheit sowie einer elektrisch leitfähigen Elektrode, die in Verbindung mit dem zu messenden Objekt einen Kondensator bildet. Je nach Abstand oder Verformung des Objekts ändert sich die messbare Kapazität dieses Kondensators. Die Einsatzgebiete von kapazitiven Sensoren umfassen die Messung von Druck in Gasen oder Flüssigkeiten, sowie die Erkennung von Abständen, Annäherungen und Bewegungen.

Für das in dieser Arbeit entwickelte System werden jedoch nur Technologien betrachtet, die zur Messung von Abständen und Annäherungen dienen, da die eingesetzten Sensoren in Böden oder Teppichen integriert werden sollen.

RC-Zeitkonstante

Erste Überlegungen ergeben, dass die Elektrode nichts anderes als einen simplen, wenn auch großflächigen Kondensator darstellt und somit die Zeit, welche die Elektrode zur Auf- und Entladung benötigt, wie bei jedem anderen Kondensator auch gemessen werden kann. Hierfür wird die Elektrode mit einem definierten Strom aufgeladen, bis die Spannung eine festgelegte Schwelle überschreitet. Anschließend wird die Elektrode wieder entladen und der Vorgang von Neuem begonnen. Durch Berühren oder Annähern an die Elektrode vergrößert sich deren Kapazität, wodurch das Auf- und Entladen mehr Zeit in Anspruch nimmt. Die Kapazitätsänderung der Elektrode bewegt sich im Bereich von wenigen pF, was eine sehr genaue Zeitmessung voraussetzt.

Ein erster Versuchsaufbau bestand aus einem ca. 5 cm x 5 cm großen Aluminiumquadrat, welches über einen 3 Mega-Ohm Widerstand mit der Stromversorgung von 5 V verbunden war. Anschließend konnte mittels eines Mikrocontrollers, dessen Anschluss wie in Abb. 24 dargestellt, ebenfalls mit der Schnittstelle von Elektrode und Widerstand verbunden wurde, die Zeit

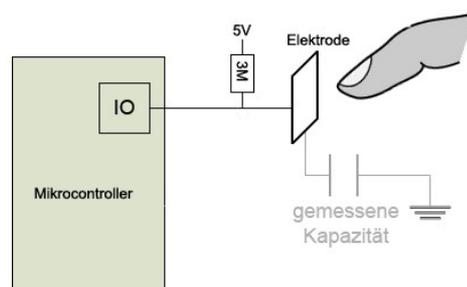


Abbildung 24: Erster Testaufbau

gemessen werden, die nötig war, um das Metallquadrat auf die entsprechende Spannung aufzuladen. Hierfür wurde der Anschluss des Mikrocontrollers als hochohmiger Eingang geschaltet, so dass sich die Elektrode langsam über den Widerstand aufladen konnte. Sobald die Spannung an der Elektrode die Schwellenspannung für einen High-Pegel des Mikrocontroller-Eingangs überstieg, wurde der Zählerstand im Controller ausgelesen und an den PC zur weiteren

Verarbeitung übertragen. Anschließend wurde der Anschluss des Mikrocontrollers auf Ausgang geschaltet und die Elektrode auf diese Weise wieder entladen, so dass der Zyklus neu beginnen konnte.

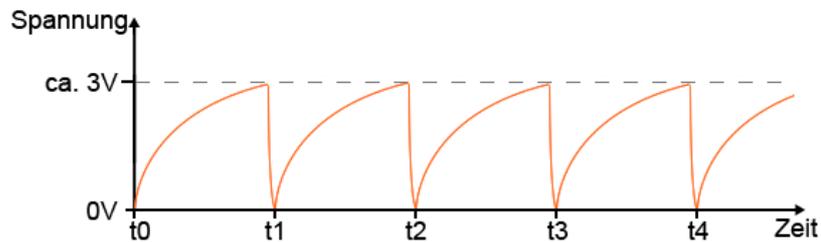


Abbildung 25: Spannungsverlauf an der Elektrode im ersten Testaufbau

Abbildung 25 illustriert den Ablauf der Messungen anhand der Spannung an der Elektrode. Zum Zeitpunkt t_0 wird der Anschluss des Mikrocontrollers als Eingang geschaltet, so dass sich die Elektrode langsam aufladen kann. Bei Erreichen von ca. 3V (Zeitpunkt t_1), schaltet der Anschluss auf Ausgang um und entlädt die Elektrode innerhalb kürzester Zeit. Anschließend wird wieder auf Eingang geschaltet und der Vorgang beginnt von Neuem.

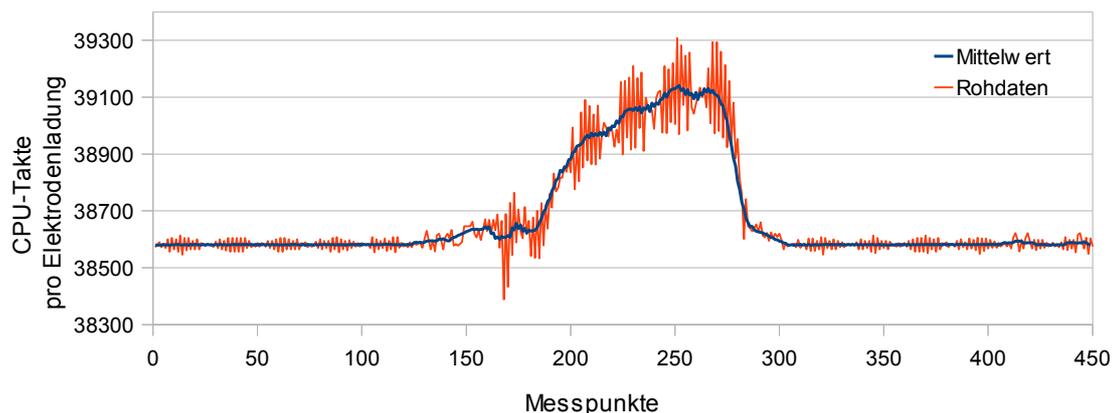


Abbildung 26: Kapazitätsmessung eines Feldes durch Zeitmessung

Abbildung 26 zeigt das Messergebnis des Aufbaus. Die horizontale Achse der Abbildung zeigt den zeitlichen Verlauf über mehrere Messungen. Bedingt durch die Ladezeit der Elektrode beträgt der Abstand zwischen zwei Messungen 4.6 bis 5 Millisekunden. Eine Berührung der Elektrode mit dem Finger fand etwa zwischen dem 150 ten und 300 ten Messpunkt statt. Neben der Änderung der benötigten CPU-Takte zum Laden der Elektrode, sind deutlich die extremen Schwankungen in diesem Bereich auf der Y-Achse zu erkennen. Diese entstehen vermutlich durch induzierte Spannungen durch den menschlichen Körper. Ebenfalls deutlich zu sehen sind die minimalen periodischen Schwankungen vor und nach der Berührung. Hierbei handelt es sich höchst wahrscheinlich um nicht gefilterte Reste der 50Hz Netzspannung oder Schwankungen in der Spannungsversorgung.

Bei Anschluss eines ca. 2 m langen Kupferkabels mit 0.14 mm Querschnitt hingegen ergibt sich ein Messbild wie in Abbildung 27. Die Anzahl der nötigen CPU-Takte, um die Elektrode aufzuladen, ist gegenüber der Ruhelage der vorherigen Messung nur um knapp 500 Punkte gestiegen, dennoch ist es nicht mehr möglich, in dem Signalverlauf eine Berührung zu detektieren. Aufgrund der Länge verhält sich das Kabel vermutlich wie eine Antenne und bewirkt durch den Empfang eines bestimmten Frequenzspektrums die Einstreuung von Störungen.

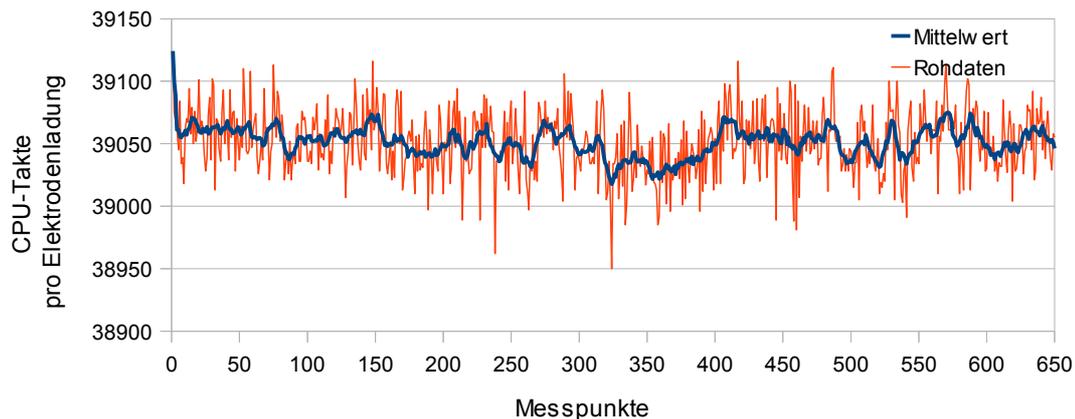


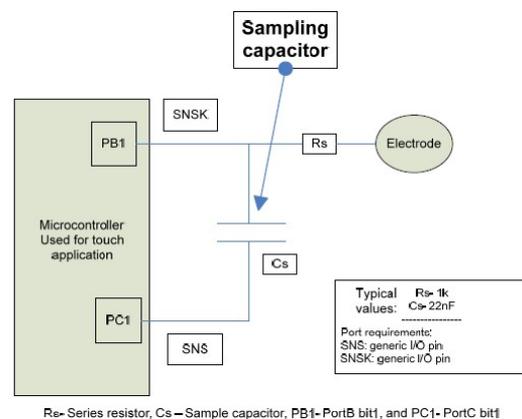
Abbildung 27: Kapazitätsmessung eines Kabels durch Zeitmessung

Wie aus den Messergebnissen sichtbar, ist dieses Messverfahren auf eine sehr stabile Versorgungsspannung und eine störungsfreie Umgebung angewiesen. Beides lässt sich unter realistischen Bedingungen nicht oder nur sehr schwer sicherstellen. Das in diesem Abschnitt beschriebene Verfahren ist somit ungeeignet für den Einsatz außerhalb eines entsprechend eingerichteten Labors.

QTouch®

Nachdem sich die soeben beschriebene Technik als ungeeignet erwiesen hat, wurde ein auf der *QTouch*® Technologie von Atmel basierendes Verfahren getestet. Dieses arbeitet durch ein Verschieben der Ladung aus der Elektrode in einen Speicher-Kondensator mit bekannter Größe [3].

Hierfür werden zunächst sowohl die Elektrode, wie auch der Kondensator C_s entladen. Dies geschieht, indem die Mikrocontroller-Anschlüsse *PB1* und *PC1* als Ausgang konfiguriert und Masse gezogen werden. Durch Umschalten von *PC1* als hochohmigen Eingang, wird anschließend C_s



R_s - Series resistor, C_s - Sample capacitor, PB1- PortB bit1, and PC1- PortC bit1

Abbildung 28: Aufbau QTouch®
(Quelle: [3])

deaktiviert. Durch kurzes Umschalten von *PB1* nach 5 V wird nun die Elektrode aufgeladen (Abbildung 29, t_0 bis t_1).

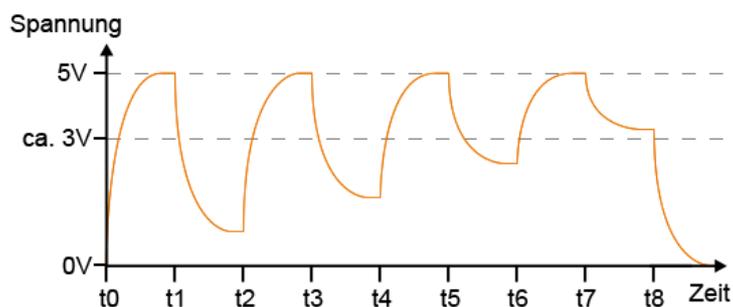


Abbildung 29: Spannungsverlauf an der Elektrode bei QTouch

Da durch den hochohmigen Anschluss *PC1* *Cs* quasi deaktiviert ist, bleibt seine gespeicherte Ladung unverändert. Anschließend wird *PB1* als Eingang geschaltet und *PC1* als Ausgang, um eine Seite von *Cs* gegen Masse zu ziehen. Auf diese Weise verteilt sich die in der Elektrode gespeicherte Ladung gleichmäßig auf Elektrode und den Kondensator, wodurch die Spannung an beiden einbricht (Abbildung 29, t_1 bis t_2). So lange die Spannung unter der Schaltschwelle von 3V fällt, wird die Elektrode weiter zyklisch aufgeladen und die Ladung in den Kondensator übertragen (Abbildung 29, t_2 bis t_8). Sobald die Spannung nicht mehr unter 3V fällt, gilt *Cs* als vollständig geladen (Abbildung 29, t_8). Die benötigte Anzahl der Zyklen wird gespeichert und der komplette Vorgang beginnt von Neuem. Durch Annäherung oder Berühren der Elektrode werden Ladungen von ihr abgezogen, wodurch sich *Cs* langsamer laden lässt. Dies macht sich durch eine höhere Anzahl an benötigten Zyklen bemerkbar.

Um dieses Verfahren zu untersuchen, wurde ein Mikrocontroller der AVR-Serie mit 4 Sensorkanälen ausgestattet. Jeder Kanal verbindet eine Elektrode über einen 1k Ohm Widerstand mit dem Sample-Kondensator. Als Kommunikationsschnittstelle wurde ein RS-485 Bus gewählt. Dieser hat den Vorteil, dass er eine große Toleranz gegenüber Störungen aufweist und es zudem erlaubt, mehrere Module mit geringem Aufwand über einen Anschluss mit dem PC zu verbinden. Als Schnittstelle zwischen Bus und PC dient ein RS-485 nach USB Wandler. Abbildung 30 veranschaulicht den Aufbau anhand eines Blockschaltbildes.

Damit die Module möglichst klein ausfallen, wurden ausschließlich Bauteile im SMD Format verwendet. Als Mikrocontroller kam ein Attiny2313 zum Einsatz, da dieser Typ der momentan günstigste mit Hardware-UART² ist. Durch die UART-Schnittstelle in Hardware-Form vereinfacht sich die Softwareentwicklung enorm.

² UART (Universal Asynchron Receiver Transmitter): Schnittstelle zum seriellen Übertragen von Informationen über eine Datenleitung.

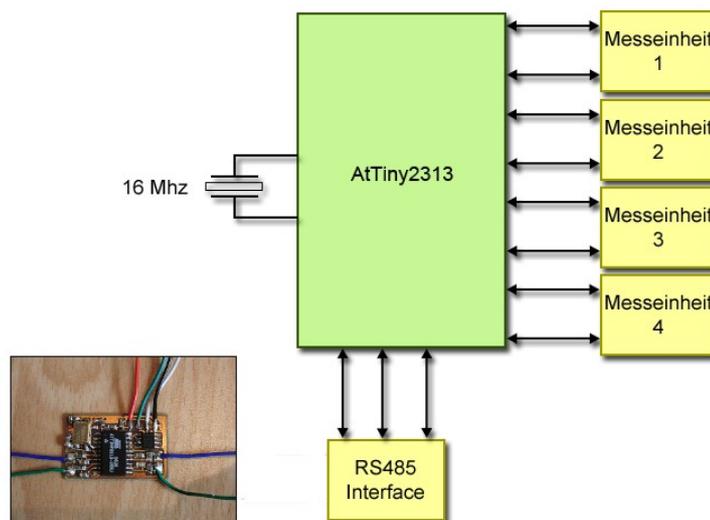


Abbildung 30: Blockdiagramm des ersten Prototyps

Der Vorteil dieses Sensors besteht in der geringen Anzahl an Bauteilen, was eine kostengünstige Herstellung ermöglicht. Zudem wird hierdurch die Möglichkeit von fehlerhaften und defekten Bauteilen deutlich reduziert. Um einen Vergleichswert zu erhalten, wurden die Messungen mit dem selben Aluminiumstück durchgeführt, das auch bei den ersten Messungen bereits zum Einsatz kam. Das folgende Diagramm veranschaulicht die Messung der einzelnen Kanäle. Die einzelnen Ausschläge bei einer Berührung der Elektrode sind deutlich durch die steigenden Ladezyklen sichtbar.

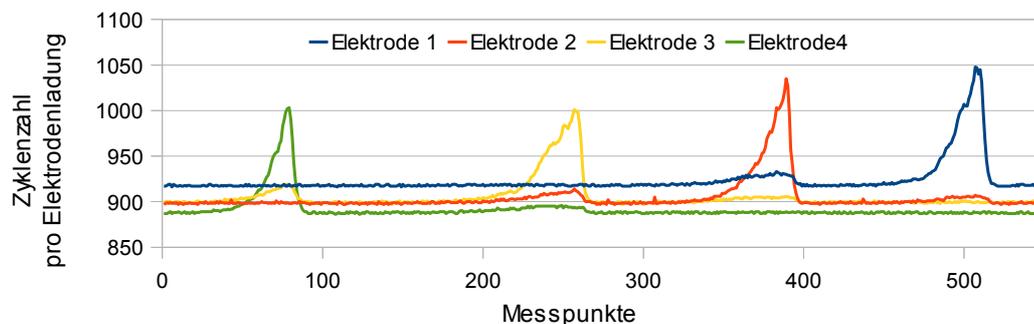


Abbildung 31: Annäherungsmessung – Erster Prototyp

Wie in der Abbildung ersichtlich, bewirkt eine Berührung der Elektroden einen große Änderung der Ladezyklen, wodurch sich eine Berührung leicht detektieren lässt. Auch die Schwankungen in den Messwerten, wie sie im ersten Versuch auftraten, sind nicht zu finden, was auf eine gute Störungsresistenz deutet.

Bevor jedoch ein abschließendes Urteil gefällt werden kann, soll noch eine weitere Technik untersucht werden.

CapToolKit

Hierfür wird das von Raphael Wimmer an der Universität München entwickelte Toolkit *CapToolKit (CTK)* eingesetzt. Dieses besteht aus einer Steuereinheit, an welcher bis zu 8 einzelne Sensoren angeschlossen werden können (Abb. 32). Jeder einzelne Sensor besteht im wesentlichen aus einem Zeitgeber-IC vom Typ 555 und einem Operationsverstärker.

Das Zeitgeber-IC fungiert als Oszillator und erlaubt es, aus der Kapazität der Elektrode eine messbare Frequenz zu erzeugen. Hierfür wird die Elektrode durch den IC kontinuierlich elektrisch aufgeladen und anschließend wieder entladen. Dies

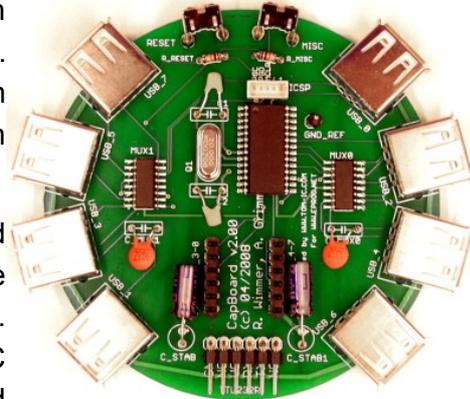


Abbildung 32: CapToolKit – CapBoard
(Quelle: [57])

geschieht, indem der Elektrode ein geringer Strom zugeführt wird, um sie auf ein bestimmtes Spannungslevel zu bringen. Sobald die Spannung an der Elektrode einen fest definierten Wert überschreitet, schaltet der IC um und entlädt sie wieder. Die Zeit, welche zwischen dem Umschalten und dem Erreichen der Spannungsgrenzen verstreicht, wird von der zentralen Steuereinheit gemessen und in einen normierten Wert übertragen. Hierbei werden immer zwei Sensoren zeitgleich ausgewertet.

Bevor die Signale letztendlich an den PC weitergeleitet werden, durchlaufen sie noch eine Filterung, welche fehlerhafte Daten aussortiert und für einen gleichmäßigeren Verlauf sorgt [57]. Hierfür kommt ein adaptiver, fließender Mittelwert-Filter zum Einsatz. Abhängig von der Schwankung der Sensorwerte werden unterschiedlich viele Datensätze zur Bildung des Mittelwertes verwendet. Bei starken Schwankungen wird der Mittelwert aus weniger Sensorwerten gebildet als bei schwachen. Dies bewirkt, dass kurzzeitige und starke Ausschläge nicht herausgefiltert werden und erkennbar bleiben, während schwache Änderungen herausgefiltert werden. Anschließend werden die Sensorwerte normalisiert. Dies bedeutet, dass ein bestimmter Wert als Nulllinie festgelegt wird. Von diesem wird der aktuelle, gemittelte Sensorwert abgezogen. Hierdurch ergibt sich für alle angeschlossenen Elektroden ein vergleichbarer Wert, unabhängig von Fertigungstoleranzen der Bauteile und Elektroden.

Als Elektrode wurde wieder ein Aluminiumquadrat und das im ersten Test bereits eingesetzte Kabel verwendet. Abbildung 33 zeigt die Messung der Aluminiumelektrode. Durch die Normierung ergab sich in Ruhelage ein Wert von etwa 0, welcher bei Annäherung an die Elektrode stark anstieg. Das CapToolKit erwies sich als erstaunlich empfindlich. Wie durch den Anstieg der Werte zwischen Messpunkt 50 und 150 ersichtlich, ließ sich bereits die Annäherung an

die Elektrode detektieren. Des Weiteren zeigten sich deutliche Schwankungen beim Ansteigen und Abfallen der Werte.

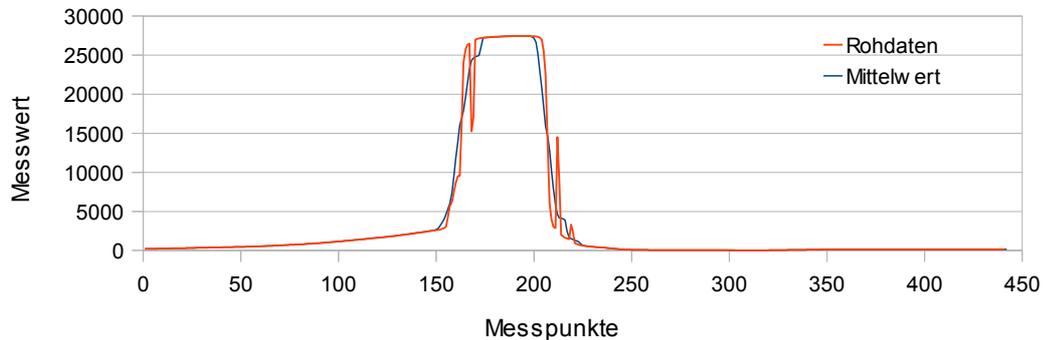


Abbildung 33: Kapazitätsmessung einer Aluminiumelektrode durch CapToolKit

Bei Anschluss des Kabels als Elektrode zeigten sich kaum Unterschiede zu der vorherigen Messung, was sehr für die Stabilität dieses Systems spricht. Lediglich der Ausschlag des Zählers sank (Abbildung 34). Eine Messung dauerte bei beiden Messreihen etwa 100 Millisekunden, was eine Abtastrate von 10 Messungen pro Sekunde bedeutet.

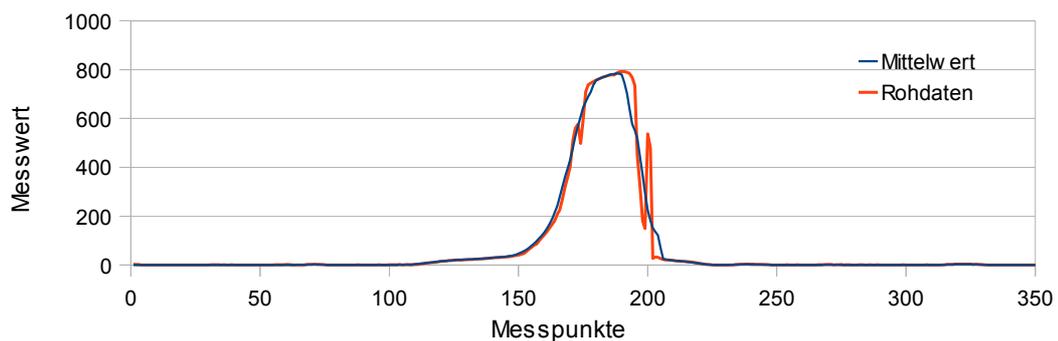


Abbildung 34: Kapazitätsmessung eines Kabels durch CapToolKit

Die Stabilität und Empfindlichkeit sprechen zwar eindeutig für das CapToolKit, da jedoch die Rohdaten bereits in dem Mikrocontroller gefiltert und geglättet werden, ist auf diese Weise keine objektiver Vergleich mit den bisherigen Verfahren möglich.

Aus diesem Grund wurde ein weiteres Sensormodul angefertigt, welches auf dem selben Funktionsprinzip basiert wie das CapToolKit von Raphael Wimmer, jedoch ungefilterte Daten liefert.

Damit es nicht zu Beeinflussung durch Spannungsschwankungen kommen kann, wurde das Modul mit einem eigenen Spannungsregler ausgestattet. Somit ist sichergestellt, dass die Schaltung auch bei einer instabilen Stromversorgung mit den benötigten 5V betrieben wird.

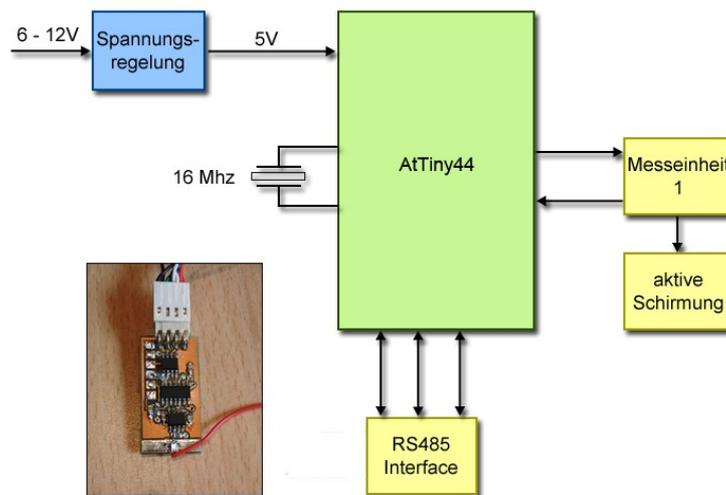


Abbildung 35: Schaltung des dritten Prototyps

Mit diesem Modul sollte zudem demonstriert werden, wie weit sich die Module minimieren lassen. Daher wurde der bisher verwendete Mikrocontroller durch Attiny44 ersetzt, welcher ein wesentlich kleineres Gehäuse besitzt als der vorher benutzte Attiny2313. Der Vorteil der kleineren Größe wurde jedoch durch geringere Ausstattung erkauft. So besitzt der neue Controller kein Hardware-UART. Dieser musste in Software nachgebildet werden, wodurch sich die maximale Busgeschwindigkeit auf 76800 Bits/Sekunde senkte. Des weiteren wurde dieses Sensormodul um einen sogenannten Guard erweitert, welcher auch im CapToolKit vorhanden ist. Hierbei handelt es sich um eine Abschirmmöglichkeit, um die Elektrode vor äußeren Einflüssen zu schützen. Der Guard besteht aus einem Operationsverstärker, welcher dafür sorgt, dass die Abschirmfläche immer das gleiche Spannungspotential hat, wie die Messelektrode.

Für die Versuche mit diesem Modul wurden die bereits beschriebene Aluminiumelektrode verwendet. Abbildung 36 veranschaulicht die Messwerte. Wie bereits beim CapToolKit sichtbar, weist das System eine gute Stabilität in den Messwerten aus. Auch die Berührung der Elektrode ist, wenn auch nicht so stark, ebenfalls deutlich durch ein Abfallen des Zählerstands ab Messpunkt 100 erkennbar. Auch eine versehentliche Annäherung zwischen den Messpunkten 60 und 80 zeigt bereits einen deutlichen Ausschlag.

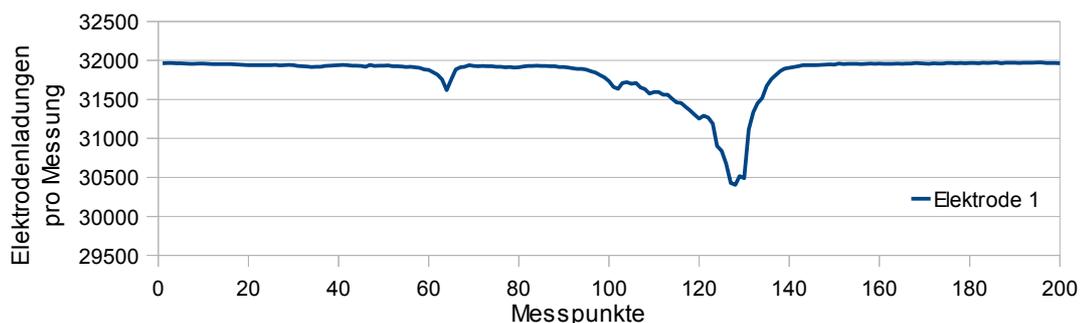


Abbildung 36: Annäherungsmessung - Prototyp 3

3.4.3 Auswertung der Messverfahren

Die Testreihen mit den Erschütterungssensoren haben gezeigt, dass in einem nahen Umfeld zum Sensor Erschütterungen zwar zuverlässig erkannt werden, jedoch bereits eine Distanz von wenigen Metern ausreicht, um eine Detektion unmöglich zu machen. Um dies zu kompensieren, müssten mehrere Sensoren in entsprechenden Abständen aufgestellt werden. Dies ist, besonders in größeren Räumen, jedoch nicht immer möglich. Im Falle des Living Places besteht der Boden zudem aus einzelnen Holzplatten, welche nur lose miteinander verbunden sind. Die Körperschall-Schwingungen würden an den Verbindungsstellen so stark gedämpft werden, dass eine Detektion über mehrere Platten hinweg nicht möglich ist. Für ein System zur Erkennung von gestürzten Personen im Living Place erweist sich diese Technik somit als ungeeignet und wird in dieser Arbeit nicht weiter verfolgt.

Bei den kapazitiven Sensoren hingegen haben sich das CapToolKit und Atmels QTouch® Technologie als äußerst zuverlässig und stabil erwiesen. Letztendlich fiel die Wahl auf das vom CapToolKit verwendete Verfahren, da mit diesem bereits aus bis zu 20 cm eine Annäherung an die Elektrode erkannt werden kann, während dies bei den Sensoren auf Qtouch®-Basis erst kurz vor Berühren der Elektrode möglich ist. Hier zeigt sich deutlich, dass Atmels QTouch® auf die Erkennung von Berührungen ausgelegt ist und nicht auf die Messung von Distanzen. Durch die stärkeren Ausschläge des CapToolKit ist es zudem leichter, Störungen und ungewünschte Einstreuungen in die Daten herauszufiltern. Tabelle 2 fasst die Vor- und Nachteile der kapazitiven Messverfahren noch einmal zusammen.

	RC-Zeitkonstante	QTouch®	CapToolKit
Pro	-Günstig -Einfach aufzubauen -Einfaches Verfahren	-Günstig -Stabil - Unempfindlich gegen Störungen	-Stabil - Unempfindlich gegen Störungen -Detektiert Annäherungen aus bis zu 20cm Entfernung
Contra	-Benötigt stabile Spannung und Umgebung	-Nur für geringe Distanzen -Komplexer Ablauf	-Teurer als die anderen Verfahren

Tabelle 2: Vor- und Nachteile der vorgestellten kapazitiven Messverfahren

3.5 Prototyp der Sensorfläche

Um die kapazitiven Sensoren unter realitätsnahen Bedingungen testen zu können und nicht von etwaigen Veranstaltungen im Living Place abhängig zu sein, wurde eine Unterlage beschafft, deren Größe etwa des im Living Place später verwendeten Teppichs entspricht. Als Material dienten zwei Reihen Malervlies, die durch Klebeband miteinander verbunden wurden. Auf diese Weise konnte eine Testfläche von 2 Meter Kantenlänge erstellt werden. Mit dieser Unterlage wurde ein Testsetup erstellt, an dem verschiedene Konstellationen von Sensoren, welche im folgenden Abschnitt näher beschrieben werden, erprobt werden konnten.



Abbildung 37: Unterlage

Anordnung der Sensoren

Erste Versuche wurden mit einer matrixförmigen Anordnung der Sensorflächen durchgeführt. Hierfür wurden Aluminiumstreifen rasterförmig auf die Unterlage geklebt (Abbildung 37) und an ihren Schnittstellen durch eine Isolierschicht getrennt. Die Sensoren wurden am Rande der Unterlage angebracht und jeweils mit einem Streifen verbunden. Auf diese Weise konnte eine 2-dimensionale Matrix gebildet werden, in der jeder Schnittpunkt der Alustreifen durch zwei Sensoren bestimmt ist. Bei diesem Aufbau kam es jedoch zu Fehlerinterpretationen, wenn sich mehr als eine Person auf der Unterlage befand und zwischen ihnen ein großer Abstand lag. Abbildung 38 illustriert verschiedene Interpretationsergebnisse, dabei symbolisieren rote Punkte Positionen, an denen eine Person erkannt wurde.

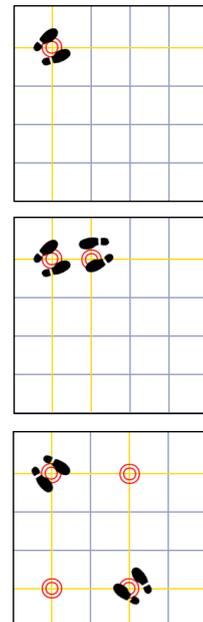


Abbildung 38:
Fehlinterpertation der
Messwerte

Da sich diese Ungenauigkeiten nicht beheben lassen, eignet sich diese Anordnung der Sensoren nur für sehr kleine Flächen oder wenn sichergestellt wird, dass sich nur eine Person im Sensorfeld befindet. Als zufriedenstellende Alternative hat sich jedoch die Anbringung der Sensoren als einzelne Felder direkt unter die Unterlage herausgestellt.

3.6 Software-Bibliotheken zur Bildverarbeitung

Bevor ein Softwaresystem in die Praxis umgesetzt werden kann, muss sich die Frage gestellt werden, wie es am effizientesten realisiert werden kann. So macht es bei einem Bildanalyse-System wenig Sinn, das Rad neu zu erfinden und alle Bildverarbeitungsfunktionen selber zu schreiben, zumal dies tiefgreifende Kenntnisse über die theoretischen Zusammenhänge der Bildverarbeitung voraussetzt und mit einem erheblichen Zeitaufwand verbunden ist. Stattdessen bietet es sich an, eine der zahlreichen kostenpflichtigen oder kostenlosen Bibliotheken zu verwenden. Diese haben die wichtigsten Funktionen bereits implementiert und können in eigene Systeme integriert werden.

OpenCV

Eine der am weitesten verbreiteten kostenlosen Bibliotheken für Bildverarbeitung ist die *Open Computer Vision Library (OpenCV)*. Das *OpenCV* Projekt wurde ursprünglich von *Intel* entwickelt und wurde später von *Willow Garage* übernommen. Die Bibliothek wird unter BSD (Berkeley Software Distribution)-Lizenz veröffentlicht, was bedeutet, dass sie frei eingesetzt und auch in kommerziellen Projekten verwendet werden darf. Lediglich der Copyright-Vermerk muss erhalten bleiben.

OpenCV ist vollständig in C/C++ programmiert und von Anfang an auf eine hohe Verarbeitungsgeschwindigkeit ausgelegt. Im Laufe der Zeit wurden die implementierten Algorithmen immer wieder den aktuellen Forschungsergebnissen angepasst und erweitert. Die Bibliothek umfasst in ihrer aktuellen Version (2.3.1 – August 2011) mehr als 2500 Algorithmen, darunter Funktionen zum maschinellen Lernen, Gesichtsdetektion sowie verschiedene Filter [68].

ITK

Eine weitere quelloffene Bibliothek ist das *Insight Toolkit (ITK)*. Das Projekt entstand 1999 aus einer Ausschreibung der *US National Library of Medicine*. Entwickler der Bibliothek ist ein Konsortium aus mehreren Universitäten und Firmen. Der Entwicklungsschwerpunkt von *ITK* ist die Visualisierung medizinischer Daten, wie sie von Computertomographen oder ähnlichen Geräten geliefert werden. Als Programmiersprache dient C/C++, zusätzlich lässt sich die Bibliothek jedoch an verschiedene Programmiersprachen anbinden. *ITK* wird ebenfalls unter der BSD Lizenz veröffentlicht und unterstützt in der aktuellen Version (3.20 – Juli 2010) Algorithmen zur Segmentierung und Darstellung von Daten, sowie verschiedene Filter (u.a. LaPlace, ColourConversion, Gradienten) [30].

HALCON

Bei *HALCON* handelt es sich um eine weit verbreitete kommerzielle Bildverarbeitungsbibliothek. Entwickler ist die *MVTec Software GmbH*. *HALCON* ist auf die Verwendung im technischen und medizinischen Umfeld ausgelegt. Neben einer hohen Verarbeitungsgeschwindigkeit bietet sie Schnittstellen zu vielen gängigen industriellen Kamerasystemen. Des Weiteren lässt sich die Bibliothek mit den Programmiersprachen C/C++, C#, Visual Basic, .NET sowie Delphi verwenden. Die Anbindung erfolgt wahlweise über eine API-Schnittstelle (Abb. 39) oder der mitgelieferten Entwicklungsumgebung *HDevelop*. Da komplexe Abläufe bereits zusammengefasst als sogenannte *procedures* vorliegen, erlaubt *HDevelop* schnelles Prototyping und eine komfortable Erstellung von komplexen Bildverarbeitungssystemen. Nach Abschluss der Entwicklung lassen sich die verwendeten *procedures* direkt in die entsprechende Programmiersprache exportieren.

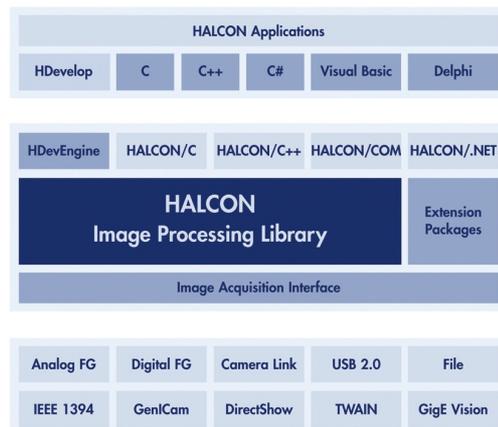


Abbildung 39: HALCON-Architektur
(Quelle: [38])

Neben der normalen Version von *HALCON*, welche für Shared-Memory- und Multicore-Systeme ausgelegt, steht noch eine spezielle Version für leistungsschwächere Systeme zur Verfügung. *HALCON embedded* erlaubt die Integration von Bildverarbeitungssystemen auf verschiedenen eingebetteten und mobilen Systemen. Neben vielen gängigen Prozessorarchitekturen wird eine große Anzahl an Betriebssystemen, wie Windows CE, VxWorks oder Linux unterstützt. Beide *HALCON* Varianten bieten in der aktuellen Version 9 mehr als 1400 unterschiedliche Operationen und Abläufe [37]. So sind bereits komplexe Vorgänge wie 3D- oder Stereo-Kamera-Kalibrierung, Linienextraktion und Objektmatching fertig hinterlegt. Die Kosten für *HALCON* liegen bei 1.750 Euro für Hochschulen und 5.500 Euro für kommerzielle Kunden (Stand: August 2010). Die Lizenzierung kann wahlweise über die MAC-ID des PCs, was das Produkt an einen einzelnen Rechner bindet, oder über einen USB-Key geschehen.

3.6.1 Resultat

Alle aufgeführten Bibliotheken bieten einen Funktionsumfang, der das Arbeiten erheblich erleichtert und wären somit für die weitere Verwendung im Projekt geeignet. Jedoch ermöglicht die frei verfügbare Demo-Version von HALCON nur einen sehr eingeschränkten Testbetrieb. So lassen sich zwar eigene Programme in der Entwicklungsumgebung *HDevelop* erstellen und Beispielprogramme ausführen, die Export-Funktion sowie die Schnittstellen-API zu verschiedenen Programmiersprache steht leider nicht zur Verfügung.

Ein weiterer entscheidender Faktor, welcher gegen HALCON spricht, ist der hohe Anschaffungspreis. *OpenCV* und *ITK* haben den Vorteil, dass beide Bibliotheken kostenlos erhältlich sind und in kommerziellen Projekten verwendet werden dürfen. Die Entscheidung fiel schlussendlich auf *OpenCV*. Ausschlaggebend hierfür war die weite Verbreitung der Bibliothek und die damit zusammenhängenden umfangreichen Dokumentationen und Anleitungen. Tabelle 3 zeigt eine Übersicht der Vor- und Nachteile der drei Bibliotheken.

	OpenCV	ITK	HALCON
Pro	<ul style="list-style-type: none"> - kostenlos - sehr umfangreich - gute Dokumentation - viele Anleitungen - einfacher Aufbau 	<ul style="list-style-type: none"> - kostenlos 	<ul style="list-style-type: none"> - sehr umfangreich - Anbindung an viele Kamerasysteme
Contra	<ul style="list-style-type: none"> - Funktionen teilweise schwer zu finden 	<ul style="list-style-type: none"> - Kamerasysteme nur über Umwege integrierbar 	<ul style="list-style-type: none"> - kostenpflichtig - Test vieler Funktionen in Demo-Version nicht möglich

Tabelle 3: Vergleich *OpenCV*, *ITK* und *HALCON*

4 Entwicklung eines kapazitiven Sensors

Für den Aufbau des kapazitiven Sensorsystems werden mehr Sensoren benötigt, als das CapToolKit von Raphael Wimmer bietet. Es bestand somit die Wahl, weitere Sets zu erwerben oder eigene Sensoren anzufertigen. Da die Sets im Handel nicht mehr erhältlich sind, mussten diese eigenständig nachgebaut werden. Dies wäre problemlos möglich, da die Schaltpläne und Software des CapToolKits kostenlos erhältlich sind [57]. Letztendlich wurde jedoch entschieden, eigene Sensoren zu entwickeln, die speziell auf die Bedürfnisse im Living Place und den daran beteiligten Projekten angepasst sind.

Anforderung an die Sensor-Module war, dass sie

- möglichst klein
- erweiterbar
- einfach anzusteuern
- universell einsetzbar
- und kostengünstig in der Herstellung

sein sollten.

Nach diesen Vorgaben wurde ein Sensor entwickelt, dessen Aufbau in den folgenden Kapiteln beschrieben wird.

4.1 Technische Umsetzung

4.1.1 Stromversorgung

Zur Spannungsbegrenzung wird ein Low-Drop Linearregler vom Typ AP1117 verwendet. Dieser bietet eine stabile Ausgangsspannung von 5V bei einer maximalen Belastung von 1A. Die Verlustspannung bei Vollast beträgt maximal 1.4V wodurch sich eine mögliche Eingangsspannung im Bereich von 6.4V - 18V ergibt. Um Störungen in der Eingangsspannung zu unterdrücken und auch bei schwankendem Strombedarf eine stabile Ausgangsspannung zu erhalten, sind vor dem Ein- und Ausgang des Spannungsreglers je ein 100nF Kondensator sowie ein 10µF/16V Tantal-Elko vorgesehen.

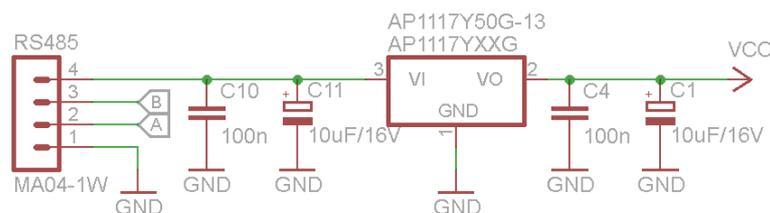


Abbildung 40: Stromversorgung

4.1.2 Mikrocontroller

Als Mikrocontroller kommt ein ATmega88PA im MLF32 Gehäuse zum Einsatz. Bei diesem handelt es sich um einen 8-Bit Prozessor der Firma Atmel. Er besitzt neben einem UART-Modul zwei 8-Bit und einen 16-Bit Zähler, einen 10-Bit Analog-Digital Wandler mit 8 Kanälen sowie eine Unterstützung für einen Bootloader, verschiedene externe Interrupt Möglichkeiten und mehrere Standby Modi.

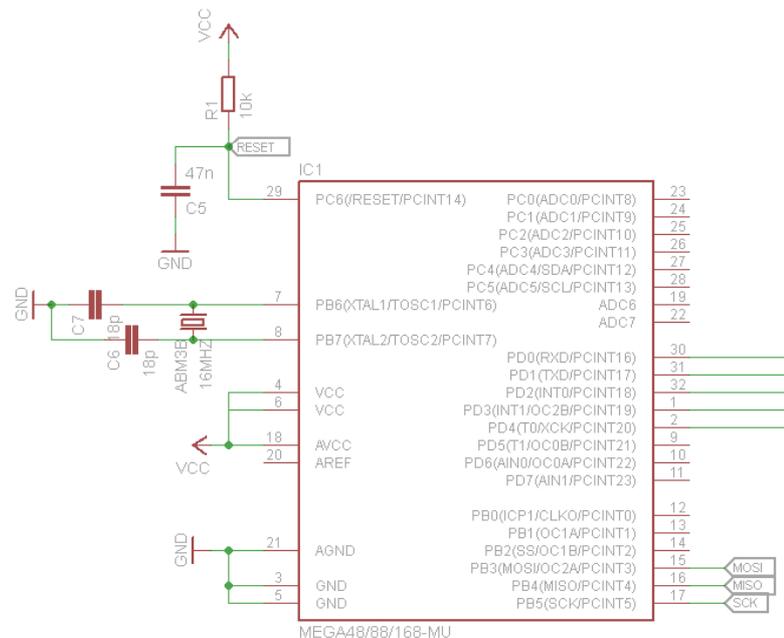


Abbildung 41: Mikrocontroller

Der Controller wurde gewählt, da er bei einer Versorgungsspannung von 5V und maximaler Taktrate von 20Mhz einen Eigenverbrauch von lediglich 9mA aufweist. Als Taktquelle dient ein 16Mhz Quarz mit zwei 18pF Lastkondensatoren. Die Resetleitung wird durch einen 10k Ohm Widerstand dauerhaft auf 5V gehalten. Zusätzlich ist zwischen der Resetleitung und Masse ein 47nF Kondensator angebracht. Dieser bewirkt, dass die Resetleitung nach anlegen der Versorgungsspannung einige Millisekunden lang den Controller im Resetzustand hält, bis sich die Spannung stabilisiert hat.

4.1.3 RS-485 Bus

Eine der Anforderungen war, dass ein Projekt, welches die Sensoren verwendet, leicht um weitere Module ausgebaut werden kann. Aus diesem Grunde wurde für die Verbindung der Sensoren ein Bussystem gewählt. Dieses erlaubt es, in einen existierenden Bus weitere Module einzuhängen, ohne zusätzliche Kabel verlegen zu müssen. Durch die Verwendung eines Busses ergibt sich zudem ein wesentlich sauberer Aufbau, da nur eine geringe Anzahl an Leitungen für die Daten und Stromversorgung benötigt werden.

Die Kommunikation der Module erfolgt über den in Kapitel 3.4.2 bereits eingesetzten RS-485 Bus. Aufgrund des symmetrischen Aufbaus der Leitungen und der differentiellen Signalübertragung weist er eine hohe Toleranz gegenüber elektromagnetischen Störungen auf und wird daher häufig im industriellen Umfeld eingesetzt. Da es sich, anders als bei Ethernet oder RS-232, um ein Bussystem handelt, sollten die Leitungsenden durch Widerstände abgeschlossen werden. In der Regel wird hierfür ein passiver Abschluss in Form eines $120\ \Omega$ Widerstandes verwendet, welcher an beiden Enden des Busses zwischen die Datenleitungen geschaltet wird. Um undefinierte Zustände auf dem Bus zu vermeiden, wird an einem Ende häufig noch ein zusätzliches Bias-Netzwerk installiert (Abb. 42). Dieses besteht aus zwei Widerständen, welche die Datenleitungen des Busses bei inaktiven Transceivern auf einem definierten Spannungspegel halten. Auf diese Weise wird vermieden, dass die Empfänger durch Störeinstrahlungen ungültige Daten empfangen. Genaue Berechnungen zur Größe der Widerstände des Bias-Netzwerk finden sich in [62].

Bei den in dieser Arbeit entwickelten Modulen dient als Schnittstelle zwischen Mikrocontroller und dem RS-485 Bus ein Transceiver vom Typ ST485CDR. Dieser wurde gewählt, da er eine maximale Datenrate von 2.5Mbps erlaubt und bei ausgeschaltetem Treiber lediglich $300\ \mu\text{A}$ Strom verbraucht. Des Weiteren sind die integrierten Treiber gegen Überlastung und Kurzschlüsse auf dem Bus geschützt. Zudem ist es mit dem Transceiver möglich bis zu 64 Geräte an einem Bus zu betreiben, was somit gleichzeitig die maximale Anzahl an Sensoren festlegt. Da für die Übertragung der Daten nur zwei Datenleitungen vorhanden sind, erfolgt die Kommunikation im Halb-Duplex Modus, d.h. jeder Transceiver kann entweder Daten senden oder empfangen. Zum Einschalten des Sendetreibers im Transceiver steht jeweils ein Anschluss zur Verfügung. Einer schaltet bei einem High-Pegel den Sendetreiber ein, während der andere den Empfangstreiber bei einem Low-Pegel aktiviert. Dies hat den Vorteil, dass man beide Anschlüsse mit nur einem Ausgang des Mikrocontrollers verbinden kann. Durch Umschalten des Ausgangs zwischen High und Low lässt sich somit leicht die Datenrichtung des Transceivers festlegen. Des Weiteren besitzt der Transceiver zwei Anschlüsse für die Daten - einen Ausgang und einen Eingang. Diese sind mit den Anschlüssen des UART im Mikrocontroller verbunden. Dies erleichtert die Entwicklung der Software, da es möglich ist, die Eigenschaften des UART-Moduls zur Datenübertragung zu verwenden.

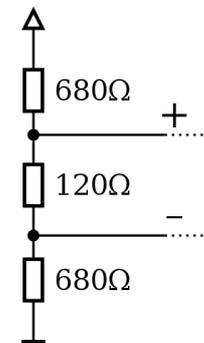


Abbildung 42: Abschluss mit Bias-Netzwerk

4.1.4 Sensor

Die Messeinheit besteht aus einem CMOS Timer vom Typ 555. Dieser zeichnet sich durch einen sehr geringen Eigenstromverbrauch von max. 250µA bei 5V Betriebsspannung sowie eine Arbeitsfrequenz von bis zu 3Mhz aus. In den Modulen wird der Timer verwendet, um die Elektrode kontinuierlich zwischen zwei Spannungspotentialen schwanken zu lassen. Hierfür schaltet der Timer seinen Ausgang auf High und lädt die Elektrode über einen Widerstand bis auf 2/3 der Betriebsspannung, also ca. 3,3V, auf. Anschließend kippt der Ausgang und entlädt die Elektrode auf 1/3 der Spannung. Dieser Ablauf wiederholt sich fortlaufend, so lange am Reset-Eingang kein Low-Signal anliegt.

Die Zeit, die benötigt wird, um die Elektrode auf das entsprechende Potential zu bringen, ist abhängig von der Kapazität der Elektrode und dem Strom, mit dem diese geladen wird. Letzteres lässt sich durch den Widerstand zwischen Ausgang des Timers und Elektrode regeln. Um auch bei Elektroden mit einer

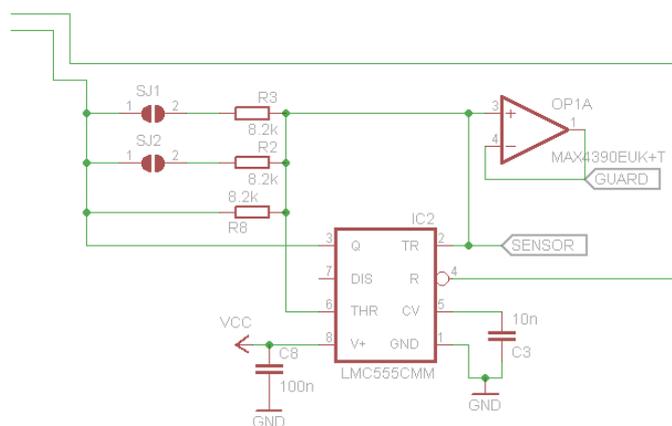


Abbildung 43: Sensor

großen Eigenkapazität eine genaue Messung zu ermöglichen, sind drei Widerstände parallel geschaltet, von denen sich zwei bei Bedarf über Lötbrücken zuschalten lassen (Abbildung 43). Auf diese Weise verringert sich der Gesamtwiderstand zwischen Timer und Elektrode, was einen höheren Strom und somit ein schnelleres Laden und Entladen der Elektrode zur Folge hat.

Der Ausgang des Timers ist zusätzlich mit einem Trigger-Eingang des Mikrocontrollers verbunden und erlaubt so die Überprüfung, ob die Elektrode auf- oder entladen wird, sowie die Zählung der Ladevorgänge. Damit der Timer bei Bedarf abgeschaltet werden kann, ist auch die Resetleitung mit dem Mikrocontroller verbunden.

4.1.5 Guard

Die Messelektrode eines Sensormoduls verhält sich wie eine omnidirektionale Antenne. Das bedeutet, sie detektiert Änderungen des elektrischen Feldes unabhängig von ihrer Ausrichtung. So werden beispielsweise bei einem frei im Raum stehenden Gerät sowohl Personen detektiert, die sich vor dem Gerät

befinden, wie auch solche, die sich dahinter bewegen. Damit dieses Gerät nur noch auf die Person vor ihm reagiert, ist es nötig, das elektrische Feld auf einen bestimmten Bereich zu begrenzen. Hierfür wird ein sogenannter *Guard* verwendet [56].

Der Guard stellt eine Möglichkeit zur Abschirmung der Elektrode dar. Hierbei handelt es sich um eine weitere Elektrode, im folgenden Abschirmungselektrode bezeichnet, die auf dem selben Potential wie die Messelektrode liegt. Wird die Messelektrode mit der Abschirmelektrode umgeben, beeinflussen die umgebenden elektrischen Felder nur die äußere Abschirmelektrode und nicht die in der Mitte liegende Messelektrode. Da beide das selbe Spannungspotential besitzen, kann sich auch zwischen ihnen kein Feld aufbauen, wodurch die Kapazität der Messelektrode nicht beeinflusst wird. Damit dies möglich ist, benötigt die Abschirmelektrode einen Treiber, der die Spannung möglichst ohne Verzögerung an die der Messelektrode anpasst und auch bei externen Einflüssen stabil hält. Hierfür wird ein Operationsverstärker des Typ MAX4390 eingesetzt. Bei diesem handelt es sich um einen High-Speed Rail-2-Rail OpAmp mit einer Anstiegsgeschwindigkeit von $500\text{V}/\mu\text{S}$. Eine hohe Umschaltgeschwindigkeit und Anstiegszeit sind erforderlich, da der Timer die Spannung der Messelektrode mit bis zu 3Mhz zwischen 3.3V und 1.6V ändert. Bei einer zu geringen Anstiegs- und Umschaltgeschwindigkeit würde die Ausgangsspannung des Operationsverstärkers sich um den Mittelwert der Messelektroden-Spannung einpendeln. Dies hätte zur Folge, dass zwischen den Elektroden ein Potentialunterschied besteht, der die Messungen beeinträchtigen würde. Der Operationsverstärker ist als Spannungsfolger geschaltet und entkoppelt somit die Abschirmelektrode von der Messelektrode.

4.2 Kommunikationsprotokoll

Die Kommunikation der Module untereinander und mit einem Rechner erfolgt asynchron durch ein serielles Übertragungsprotokoll. Als Grundgerüst dient der Datenrahmen des RS-232 Standards.

Das aufbauende Übertragungsprotokoll ist so gehalten, dass es möglichst allgemein verwendbar ist, gleichzeitig jedoch unnötigen Overhead auf dem Bus vermeidet. Es arbeitet mit einer paketorientierte Übertragung. Jedes Paket beginnt mit einem Startbyte (üblicherweise #), welches fehlerhafte Übertragungen verhindern soll. Anschließend folgt die Größe des Pakets.

Das eigentliche Paket besteht aus einer Quell- und Zieladresse sowie dem auszuführende Befehl. Für Parameter o.ä. können zusätzlich noch bis zu 50 Bytes übertragen werden. Den Abschluss eines Pakets bildet die Prüfsumme. Bei dieser handelt es sich um einen 8-Bit großen Prüfwert nach dem *cyclic redundancy check* (CRC) Verfahren, welcher über Quell- und Zieladresse, Befehlsbyte sowie alle Datenbytes gebildet wird. Als zulässige Zieladressen gilt der Bereich von 1 (0x01) bis 254 (0xFE). Adresse 255 (0xFF) wird als Broadcastadresse behandelt.

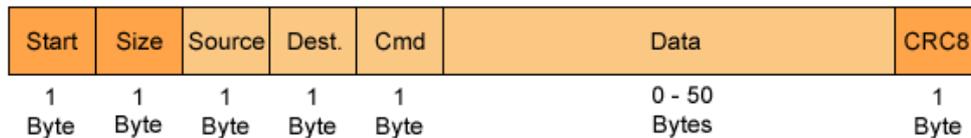


Abbildung 44: Datenpaket

4.3 Software

4.3.1 Bootloader

Der Bootloader ist ein spezieller Teil der Firmware, welcher nach Anlegen der Versorgungsspannung oder nach einem Reset vor dem eigentlichen Programm ausgeführt wird. Er dient dazu, die Firmware der Module über den Bus zu aktualisieren. Somit lässt sich, ohne die Module auszubauen, ein neues Programm aufspielen.

Nach dem Start des Bootloaders, prüft dieser zunächst, ob eine gültige Firmware vorhanden ist. Ist dies der Fall, wartet der Bootloader ca. 500mS auf eine Nachricht vom Bus-Master, bevor das eigentliche Programm gestartet wird. Wird keine gültige Firmware gefunden, verharrt der Bootloader in der Warteschleife. Ein neues Programm kann erst aufgespielt werden, nachdem dem Bootloader mitgeteilt wurde, das er sich in den Programmiermodus versetzen soll. Diese Sicherheitsvorkehrung soll verhindern, dass ein Programm versehentlich überschrieben wird. Um den Programmiermodus zu aktivieren, wird eine entsprechende Nachricht an das gewünschte Modul verschickt. Das Modul versetzt sich anschließend in den Programmiermodus und antwortet mit einem Status sowie der Versionsnummer der Bootloaders. Hierbei wird der Countdown, der zum Starten des Hauptprogrammes führen würde, im entsprechendem Modul angehalten. Nach Aktivieren des Programmiermodus kann somit beliebig Zeit verstreichen, bevor das eigentliche Update beginnt.

Da die Module aus Kompatibilitätsgründen standardmäßig mit einer Busgeschwindigkeit von 38.4kBit/s arbeiten, lässt diese sich nach Aktivieren des Programmiermodus auf einen höheren Wert setzen, um ein Hochladen der Firmware zu beschleunigen.

Die Daten eines Programms liegen auf dem PC im sogenannten Intel-Hex-Format vor. Bei diesem handelt es sich um ein Dateiformat, in dem Binärdaten als Hexadezimalzahlen kodiert gespeichert werden. Jede Zeile einer Datei enthält neben einer Typenbezeichnung, eine Speicheradresse und üblicherweise 16 Bytes an Daten, welche durch eine Checksumme gesichert sind. Da die Daten für ein neues Programm in mehreren Paketen auf das Modul übertragen werden müssen, ist es sinnvoll, sich mit der Paketstruktur nahe an dem Intel-Hex-Format zu orientieren. Somit wird eine leichte Konvertierung zwischen Datei und Datenpaket gewährleistet. Um unnötigen Overhead zu vermeiden, ist es jedoch unabdingbar, einige Teile einer Dateizeile vor der Übertragung zu entfernen. Für das Bootloader des Moduls sind lediglich die Speicheradresse sowie die jeweiligen Daten von Interesse. Jedes Datenpaket besteht somit aus einer 16-Bit großen Speicheradresse und bis zu 48 Byte an Daten, die an die entsprechende

Adresse im Speicher geschrieben werden sollen. Jedes empfangene Datenpaket wird von dem Modul mit Rückmeldung quittiert. Diese gibt an, ob bei der Verarbeitung der Daten ein Fehler aufgetreten ist. Aus diesem Grund müssen alle am Bus angeschlossenen Module einzeln programmiert werden. Würden sich mehrere Module im Programmiermodus befinden, könnte es beim Bestätigen der Pakete zu Datenkollisionen auf dem Bus kommen. Nachdem ein Programm vollständig übertragen wurde, kann der Programmiermodus wieder verlassen und das Hauptprogramm gestartet werden.

4.3.2 Firmware

Die Software der Module wurde in mehrere Sektionen unterteilt, dessen Funktionen in den nachfolgenden Abschnitten näher erläutert werden. Das Bus-Interface behandelt alle Funktionen, welche zur Kommunikation mit dem Master nötig sind. Durch die Nachrichten-Queue wird das Bus-Interface vom Rest des Systems entkoppelt. Sie speichert und verwaltet ein- und ausgehende Nachrichten, bis diese verarbeitet wurden. In der Befehls-Sektion werden alle eingehenden Befehle verarbeitet. Der Bereich Device Information enthält Informationen über Seriennummer, ID und Softwareversion des Moduls. Die Sektion Messung umfasst alle Funktionen, die zur Erfassung der Messwerte nötig sind.

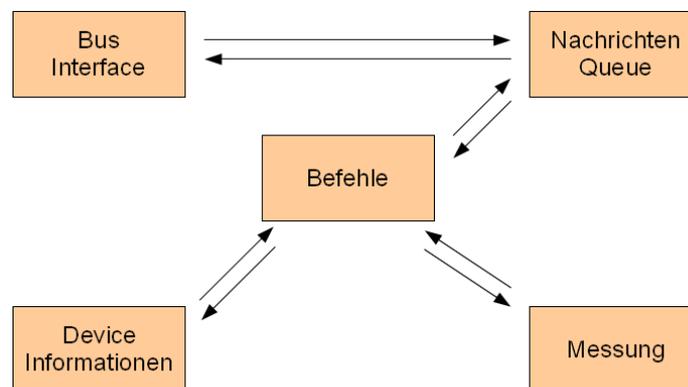


Abbildung 45: Softwareaufbau des Sensormoduls

Bus Interface

Das Bus-Interface regelt die Kommunikation mit dem Bus-Master sowie anderen Modulen. Es sorgt zudem für die Einhaltung des in Kapitel 4.2 beschriebenen Protokolls. Damit kein Byte eines Pakets verloren geht, arbeitet der Datenempfang komplett Interrupt-basiert. Hierfür dient ein einfacher Zustandsautomat, wie in Abbildung 46 dargestellt. Nachdem ein gültiges Startzeichen empfangen wurde, wird ein Wechsel des Zustandes ausgelöst. Das anschließend eintreffende Byte legt fest, wie viele Daten noch folgen und

überführt den Automaten in den Empfangszustand für die Paketdaten. Dieser bleibt so lange aktiv, bis die entsprechende Anzahl an Daten empfangen wurde. Abschließend wird der empfangene CRC-Wert mit dem intern berechneten verglichen. Sind beide identisch, wird das Datenpaket an die Message-Queue weitergeleitet und der Automat in seinen Ausgangszustand versetzt. Um zu verhindern, dass der Automat durch einen zufälligen Verbindungsabbruch oder andere Umstände in einem Zustand hängen bleibt, wird er nach einer festgelegten Zeit automatisch von einem Timer in seinen Ausgangszustand versetzt.

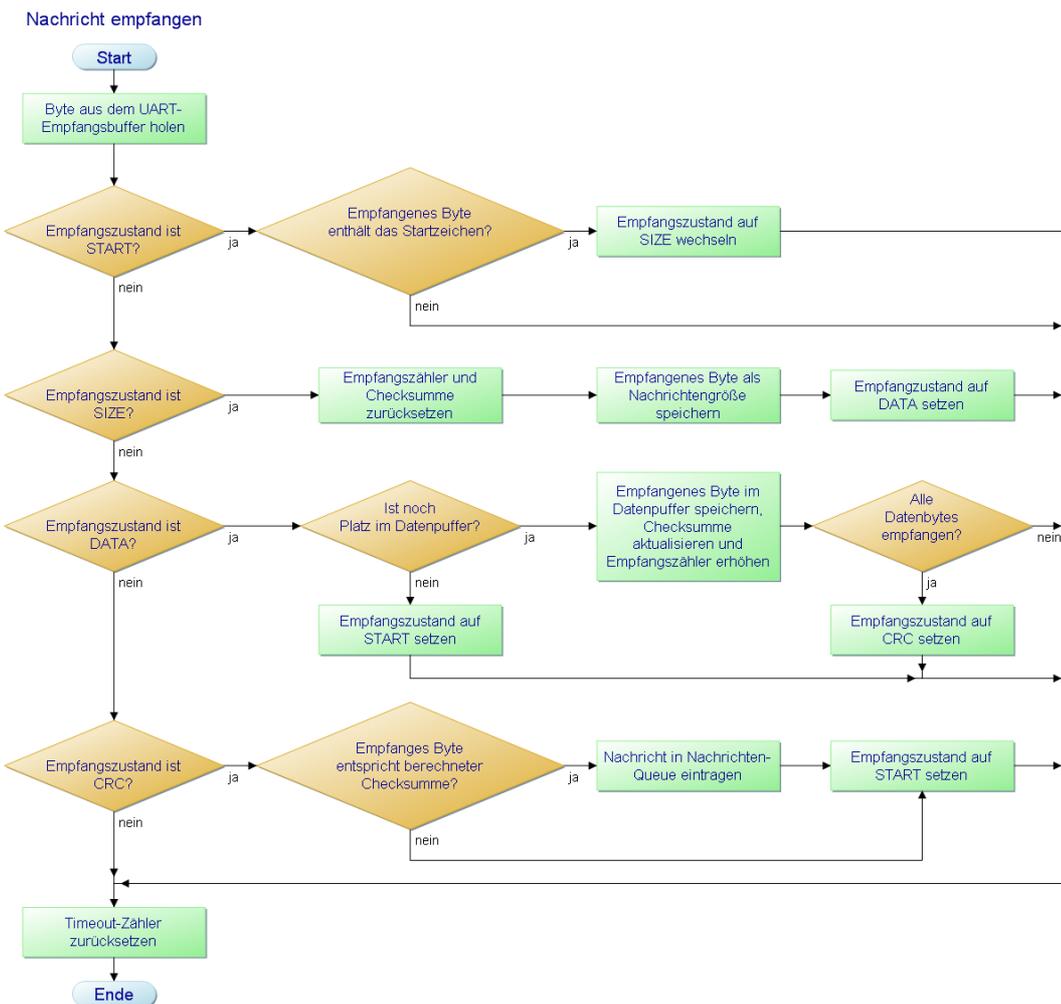


Abbildung 46: Sensor-Modul - Empfang einer Nachricht

Das Übertragen von Daten übernimmt eine Funktion, welche aus der Ziel- und Quelladresse, dem zu sendenden Befehl sowie den optionalen Daten und deren Größe die Checksumme berechnet, ein Datenpaket zusammensetzt und dieses auf den Bus überträgt. Damit es nicht zu Problemen mit der Ansteuerung des RS485-Transceivers kommt, blockiert diese Funktion, bis die Übertragung abgeschlossen ist.

Nachrichten Queue

Die Nachrichten-Queue entkoppelt den Interrupt basierten Teil des Bus-Interface von dem Rest des Systems. Sie besitzt zwei FIFO-Speicher in denen Nachrichten, welche empfangen wurden oder verschickt werden sollen, zwischengespeichert werden. Empfangene Nachrichten werden durch das Bus-Interface in die Nachrichten-Queue eingetragen. Parallel dazu wird im Hauptprogramm kontinuierlich überprüft, ob Nachrichten vorliegen. Ist dies der Fall, wird die älteste Nachricht an die Befehls-Sektion zur Bearbeitung weitergeleitet. Dies erlaubt es, dem Modul Nachrichten zu senden, während es mit anderen Aufgaben beschäftigt ist.

Entsprechend ist der Ablauf beim Versenden von Nachrichten. Hierbei schreibt die Befehls-Sektion die Daten in den entsprechenden Speicher der Nachrichten-Queue. Sobald es möglich ist, werden diese ausgelesen und an das Bus-Interface weitergeleitet, welches anschließend die Übertragung einleitet.

Befehls-Sektion

Die Befehls-Sektion ist die zentrale Komponente des Systems. Sie dient zur Interpretation und Abarbeitung der empfangenden Nachrichten. In ihr werden alle eingehenden Nachrichten und Befehle auf Plausibilität überprüft und ausgeführt oder verworfen. Hierbei wird verifiziert, ob die Zieladresse der Nachricht mit der eigenen Id des Moduls übereinstimmt oder es sich um eine Broadcast-Nachricht handelt. Zudem wird sichergestellt, dass die Nachricht einen gültigen Befehl enthält und die Anzahl der Datenbytes mit den erwarteten übereinstimmt. Liefern alle Prüfungen ein positives Ergebnis, wird die Nachricht entsprechend dem enthaltenen Befehls ausgeführt – ansonsten verworfen. Anhang B enthält eine komplette Auflistung der verfügbaren Befehle und deren Parameter.

Messung

Der Bereich der Messung enthält alle Funktionalitäten, welche zum Einstellen, Ansteuern und Auswerten des eigentlichen Sensors nötig sind. Eine Messung hat immer den gleichen Ablauf, unabhängig davon ob sie einmal oder kontinuierlich ausgeführt wird. Um das Modul auch nach dem Start einer Messung

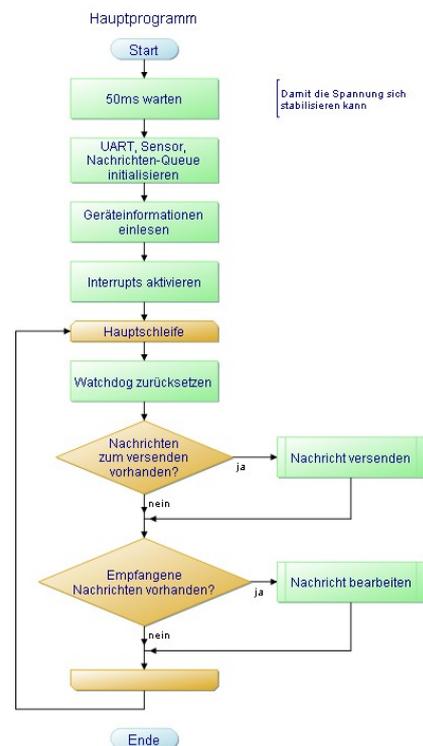


Abbildung 47: Ablauf des Hauptprogramms

reaktionsfähig zu halten, wird ein asynchroner Timer verwendet. Dieser wird so eingestellt, dass er durch einen externen Takt – welcher in diesem Fall durch den Sensor erzeugt wird – betrieben wird. Zusätzlich wird ein zweiter Timer auf die Dauer der Messung eingestellt und gestartet. Bei Erreichen der eingestellten Dauer wird ein Interrupt ausgelöst und der Sensor deaktiviert. Anschließend braucht nur noch der Zählerstand des asynchronen Timers ausgelesen werden und ggf. eine neue Messung gestartet werden.

Device Informationen

Diese Komponente liefert Informationen zur Identifizierung des Moduls. Sie enthält Funktionen zum Auslesen und Einstellen der ID, Seriennummer sowie der aktuellen Softwareversion. Da es sich bei diesen Informationen um wichtige Daten handelt, ohne die eine reibungslose Funktionalität der Sensormodule und unter Umständen des gesamten Bussystems nicht mehr möglich ist, sind diese Informationen in redundanter Ausführung im EEPROM des Mikrocontrollers abgelegt. Dies erlaubt es, defekte Speicherzellen oder Lesefehler zu erkennen und entsprechend zu reagieren. Hierfür wird der Block mit den oben genannten Daten an drei verschiedenen Stellen im EEPROM gespeichert. Beim Laden der Daten werden alle 3 Blöcke gelesen und miteinander verglichen. Weisen alle drei identische Daten auf, traten beim Laden keinerlei Fehler auf. Weicht jedoch einer der Datenblöcke von den anderen beiden ab, kann davon ausgegangen werden, dass dessen Daten nicht gültig sind. In dem Fall wird der entsprechende Block ignoriert. Besitzen jedoch alle drei Blöcke unterschiedliche Daten, kann nicht mehr sicher bestimmt werden, welche Daten gültig sind. In diesem Fall wird versucht, den Fehler durch ein Neustart des Sensormoduls zu beheben.

4.4 Evaluierung

Resultat der beschriebenen Entwicklung ist das in Abbildung 48 gezeigte Sensormodul. Mit seine geringen Abmessungen von 25 x 18 mm und einer Dicke von knapp 4mm ist es nur unwesentlich größer als die im CapToolKit eingesetzten Sensoren. Dank des verwendeten Datenbusses lassen sich mehrere Module zu einem System zusammenschließen. Dies erlaubt eine übersichtliche Anordnung des Aufbaus, da nicht für jeden Sensor ein eigenes Kabel benötigt wird. Darüber hinaus wird durch den Datenbus nur ein Anschluss am PC belegt.

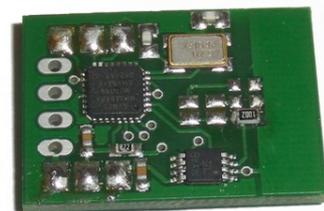


Abbildung 48: Entwickelter kapazitiver Sensor

Infolge der Struktur des Datenbusses und der dadurch variablen Anzahl an Modulen ist jedoch eine komplexe Ansteuerung nötig. Da es zu Kommunikationsstörungen kommen würde, wenn mehr als ein

Modul Daten über den Bus überträgt, übernimmt der angeschlossene PC die Arbitrationslogik auf den Bus. Zudem ist aufgrund des geringen Datenspeichers der Module, anders als beim CapToolKit, keine Filterung und Vorverarbeitung der Daten möglich. Von den Sensoren können daher nur die reinen Rohdaten abgefragt werden. Eine eventuelle Filterung muss softwaretechnisch im PC vorgenommen werden.

Zur einfacheren Konfiguration der Module wurden zusätzlich zwei Java-Anwendungen entwickelt.

Mit Hilfe des *Firmware-Tools* ist es möglich, die Module zu verwalten sowie Einstellungen eines Moduls den Wünschen nach anzupassen (Abbildung 49). Hierfür stehen beispielsweise Funktionen zum Auflisten aller angeschlossenen Sensoren sowie eine grafische Anzeige des Sensorwertes zur Verfügung. Darüber hinaus lässt sich mit dieser Anwendung eine andere Firmware auf das Modul aufspielen (Abbildung 50). Dies geschieht wahlweise durch einen Programmier-Adapter oder über den integrierten Bootloader (Kapitel 4.3.1).

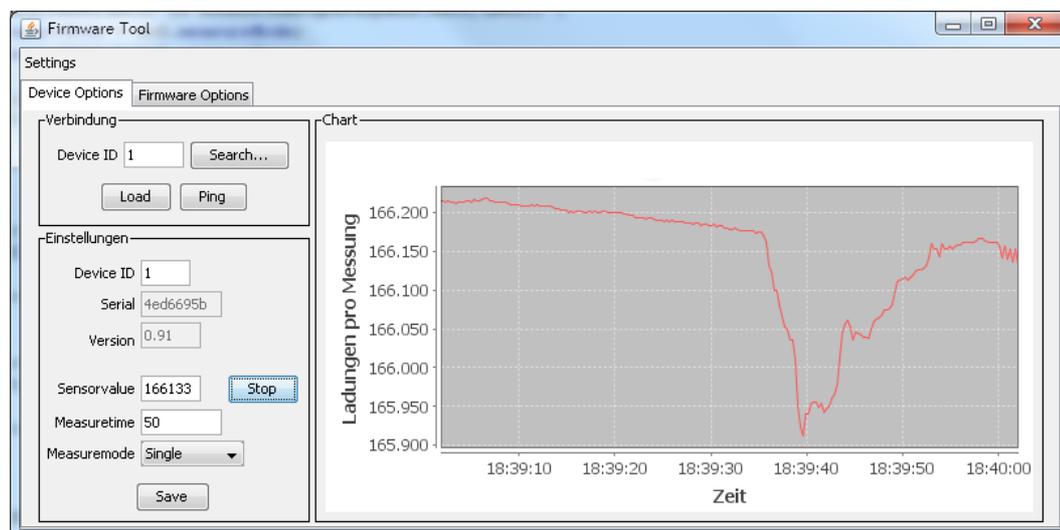


Abbildung 49: Firmware-Tool - Konfigurationsoberfläche

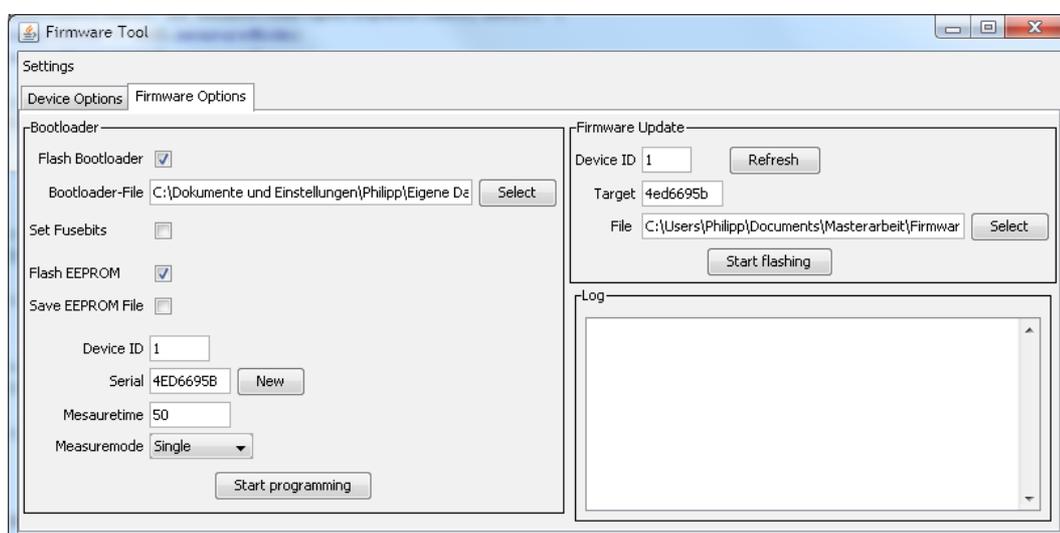


Abbildung 50: Firmware-Tool - Updateoberfläche

Eine zweite Anwendung erlaubt die komfortable, grafische Anzeige der Daten mehrerer Sensoren. Dies ist speziell beim Vergleich verschiedener Sensoren hilfreich. Hierfür bieten die einzelnen Graphen unabhängig voneinander mehrere Skalierungs- und Zoomfunktionen. Auch Langzeittest sind auf diese Weise möglich. Für weitere Auswertungen lassen sich die aufgezeichneten Daten jedes Sensors separat in einer CSV-Datei abspeichern.

Abbildung 51 zeigt die Anwendung mit dem Werteverlauf eines Sensors mit der ID 1. Anhand dieses Bildes ist auch sehr gut der Ausschlag der Werte bei Annäherung an die Elektrode zu erkennen.

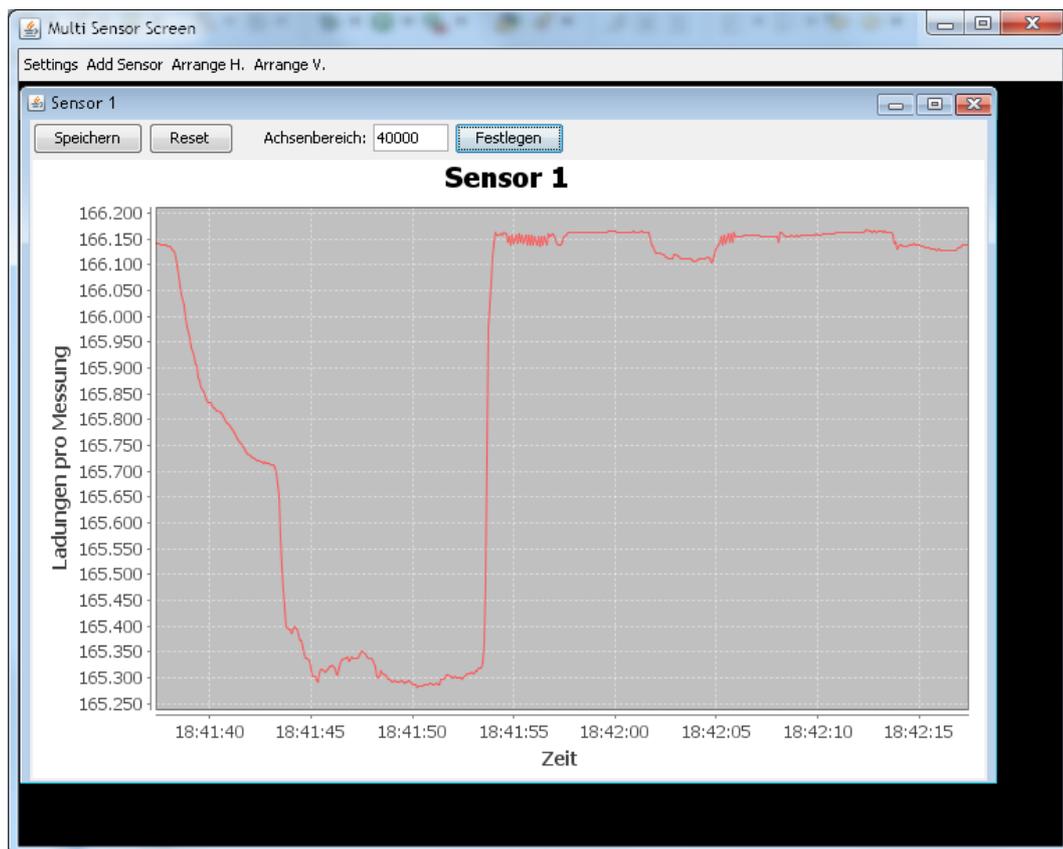


Abbildung 51: Multi-Sensor-Tool - Aufzeichnung des Werteverlaufs eines Sensors

5 Konzept und Architektur der Software

5.1 Aufbau der Software

In diesem Kapitel wird, aufbauend auf die in Kapitel 3 und 4 erlangten Erkenntnisse, ein System zur zuverlässigen Erkennung von Stürzen entwickelt. In den ersten Abschnitten wird die Struktur des Gesamtsystems und dessen Komponenten erläutert. Anschließend wird näher auf die Vorverarbeitung und Interpretation der Bild- und Sensordaten eingegangen. Den Abschluss des Kapitels bildet eine Risikoanalyse des entworfenen Systems.

5.2 Systembeschreibung

Um ein durchdachtes System zu erhalten, ist es unabdingbar, dass bereits in der Entwurfsphase der Software wichtige Aspekte, wie

- leichte Wartbarkeit
- einfache Konfigurierbarkeit
- problemlose Erweiterbarkeit

berücksichtigt werden.

Die Flexibilität von Softwaresystemen lässt sich unter anderem durch ihre Konfigurierbarkeit beeinflussen. Die Konfiguration eines Programmes lässt sich sowohl während der Entwicklungsphase, z.B. durch direktes Festlegen im Programmcode, als auch zur Laufzeit festlegen. Je leichter die Einstellungen eines Programmes geändert werden können, desto einfacher ist die Bedienung für den Benutzer. Eine einfache Konfigurierbarkeit lässt sich erreichen, indem alle relevanten Einstellungen der Software in einer übersichtlichen und leicht zu bearbeitenden Form gespeichert werden. Als Ablageort bieten sich daher Datenbanksysteme oder auf Textdateien basierende Systeme, wie XML oder CSV an. Zusätzlich trägt die Möglichkeit, Einstellungen während der Laufzeit zu verändern, sehr zum Bedienungskomfort einer Software bei.

Eine leichte Wartbarkeit ist um so wichtiger, je länger eine Software eingesetzt werden soll. Da sich nicht absehen lässt, wie lange die entwickelte Software Verwendung finden wird, es jedoch wahrscheinlich ist, dass sie zu einem späteren Zeitpunkt noch einmal verändert wird oder einzelne Teile der Software genutzt werden sollen, ist es von besonderer Relevanz, die Wartbarkeit und Erweiterbarkeit so leicht wie möglich zu gestalten. Um dies zu erreichen, sollte ein möglichst stark gegliederter und modularer Aufbau gewählt werden.

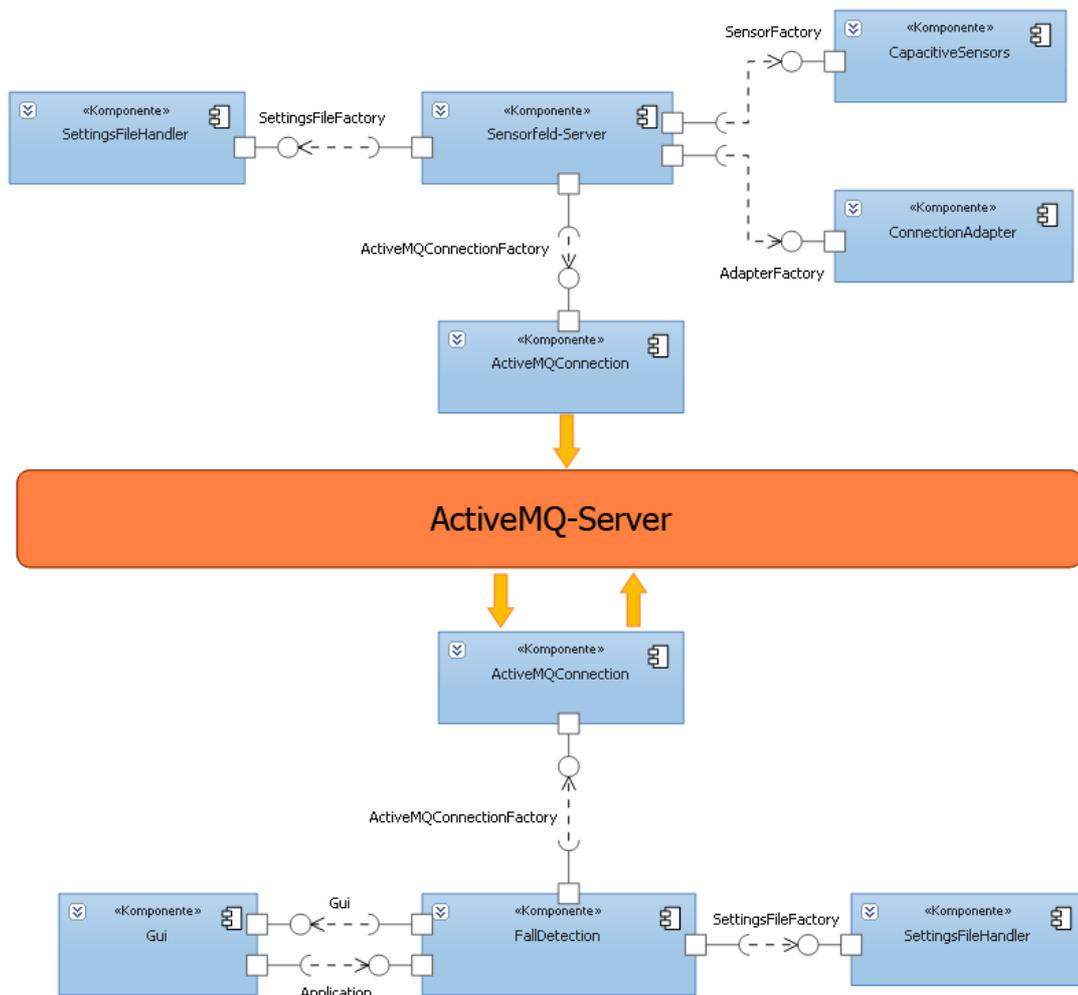


Abbildung 52: Aufteilung des Systems in einzelne Komponenten

Ausgehend von diesen Anforderungen wurde das in Abbildung 52 dargestellte Gesamtsystem entworfen. Die Software wurde in zwei Anwendungen unterteilt, welche über einen ActiveMQ-Server miteinander kommunizieren. Die in der Abbildung oberhalb des Servers dargestellten Komponenten bilden den *Sensorfeld-Server*. Er liest die kapazitiven Sensoren aus und bereitet die Daten für die weitere Verarbeitung auf. Anschließend stellt er diese anderen Projekten über den ActiveMQ-Server zur Verfügung. Auf diese Weise gelangen die Daten des Sensorfeldes zu der Anwendung *FallDetection*. In ihr werden sie zusammen mit den Bildern der Kamera zur Erkennung von Stürzen ausgewertet. Die Aufteilung des Systems in eine verteilte Anwendung erlaubt eine gewisse Dynamik beim Betrieb. So kann der Sensorfeld-Server beispielsweise auf einem leistungsschwachen PC im Wohnraum betrieben werden, während die rechenintensive Auswertung der Kameradaten auf einem Hochleistungsrechner in einem Kontrollraum oder Serverschrank ausgeführt wird. Um Teile der Software für andere Projekte verwenden zu können und leichter Anpassungen

vorzunehmen, wurden die Anwendungen zusätzlich in mehrere Komponenten unterteilt, welche in den nachfolgenden Abschnitten beschrieben werden.

ConnectionAdapter

Die Verbindung zu den Sensoren kann sowohl über eine serielle Schnittstelle des PCs wie auch über einen USB-Adapter erfolgen. Um die Software unabhängig von der Art der Verbindung zu halten, wurden alle hierfür nötigen Funktionen in eine Komponente ausgegliedert. Der Zugriff auf eine Schnittstelle läuft über ein generalisiertes Interface, welches für alle Verbindungsarten gleich ist. Hierbei ist es unerheblich, ob es sich um eine serielle Schnittstelle, einen USB-Adapter oder eine anderes Kommunikationsmedium handelt. Das Interface bietet rudimentäre Funktionen zum Senden und Empfangen von Daten sowie Methoden zur Konfiguration der Datenübertragung. Dieses Interface wird für den Zugriff auf die Sensoren benötigt.

CapacitiveSensors

In dieser Komponente sind alle Funktionen integriert, die zur Kommunikation mit den Sensoren benötigt werden. Einzelne Sensoren sind auf Objekte abstrahiert. Sämtliche von den Sensoren unterstützten Befehle sind als Funktionen herausgeführt, so dass die Kommunikation mit den Modulen verdeckt vom Benutzer stattfinden kann. Da an ein und dem selben PC durchaus mehrere Sensoren an unterschiedlichen Schnittstellen angeschlossen sein können, lassen sich durch die CapacitiveSensors Komponente zusätzlich sogenannte Factory-Instanzen erzeugen. Diese verwalten die angeschlossenen Sensoren und erlauben den Versand busweiter Broadcast-Nachrichten.

ActiveMQConnection

Zur Veröffentlichung des Interpretationsergebnisses der Sturzerkennung dient im Living Place ein zentraler ActiveMQ-Server. Auf diesem wird bei Erkennen eines Sturzes für potentielle Verbraucher eine Nachricht abgelegt. Die Komponente ActiveMQConnection stellt hierfür Producer, Publisher und Subscriber-Objekte zur Verfügung, über die sich auf einem beliebigen Server Daten publizieren und abrufen lassen (Siehe [31]). Die Kommunikation mit dem ActiveMQ-Server findet eigenständig innerhalb der Komponente statt. Die Basis dieser Komponente stellt das Apache ActiveMQ-CPP Projekt dar [63].

SettingsFileHandler

Wie bereits beschrieben, soll eine möglichst leichte Konfiguration der Software erreicht werden. Hierfür wurde eine auf dem XML-Protokoll basierende Datenstruktur gewählt. Diese hat den Vorteil, dass sie sowohl in Dateiform, wie auch in Datenbanken abgelegt werden kann. Durch die hierarchisch strukturierte Form von XML-Daten ist es auch leicht möglich, außerhalb der Software Änderungen an den Daten vorzunehmen.

Für die Verarbeitung der Daten dient tinyxml [32]. Tinyxml ist ein lightweight OpenSource XML-Parser zum Lesen und Schreiben von XML-Dateien. Um ein weiteres Maß an Flexibilität zu erreichen, ist das Handling der Konfigurationsdateien in eine eigene Komponente ausgegliedert. Der Zugriff auf die Daten erfolgt über ein generalisiertes Interface. Sollte die Datenstruktur oder die Speicherart der Konfigurationsdaten sich ändern, reicht es somit aus, die entsprechende Komponente anzupassen.

Gui

Über die grafische Benutzeroberfläche sollen dem Benutzer Informationen und Einstellmöglichkeiten zugänglich gemacht werden. Aufgabe der Benutzeroberfläche wird es sein, Informationen wie den Zustand des Sensorfeldes oder den aktuellen Status der Software anzuzeigen. Zudem soll es möglich sein, das Verhalten der Sturzerkennung durch Parameter anpassen zu können. Die Benutzeroberfläche nimmt hierdurch einen besonderen Status in den Komponenten ein. Durch die Möglichkeit zur Laufzeit Einstellungen zu ändern, muss die Gui-Komponente aktiv in den Prozess der Sturzerkennung eingreifen können. Um dies zu ermöglichen, stellt die Hauptkomponente eine Schnittstelle für die Komponente, über die der Zugriff erfolgen kann.

Als Grundgerüst für die grafische Oberfläche soll eine Grafikbibliothek dienen, die möglichst viele der nachstehende Kriterien erfüllt:

- Frei erhältlich
- Quelloffen
- Plattformunabhängig
- Anzeigemöglichkeiten für Graphen
- Gute Dokumentation / große Community

Für diesem Zweck wurden mehrere Bibliotheken für grafische Benutzeroberflächen getestet, welche in den folgenden Abschnitten näher erläutert werden.

wxWidgets

wxWidgets wurde ursprünglich an der Universität von Edinburgh als Projekt zur Portierung von Programmen zwischen Windows und Unix entwickelt [47]. Aktuell wird es von einer große Community weiterentwickelt in der auch namenhafte internationale Firmen vertreten sind. Durch die große Community existieren dementsprechend viele Dokumentationen und Anleitungen, was den Einstieg in diese Bibliothek erheblich erleichtert. Gleichwohl ist es für den privaten wie auch den kommerziellen Gebrauch kostenlos erhältlich. Neben gängigen Windows und Unix Versionen werden auch MacOS, Windows Mobile sowie das iPhone SDK unterstützt. wxWidgets ist vollständig in C++

geschrieben, lässt sich jedoch auch in Verbindung mit C#, Java, Python und weiteren Sprachen verwenden. Die Ausgabe von Graphen lässt sich jedoch im benötigten Umfang nur durch Plugins erreichen. Anwendungen, die auf wxWidgets basieren sind beispielsweise SYSmark 2012 [6], KICAD [26] und Audacity [46].

Qt

Bei Qt handelt es sich ein von Nokia entwickeltes Toolkit für grafische Oberflächen [41]. Neben einer großen Community setzen auch viele bekannte Firmen auf den Einsatz von Qt. Neben einer großen Anzahl von Dokumentationen und Anleitungen finden regelmäßig Schulungen und Zertifizierungen statt. Qt bietet sowohl eine OpenSource-Lizenz in Form der LGPL wie auch eine kommerzielle Lizenz. Ebenso wie wxWidgets unterstützt Qt eine Vielzahl von Betriebssystemen und Plattformen, darunter auch Nokias Betriebssysteme Symbian, Maemo und MeeGo. Neben einem in C++ verfassten Framework bietet Qt eine komplette Entwicklungsumgebung zur Erstellung von grafischen Oberflächen an. Diese umfasst neben einem grafischen Editor auch einen Compiler, Simulator sowie die Unterstützung von mehrsprachigen Benutzeroberflächen. Dennoch ist eine Ausgabe von Graphen ebenfalls nur über eine Erweiterung möglich.

Genutzt wird Qt u.a. in den Anwendungen Stellarium [45], EDIS Interactive [11] und dem VoIP Development Kit [40].

GTK+

Das ursprünglich für Unix Systeme entwickelte Gimp Toolkit (GTK+) ist ein auf der GLib Bibliothek aufbauendes Framework. Neben einer Vielzahl von Entwicklern wird es auch von mehreren Firmen unterstützt. Dennoch dient als aktuelles Lizenzierungsmodell die GNU LGPL. Als native Entwicklungssprache von GTK+ kommt C zum Einsatz. Durch eine Vielzahl sogenannter *Wrapper* lässt sich GTK+ auch in anderen Programmiersprachen verwenden. Hier seien beispielsweise C++, Python, oder Javascript genannt. Als mögliche Plattformen hierfür dienen Unix-Systeme ebenso wie Windows und MacOS. Für die Ausgabe von Graphen ist jedoch wie bei den bereits genannten Bibliotheken ein extra Plugin nötig. GTK+ wird u.a in Openmoko [50], Inkscape [44] und GIMP [43] eingesetzt.

Alle getesteten Frameworks weisen ähnliche Eigenschaften auf und eignen sich somit im gleichen Maße. Nach mehreren Versuchen wurde Qt als Framework für die Benutzeroberfläche gewählt. Auf Grund seines Aufbaus lässt es sich leicht in Anwendungen integrieren und liefert eine solide, multithreadfähige Struktur. Zudem lassen sich mit dem mitgelieferten Editor unkompliziert komfortable Benutzeroberflächen erstellen. Für die Anzeige von Graphen kommt die

kostenlose Erweiterung Qwt [66] zum Einsatz.

FallDetection

Die Hauptkomponente übernimmt die komplette Ablaufsteuerung der Sturzerkennung, welche im folgenden Abschnitt näher erläutert wird. Sie vereint neben der Kontrolle der Benutzeroberfläche sowohl die Analyse des Sensorfeldes wie auch die der Kamerabilder. Während sich die anderen, in den vorherigen Abschnitten beschriebenen Komponenten, auch in andere Projekte und Anwendungen einsetzen lassen, ist die FallDetection-Komponente speziell auf die Erkennung von gestürzten Personen ausgelegt.

Sensorfeld-Server

Die Hauptkomponente des Sensorfeld-Servers dient zum Auslesen und Auswerten der Sensoren. Hierfür wird aus den Daten ein 2-dimensionales Modell des Sensorfeldes erstellt, in dem der Zustand jedes Feldes hinterlegt ist. Somit lässt sich leichter erkennen, auf welchem Feld sich ein Objekt befindet, oder ob ein Feld ausgefallen ist. Neben dem Datenmodell, welches nach Abschluss der Auswertung auf einem ActiveMQ-Server veröffentlicht wird, werden auch die unbearbeiteten Rohdaten der Sensoren für andere Projekte auf dem Server publiziert.

5.3 Ablaufsteuerung

Für eine erfolgreiche Erkennung von gestürzten und hilflosen Personen ist eine effiziente Verknüpfung der Sensordaten und Kamerabilder nötig. Theoretisch betrachtet wären beide Methoden unabhängig von einander in der Lage, Stürze zu erkennen. In der Praxis zeigen sich jedoch Restriktionen, die es durch die jeweils andere Technik auszugleichen gilt. Auf Grund der Position der Kamera beispielsweise, ist es nur sehr schwer möglich, eine Person zu detektieren, die sich auf die Kamera zu bewegt hat und dabei gestürzt ist. Die Sensoren hingegen haben den Nachteil, dass für eine lückenlose Überwachung der komplette Raum ausgestattet sein muss. Um diese Unzulänglichkeiten zu kompensieren, ist es nötig, beide Methoden in geeigneter Weise miteinander zu verbinden. In den folgenden Abschnitten werden die Vor- und Nachteile verschiedener Ansätze erläutert.

Ein erster Ansatz ist es, beide Systeme unabhängig voneinander laufen zu lassen und ihre Ergebnisse miteinander zu verrechnen. Dies hat den Vorteil, dass in Bereichen, in denen keine Sensoren vorhanden sind, mit Hilfe der Kamera dennoch Stürze erkannt werden können. Auf der anderen Seite lässt sich durch richtige Anordnung der Sensoren das oben beschriebene Defizit der Kamera ausgleichen. Die Ergebnisse beider Systeme ließen sich anschließend durch analytische oder maschinelle Lernverfahren miteinander verknüpfen. Bei diesem Verfahren besteht jedoch der Nachteil, dass die Personen im Raum durch die Sensoren ständig einem elektromagnetischem Feld ausgesetzt sind. Zudem ist für die parallele Bearbeitung beider Verfahren viel Systemleistung nötig. Dies erfordert die Unterbringung eines entsprechend leistungsstarken Rechensystems in der Wohnumgebung. Des weiteren ist eine komplexe Synchronisation zwischen den beiden Abläufen nötig, damit sichergestellt sein kann, dass die Daten der Sensoren mit den aktuellen Kamerabildern übereinstimmen.

Weiterhin denkbar wäre es, die Kamerabilder durchgehend auszuwerten und je nach Bedarf die Sensoren hinzuschalten, beispielsweise bei einer Annäherung an diese. Hierbei würden die Kameras nicht zur Detektion der Stürze dienen sondern die Aufgabe eines Bewegungsmelders einnehmen. Auf diese Weise ist es möglich festzustellen, ob eine gestürzte Person in der Lage ist, sich zu bewegen oder ob ihr von jemand anderem geholfen wird. Ein weiter Vorteil dieses Verfahrens ist die geringe Strahlung, die von den Sensoren ausgeht, da diese vorrangig abgeschaltet wären. Jedoch benötigt dieser Ansatz auf Grund der fortlaufenden Bildanalyse ebenfalls eine große Menge an Rechenleistung.

Der in dieser Arbeit umgesetzte Ansatz basiert auf der eben beschriebenen Arbeitsweise, wertet jedoch die Sensoren permanent aus und greift nur bei Bedarf auf die Kamerabilder zu. Dies hat zwar den Nachteil, dass ständig ein

elektrisches Feld im Bereich der Sensoren herrscht, jedoch lässt sich auf diese Weise die benötigte Rechenleistung drastisch senken. Des weiteren wird dadurch die Privatsphäre der Person nicht verletzt, da die Kamerabilder nur ausgewertet werden, wenn dies von dem System als nötig angesehen wird. Die Sensoren dienen hierbei zur eigentlichen Erkennung des Sturzes. Sobald diese eine am Boden liegende Person melden, wird mit Hilfe der Kamerabilder geprüft, ob sich weitere Personen im Raum befinden, welche sich um die hilflose Person kümmern können. Ist dies nicht der Fall oder kommt ihr keiner zur Hilfe wird nach einer gewissen Zeit ein Alarm ausgelöst.

Abbildung 53 zeigt den Ablauf der Sturzerkennung in Form eines Zustandsautomaten.

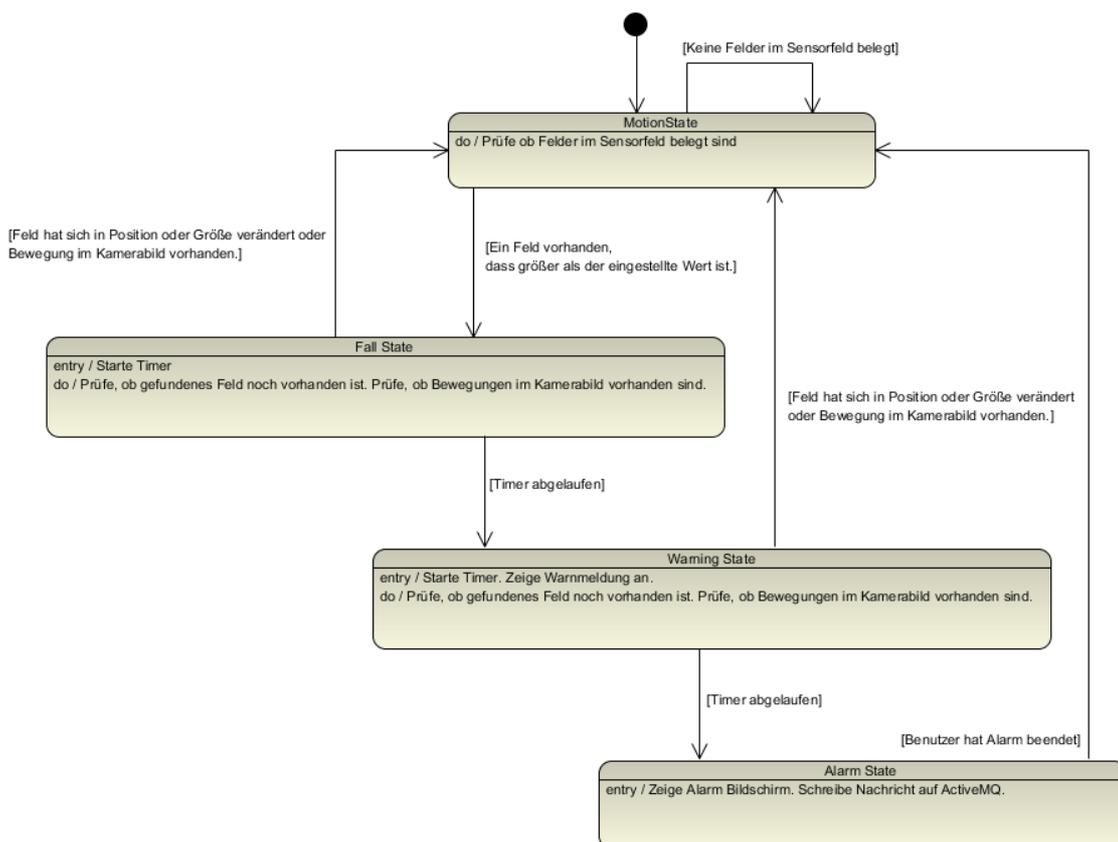


Abbildung 53: Ablauf der Sturzerkennung als Zustandsautomat

5.4 Verarbeitung der Sensordaten

Für eine erfolgreiche Interpretation ist es erforderlich, die Sensordaten vor der Verarbeitung zu bereinigen und markante Merkmale zu extrahieren. Hierzu ist es nötig, die Rohdaten der Sensoren zu filtern und anschließend zu normieren. Erst danach können signifikante Merkmale, wie beispielsweise Bereiche, in denen sich ein Objekt befindet, extrahiert werden.

Die Rohdaten müssen zuerst gefiltert werden, um Messfehler, verlorene Messwerte oder Schwankungen in den Daten zu auszugleichen. Geeignete Verfahren hierfür sind beispielsweise die Berechnung eines Mittelwertes, der Standardabweichung oder aber ein Tiefpassfilter. Für die kapazitiven Sensoren hat sich ein gleitender Mittelwert der 5. Ordnung als effektiv erwiesen (Abb. 54).

Nachdem die Rohdaten gefiltert wurden, ist es nötig, diese auf ein einheitliches Maß zu bringen. Die Normierung ist notwendig, da die Wertebereiche der Sensoren, bedingt durch Umgebungseinflüsse und Toleranzabweichungen der Bauteile, untereinander stark abweichen können.

Der grüne und gelbe Graph in Abbildung 54 illustriert die unterschiedlichen Bereiche zweier Sensoren.

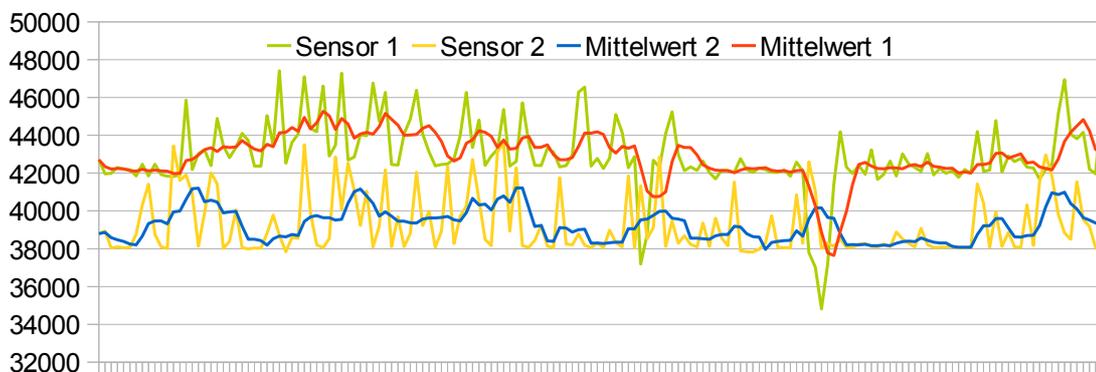


Abbildung 54: Mittelwerte und Abweichung zweier Sensoren

Die Versuche haben gezeigt, dass sich die Werte aller Sensoren bei Annäherung in der selben Größenordnung ändern. Somit bietet es sich an, die Abweichung von der Ruhelage der Sensoren zum Normieren zu verwenden. Durch die Differenz direkt aufeinanderfolgender Punkte lässt sich jedoch keine konkrete Aussage über den Zustand des Sensors treffen. Es ist also erforderlich, eine Methode zu finden, mit der sich die Ruhelage des Sensors abbilden lässt. Als Vorgehen hierfür kommen beispielsweise die Analyse der Steigung zwischen zwei Messwerten, die Bildung eines Mittelwertes und der Vergleich mit diesem oder aber die Berechnung der Standardabweichung über mehrere Datensätze in Frage. Als Ansatz wurde die Berechnung eines gleitenden Mittelwertes, wie er bereits beim Filtern der Rohdaten zum Einsatz kommt, gewählt. Dieser muss so gewählt werden, dass er sich an langfristige Änderungen, wie sie durch Schwankungen der Temperatur und Luftfeuchtigkeit erzeugt werden, angleicht. Gleichzeitig muss er sich jedoch möglichst schnell an dauerhafte Änderungen in den Messwerten anpassen können. Dies ist wichtig, wenn beispielsweise Gegenstände für längere Zeit in das Sensorfeld gestellt werden.

Anhand des aktuellen Sensorwertes und der berechneten Ruhelage des Sensors lässt sich nun die Abweichung berechnen. Diese ist unter gleichen Bedingungen

bei allen Sensoren nahezu identisch.

Mit Hilfe der Abweichung kann anschließend die Entfernung zu einem Objekt sehr genau bestimmt werden.

Für die Erkennung eines Sturzes hingegen ist nur relevant, ob sich etwas auf der Sensorfläche befindet oder nicht. Es bietet sich daher an, dies über ein Schwellwertverfahren zu lösen. Das bedeutet, sobald die Abweichung von der Ruhelage des Sensors einen festgelegten Werte überschreitet, kann angenommen werden, dass sich ein Objekt auf dem Sensor befindet.

Im weiteren Verlauf müssen die eingesetzten kapazitiven Sensoren als komplettes Sensorfeld angesehen werden. Durch das soeben beschriebene Schwellwertverfahren lässt sich dieses Feld auf einen 2-dimensionalen Raum abbilden, in dem jeder Wert den Zustand – belegt oder unbelegt - eines Sensors repräsentiert. Auf diese Weise erhält man eine monochrome Abbildung, welche sich mit Operationen aus dem Bereich der Bildverarbeitung weiter analysieren lässt.

Von besonderem Interesse dabei ist die Größe der belegten Bereiche. Die Aufteilung der Abbildung in einzelne Bereiche erfolgt durch einen Segmentierung-Algorithmus, welche auch auch in der Bildverarbeitung eingesetzt wird. Da es sich um eine binäre Abbildung handelt, bietet sich ein pixelorientiertes Verfahren, wie beispielsweise der klassische *Connected Components Labeling* Algorithmus an [54].

Bei diesem auch als Two-Pass-Algorithmus bezeichnetem Verfahren wird die Abbildung zeilenweise zweimal hintereinander durchlaufen. Der erste Durchlauf dient dem Finden von Gleichheiten bei benachbarten Pixeln sowie dem Zuweisen von temporären IDs. Hierfür wird das Bild einmal komplett durchlaufen und je nach Implementation 2 (4-connectivity) oder 4 (8-connectivity) mögliche Nachbarn mit einbezogen. Sollte kein Nachbar existieren, dem bereits eine ID zugewiesen wurde, bekommt der Pixel eine neue ID. Sollten jedoch ein oder mehrere Nachbarn bereits eine ID besitzen, wird dem Pixel die kleinste, angrenzende ID zugewiesen. Zusätzlich werden alle IDs der Nachbarn gespeichert und markiert, dass diese untereinander verbunden sind. Im zweiten Durchlauf werden die IDs durch ein endgültiges Label ersetzt. Hierfür wird das Bild erneut durchlaufen und allen IDs, die als untereinander verbunden markiert sind, der kleinste Wert aus dieser Gruppe zugewiesen.

Eine Optimierung des Verfahrens ist der One-Pass Algorithmus. Dieser identifiziert und benennt die verbundenen Bereiche in einem Durchgang. Durch den Wegfall des zweiten Durchgangs ergibt sich bei wenigen großen Bereichen eine signifikante Abarbeitungsgeschwindigkeit gegenüber der Two-Pass Variante.

```

algorithm TwoPass(data)
  linked = []
  labels = structure with dimensions of data, initialized with 0

  First pass

  for row in data:
    for column in row:
      if data[row][column] is not Background

        neighbors = labels of all neighboring elements

        if neighbors is empty
          linked[NextLabel] = set containing NextLabel
          labels[row][column] = NextLabel
          NextLabel += 1

        else

          Find the smallest label
          labels[row][column] = min(neighbors labels)

          connect all linked labels
          for label_A in neighbors labels
            for label_B in neighbors_labels
              for label_C in linked[label_B]
                linked[label_A] += label_C

  Second pass

  for row in data
    for column in row
      if labels[row][column] is not 0
        labels[row][column] = min(linked[labels[row][column]])

```

Pseudocode des Two-Pass-Algorithmus

Da jedoch alle bereits detektierten Bereiche mit einbezogen werden, sinkt mit zunehmender Anzahl die Arbeitsgeschwindigkeit [42]. Der obige Pseudocode verdeutlicht den Ablauf des Algorithmus.

Die auf diese Weise erhaltenen Datensätze geben Aufschluss über die aktuelle Größe und Position von Objekten auf dem Sensorfeld. Anhand der Felder soll nun eine Aussage getroffen werden, ob eine Person innerhalb des Sensorfeldes liegt.

Hierfür werden nachfolgend die Vor- und Nachteile von zwei möglichen Interpretationsansätzen erläutert.

Maschinelle Lernverfahren

Durch die große Anzahl an Permutation, die im Sensorfeld entstehen können, bietet sich die Verwendung eines maschinellen Lernverfahren an. Im einfachsten Fall ist eine binäre Klassifikation nötig, um eine liegende Person im Sensorfeld zu bestimmen. Damit ein maschinelles Lernverfahren die eingehenden Daten jedoch erfolgreich klassifizieren kann, muss es vorab trainiert werden. Um möglichst realistische Trainingsdaten zu bekommen, ist es erforderlich, je nach

verwendetem Verfahren nötig, mehrere hunderte Stürze auf die Sensorfläche zu simulieren und aufzuzeichnen. Um Verletzungen zu vermeiden, wäre es erforderlich das Sensorfeld entsprechend zu polstern. Dies würde jedoch zu einer Verfälschung der Messwerte führen. Ein weiteres Risiko besteht in der Über- oder Unterklassifikation des Verfahrens. Bei einem nicht ausreichend trainierten Lernverfahren findet keine korrekte Klassifikation statt. Bei einem zu sehr trainierten Lernverfahren hingegen kann man von einem Auswendiglernen sprechen. In diesem Fall werden nur noch Situationen erkannt, die nahezu identisch mit den Trainingsdaten sind. In beiden Fällen ist ein maschinelles Lernverfahren dem folgenden Ansatz unterlegen.

Hard-Coded

Eine liegende Person hinterlässt, abhängig von Größe und Ausrichtung, ein signifikantes Muster auf der Sensorfläche. Diese lassen sich als Vergleichsmuster in der Software hinterlegen, mit den aktuellen Datensätzen vergleichen und auf Übereinstimmung prüfen. Dabei wäre ein mögliches Muster die Anzahl der Sensorfelder, die eine Person bedeckt oder die Ausrichtung dieser Fläche. Da dieses Verfahren rein auf dem Vergleich verschiedener Parameter basiert, ist es wesentlich performanter als ein maschinelles Lernverfahren. Durch die große Anzahl an Möglichkeiten, wie eine Person liegen kann, ergibt sich jedoch eine Vielzahl von Messwertkombinationen, die als Vergleichsmuster abgebildet werden müssten. Um dem entgegenzuwirken, könnten, durch die Software einstellbare Parameter dienen.

Bei der Auswahl des Interpretationsverfahren muss bedacht werden, dass nicht nur der aktuelle Zustand des Sensorfeldes beachtet werden darf. Um eine Person zu erkennen, die sich nicht mehr bewegt, müssen auch zeitlich zurückliegende Zustände in die Interpretation einbezogen werden. Hier gilt es abzuwägen, ob es Sinn macht, ein komplexes Interpretationsverfahren zu wählen, welches durch die Einbeziehung des zeitlichen Aspektes noch ungleich umfangreicher wird, oder ob es effizienter ist, einen analytischen Ansatz zu wählen.

Für die Implementation wurde ein Hard-Coded Ansatz gewählt, welcher neben der Größe eines Feldes auch dessen Veränderungen und die Dauer, die es vorhanden ist, mit einbezieht. Die Parameter hierfür sind mit in der Software-Konfiguration abgelegt. Somit lässt sich das Verhalten des Verfahrens bei Bedarf anpassen.

5.5 Bildanalyse

Neben den kapazitiven Sensoren stellt das von der Kamera gelieferte Videomaterial eine wichtige Komponente in dem System dar. Durch dieses ist es möglich, die Anzahl und den genauen Aufenthaltsort der anwesenden Personen zu bestimmen. Um dies zu erreichen, müssen Personen in den Bildern erst einmal erkannt werden.

Die Erkennung von Personen in Standbildern und Videos stellt eine der großen Herausforderungen in der automatischen Video- und Bildverarbeitung dar. Dies liegt nicht zuletzt an der großen Variation der Daten, welche das Problem extrem komplex machen. Jeder Mensch sieht unterschiedlich aus und bewegt sich anders. Dennoch gibt es Gemeinsamkeiten, anhand derer versucht werden kann, einen Menschen zu detektieren. Von besonderem Interesse sind hierbei Merkmale im Gesicht, wie beispielsweise Augen, Mund und Nase. Auch andere Körperstrukturen wie die Hände oder die Haut lassen sich verwenden. Techniken, welche mit diesen Strukturen arbeiten, werden als *Face detection* und *Skin detection* Algorithmen bezeichnet. Beispiele hierfür sind die Arbeiten von Heisele, Poggio und Pontil [4] sowie Jones und Rehg [34].

Ein weitere Ansatz ist die Erkennung von Bewegungen mittels modellbasierter Methoden. Diese arbeiten üblicherweise mit einem Hintergrundmodell, anhand dessen es möglich ist, bewegende Objekte zu detektieren und mit Hilfe ihrer Kontur, Farbe oder Bewegung zu klassifizieren. Als Beispiele hierfür seien die Arbeiten von Elzein, Lakshmanan und Watta [20], Haga sowie Sumi und Yagi [60] genannt.

Im folgenden Abschnitt soll der gewählte Ablauf der Bewegungserkennung näher erläutert werden.

Vorverarbeitung

Bevor die Bewegungsinformationen aus den Bildern extrahiert werden können, müssen alle irrelevanten Daten entfernt werden. Hierfür wurde ein angepasster, modellbasierter Ansatz gewählt, wie er in den Arbeiten [21] und [27] beschrieben wird.

Hierfür wird zunächst mit Hilfe eines Hintergrundmodells ein Differenzbild erzeugt. Dieses wird durch verschiedenen Verfahren in mehrere Ebenen unterteilt. Auf diese Weise lassen sich Schatten und Reflexionen detektieren und entfernen. Abschließend werden die Daten für die weitere Verarbeitung durch ein Schwellwertverfahren in ein Binärbild überführt.

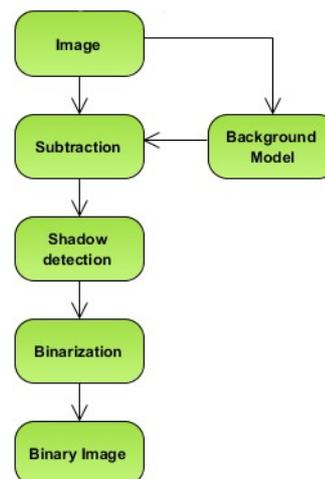


Abbildung 55: Bild-Vorverarbeitung

Für die Erstellung des Differenzbildes werden die von der Kamera gelieferten Bilder zuerst zu einem Modell verrechnet, welches den Hintergrund des überwachten Bereichs repräsentiert (Abbildung 56a). Es enthält Informationen über statische oder sich nur langsam verändernde Objekte, vergleichbar mit der in Kapitel 5.4 beschriebenen Mittelwertbildung. Für die weitere Verarbeitung müssen zunächst Störungen und Rauschen in den Bildern entfernt werden. Hierfür wird sowohl aus dem aktuellen Bild der Kamera, wie auch des Hintergrundmodells die Helligkeitskomponente berechnet. Anschließend kann durch Subtraktion beider Komponenten ein Differenzbild erzeugt werden. Abbildung 56 verdeutlicht den Vorgang anhand eines Hintergrundbildes (a), des aktuellen Kamerabildes (b), sowie dem daraus resultierendem Differenzbild (c).

Da sich die Helligkeitskomponente durch Abschattungen oder Reflexionen drastisch ändern kann, ist es nötig diese aus dem Differenzbild heraus zu filtern. Hierfür wird jedes Pixel des Differenzbildes anhand eines einstellbaren Gewichtungsfaktor sowie seines Wertes in die Kategorien *Vordergrund*, *Hintergrund*, *möglicher Hintergrund* sowie *möglicher Vordergrund* eingeteilt. Durch diese Einteilung lassen sich bereits grob Bereiche abgrenzen, in denen Bewegungen vorhanden sind. Durch Schatten oder Lichtreflexionen wird es allerdings noch Bereiche geben, die nicht korrekt klassifiziert wurden. Aus diesem Grund, werden Bereiche, die der Kategorie *möglicher Vordergrund* zugewiesen wurden, einem weiteren Arbeitsschritt unterzogen. Der Grundgedanke hierbei ist, dass Schatten und Reflexionen zwar die Helligkeit in einem Bereich verändern, nicht jedoch die Farbton. Für den nachfolgenden Schritt ist es also nötig, aus dem Hintergrundmodell und dem aktuellen Kamerabild die Farbkomponente zu extrahieren. Mit ihrer Hilfe lassen sich durch Anwenden der in Abbildung 57 dargestellten Formel falsch eingeordnete Bereiche korrigieren.



Abbildung 56: Differenzbild erstellen
Hintergrundmodell (a), aktuelles Bild (b),
Differenzbild (c)

wenn $p \in \text{möglicher Vordergrund} \wedge |H_I(p) - H_b(p)| < K_4$ dann
 $p \Rightarrow \text{möglicher Hintergrund}$

Abbildung 57: Formel zur Korrektur der Klassifizierung

Diese besagt, dass die Klassifizierung eines Pixel p in die Klasse *möglicher Hintergrund* geändert wird wenn, p als *möglicher Vordergrund* klassifiziert wurde und die Helligkeitskomponente $H_I(p)$ des Pixels abzüglich der Helligkeitskomponente des Hintergrund-Modells $H_b(p)$ kleiner als der Schwellwert K_4 ist.

Im weiteren Verlauf werden mittels verschiedener morphologischer Filter kleine, freistehende Bereiche entfernt und wenige Pixel große Lücken in den verbliebenen Bereichen geschlossen. Abschließend werden alle als *Vordergrund* oder *möglicher Vordergrund* klassifizierten Pixel in ein Binärbild übertragen. Auf diese Art entsteht ein monochromes Bild, in dem nur die Pixel markiert sind, welche signifikant auf eine Bewegung hindeuten.

Abbildung 58 illustriert die Erstellung des Differenzbildes anhand der Einteilung des Bildes in mehrere Kategorien (a), der anschließenden Anwendung verschiedener Filter zur Reduktion unerwünschter Bereiche (b), sowie der abschließenden Erstellung des Binärbildes mit Hilfe eines Schwellwertes (c).

Segmentierung

Auf das entstandene Binärbild kann, wie bereits für die kapazitiven Sensoren in Kapitel 5.4 beschrieben, ein Segmentierungsverfahren angewendet werden, um einzelne Bereiche zu unterscheiden. Möglich wäre hierfür ebenfalls der TwoPass-Algorithmus oder einer seiner Derivate. Die Wahl fiel jedoch auf ein Verfahren, welches von der verwendeten OpenCV-Bibliothek zur Verfügung gestellt wird. Dieses basiert auf der Detektion von Konturen, was besonders bei großen Datenmengen zu einer signifikanten Performanzsteigerung gegenüber dem beschriebenen Algorithmus führt. Ein weiterer Vorteil dieses Verfahrens ist die Integration eines Tracking Algorithmus. Dies erlaubt die eindeutige Identifizierung eines Segmentes über einen längeren Zeitraum.

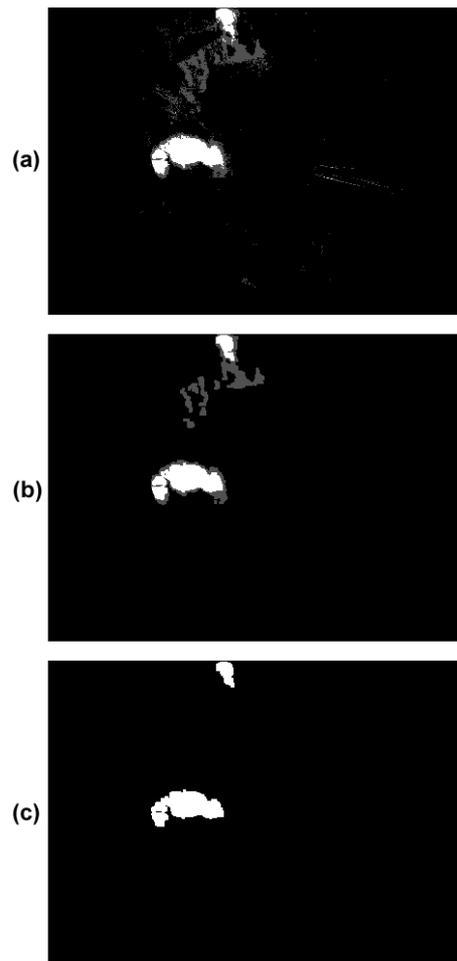


Abbildung 58: Binärbild erstellen
 Klassifiziertes Bild (a), Nach Anwendung morphologischer Filter (b), Binärbild (c)

Klassifizierung

Anhand der extrahierten Segmente lassen sich bereits erfolgreich Aussagen über einzelne Bewegungen in der Szene treffen. Allerdings kann die Quelle der Bewegung noch nicht bestimmt werden. So kann es sich bei einer Bewegung zwar um einen Menschen handeln, jedoch ist es ebenso möglich, dass sie von einem Tier oder einem sich bewegenden Gegenstand stammt. Um dies auszuschließen, ist eine Klassifizierung der einzelnen Segmente nötig. Beispiele für Verfahren zur Bestimmung von Personen und deren Körperhaltung finden sich in den Arbeiten von Zisserman, Marín-Jiménez und Ferrari [67] sowie Mojarrad, Rahmani und Mohebi [35].

Für das in dieser Arbeit beschriebene Verfahren ist allerdings nur von Interesse, ob sich eine Person im Kamerabereich bewegt, womit eine Einteilung der Segmente in „*ist ein Mensch*“ und „*ist kein Mensch*“ ausreichend ist. Hier kommt einem die sphärische Optik der Kameras zu Gute. Durch ihre optische Verzerrung neigen sich alle Objekte im Blickfeld strahlenförmig zum Mittelpunkt des Bildes. Somit lassen sich mit einfachen geometrischen Berechnungen Größe und Entfernung von Objekten bestimmen. Eine stehende Person beispielsweise stellt ein Bewegungssegment dar, dessen längere Seite zum Mittelpunkt des Bildes zeigt. Anhand dieses Merkmals, sowie der Größe des Segments lässt sich sagen, ob es sich um eine stehende und bewegende Person handelt.

Vorangegangene Versuche haben gezeigt, dass das in diesem Kapitel beschriebene Verfahren in Verbindung mit der verwendeten Kamera sehr gute Ergebnisse bezüglich Stabilität, Erkennungsrate und Performance bietet.

5.6 Risiken und Probleme

Das größte Risiko besteht in einem fehlerhaften Melden eines Sturzes. Dies kann durch verschiedene Bedingungen ausgelöst werden.

In der Videoüberwachung kann dies durch ungewöhnliche Bewegungen hervorgerufen werden. Hierzu zählen u.a. Bewegungen in Fernsehern oder Spiegeln. Ebenso können Bewegungen vor Fenstern, wie sie z.B. von Bäumen und sich im Wind bewegende Ästen oder vorbeigehenden Passanten erzeugt werden, irrtümlich als Stürze klassifiziert werden. Ein weiteres Risiko stellen kleine Kinder und Hunde dar. Besonders kleine Kinder können aufgrund ihrer häufig krabbelnden Körperhaltung nicht eindeutig identifiziert werden, gleiches gilt für Hunde. Während kleinere Tiere aufgrund ihrer Größe noch durch den Filter entfernt werden, ist dies bei größeren nicht mehr möglich. Da die Silhouette eines Menschen, der sich auf Hände und Knie stützt der eines großen Hund sehr ähnelt, ist es nicht möglich diese nur anhand des Umrisses und der Bewegung zu unterscheiden. Hier wären weitreichendere Analysen des entsprechenden

Bildausschnittes nötig.

Entsprechend verhält es sich mit den kapazitiven Sensoren. Hier kann durch Ausfälle einzelner Sensoren an ungünstigen Positionen das gesamte Sensorfeld unbrauchbar werden. Ebenso können große oder wasserhaltige Objekte, die auf das Sensorfeld gestellt werden, dieses beeinträchtigen. Zwar passt es sich der Änderung nach einer Weile an, jedoch kann es in dieser Zeit zu Fehlmeldungen kommen.

In allen Fällen ist es jedoch sinnvoller, ein ungültigen Sturz zu melden, statt einen ggf. stattgefunden Sturz zu ignorieren. Da das System anhand der Indizien nur auf einen möglichen Sturz hinweisen kann, ist eine Sichtung der Auswertung durch einen Menschen unabdingbar. Somit bedeuten ungültige Stürze zwar einerseits Mehrarbeit für die überwachende Person, andererseits kann Leben gerettet werden.

6 Technische Umsetzung und Implementation

6.1 Aufbau des Sensorfeldes

Durch die Vorarbeiten in Kapitel 3.5 ergab sich, dass für den im Living Place verwendeten Teppich ein Sensorfeld mit 3 x 4 Elementen ausreichend ist. Dafür wurden zunächst die Flächen der späteren Messfelder auf der Unterseite des Teppichs mit Gewebeklebeband abgeklebt. Dies sorgte für die nötige Haftung der anschließend aufgeklebten Aluminiumstreifen (Abb. 60).

In die Mitte jedes Feld wurde einer der in Kapitel 4 entwickelten Sensoren platziert und mit dem Metall verbunden. Um den Sensor vor übermäßiger Belastung zu schützen, wurde er abschließend mit einer elastischen Kunststoffmatte umgeben (Abbildung 61). Da es bereits während der Versuche zu Kommunikationsproblemen durch ungeschirmte Kabel kam, wurde für den Aufbau eine extra flache, doppelt geschirmte Datenleitung vom Typ CAT-7 verwendet. Der rahmenartige Aufbau der Sensorfelder hat den Vorteil, dass eine große Fläche bei vergleichsweise geringer Eigenkapazität abgedeckt werden kann.



Abbildung 59: Fertige Sensorfläche



Abbildung 60: Teilweise beklebte Sensorfläche



Abbildung 61: Vollständige Sensorfläche

6.2 Implementation der Software

Mit Berücksichtigung der in Kapitel 5.2 bereits aufgelisteten Aspekte für eine leichte Wartbarkeit, einfache Konfigurierbarkeit und problemlose Erweiterbarkeit entstanden zwei in C/C++ geschriebene Multithread-Anwendungen mit Qt-basierter Benutzeroberfläche, dessen Aufbau nachfolgend genauer beschrieben ist.

Konfigurierbarkeit

Damit die Anwendungen leicht konfigurierbar sind, sind die Einstellungen in mehrere Bereiche unterteilt. Um dies auch physisch voneinander zu trennen, wurden die Einstellungen entsprechend ihrer Zugehörigkeit in separaten Dateien untergebracht. Listing 1 zeigt exemplarisch den Aufbau der Konfigurationsdatei der Hauptanwendung.

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <settings>
3   <activemq>
4     <server>tcp://127.0.0.1:61616</server>
5     <topic_alarm>LP.FALLDETECTION.ALARM</topic_alarm>
6     <topic_states>LP.FALLDETECTION.STATES</topic_states>
7     <id>sensorfieldServer</id>
8     <version>0.1</version>
9   </activemq>
10  <alarmtime>5</alarmtime>
11  <warningtime>5</warningtime>
12  <ignore_image>>false</ignore_image>
13  <send_msg_by_alarm>>false</send_msg_by_alarm>
22 </settings>
```

Listing 1: Konfigurationsdatei

Hierbei wurde bewusst auf XML-spezifische Elemente wie Parameter oder Namespaces verzichtet, um eine mögliche Portierbarkeit in ein anderes Format zu gewährleisten.

Die im XML-Format gespeicherten Daten werden mit Hilfe von tinyXml eingelesen und der Software über ein einheitliches Interface zur Verfügung gestellt. Die Identifizierung eines Eintrages erfolgt dabei über ihren Pfad in der XML-Struktur (Listing 2, Zeile 4).

```

1 m_Settings = CSettingsFileFactory::GetSettingsFile(filename);
2 m_Settings->Open();
3 m_Alarmtime = atoi(m_Settings->GetAttribute("alarmtime").data());
4 string server = m_Settings->GetAttribute("activemq/server");

```

Listing 2: Laden der Einstellungen

Auf diese Weise ist ein einfacher Zugriff auf die Konfigurationsdaten gewährleistet, unabhängig in welchem Format sie gespeichert werden.

Verarbeitung der Kamerabilder

Die einzelnen Schritte der Bildanalyse, wie sie in Kapitel 5.5 beschrieben wurden, sind konsequent für die Verwendung auf Multicore-Prozessoren ausgelegt. Hierfür wurden die einzelnen Aufgaben jeweils auf mehrere Threads verteilt. Da OpenCV die Bildinformationen zeilenweise im Speicher ablegt, ließ sich eine einfache Unterteilung in verschiedene Threads realisieren, ohne dass diese in den selben Speicherbereichen arbeiten. Obwohl sich diese Einteilung nur bei Aufgaben anwenden lässt, bei denen Pixel unabhängig von ihrem Nachbarn bearbeitet werden, ergibt sich durch die gleichmäßige Lastverteilung ein großer Performancegewinn. Dies wird nicht zuletzt dadurch begünstigt, dass jeder Thread nahezu die gleiche Abarbeitungsdauer hat.

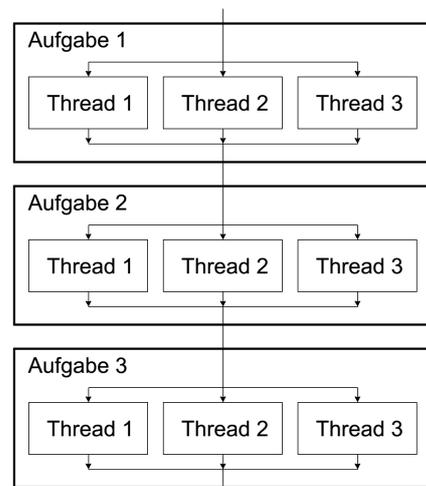


Abbildung 62: Lastverteilung in mehrere Threads

Listing 3 verdeutlicht die Aufteilung anhand der Differenzbilderstellung. Da einem Thread nur ein Zeiger übergeben werden kann, sind alle benötigten Informationen in der Struktur *createDifference_t* zusammengefasst, dessen Adresse beim Erstellen des Threads mit übergeben wird. In der Threadfunktion *tCreateDiff* wird anhand der mit *createDifference_t* übergebenen Informationen die eigentliche Operation durchgeführt. Damit es nicht zu Überschneidungen der Threads kommt, sind die Zeilen, für die der jeweilige Thread zuständig ist, durch die Parametern *startAt* und *endAt* vorgegeben.

```
1 void CMotionDetector::CreateDiff(IplImage *frame) {
2     HANDLE *thread = new HANDLE[m_nThreads];
3     for(int i=0;i<m_nThreads;i++) {
4         createDifference_t *p = new createDifference_t;
5         p->background = m_bgModel->GetRGBImage();
6         p->frame = frame;
7         p->startAt = (frame->height/m_nThreads)*i;
8         p->endAt = (i+1==m_nThreads)?frame->height:
9                 (frame->height/m_nThreads)*(i+1);
10        p->ret = m_imgDiff;
11        thread[i] = CreateThread(NULL,0,
12                                (LPTHREAD_START_ROUTINE)tCreateDiff,
13                                p,0,0);
14    }
15    for(int i=0;i<m_nThreads;i++) {
16        WaitForSingleObject(thread[i], INFINITE);
17        CloseHandle(thread[i]);
18    }
19    delete[] thread;
20 }
```

Listing 3: Differenzbilderstellung mit mehreren Threads

Äquivalent zu diesem Aufbau laufen die Schritte zur Eliminierung von Schatten und der Bildung des Hintergrund-Modells ab.

Analyse der Sensordaten

Wie in Kapitel 5 erläutert wurde, findet die Vorverarbeitung der kapazitiven Sensordaten in einer separaten Anwendung statt. Zu der Vorverarbeitung zählen neben dem Auslesen der Sensormodule die Bereinigung der Daten sowie eine Filterung, um eine vergleichbare Datenbasis zu schaffen. Erste Versuche mit dem Teppich haben gezeigt, dass sich die einzelnen Sensoren gegenseitig stören, wenn mehrere zur gleichen Zeit eine Messung ausführen. Daher wurde das System entsprechend abgeändert, so dass alle Sensoren hintereinander ihre Messung durchführen (Listing 4). Für eine leichtere Handhabung, sowohl der ausgelesenen Sensorwerte wie auch der einzelnen Sensoren, wurde eine Template-Klasse geschrieben, in welcher auf die Daten wie in eine Tabelle über ihre X- und Y-Koordinaten zugegriffen werden kann. Abhängig von den Sensordaten wird eine weitere Instanz dieser Klasse gepflegt, in welcher der jeweilige Zustand der Sensorfelder vermerkt ist. So sind Felder, bei denen der Sensorwert nicht ausgelesen werden konnte, oder für die kein Sensor eingestellt ist, als ungültig markiert. Felder, bei denen der Sensorwert entsprechend stark

von der Ruhelage abweicht, werden als aktiv markiert. Diese Karte bewirkt im weiteren Verlauf der Analyse eine leichtere Überprüfung des Sensorfeldes. Zudem dient sie der Visualisierung in der Benutzeroberfläche, auf die in den kommenden Abschnitten noch näher eingegangen wird.

```
1 CValueMap<int> CCapacitiveSensorComponent::AcquireSensorData() {
2   CValueMap<int> valueMap(GetNumRows(), GetNumColumns());
3   for(int x=0; x<m_SensorMap.GetNumColumns();x++) {
4     for(int y=0; y<m_SensorMap.GetNumRows();y++) {
5       if(m_SensorMap[y][x] != NULL) {
6         m_SensorMap[y][x]->StartMeasure();
7         Sleep(60);
8         valueMap[y][x] = m_SensorMap[y][x]->GetValue();
9       }
10    }
11  }
12  return valueMap;
13 }
```

Listing 4: Auslesen der Sensoren

Die auf diese Weise bereinigten und normierten Daten werden anschließend an den ActiveMQ-Server weitergeleitet. Hierfür werden sowohl die Rohdaten, wie auch die erstellte Status-Karte mit Hilfe von JsonCpp [5] in ein kompaktes Datenformat umgewandelt und für andere Projekte auf dem Server veröffentlicht.

```
1 {
2   „Id“: „sensorfieldServer“,
3   „Version“: „0.1“,
4   „Label“: „2451“,
5   „Rows“: „3“,
6   „Columns“: „4“,
7   „States“: „1;0;0;3;\n0;1;1;3;\n1;1;0;0;\n“
8 }
```

Listing 5: Aufbau der Json-Nachricht

Listing 5 zeigt den Aufbau einer JSON-Nachricht am Beispiel der Status-Karte. Die Richtlinien des Living Places geben vor, dass jede Nachricht auf dem ActiveMQ-Server die Id und Versionsnummer der Anwendung erhalten muss, welche die Nachricht verschickt hat. Das Element *Label* dient zum Detektieren von verlorengegangenen Nachrichten, sowie zum Synchronisieren der Anwendungen. Jedes Mal, wenn der Sensorfeld-Server eine Nachricht verschickt, wird dieses Feld um einen Wert erhöht. Des weiteren enthält die

Struktur Felder mit Informationen zur Größe des Sensorfelds sowie den eigentlichen Daten. Dieses, in der Nachricht als *States*-Element bezeichnete Feld, enthält die eigentlichen Informationen. Die einzelnen Felder sind hierbei zeilenweise, durch Semikolons getrennt, in einem String untergebracht. Die Zahlenwerte geben dabei den jeweiligen Status eines Feldes wieder, beispielsweise steht 3 für ein Feld, welches ignoriert werden soll.

Ablaufsteuerung

Nachdem die Daten auf dem ActiveMQ-Server eingetroffen sind, werden diese an die angemeldeten Empfänger weitergeleitet. Im Falle dieses Projektes an die FallDetection-Anwendung, welche die Auswertung übernimmt. Jedes Mal, wenn eine Nachricht vom Server eintrifft, werden die darin enthaltenen Sensordaten ausgewertet. Bei Anzeichen eines Sturzes wird zusätzlich das aktuelle Kamerabild für die weitere Auswertung gespeichert. Auf diese Weise ist sichergestellt, dass Bilder und Daten stets synchron bleiben.

Zur Steuerung der Sturzerkennung wurde ein einfacher Zustandsautomat implementiert, wie er in Kapitel 5.3 erläutert wurde. Jeder Zustand dieses Automaten muss die abstrakten Funktionen der Klasse *IState* implementieren, da über sie die Steuerung der einzelnen Zustände im Automaten erfolgt.

```
1 class IState {
2 public:
3     virtual void Entry() = 0;
4     virtual void Step() = 0;
5     virtual void Exit() = 0;
6 };
```

Listing 6: *IState* Klasse

Die Funktionen *Entry* und *Exit* werden nur bei einem Wechsel zwischen zwei Zuständen aufgerufen und dienen bei entsprechenden Zuständen beispielsweise zur Anzeige der Warnmeldung und Alarmmeldung eines Sturzes oder der Ausgabe von ActiveMQ-Nachrichten. In der Funktion *Step* hingegen laufen alle Aufgaben ab, die kontinuierlich durchgeführt werden müssen. Hierzu zählen u.a. die Analyse der Bild- und Sensordaten.

Benutzeroberfläche

Nach jedem Schritt des Automaten werden die aktuellen für die Anzeige relevanten Daten an die Benutzeroberfläche übertragen. Diese läuft in einem separatem Thread, damit auch bei stagnieren der Analyse, z.B. durch Fehlkonfiguration, noch eine Bedienung der Anwendung möglich bleibt. Hierdurch ist es jedoch nicht möglich, aktualisierte Daten direkt anzuzeigen. Stattdessen müssen diese zwischengespeichert werden, bis sich die Oberfläche aktualisiert.

Die Hauptansicht der Benutzeroberfläche dient zur Überwachung der Software. Neben dem aktuellen Kamerabild wird der Status der einzelnen Sensorfelder grafisch dargestellt. Mit diesen lässt sich erkennen, wo aktuell eine Bewegung stattfindet. Zudem gibt die Hauptansicht Aufschluss über den aktuellen Zustand des Sturzerkennungs-Automaten (Abbildung 63).

Weiterhin bietet die Benutzeroberfläche die Möglichkeit, jegliche Parameter der Software komfortabel zu konfigurieren (Abbildung 64).

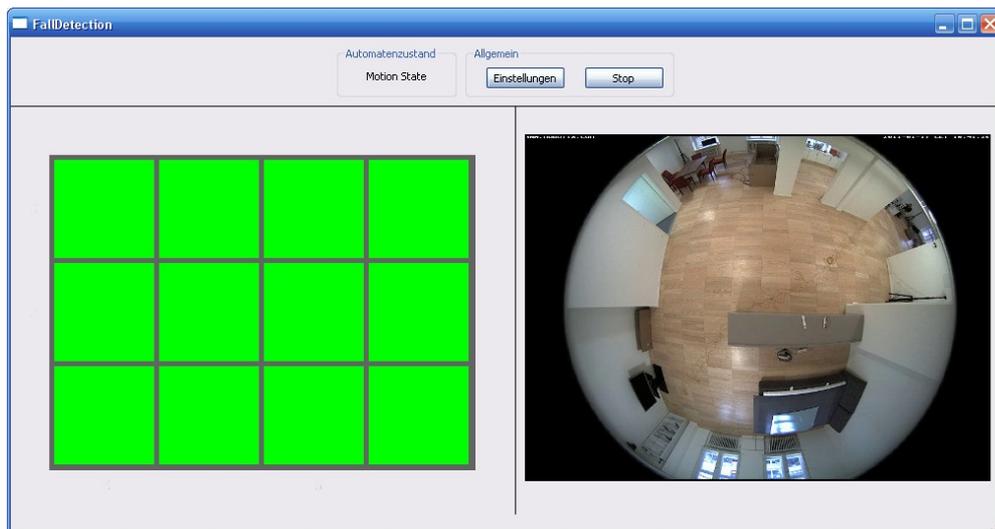


Abbildung 63: Hauptansicht der Software

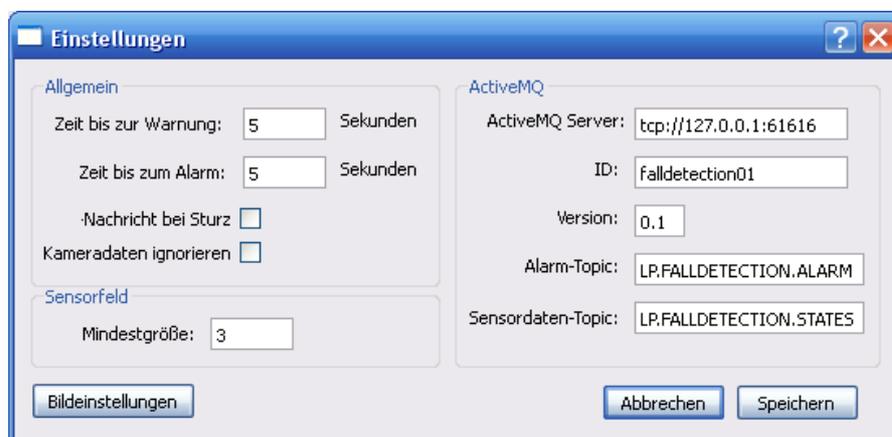


Abbildung 64: Fenster zur Konfiguration der Sensoren

7 Evaluierung

Im Rahmen dieser Masterarbeit wurde ein Konzept entwickelt und umgesetzt, mit dem eine berührungslose Erkennung von gestürzten und hilflos am Boden liegenden Personen möglich ist. Die Umsetzung erfolgte hierbei als ein Multisensor-System in der Projektumgebung des Living Places Hamburg. Hierfür war es nötig, neue und bereits vorhandene Sensoren auf ihre Verwendbarkeit hin zu evaluieren. Neben Erschütterungssensoren, welche sich auf Grund ihrer geringen Empfindlichkeit und Reichweite als ungeeignet erwiesen haben, wurden mehrere Versuche mit kapazitiven Messtechniken durchgeführt. Nach mehreren Untersuchungen fiel die Entscheidung auf einen selbst entwickelten Sensor, welcher speziell auf die Anforderung im Living Place angepasst wurde. Nach anfänglichen Problemen mit der Hard- und Software entstand ein kostengünstiger und hochempfindlicher Sensor zur Messung von Kapazitätsänderungen. Durch die Montage mehrere dieser Sensoren unter einen Teppich konnte ein Sensorfeld aufgebaut werden, welches sich leicht transportieren und in eine bestehende Wohnumgebung integrieren lässt. Das langfristige Verhalten von Sensoren und verwendeten Materialien konnte jedoch noch nicht ausreichend erprobt werden und muss über die Zeit weiter beobachtet werden.

Neben den Sensoren kommt eine fest im Living Place installierte Dome-Kamera zum Einsatz. Bei dieser handelt es sich um eine Kamera aus der Sicherheitstechnik, wodurch sie mit einem besonders lichtempfindlichen Bildsensor ausgestattet ist. Dies hat zur Folge, dass die von der Kamera gelieferten Bilder von einem relativ starken Bildrauschen durchzogen sind, was sie für die Bildverarbeitung eigentlich unbrauchbar macht. Durch aufwendige Bildfilterungen konnte das Rauschen jedoch auf ein Minimum reduziert werden. Durch die hohe Lichtempfindlichkeit der Kamera reagiert die Anwendung entsprechend sensibel auf Änderungen der Raumbeleuchtung. Zusätzlich bedingt durch die implementierte Differenzbild-Methode führen wechselnde Lichtverhältnisse zur fehlerhaften Erkennung von Bewegungen. Durch die adaptive Auslegung des Hintergrundmodells passt es sich zwar den neuen Verhältnissen an, dies benötigt jedoch einige Zeit, während der keine brauchbare Auswertung der Bilder möglich ist.

Eingeschränkt durch die Infrastruktur im Living Place musste die implementierte Software auf mehrere Rechner verteilt werden. Dies hat zur Folge, dass die Auswertung der kapazitiven Sensoren auf einem MacMini im Wohnbereich stattfindet, während die Verknüpfung mit den Bilddaten und die Auswertung dieser auf einem Server im Kontrollraum ausgeführt wird. Der für die Auswertung der Sensoren zuständige Teil des Systems erstellt aus den Sensordaten ein 2-dimensionales Model des Sensorfeldes und veröffentlicht es über den zentralen

ActiveMQ-Server des Living Places. Die im Kontrollraum laufende Auswertungs-Anwendung verknüpft diese Daten mit dem aktuellen Bild der Kamera. Durch das verwendete, parametrierbare Interpretationsverfahren lässt sich so zuverlässig erkennen, ob eine Person auf dem Sensorfeld steht oder liegt. Abbildung 65 zeigt wie eine auf dem Sensorfeld liegende Person von der Kamera und den kapazitiven Sensoren erkannt wird. Das Muster des Sensorfeldes auf der linken Seite des Bildes deutet auf eine gestürzte Person hin, da die Kamera jedoch eine weitere Bewegung (rechtes Bild, weiße Silhouette am oberen Rand) im Raum detektiert, wird kein Alarm ausgelöst werden.

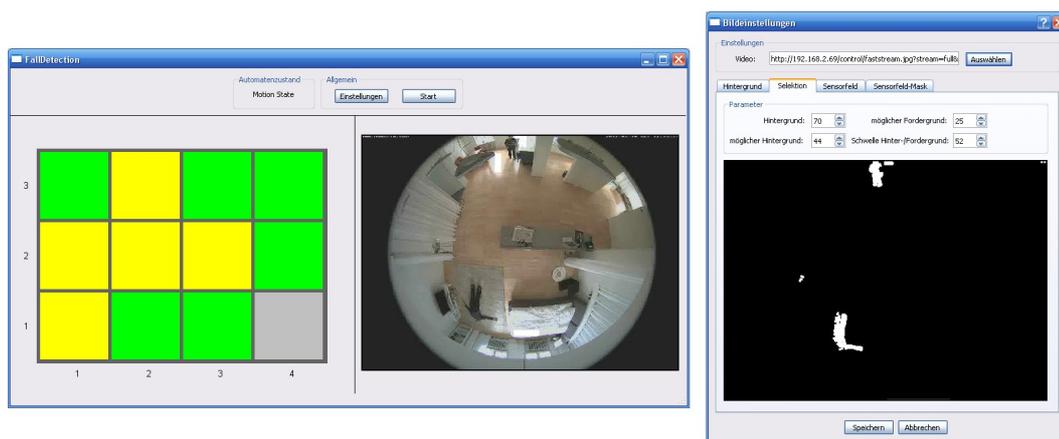


Abbildung 65: Erkennung einer liegenden Person durch Kamera und Sensorfeld

Befinden sich jedoch mehrere Personen auf dem Sensorfeld, kann nicht mehr sicher zwischen diesen unterschieden werden, was unter bestimmten Bedingungen zu Fehlmeldungen führt. Das umgesetzte System eignet sich daher vorwiegend für allein lebende Personen.

Rückblickend auf den Verlauf der Masterarbeit hat sich die Verwendung eines kapazitiven Messsystems mit Unterstützung durch Auswertung von Videodaten als gute Entscheidung erwiesen. Das entwickelte System bietet gute Erweiterungsmöglichkeiten und verwendet leicht verständliche Algorithmen. Auch die Entwicklung eigener kapazitiver Sensoren war eine richtige Entscheidung. Nach anfänglichen Hürden in der Herstellung werden die Sensoren bereits in mehreren Projekten eingesetzt bzw. ist deren Einsatz in Planung.

Während der Einsatz einer Kamera aus Gründen der Privatsphäre und des Datenschutzes im produktiven Einsatz außerhalb des Living Places vermutlich nicht möglich ist, bietet das entwickelte Sensorfeld großes Potential für die Unterstützung von Überwachungs- und Notrufsystemen.

8 Fazit

8.1 Zusammenfassung

In dieser Arbeit wurde im Kontext einer intelligenten Wohnung ein System zur kontaktlosen Erkennung von gestürzten und hilflosen Personen entwickelt. Hierzu wurden zunächst zwei Anwendungsszenarien vorgestellt, auf deren Basis anschließend die Rahmenbedingungen und Anforderungen an das System konkretisiert werden konnten. In Kapitel 3.3 erfolgte die Vorstellung verschiedener bereits verfügbarer, kommerzieller Produkte sowie aktuelle Forschungsprojekte, welche sich mit der Problematik der Sturzerkennung befassen. Im Anschluss daran wurden die zu verwendenden Sensoren näher betrachtet. Dies umfasst neben dem Einsatz einer Dome-Kamera, welche durch ihre spezifische Optik eine vollständige Rundumsicht ermöglicht, die Evaluierung verschiedener Messverfahren und Sensoren (Kapitel 3.4). Da sich Erschütterungssensoren aufgrund ihrer geringen Reichweite als ineffektiv erwiesen haben, fiel die Entscheidung auf den Einsatz kapazitiver Sensorik. Zur Erfassung der Daten wurden hierfür spezielle Sensor-Module entwickelt (Kapitel 4).

Aufbauend hierauf ließ sich anschließend ein aus mehreren Anwendungen bestehendes System entwickeln, welches durch Interpretation der Sensor- und Kameradaten am Boden liegende Personen erkennt. In Kapitel 5 wird näher auf den Aufbau des Systems eingegangen. Dies umfasst neben der Grundstruktur auch die Vorstellung verschiedener Verfahren zur Bildverarbeitung sowie unterschiedliche Verknüpfungsmöglichkeiten von Kamera und Sensoren. Im Anschluss hieran folgen Besonderheiten bei der technischen Umsetzung des Sensorfeldes sowie Feinheiten in der Software-Implementation. Den Abschluss dieser Arbeit bildet eine Evaluierung des Systems, in welcher auf die Ergebnisse und Beschränkungen der Software eingegangen wird.

8.2 Ausblick

Das in dieser Arbeit entwickelte System ist ein erster Ansatz zur berührungslosen kamera- und sensorgestützten Erkennung von hilflosen Personen. Als Ergebnis lassen sich zuverlässig regungslos am Boden liegende Personen erkennen. Die verwendete Technik bietet je nach Ausbaustufe jedoch noch großes Potential. So ist es durch den im Living Place vorhandenen Doppelboden beispielsweise denkbar, das System um mehrere kapazitive Sensoren zu erweitern, welche unter dem Boden installiert werden. Auch die Anbindung an einen leistungsstarken Rechner für eine erweiterte Analyse der Videobilder und eine Synchronisation mit weiteren Kameras wäre denkbar.

Des Weiteren bieten die entwickelten Sensoren Verwendungsmöglichkeiten für viele andere Szenarien. Beispielsweise lassen sich mit ihnen in einer entsprechend ausgestatteten Wohnung Bewegungen auf Polstermöbeln oder Betten detektieren und auswerten.

Durch den verteilten Aufbau des Systems ist es zusätzlich möglich, Teile des Aufbaus für andere Projekte zu verwenden. Da sich das Auslesen der Sensorfläche und die Auswertung der Daten in einer separaten Anwendung befindet, ist diese in ihrer Funktion vollständig unabhängig vom Rest des Systems. Auf diese Weise ist eine Verwendung des Sensorfeldes für andere Benutzerinteraktionen denkbar.

Als Ergebnis dieser Arbeit im Rahmen des Masterstudiums liefert das entwickelte System durch kontaktlose Sensorik gute Ergebnisse und Erkenntnisse im Bereich Ambient Assisted Living (AAL).

Technische Hilfsmittel stärken das Sicherheitsempfinden der allein lebenden Menschen, machen sie unabhängiger und stützen ihre Autonomie und somit ihre Würde.

AAL-Systeme können und sollen den helfenden Menschen dennoch nicht ersetzen. Sie können ihn jedoch im Bedarfsfall informieren und die Kommunikation mit ihm erleichtern. So muss eine Pflegekraft anhand der empfangenen Sensordaten immer noch selbst entscheiden, ob und wo persönliches Eingreifen von Nöten ist. Die Interpretationskompetenz hierfür ergibt sich aus Erfahrung und der persönlichen Kenntnis der Patienten und kann von keinem technischen Gerät ersetzt werden.

Gleichzeitig birgt der Einsatz von technischen Hilfsmitteln aber auch die Gefahr, dass die menschliche Zuwendung vernachlässigt wird. Z.B. könnte das Management einer Pflegeeinrichtung Kontrollsysteme einsetzen, um die Personalkosten zu senken. Vom ethischen Standpunkt gesehen, sollte das Ziel der Entwicklung von AAL-Systemen nicht die Gewinnmaximierung sein, sondern dem Pflegepersonal als Unterstützung bei der Ausübung ihrer verantwortungsvollen Arbeit dienen.

Abbildungsverzeichnis

Abbildung 1: Modell des Living Place Hamburg.....	5
Abbildung 2: Schlafbereich.....	5
Abbildung 3: Küche mit Multi-Touch-Tresen.....	5
Abbildung 4: Wohnbereich.....	5
Abbildung 5: Leidener Flasche.....	6
Abbildung 6: Seismograph.....	7
Abbildung 7: Ferroelektrischer Lautsprecher.....	7
Abbildung 8: Aufnahme einer Dome-Kamera.....	11
Abbildung 9: QuietCare Aufbau	13
Abbildung 10: Lifeline Button mit AutoAlert.....	14
Abbildung 11: SensFloor Matte.....	14
Abbildung 12: Beschleunigungsänderung während eines Sturzes.....	17
Abbildung 13: Luftdruckänderung während eines Sturzes.....	17
Abbildung 14: timed Motion History Image.....	18
Abbildung 15: Projektions-Histogramme.....	18
Abbildung 16: Änderung des Höhen/Breiten-Verhältnisses.....	19
Abbildung 17: Höhen/Breiten-Verhältnis.....	19
Abbildung 18: Vertikale Änderung der Kopfposition.....	20
Abbildung 19: Ferroelektrischer Lautsprecher.....	24
Abbildung 20: Erschütterungssensors Auswertung.....	24
Abbildung 21: Blockschaltbild des SCA3000.....	25
Abbildung 22: SCA3000 - Erschütterungen in geringem Abstand.....	26
Abbildung 23: SCA3000 - Erschütterungen in großem Abstand.....	26
Abbildung 24: Erster Testaufbau.....	28
Abbildung 25: Spannungsverlauf an der Elektrode im ersten Testaufbau.....	29
Abbildung 26: Kapazitätsmessung eines Feldes durch Zeitmessung.....	29
Abbildung 27: Kapazitätsmessung eines Kabels durch Zeitmessung.....	30
Abbildung 28: Aufbau QTouch®.....	30
Abbildung 29: Spannungsverlauf an der Elektrode bei QTouch.....	31
Abbildung 30: Blockdiagramm des ersten Prototyps.....	32
Abbildung 31: Annäherungsmessung – Erster Prototyp.....	32
Abbildung 32: CapToolKit – CapBoard.....	33
Abbildung 33: Kapazitätsmessung einer Aluminiumelektrode durch CapToolKit.....	34
Abbildung 34: Kapazitätsmessung eines Kabels durch CapToolKit.....	34
Abbildung 35: Schaltung des dritten Prototyps.....	35
Abbildung 36: Annäherungsmessung - Prototyp 3.....	35
Abbildung 37: Unterlage.....	37
Abbildung 38: Fehlinterpretation der Messwerte.....	37
Abbildung 39: HALCON-Architektur.....	39
Abbildung 40: Stromversorgung.....	41
Abbildung 41: Mikrocontroller.....	42
Abbildung 42: Abschluss mit Bias-Netzwerk.....	43
Abbildung 43: Sensor.....	44
Abbildung 44: Datenpaket.....	46
Abbildung 45: Softwareaufbau des Sensormoduls.....	48
Abbildung 46: Sensor-Modul - Empfang einer Nachricht.....	49
Abbildung 47: Ablauf des Hauptprogramms.....	50

Abbildung 48: Entwickelter kapazitiver Sensor.....	51
Abbildung 49: Firmware-Tool - Konfigurationsoberfläche.....	52
Abbildung 50: Firmware-Tool - Updateoberfläche.....	52
Abbildung 51: Multi-Sensor-Tool - Aufzeichnung des Werteverlaufs eines Sensors.....	53
Abbildung 52: Aufteilung des Systems in einzelne Komponenten.....	55
Abbildung 53: Ablauf der Sturzerkennung als Zustandsautomat.....	61
Abbildung 54: Mittelwerte und Abweichung zweier Sensoren.....	62
Abbildung 55: Bild-Vorverarbeitung.....	66
Abbildung 56: Differenzbild erstellen.....	67
Abbildung 57: Formel zur Korrektur der Klassifizierung.....	68
Abbildung 58: Binärbild erstellen.....	68
Abbildung 59: Fertige Sensorfläche.....	71
Abbildung 60: Teilweise beklebte Sensorfläche.....	71
Abbildung 61: Vollständige Sensorfläche.....	71
Abbildung 62: Lastverteilung in mehrere Threads.....	73
Abbildung 63: Hauptansicht der Software.....	77
Abbildung 64: Fenster zur Konfiguration der Sensoren	77
Abbildung 65: Erkennung einer liegenden Person durch Kamera und Sensorfeld.....	79

Tabellenverzeichnis

Tabelle 1: Vor- und Nachteile mehrerer kommerzieller Produkte	22
Tabelle 2: Vor- und Nachteile der vorgestellten kapazitiven Messverfahren	36
Tabelle 3: Vergleich OpenCV, ITK und HALCON	40

Listings

Listing 1: Konfigurationsdatei.....	72
Listing 2: Laden der Einstellungen.....	73
Listing 3: Differenzbilderstellung mit mehreren Threads.....	74
Listing 4: Auslesen der Sensoren.....	75
Listing 5: Aufbau der Json-Nachricht.....	75
Listing 6: IState Klasse.....	76

Literaturverzeichnis

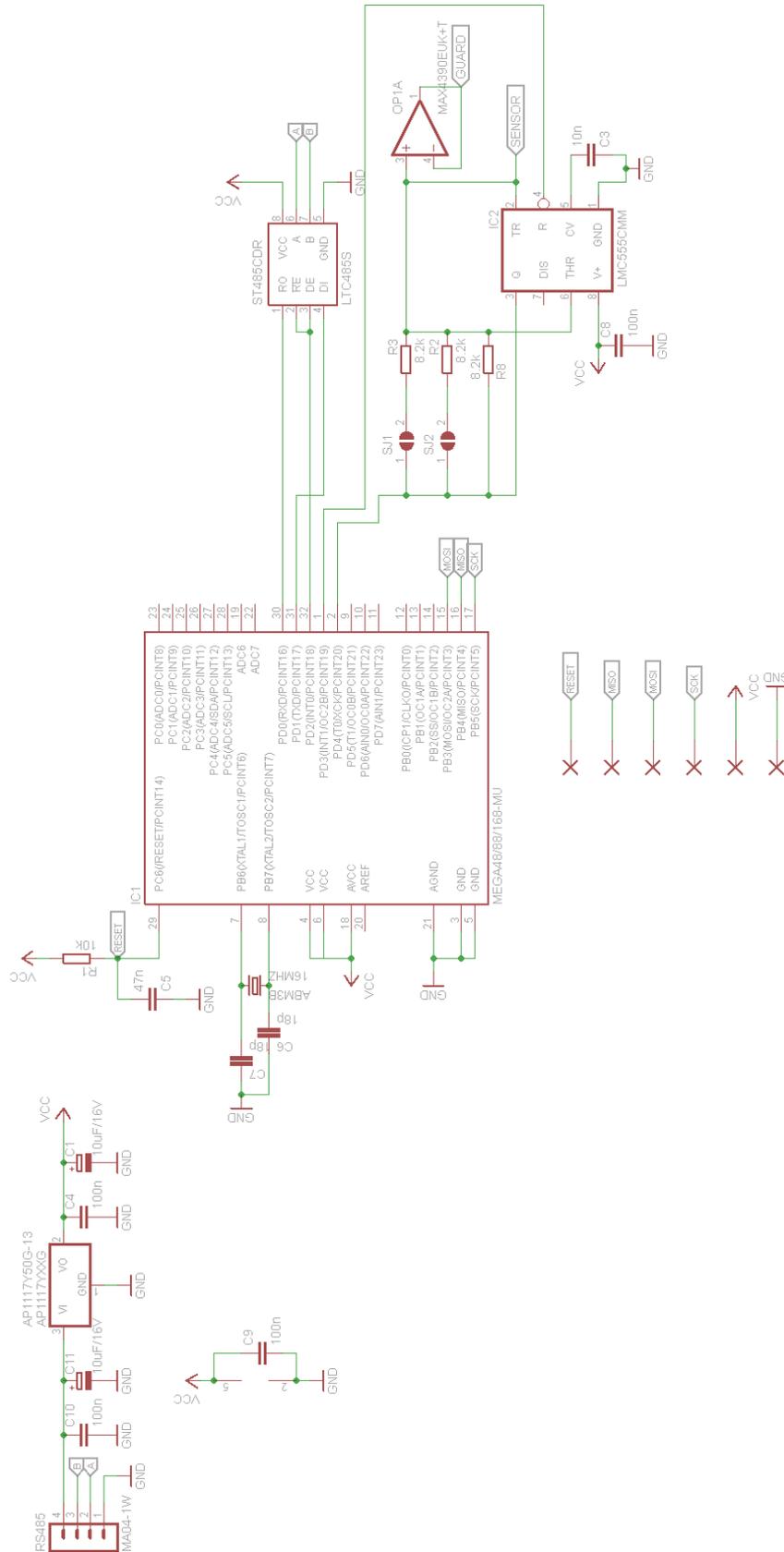
- 1 Allan Mills: *Studies in Electrostatics - Part 6: The Leyden jar and Other Capacitors*, In: Bulletin of the Scientific Instrument Society No. 99, 2008, S. .
- 2 Ann-Kathrin Weiss: *Entwicklung textiler Sensoren für intelligente Umgebungen am Beispiel eines Sofas*. Hamburg, HAW Hamburg, 2011
- 3 Atmel : *QTouch Library User Guide*, Mai 2010.
- 4 B. Heisele, T. Poggio, M Pontil: *Face detection in still gray images*, 2000.
- 5 Baptiste Lepilleur: *JsonCpp*. - URL <http://jsoncpp.sourceforge.net>. - Zugriffsdatum: 17.12.2011
- 6 Business Applications Performance Corporation: *SYSmark 2012*. - URL <http://www.bapco.com/products/sysmark2012/>. - Zugriffsdatum: 17.12.2011
- 7 Chris Burks: *Leyden jar*. - URL http://http://en.wikipedia.org/wiki/Leyden_jar. - Zugriffsdatum: 23.01.2012
- 8 Deutsches Rotes Kreuz e.V.: *Hausnotruf - DRK*. - URL <http://www.drk.de/angebote/senioren/hausnotruf.html>. - Zugriffsdatum: 23.01.2012
- 9 Dipl.-Ing. Marian Lüder, Dipl.-Ing. Gerald Bieber, Prof. Dr.-Ing. habil. Ralf Salomon: *Air Pressure- and Acceleration-Based Fall Detector*, 2009.
- 10 Dipl.-Ing. Marian Lüder, Prof. Dr.-Ing. habil. Ralf Salomon, Dipl.-Ing. Gerald Bieber: *StairMaster: A New Online Fall Detection Device*, 2009.
- 11 EDIS Interactive: *EDIS Interactive*. - URL <http://edis.mx>. - Zugriffsdatum: 17.12.2011
- 12 Forschungszentrum Karlsruhe: *Mikro-Vibrationsschalter*. - URL http://www.mstbw.de/imperia/md/content/mstbw/bestpractice/fzk_mikro_vibrationsschalter.pdf. - Zugriffsdatum: 17.12.2011
- 13 Frank Hardenack: *Das intelligente Bett - Sensorbasierte Detektion von Schlafphasen*. Hamburg, HAW Hamburg, 2011
- 14 Frank Sposaro, Gary Tyson: *iFall: An Android Application for Fall Monitoring and Response*, In: 31st Annual International Conference of the IEEE EMBS, 2009, S. 6119-6122. - ISSN: 1557-170X
- 15 Fraunhofer-Institut für Integrierte Schaltungen IIS: *MotionSENS - Bewegungssensor mit Sturzdetection – Fraunhofer-Institut für Integrierte Schaltungen IIS*. - URL <http://www.iis.fraunhofer.de/bf/med/sens/motionsens>. - Zugriffsdatum: 23.01.2012
- 16 Future-Shape GmbH: *SensFloor® Matte - für den Einsatz in der professionellen Pflege*. - URL www.future-shape.de/download/21/Preis+Pflege+Stationär+SE2-R.pdf. - Zugriffsdatum: 23.01.2012
- 17 Future-Shape GmbH: *Future Shape | HighTech mit hoher Flexibilität*. - URL <http://www.future-shape.com>. - Zugriffsdatum: 17.12.2011
- 18 Future-Shape GmbH: *Startseite - Verbundprojekt SensFloor*. - URL <http://www.sensfloor.de>. - Zugriffsdatum: 23.01.2012
- 19 Future-Shape GmbH: *SensFloor® Matten / SensFloor® Mats*, 2010.

- 20 H. Elzein, S. Lakshmanan, P. Watta: *A motion and shape-based pedestrian detection algorithm*, In: Intelligent Vehicles Symposium, 2003, S. 500-504. - ISBN: 0-7803-7848-2
- 21 Hansung Kim, Ryuuki Sakamoto, Itaru Kitahara, Tomoji Toriyama, Kiyoshi Kogure: *Robust Foreground Segmentation from Color Video Sequences Using Background Subtraction with Multiple Thresholds*, 2007.
- 22 Homa Foroughi, Baharak Shakeri Aski, Hamidreza Pourreza: *Intelligent Video Surveillance for Monitoring Fall Detection of Elderly in Home Environments*, In: 11th International Conference on Computer and Information Technology, Dezember 2008, S. 219-224. - ISBN: 978-1-4244-2135-0
- 23 Home For Life Solutions : *Aging In Place Technology | HomeFor Life Solutions*. - URL <http://www.homeforlifesolutions.com>. - Zugriffsdatum: 23.01.2012
- 24 Hosnia Najem: *Projektbericht PO2 - Modellbasiertes Suchen von Objekten in einer Smart-Home-Umgebung*, 2011.
- 25 Jacob Fraden: *Handbook of modern sensors*, Springer, - ISBN 0-387-00750-4
- 26 Jean-Pierre Charras: *KiCad*. - URL http://www.lis.inpg.fr/realise_au_lis/kicad/. - Zugriffsdatum: 17.12.2011
- 27 Jianpeng Zhou, Jack Hoang: *Real Time Robust Human Detection and Tracking System*, In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, 2005, S. 149-149. - ISSN: 1063-6919
- 28 Johanniter-Unfall-Hilfe e. V.: *Hausnotruf*. - URL <http://www.johanniter.de/dienstleistungen/hausnotruf/hausnotruf>. - Zugriffsdatum: 11.01.2012
- 29 K. Kim, T. H. Chalidabhongse, D. Harwood, L. Davis: *Real-time foreground-background segmentation using codebook model*, In: , Juni 2005, S. 172-185. - ISSN: 1077-2014
- 30 Kitware, Inc.: *ITK - Segmentation & Registration Toolkit*. - URL <http://www.itk.org>. - Zugriffsdatum: 23.01.2012
- 31 Kjell Otto, Sören Voskuhl: *Projektbericht Sommersemester 2010 - Entwicklung einer Architektur für den Living PlaceHamburg*, 2010.
- 32 Lee Thomason: *TinyXml Main Page*. - URL <http://www.grinninglizard.com/tinyxml/>. - Zugriffsdatum: 23.01.2012
- 33 Malteser Hilfsdienst e.V.: *Malteser Hausnotruf*. - URL <http://www.malteser-hausnotruf.de>. - Zugriffsdatum: 11.01.2012
- 34 Michael J. Jones, James M. Rehg: *Statistical Color Models with Application to Skin Detection*, In: International Journal of Computer Vision Volume 46 Issue 1, 2002, S. 81-96. - ISSN: 0920-5691
- 35 Mousa Mojarrad, Amir Masoud Rahmani, and Mehrab Mohebi: *Detection and Pose Estimation of People in Images*, In: World Academy of Science 49, 2009, S. 985-990. - ISSN: 2010376X
- 36 Muhammad Jamil Khan, Hafiz Adnan Habib: *Video Analytic for Fall Detection from Shape Features and Motion Gradients*, In: Lecture Notes in Engineering and Computer Science, 2009, S. 1311-1316. - ISSN: 20780958
- 37 MVTec Software GmbH: *MVTec HALCON*. - URL <http://www.mvtec.com/halcon/>. - Zugriffsdatum: 17.12.2011
- 38 MVTec Software GmbH: *HALCON 9.0 Quick Guide*, 2008.
- 39 N Noury, A Fleury, P Rumeau, A Bourke, G Laighin, VRialle, J Lundy: *Fall detection - Principles and Methods*, In: Conf Proc IEEE Eng Med Biol Soc. 2007, Oktober 2007, S. 1663-1666. - ISSN: 1557-170X

- 40 NetResults S.r.l.: *VDK - VoIP Development Kit*. - URL <http://www.voipdevelopmentkit.com>. - Zugriffsdatum: 17.12.2011
- 41 Nokia Corporation: *Qt - A cross-platform application and UI framework*. - URL <http://qt.nokia.com>. - Zugriffsdatum: 17.12.2011
- 42 o,V.: *Connected Component Labeling*. - URL http://en.wikipedia.org/wiki/Blob_extraction. - Zugriffsdatum: 17.12.2011
- 43 o,V.: *GIMP - The GNU Image Manipulation Program*. - URL <http://www.gimp.org>. - Zugriffsdatum: 17.12.2011
- 44 o,V.: *Inkscape. Draw Freely*. - URL <http://inkscape.org>. - Zugriffsdatum: 17.12.2011
- 45 o,V.: *Stellarium*. - URL <http://www.stellarium.org>. - Zugriffsdatum:
- 46 o,V.: *Audacity: Freier Audioeditor und Rekorder*. - URL <http://audacity.sourceforge.net/>. - Zugriffsdatum: 17.12.2011
- 47 o.V.: *wxWidgets*. - URL <http://www.wxwidgets.org>. - Zugriffsdatum: 17.12.2011
- 48 o.V.: *Ferroelektrischer Lautsprecher*. - URL http://de.wikipedia.org/wiki/Ferroelektrischer_Lautsprecher. - Zugriffsdatum: 23.01.2012
- 49 Oliver Dreschke: *Entwicklung kontextsensitiver Möbel für intelligente Wohnumgebungen*. Hamburg, HAW Hamburg, 2011
- 50 Openmoko, Inc.: *Openmoko, Inc.*. - URL <http://www.openmoko.com>. - Zugriffsdatum: 17.12.2011
- 51 Philips Electronics: *News - Philips*. - URL <http://www.healthcare.philips.com/wpd.aspx?p=/Resources/News.wpd&id=2075&c=global>. - Zugriffsdatum: 23.01.2012
- 52 Philips Electronics: *Lifeline ® - the trusted medical alert service provider*. - URL <http://www.lifelinesys.com>. - Zugriffsdatum: 23.01.2012
- 53 Prof. Dr. K. von Luck, Prof. Dr. G. Klemke, S. Gregor, Mohammad A. Rahimi, M. Vogt: *Living Place Hamburg - A place for concepts of IT based modern living*, 2010.
- 54 Prof. G. Stockman & Prof. L. Shapiro: *Computer Vision*, Prentice Hall, 2001 - ISBN: 0130307963
- 55 QuietCare: *QuietCare*. - URL <http://www.careinnovations.com/Products/QuietCare/Default.aspx>. - Zugriffsdatum: 17.12.2011
- 56 Raphael Wimmer: *Capacitive Sensors for Whole Body Interaction*, 2011.
- 57 Raphael Wimmer: *CapToolKit*. - URL <http://www.capsense.org>. - Zugriffsdatum: 17.12.2011
- 58 Steffen Kröhnert, Franziska Medicus, Rainer Klingholz: *Die demographische Lage der Nation - Kurzfassung*, 2006.
- 59 Sven Stegelmeier, Piotr Wendt, Kai von Luck: *iFlat - Eine dienstorientierte Architektur für intelligente Räume*, 2009.
- 60 T. Haga, K. Sumi, and Y. Yagi.: *Human detection in outdoor scene using spatio-temporal motion analysis*. In: 17th International Conference on Pattern Recognition, 2004, S. 331 - 334. - ISSN: 1051-4651
- 61 T. Lee, A. Mihailidis: *An intelligent emergency response system: preliminary development and testing of automated fall detection*, In: Journal of telemedicine and telecare vol. 11 no. 4, Juni 2005, S. 194-198.

- 62 Texas Instruments Incorporated: *RS-485: Passive failsafe for an idle bus*, In: Analog Applications Journal, 2009, S. 22-24.
- 63 The Apache Software Foundation: *Apache ActiveMQ CPP*. - URL <http://activemq.apache.org/cms/index.html>. - Zugriffsdatum: 23.01.2012
- 64 Tong Zhang, Jue Wang, Ping Liu and Jing Hou: *Fall Detection by Embedding an Accelerometer in Cellphone and Using KFD Algorithm*, Oktober 2006.
- 65 Uwe Mnich: *umni's Homepage - Projekte - Wetterstation - Sensoren - Vertikal Seismometer*. - URL <http://umnicom.de/Elektronik/Projekte/Wetterstation/Sensoren/VertikalSeismometer/VertikalSeismometer.htm>. - Zugriffsdatum: 23.01.2012
- 66 Uwe Rathmann, Josef Wilgen: *Qwt - Qt Widgets for Technical Applications*. - URL <http://qwt.sourceforge.net>. - Zugriffsdatum: 17.12.2011
- 67 V. Ferrari, M. Marín-Jiménez, A. Zisserman: *Pose Search: retrieving people using their pose*, In: IEEE Conference on Computer Vision and Pattern Recognition, 2009, S. 1 - 8. - ISSN: 1063-6919
- 68 Willow Garage: *Welcome - OpenCV Wiki*. - URL <http://http://opencv.willowgarage.com/wiki/>. - Zugriffsdatum: 17.12.2011

A Schaltplan des Sensors



B Befehlsübersicht des Sensors

Byte-Nr	1	2	3	4	5	6	7	8	9	10	11
Reset											
PC → Sensor	#	Size	Src	Dst	0	Checksum					
Ping											
PC → Sensor	#	Size	Src	Dst	1	Number	Checksum				
Sensor → PC	#	Size	Src	Dst	1	Number	Checksum				
Get ID											
PC → Sensor	#	Size	Src	FF	2	Serial	Serial	Serial	Serial	Checksum	
Sensor → PC	#	Size	Src	Dst	2	Serial	Serial	Serial	Serial	Checksum	
Set ID											
PC → Sensor	#	Size	Src	FF	3	Id	Serial	Serial	Serial	Serial	Checksum
Get Serial											
PC → Sensor	#	Size	Src	Dst	4	Checksum					
Sensor → PC	#	Size	Src	Dst	4	Serial	Serial	Serial	Serial	Checksum	
Get Version											
PC → Sensor	#	Size	Src	Dst	5	Checksum					
Sensor → PC	#	Size	Src	Dst	5	Version	Version	Checksum			
Set Baud											
PC → Sensor	#	Size	Src	Dst	6	Baud	Baud	Baud	Baud	Checksum	
Set MeasureTime											
PC → Sensor	#	Size	Src	Dst	7	Time	Time	Checksum			
Get MeasureTime											
PC → Sensor	#	Size	Src	Dst	8	Checksum					
Sensor → PC	#	Size	Src	Dst	8	Time	Time	Checksum			
Set MeasureMode											
PC → Sensor	#	Size	Src	Dst	9	Mode	Checksum				
Get MeasureMode											
PC → Sensor	#	Size	Src	Dst	10	Checksum					
Sensor → PC	#	Size	Src	Dst	10	Mode	Checksum				
Get Value											
PC → Sensor	#	Size	Src	Dst	11	Checksum					
Sensor → PC	#	Size	Src	Dst	11	Value	Value	Value	Value	Checksum	
StartMeasuring											
PC → Sensor	#	Size	Src	Dst	12	Checksum					

0X00 – Reset

Startet das Modul neu.

Parameter:

keine

Rückgabewert:

keine

0x01 – Ping

Prüft ob ein Modul erreichbar ist.

Parameter:

beliebige 8-Bit Zahl

Rückgabewert:

Die als Parameter übergebenen Zahl

0x02 – Get ID *(In der aktuellen Softwareversion nicht mehr implementiert)*

Wird als Broadcast-Nachricht versandt.

Liefert die ID eines Moduls.

Parameter:

4-Byte Seriennummer

Rückgabewert:

Die als Parameter übergebenen Seriennummer, im Source-Feld des Datenpakets steht anschließend die ID des Moduls.

0x03 – Set ID

Wird als Broadcast-Nachricht versandt.

Legt die ID eines Moduls fest.

Parameter:

4-Byte Seriennummer

Rückgabewert:

keiner

0x04 – Get Serial

Liefert die Seriennummer eines Moduls.

Parameter:

keine

Rückgabewert:
4-Byte Seriennummer

0x05 – Get Version

Liefert die Versionsnummer der Software eines Moduls.

Parameter:

keine

Rückgabewert:

2-Byte Softwareversion. Das erste Byte beschreibt die Nummer vor der Dezimalstelle, das zweite Byte die Zahl hinter der Dezimalstelle.

0x06 – Set Baudrate

Wird als Broadcast-Nachricht versandt.
Legt die Übertragungsgeschwindigkeit fest.

Parameter:

4-Byte Baudrate

Rückgabewert:

keine

0x07 – Set Measure Time

Legt die Dauer einer Messung fest.

Parameter:

2-Byte Messdauer

Rückgabewert:

keine

0x08 – Get Measure Time

Liefert die Messdauer eines Moduls.

Parameter:

keine

Rückgabewert:

2-Byte Messdauer

0x09 – Set Measure Mode

Legt den Messmodus fest.

00: Einfach Messung

01: Kontinuierlich Messung. Sobald eine Messung abgeschlossen ist, wird die nächste gestartet.

Parameter:

1-Byte Messmodus

Rückgabewert:

keine

0x0A – Get Measure Mode

Liefert den Messmodus eines Moduls.

00: Einfache Messung

01: Kontinuierlich Messung

Parameter:

keine

Rückgabewert:

1-Byte Messmodus

0x0B – Get Value

Liefert das Ergebnis der letzten Messung.

Parameter:

keine

Rückgabewert:

4-Byte Messergebnis

0x0C – Start Measuring

Startet die Messung.

Parameter:

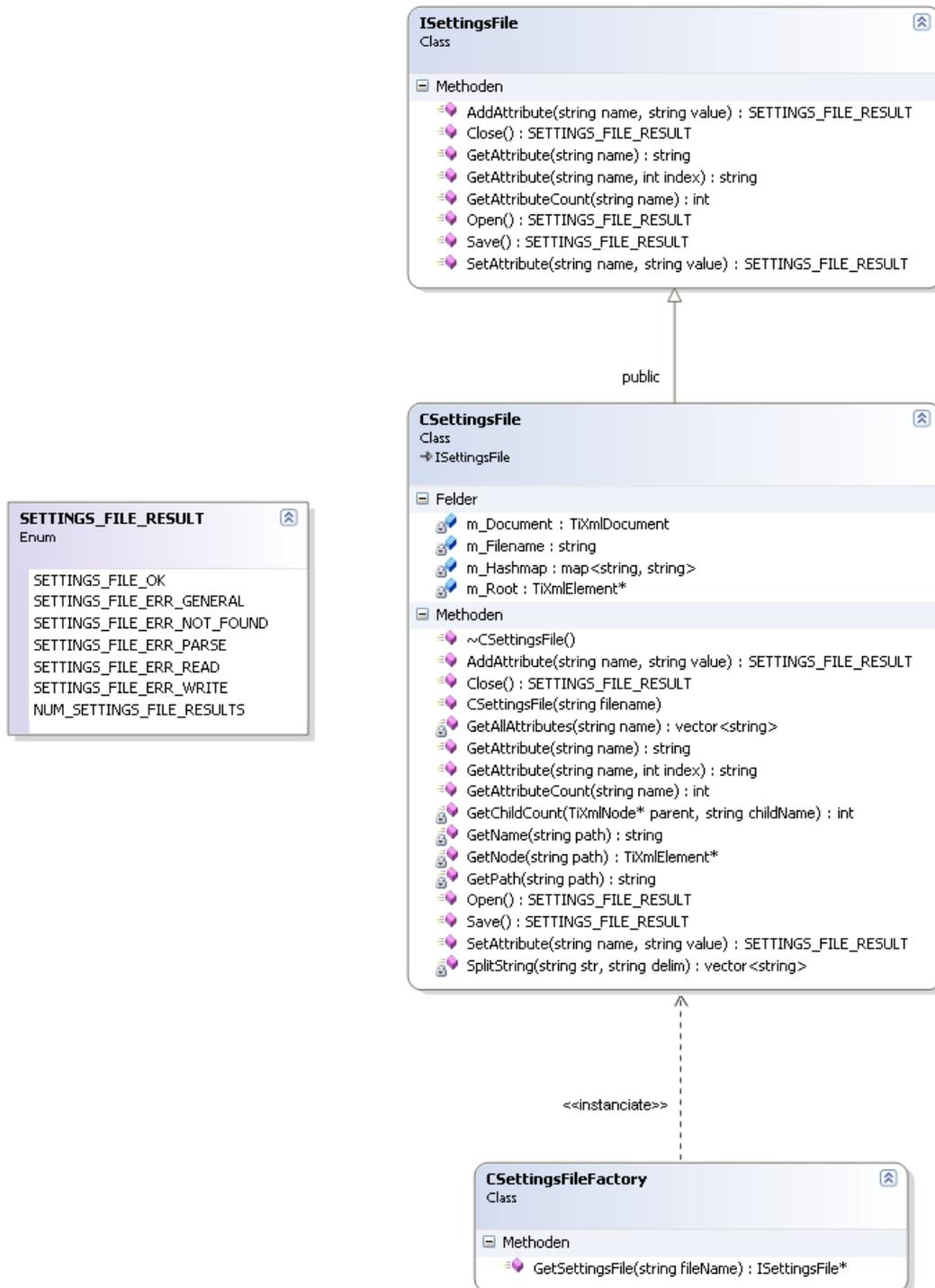
keine

Rückgabewert:

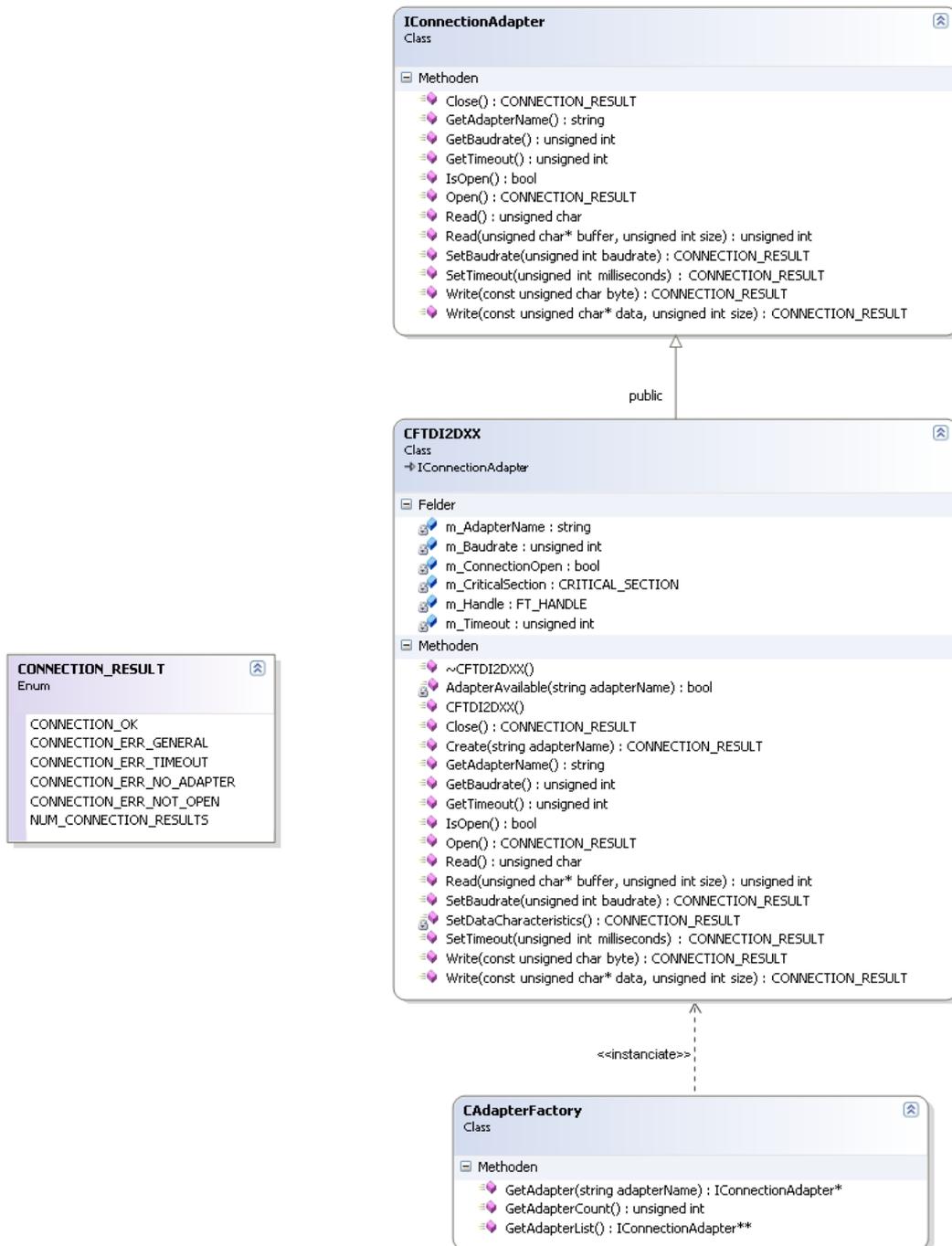
keine

C Klassendiagramme

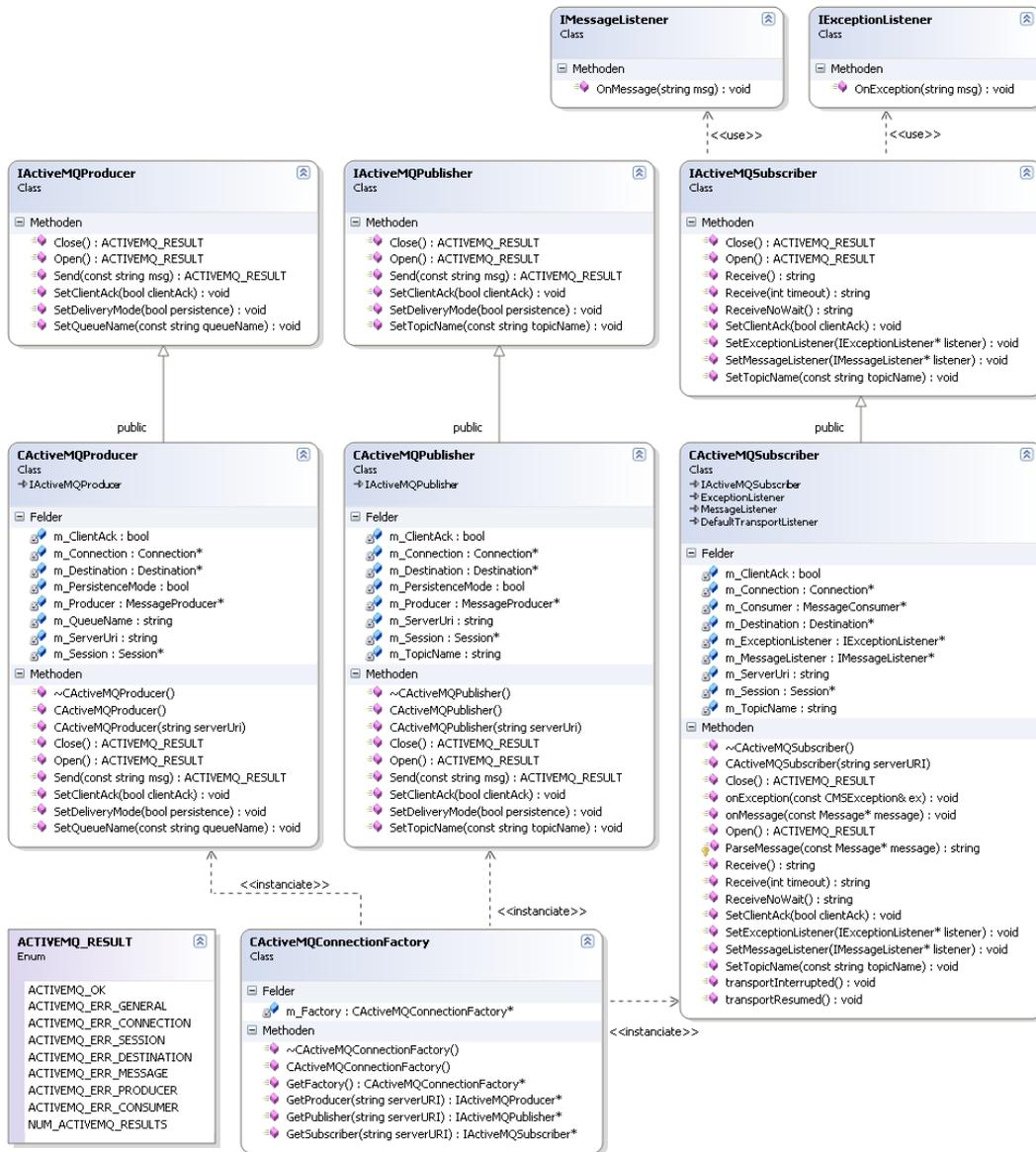
C.1 SettingsFileHandler



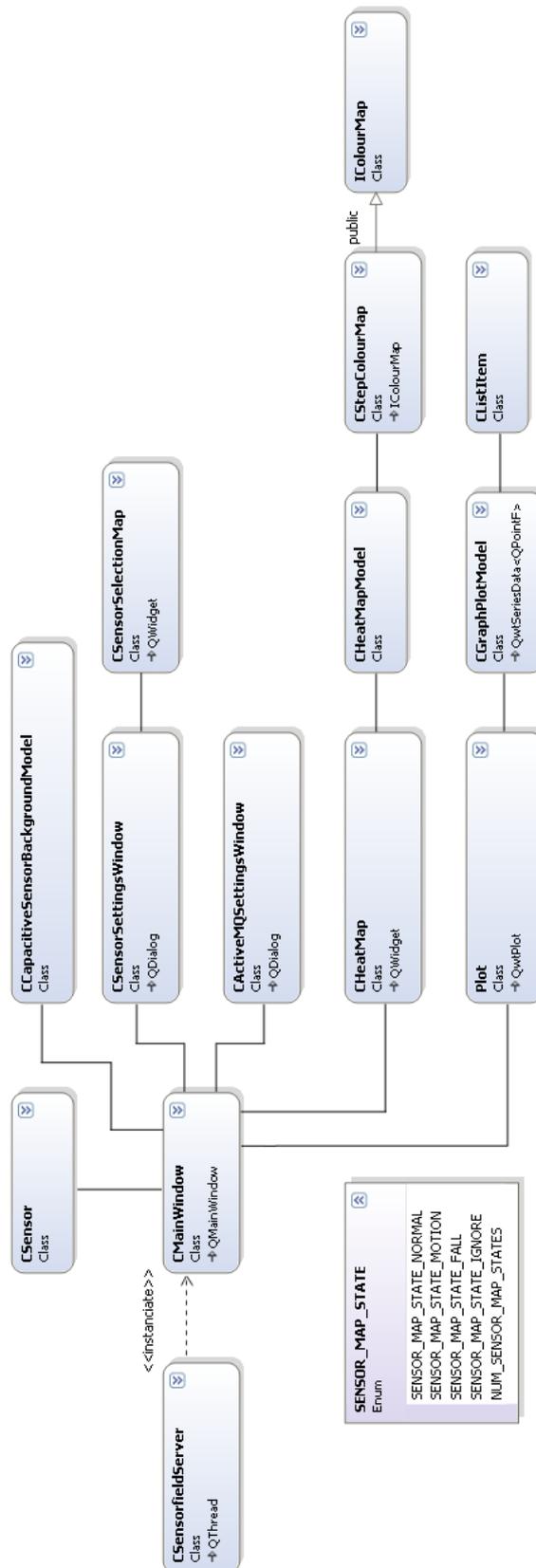
C.2 ConnectionAdapter



C.4 ActiveMQConnection



C.7 Sensorfeld-Server



Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §22(4) bzw. §24(4) ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, _____

Ort, Datum

Unterschrift