



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Projektbericht Masterprojekt 2

Frank Hardenack

Kapazitive Body-Sense Funktionen in der Praxis

Inhaltsverzeichnis

1	Einleitung	1
2	Erfahrungen mit dem CapSense Toolkit	2
2.1	Einbau im intelligenten Bett	2
2.2	Sturzerkennung durch kapazitive Sensoren im Teppich	3
3	Erstellung des Prototypen	4
3.1	Modifikation des entwickelten Lattenrost-Sensors	4
3.2	Bau des Prototypen	4
3.2.1	Positionierung der Sensoren	6
4	Visualisierungssoftware	7
5	Vorarbeiten zur Masterarbeit	8
5.1	Nutzung maschineller Lernverfahren	8
5.1.1	Die JavaML-Bibliothek	8
5.2	Installation der MacMinis	9
6	Ausblick	10
	Literaturverzeichnis	11

Abbildungsverzeichnis

1	CapSense Board (http://www.capsense.org)	2
2	Kabel-Matrix der Sturzerkennung	3
4	Prototyp im Projektraum	4
3	Modifizierter Lattenrost-Sensor	5
3.1	(a)	5
3.2	(b)	5
3.3	(c)	5
3.4	(d)	5
3.5	(e)	5
3.6	(f)	5
5	Liegepositionen (www.wickerklinik.de)	6
6	Messwerte bei einer vorsichtigen Umdrehbewegung (90°)	6
6.1	Schulter	6
6.2	Hüfte	6
6.3	Knie	6
7	MacMini Server	9

1 Einleitung

Bei dieser Arbeit handelt es sich um einen Projektbericht zum Masterprojekt 2. Neben der im Masterprojekt 1 (Hardenack, 2010) in Aussicht gestellten Untersuchung des CapSense Toolkits¹ auf Qualität der Messwerte und Einsetzbarkeit im Bereich der Smart Furniture wurde der Schwerpunkt auf die Erstellung eines Prototypen des intelligenten Bettes sowie der Vorbereitung der Hardware für die Masterarbeit gelegt.

Ebenso war es ein Ziel des Masterprojekt 2, eine geeignete Java-Bibliothek zur Nutzung maschineller Lernverfahren für den Einsatz bei der Interpretation der erfassten Messwerte zu finden.

Die zusammen mit Oliver Dreschke entwickelte Visualisierungssoftware wurde im Verlauf des Masterprojekt 2 im großen Stil refaktoriert und um Funktionalität für die Darstellung und Nutzung weiterer Sensortypen sowie Möglichkeiten der Datenübertragung erweitert. In Abschnitt 2 werden die ersten Erfahrungen mit dem CapSense-Toolkit sowie die aufgetretenen Probleme bei der Verwendung im intelligenten Bett beschrieben. Abschnitt 3 beschäftigt sich mit der Erstellung des Prototypen sowie der Modifikation des Lattenrost-Sensor, während Abschnitt 4 sich mit der Visualisierungssoftware und den darin vorgenommenen Änderungen befasst. Getätigte vorbereitende Arbeiten zur Masterarbeit wie die Versuche mit maschinellen Lernverfahren und Vorbereitungen im Hardwarebereich werden in Abschnitt 5 näher erläutert. Ein Ausblick auf das weitere Vorgehen schließt diese Arbeit in Abschnitt 6 ab.

¹<http://www.capsense.org>

2 Erfahrungen mit dem CapSense Toolkit

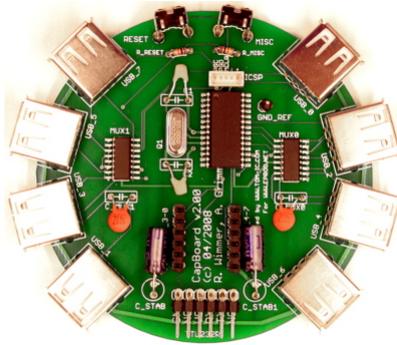


Abbildung 1: CapSense Board
(<http://www.capsense.org>)

Dem in (Hardenack, 2010) vorgenommenen Vergleich von Schwingkreiserzeugern muss die Prüfung des CapSense-Toolkits auf Einsetzbarkeit im intelligenten Bett hinzugefügt werden, da das Toolkit zum Masterprojekt 1 noch nicht verfügbar war.

Das CapSense-Toolkit bedient sich bei der Funktionsweise intelligenter Sensoren. Das verwendete Messverfahren für die Annäherung an eine Sensorfläche funktioniert dabei mittels eines Timer-IC (LMC555), der in Abhängigkeit von der Kapazität eines angeschlossenen Kondensators eine nichtlineare Frequenz bis 3 MHz ausgibt. Dadurch wird ein elektrisches Feld rund um die Elektrode erzeugt, durch welches eine Annäherung von Objekten an die

Elektrode detektiert werden kann, da sich die Frequenz des Timers durch eben diese Annäherung verändert (Raphael Wimmer, 2007).

Sehr interessant ist beim CapSense-Toolkit der vergleichsweise geringe Anschaffungspreis sowie die komplett quelloffene Firmware sowie das Protokoll. Leider ist aber anzumerken, dass nicht immer alle Messwerte vom CapSense-Toolkit an den PC übertragen werden, ebenso wie die Tatsache, dass das gezielte Ein- und Ausschalten einzelner Sensoren nur bedingt funktioniert. Damit erfüllt das CapSense-Toolkit zwar nicht die Anforderungen, die an die Produkte aus der Sicherheitstechnik gestellt werden, reicht aber für den Einsatz im Bereich der Smart Furniture vollkommen aus.

2.1 Einbau im intelligenten Bett

Nach der in Abschnitt 3.1 beschriebenen Modifikation des Lattenrost-Sensors im Rahmen der Erstellung des Prototyps wurde das CapBoard zur Erfassung der Messdaten an den Prototyp angeschlossen. Dabei ergab sich, dass die Kapazitäten in dem entwickelten Sensor zu hoch sind, um mit dem CapSense-Toolkit verwertbare Messwerte zu erfassen. Anstatt einen Rückschluss über die Auslenkung des Lattenrosts aus der Ruhelage ziehen zu können wurde in den Messwerten eine gestreckte Sinusschwingung sichtbar.

Das CapSense-Toolkit eignet sich somit nicht für den Einsatz im intelligenten Bett. An dieser Stelle sei aber auf die Bachelorarbeit von Ann-Kathrin Weiss (2011) sowie die Masterarbeit von Oliver Dreschke (2011) verwiesen. In diesen Arbeiten geht es um die Entwicklung eines intelligenten Sofas unter Verwendung textiler, kapazitiver Sensoren, für die unter anderem das CapSense-Toolkit benutzt wurde. Neben diesen Abschlussarbeiten entwickelte Oliver Dreschke eine simple Sturzerkennung, für die ebenfalls das CapSense-Toolkit verwendet wurde.

2.2 Sturzerkennung durch kapazitive Sensoren im Teppich

Die Beschreibung der Sturzerkennung soll an dieser Stelle nur ein kleiner Exkurs sein, um einen Blick über den Tellerrand zu wagen, in welchen Bereichen das CapSense-Toolkit sinnvoll einsetzbar ist.

Die Sturzerkennung auf Basis des CapSense-Toolkits verwendet eine Matrix aus unter dem Teppich angebrachten Kabeln. Anhand der Veränderung der belasteten („belegten“) Kreuzungspunkte lässt sich eine Aussage treffen, ob ein Mensch gestürzt ist, indem schlagartig mehrere Kreuzungspunkte belegt sind, oder sich normal fortbewegt, wenn sich nur einzelne bzw. wenige Kreuzungspunkte verändern. Die Visualisierungssoftware (siehe Abschnitt 4) wurde entsprechend angepasst, um die belasteten Kreuzungspunkte anzuzeigen. Für die genaue Funktionsweise sowie die technische Realisierung sei auch an dieser Stelle auf die Masterarbeit von Oliver Dreschke (2011) verwiesen.

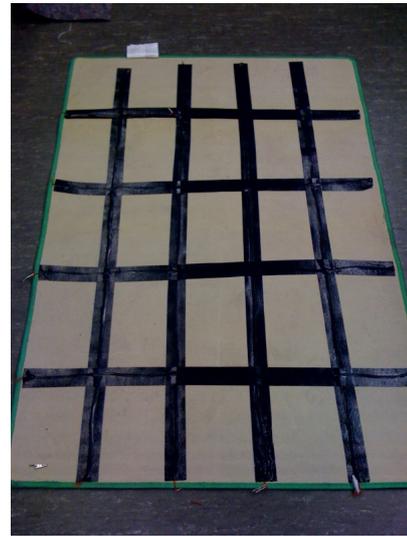


Abbildung 2: Kabel-Matrix der Sturzerkennung

3 Erstellung des Prototypen

In diesem Abschnitt wird die vorgenommene Modifikation des Lattenrost-Sensors sowie die Erstellung eines ersten Prototypen des intelligenten Bettes beschrieben.

3.1 Modifikation des entwickelten Lattenrost-Sensors

Der im Masterprojekt 1 entwickelte Sensor in Form einer modifizierten Latte des Lattenrosts war noch nicht ausgereift genug, um im Prototyp des Bettes verbaut zu werden. Die über und unter der Latte angebrachte Schirmung aus leitfähigem Gewebe verformte sich bei mehrfachem Durchbiegen des Lattenrosts nachhaltig und führte so zu stark schwankenden Messergebnissen.

Um die Schirmung besser aufzubringen wurde diese ebenfalls in Form von leitfähigem Aluminiumklebeband vorgenommen. Dadurch entstand der in Abbildung 3 von oben links nach unten rechts dargestellte, siebenschichtige Aufbau aus Aluminiumklebeband und Isolierband. Der Anschluss des Sensors erfolgt nach wie vor über ein Koaxialkabel. Der weiterentwickelte Aufbau hat den Vorteil, dass die Schirmung nun immer den gleichen Abstand zur Sensorfläche behält und der aus der locker aufliegenden Schirmung resultierende Seiteneffekt behoben wurde. Durch das Aufbringen des Sensors auf der Unterseite des Lattenrosts verschwindet dieser nun auch optisch „unter“ dem Bett und fügt sich noch besser als bisher in ein bestehendes Wohnumfeld ein. Diese Integration in den Wohnbereich sowie die komplette Schirmung wirken eventuellen Akzeptanzproblemen späterer Nutzer entgegen. Zum einen trägt die neue Bauform gar nicht mehr gegenüber einem Standard-Lattenrost auf, zum anderen wird Argumentationen gegen ein den Körper während der Nacht umgebendes elektrisches Feld durch die komplette Schirmung präventiv entgegengewirkt.

3.2 Bau des Prototypen



Abbildung 4: Prototyp im Projekttraum

Mit dem modifizierten Lattenrost-Sensor wurde ein Lattenrost im Projekttraum prototypisch bestückt. Dabei wurden insgesamt drei Latten mit einem Sensor ausgestattet, deren Positionierung dabei sehr wichtig für die Erfassung der Bewegungen ist und in Abschnitt 3.2.1 näher beschrieben wird. Mit der gewählten Positionierung lassen sich mit dem Prototypen unter Verwendung der Schwingkreiserzeuger von RSI bereits minimale Bewegungen wie das Anheben eines Beines oder leichte Drehbewegungen detektieren. Die Sensibilität geht so weit, dass für den geübten Betrachter bei Bauchschläfern Atembewegungen sichtbar sind.

Bei Verformung des Lattenrosts verändern sich die Messwerte nachhaltig und kehren erst zum Ausgangswert zurück, wenn die Latte des Lattenrosts

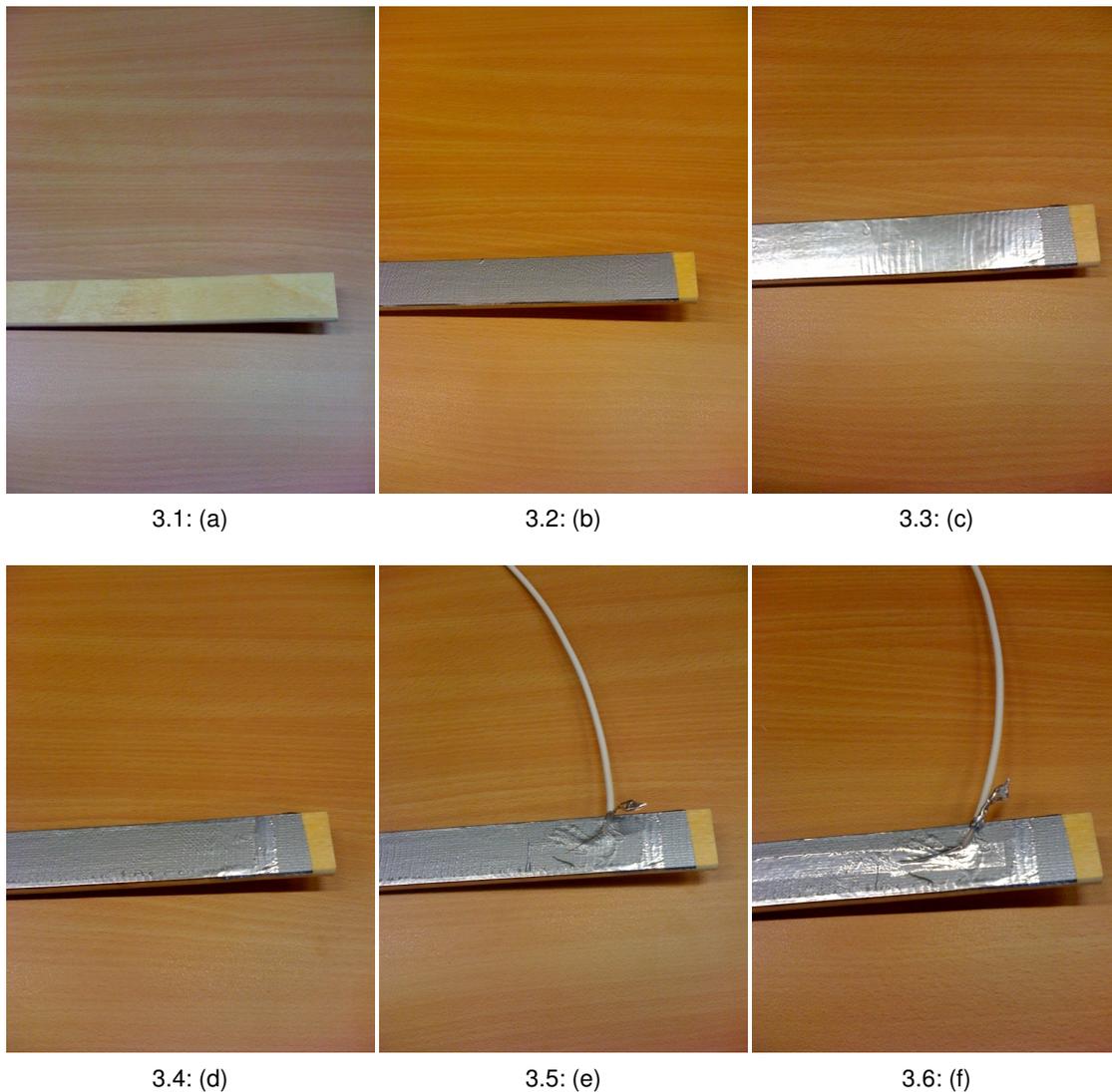


Abbildung 3: Modifizierter Lattenrost-Sensor

komplett entlastet wird. Diese anhaltende Veränderung birgt Schwierigkeiten bei der Verarbeitung und Interpretation der absoluten Messwerte, kann aber durch die Verwendung der Streuung oder der Varianz über eine noch festzulegende Anzahl der letzten Messwerte ermöglicht werden.

Durch das Loslösen von den absoluten Messwerten und den Übergang zu relativen Messwerten (Streuung, Varianz) lassen sich gleichzeitig Umwelteinflüsse wie Temperaturschwankungen und Veränderungen der Luftfeuchtigkeit ebenso wie auftretende Gewichtsunterschiede verschiedener Nutzer minimieren oder gar kompensieren.

3.2.1 Positionierung der Sensoren

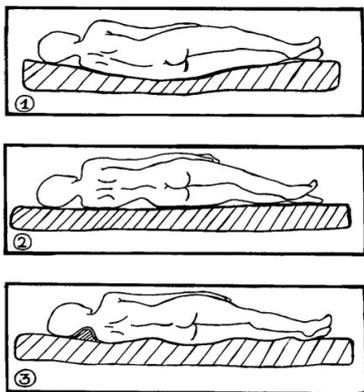
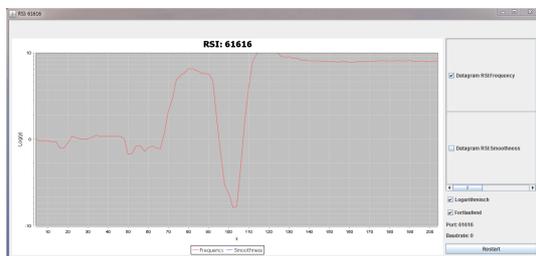


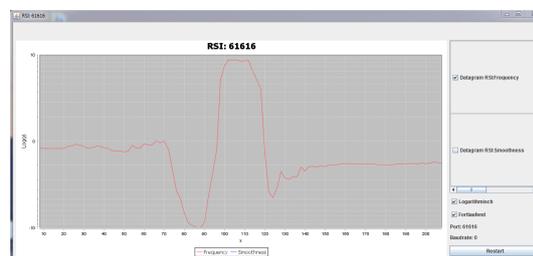
Abbildung 5: Liegepositionen (www.wickerklinik.de)

Die unter Punkt drei der nebenstehenden Abbildung gezeigte Liegeposition zeigt zum einen die optimale Liegeposition mit einer geraden Wirbelsäule, zum anderen aber auch zwei Körperschwerpunkte auf Schulter- und Hüfthöhe. An eben diesen Schwerpunkten wurden zwei Latten des Lattenrosts mit dem modifizierten Sensor aus Abschnitt 3.1 versehen. Bewegt sich ein Mensch im Bett, oder dreht er sich um, so findet an diesen beiden Schwerpunkten, die sich am tiefsten in die Matratze drücken, die stärkste Belastung des Lattenrosts statt. Diese Tatsache wird genutzt, um eine möglichst starke Auslenkung der Messwerte der Lattenrost-Sensoren zu erreichen. Eine dritte Latte wurde auf Höhe der Knie beziehungsweise der Unterschenkel modifiziert, um auch Bewegungen der Beine

zusätzlich zu den Bewegungen an den beiden Körperschwerpunkten erfassen zu können. Mit dieser Aufteilung der Sensoren wurden bereits erste, sehr zufriedenstellende Versuche zur Bewegungserkennung durchgeführt (Abbildung 6 zeigt die Veränderung der Messwerte der einzelnen Sensoren bei einer Drehung auf der Matratze um 90° von der Seitenlage in die Rückenlage).



6.1: Schulter



6.2: Hüfte



6.3: Knie

Abbildung 6: Messwerte bei einer vorsichtigen Umdrehbewegung (90°)

4 Visualisierungssoftware

Die im Verlauf des Masterprojekt 1 zusammen mit Oliver Dreschke entwickelte Software wurde im Rahmen vom Masterprojekt 2 um viele wichtige Punkte in Bezug auf Funktionalität ergänzt und in ihrer Struktur so umgebaut, dass sie leichter um neue Sensortypen oder Darstellungsformen zu erweitern ist.

Kommunikation über ActiveMQ

Neben der Ausführung der Software auf einem PC war die Vorbereitung und Erweiterung der Software zur Erfassung und Darstellung der Messwerte auf unterschiedlichen PCs eine wichtige Verbesserung der Software. Diese kann nun die Messdaten auf einem PC erfassen und sie mittels ActiveMQ² für andere Anwendungen oder die Visualisierung auf einem anderen PC nutzbar machen. Die auf dem ActiveMQ abgelegten Werte können von einem anderen PC mit der entwickelten Software abgerufen und in der Chart-Darstellung visualisiert oder von einer komplett anderen Anwendung des Living Place Hamburg weiterverarbeitet werden.

In der momentanen Konfiguration läuft die Kommunikation über einen eigenen ActiveMQ, da der produktive ActiveMQ des Living Place Hamburg in der Testphase der Software nicht unnötig belastet werden sollte. Über Konfigurationsparameter kann der genutzte ActiveMQ leicht umgestellt werden.

Plugin-Konzept

Das in der Software verwendete Plugin-Konzept ermöglicht eine schnelle Erweiterung um neue Sensortypen oder Darstellungsformen (zur Zeit: 3D-Modell des Sofas, Chart-Darstellung der Messwerte wie in Abbildung 6). Dazu muss nur von der jeweiligen abstrakten Oberklasse abgeleitet und die dort spezifizierten Methoden implementiert werden. Zur Repräsentation der Daten einer Messreihe muss zusätzlich ein Datagramm implementiert werden, das ebenfalls von einer abstrakten Oberklasse abgeleitet wird und Informationen über die einzelnen Messwerte sowie einen Timestamp enthält.

Refactoring

Die teils historisch gewachsenen Paket- und Klassenstrukturen der Software genügten nicht mehr den Ansprüchen, die man an eine wartbare und verständliche Software stellt. Dazu kamen irritierende Klassennamen sowie nicht saubere Code-Passagen. Aus diesen Gründen und in Hinblick auf das Plugin-Konzept wurde die Paketstruktur komplett überarbeitet und die Klassen refaktoriert, um eine saubere, verständliche und wartbare Struktur zu erhalten.

²<http://activemq.apache.org>

5 Vorarbeiten zur Masterarbeit

Das Masterprojekt 2 wurde auch genutzt um notwendige Vorarbeiten zur Masterarbeit wie die Installation von Hardware oder die Einarbeitung in eine Machine-Learning-Bibliothek vorzunehmen.

5.1 Nutzung maschineller Lernverfahren

Um erste Erfahrung mit maschinellen Lernverfahren und deren Implementierung und Nutzung in Java zu sammeln, wurden verschiedene freie Java-Bibliotheken ausprobiert. Dabei ist anzumerken, dass es sich bei vielen der online verfügbaren Bibliotheken um mittlerweile eingeschlafene Projekte handelt, deren Verwendung aufgrund schlechter Dokumentation und Literatur nur bedingt nützlich ist.

Als erster Schritt wurden im Speziellen Bibliotheken zur Nutzung von Support Vector Machines benutzt, da diese für den Grundgedanken der Klassifikation von Schlafphasen sehr geeignet schienen. Dabei fiel die engere Wahl auf LibSVM³ (inkl dem refaktorierten Javacode für schnelleres Training und Testing, der JLibSVM⁴), svmLight⁵ sowie JavaML⁶. Bei den ersten drei Frameworks waren die Beispiele und die Dokumentation leider sehr dürftig, weshalb diese frühzeitig ausschieden und die Wahl auf JavaML fiel.

5.1.1 Die JavaML-Bibliothek

Die JavaML-Bibliothek ist eine gut dokumentierte und mit reichlich Beispielen versehene Java-Bibliothek, die verschiedene, leicht austauschbare Lernverfahren (Support Vector Machine, K-Nearest-Neighbour, Bayesnetze uvm.) anbietet und dabei durch eine gute API leicht zu bedienen ist.

Anhand der mitgelieferten Beispiele wurden erste Implementierungen vorgenommen, um die Funktionsweise der Bibliothek zu durchdringen und erste Eindrücke zur Featurwahl zu erhalten. Die leichte Austauschbarkeit des genutzten Lernverfahrens macht diese Bibliothek flexibel in der Anpassung auf erweiterte beziehungsweise neue Problemstellungen im Bereich der Smart Furniture.

Weitere, nützliche Features dieser Bibliothek sind die Möglichkeit, ein Feature-Scoring zu nutzen, bei dem einzelnen Features eines Feature Vektors eine unterschiedliche Gewichtung bei der Klassifikation zugeordnet werden kann, sowie die Möglichkeit eines Feature-Rankings und der Nutzung von Feature-Subsets, bei denen für das eingesetzte Lernverfahren nur eine Untermenge der zur Verfügung stehenden Features unabhängig ihrer Qualität zur Klassifikation herangezogen wird.

³<http://www.csie.ntu.edu.tw/~cjlin/libsvm>

⁴<http://dev.davidsoergel.com/trac/jlibsvm>

⁵<http://svmlight.joachims.org>

⁶<http://java-ml.sourceforge.net>

5.2 Installation der MacMinis

Die sowohl für das intelligente Bett, als auch für das intelligente Sofa benötigte Rechenleistung soll durch MacMinis der fünften Generation in den Living Place Hamburg eingebracht werden. Die Wahl fiel dabei auf die Servervariante, die sich durch zwei Festplatten auszeichnet und so eine gute Basis zur Installation eines weiteren Betriebssystems neben MacOS darstellt.

Die Installation gestaltete sich durch die fehlende Unterstützung von Apple's Bootcamp in der MacOS X Server-Version allerdings komplizierter als anfangs vermutet. Es war nicht möglich, parallel zum vorinstallierten MacOS X Server ein Windows-Betriebssystem zu installieren. Erst durch die Installation eines *normalen* MacOS und der damit verbundenen Bootcamp-Unterstützung konnte ein weiteres Betriebssystem installiert werden.

Da die von uns gewählten Java-Bibliotheken zur Nutzung der seriellen Schnittstelle zum Erfassen der Sensordaten ein Windows-Betriebssystem benötigen war eine Installation von Windows 7 unumgänglich.



Abbildung 7: MacMini Server

6 Ausblick

Nachdem im Masterprojekt 2 mehrere Möglichkeiten zum Einsatz maschineller Lernverfahren untersucht wurden, tritt die Frage auf, wie man mit Featurevektoren variabler Größe umgehen kann. Nicht immer sind alle Feature-liefernden Services erreichbar und es kann beispielsweise keine Information über das Wetter, die Termine oder den Kontext des Nutzers zur Erstellung des Featurevektors herangezogen werden. Eine andere Möglichkeit für unvollständige Featurevektoren wäre das Ausfallen eines Sensors oder die Erfassung korrupter Messwerte.

Als mögliche Lösung ließen sich Dummywerte für die fehlenden Features einsetzen, womit man zwar das Problem wegfallender, nicht aber das Problem neu hinzukommender Features abdecken kann. Hierfür müssten für das Lernverfahren neue Trainingsdaten erstellt und das Verfahren neu trainiert und getestet werden. Für den laufenden Betrieb im Living Place Hamburg stellt dies einen großen Overhead dar, der in dieser Form aufgrund der dynamischen Entwicklung neuer Anwendungen im Living Place nicht umsetzbar ist. Abhilfe könnten in diesem Fall Regelsysteme (zum Beispiel Jess⁷ oder Drools⁸) und ein für die Anwendung zugeschnittenes, leicht erweiterbares Regelsystem bieten.

Im Hinblick auf die Masterarbeit kann man zusammenfassen, dass die Erfassung von Bewegungen mit dem selbst entwickelten Sensor soweit funktionstüchtig ist. Es fehlt nun die Aufbereitung sowie die semantische Interpretation der erfassten Daten, zu der erste Ansätze bereits in (Hardenack, 2011) erläutert wurden. Neben dem Ansatz zur Nutzung maschineller Lernverfahren soll als Alternative der Einsatz von Regelsystemen zur Interpretation von Schlafphasen genauer betrachtet werden.

⁷<http://www.jessrules.com/jess>

⁸<http://www.jboss.org/drools>

Literatur

- [Hardenack 2010] HARDENACK, Frank: Bodymonitoring in Smart Homes / HAW Hamburg. URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2010-proj/Hardenack.pdf>, 2010. – Forschungsbericht. [Projektbericht zum Master-Projekt 1, Sommersemester 2010]
- [Hardenack 2011] HARDENACK, Frank: Das intelligente Bett - Semantische Interpretation auf Basis kapazitiver Sensoren / HAW Hamburg. URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master10-11-seminar/hardenack/bericht.pdf>, 2011. – Forschungsbericht. [Ausarbeitung Master-Seminar]
- [Raphael Wimmer 2007] RAPHAEL WIMMER, Sebastian Boring Albrecht S.: A Capacitive Sensing Toolkit for Pervasive Activity Detection and Recognition. In: *Proceedings of the Fifth Annual IEEE Conference on Pervasive Computing and Communications (PerCom)*, März 2007