Bachelorarbeit bei MCS DataLabs

Vorbereitung eines Smartbracelets für eine Machine-learning basierte Sturzerkennung

Kai Kruschel
Matrikelnummer: 4304673
kai.kruschel@fu-berlin.de

Betreuer: Fihmi Mousa
Eingereicht bei: Prof. Dr. Raúl Rojas

Berlin, den 9. März 2017

Zusammenfassung

Es soll eine erste Vorauswahl für eine Sturzerkennung in die Funktionalität eines bestehenden Smart Bracelets integriert und die erhobenen Daten im Anschluss zur weiteren Verarbeitung an einen Server gesendet werden. Zunächst werden unterschiedliche aktuelle Formen der Sturzerkennung diskutiert. Im Anschluss werden einige repräsentative Beispiele sowie das verwendete Gerät vorgestellt. Danach wird der Algorithmus für die grobe Sturzerkennung auf dem Gerät selbst erklärt und die konkrete Implementierung mit ihren speziellen Herausforderungen und Problemen erläutert. Schließlich werden mögliche Verwendungen der generierten Daten und nächste Schritte bei der weiteren Verbesserung des Algorithmus aufgezeigt und diskutiert.

Eidesstattliche Erklärung

Ich versichere hiermit an Eides Statt, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder ähnliches sind im Literaturverzeichnis angegeben, Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Berlin, den 9. März 2017

Kai Kruschel

Inhaltsverzeichnis

Es wurde versucht, wenn möglich geschlechtsneutrale Begriffe und Bezeichnungen zu verwenden. An den Stellen, wo das nicht möglich war, wird der Lesbarkeit halber nur die weibliche Form genutzt. Diese soll hier aber, solange keine bestimmte Person beschrieben wird, als genderneutral verstanden werden.

1 Einleitung

Das Durchschnittsalter in Deutschland und vielen anderen Industrienationen steigt mit besserer medizinischer Pflege beständig an. Bis 2050 wird laut Aussage des Berlin-Instituts für Bevölkerung und Entwicklung die Anzahl der über 60 Jährigen von ca. 25% auf 37% gestiegen sein[Kröhnert et al., 2006]. Das bedeutet, ein immer größerer Bevölkerungsteil ist altersbedingten Risiken ausgesetzt. Eine der größten Gefahren sind dabei Stürze. Laut WHO stürzen 28-35% aller Menschen über 65 weltweit[Ageing and Unit, 2008].

Definition eines Sturzes Funk und Pierobon definieren einen Sturz wie folgt: „Ein Sturz ist ein plötzliches, nicht willentlich beeinflussbares Gelangen auf den Boden oder eine andere, im Vergleich zur Ausgangslage deutlich tiefer gelegene Ebene.[...]“[Funk and Pierobon, 2007].

Sturzrisiken Während Kinder und jüngere Menschen häufig keinen Schaden davon tragen oder Verletzungen wie Prellungen und Knochenbrüche in vielen Fällen schnell heilen, werden Stürze mit zunehmendem Alter zu einem immer schwerwiegenderen Problem. Fragilere Knochen und schwindende Muskelmasse bedeuten, dass Stürze häufiger zu ernsthaften Verletzungen oder sogar bleibenden Schäden führen. Infektionen infolge von Verletzungen stellen eine größere Gefahr dar und ältere Menschen haben oft Schwierigkeiten, sich nach einem Sturz von selbst wieder aufzurichten oder Hilfe zu holen.

Möglichkeiten zur Prävention Es können Vorkehrungen getroffen werden, um die negativen Folgen von Stürzen zu minimieren. Das Präparieren des Lebensbereichs, z.B. durch das Abrunden von Ecken, die Vermeidung von verwinkelten und schmalen Durchgängen und das Tragen von Protektoren in und an der Kleidung stellt eine Möglichkeit dar, Stürze zu vermeiden und das Verletzungsrisiko zu vermindern[Igual et al., 2013]. Vollends vermieden werden können diese dadurch allerdings nicht. Auch wenn der Sturz selbst ohne Verletzung überstanden wurde, bleibt die Unfähigkeit der Gestürzten, sich selbstständig aufzurichten als mögliche Ursache für Folgeverletzungen. All das spricht für eine Überwachung durch Pflegepersonal, das bei Bedarf schnell eingreifen kann. Diese Überwachung setzt jedoch voraus, dass die Patientin sich in einer betreuten Wohneinrichtung befindet und große Abstriche in der Privatsphäre und Selbstbestimmung in Kauf nehmen muss.

Medizinische Überwachung und Selbstbestimmtes Leben Um eine hohe Lebensqualität auch im Alter zu gewährleisten ist es unabdingbar, diese Einschränkungen soweit wie möglich zu minimieren.

Automatisierte Sturzüberwachung kann sowohl die Privatsphäre schützen als auch ältere Menschen länger befähigen, selbstständig in der eigenen Wohnung zu leben. Dabei muss das System diesen sowohl die Möglichkeit bieten, selbstständig Hilfe anzufordern, als auch Notfälle selbst erkennen und darauf reagieren können.
Eine robust funktionierende Sturzerkennung stellt hierbei eine Kernfunktionalität dar. Insbesondere sind das Risiko von Stürzen und die Angst vor Stürzen direkt miteinander verknüpft. Das Wissen um die Sturzerkennung kann dabei Betroffenen Sicherheit geben und damit die Angst vor Stürzen reduzieren. Dies hat wiederum direkte Auswirkungen nicht nur auf das Sturzrisiko, sondern auch auf die generelle Lebensqualität[Friedman et al., 2002].

Ziel dieser Arbeit Im Rahmen dieser Arbeit soll der erste Schritt für die Integration einer solchen Sturzerkennung auf einem bestehenden Smart Bracelet getan werden. Als Smart Bracelet wird hier ein Microcontroller mit verschiedenen Sensoren bezeichnet, der ähnlich einer Armbanduhr am Handgelenk getragen wird. Als Teil der SmarKo Plattform soll dieses Smart Bracelet auf Basis eines 9-Achsigen Bewegungssensors sowie verschiedener Vitalsensoren und anderer Parameter eine Vorauswahl von möglichen Stürzen treffen können. Die gesammelten Daten sollen dann aufbereitet und zur weiteren Verarbeitung an einen Server gesendet werden.

2 Stand der Technik

Igual et al. unterscheiden zwischen zwei Hauptkategorien: Kontextsensitive Systeme und Wearables[Igual et al., 2013]. Dabei sind kontextsensitive Systeme solche, die im Umfeld der Patientin fest installiert sind und beinhalten Sensoren wie Kameras, Drucksensoren im Boden, Infrarotsensoren und Mikrophone. Wearables sind relativ kleine sensorbasierte Geräte, die am Körper getragen werden. Diese basieren in der Regel auf Accelerometern, können aber auch Gyroskope und andere Sensoren beinhalten. Wearables gibt es sowohl in der Form dedizierter Geräte, als auch integriert in Hörgeräte[Lindemann et al., 2005] und Mobiltelefone sowie in Softwareform, die sich die bestehende Sensorik in Smartphones zu Nutze macht([Dai et al., 2010], [Abbate et al., 2012])

2.1 Kontextsensitive Systeme

Diese Kategorie beinhaltet Systeme die mit Hilfe von digitaler Bildverarbeitung Videos auswerten sowie solche, die fix verbaute Sensoren verwenden. Die Hauptprobleme in dieser Kategorie sind Datenschutz und die Ortsbindung. Eine durchgängige Videoüberwachung stellt eine starke Einschränkung der Privatsphäre dar. Ein Versuch zur Lösung dieses Problems ist, das Kamerabild zu verfremden, um ein Minimum an Privatsphäre zu gewährleisten(Vgl. [Edgcomb and Vahid, 2012]). Allerdings lässt auch dieser Ansatz zu Wünschen übrig. Druck- oder Infrarotsensoren können an Stellen verwendet werden, an denen ein Sturz besonders wahrscheinlich ist, als Beispiel vor einem Bett oder einer Sitzgelegenheiten. Allerdings müssen dabei diese Stellen vorhergesehen werden und die Kosten steigen mit der Anzahl an verbauten Sensoren schnell an.

2.2 Wearables

Am Körper getragene Sensoren haben den Vorteil, dass sie abhängig von der Implementierung ortsunabhängig funktionieren und im Vergleich mit verbauten Lösungen häufig einen wesentlich geringeren Kostenaufwand verursachen.
In verschiedenen Arbeiten zu dem Thema wurden gute Erfolge damit erziehlt, Beschleunigungssensoren am Oberkörper([Chen et al., 2006], [Bourke et al., 2007], [Zhang et al., 2006]) oder am Kopf([Lindemann et al., 2005]) zu befestigen und die dadurch gewonnenen Daten mit denen eines Lagesensors zu kombinieren. Die Nähe zum Körperschwerpunkt und die einfache Lagebestimmung des ganzen Körpers machen dabei die Sturzerkennung bedeutend einfacher, als an anderen Körperstellen, insbesondere am Handgelenk([Kangas et al., 2008]).
Allerdings kann ein Gerät, das konstant am Oberkörper getragen werden

muss, störend und ungewohnt wirken und dadurch auch häufiger vergessen werden. Die Folge kann ein stark reduzierter Nutzen sein. Insbesondere Sturzsensoren, die keine weiteren Funktionen bieten können schnell von diesem Problem betroffen sein.
Eine integrierte Lösung, insbesondere in einer Form, die die Nutzerin schon gewohnt ist, bietet sich daher an. Sturzsensortechnik existiert in Hörgeräten[Lindemann et al., 2005], in Mobiltelefonen, in Smart Phones[Aguiar et al., 2014] sowie seit kurzer Zeit in Smart Watches[Kostopoulos et al., 2016]. Insbesondere Smart Watches und am Handgelenk getragene Sensoren bieten eine leichte Gewöhnung an das Gerät aufgrund der Ähnlichkeit zu herkömmlichen Uhren. Einen solchen Sensor um eine Zeitanzeige zu Erweitern und damit die Funktionalität für die Nutzerin deutlich zu erhöhen ist ebenfalls sehr einfach möglich. Da das verwendete Gerät in diese Gruppe fällt, soll sich bei den Beispielen auf diese Art von Lösungen konzentriert werden.

2.3 Beispiele für Sturzerkennung auf Wearables

Gemein haben alle diese Beispiele eine Sturzerkennung die im wesentlichen auf den Daten eines Accelerometers basiert.

2.3.1 SPEEDY

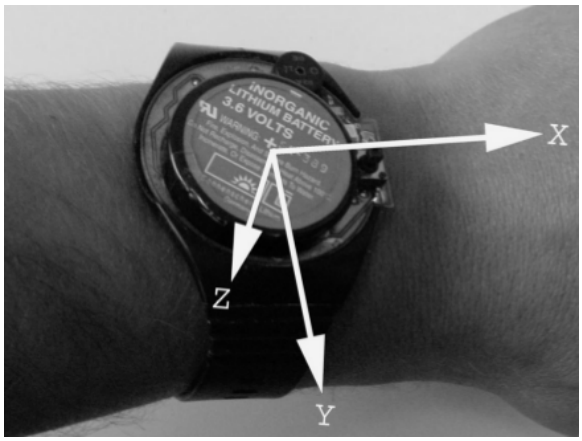

Abbildung 1: Speedy und die drei Achsen des Beschleunigungssensors
Quelle: [Degen et al., 2003]

2003 stellten Degen, Jaeckel, Rufer und Wyss eine handgelenkbasierte Lösung basierend auf einem Microcontroller, nämlich ein MSP430 von Texas Instruments, einem 3-Achsen Beschleunigungssensor(Accelerometer) und einer über Funk verbundenen Basisstation vor(Abb.1). Die Sturzerkennung wird nahezu ausschließlich auf dem Gerät selbst durchgeführt. Der Ablauf

hierbei ist in drei Phasen aufgeteilt. Abb. 2 zeigt die Messung eines repräsentativen Sturzes und seine Aufteilung in diese drei Phasen. Phase 1 ist eine schnelle Beschleunigung Richtung Boden, gemessen an der Norm der drei Accelerometerachsen($|a| = \sqrt{a_x^2 + a_y^2 + a_z^2}$). Ausgehend von der Erdbeschleunigungskonstanten $g \approx 9,81 m/s^2$ bedeutet das eine Beschleunigung von deutlich weniger als ein g. Phase 2 ist der Aufprall mit einer kurzen, starken Beschleunigung. Für Phase 3 wird über einen Zeitraum von 60 Sekunden die Aktivität gemessen. Verbleibt sie über mindestens 40 Sekunden unter einem Schwellwert, wird ein Alarm ausgelöst. Dieser kann von der Nutzerin manuell abgebrochen werden. Wird der Alarm nicht innerhalb einer gewissen Zeitspanne abgebrochen, wird ein Signal an die Basisstation gesendet, von der wiederum ein Notruf abgesetzt werden kann.

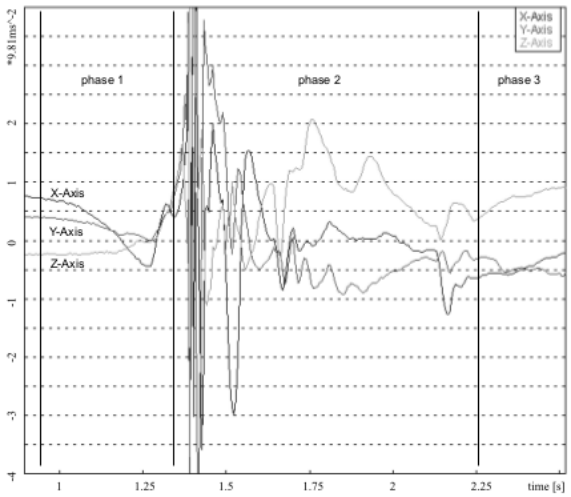

Abbildung 2: Plot eines von SPEEDY gemessenen repräsentativen Sturzes
Quelle: [Degen et al., 2003]

Um False Positives zu minimieren, werden neben der Norm des Beschleunigungsvektors auch zwei weitere Formeln verwendet, die auf unterschiedliche Weise über die einzelnen Beschleunigungsachsen integrieren und laut Aussage der Autoren auf verschiedene Stürze spezialisiert sind. Gemeinsam sollen sie sicherstellen, dass alle Stürze abgedeckt sind.([Degen et al., 2003])
Ein weiteres Beispiel für eine am Handgelenk angebrachte Lösung mit Basisstation ist das Tunstall Vibby([Tunstall, 2016]), das in einer so genannten „Ambient Assisted Living“(AAL) Umgebung verwendet wird. Diese beinhaltet verschiedene Sensoren, Alarmknöpfe und Konzepte, um eine Wohnung altersgerecht zu gestalten.

2.3.2 F2D

Mit dem vermehrten Aufkommen von Smartwatches wird es immer attraktiver, diese als Basis für eine handgelenkbasierte Sturzerkennung zu verwenden. Basierend auf einer am Markt erhältlichen auf Android basierenden Smartwatch mit integriertem GSM-Modem wurde an der University of Geneva eine ebenfalls schwellwertbasierte Sturzerkennung als App entwickelt. Im Gegensatz zum obengenannten SPEEDY Projekt ermöglicht das integrierte GSM Modem der Smartwatch den Aufbau einer Verbindung mit Pflegepersonal oder Angehörigen ohne den Umweg über eine Basisstation oder ein Smartphone.
Auch hier werden selbst erstellte Muster verwendet um Stürze von „Activities of Daily Living"(ADL) zu unterscheiden. Die Parameter der Stürze wie Schwellwerte und Timeouts werden aber im Rahmen von Nutzerprofilen definiert. Wie diese Profile aussehen und wie sie generiert werden, wird nicht erläutert.
Der Sturz wird auch hier anhand der Norm der drei Accelerometerachsen berechnet. Wird eine starke Beschleunigung über einem oberen Schwellwert gefolgt von einer Beschleunigung über einem unteren Schwellwert gemessen, so wird ein Sturz vermutet. Die Zeitspanne zwischen beiden Ausschlägen ist flexibel, übersteigt aber 0.5 Sekunden nicht.
Wurde ein Sturz erkannt, wird wie auch bei SPEEDY der Grad der Aktivität untersucht.
Im Gegensatz zu SPEEDY wird aber zwischen drei Graden unterschieden:
keine Aktivität: Alarm wird sofort gesendet
wenig Aktivität: möglicher Sturz wird signalisiert, Nutzer kann das Senden abbrechen.
normale Aktivität: Hierzu wird im Text keine Aussage gemacht, allerdings ist zu vermuten, dass der Sturz als nicht kritisch eingestuft wird und die Nutzerin bei Bedarf manuell Hilfe anfordern kann.
Die App beinhaltet außerdem noch eine Ortungsmöglichkeit innerhalb von Gebäuden mithilfe eines Bluetooth Beacons.
[Kostopoulos et al., 2016]

2.3.3 Machinelearning-basierter Ansatz

Ein Beispiel für die Sturzerkennung mit einem Machinelearning Ansatz auf einem Wearable Gerät wird in der Arbeit „Detecting Falls with Wearable Sensors using Machine Learning Techniques"[Özdemir and Barshan, 2014] aufgezeigt.
Als Sensor dient hier ein 9-Achsiger Bewegungssensor bestehend aus jeweils 3-Achsigem Accelerometer zur Beschleunigungsmessung, Gyroskop zur Messung von Lageveränderungen und Magnetometer zur Messung der Ausrichtung am Erdmagnetfeld.

Mehrere Sensoren wurden dabei an Kopf, Brust, Hüfte, rechtem Handgelenk, rechtem Oberschenkel und und rechtem Knöchel der Testpersonen fixiert. Die Daten wurden über den Zeitraum des Experimentes kontinuierlich per Funk an eine Basisstation übertragen und im Anschluss ausgewertet. [Özdemir and Barshan, 2014]

3 Problemstellung und Lösungsansatz

3.1 Problemstellung

Es soll ein Algorithmus für ein bestehendes am Handgelenk getragenes Gerät entwickelt und implementiert werden, der mithilfe der bestehenden Sensorik eine grobe Sturzerkennung zur Vorauswahl durchführt. Anschließend sollen relevante Daten von den unterschiedlichen Sensoren des Gerätes zusammengeführt und an einen Server gesendet werden. Durch die Ausführung des Algorithmus darf die bestehende Funktionalität des Gerätes nicht beeinträchtigt werden. Außerdem müssen genügend mögliche Stürze an den Server gesendet werden, damit auch ungewöhnlichere Stürze erkannt werden können. Gleichzeitig ist aber die Datenübertragung an den Server sehr energieaufwändig. Daher muss eine Vorauswahl getroffen werden, um die Anzahl der Verbindungen einzuschränken.

3.2 Lösungsansatz

Die meisten Ansätze, die auf der Verwendung von Bewegungssensoren basieren, beschränken sich entweder auf die Verwendung des Accelerometers zur Beschleunigungsmessung oder integrieren noch ein Gyroskop zur Bestimmung von Lageveränderungen(Vgl. [Degen et al., 2003], [Özdemir and Barshan, 2014]). Wenn der Sensor am Oberkörper und damit nahe am Körperschwerpunkt der Nutzerin angebracht ist, sind diese Werte gute Indikatoren für einen Sturz(Vgl. [Bourke et al., 2007], [Li et al., 2009]). Wurde eine starke Beschleunigung mit gleichzeitiger Lageänderung gemessen, dann kann schon mit hoher Wahrscheinlichkeit von einem Sturz ausgegangen werden.
Am Handgelenk getragen verlieren diese Werte allerdings viel ihrer Aussagekraft, da sich der Arm sehr unabhängig vom Körper bewegen kann. Während eine Lageveränderung des aufrechten Oberkörpers um 90 Grad eine Positionsveränderung zum Liegen bedeutet, ist das am Arm deutlich weniger ersichtlich. Ebenso kann eine starke Beschleunigung des Handgelenks auch die Folge eines Handschlags oder einer sonstigen plötzlichen Bewegung des Arms sein.
Ein großer Vorteil der Position am Handgelenk ist allerdings die Möglichkeit, Vitaldaten wie den Puls oder die Körpertemperatur leicht und schnell messen und in die Sturzerkennung mit einfließen lassen zu können. Außerdem kann einfacher der Knopf am Gerät selbst erreicht werden. Dieser kann also zur Interaktion mit dem Nutzer in den Algorithmus eingebunden werden.
Der hier verfolgte Ansatz besteht darin, einen einfachen Algorithmus zu entwickeln, der die Beschleunigung und Lage des Gerätes misst und bei Überschreiten eines Schwellwertes diese Daten zusammen mit dem Puls und weiteren Vitaldaten vorbereitet und sendet.
Da zu diesem Zeitpunkt noch nicht feststeht, welche Daten vom Server benötigt werden, um die weitere Sturzerkennung durchzuführen, sollen so de-

taillierte Rohdaten wie möglich gesendet werden. Sobald das Design des serverseitigen Algorithmus feststeht kann die Menge der gesendeten Daten und deren Vorverarbeitung darauf angepasst werden.

4 Die SmarKo Plattform

Das Armband, auf dem die grobe Sturzerkennung implementiert werden soll, ist Teil der von MCS DataLabs entwickelten SmarKo Plattform[MCS DataLabs,]. Da es die physische Kernkomponente darstellt werden hier „SmarKo Armband“ und „SmarKo“ häufig synonym benutzt. Wenn in dieser Arbeit von der SmarKo Plattform als Ganzes geredet wird, wird dies auch im Text deutlich gemacht.
Der Inhalt dieser Arbeit ist, wie schon erwähnt, die Implementierung und Integration einer Vorauswahl oder groben Sturzerkennung auf dem Armband selbst. In diesem Teil soll die ursprüngliche Funktionalität und der Aufbau der Plattform im Ganzen und des Armbandes im Speziellen beschrieben werden. Diese wurden nicht im Rahmen dieser Arbeit entwickelt. Die Beschreibung wird hier zum besseren Verständnis geliefert.

4.1 Entstehung und Ziel

Das Armband SmarKo, zum Zeitpunkt seiner Entstehung noch KiKo genannt, begann als Projekt des Fraunhofer Instituts für Offene Kommunikationssysteme[FOKUS,] als Möglichkeit für Eltern, den Aufenthaltsort ihrer Kinder auf dem Schulweg prüfen und bei Bedarf schnell mit ihnen kommunizieren zu können. Insbesondere für junge Kinder, die kein eigenes Telefon besitzen, kann das die einzige Möglichkeit sein, in einem Notfall schnell eine Verbindung zu den Eltern aufzubauen. Eine Voraussetzung für das Gelingen war, dass das Gerät unabhängig von einem Smartphone oder anderem Computer agieren kann und eine Möglichkeit besitzt, den eigenen Standort zu bestimmen. Außerdem sollte es einfache Vitaldaten messen und senden können und diese auch auf Anfrage senden. Schließlich musste noch die Möglichkeit für die Trägerin bestehen, manuell einen Notruf auszulösen.
Auf Serverseite sollte die Möglichkeit bestehen, sowohl als Privatperson einen eigenen Server für ein oder zwei Geräte selbst zu betreiben, als auch als Institution eine größere Anzahl an Geräten gleichzeitig zu betreuen. In diesem Kontext könnte das zum Beispiel eine Schule sein, die diesen Service anbietet. Gleichzeitig musste aber auch ein hohes Maß an Datenschutz durch Verschlüsselung der Daten gewährleistet sein.
Diese Funktionen bieten sich aber auch für eine Vielzahl anderer Einsatzmöglichkeiten an. Insbesondere in der Altenpflege ist die konstante Überwachung von Vitaldaten und die Unabhängigkeit von einem Smartphone oder einer Basisstation von großem Wert.
Die einfache Bedienbarkeit des Gerätes kombiniert mit der Ähnlichkeit zu einer normalen Armbanduhr macht SmarKo wesentlich attraktiver für Menschen, die sich wenig mit Technik und Technologie auseinander setzen wollen. Da das Gerät konstant getragen werden muss, um seine Aufgabe zu erfüllen ist das ein großer Vorteil gegenüber anderen ähnlichen Geräten, die zum

Beispiel an einem Band um den Oberkörper oder an einem Gürtelclip getragen werden. Auch die Verwendung für demenzerkrankte Patientinnen sollte durch die vertraute Form stark vereinfacht werden
Geolocation und Geofencing ist ebenfalls eine Funktionalität, die insbesondere für Demenzpatientinnen wertvoll ist. „Wandering“ ist ein häufiges Phänomen von Demenzerkrankungen, bei dem Betroffene plötzlich ihren Aufenthaltsort verlassen und ziellos umherirren([Warren et al., 1999]). In der Folge können sie sich schnell verletzen oder verirren. Für pflegende Angehörige ist dieses Verhalten oft frustrierend und zeitaufwändig. Eine Alarmierung der Trägerin sowie einer pflegenden Person bei Verlassen eines designierten Bereiches kann viele dieser Probleme verringern.

4.2 Überblick

Während KiKo noch einen relativ eingeschränkten Leistungsumfang und eine sehr spezifische Aufgabe besaß, soll mit SmarKo eine Plattform geschaffen werden, die schnell und leicht auf unterschiedliche Anforderungen anpassbar ist. Die SmarKo Plattform beinhaltet dabei neben Armband und Server auch mehrere mobile Apps, die die Kommunikation zwischen Pflegebedürftigen, Angehörigen und Ärztinnen erleichtern sollen. Der Kern der SmarKo Plattform besteht aus dem Armband und dem zugehörigen Server. Daten werden auf dem Armband gesammelt und an den Server gesendet, wo sie verarbeitet und an registrierte Mobilgeräte weitergesendet werden.

4.3 Armband

Das SmarKo Armband besitzt ein GSM Modem und kann darüber unabhängig von einem Smartphone oder einer Basisstation Daten an den Server senden. Diese Daten umfassen den Standort, den Puls und den Blutsauerstoffgehalt der Nutzerin und werden in regelmäßigen, vom Server festgesetzten Zeitintervallen in einen JSON String verpackt und gesendet. Außerdem ist ein Update der Firmware über die mobile Datenverbindung möglich. Bei Bedarf können Nutzerinnen durch zweimalige Betätigung des seitlich angebrachten Knopfes einen Notruf absetzen. Dieser wird vom Server registriert und an Angehörige und/oder Pflegepersonal weitergeleitet. SmarKo ist dabei auf einfachste Bedienung ausgelegt, um seine Benutzung sowohl durch ältere Menschen als auch durch Kinder zu ermöglichen.

4.4 Server

Der Server kann entweder von einer Privatperson oder einer größeren Institution wie einem Krankenhaus oder einer anderen Pflegeeinrichtung betrieben werden. Das bedeutet, Nutzerinnen von SmarKo haben im Gegensatz zu vielen anderen auf dem Markt erhältlichen Smart Bracelets die Datenhoheit über ihre eigenen Gesundheitsdaten.

Das Webinterface des Servers ist aufgeteilt in ein Administrations- und ein Userinterface.
Im Administrationsinterface können Geräte und Trägerinnen hinzugefügt und einander zugeordnet werden. Außerdem können die aktuellsten Datensätze mehrerer Patientinnen gleichzeitig eingesehen werden. Dieses Interface ist insbesondere in größeren Einrichtungen relevant, die viele SmarKo Geräte gleichzeitig betreuen.
Das Userinterface bietet Zugriff auf die Daten und Einstellmöglichkeiten der diesem Account zugeordneten Geräte.
Die Trägerin selbst oder wahrscheinlicher eine Vertrauensperson, im folgenden Nutzerin genannt, kann dann auf die Positionsdaten der zugeordneten Geräte zugreifen, sich die verschiedenen gesammelten Daten anzeigen lassen und über eine SMS ein Senden aktueller Daten von einzelnen Geräten auslösen. Außerdem besteht die Möglichkeit des Geofencing. Die Nutzerin kann dabei für jedes Gerät einen Bereich auf einer Karte festlegen. Wird das Gerät aus diesem Bereich entfernt, wird ein Alarm ausgelöst.
Weiterhin werden die Nutzerinnen über das Interface und per SMS über jeden Alarm informiert inklusive einer Anzeige, um was für einen Alarm es sich handelt, sei er manuell, durch einen Sturz oder durch eine andere Ursache ausgelöst.
Im Falle einer größeren Einrichtung würde das Administrationsinterface also zentral verwaltet, während Angehörige und Pflegepersonal über das Userinterface direkten Zugriff auf die ihren Accounts zugeordneten Geräte haben.

4.5 Hardware des Armbands

Das Bracelet basiert auf einem ARM Cortex M3 Mikroprozessor von Silicon Labs, der mit 24MHz getaktet ist. An Komponenten existieren auf dem Gerät ein GSM und ein GPS Modul, die beide über die UART Schnittstelle angeschlossen sind. Weiterhin sind über die I2C Schnittstelle ein 9-Achsen Bewegungssensor sowie ein kombinierter Blutsauerstoff- und Pulsmesser angeschlossen. Als Stromquelle dient ein integrierter, nicht auswechselbarer Akku. Dieser ist mit einer Spule verbunden, die es ermöglicht, das Gerät kabellos mit entsprechenden Ladegeräten zu laden. Die drahtlose Ladetechnologie entspricht dem qi-Standard[Wireless Power Consortium, 2017]. Eingefasst ist das Gerät dabei von einer wasserdicht verschlossenen Hülle die mit einem separaten Armband oder Clip kombiniert werden kann. Abb. 3 zeigt die einzelnen Komponenten und ihre Verbindungen zum Microcontroller.

4.5.1 Eingabe

Ein Knopf an der Seite ist das einzige Eingabegerät und kann verwendet werden, um mit einem zweimaligen Druck einen Notruf abzusetzen oder bei langem Druck(mehr als 10 Sekunden) das Gerät neuzustarten. Wird mit dem

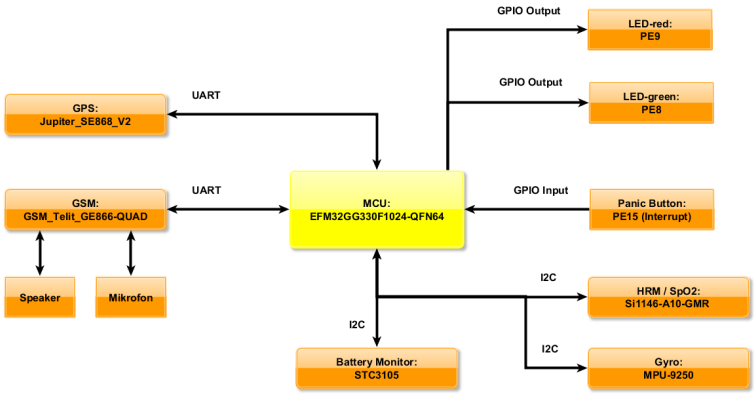

Abbildung 3: Komponenten SmarKo
Quelle: Fraunhofer Institut für Offene Kommunikationssysteme FOKUS

Knopf ein Notruf ausgelöst, wechselt das Gerät in den SOS Modus. Dessen Parameter können unabhängig vom normalen Modus gesetzt werden um z.B. die Sendefrequenz zu erhöhen oder den Timeout für einen eine genaue GPS Positionsermittlung zu erhöhen oder verringern.

4.5.2 Ausgabe

Eine einfache Ausgabe erfolgt über eine rote und eine grüne LED, die kontextabhängig auf unterschiedliche Arten blinken oder leuchten. Sämtliche relevanten Daten werden an den Server gesendet. Dieser bildet das primäre Interface mit dem Bracelet und ermöglicht weitreichende Einstellungen und Anpassungen auf den aktuellen Verwendungszweck des Gerätes durch das Webinterface. Außerdem stellt er die Basis für die Erweiterung der ganzen Plattform um weitere Geräte oder Apps dar. Nutzerinnen und deren Angehörige oder Pflegekräfte können die Daten vom Server über das Webinterface sowie über eine spezielle App beziehen. In Notfällen können sie außerdem per SMS kontaktiert werden.

5 Algorithmus zur Sturzvorauswahl

Der ursprüngliche Plan, eine Sturzerkennung zu implementieren, die komplett auf dem Gerät ausgeführt wird, hat sich im Kontext der anderen Funktionen des Gerätes als unpraktikabel erwiesen. Daher wurde eine Aufteilung des Algorithmus auf Server und Armband angestrebt. Eine Sturzbestimmung ausschließlich auf dem Gerät selbst wäre zwar möglich, die Notwendigkeit, schnell auf andere Ereignisse reagieren zu können und die begrenzte Batteriekapazität machen es allerdings sinnvoll, die Berechnung selbst auf einen externen Server auszulagern. Außerdem wird so die Möglichkeit geschaffen, die gesammelten Daten mit einem Machine Learning Ansatz auszuwerten, der unter Umständen mehr Rechenleistung benötigt, als auf dem Armband zur Verfügung steht.

5.1 Aufteilung des Algorithmus

Diese Aufteilung sieht vor, dass auf dem Gerät selbst eine Vorauswahl getroffen wird, die grob genug ist, um sämtliche Stürze zu beinhalten, dabei aber dennoch die Menge der zu analysierenden Fälle stark verringert. Wurde ein potentieller Sturz erkannt, werden gesammelte Bewegungsdaten zusammen mit Vitaldaten und Position an den Server übertragen, wo sie weitergehend analysiert werden können. Liegt tatsächlich ein Sturz vor, kann der Server weitere Schritte ergreifen, wie z.B. einen Notruf absetzen, Pflegepersonal oder Familienangehörige informieren oder eine Sprachverbindung zur Patientin aufbauen.

5.2 Server

Der Server wird vom SmarKo Armband über eine SSL verschlüsselte Datenverbindung kontaktiert und erhält so die Geoposition, Vitaldaten und einige Statusinformationen. Im Rahmen dieses Kontakts kann der Server auch veränderte Werte und Steuerdaten an das SmarKo Armband übertragen. Das sind insbesondere Zeitintervalle für die Funktionen des Armbands wie das Sendeintervall, die Dauer des SOS Modus und die Zeit, bis der Versuch die Position per GPS festzustellen abgebrochen wird.

5.3 Entstehung des Client-Algorithmus

Trotzdem die genauere Sturzerkennung und die Verarbeitung der gesammelten Daten vor allem auf dem Server passieren, muss das Gerät selbst gleichzeitig verschiedene Aufgaben erfüllen. Ursprünglich bestanden diese nur im Senden in fixen Zeitabständen sowie in der Behandlung eines durch den am Gerät angebrachten Knopf ausgelösten manuellen Alarms. Die Datenermittlung konnte zum Zeitpunkt des Sendens geschehen und damit in den

Sendevorgang integriert werden. Dazu kommen jetzt zusätzlich die regelmäßige Messung der Bewegungsdaten sowie die grobe Sturzerkennung selbst. Beides muss parallel zu den anderen Funktionen möglich sein.
Hier soll der Entwicklungsprozess des Algorithmus dargestellt werden, um die Wichtigkeit seiner einzelnen Komponenten zu verdeutlichen.

5.3.1 Einfacher Bewegungsinterrupt

Beschreibung Die einfachste und erste implementierte Version des Algorithmus verwendet die Fähigkeit des Bewegungssensors, beim Überschreiten eines festgelegten Schwellwertes auf einer der drei Achsen einen Interrupt auszulösen. Hierbei werden die Beschleunigungswerte der drei Achsen bereits vom Sensor selbst hochpassgefiltert um die Erdbeschleunigung zu eliminieren und anschließend mit einem Schwellwert verglichen, der in ein spezielles Register des Sensors geschrieben wurde. Übertritt eine der drei Achsen diesen Wert, wird der Interrupt ausgelöst.

Probleme Diese Funktion des Bewegungssensor ist für die generelle Erkennung von Bewegung gedacht. Der sensorinterne Schwellwert kann daher zwar relativ fein eingestellt werden, der Maximalwert liegt aber nur bei 1024mg. Als Grundlage für eine Sturzerkennung ist das um einiges zu wenig, da schon eine etwas stärkere Bewegung des Handgelenks ausreicht, um den Interrupt auszulösen. Es werden so erwartungsgemäß zwar nahezu alle Stürze erkannt, allerdings macht die Menge an False Positives und damit ein extrem hohes Datenaufkommen diese Implementierung unpraktikabel. Es wird eine Möglichkeit benötigt, diese einzugrenzen.

5.3.2 Manuelle Unterbrechung des Sendevorgangs

Beschreibung Die erste Erweiterung des Algorithmus besteht darin, eine Möglichkeit für den Nutzer zu bieten, das Senden selbst abzubrechen. Das SmarKo Armband verfügt über einen Knopf, der sowohl zum manuellen Auslösen eines Alarms, als auch zum Neustart des Gerätes verwendet wird. Außerdem existieren zur Ausgabe eine grüne und eine rote LED. Wird durch den vorgenannten Interrupt ein Sturz signalisiert, dann fängt die rote LED an, gleichmäßig zu blinken. Wenn die Nutzerin innerhalb von einer festen Zeitspanne den Knopf 2 Sekunden lang betätigt, wird der Sendevorgang abgebrochen.
Sonst wird er wie bisher fortgesetzt.

Problem Wir haben nun zwar die Möglichkeit, eine große Anzahl an Fehlsendungen zu unterbinden, allerdings muss das durch die Nutzerin geschehen. Bemerkt diese das Blinken der LED nicht, wird trotzdem ein Sturzalarm gesendet. Außerdem hängt die Akzeptanz eines solchen Gerätes stark mit

der Benutzbarkeit zusammen. Wenn eine Nutzerin immer wieder Sendevorgänge abbrechen muss, wird sie das Armband irgendwann nicht mehr tragen wollen.

5.3.3 Softwareprüfung der Beschleunigungswerte

Beschreibung Um die automatische Sturzerkennung zu verbessern wird zwar weiterhin der Interrupt als Grundlage verwendet. Statt aber in der Behandlung des Interrupts nur zu signalisieren, dass ein Sturz vorliegt, werden jetzt die Beschleunigungsdaten des Accelerometers ausgelesen und die Norm der drei Achsen a_x, a_y und a_z berechnet: $|a| = \sqrt{a_x^2 + a_y^2 + a_z^2}$. Jetzt kann in Software ein hardwareunabhängiger Schwellwert festgesetzt und mit der berechneten Norm verglichen werden. Da die ausgelesenen Rohdaten des Accelerometers nicht hochpassgefiltert sind, ist in ihr die Erdbeschleunigung enthalten. Wir ziehen daher die Konstante für die Erdbeschleunigung g von der Norm ab. Der resultierende Wert soll hier $n = |a| - g$ heißen. Bei einer Bewegung in Richtung des Erdmittelpunkt ohne horizontale Bewegung, also einem freien Fall, können dabei Ergebnisse mit $n < 0$ entstehen. Hier soll aber der Aufprall und die daraus resultierende starke Beschleunigung als Indikator für einen Sturz dienen. Damit können negative Werte ignoriert werden.

Problem Es besteht jetzt die Möglichkeit, erst bei einer beliebig gewählten Beschleunigung die Sturzbehandlung auszulösen. Danach hat die Nutzerin noch die Möglichkeit, das Senden eines Alarms abzubrechen. Allerdings gibt es einige Bewegungen, die ähnliche starke Beschleunigungen auslösen wie ein Aufprall, insbesondere, wenn die Messung am Handgelenk statt findet. Ein enthusiastisches Winken oder etwas stärkeres Aufsetzen mit der Hand z.B. auf dem Tisch würde immer noch als Sturz gewertet. Es reicht also nicht, nur die Beschleunigungswerte zum Zeitpunkt des Interrupts selbst zu betrachten.

5.3.4 Messung von mehreren Datensätzen

Beschreibung Der Algorithmus wird so angepasst, dass mit dem Interrupt mehrere Messungen in festen Abständen angestoßen werden. Mit diesen Daten können auch die Bewegungen nach dem Sturz überprüft werden. So kann ein Alarm erst dann gesendet werden, wenn sich die Trägerin über einen festgelegten Zeitraum nur wenig oder nicht bewegt hat.

Problem Wurde der Interrupt durch eine starke Bewegung der Hand z.B. durch eine Winkbewegung ausgelöst, dann kann es passieren, dass die Hand in der Folge nicht viel bewegt wird und wiederum fälschlicherweise ein Sturz

angenommen wird. Es werden also noch mehr Indikatoren für einen Sturz benötigt.

5.3.5 Datensammlung in einem Ringpuffer

Beschreibung Es wird ein Timerinterrupt verwendet um die Daten aus dem Bewegungssensor in einen Ringpuffer zu schreiben. Dieser wird in festen Zeitabständen ausgelöst. Wird der Bewegungsinterrupt ausgelöst, dann können noch über eine kurze Zeitspanne weiter Daten gesammelt werden um so eine Menge von Messungen zu generieren in deren Mitte sich der Sturz befindet.

Ringpuffer Ein Ringpuffer ist eine Datenstruktur die sich insbesondere dadurch auszeichnet, dass beliebig viele Daten hinein geschrieben werden können. Ist der zugewiesene Speicher voll, werden bestehende Daten beginnend mit den ältesten überschrieben. Da in diesem Fall ein fest definiertes Fenster von Messungen um den Sturz herum benötigt wird, bietet sich diese Datenstruktur hier besonders an.

Problem Das Auslösen der Messung geschieht immer noch im oben genannten Bewegungsinterrupt und ist dementsprechend durch die einmalige Messung eingeschränkt.

5.3.6 Analyse der Bewegungsdaten statt Motioninterrupt

Beschreibung Statt den Bewegungsinterrupt zum Auslösen einer Messung zu nutzen, kann eine komplexere Überprüfung der Daten in der selben timerbasierten Interruptverarbeitung passieren, in der die Daten gesammelt werden. So kann in Echtzeit überprüft werden, ob ein freier Fall, ein Aufschlag oder wenig Bewegung vorliegen. Da die Daten letztendlich auf dem Server final verarbeitet werden sollen, müssen immer noch Rohdaten an den Server gesendet werden. Außerdem darf die Sturzerkennung auf dem Armband nicht so restriktiv sein, dass Stürze, die nicht genau in das verwendete Muster passen, nicht an den Server weiter geleitet werden. Jetzt können die unterschiedlichen Parameter wie Zeitrahmen der Messung, Anzahl der Messungen in diesem Zeitrahmen, Einteilung des Sturzes in Phasen etc. flexibel gesetzt und mit Hilfe von Experimenten die besten Werte dafür gefunden werden.

6 Implementierung

In diesem Teil soll die konkrete Implementierung des im Kapitel *Algorithmus zur Sturzvorauswahl* beschriebenen Algorithmus auf dem SmarKo Armband beschrieben werden.
Die Implementierung wurde in der Programmiersprache C vorgenommen. Dabei wurde die vom Hersteller des verbauten Microcontrollers, Silicon Labs, zur Verfügung gestellte Software Suite „Simplicity Studio“ verwendet. Diese beinhaltet neben einer Entwicklungsumgebung(IDE) auch Funktionalität zum Aufspielen der Firmware auf den Microcontroller und zur Fehlersuche.

6.1 Anforderungen an die Implementierung

Hier sollen kurz einige Anforderungen herausgearbeitet werden, die bei der Implementierung eine Rolle gespielt haben.

Rückwärtskompatibilität Da der ursprüngliche Verwendungszweck des Gerätes ein anderer war, wurde die Firmware auch mit anderem Fokus geschrieben. Oft konnten in der hier beschriebenen Implementierung neue Funktionen nicht einfach hinzugefügt werden, ohne alte Funktionalität in ihrer Implementierung stark anzupassen und zu verändern.

effiziente Speichernutzung Das Gerät verfügt aktuell über einen Festspeicher von einem Megabyte. Da dieser nicht allein der Sturzerkennung zur Verfügung steht, muss eine Balance gefunden werden zwischen der Menge der gespeicherten Daten und der Häufigkeit des Sendevorgangs. Im Hinblick auf zukünftige Funktionserweiterungen und den daraus resultierenden Speicherbedarf muss eine Möglichkeit geschaffen werden, die Menge an gespeicherten Daten einfach an die Anforderungen und den verwendbaren Speicher anzupassen. Um das gewährleisten zu können wurde eine Steueroption eingebaut, um sowohl die zeitliche Länge des Messfensters, als auch die Anzahl der Messungen an einer zentralen Stelle zu setzen.

Notfallplan bei gescheitertem Verbindungsaufbau Der Informationsaustausch kommt nur zustande, wenn das Gerät den Server kontaktiert. Insbesondere kann der Server nur auf diesem Wege mit dem Gerät kommunizieren. Es muss also einen Notfallplan geben, für den Fall, dass eine Verbindung nicht zum erst möglichen Zeitpunkt zu Stande kommt. Als Lösung für dieses Problem wird geprüft, ob eine Verbindung zustande gekommen ist. Erst nach der Bestätigung durch den Server werden die relevanten Messungen nicht mehr gesendet und der normale Betrieb wird wieder aufgenommen.

6.2 Mainloop

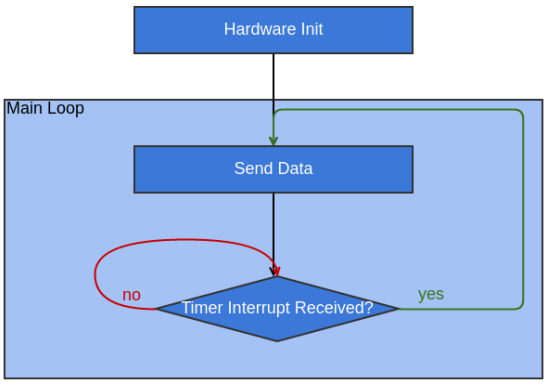

Abbildung 4: ursprünglicher Main Loop

In der ursprünglichen Version des Gerätes vor der Sturzerkennung wurde nur die Datensendefunktion in fixen Abständen ausgeführt. Die Möglichkeit, manuell einen Notruf auszulösen existierte allerdings schon. Der Knopf löste einen Interrupt aus, der das Gerät in den SOS Modus versetzte und die Sendefrequenz erhöhte. Dem Server wurde dann beim nächsten Sendevorgang der SOS Zustand mitgeteilt, sodass darauf reagiert werden konnte. Der ursprüngliche Ablauf ist in Abb. 4 zu sehen.
Die Implementierung der Sturzerkennung machte es nötig, zwischen verschiedenen SOS Modi zu unterscheiden und komplexere Funktionalität beim Wechsel in den SOS Modus zu ermöglichen. Wie in Abbildung 5 zu erkennen, wurde der ursprüngliche Main Loop stark erweitert.
SmarKo hat am Gerät selbst bis auf den Knopf sehr wenige Eingabemöglichkeiten. Deshalb wurde von Anfang an ein großer Teil der Konfiguration auf den Server ausgelagert. Die vom Server auslösbaren Funktionen beinhalten unter anderem das Auslösen eines Reset, das Abbrechen des SOS Modus und ein Over-the-Air Update der Firmware. Allerdings gab es wenige Möglichkeiten, den Erfolg oder Misserfolg dieser Befehle zu überprüfen.
Eine Bemühung im Rahmen dieser Arbeit war es, die Expressivität des Gerätes gegenüber dem Server zu erhöhen. Darum wurden mehrere Flags integriert, die dem Server mitteilen, ob das Gerät neu gestartet wurde, ob ein angesetztes Softwareupdate erfolgreich oder erfolglos durchgeführt wurde und ob der SOS Modus abgebrochen wurde. Diese Flags werden im ersten Schritt des neuen Main Loops gesetzt und die Informationen für das Senden an den Server vorbereitet.
Im Anschluss wird überprüft, ob durch einen der Interrupts eine der Flags gesetzt wurde, die die Behandlung des jeweiligen Interrupts signalisieren.
Diese Behandlung wurde nach dem Bottom-Half Designpattern(vgl. [Rubini

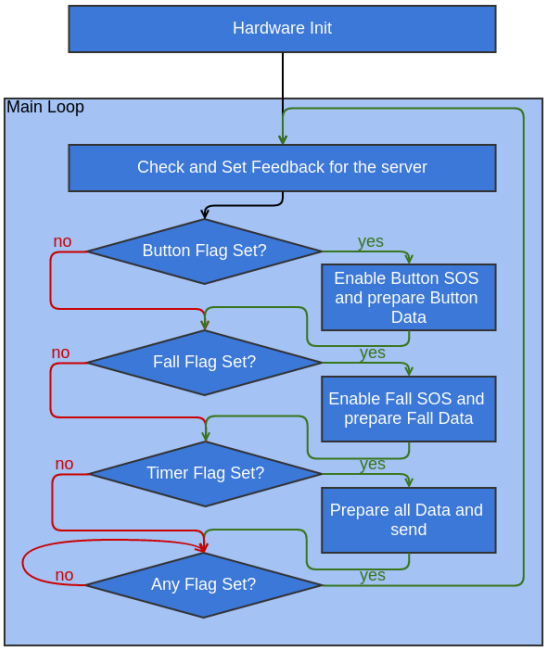

Abbildung 5: angepasster Main Loop

and Corbet, 2001], S. 275) aufgebaut. Dabei wird soviel wie möglich der Funktionalität aus der Interrupt Service Routine(ISR) ausgelagert um einen schnellen Rücksprung in den Main Loop und aus der ISR zu gewährleisten. Die ISR möglichst kurz zu halten ist wünschenswert, da unter Umständen während einer laufenden Interruptbehandlung weitere Interrupts ignoriert werden.
Im Anschluss soll auf die einzelnen Komponenten eingegangen werden.

6.3 SOS Button Interrupt

Die Flag für den manuellen Notruf wird gesetzt, wenn der seitlich am Gerät befestigte Knopf zweimal in kurzer Abfolge gedrückt wird. Ist das passiert, dann wird, wie in Abb. 6 zu sehen, das SmarKo Armband in den Notrufmodus versetzt und als Grund dafür ”Button”gespeichert. Im Notrufmodus wird dann wie in der alten Version der Firmware die Sendefrequenz erhöht und dem Server mitgeteilt, dass ein Notfall vorliegt.

6.4 Sturz Interrupt

Obgleich ein Großteil der hier beschriebenen Implementierungsarbeit darin bestand, die bestehende Software an die veränderten Anforderungen anzupassen, stellt dieser Teil die Kernfunktionalität der groben Sturzerkennung

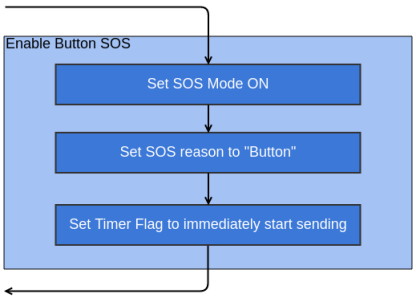

Abbildung 6: SOS Button Behandlung

dar.

6.4.1 Erkennung eines möglichen Sturzes

Die Behandlung eines möglichen Sturzes ist um einiges komplexer als die des manuellen Alarms. Aufgrund von Schwierigkeiten mit dem Senden größerer Datenmengen entspricht die aktuelle Implementierung dem Abschnitt *Datensammlung in einem Ringpuffer* aus dem Kapitel *Algorithmus zur Sturzvorauswahl*.
Bewegungsdaten werden innerhalb eines Timerinterrupts in festen Abständen in einen Ringpuffer geschrieben und können bei Bedarf ausgelesen werden. Ein Sturz wird aber unabhängig davon über den Bewegungsinterrupt signalisiert. Dabei wird die Norm der ermittelten Beschleunigung mit einem Schwellwert verglichen. Ist die Norm größer als der Schwellwert, wird die Sturzflag gesetzt. Außerdem wird eine Markierung gesetzt, sodass bei der nächsten Messung für den Ringpuffer markiert werden kann, dass diese nahe am Sturz war. Bei der gegenwärtig groben Auflösung aufgrund des Sendeproblems ist das nur eingeschränkt sinnvoll. Ist die Auflösung allerdings hoch genug, dann ist die markierte diejenige Messung, welche zum Zeitpunkt des Sturzes durchgeführt wurde. Wenn der Bewegungsinterrupt wegfällt, dann wird diese Markierung direkt mit der Messung ausgeführt und stellt nur noch eine redundante Möglichkeit für den Server da, den auslösenden Zeitpunkt festzustellen.

6.4.2 Behandlung eines möglichen Sturzes

Wurde ein möglicher Sturz registriert, so wird dies der Trägerin des Armbands durch das Blinken der roten LED signalisiert. Zu diesem Zeitpunkt kann das Senden eines Alarms an den Server durch das zweisekündige Betätigen des Knopfes abgebrochen werden. Es wurde bewusst eine andere

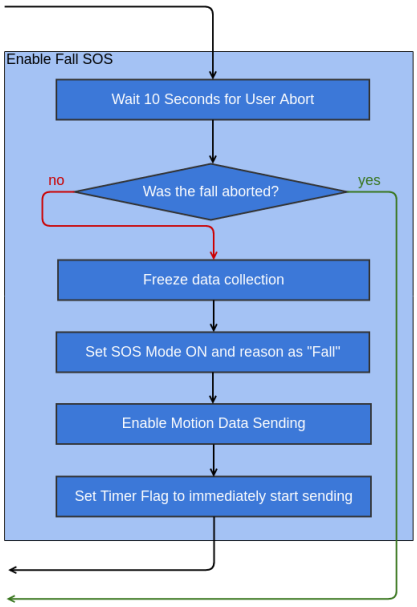

Abbildung 7: Fall Interrupt Behandlung

Bedienung des Knopfes als für den manuellen Alarm gewählt, um Verwechslungen zu vermeiden. Das Senden eines manuellen Alarms ist währenddessen nicht möglich, um zu verhindern, dass der Abbruch des einen Alarms zum Auslösen des anderen führt. Da das zweifache Drücken des Alarmknopfes(manuelles SOS Signal senden) nicht zu einem Abbruch führt, wird in diesem Fall einfach das Sturz SOS gesendet.
Wurde der Sturzalarm nicht manuell abgebrochen, dann wird das Sammeln der Bewegungsdaten nach 10 Sekunden eingefroren. Damit haben wir einen Datensatz von aktuell 10 Sekunden nach dem Sturz plus eine kurze Zeit vor dem Sturz. Beide Zeitrahmen können eingestellt werden, um einen optimalen Datensatz zu generieren. Es muss noch ermittelt werden, welche Messfrequenz und welcher Messzeitraum für die für die weitere Verwendung optimal sind.
Weiterhin wird als Grund für den ausgelösten Alarm ein möglicher Sturz eingetragen und das Senden der Bewegungsdaten eingeschaltet. Um Strom und Datenvolumen zu sparen, werden im normalen Betrieb die Bewegungsdaten nur gesendet, wenn tatsächlich ein potentieller Sturz vorliegt.
Schließlich wird wie beim manuellen Notruf die Timerflag gesetzt, um direkt den Sendevorgang anzustoßen.

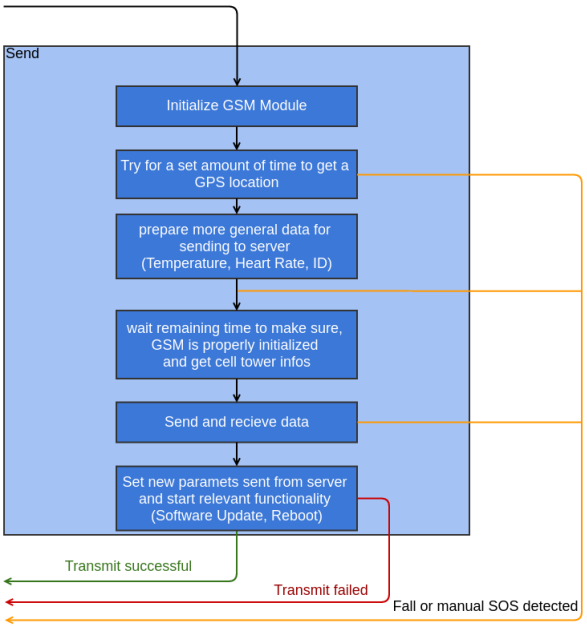

Abbildung 8: Sendevorgang

6.5 Sendevorgang

Die Ausführung des kompletten Sendevorgangs kann mit Initialisierungen, dem Sammeln unterschiedlicher Daten und erneuten Versuchen bei fehlgeschlagenen Verbindungen mehrere Minuten dauern. Um zu gewährleisten, dass Stürze und manuell ausgelöste Notrufe trotzdem so schnell wie möglich behandelt werden können, wurden an mehreren relevanten Stellen Abbruchbedingungen integriert. Im Abschnitt *Sturzerkennung während des Sendevorgangs* wird das Vorgehen genauer erörtert.
Wie in Abb. 8 zu erkennen, beginnt die Sendefunktionalität mit der Initialisierung des GPS Moduls and der anschließenden Suche nach einer Position durch das GPS. Der Timeout dafür kann flexibel vom Server eingestellt werden, sollte aber größer als 1 Minute sein. Aus diesem Grund existiert an dieser Stelle die erste Möglichkeit zum Abbruch durch einen Sturz oder Notruf. Vor der Initialisierung des GPS werden noch das GSM Modul gestartet und eine Messung des kombinierten Puls- und Blutsauerstoffsensors veranlasst. Da beide Funktionen nahezu unabhängig vom Prozessor laufen, können wir sie parallel zur GPS Positionssuche ausführen.
Im Anschluss an entweder eine erfolgreiche Positionsbestimmung oder einen GPS Timeout werden zuerst die Vitaldaten - Puls und Blutsauerstoff - für das Senden vorbereitet. Sollte die Positionssuche per GPS erfolglos gewesen

sein, werden im Anschluss Daten über die nächsten Mobilfunkmasten aufbereitet, um dem Server zumindest eine grobe Ortsbestimmung zu ermöglichen. Dies muss erfahrungsgemäß mindestens 20 Sekunden nach Initialisierung des GSM Moduls geschehen um fehlerhaften Messungen vorzubeugen. Danach werden die gesammelten Daten in einem JSON String vereint und per SSL verschlüsselt über eine GPRS Verbindung an den Server übertragen. Dieser antwortet mit einem OK und den aktuellen Einstellungen für das sendende Gerät.
Schließlich wird die Antwort des Servers ausgewertet und bei Bedarf der SOS Zustand ausgeschaltet oder ein Neustart oder Firmware Update veranlasst. Daran anschließend wechselt das Armband in einen Leerlaufmodus, aus dem es nur durch das Setzen einer der Flags für den Button-Notruf, einen möglichen Sturz oder das erneute Senden der Daten durch einen Interrupt wieder geweckt wird.

6.6 Schwierigkeiten bei der Implementierung

6.6.1 Ausbau des Mainloops mit Bottom-Half Interruptbehandlung

Da die Funktion des Gerätes ursprünglich hauptsächlich das Senden von gesammelten Daten in fixen Zeitabständen war, reichte es aus, im Mainloop die Daten zu sammeln, zu senden und anschließend zu warten, bis die Zeit wieder abgelaufen ist, um von neuem zu starten. Das manuelle Auslösen des SOS Modus war per Button Interrupt möglich, allerdings wurde direkt in der Interrupt Service Routine(ISR) die Sendezeit verkürzt und ein Flag für den SOS Modus gesetzt. Die Funktionalität im Mainloop war essentiell dieselbe bis auf eine schnellere Sendefrequenz und ein Signal an den Server, dass sich das Gerät im SOS Modus befindet.
Da durch die Sturzerkennung eine weitere Ursache für einen Notruf hinzukam und außerdem die Behandlung des Bewegungsinterrupts weit komplexer ist als die des Buttoninterrupts, musste der Mainloop stark angepasst werden.
Die umfangreiche Interruptverarbeitung direkt in der ISR durchzuführen ist nicht erwünscht, weil so andere Interrupts bis zum Ende der Ausführung blockiert würden. Eine Lösung für dieses Problem ist die Aufteilung der Behandlung in Top-Half und Bottom-Half. Diese Technik wird unter anderem auch im Linux Kernel verwendet([Rubini and Corbet, 2001], S. 275). Die ISR selbst stellt dabei die Top-Half dar. In ihr wird so wenig wie möglich gearbeitet, um andere potentielle Interrupts so kurz wie möglich zu blockieren. Es wurden Flags für die unterschiedlichen Interruptauslöser erstellt, so dass diese in den ISRs nur noch gesetzt werden mussten.
Die Bottom-Half ist die eigentliche Behandlung. Bei dieser Implementierung werden die Flags im Mainloop überprüft und die Behandlung bei Bedarf

ausgeführt. Wurde der Mainloop durchlaufen, wird in einer Fangschleife gewartet, bis wieder ein Flag gesetzt wurde und in diesem Fall wieder an den Start des Mainloops gesprungen.

6.6.2 Interferenzen zwischen unterschiedlichen Sensoren

Dieses Problem trat mehrmals in unterschiedlichen Formen auf.
Da SmarKo kein Betriebssystem besitzt, muss die Aufteilung der Rechenzeit manuell erfolgen. Nach der Implementierung des konstanten Speicherns der Bewegungsdaten in einem Ringpuffer traten hin und wieder Hardfaults auf. Ein Hardfault ist im Kontext von ARM Cortex-M Prozessoren eine Exception, die durch mehrere unterschiedliche Probleme ausgelöst werden kann. So kann ein falscher Speicherzugriff oder ein Problem beim Zugriff auf Sensoren über eine serielle Schnittstelle einen Hardfault auslösen.
In diesem konkreten Fall stellte sich als Ursache heraus, dass der Temperatursensor, der im Bewegungssensor integriert ist, beim Ladevorgang zur Überprüfung der Chiptemperatur verwendet wurde. Der Hardfault trat auf, wenn diese Anfrage mit der Bewegungsmessung auf dem gleichen Chip kollidierte.
Die Lösung für dieses Problem war, eine kurze Unterbrechung der Bewegungsmessung zu veranlassen, um die Temperatur auszulesen. Anschließend wird die Bewegungsmessung normal weiter geführt.

6.6.3 Sturzerkennung während des Sendevorgangs

Der Sendevorgang enthält einen Großteil der Funktionalität des SmarKo Gerätes. Hier werden das GPS und das GSM Modul initialisiert, der Puls gemessen, der Standort ermittelt und im Anschluss die gesammelten Daten inklusive der Daten der nächsten Mobilfunktürme und bei Bedarf Bewegungsdaten in einem JSON-String zusammengeführt und über eine GPRS Verbindung an den Server übermittelt.
Abhängig von der Position kann die Ermittlung des Standortes mehrere Minuten dauern. Der Sendevorgang kann aufgrund einer schlechten Verbindung abbrechen und muss dann neu begonnen werden.
Sollte in dieser Zeit ein potentieller Sturz erkannt werden, dann würden durch die aufgeteilte Behandlung erst nach dem Sendevorgang die relevanten Daten gesammelt. Um dem entgegenzuwirken wurden innerhalb der Sendefunktion an verschiedenen Stellen, insbesondere aber in Warteschleifen, Kontrollen der gesetzten Flags eingebaut. In Abb. 8 sind diese Stellen orange markiert. Wurde vom Sturzinterrupt eine Flag gesetzt während der Sendevorgang läuft, dann wird dieser abgebrochen, der Sturz behandelt und im Anschluss ein neuer Sendevorgang angestoßen.

6.6.4 Fehlschlag des Sendevorgangs bei zu großer Datenmenge

Die ursprüngliche Datenmenge pro Sendevorgang ist fix und beinhaltet ca. 15 Werte, die in einem JSON String gespeichert und übertragen werden. Es gab hier keinen Grund, den Sendevorgang auch für große Datenmengen verwendbar zu machen. Da bei einem Sturz eine große Menge Daten übertragen werden müssen, gab es an dieser Stelle einige Probleme. Insbesondere wurde nur ein Teil des JSON Strings übertragen, was einen Abbruch der Verbindung seitens des Servers zur Folge hatte. Die Lösung für dieses Problem war es schließlich, die Daten aufzuteilen und als Chunks an den Server zu senden. Das gestaltete sich aufgrund der speziellen Architektur schwieriger, als bisher gedacht, da es nicht mehr möglich war, den JSON-String erst komplett zu erstellen, in den Sendebuffer zu schreiben und dann mit einem Mal zu versenden. Die Erstellung des Sendebuffers wurde in den Sendevorgang integriert und dieser so angepasst, dass er den JSON-String Stück für Stück in den Buffer schreibt und sendet. Diese Funktionalität ist zum Zeitpunkt dieser Arbeit noch unvollständig, da auch der Server angepasst werden muss.

7 Verbesserungsmöglichkeiten und Ausblick

Nachdem die bisher erbrachte Grundlagenarbeit in den vorangegangenen Kapiteln beschrieben wurde, soll hier ein Ausblick geboten werden auf die Möglichkeiten die sich dadurch bieten. Insbesondere sollen mögliche Verbesserungen der generellen Firmware des Armbands und des groben Sturzerkennungsalgorithmus im speziellen aufgezeigt, sowie erste Ansätze zur Verwendung der Daten und zur weiteren Erweiterung der Funktionalität geliefert werden.

7.1 Echtzeit Betriebssystem

Die Anforderungen an die Funktionen von SmarKo haben sich seit der initialen Konzeptionierung stark verändert. Während das ursprüngliche Gerät in fixen Abständen eine kleine Menge an Daten senden musste und darüber hinaus wenig komplexe Funktionalität besaß, war es einfacher, die Software ohne Echtzeit Betriebssystem(Real Time Operating System, RTOS) direkt zu implementieren. Durch immer komplexere Funktionen, die auch unabhängig von einander laufen müssen, wird diese Architektur immer schwieriger zu warten. Neue Funktionen sind komplizierter einzufügen und der Zeitaufwand für ihre Implementierung steigt, weil sie manuell daran gehinderter werden müssen sich gegenseitig zu behindern. Ein Real RTOS kann hier immense Vorteile bringen. Das wichtigste Feature eines solchen RTOS ist ein Scheduler sowie mehr oder weniger ausgereifte Blocking-Mechanismen, um die parallele Ausführung mehrerer Tasks zu ermöglichen. Weiterhin implementieren einige RTOS auch ein Hardware Abstraction Layer um den Zugriff auf verschiedenste Sensoren und andere Hardwarekomponenten zu standardisieren und vereinfachen. Die Einführung eines solchen Betriebssystems würde initial einen großen Zeitaufwand bedeuten, aber die anschließende Entwicklungszeit von neuen Features und die Integration neuer Hardware würde damit massiv vereinfacht.

7.2 Integration von Bluetooth und GPRS als Fallback

Die nächste Hardwareiteration des Armbandes wird ein integriertes Bluetoothmodul beinhalten. Dieses eröffnet einige Möglichkeiten zur Kommunikation und Standortbestimmung. Die Datenübertragung könnte, solange möglich, per Bluetooth und damit stromsparender durchgeführt werden. Eine mobile Datenverbindung müsste nur nach Verlassen des Gebäudes aufgebaut werden. Eine Möglichkeit zur Standortbestimmung per Bluetooth wird weiter unten aufgezeigt. Diese Lösung würde die Vorteile einer Basisstation mit der Autonomie verbinden, die GSM Modem und GPS im freien bieten.

7.3 komplexere Auswertung der Accelerometer und Gyroskop Daten

Mit einem Wechsel vom Bewegungsinterrupt zur konstanten Überprüfung der erhobenen Werte, kann auch die Vorauswahl direkt auf dem Gerät komplexer gestaltet werden. Statt alle Vorkommnisse einer starken Beschleunigung zur weiteren Verarbeitung an den Server zu senden, wäre es dann möglich, eine ähnliche Strategie zu verfolgen, wie Degen[Degen et al., 2003] oder Kostopoulos[Kostopoulos et al., 2016] und einen Sturz in mehrere Phasen aufzuteilen. Dabei muss ein Kompromiss gefunden werden, sodass immer noch potentielle Stürze an den Server gesendet werden, selbst wenn sie einen weniger häufig auftretenden Verlauf haben. Ein langsamer Sturz zum Beispiel, bei dem sich der Stürzende an einem Möbelstück oder einer Wand festhält, wäre mit dieser Technik leicht zu übersehen.

7.4 Vitaldaten als zusätzliche Klassifikatoren für Stürze

Ein großer Vorteil des SmarKo Armbandes ist, dass neben den Orts- und Bewegungsdaten auch Vitaldaten unkompliziert erhoben werden können. Insbesondere können schon jetzt Puls und Blutsauerstoffwert abgefragt werden. Eine Messung der Körpertemperatur ist in der nächsten Hardwareiteration integriert.
Ein Sturz stellt eine starke Stressquelle dar. Symptome von Stress können unter anderem erhöhte Körpertemperatur und ein erhöhter Puls sein. Wang et al. haben mit einem Smart Home basierten Sturzerkennungssystem gute Erfolge erzielt, das den Puls in seine Berechnung mit einbezieht.([Wang et al., 2014])

7.5 Einbeziehung des Standortes

Das SmarKo Armband besitzt schon jetzt einen GPS Empfänger, um den Träger schnell unter offenem Himmel zu orten. Bei einem Notruf können Helfer so schnell zum Aufenthaltsort des Trägers dirigiert werden. Sehr viele Stürze, insbesondere bei älteren Menschen, ereignen sich allerdings in Gebäuden, insbesondere am Wohnort. Die nächste Iteration der SmarKo Hardware besitzt einen Bluetooth Sender und Empfänger. Damit wäre eine Kombination von mehreren Techniken möglich. Durch das Aufstellen einer Basisstation mit einem Bluetooth Beacon wäre am Wohnort des Trägers, sei es in der eigenen Wohnung oder in einer Pflegeeinrichtung, die genaue Ortung bei einem Unfall möglich. Gleichzeitig ist durch GPS und Mobilfunkmasten basierte Geolocation auch die Möglichkeit der Ortung außerhalb des Wohnortes gegeben. Es können auch mehrere Beacons verwendet werden, beispielsweise in der Wohnung und in einer Tagespflegeeinrichtung.

7.6 Erweiterung der Bewegungserkennung

Umfassende Bewegungsdaten von Accelerometer und Gyroskop können nicht nur in Hinblick auf Stürze ausgewertet werden. Insbesondere die Funktion SmarKos als Teil eines größeren Systems bestehend aus einem oder mehreren SmarKo Geräten, dem zentralen Server sowie anderen Komponenten wie Smartphone Apps oder zusätzlichen Sensoren kann stark von einer umfassenderen Bewegungserkennung profitieren.
Viele Krankheiten äußern sich in einer charakteristischen Gangart(Gait). Gait Analysis ist ein gängiges Konzept in der Medizin(z.B.[Whittle, 2014]). Ein SmarKo an der Hüfte oder am Bein könnte hier andere Methoden unterstützen und die Möglichkeit bieten, die Gangart konstant zu messen und Veränderungen schnell zu erkennen, selbst wenn sie nicht während einer Beobachtung durch z.B. eine kamerabasierte und damit ortsabhängige Methode wahrgenommen werden können.

7.7 Verwendung der gesammelten Daten, um andere Notfälle zu erkennen

Die erhobenen Daten könnten schließlich auch verwendet werden, um Notfälle zu erkennen, die kein Sturz sind. Ist zum Beispiel der Puls einer Patientin über längere Zeit außerhalb der Norm oder bewegt sich die Trägerin nicht oder sehr wenig zu einer Zeit, in der normalerweise hohe Aktivität herrscht, könnten das Indizien für einen Notfall sein.

7.8 Nutzerprofile

Das Sammeln der Benutzerdaten ermöglicht natürlich auch das Erstellen individueller Nutzerprofile. Diese stellen eine große Herausforderung für den Datenschutz dar, da personenbezogene Daten, insbesondere medizinischer Natur, auch zum Schaden der Betroffenen verwendet werden können. Da der Server, auf dem die Daten gesammelt werde, jedoch von der Nutzerin selbst oder einer Vertrauensperson betrieben wird, eröffnen sich so Möglichkeiten, Notfälle und Probleme schnell zu erkennen und gleichzeitig ein hohes Maß an Datenschutz zu gewährleisten.
Ähnlich wie bei Kostopoulos[Kostopoulos et al., 2016] könnten diese Nutzerprofile die Wahrscheinlichkeit von Stürzen mithilfe der Kombination von Ort, Zeit und anderen Werten dynamisch anpassen. Wurde beispielsweise ein Sturz mit verhältnismäßig geringer Überschreitung der Grenzwerte im Schlafzimmer registriert zu einer Zeit, in der die Trägerin sich normalerweise gerade zum Schlafen ins Bett legt, dann wäre es wahrscheinlich, dass kein Sturz vorliegt, sondern nur die Bewegung des Hinlegens registriert wurde. Wurde der schwache Sturz allerdings zu einem Zeitpunkt registriert, an dem die Trägerin laut Profil relativ aktiv ist, dann ist ein tatsächlicher Sturz wahrscheinlich.

7.9 Anpassung auf krankheits- und nutzerspezifische Risiken

Nutzerprofile in Verbindung mit den anderen gesammelten Daten könnten individuell an unterschiedlichste Risikofaktoren oder Krankheitsbilder angepasst werden und dabei zum Beispiel Patientinnen mit Diabetes- oder Herzerkrankung frühzeitig auf Probleme hinweisen.

8 Fazit

Die Implementierung der Sturzerkennung auf einem bestehenden Gerät stellte sich trotz des verhältnismäßig simplen Algorithmus als schwerer heraus als anfangs gedacht. Das Fehlen eines Schedulers und die daraus resultierende Notwendigkeit, die Ausführung und Unterbrechung der einzelnen Funktionen manuell zu verwalten stellten dabei wohl eines der größten Probleme dar. Mit zunehmender Komplexität des Gerätes und seiner Funktionen wird es deshalb wohl unabdingbar sein, ein Echtzeitbetriebssystem zu verwenden.
Da eine Studie über den vorgeschlagenen Algorithmus und seine Implementierung den Rahmen dieser Arbeit gesprengt hätte, sind die einzelnen Parameter der Implementierung wie Schwellwert und Anzahl der Messungen nach eigenem Ermessen des Autors gewählt worden.
Die nächsten Schritte müssen also die Erstellung des serverseitigen Teils des Algorithmus sowie die Erstellung eines aussagekräftigen Datensatzes sein, um die einzelnen Parameter einzustellen und den gesamten Algorithmus weiter verbessern zu können.
Das Handgelenk ist für eine auf einem Bewegungssensor basierende Sturzerkennung keine optimale Position. Ein Gerät, dessen Funktion ausschließlich die Sturzerkennung ist, sollte besser an der Hüfte oder dem Oberkörper getragen werden.
Allerdings kann die Verwendung solcher Single-Use-Geräte schnell dazu führen, dass die Trägerin drei oder mehr Geräte am Körper tragen muss. Es wird schwer möglich sein, eine Patientin davon zu überzeugen, einen Panikknopf um den Hals, einen Sturzsensor an der Hüfte und ein Gerät zur Pulsüberwachung am Handgelenk zu tragen. Mit zunehmender Anzahl der Geräte wird das Anlegen immer mühsammer und die Chance, dass eines oder mehrere der Geräte vergessen oder aus Frust nicht mehr getragen werden, steigt.
Sind hingegen diese Funktionen in einem kleinen und unaufdringlichen Gerät vereint, so steigt die Akzeptanz bei Endnutzerinnen. Da das für diese Arbeit verwendete Gerät schon bestand, stellte sich die Frage der Wahl nicht.
Sie wäre jedoch mit großer Wahrscheinlichkeit auf ein Armband gefallen. Die Ähnlichkeit zu einer Armbanduhr und die gute Postition zur Messung diverser Vitaldaten macht diesen Formfaktor zu einer guten Allroundlösung für die unterschiedlichen Aufgaben, die das SmarKo Armband zu erfüllen hat. Unabhängig davon, ob das Gerät allein oder in Verbindung mit anderen Sensoren verwendet wird, besteht hier großes Potential, eine Lösung zu schaffen, die es effektiv und kostengünstig ermöglicht, eine umfassendere medizinische Überwachung zu gewährleisten.

Abbildungsverzeichnis

Literatur

[Abbate et al., 2012] Abbate, S., Avvenuti, M., Bonatesta, F., Cola, G., Corsini, P., and Vecchio, A. (2012). A smartphone-based fall detection system. *Pervasive and Mobile Computing*, 8(6):883–899.

[Ageing and Unit, 2008] Ageing, W. H. O. and Unit, L. C. (2008). *WHO global report on falls prevention in older age*. World Health Organization.

[Aguiar et al., 2014] Aguiar, B., Rocha, T., Silva, J., and Sousa, I. (2014). Accelerometer-based fall detection for smartphones. In *Medical Measurements and Applications (MeMeA), 2014 IEEE International Symposium on*, pages 1–6. IEEE.

[Bourke et al., 2007] Bourke, A., O'brien, J., and Lyons, G. (2007). Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. *Gait & posture*, 26(2):194–199.

[Chen et al., 2006] Chen, J., Kwong, K., Chang, D., Luk, J., and Bajcsy, R. (2006). Wearable sensors for reliable fall detection. In *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pages 3551–3554. IEEE.

[Dai et al., 2010] Dai, J., Bai, X., Yang, Z., Shen, Z., and Xuan, D. (2010). Perfalld: A pervasive fall detection system using mobile phones. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pages 292–297. IEEE.

[Degen et al., 2003] Degen, T., Jaeckel, H., Rufer, M., and Wyss, S. (2003). Speedy: A fall detector in a wrist watch. In *ISWC*, pages 184–189.

[Edgcomb and Vahid, 2012] Edgcomb, A. and Vahid, F. (2012). Privacy perception and fall detection accuracy for in-home video assistive monitoring with privacy enhancements. *ACM SIGHIT Record*, 2(2):6–15.

[FOKUS,] FOKUS. Fraunhofer-institut für offene kommunikationssysteme(fokus). Website: www.fokus.fraunhofer.de.

[Friedman et al., 2002] Friedman, S. M., Munoz, B., West, S. K., Rubin, G. S., and Fried, L. P. (2002). Falls and fear of falling: which comes first? a longitudinal prediction model suggests strategies for primary and secondary prevention. *Journal of the American Geriatrics Society*, 50(8):1329–1335.

[Funk and Pierobon, 2007] Funk, M. and Pierobon, A. (2007). *Sturzprävention bei älteren Menschen: Risiken-Folgen-Maßnahmen*. Georg Thieme Verlag.

[Igual et al., 2013] Igual, R., Medrano, C., and Plaza, I. (2013). Challenges, issues and trends in fall detection systems. *Biomedical engineering online*, 12(1):66.

[Kangas et al., 2008] Kangas, M., Konttila, A., Lindgren, P., Winblad, I., and Jämsä, T. (2008). Comparison of low-complexity fall detection algorithms for body attached accelerometers. *Gait & posture*, 28(2):285–291.

[Kostopoulos et al., 2016] Kostopoulos, P., Kyritsis, A. I., Deriaz, M., and Konstantas, D. (2016). F2d: A location aware fall detection system tested with real data from daily life of elderly people. *Procedia Computer Science*, 98:212–219.

[Kröhnert et al., 2006] Kröhnert, S., Medicus, F., and Klingholz, R. (2006). Die demografische lage der nation. *Wie zukunftsfähig sind Deutschlands Regionen*, 2.

[Li et al., 2009] Li, Q., Stankovic, J. A., Hanson, M. A., Barth, A. T., Lach, J., and Zhou, G. (2009). Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information. In *Wearable and Implantable Body Sensor Networks, 2009. BSN 2009. Sixth International Workshop on*, pages 138–143. IEEE.

[Lindemann et al., 2005] Lindemann, U., Hock, A., Stuber, M., Keck, W., and Becker, C. (2005). Evaluation of a fall detector based on accelerometers: A pilot study. *Medical and Biological Engineering and Computing*, 43(5):548–551.

[MCS DataLabs,] MCS DataLabs. MCS DataLabs. Website: www.mcs-datalabs.com.

[Özdemir and Barshan, 2014] Özdemir, A. T. and Barshan, B. (2014). Detecting falls with wearable sensors using machine learning techniques. *Sensors*, 14(6):10691–10708.

[Rubini and Corbet, 2001] Rubini, A. and Corbet, J. (2001). *Linux device drivers*. Ö'Reilly Media, Inc.".

[Tunstall, 2016] Tunstall (2016). Tunstall vibby fall detector.

[Wang et al., 2014] Wang, J., Zhang, Z., Li, B., Lee, S., and Sherratt, R. (2014). An enhanced fall detection system for elderly person monitoring using consumer home networks. *IEEE transactions on consumer electronics*, 60(1):23–29.

[Warren et al., 1999] Warren, A., Rosenblatt, A., and Lyketsos, C. G. (1999). Wandering behaviour in community-residing persons with dementia. *Int. J. Geriat. Psychiatry*, 14:272–279.

[Whittle, 2014] Whittle, M. W. (2014). *Gait analysis: an introduction*. Butterworth-Heinemann.

[Wireless Power Consortium, 2017] Wireless Power Consortium (2017). qi Wireless Charging Standard.

[Zhang et al., 2006] Zhang, T., Wang, J., Xu, L., and Liu, P. (2006). Fall detection by wearable sensor and one-class svm algorithm. In *Intelligent computing in signal processing and pattern recognition*, pages 858–863. Springer.