

Institut für Visualisierung und Interaktive Systeme
Abteilung Intelligente Systeme
Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Studienarbeit Nr. 2318

Erkennung von Aktivitäten mit Hilfe mobiler Geräte

Attila Györkös

Studiengang: Informatik

Prüfer: Prof. Dr. Gunther Heidemann

Betreuer: Dipl.-Inf. Julia Möhrmann

begonnen am: 20. Januar 2011

beendet am: 26. Mai 2011

CR-Klassifikation: G.3, I.2.6, I.4.8, I.5.4

Inhaltsverzeichnis

1 Einleitung	9
2 Technische Grundlagen	13
2.1 Hardware	13
2.1.1 Beschleunigungssensor	13
2.1.2 Magnetfeldsensor	14
2.2 Android	14
2.2.1 Sensoren Schnittstelle	14
2.2.2 Sensorkoordinatensystem	15
2.2.3 Samplingrate	15
2.3 Matlab	16
2.3.1 Entscheidungsbäume	16
2.3.2 HMM Toolbox	16
3 Datenakquisition	17
3.1 Software	17
3.2 Gerätelocation	17
3.3 Aktivitäten	18
4 Datenaufbereitung	25
4.1 Generierung zusammenhängender Datensätze	25
4.2 Glättung	25
4.3 Einteilung in Fenster	26
4.4 Koordinatentransformation	27
4.4.1 Exakte Transformation	27
4.4.2 Näherungsweise Transformation	27
4.5 Zyklen Erkennung	28
4.5.1 Vorverarbeitung	29
4.5.2 Abschätzen der Periodenlänge	31
4.5.3 Einteilung nach Zyklen	31
4.6 Merkmale	32
5 Entscheidungsbäume	33
5.1 Grundlagen	33
5.1.1 CART Algorithmus	33
5.1.2 Teilungskriterium	33
5.2 Merkmalsextraktion	34

5.3	10-fold Cross Validation	35
5.4	Ergebnisse	35
6	Hidden Markov Modelle	37
6.1	Grundlagen	37
6.2	Rohdaten	38
6.2.1	Struktur	38
6.2.2	Erkennungsraten	39
6.3	Einteilung in Intervalle	39
6.3.1	Merkmalsextraktion	40
6.3.2	Struktur	41
6.3.3	Erkennungsraten	42
7	Diskussion	45
7.1	Entscheidungsbäume	45
7.2	Hidden Markow Modelle	45
7.3	Vergleich der Verfahren	46
8	Zusammenfassung und Ausblick	49
A	Anhang	51
A.1	Weitere Konfusionsmatrizen	51
A.1.1	Entscheidungsbäume	51
A.1.2	HMM	53

Abbildungsverzeichnis

2.1	Sensorkoordinatensystem	15
3.1	Aufbau einer Datei	18
3.2	Gerätepositionen	19
3.3	Beschleunigungsverlauf: Aufzug fahren	21
3.4	Beschleunigungsverlauf: Rolltreppe fahren	21
3.5	Beschleunigungsverlauf: Ruhezustand	22
3.6	Beschleunigungsverlauf: Rennen	22
3.7	Beschleunigungsverlauf: Treppe hinuntersteigen	23
3.8	Beschleunigungsverlauf: Treppe hochsteigen	23
3.9	Beschleunigungsverlauf: Gehen	24
4.1	Generierung von Fenstern	26
4.2	Körperfeste Koordinatensysteme	29
4.3	Ergebnisse der Koordinatentransformation	30
4.4	Bestimmung der Zyklen	31
6.1	HMM Beispiel	37
6.2	HMM Struktur, Rohdaten	38
6.3	Intervalleinteilung für HMM	40
6.4	HMM Struktur, Features aus Intervallen	41
6.5	HMM Erkennungsraten	42
7.1	Vergleich der Erkennungsraten	47

Tabellenverzeichnis

2.1	T-Mobile G1 Technische Daten	13
2.2	SENSOR_DELAY_* Konstanten	16
4.1	Anzahl der Daten nach manueller Aufbereitung	25
4.2	Verwendete Fenstergrößen. Samples entsprechen den angegebenen Zeiträumen .	26

5.1	Erkennungsrate für Entscheidungsbäume nach Fenstergröße	36
6.1	Erkennungsrate für HMM, Rohdaten nach Fenstergröße	39
6.2	Gesamterkennungsraten HMM	43
6.3	Erkennungsrate HMM, Features aus Fenstergröße 256	43
6.4	Erkennungsraten HMM mit Zyklenerkennung	44
7.1	Vergleich der Erkennungsraten	48
7.2	Vergleich Erkennungsraten für Fenstergröße 256	48
A.1	Konfusionmatrix Entscheidungsbaum, Fenstergröße 32	51
A.2	Konfusionmatrix Entscheidungsbaum, Fenstergröße 64	51
A.3	Konfusionmatrix Entscheidungsbaum, Fenstergröße 256	52
A.4	Konfusionmatrix Entscheidungsbaum, Fenstergröße 512	52
A.5	Konfusionmatrix Entscheidungsbaum, Neuster Zyklus	52
A.6	Konfusionmatrix HMM, Fenstergröße 512, Features aus 5 Intervallen	53
A.7	Konfusionmatrix HMM, Fenstergröße 512, Features aus 7 Intervallen	53
A.8	Konfusionmatrix HMM, Fenstergröße 512, Features aus 11 Intervallen	53
A.9	Konfusionmatrix HMM, Fenstergröße 512, Features aus 13 Intervallen	54
A.10	Konfusionmatrix HMM, Fenstergröße 512, Features aus 15 Intervallen	54
A.11	Konfusionmatrix HMM, Fenstergröße 256, Features aus 5 Intervallen	54
A.12	Konfusionmatrix HMM, Fenstergröße 256, Features aus 7 Intervallen	55
A.13	Konfusionmatrix HMM, Fenstergröße 256, Features aus 9 Intervallen	55
A.14	Konfusionmatrix HMM, Fenstergröße 256, Features aus 11 Intervallen	55
A.15	Konfusionmatrix HMM, Fenstergröße 256, Features aus 13 Intervallen	56
A.16	Konfusionmatrix HMM, Fenstergröße 256, Features aus 15 Intervallen	56
A.17	Konfusionmatrix HMM, Fenstergröße 128, Features aus 5 Intervallen	56
A.18	Konfusionmatrix HMM, Fenstergröße 128, Features aus 7 Intervallen	57
A.19	Konfusionmatrix HMM, Fenstergröße 128, Features aus 9 Intervallen	57
A.20	Konfusionmatrix HMM, Fenstergröße 128, Features aus 11 Intervallen	57
A.21	Konfusionmatrix HMM, Fenstergröße 128, Features aus 13 Intervallen	58
A.22	Konfusionmatrix HMM, Fenstergröße 128, Features aus 15 Intervallen	58
A.23	Konfusionmatrix HMM, Fenstergröße 64, Features aus 3 Intervallen	58
A.24	Konfusionmatrix HMM, Fenstergröße 64, Features aus 5 Intervallen	59
A.25	Konfusionmatrix HMM, Fenstergröße 64, Features aus 7 Intervallen	59
A.26	Konfusionmatrix HMM, Fenstergröße 32, Features aus 3 Intervallen	59
A.27	Konfusionmatrix HMM, Fenstergröße 32, Features aus 5 Intervallen	60
A.28	Konfusionmatrix HMM, Fenstergröße 32, Features aus 7 Intervallen	60
A.29	Konfusionmatrix HMM, Neuster Zyklus, Features aus 3 Intervallen	60
A.30	Konfusionmatrix HMM, Neuster Zyklus, Features aus 5 Intervallen	61
A.31	Konfusionmatrix HMM, Neuster Zyklus, Features aus 7 Intervallen	61
A.32	Konfusionmatrix HMM, Fenstergröße 32, Rohdaten	61
A.33	Konfusionmatrix HMM, Fenstergröße 64, Rohdaten	62
A.34	Konfusionmatrix HMM, Fenstergröße 128, Rohdaten	62
A.35	Konfusionmatrix HMM, Fenstergröße 256, Rohdaten	62

A.36 Konfusionmatrix HMM, Fenstergröße 512, Rohdaten	63
A.37 Konfusionmatrix HMM, Neuster Zylus, Rohdaten	63

1 Einleitung

Systeme zur Erkennung von Aktivitäten bieten viele Anwendungsmöglichkeiten, beispielsweise im Gesundheitswesen zur Überwachung von Patienten oder Navigation innerhalb von Gebäuden. Beschleunigungssensoren werden häufig für diese Aufgabe verwendet, da die entsprechenden Sensormodule heute klein, preiswert und weit verfügbar sind. Viele Mobiltelefone sind heute mit entsprechenden Sensormodulen ausgestattet und bieten außerdem oft weitere Sensoren, wie Magnetfeldsensoren, Gyroskope und GPS. Sie haben zudem eine ausreichend hohe Rechenleistung, um die Erkennung direkt auf dem Gerät durchzuführen. Die Verwendung dieser Hardware bietet sich also an.

Bisherige Arbeiten

Zur Bewegungserkennung mit Beschleunigungssensoren existieren bereits zahlreiche Arbeiten. Viele verwenden spezielle Hardware und Befestigungssysteme, wie zum Beispiel mehrere am Körper verteilte Beschleunigungssensoren [2] [24]. Andere setzen zwar nur einen Sensor ein, der aber zwingend an einer bestimmten Position am Körper befestigt werden muss und die eigentliche Erkennung auf einem zusätzlichen Gerät, z.B. einem PC durchführt [6] [8] [15].

Najafi et al. [15] haben ein System zur Aktivitäten-Erkennung für ältere Menschen entwickelt. Es wird ein auf der Brust befestigtes Sensormodul, bestehend aus einem Beschleunigungssensor und einem Gyroskop genutzt. Die Daten werden auf einem zusätzlichen Gerät aufgezeichnet und die Erkennung später am Computer durchgeführt. Es werden die Aktivitäten *Stehen*, *Liegen*, *Sitzen* und *Gehen*, sowie Übergänge zwischen diesen erkannt. Dazu wird eine Wavelet Transformation und ein komplexes, von Hand erstelltes Zustandsübergangsmodell verwendet, mit dessen Hilfe Erkennungsraten von über 90% erreicht werden.

Karantonis et. al. [8] verwenden ein Modul befestigt an einem Gurt, der um die Hüfte getragen wird. Das Modul enthält einen 3-achsen Beschleunigungssensor und einen Mikrocontroller für die Weiterverarbeitung. Aufgrund der begrenzten Leistung des Mikrocontrollers wird ein Zeitraum von einer Sekunde betrachtet und wie in [15] ein per Hand erstelltes Zustandsübergangsmodell genutzt. Die untersuchten Aktivitäten sind *Stehen*, *Gehen*, *Rennen*, *Hinfallen* und *Liegen* in verschiedenen Positionen. Für die Erkennung von periodischen Aktivitäten, wie *Gehen*, *Rennen* werden die Koeffizienten der Fouriertransformierten des Beschleunigungsverlaufs herangezogen. Für die Erkennung von *Hinfallen* und *Liegen* werden dagegen Schwellenwerte für die Beschleunigung und den Neigungswinkel des Gerätes verwendet. Damit werden Erkennungsraten von durchschnittlich 90% erreicht.

1 Einleitung

In [6] wird ein einzelnes, in der vorderen Hosentasche getragenes Sensormodul zur Datenakquisition verwendet, welches Beschleunigungsdaten mit Hilfe von Bluetooth an einen PC überträgt. Die betrachteten Aktivitäten *Gehen*, *Stehen*, *Springen* und *Rennen* wurden in Fenster von etwa fünf Sekunden Länge eingeteilt. Um die Abhängigkeit der Daten von der Position des Beines zu reduzieren, wurde die Norm jedes Datenpunktes bestimmt. Aus jedem Fenster wurden anschließend die Koeffizienten der diskreten Kosinustransformation bestimmt. Da der Sensor mit einer Samplingrate von 100 Hz arbeitet erhält man aus jedem der Fenster 512 Koeffizienten. Um die Dimensionalität der Daten zu reduzieren, wurde daher eine Principal Component Analysis durchgeführt und die ersten 48 Koeffizienten als Merkmale für die Klassifikation verwendet. Als Klassifikator werden Support Vektor Maschinen genutzt und damit eine durchschnittliche Erkennungsrate von 97.51 % erreicht, wobei *Springen* am schlechtesten abgeschnitten hat und häufig als *Rennen* klassifiziert wurde.

Olguín et al. [16] werden mehrere Beschleunigungssensoren, die an Handgelenk, Hüfte und Brustkorb angebracht werden. Dabei werden verschiedene Konfiguration getestet, z.B. je ein Sensor an Hüfte und Handgelenk oder ein Sensor an jeder der drei Positionen. Es wurden Daten für die Aktivitäten *Sitzen*, *Rennen*, *Kauern*, *Gehen*, *Stehen*, *Krabbeln*, *Liegen* und *Handbewegungen* (im Stehen) gesammelt. Die Daten wurden anschließend in Zeiträume von einer Sekunde eingeteilt. Anschließend werden Mittelwert und Standardabweichung aus den Zeiträumen bestimmt und als Merkmale für Hidden Markov Modelle (HMM) verwendet. Mit Hilfe von je einem Sensor an Handgelenk, Hüfte und Brustkorb wurde eine durchschnittliche Erkennungsrate von 92.13% erreicht. Bei Nutzung eines einzelnen Sensors wurde eine Erkennungsrate von durchschnittlich 62.45% erreicht, wobei *Rennen* hier mit über 90% sehr gut abschnitt.

Die bisher beschriebenen Arbeiten haben alle spezielle Sensormodule verwendet. Sie sind an sich zwar preiswert, aber für die Weiterverarbeitung der Sensordaten auf zusätzliche Hardware angewiesen. Zudem sind solche System unbequem und für den täglichen Gebrauch daher weniger gut geeignet. Hier bieten sich Mobiltelefone an, da sie ohnehin meist mitgeführt werden und kaum behindern. Arbeiten, die Mobiltelefone verwenden sind jedoch selten und die Forschung in diesem Bereich noch nicht sehr fortgeschritten.

In [4] wird ein externer Sensor verwendet, der ähnlich einer Armbanduhr um das Handgelenk getragen wird. Die Verarbeitung der Sensordaten wird jedoch auf einem Smartphone durchgeführt. Es werden die Aktivitäten *Ruhezustand*, *Tippen*, *Gestikulieren*, *Gehen*, *Rennen* und *Fahrrad fahren* über Zeiträume von 2 Sekunden untersucht. Als Merkmal wird die Intensität der Beschleunigung verwendet. Als Klassifikatoren werden Entscheidungsbäume und Neuronale Netze verwendet, wobei letztere in der Praxis bessere Ergebnisse lieferten. Mit den Neuronalen Netzen werden Erkennungsraten von 82% erreicht.

Reddy et al. [22] verwenden ein Mobiltelefon zur Bestimmung der Fortbewegungsweise (*Stillstand*, *Rennen*, *Gehen*, *Fahrrad fahren* und *Motorisiert*) des Benutzers. Das Gerät enthält einen Beschleunigungssensor, sowie einen GPS Empfänger, die beide zur Erkennung verwendet werden. Es werden Beschleunigungsverläufe über einem Fenster von einer Sekunde betrachtet. Aus dem Fenster wurde unter anderem der Mittelwert, Standardabweichung und die Energie extrahiert. Da allein mit den Beschleunigungsdaten kaum Unterschiede zwischen den beiden Aktivitäten *Stillstand* und *Motorisiert* ausgemacht werden können, wird

der GPS Empfänger zur Unterscheidung der beiden Aktivitäten zur Hilfe genommen. Die Arbeit testet verschiedene Klassifikatoren und die besten Ergebnisse liefert ein zweistufiges Verfahren bestehend aus einem Entscheidungsbaum und Hidden Markov Modellen, welche als Eingaben die Ausgabewahrscheinlichkeiten des Entscheidungsbaums verwendet. Als Merkmale für die Entscheidungsbäume dienen Varianz, Energie und Summe der Fourierkoeffizienten, außerdem die Geschwindigkeit, die mit Hilfe des GPS Empfängers bestimmt wird. In dieser Arbeit werden Erkennungsraten von 98% erreicht.

In [25] kommt ein Mobiltelefon zum Einsatz, welches mit einem Beschleunigungssensor ausgestattet ist. Die untersuchten Aktivitäten sind *Sitzen*, *Stehen*, *Gehen*, *Rennen*, *Fahrrad fahren* und *Auto fahren*. Für jede Aktivität wurde ein Fenster von 10 Sekunden betrachtet. Die Besonderheit an dieser Arbeit ist, dass eine Transformation der Sensordaten auf ein Körperfestes Koordinatensystem durchgeführt wird. Andere Arbeiten verwenden statt einer Transformation lediglich die Norm der Beschleunigungsvektors, wodurch Richtungsinformationen verloren gehen. Aus den 10 Sekunden Fenstern werden folgende Merkmale extrahiert: Mittelwert, Standardabweichung, Anzahl der Nulldurchgänge, 75% Perzentile und Quartilabstand. Diese werden schließlich einem Entscheidungsbaum übergeben und damit Erkennungsraten von durchschnittlich 90% erreicht.

Kwapisz et al. [11] verwenden ebenfalls den Beschleunigungssensor eines Mobiltelefons um die Aktivitäten *Gehen*, *Laufen*, *Sitzen*, *Stehen*, *Treppen hinaufsteigen* und *Treppen hinuntersteigen* zu erkennen. Aus Zeiträumen von 10 Sekunden werden unter anderem Mittelwert, Standardabweichung und Abstand zwischen den Spitzen verwendet. Anschließend werden die Merkmale mit Hilfe von Entscheidungsbäumen, Logistischer Regression und Neuronalen Netzen klassifiziert, wobei die Neuronalen Netze mit durchschnittlich 91.70% die beste Gesamterkennungsrate aufweisen. *Treppen hinaufsteigen* und *Treppen hinuntersteigen* weisen dabei mit 55%, bzw. 61.5% die schlechteste Erkennungsraten auf, alle anderen Aktivitäten liegen über 90%.

Ziele

In den meisten Arbeiten, die einzelne Sensoren verwenden wurden recht lange Zeitfenster verwendet, meistens 10 Sekunden oder 5 Sekunden. In dieser Arbeit soll untersucht werden wie genau sich eine Wahl von kürzeren Zeitfenstern auswirkt und bis zu welcher Fenstergröße noch gute Erkennungsraten erreicht werden können. Außerdem wurden bisher kaum betrachtete Aktivitäten untersucht: *Treppen hochsteigen*, *Treppen hinuntersteigen*, *Aufzug fahren* und *Rolltreppe fahren*.

Es werden zwei Klassifikationsverfahren gegenübergestellt: Entscheidungsbäume und Hidden Markov Modelle. Entscheidungsbäume wurden in den vorhergehenden Arbeiten häufig verwendet, da sie gute Erkennungsraten bei wenig Rechenaufwand bieten. Da sich die verwendete Hardware und Datensätze natürlich von anderen Arbeiten unterscheiden, wurden hier Entscheidungsbäume als Referenzimplementierung gewählt, um eine bessere Vergleichbarkeit zu gewährleisten. Hidden Markov Modelle wurden nur in einige wenigen Arbeiten zur Klassifikation von Bewegungen verwendet. Da hier beide Verfahren auf dem selben

Datensatz getestet werden, lassen sich die Ergebnisse direkt vergleichen. Weiterhin wird das untersuchte Zeitfenster für die Hidden Markov Modelle nochmals in Intervalle eingeteilt. Dieses Verfahren wurde bereits in [5] angewendet, allerdings wurden nur mäßige Erkennungsraten von maximal 78% erreicht. Bisher wurde kaum untersucht welche Einteilung sich als am Besten erweist, deshalb werden verschiedene Verfahren auf ihre Erkennungsraten hin untersucht.

Gliederung

Diese Arbeit ist in folgender Weise gegliedert:

Kapitel 2 – Technische Grundlagen bietet eine Übersicht die verwendete Hardware, insbesondere die Eigenschaften und der Zugriff auf die Sensoren wird erläutert. Außerdem wird kurz auf die Entwicklungsumgebungen eingegangen, unter denen die Software zur Datenakquisition und Klassifikation entwickelt wurde.

Kapitel 3 – Datenakquisition widmet sich den in dieser Arbeit betrachteten Aktivitäten, den gesammelten Daten, sowie der für die Datenakquisition entwickelten Software.

Kapitel 4 – Datenaufbereitung geht auf die Aufbereitung der Daten ein. Hier wird auf die Einteilung der gesammelten Daten in Fenster und auch auf die verwendete Koordinatentransformation eingegangen.

Kapitel 5 – Entscheidungsbäume geht auf eines der verwendeten Klassifikationsverfahren ein, den Entscheidungsbäume. Diese dienen in dieser Arbeit als Basis, um die Erkennungsraten der in Kapitel 6 betrachteten Hidden Markov Modelle besser bewerten zu können und mit anderen Arbeiten zu vergleichen.

Kapitel 6 – Hidden Markov Modelle behandelt schließlich Grundlagen und Klassifikationsergebnisse der Hidden Markov Modelle. Es werden verschiedene Eingaben für die HMM betrachtet und miteinander verglichen.

Kapitel 7 – Diskussion bewertet und vergleicht die beiden verwendeten Klassifikationsverfahren.

2 Technische Grundlagen

2.1 Hardware

Zur Datenaufnahme wurde das Smartphone T-Mobile G1 (auch als HTC Dream vermarktet) verwendet. In Tabelle 2.1 sind die technischen Details des Gerätes aufgelistet [7].

Im Gerät ist ein Sensorchip vom Typ Asahi Kasei AK8976A verbaut. Er bietet einen 3-Achsen Beschleunigungssensor, einen 3-Achsen Magnetfeldsensor, sowie einen Temperatursensor [9].

2.1.1 Beschleunigungssensor

Ein Beschleunigungssensor misst die Abweichung der aktuellen Beschleunigung von der Beschleunigung im Freien Fall. Das bedeutet, dass ein Beschleunigungssensor der sich im Freien Fall befindet stets eine Beschleunigung von 0 gemessen wird, obwohl die Erdbeschleunigung auf ihn einwirkt. Wird der Sensor dagegen auf flach einem Tisch platziert, gibt er eine Beschleunigung von $1,0 \text{ G}$ ¹ nach oben an (siehe Abbildung 2.1)

Prozessor	Qualcomm®MSM7201A™, 528 MHz
Betriebssystem	Android™(Version 1.6)
Speicher	192 MB RAM 256 MB ROM Erweiterbar durch Speicherkarte
Schnittstellen	Bluetooth 2.0 IEEE 802.11 b/g WLAN Mini USB Anschluss
Sensoren	3-Achsen Beschleunigungssensor 3-Achsen Magnetfeldsensor Temperatursensor
Sonstiges	GPS-Empfänger

Tabelle 2.1: T-Mobile G1 Technische Daten

¹Gravitation, bzw. Erdbeschleunigung (in SI Einheiten 9.81 m/s^2)

Beschleunigungssensoren existieren in verschiedenen Bauarten. Der im T-Mobile G1 eingesetzte Sensor ist ein piezoresistiver Sensor [9]. Er macht sich den sogenannten piezoresistiven Effekt² zu Nutze. Damit kann die Wirkung der Beschleunigung auf eine Masse in eine Widerstandsänderung umgesetzt werden, welche sich anschließend messen lässt.

2.1.2 Magnetfeldsensor

Es steht ein Magnetfeldsensor zur Verfügung, der die Stärke des Magnetfeldes auf den drei Achsen in Millitesla liefert. Die Android API stellt Funktionen zur Verfügung, um aus Magnetfeld- und Beschleunigungsdaten eine Rotationsmatrix zu berechnen, die die Gerätekordinaten auf Weltkoordinaten transformiert. Die Weltkoordinaten entsprechen:

x-Achse Richtung Norden

y-Achse Richtung Westen

z-Achse Richtung Himmel

Dies funktioniert allerdings nur, solange das Gerät im Stillstand ist. Die Transformation der Beschleunigung auf Weltkoordinaten ist mit dem von der Android API zur Verfügung gestellten Verfahren nicht möglich, denn die Beschleunigung wird stets auf $(0, 0, |a|)$ transformiert, wobei $|a|$ die Norm des Beschleunigungsvektors ist [17].

2.2 Android

Auf dem T-Mobile G1 ist das Betriebssystem Android installiert. Als Sprache für die Entwicklung von Anwendungen für Android dient Java. Es werden zahlreiche Entwicklungswerkzeuge zur Verfügung gestellt, darunter Emulatoren und ein Plugin für die integrierte Entwicklungsumgebung Eclipse. Daher wurde die Anwendung zur Datenakquisition vollständig unter Eclipse entwickelt.

2.2.1 Sensoren Schnittstelle

Die Kommunikation mit den Sensoren erfolgt ereignisorientiert. Die Android API bietet dazu eine Klasse *SensorManager* an. Nachdem ein Objekt der Klasse instanziiert und mit seiner Hilfe die Anwendung als Listener registriert wurde, liefert das Betriebssystem Sensorwerte an die Anwendung. Diese bestehen aus einem Zeitstempel, sowie den eigentlichen Werten.

Sobald das Gerät in den Standby Modus wechselt und das Display abschaltet, werden keine Sensordaten mehr geliefert. Das Umschalten in den Standby Modus lässt sich mit Hilfe eines sogenannten Wakelocks verhindern, welches die Android API anbietet. Der Nachteil daran ist jedoch, dass das Display weiter auf Benutzereingaben reagiert. Dies schränkt die möglichen Positionen des Gerätes am Körper ein, da versehentlich Benutzereingaben ausgelöst werden

²Widerstandsänderung eines elektrischen Leiters unter Druck oder Zug

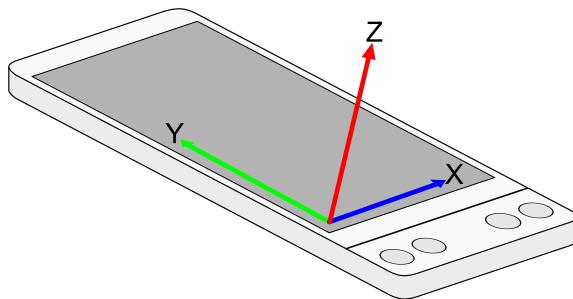


Abbildung 2.1: Das Sensorkoordinatensystem ist abhängig von der Lage des Displays. x-Achse und y-Achse entsprechen den jeweiligen Display Seiten und die z-Achse zeigt gen Himmel, falls das Gerät flach auf dem Boden liegt.

könnten. Außerdem verkürzt sich die Laufzeit des Geräts durch das aktivierte Display und den damit verbundenen erhöhten Stromverbrauch.

Allerdings lässt sich trotz des aktivierten Displays eine Laufzeit von über einer Stunde erreichen, was für diese Arbeit ausreichend ist. Mit voranschreitender Entwicklung wird es auf irgendeine Art möglich sein, die Sensordaten auch bei abgeschalteten Displays auszulesen. Die Einschränkungen für diese Arbeit sind also vernachlässigbar und werden für zukünftige Entwicklungen voraussichtlich nicht mehr gegeben sein.

2.2.2 Sensorkoordinatensystem

Das Sensorkoordinatensystem ist abhängig von der Ausrichtung des Displays. Im Porträtmodus entspricht die x-Achse der kurzen Displayseite, die y-Achse der langen Displayseite. Im Landschaftsmodus entspricht die x-Achse der langen Seite, die y-Achse der kurzen Seite. Die z-Achse ist stets orthogonal zu x und y-Achse und zeigt in Richtung Himmel, falls das Gerät flach auf dem Boden liegt [17].

Durch Drehung des Gerätes wird zwischen den beiden Displaymodi gewechselt. Dabei ändert sich aber zugleich das Koordinatensystem. Um dies zu vermeiden wird der automatische Wechsel des Displaymodus deaktiviert. Das Display befindet sich somit ständig im Porträtmodus. Das verwendete Sensorkoordinatensystem ist in Abbildung 2.1 nochmals dargestellt.

2.2.3 Samplingrate

Die Samplingrate lässt sich mit Hilfe der in Tabelle 2.2 angegebenen Konstanten beeinflussen. Eine genaue Festlegung der Samplingrate ist jedoch nicht möglich, denn die Konstanten legen lediglich den Mindestabstand zwischen je 2 aufeinanderfolgenden Sensorwerten fest, die Werte können jedoch auch mit größerem Abstand geliefert werden.

Name	Mindestabstand in Millisekunden
SENSOR_DELAY_NORMAL	200
SENSOR_DELAY_UI	60
SENSOR_DELAY_GAME	20
SENSOR_DELAY_FASTEST	0

Tabelle 2.2: SENSOR_DELAY_* Konstanten

Genutzt wird die schnellste verfügbare Samplingrate *SENSOR_DELAY_FASTEST*. Sie setzt den Mindestabstand auf 0 Sekunden, das bedeutet die Sensorwerte werden so schnell geliefert, wie möglich. Kurz nach dem Start der Sensoraufnahme pendelt sich der Abstand jedoch auf 20 Millisekunden ein [18]. Nur selten gibt es leichte Schwankungen von einer Millisekunde. Daher kann von einer konstanten Samplingrate von 50 Hz ausgegangen werden.

2.3 Matlab

Zur Implementierung der Weiterverarbeitung und Klassifikation der Sensordaten wurde Matlab verwendet. Matlab ermöglicht eine schnelle Entwicklung aufgrund der zahlreichen bereits existierenden Funktionen, wodurch das Testen verschiedener Verfahren erleichtert wird. Dies stellt natürlich keine große Herausforderung dar. Die Implementierung kann auch direkt auf dem Gerät erfolgen und die Erkennung kann vollständig portiert werden. Die Implementierung auf dem Gerät nimmt jedoch ein Vielfaches der Zeit in Anspruch, weshalb matlab gewählt wurde.

2.3.1 Entscheidungsbäume

Für die Entscheidungsbäume wurde die Statistics Toolbox von Matlab verwendet. Sie ist in der Lage Klassifikations- und Regressionsbäume mit Hilfe des CART Algorithmus und verschiedener Teilungskriterien zu erstellen. In 5 wird genauer auf das verwendete Verfahren eingegangen.

2.3.2 HMM Toolbox

Zum Trainieren und Testen der verschiedenen Hidden Markov Modelle wurde die HMM Toolbox für Matlab verwendet [14]. Sie ist frei verfügbar und steht unter der MIT Lizenz zur Verfügung. Die Toolbox erlaubt das Trainieren von Hidden Markov Modellen mit Gaußschen Mischungen mit Hilfe dem EM Algorithmus. In 6 wird genauer auf das verwendete Verfahren eingegangen.

3 Datenakquisition

3.1 Software

Zur Aufnahme der Daten wurde eine Anwendung für das G1 entwickelt, die es ermöglicht Beschleunigungsdaten über einen beliebig gewählten Zeitraum aufzunehmen. Der Benutzer kann die Datenakquisition manuell starten und wieder stoppen. Die Dauer der Aufnahme kann also vom Benutzer gewählt werden, und wird nur durch den vorhandenen Speicher begrenzt. Die Daten werden auf der Speicherkarte des Gerätes abgelegt und können von dort aus zur Weiterverarbeitung auf einen PC übertragen werden. Natürlich kann die Weiterverarbeitung auch direkt auf dem Smartphone stattfinden, in dieser Arbeit wurde sie jedoch auf dem PC durchgeführt.

Die Anwendung speichert die Sensordaten als Textdatei. Für jedes Sensorereignis wird der Zeitstempel (Die Zeit seit dem Beginn der Aufzeichnung in Millisekunden), sowie die drei Elemente des Beschleunigungsvektors gespeichert. Es stehen verschiedene vorgefertigte Aktivitäten zur Auswahl, für die automatisch ein Dateiname und ein Header erzeugt wird. Alternativ lassen sich diese auch manuell bearbeiten. Der Header enthält den Gerätenamen, den Namen der Aktivität, sowie das Datum und die Uhrzeit der Aufnahmebeginns. Optional kann der Benutzer einen Kommentar eingeben. Dieser wurde genutzt, um die Gerätelocation oder den genauen Ort der Aufnahme zu speichern. Abbildung 3.1 zeigt den Aufbau einer Datei.

3.2 Gerätelocation

Das Gerät wird in einer der vorderen Hosentaschen getragen, da Mobiltelefone häufig an dieser Position getragen werden. Aufgrund der in Kapitel 2 genannten Einschränkungen kann das Gerät nicht mit dem Display in Richtung Oberschenkel getragen werden, denn durch die ständigen Berührungen mit dem Bein werden Benutzereingaben ausgelöst, die die Datenaufnahme stoppen könnten.

Das schränkt die möglichen Positionen pro Hosentasche auf zwei ein. Damit sind insgesamt die vier Positionen nutzbar, die in Abbildung 3.2 gezeigt werden. Untersucht wurden jedoch nur zwei Positionen in der rechten Hosentasche (Abbildung 3.2a und 3.2b). Die Untersuchung weiterer Positionen macht auch das Sammeln zusätzlicher Daten notwendig, was den Zeitrahmen dieser Arbeit sprengen würde. Es wurden jedoch auch einige Datensätze in den anderen beiden Positionen gesammelt, um die Koordinatentransformation (Abschnitt 4.4) zu testen.

```

Device: T-Mobile G1
walk
2010-12-15 13:44
Position A, contains passage downhill & uphill
#
89 -2.6014864 4.903325 7.5865335
128 -2.6014864 5.1212506 7.5865335
149 -2.6014864 5.012288 7.5865335
169 -2.6559677 5.066769 7.5320525
189 -2.7104492 5.1212506 7.4230895
210 -2.7649305 5.1212506 7.3141265
230 -2.819412 5.1212506 7.205164
250 -2.7649305 5.2302136 7.096201
270 -2.7649305 5.339176 7.04172
291 -2.7104492 5.611583 6.8237944
310 -2.547005 6.033814 6.66035
330 -2.6014864 6.033814 6.7148314

```

The diagram illustrates the structure of a sensor data file. It consists of two main sections: a header and a data block. The header is located at the top and includes the device name ('Device: T-Mobile G1'), activity type ('walk'), timestamp ('2010-12-15 13:44'), and a descriptive comment ('Position A, contains passage downhill & uphill #'). The data block follows the header, containing multiple lines of numerical data representing acceleration vectors. Each line in the data block is preceded by a timestamp and followed by three values representing the x, y, and z components of the acceleration vector.

Abbildung 3.1: Aufbau einer durch die Anwendung erzeugten Datei. Die Abbildung zeigt den Header der Datei und den Beginn des Datenblocks.

Bedingt durch die Wahl der Position wird nicht nur die Vorwärtsbeschleunigung und die Gravitation, sondern auch die Beinbewegung des Benutzers durch den Sensor registriert. Die Erkennung kann deshalb nicht auf Sensordaten angewendet werden, die in einer anderen Position aufgenommen wurden, beispielsweise falls sich das Gerät in einem Rucksack befindet.

3.3 Aktivitäten

Die gesammelten Aktivitäten werden aufgeteilt in periodische Aktivitäten, d.h. Aktivitäten die eine gewisse Periodizität aufweisen, wie z.B. *Gehen* und statischen Aktivitäten bei denen sich der Benutzer selbst kaum bewegt, wie z.B. *Rolltreppe fahren*. Die Aktivitäten wurden auf dem Universitätsgelände, d.h. unter realen Bedingungen, von einer einzigen Person gesammelt.

Statischen Aktivitäten:

Aufzug fahren Das Sammeln der Daten für die Aktivität Aufzug fahren ist problematisch, denn moderne Aufzüge verursachen kaum Vibrationen oder Schwankungen, wodurch sich Aufzug fahren kaum von Stehen unterscheiden lässt. Je nach Aufzug sind selbst

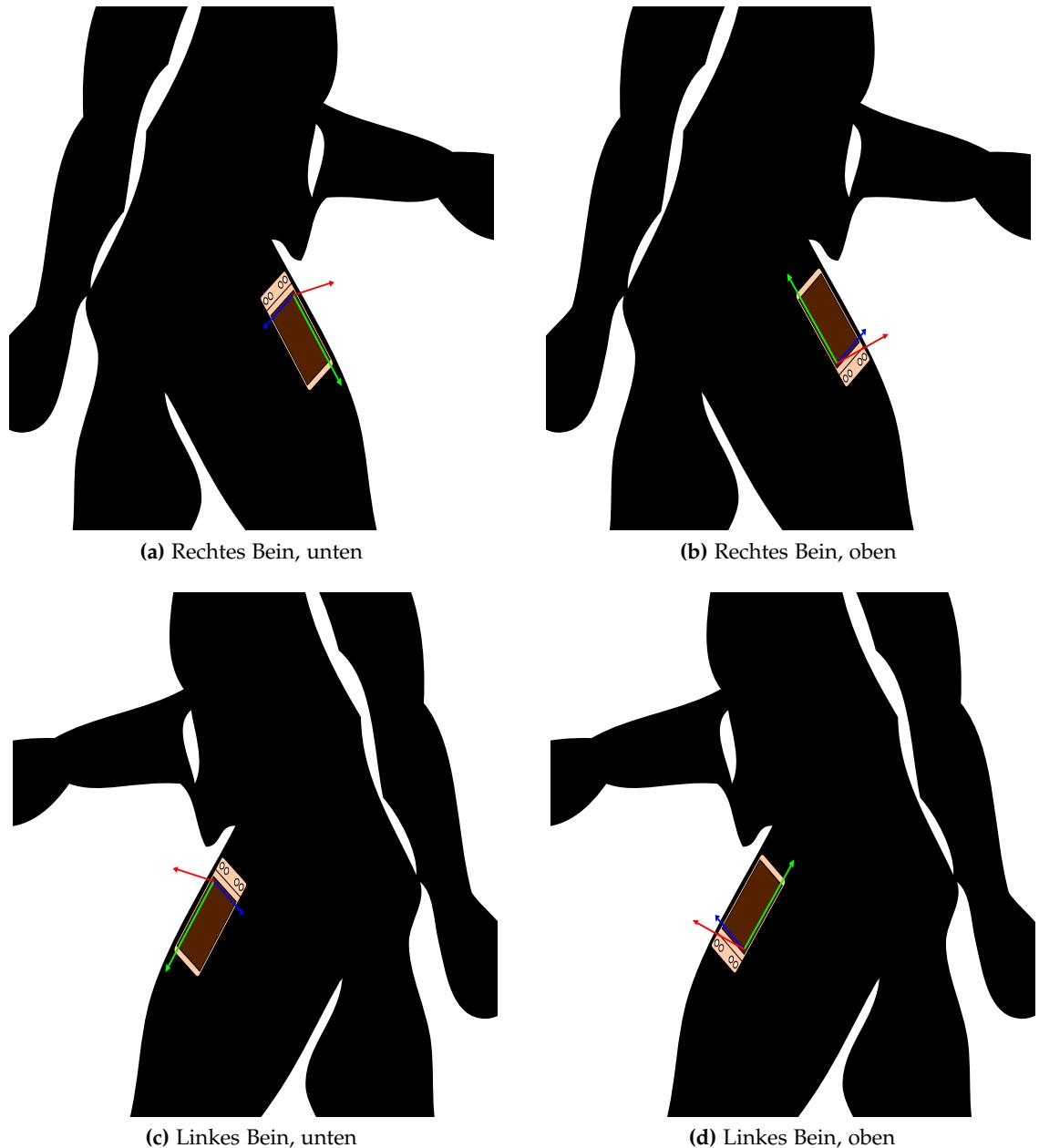


Abbildung 3.2: Gerätepositionen in den Hosentaschen mit zugehörigen Sensorkoordinatensystemen

Beschleunigungs- und Bremsphasen kaum in den Beschleunigungsverläufen zu erkennen.

Rolltreppe fahren Rolltreppendaten wurden zwar an verschiedenen Rolltreppen gesammelt, leider stammen alle von dem gleichen Hersteller, so dass kaum Unterschiede in den Beschleunigungsverläufen existieren. Ein leichtes Rütteln ist dennoch zu beobachten und wie in Kapitel 5 und 6 gezeigt wird es reicht aus, um die Aktivität von *Ruhezustand* und *Aufzug fahren* zu unterscheiden.

Ruhezustand In der Aktivität *Ruhezustand* werden die Aktivitäten Sitzen und Stehen zusammengefasst. Sie unterscheiden sich nur durch eine Drehung (der Drehung der Beins), dieser Unterschied verschwindet bei der späteren Aufbereitung der Daten (Kapitel 4).

Periodische Aktivitäten:

Rennen Wie aus Abbildung 3.6 ersichtlich unterscheidet sich Rennen deutlich von den übrigen Aktivitäten. Frequenz und Amplitude des Beschleunigungsverlaufs übersteigen die der übrigen Aktivitäten deutlich. In Kapitel 5 und 6 wird gezeigt, dass die Erkennungsraten für Rennen entsprechend hoch sind.

Treppen hinuntersteigen und Treppen hinaufsteigen Für die beiden Aktivitäten *Treppen hinaufsteigen* und *Treppen hinuntersteigen* wurden Daten von fünf verschiedenen Treppen aufgenommen. Es wurde darauf geachtet Datensätze von Treppen unterschiedlicher Steigung zu sammeln. Die Beschleunigungsverläufe für flache Treppen, d.h. Treppen mit niedriger Steigung, ähneln dabei den Verläufen für Gehen. Je nach Art der Treppe sind längere Gehphasen zwischen dem eigentlichen Treppendaten vorhanden. Diese werden bei der Aufbereitung entfernt.

Gehen ist von allen Aktivitäten die am stärksten periodische, d.h. die einzelnen Perioden gleichen sich stark und lassen sich in den Beschleunigungsverläufen stets gut erkennen. Die Daten wurden an Stellen mit verschiedenen Untergründen (Asphalt, Schnee, Gras), sowie unterschiedlichen Steigungen gesammelt. Da die Daten unter realen Bedingungen gesammelt wurden, mussten auch Hindernisse umgangen, d.h. die Richtung geändert werden, was zusätzlich Varianz in den Datensatz bringt. Gravierende Unterschiede in den Beschleunigungsverläufen bestehen jedoch nicht.

Einige typische Beschleunigungsverläufe sind in Abbildung 3.3–3.9 aufgezeigt. Zur besseren Vergleichbarkeit wurde für alle Aktivitäten die Position A (Abbildung 3.2a) ausgewählt.

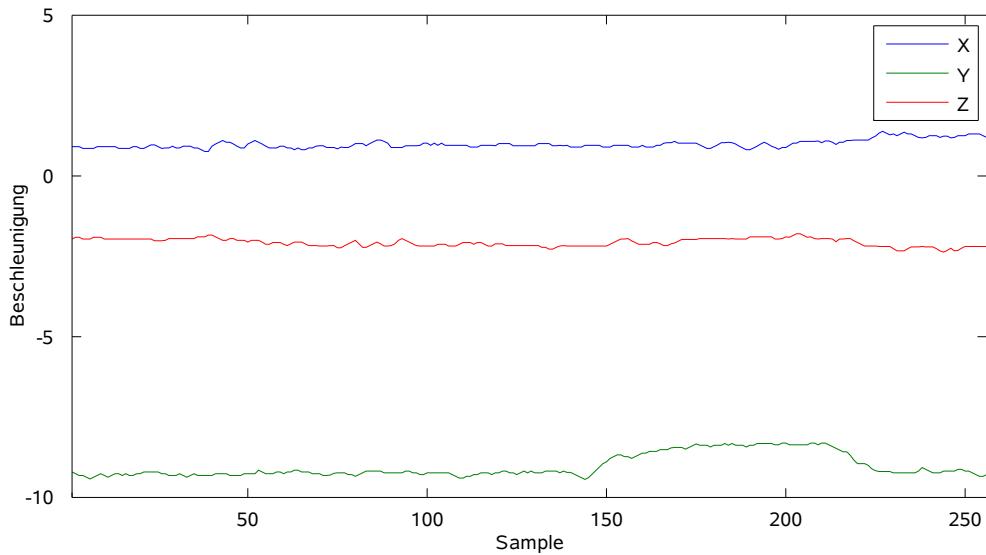


Abbildung 3.3: Beschleunigungsverlauf von *Aufzug fahren*, 256 Datenpunkte, 5 Sekunden. Erkennbar ist ein leichtes "Rütteln", sowie die Bremsphase des Aufzugs zwischen 150-225

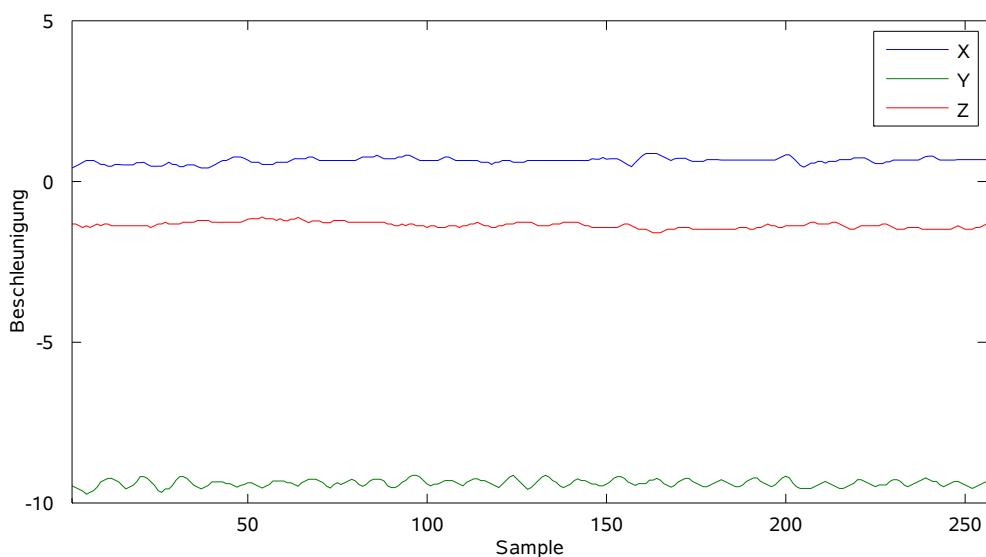


Abbildung 3.4: Beschleunigungsverlauf von *Rolltreppe fahren*, 256 Datenpunkte, 5 Sekunden. Gut erkennbar ist das "Rütteln" der Rolltreppe auf der Y-Achse

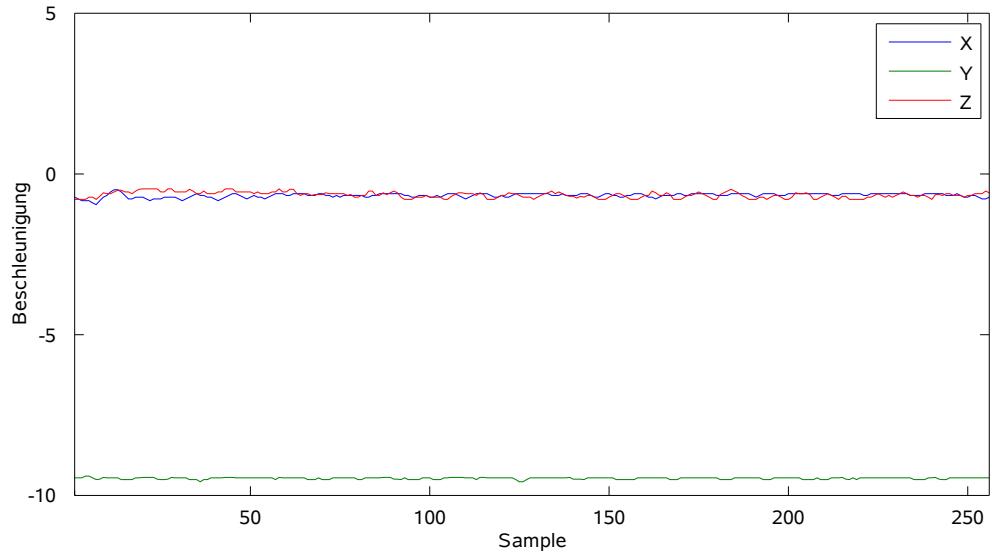


Abbildung 3.5: Beschleunigungsverlauf von *Ruhezustand*, 256 Datenpunkte, 5 Sekunden. In *Ruhezustand* wurden Stehen und Sitzen zusammengefasst. Hier ist Stehen abgebildet.

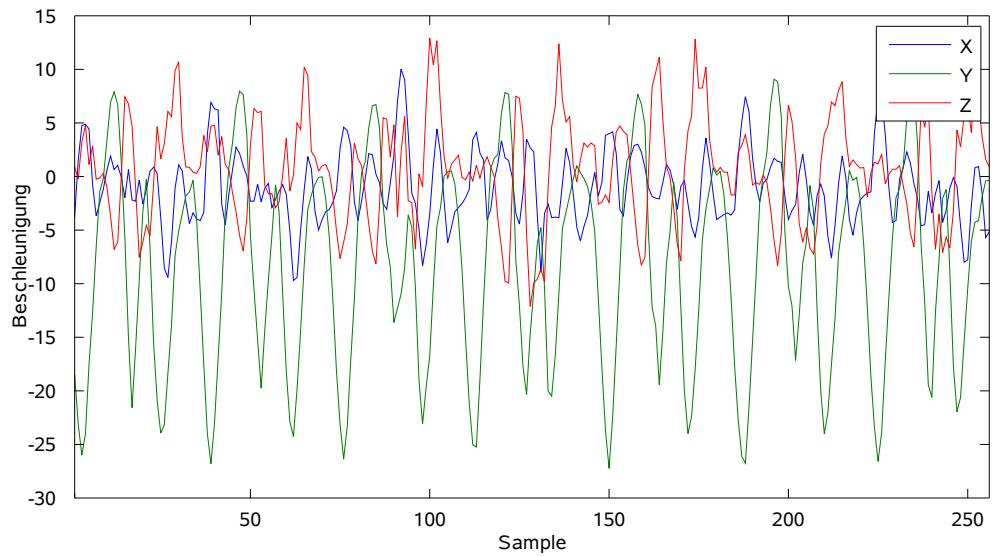


Abbildung 3.6: Beschleunigungsverlauf von *Rennen*, 256 Datenpunkte, 5 Sekunden. Hohe Amplitude und Frequenz im Vergleich zu den übrigen Aktivitäten

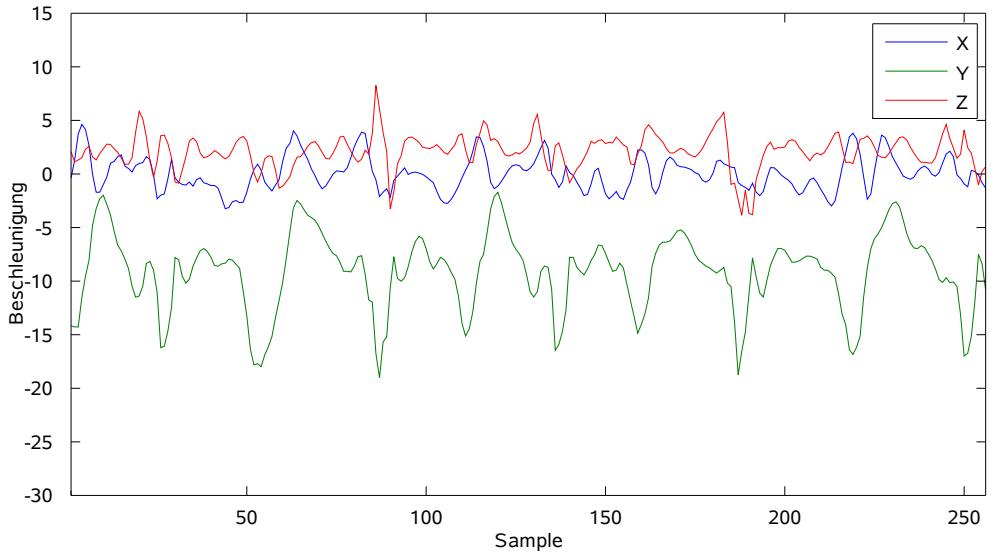


Abbildung 3.7: Beschleunigungsverlauf von *Treppen hinuntersteigen*, 256 Datenpunkte, 5 Sekunden. Diese Aktivität neigt stärker zu Unregelmäßigkeiten, d.h. sie ist weniger periodisch als z.B. *Treppe hinaufsteigen* oder *Gehen*

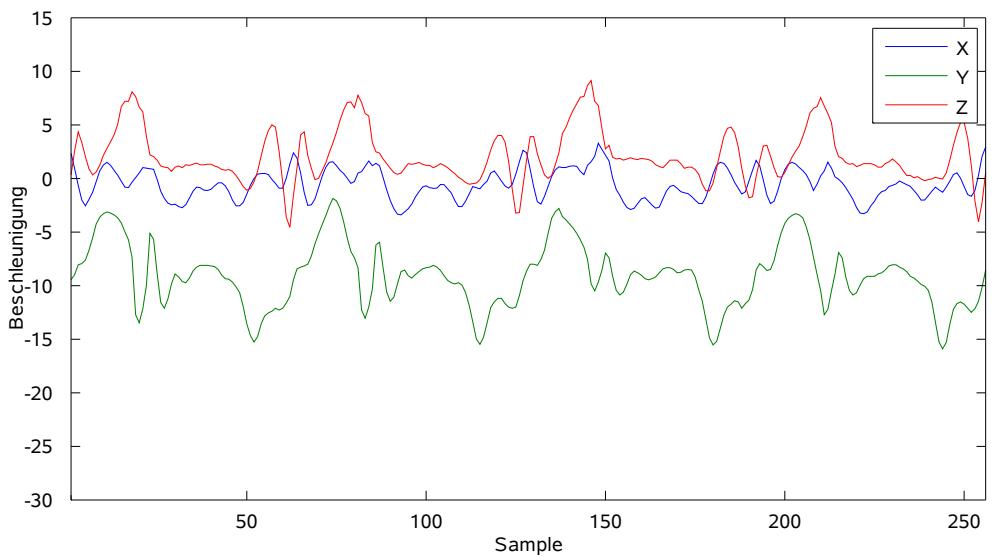


Abbildung 3.8: Beschleunigungsverlauf von *Treppen hinaufsteigen*, 256 Datenpunkte, 5 Sekunden

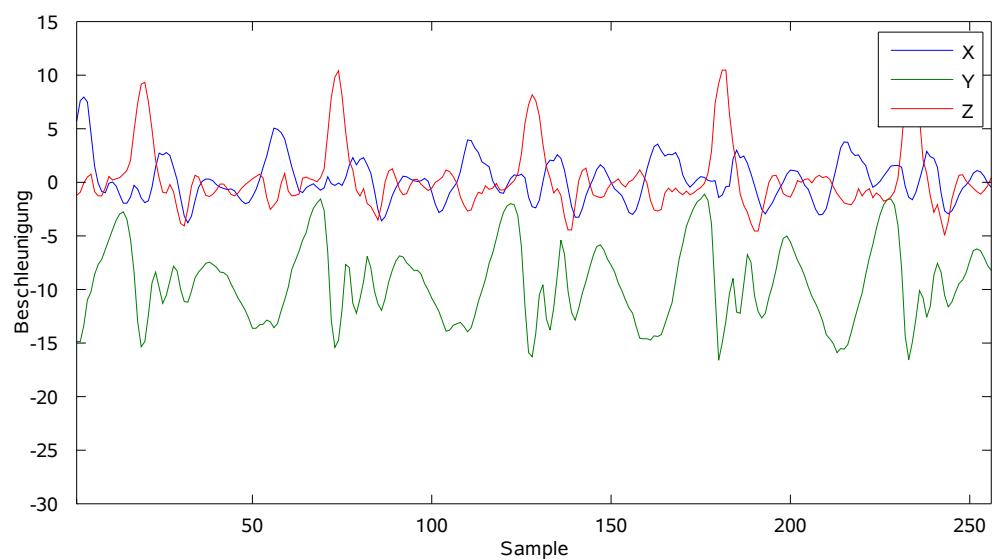


Abbildung 3.9: Beschleunigungsverlauf von *Gehen*, 256 Datenpunkte, 5 Sekunden

4 Datenaufbereitung

4.1 Generierung zusammenhängender Datensätze

Die gesammelten Rohdaten liegen als zahlreiche Einzeldateien vor. Um die Weiterverarbeitung zu vereinfachen wurden die Daten per Hand zusammengefügt, so dass für jede der Aktivitäten ein einziger durchgängiger Datensatz vorliegt. Aus den daraus entstandenen durchgängigen Datensätzen können anschließend beliebige Fenstergrößen generiert werden.

Da der Benutzer das Gerät beim Starten und Beenden der Datenakquisition in der Hand hält, müssen ein Stück am Anfang und am Ende einer Datei entfernt werden. Außerdem werden die Sprünge zwischen den einzelnen Beschleunigungsverläufen manuell entfernt, indem Datenpunkte gelöscht werden, bis die Daten nahtlos zusammenhängen. Das ist möglich, da die Periodizität der Daten ausreichend hoch ist. Zudem müssen Fragmente fremder Aktivitäten entfernt werden. Insbesondere *Treppen steigen* erfordert eine zeitaufwändige Aufbereitung, aufgrund der vielen dazwischenliegenden Gehphasen.

Die nach der manuellen Aufbereitung nutzbaren Datensätze sind in Tabelle 4.1 aufgelistet.

4.2 Glättung

Eine Glättung der Daten gestaltet sich schwierig. Nur für periodische Aktivitäten, wie *Gehen* oder *Rennen* wäre ein Glättung möglich. Würde man jedoch die Aktivität *Rolltreppe* oder *Aufzug fahren* fahren glätten, so wären sie kaum noch von *Ruhezustand* zu unterscheiden. Da

Aktivität	Anzahl Samples	= Minuten
Aufzug fahren	47408	15.8
Rolltreppe fahren	26130	8.7
Ruhezustand	42903	14.3
Rennen	45882	15.3
Treppen hinuntersteigen	43560	14.5
Treppen hochsteigen	63409	21.1
Gehen	155464	51.8

Tabelle 4.1: Anzahl der Daten nach manueller Aufbereitung

Fenstergröße (Samples)	Zeitraum (Sekunden)
32	0.64
64	1.28
128	2.56
256	5.12
512	10.24

Tabelle 4.2: Verwendete Fenstergrößen. Samples entsprechen den angegebenen Zeiträumen

bereits eine leichte Glättung die Unterscheidung stark erschwert, wurde auf eine Glättung komplett verzichtet.

4.3 Einteilung in Fenster

Um Trainings- und Testdatensätze zu gewinnen, werden die zusammenhängenden Rohdaten wieder in Fenster aufgeteilt. Diese Fenster werden zur Merkmalsextraktion genutzt oder direkt zum Trainieren des Klassifikators verwendet. In früheren Arbeiten haben sich Fenster mit 50% Überlappung als vorteilhaft erwiesen. Es wurden verschiedene Fenstergrößen generiert. Da unter anderem die Koeffizienten der Fouriertransformation als Merkmale genutzt werden, wurden stets Fenstergrößen zu Basis zwei verwendet. In Tabelle 4.2 werden die verwendeten Fenstergrößen aufgeführt.

Nach der Generierung, wurde eine zufällige Permutation der Fenster erzeugt, dies ist insbesondere für die Aktivitäten *Treppen hinaufsteigen* und *Treppen hinuntersteigen* wichtig, da Daten von verschiedenen Treppen gesammelt wurden und es vermieden werden soll die Klassifikatoren versehentlich nur auf einer Treppenart zu trainieren.

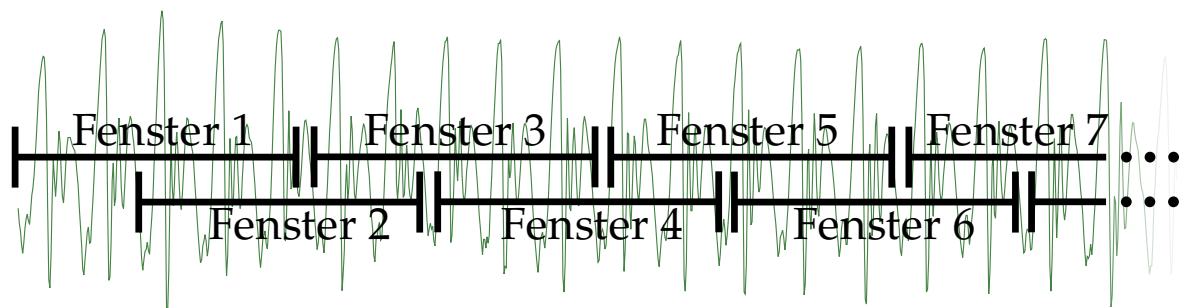


Abbildung 4.1: Die Daten werden in Fenster fester Größe mit 50% Überlappung eingeteilt.

Die Nutzung längerer Fenster erhöht die Erkennungsrate, hat jedoch den Nachteil einer trügerischen Reaktion: Wechselt die Aktivität des Benutzers, beispielsweise von *Gehen* zu *Treppen hinaufsteigen*, so wird das Fenster erst als *Treppen hinaufsteigen* klassifiziert, sobald ein ausreichend großer Anteil der Daten im Fenster die entsprechende Aktivität darstellt. Außerdem

werden kurze Abschnitte anderer Aktivitäten verschluckt, wie zum Beispiel wenige Stufen *Treppen hinaufsteigen* eingebettet in *Gehen*.

4.4 Koordinatentransformation

Aufgrund der Position des Geräts in der Hosentasche, ist das Sensorkoordinatensystem (Abbildung 4.2) durch die Beinbewegung ständigen Schwankungen ausgesetzt.

Um unabhängig von der Position des Geräts vergleichbare Sensordaten zu erhalten, ist es notwendig sie in ein Körperfestes Koordinatensystem zu transformieren. Eine andere Möglichkeit ist die Norm des Beschleunigungsvektors zu verwenden. Dabei gehen jedoch sämtliche Richtungsinformationen verloren.

4.4.1 Exakte Transformation

Eine exakte Transformation, in das in Abbildung 4.2a dargestellte Koordinatensystem wäre wünschenswert, ist mit den vorhandenen Sensoren aber nicht möglich. Für eine exakte Transformation wäre Kenntnis über die genauen Drehwinkel erforderlich, diese Winkel aus den Daten des Beschleunigungssensors zu gewinnen ist jedoch nur möglich, solange ausschließlich die Gravitation auf den Sensor einwirkt. Wirkt eine weitere, d.h. vom Benutzer erzeugte Beschleunigung auf das Gerät ein, kann man nicht zwischen dieser und der Gravitation unterscheiden. Das Verfahren lässt sich also nur im Stillstand anwenden und ist damit für diese Arbeit nicht geeignet. Es muss stattdessen auf eine näherungsweise Transformation zurückgegriffen werden.

Mit dem Aufkommen von Gyroskopen in Smartphones kann diese Einschränkung jedoch umgangen werden. Es kann eingesetzt werden, um die exakten Drehwinkel zu erhalten, mit deren Hilfe sich die Beschleunigungsdaten dann transformieren lassen. Leider stand für diese Arbeit kein Smartphone mit Gyroskop zur Verfügung, die ersten Geräte mit integriertem Gyroskop kamen erst gegen Ende dieser Arbeit auf den Markt.

4.4.2 Näherungsweise Transformation

Die näherungsweise Transformation, bzw. Projektion der Beschleunigungsdaten erfolgt mit dem in [13] vorgeschlagenen Verfahren. Dabei wird ein Beschleunigungsvektor zerlegt in eine vertikale Komponente parallel- und eine horizontale Komponente orthogonal zur Gravitation.

Die Grundidee besteht darin, die Gravitation mit Hilfe des Mittelwertes abzuschätzen. Dies ist möglich, da die Gravitation konstant ist, die Beschleunigung durch die Bewegungen des Benutzers aber schwankt. Um eine gute Abschätzung der Gravitation zu erhalten, muss das Zeitfenster über dem der Mittelwert gebildet wird jedoch ausreichend lang sein [13].

4 Datenaufbereitung

Für die vorliegenden Aktivitäten hat sich eine Fenstergröße von 64 Samples (etwa eine Sekunde) als noch ausreichend erwiesen. Bei einer Fenstergröße von 32 treten bereits Schwankungen auf, d.h. die mit Hilfe des Mittelwerts abgeschätzte Gravitation schwankt von Fenster zu Fenster.

Es wird zunächst der Mittelwert $\vec{m} = (m_x, m_y, m_z)$ der Beschleunigung $\vec{a}(t) = (a_x, a_y, a_z)$ für jede Achse, über das gewählte Zeitfenster $[t_1, t_n]$ der Länge n gebildet:

$$\vec{m} = \frac{1}{n} * \sum_{t=t_1}^{t_n} \vec{a}(t)$$

Dieser Mittelwertvektor \vec{m} weist in Richtung der Gravitation, d.h. der vertikalen Achse (Abbildung 4.2b) Anschließend lässt sich die vertikale Komponente \vec{a}_v der Beschleunigung bestimmen, indem die einzelnen Datenpunkte \vec{a} auf \vec{m} projiziert werden:

$$\vec{a}_v = \left(\frac{\vec{a} \cdot \vec{m}}{\vec{m} \cdot \vec{m}} \right) * \vec{m}$$

Die horizontale Komponente \vec{a}_h erhält man mittels:

$$\vec{a}_h = \vec{a} - \vec{a}_v$$

Die einzelnen Komponenten der horizontalen Beschleunigung lassen sich nicht unterscheiden. Es ist nur bekannt, dass sich \vec{h} in der Ebene orthogonal zum abgeschätzten Gravitationsvektor \vec{m} befindet [13].

Dies resultiert im in Abbildung 4.2b dargestellten Koordinatensystem. Die Ergebnisse der Transformation mit zugehörigen Rohdaten sind in Abbildung 4.3 dargestellt, zur leichteren Vergleichbarkeit wurde in allen 4 Positionen die Aktivität *Gehen* ausgewählt. Wie an den Ergebnissen zu erkennen ist, überführt die Transformation die Daten in ein einheitliches Koordinatensystem.

4.5 Zyklus Erkennung

Ein Zyklus entspricht eine Periode im Beschleunigungsverlauf. Bei der Aktivität Gehen wären dies beispielsweise zwei Schritte. Die Erkennung von Zyklen wird auf zwei Arten eingesetzt.

Erzeugung eines zusätzlichen Fensters Mit Hilfe der Zyklen soll zusätzlich zu den in 4.3 generierten Fenstern ein weiteres erzeugt werden. Da sich die Zyklen alle ähneln, reicht die Betrachtung eines Zyklus. Durch die Wahl des neuesten Zyklus kann außerdem die Reaktionsgeschwindigkeit des System verbessert werden.

Unterteilung des Fensters Das Fenster wird für das Training der Hidden Markov Modelle in Intervalle eingeteilt. Zusätzlich zu festen Intervallgrößen, soll das Fenster auch abhängig von den gefundenen Zyklen eingeteilt werden. Auf die verwendeten Unterteilungen wird in Kapitel 6 noch einmal genauer eingegangen.

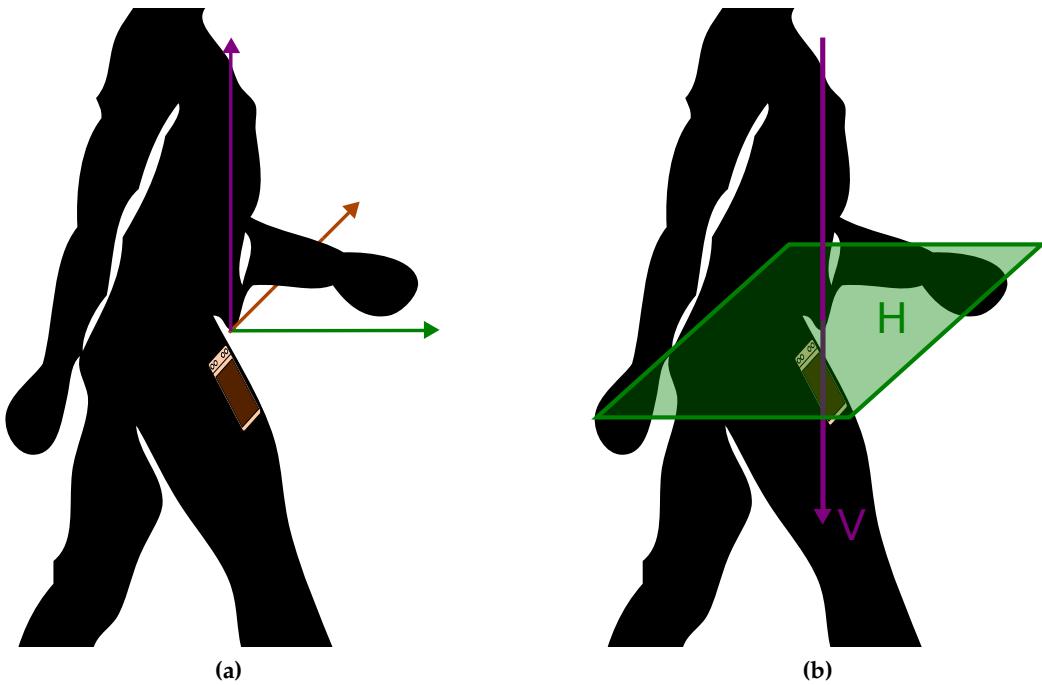


Abbildung 4.2: (a) “Ideales” körperfestes Koordinatensystem, mit den vorliegenden Sensoren nicht verfügbar. Stattdessen wird das durch Abschätzen der Gravitation entstandene Koordinatensystem (b) verwendet

4.5.1 Vorverarbeitung

Um die Zyklen in einem Fenster zu bestimmen muss zunächst die Periodenlänge des Beschleunigungsverlaufs abgeschätzt werden. Um die Abschätzung zu verbessern ist eine Vorverarbeitung notwendig. Dazu wird auf Methoden zurückgegriffen, die auch in der Herzschlagerkennung [12] und der Rhythmus Erkennung bei Musik angewendet werden [19].

Zunächst wird der Beschleunigungsverlauf geglättet. Anschließend wird das Ergebnis quadriert und ein gleitender Mittelwertfilter darauf angewandt, d.h. erneut geglättet. Diese Art der Vorverarbeitung hat sich bewährt und verbessert auch für Beschleunigungsverläufe die Abschätzung der Periodenlänge.

4 Datenaufbereitung

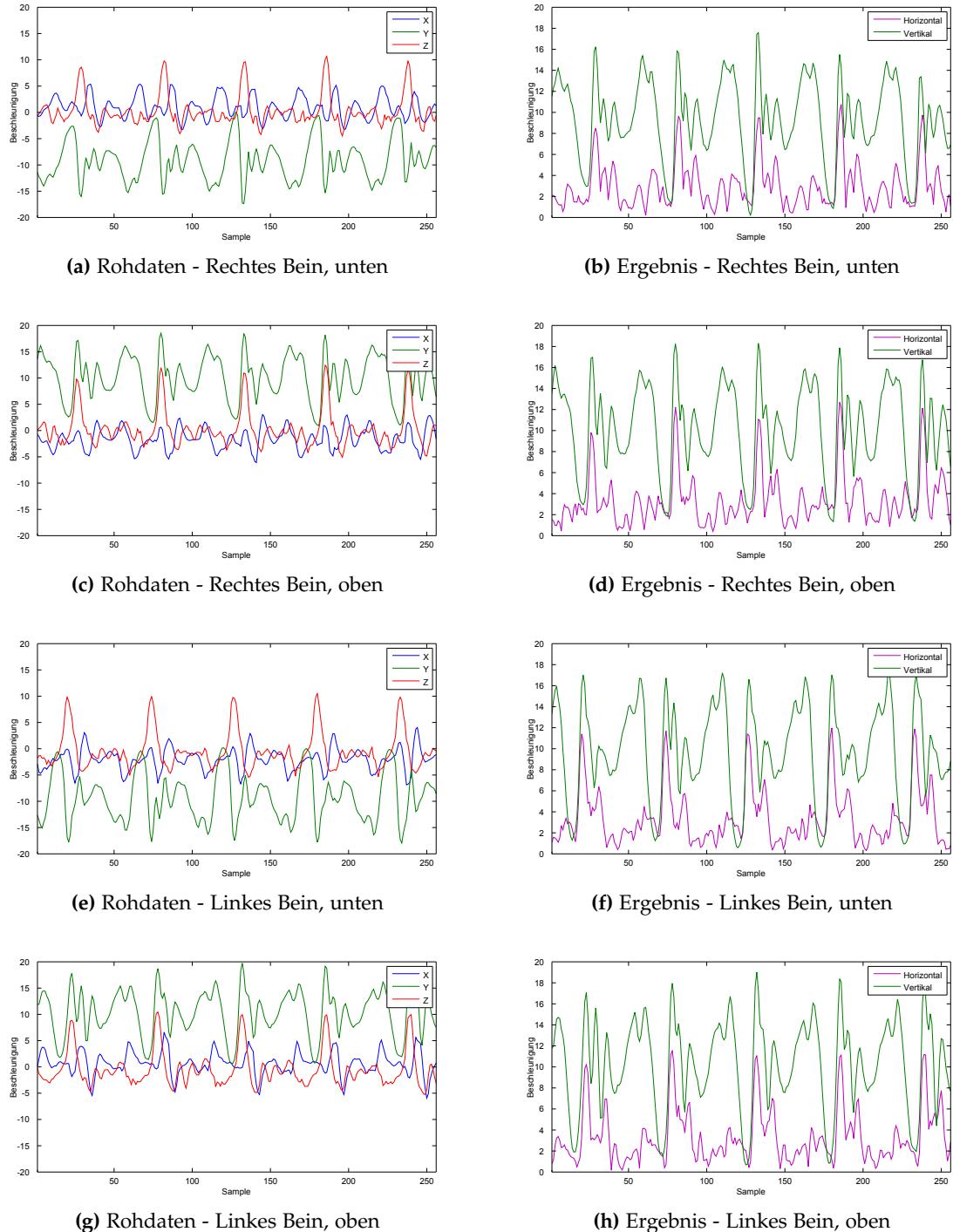


Abbildung 4.3: Ergebnisse der Koordinatentransformation am Beispiel Gehen. Jede Zeile entspricht einer Position aus Abbildung 3.2. Die Beschleunigungsverläufe in der linken Spalte sind im körperfesten Koordinatensystem und gleichen sich unabhängig von der Position des Gerätes

4.5.2 Abschätzen der Periodenlänge

Um die Periodenlänge abzuschätzen wird die Autokorrelationsfunktion verwendet. Sei $x_1 \dots x_n$ eine Folge von Werten. Dann ist die Autokorrelationsfunktion gegeben durch:

$$akf(k) = \sum_{t=1}^{n-k} (x_t * x_{t+k})$$

Nach der Vorverarbeitung, werden die Maxima $M_i = (m_i^x, m_i^y)$ der Autokorrelationsfunktion bestimmt und anschließend sortiert $\mathbf{M} = \{M_1 \dots M_n | m_i^y < m_j^y, i < j\}$. Die Differenz $p = |m_n^x - m_{n-1}^x|$ zwischen den beiden größten Maxima liefert die gewünschte Abschätzung der Periodenlänge. Die Periodenlänge wird sowohl für die vertikale, als auch die horizontale Beschleunigung bestimmt. Experimente haben gezeigt, dass die Verwendung des größeren der beiden Werte die besten Ergebnisse liefert.

4.5.3 Einteilung nach Zyklen

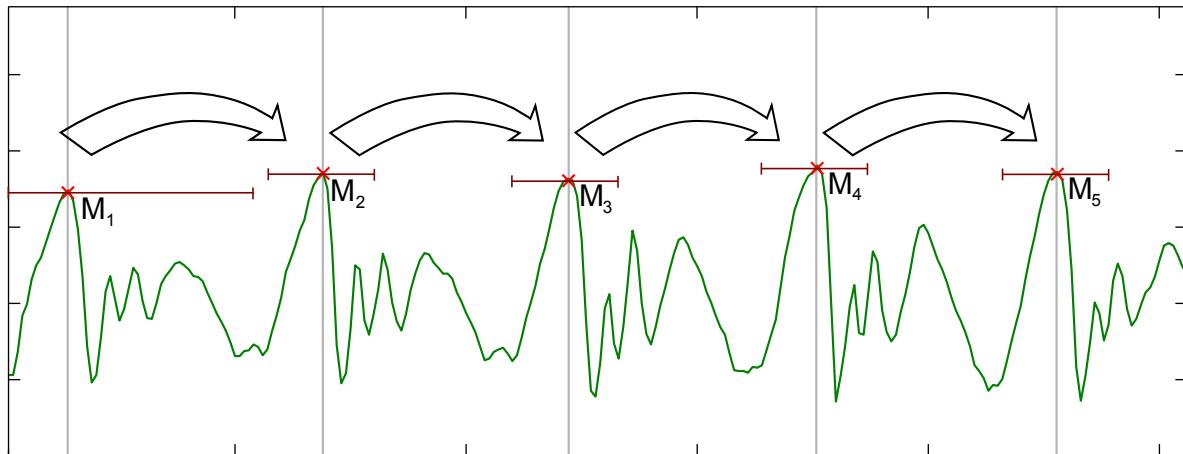


Abbildung 4.4: Suche nach lokalen Maxima zur Unterscheidung der Zyklen. Zunächst wird das globale Maximum M_1 im Intervall $[0, p]$ gesucht, wobei p die abgeschätzte Periodenlänge ist. Ausgehend von diesem wird die Position des nächsten Maximums abgeschätzt und um diesen Bereich das nächste globale Maximum berechnet.

Nachdem die Periodenlänge p nun bekannt ist, wird der Beschleunigungsverlauf in Abschnitte eingeteilt. Jeder Abschnitt entspricht einem Zyklus. Für die Einteilung werden markante Stellen in den Beschleunigungsverläufen benötigt. Dafür eignen sich die lokale Maxima, da sie in allen periodischen Aktivitäten stark ausgeprägt sind. Zunächst wird das Maximum $M_1 = (x_1, y_1)$ innerhalb $[0, p]$ gesucht. Dies ist der Startpunkt des ersten Zyklus. Ausgehend von diesem wird am vermuteten Startpunkt des zweiten Zyklus $[y_1 + 3/4 * p, y_1 + 5/4 * p]$,

nach dem nächsten Maximum M_2 gesucht. Dies wird wiederholt, bis der vermutete nächste Startpunkt zu weit außerhalb des Fensters liegt (Abbildung 4.4).

Die gefundenen Maxima werden schließlich zur Einteilung der Zyklen verwendet. Werden im Fenster keine Maxima gefunden oder entspricht die Periodenlänge dem ganzen Intervall, so wird das ganze Fenster als Zyklus verwendet. Dies ist bei allen nicht-periodischen Aktivitäten - wie zum Beispiel *Ruhezustand* - der Fall.

4.6 Merkmale

Um die Anzahl der zur Klassifikation verwendeten Eigenschaften zu verringern, werden Merkmale aus den Beschleunigungsdaten extrahiert. Sei $X = \{\vec{x}_1, \dots, \vec{x}_n\}$ die Menge der Beschleunigungsdaten. Jedes $\vec{x}_i = (x^h, x^v)$ entspricht dabei einem Datenpunkt bestehend aus der horizontalen Komponente x_h und vertikalen Komponente x_v . Folgende Merkmale wurden jeweils für beide Komponenten verwendet:

Mittelwert:

$$\mu = \frac{1}{n} * \sum_{i=1}^n x_i$$

Standardabweichung: Standardabweichung $\sigma = \sqrt{s^2}$ aus Stichprobenvarianz s^2

$$s^2 = \frac{1}{n-1} * \sum_{i=1}^n (x_i - \mu)^2$$

Fouriertransformation: Koeffizienten \hat{x}_k der diskreten Fouriertransformation einer Folge (x_0, \dots, x_n)

$$\hat{x}_k = \frac{1}{n-1} * \sum_{j=0}^{n-1} e^{-2\pi i * \frac{jk}{n}} x_j$$

5 Entscheidungsbäume

5.1 Grundlagen

Entscheidungsbäume werden zur Klassifikation und zum Data Mining eingesetzt. Jeder Knoten eines Baums entspricht dabei einer Entscheidungsregel und jedes Blatt entspricht einer Klasse. Ein großer Vorteil ist, dass sie die zur Klassifikation verwendeten Merkmale anschaulich darstellen.

5.1.1 CART Algorithmus

Der CART Algorithmus (Classification and Regression Trees) ist ein überwachtes Lernverfahren zur Generierung von Entscheidungsbäumen. Das Verfahren wurde 1984 von L. Breiman veröffentlicht [3].

Begonnen bei der Wurzel - die zunächst die gesamte Menge enthält - wird an jedem Knoten die Menge der Daten in zwei Klassen eingeteilt, wodurch zwei neue Knoten entstehen. Aufgrund der Einteilung in zwei Klassen nur Binäräbäume erzeugt werden. Für die eigentliche Teilung wird dasjenige Attribut ausgewählt, welches die optimale Aufteilung in zwei Teilmengen, hinsichtlich eines Teilungskriteriums liefert.

Da bei einer Teilung stets nur der aktuelle Knoten betrachtet wird, zählt das Verfahren zur Klasse der greedy Algorithmen. Es platziert Attribute umso höher im Entscheidungsbaum, je besser sie die Datenmenge unterteilen.

5.1.2 Teilungskriterium

Als Teilungskriterium wird der Gini Index (oder Gini Koeffizient) verwendet. Er wird als Maß für die Ungleichheit oder Unreinheit einer Menge von Daten genutzt. Es seien die Klassen $J = 1, \dots, k$ gegeben und p_j die relative Häufigkeit der Elemente in der Menge, die zu Klasse j gehören. Dann ist der Gini Index gegeben durch:

$$gi = 1 - \sum_{j=1}^k p_j^2$$

Der Gini Index wird minimal, falls jedes Element einer Menge zu der selben Klasse gehört. Umgekehrt wird der Gini Index maximal, falls jedes Element zu einer eigenen Klasse gehört.

An jedem Knoten soll die Aufteilung zwei möglichst homogene Mengen erzeugen. Dies geschieht, indem die Differenz Δ zwischen der Unreinheit des Elternknotens und der Unreinheit der Kindknoten maximiert:

$$\Delta = gi(t) - n_r * gi(t_r) - n_l * gi(t_l)$$

Wobei gilt:

- $gi(t)$ = Gini Index des Elternknotens
- $gi(t_l)$ = Gini Index des linken Kindknotens
- $gi(t_r)$ = Gini Index des rechten Kindknotens
- n_l = Anzahl der Elemente im linken Kindknoten
- n_r = Anzahl der Elemente im rechten Kindknoten

5.2 Merkmalsextraktion

Um die Anzahl der zur Klassifikation verwendeten Eigenschaften zu verringern, werden Merkmale aus den Beschleunigungsdaten extrahiert. Sei $X = \{\vec{x}_1, \dots, \vec{x}_n\}$ eine Menge von Beschleunigungsdaten, mit $\vec{x}_i = (x_h, x_v)$. Folgende Merkmale wurden jeweils für die horizontale Komponente x_h und die vertikale Komponente x_v eines Datenpunkts berechnet:

Mittelwert:

$$\mu = \frac{1}{n} * \sum_{i=1}^n x_i$$

Standardabweichung: Standardabweichung $\sigma = \sqrt{s^2}$ aus Stichprobenvarianz s^2

$$s^2 = \frac{1}{n-1} * \sum_{i=1}^n (x_i - \mu)^2$$

Fouriertransformation: Koeffizienten \hat{y}_k der diskreten Fouriertransformation einer Folge (y_0, \dots, y_n)

$$\hat{y}_k = \frac{1}{n-1} * \sum_{j=0}^{n-1} e^{-2\pi i * \frac{jk}{n}} y_j$$

Die Merkmale für den Entscheidungsbaum werden über dem gesamten Zeitfenster extrahiert. Die Zyklenerkennung wird genutzt um ein zusätzliches Fenster zu erzeugen. Dazu wurden aus den bestehenden Fenstern der Größe 256 Samples die Zyklen bestimmt und aus dem neuesten Zyklus das neue Fenster erzeugt, dessen Länge schwankt und abhängig von der Länge des Zyklus ist.

Jeweils für die horizontale und vertikale Komponente der Beschleunigung. Dadurch ergeben sich bei einer Fenstergröße von n ein Merkmalsvektor der Dimension $2 * (2 + n)$:

$$\vec{o} = (\mu^h, \mu^v, \sigma^h, \sigma^v, \hat{y}_1^h, \dots, \hat{y}_n^h, \hat{y}_1^v, \dots, \hat{y}_n^v)$$

Der Entscheidungsbaum erhält also mehr Merkmale, als die Daten eigentlich lang sind. Der CART Algorithmus (Abschnitt 5.1.1) wird diejenigen Merkmale auswählen, die die Daten am besten unterteilen, d.h. es werden nicht alle Merkmale auch verwendet.

5.3 10-fold Cross Validation

Die Ergebnisse der Erkennungsraten der Entscheidungsbäume wurden mit Hilfe von 10-fold cross validation geprüft. Für jede Fenstergröße wurde der Datensatz aufgeteilt in zehn Datensätze. Anschließend wurden die Entscheidungsbäume auf neun dieser Datensätze trainiert und mit dem verbleibenden zehnten Datensatz getestet. Dieser Vorgang wurde zehn mal wiederholt, jedes mal mit einem anderen der zehn Datensätze als Testdatensatz. Der Durchschnitt der Erkennungsraten aus den zehn Durchläufen wird schließlich als Endergebnis verwendet. 10-fold cross validation vermindert den Einfluss

5.4 Ergebnisse

Entscheidungsbäume bieten, wie in anderen Untersuchungen zuvor [25] [4], auch hier gute Erkennungsraten und erreichen für große Fenster Erkennungsraten von 93%. Die Einteilung in periodische und statische Aktivitäten spiegelt sich auch in der Struktur des Baums wieder: *Rennen* wird im Entscheidungsbaum meist sofort an der Wurzel zu einem Blatt, danach verzweigt sich der Baum in je einen Unterbaum für statische Aktivitäten und periodische Aktivitäten.

Die periodischen Aktivitäten *Rennen*, *Treppen hochsteigen*, *Treppen hinuntersteigen* und *Gehen* wiesen eine im Durchschnitt 10% höhere Erkennungsrate als die statischen Aktivitäten *textitAufzug fahren*, *Rolltreppe fahren* und *Ruhezustand* auf. Insbesondere *Rennen* weist eine sehr hohe Erkennungsrate auf. Im Gegensatz dazu kommt es zwischen *Ruhezustand* und *Aufzug fahren* häufig zu Fehlklassifikationen, wodurch für *Ruhezustand* die Erkennungsrate bei einer Fenstergröße von 32 auf 39% absinkt. Die Tiefe der Bäume variiert zwischen 9 für Fenstergröße 512 und Tiefe 41 für Fenstergrößen 32. Tabelle 5.1 zeigt die Erkennungsraten der einzelnen Aktivitäten in Abhängigkeit von der gewählten Fenstergröße.

Die Bestimmung des neuesten Zyklus aus einem Zeitfenster von 256 Samples führt zu einer Gesamterkennungsrate von 90.21%. Für periodische Aktivitäten beträgt sie 95.45% und bringt damit eine Steigerung gegenüber der Erkennungsrate von 93.69% für ein Fenster der festen Größe 64 Samples. Für statische Aktivitäten beträgt die Erkennungsrate 83.20%. Da bei statischen Aktivitäten keine Zyklen existieren, wird das Verfahren zur Zyklenerkennung das gesamte Fenster als Zyklus verwenden, d.h. 256 Samples. Deshalb wird in diesem Fall auch

5 Entscheidungsbäume

		Fenstergröße					
		32	64	128	256	512	Neu
Aktivität	Aufzug	82.63	83.25	82.75	82.00	89.00	81.00
	Rolltreppe	90.00	88.50	88.00	85.00	90.00	84.40
	Ruhezustand	39.33	75.75	81.50	89.50	94.00	84.30
	Rennen	99.46	100.00	100.00	100.00	100.00	100.00
	Treppe Runter	85.50	97.38	95.00	97.00	99.00	97.60
	Treppe Hoch	83.25	80.13	95.50	94.50	98.00	91.40
	Gehen	84.08	97.25	94.50	92.00	96.00	92.80
Gesamt		82.29	88.75	91.04	91.43	93.71	90.21

Tabelle 5.1: Erkennungsraten für Entscheidungsbäume, geordnet nach Fenstergröße. "Neu" steht für den Neuesten Zyklus, der aus einem Fenster der Größe 256 bestimmt wurde. *Rennen* weist die besten Erkennungsraten auf, *Aufzug fahren* und *Ruhezustand* die niedrigsten.

mit der festen Fenstergröße von 256 Samples verglichen, welches eine höhere Erkennungsrate (85.75%) aufweist. Dies liegt wahrscheinlich an den zusätzlichen Fehlern, die das Verfahren zur Zyklenerkennung verursacht.

6 Hidden Markov Modelle

6.1 Grundlagen

Hidden Markov Modelle werden unter anderem in der Spracherkennung, für Spamfilter und der Gestenerkennung eingesetzt. Mit Hilfe von Hidden Markov Modellen ist es möglich den zeitlichen Zusammenhang der Daten, bzw. des zugrundeliegenden Systems zu berücksichtigen [21].

Ein Hidden Markov Modell wird definiert als Quintupel $\lambda = \{S, A, Y, B, \pi\}$ mit:

$S = \{s_1, \dots, s_n\}$ Menge der (verborgenen) Zustände

$A = \{a_{ij}\}$ Zustandsübergangsmatrix,

$a_{ij} = \text{Wahrscheinlichkeit von Zustand } s_i \text{ nach Zustand } s_j \text{ zu wechseln}$

$Y = \{y_1, \dots, y_n\}$ Menge der Emissionen (Beobachtungen)

$B = \{b_{ij}\}$ Matrix der Emissionswahrscheinlichkeiten,

$b_{ij} = \text{Wahrscheinlichkeit im Zustand } s_i \text{ die Beobachtung } y_j \text{ zu machen}$

$\pi = \{\pi_i\}$ Menge der Startwahrscheinlichkeiten

$\pi_i = \text{Wahrscheinlichkeit das } s_i \text{ Startzustand ist}$

Ein Hidden Markov Modell ist ein stochastisches Modell, welches durch zwei stochastische Prozesse¹ beschrieben werden kann. Der erste Prozess gegeben durch S und A entspricht

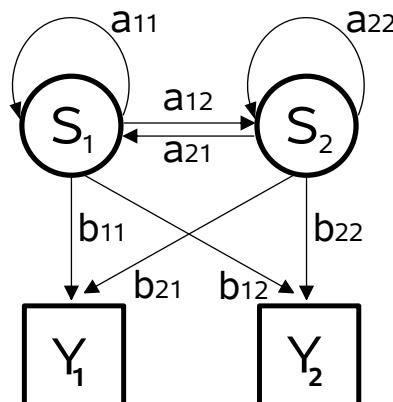


Abbildung 6.1: Beispiel für den Aufbau eines HMM mit zwei versteckten Zuständen s_i , Beobachtungen y_i , Emissionswahrscheinlichkeiten b_i und Übergangswahrscheinlichkeiten a_i

¹eine Folge von Zufallsvariablen $X = \{X_1, \dots, X_n\}$

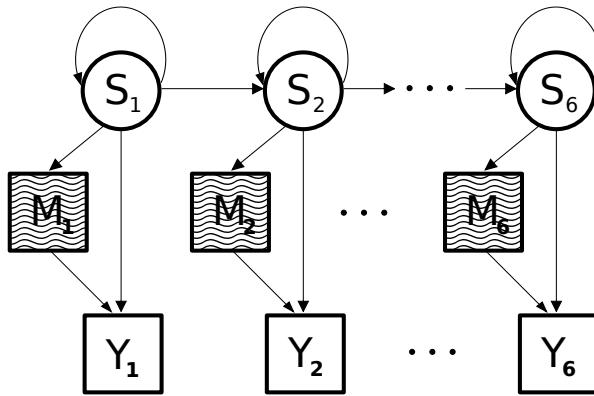


Abbildung 6.2: HMM erster Ordnung mit sechs Zuständen und Gaußschen Mischungen.
Wird verwendet für die Klassifikation mit Rohdaten.

einer Markow-Kette, deren Zustände nicht sichtbar, d.h. versteckt sind. Der zweite Prozess entspricht einer Folge von Beobachtungen.

Für jede der sieben Klassen, d.h. Aktivitäten c_1, \dots, c_7 ist ein Datensatz mit Beobachtungen $O^c = \{o_1^c, \dots, o_n^c\}$ gegeben. Für jede der Aktivitäten wird ein HMM $\lambda_1, \dots, \lambda_7$ trainiert. Diese HMM werden dann als Klassifikator verwendet, indem jedem HMM die Beobachtung o übergeben wird, die klassifiziert werden soll. Die Beobachtung gehört dann zu derjenigen Klasse, dessen zugehöriges HMM die größte Wahrscheinlichkeit liefert.

An dieser Stelle soll nicht weiter auf Hidden Markov Modelle eingegangen werden, da dies den Umfang dieser Arbeit sprengen würde. Für einen umfassenden Überblick sei auf [21] verwiesen.

6.2 Rohdaten

In diesem Abschnitt wird die Klassifikation mit Rohdaten behandelt. Die Beschleunigungsdaten werden dabei direkt den Hidden Markov Modellen übergeben. Zum Testen der Erkennungsraten wurde, wie bei den Entscheidungsbäumen 10-fold cross validation eingesetzt. Dies vermindert auch den Einfluss der Zufallsinitialisierung der HMM.

6.2.1 Struktur

Für die Klassifikation mit Rohdaten erwies sich ein HMM erster Ordnung mit sechs Zuständen und einer Mischung aus drei Gaußverteilungen (Abbildung 6.2) als am Besten. Weder die Erhöhung der Anzahl der Zustände, noch die Erhöhung der Ordnung führte zu signifikant besseren Erkennungsraten.

		Fenstergröße					
		32	64	128	256	512	Neu
Aktivität	Aufzug	35.75	52.50	58.00	66.50	78.00	59.50
	Rolltreppe	52.63	60.38	79.50	90.50	93.00	89.00
	Ruhezustand	27.13	38.75	42.00	48.50	60.00	52.50
	Rennen	89.75	98.12	99.50	100.00	100.00	96.00
	Treppe runter	63.75	85.55	91.00	94.50	98.00	83.50
	Treppe hoch	70.19	96.50	99.50	100.00	100.00	91.50
	Gehen	76.06	96.58	98.75	99.50	100.00	97.89
Gesamt		59.00	75.46	81.18	85.55	89.00	81.41

Tabelle 6.1: Erkennungsraten für HMM mit Rohdaten, geordnet nach Fenstergröße. "Neu" steht für den Neuesten Zyklus, der aus einem Fenster der Größe 256 bestimmt wurde. Gute Erkennungsraten für periodische Aktivitäten, insbesondere *Rennen* und *Treppe hinuntersteigen*. Statische Aktivitäten weisen schlechte Erkennungsraten auf, *Ruhezustand* wird oft als *Aufzug fahren* erkannt. Die Erkennung des neuesten Zyklus bringt leider keine Verbesserung.

6.2.2 Erkennungsraten

HMM mit Rohdaten erreichen gute Erkennungsraten bei periodischen Aktivitäten, wie *Treppen hochsteigen*, *Treppen hinuntersteigen*, *Gehen* und insbesondere *Rennen*. Sie haben jedoch Schwierigkeiten bei statischen Aktivitäten, wie *Aufzug fahren*, *Rolltreppe fahren* und *Ruhezustand*. Insbesondere zwischen *Aufzug fahren* und *Ruhezustand* kommt es zu Fehlklassifikationen, aufgrund der Ähnlichkeit der beiden Aktivitäten, daher wird für *Ruhezustand* maximal eine Erkennungsrate von 60% erreicht. Die Gesamt-Erkennungsrate wird von den statischen Aktivitäten stark in Mitleidenschaft gezogen. Sie reicht von 59% bei einer Fenstergröße von 32 Samples (etwa 0.5 Sekunden) bis zu 89% bei einer Fenstergröße von 512 (etwa 10 Sekunden). Eine Fenstergröße von 32 ist also für eine gute Erkennung nicht ausreichend, da selbst die periodischen Aktivitäten bei dieser Fenstergröße eine schlechte Erkennungsraten aufweisen. Tabelle 6.1 zeigt die Erkennungsraten für jede einzelne Aktivität, abhängig von der Fenstergröße.

Die Erkennung des neuesten Zyklus bringt leider keine verbesserten Erkennungsraten. Die Wahl eines Fensters mit einer festen Größe von 64 Samples bringt bessere Ergebnisse für periodische Aktivitäten. Eine feste Fenstergröße von 256 Samples bringt bei statischen Aktivitäten die besseren Erkennungsraten.

6.3 Einteilung in Intervalle

In diesem Abschnitt werden die betrachteten Fenster aus Abschnitt 4.3 in Intervalle eingeteilt und anschließend Merkmale aus ihnen extrahiert. Diese werden anschließend den Hidden

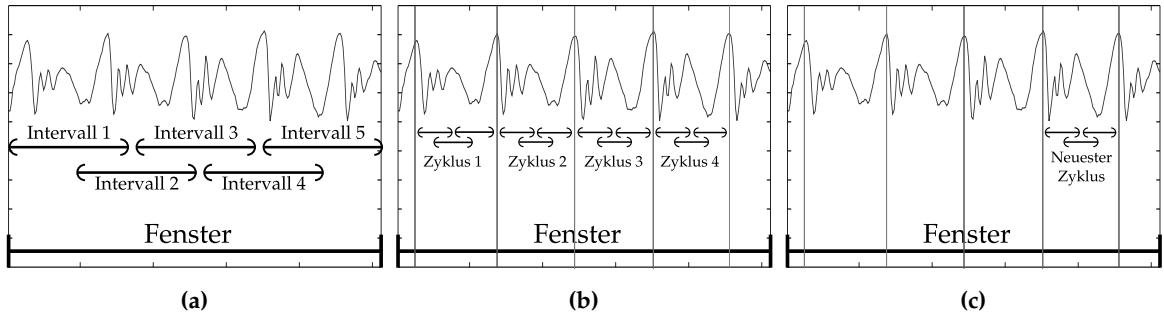


Abbildung 6.3: Weitere Einteilung der Fenster in Intervalle. (a) Einteilung des Fensters in Intervalle fester Größe. (b) Bestimmung der Zyklen und anschließend Einteilung jedes Zyklus in Intervalle fester Größe. (c) Verwendung des neuesten Zyklus und Unterteilung in Intervalle fester Größe.

Markov Modellen übergeben. Zum Testen der Erkennungsrate wurde auch hier 10-fold cross validation eingesetzt.

6.3.1 Merkmalsextraktion

Vor der Merkmalsextraktion werden die in Abschnitt 4.3 erzeugten Fenstern, nochmals in Intervalle aufgeteilt. Dabei hat sich eine 50%-ige Überlappung der Intervalle in vorherigen Arbeiten als vorteilhaft erwiesen [22] [5]. Zusätzlich zu der in bisherigen Arbeiten betrachteten festen Unterteilung werden weitere Möglichkeiten untersucht:

Unterteilung in feste Intervalle Das Zeitfenster wird in m feste Intervalle mit 50% Überlappung eingeteilt. In Abbildung 6.3a ist eine mögliche Unterteilung mit $m = 5$ dargestellt. Aufgrund der 50%-igen Überlappung ist die Intervallanzahl stets ungerade. Die Anzahl der getesteten Aufteilungen schwankt je nach Fenstergröße, da bei immer feinerer Unterteilung der Rechenaufwand rasch ansteigt, bis schließlich mehr Merkmale erzeugt werden, als Datenpunkte im Fenster vorhanden sind.

Zyklenerkennung und Unterteilung aller Zyklen Aus dem Zeitfenster werden zunächst alle Zyklen extrahiert. Jeder der Zyklen wird anschließend in m Intervalle mit 50% Überlappung eingeteilt. Abbildung 6.3b zeigt eine mögliche Unterteilung des Fensters in seine vier Zyklen und anschließende Unterteilung der Zyklen in $m = 3$ Intervalle. Das Verfahren zu Zyklenerkennung arbeitet ab einer Fenstergröße von 256 Samples am Besten, daher wird hierfür eine Fenstergröße von 256 verwendet. Kleinere Fenster führen häufig zu Fehlern in der Abschätzung der Periodenlänge, während längere Fenster keine signifikante Verbesserung mehr bringen.

Zyklenerkennung und Unterteilung des neuesten Zyklus Aus einem Zeitfenster werden zwar alle Zyklen extrahiert, jedoch wird nur der neueste Zyklus weiterverwendet. Er wird ebenfalls in m Intervalle unterteilt. Abbildung 6.3c zeigt eine mögliche Unterteilung mit $m = 3$. Auch hier wird eine Fenstergröße von 256 Samples verwendet.

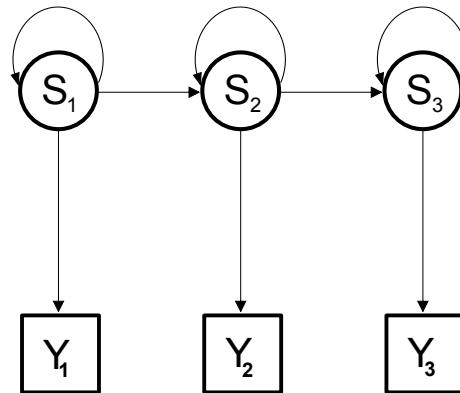


Abbildung 6.4: HMM erster Ordnung mit drei Zuständen. Wird verwendet für die Klassifikation mit Intervallen

Durch die Verwendung des neuesten Zyklus soll die Reaktionsgeschwindigkeit des Verfahrens verbessert werden, bei einer gleichzeitigen Erhöhung der Erkennungsraten im Vergleich zur Verwendung von Fenster der Länge 64 Samples.

Aus jedem dieser Intervalle werden folgende Features generiert (jeweils für die horizontale und vertikale Komponente der Beschleunigung):

- Mittelwert
- Standardabweichung
- Die 8 ersten Koeffizienten der Fourier Transformation

Auf die verwendeten Merkmale wurde bereits zuvor in Abschnitt 5.2 genauer eingegangen, daher soll dies hier nicht nochmals wiederholt werden. Für jedes Intervall erhält man einen Merkmalsvektor, bzw. Beobachtung der Dimension 20:

$$\vec{o} = (\mu^h, \mu^v, \sigma^h, \sigma^v, \hat{x}_1^h, \dots, \hat{x}_8^h, \hat{x}_1^v, \dots, \hat{x}_8^v)$$

Für ein Fenster, welches in n Intervalle unterteilt wird erhält man eine Folge von Beobachtungen:

$$O = (\vec{o}_1, \dots, \vec{o}_n)$$

6.3.2 Struktur

Verschiedene Strukturen für die HMM wurden getestet und ein Modell erster Ordnung mit drei Zuständen hat sich für die Klassifikation mit Hilfe von Merkmale, die aus Intervallen gewonnen werden, als am Besten erwiesen. Eine Erhöhung der Anzahl der Zustände, der Ordnung und die Nutzung von gaußschen Mischungen brachte keine signifikante Verbesserung der Erkennungsraten.

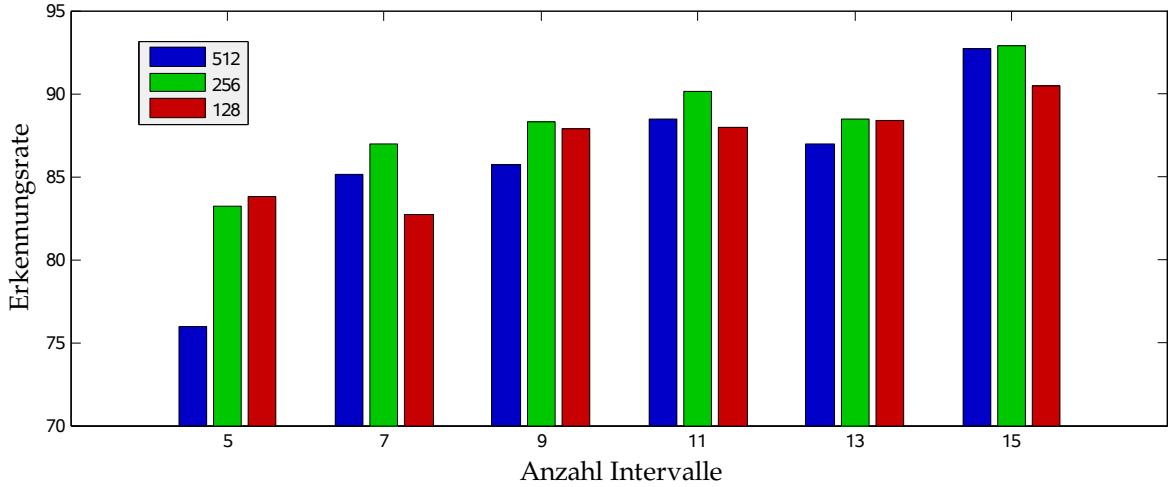


Abbildung 6.5: Erkennungsraten für HMM nach Anzahl der Intervalle, in die ein Fenster unterteilt wird. Betrachtet werden Fenster der Größe 128, 256 und 512. Eine feinere Unterteilung der Fenster hat eine größere Auswirkung auf die Erkennungsraten, als die Wahl größerer Fenster.

6.3.3 Erkennungsraten

Wie schon bei den Entscheidungsbäumen und der Verwendung von Rohdaten für die HMM weisen auch hier die periodischen Aktivitäten eine höhere Erkennungsraten als die statischen Aktivitäten auf. Die meisten Fehlklassifikationen treten zwischen *Ruhezustand* und *Aufzug fahren* auf, allerdings weit weniger als bei Verwendung von Rohdaten. Dennoch beeinflusst die schlechtere Erkennungsraten der statischen Aktivitäten die Gesamterkennungsraten negativ.

Die Nutzung einer feineren Aufteilung des Fensters verbessert die Erkennungsraten erheblich. Wie in Abbildung 6.5 dargestellt, wirkt sich die Wahl einer feineren Aufteilung des Fensters meist sogar stärker aus, als die Wahl einer größeren Fensters. Eine Steigerung der Intervallanzahl von 13 auf 15 bringt eine Steigerung von durchschnittlich etwa 4%, während die Wahl größerer Fenster bei einer gleich bleibender Intervallanzahl von 13 keine Verbesserung bringt. Im Gegenteil, mit einer Fenstergröße von 256 (etwa 5 Sekunden) lassen sich bessere Erkennungsraten erzielen, als mit einer Fenstergröße von 512 (10 Sekunden).

Bei einem Fenster der Größe 512 Samples, unterteilt in 15 Intervalle der Länge 64 Samples (etwa 1 Sekunde) beträgt die Gesamterkennungsraten beispielsweise 92.71 %. Bei einem Fenster der Größe 256 Samples (etwa 5 Sekunden), unterteilt in 15 Intervalle der Länge 32 Samples (etwa 0.5 Sekunden) beträgt die Erkennungsraten 92.85 %. Trotz des kürzeren Fensters und wird also die gleiche Erkennungsraten erreicht, indem die Intervalle feiner gewählt wurden. Allgemein lässt sich sagen, dass eine Fenstergröße von 256 die besten Erkennungsraten liefert, unabhängig von der Anzahl der zur Aufteilung verwendeten Intervalle.

		Anzahl Intervalle						
		3	5	7	9	11	13	15
Fenstergröße	512	-	76.00	85.14	85.71	88.43	87.00	92.71
	256	-	83.42	86.98	88.33	90.14	88.49	92.85
	128	-	83.82	82.71	87.86	87.93	88.39	90.48
	64	79.25	86.21	87.50	-	-	-	-
	32	60.61	72.96	80.83	-	-	-	-
	Alle Zyklen aus 256	73.57	75.71	-	-	-	-	-
	Neuester Zyklus aus 256	67.61	72.00	70.71	-	-	-	-

Tabelle 6.2: Verwendete Aufteilungen der Fenster in Intervalle und resultierende Erkennungsraten. Die höchste Erkennungsrate wird bei einem Fenster der Größe 256 erreicht, welches in 15 Intervalle aufgeteilt wird. Erkennbar ist auch, dass die Nutzung der Zyklenerkennung keine Verbesserung mit sich bringt, sondern die Erkennungsraten verschlechtert. Die Zyklen wurden aus einem Fenster der Größe 256 bestimmt.

		Anzahl Intervalle (Intervalllänge)					
		5 (85)	7 (64)	9 (51)	11 (42)	13 (36)	15 (32)
Aktivität	Aufzug	82.50	82.00	78.50	78.00	80.00	83.50
	Rolltreppe	70.50	80.50	90.50	91.00	87.00	91.00
	Ruhezustand	53.50	63.00	67.00	75.00	70.50	83.00
	Rennen	93.00	95.00	94.50	96.50	93.00	99.00
	Treppe Runter	90.50	97.00	93.00	94.00	93.50	96.50
	Treppe Hoch	95.55	93.50	98.00	98.00	97.00	98.00
	Gehen	98.42	97.89	96.84	98.50	98.42	98.95
Gesamt		83.42	86.98	88.33	90.14	88.49	92.85

Tabelle 6.3: Erkennungsraten für HMM nach Anzahl der Intervalle, Fenstergröße: 256 Samples / 5 Sekunden. Erkennungsrate steigt durch eine feinere Aufteilung des Fensters an. Periodische Aktivitäten weisen durchweg gute Erkennungsraten von über 90% auf. Statische Aktivitäten etwas schlechtere, insbesondere *Aufzug fahren* und *Ruhezustand*

		Anzahl Intervalle						
		3 (16)	5 (14)					
Aktivität	Aufzug	45.00	80.00	Aktivität	Aufzug	59.00	73.00	65.50
	Rolltreppe	70.00	65.00		Rolltreppe	39.50	49.50	56.50
	Ruhezustand	15.00	15.00		Ruhezustand	49.50	48.50	41.50
	Rennen	95.75	90.00		Rennen	81.50	79.00	86.50
	Treppe Runter	100.00	100.00		Treppe Runter	73.50	79.50	75.00
	Treppe Hoch	90.00	90.75		Treppe Hoch	79.50	81.00	81.00
	Gehen	100.00	90.00		Gehen	91.50	93.50	92.50
	Gesamt	73.57	75.71		Gesamt	67.61	72.00	70.71

(a)

(b)

Tabelle 6.4: Erkennungsraten der HMM mit Zyklenerkennung. **(a)** Erkennungsrate bei Bestimmung aller Zyklen aus einem Fenster der Größe 256 und anschließender Unterteilung aller Zyklen. Gute Erkennungsraten bei periodischen Aktivitäten, insbesondere *Treppen hinuntersteigen* und *Gehen*. Allerdings extrem schlechte bei *Aufzug fahren* und *Ruhezustand*. **(b)** Erkennungsrate bei Verwendung des neuesten Zyklus und anschließender Unterteilung. Erkennungsraten sind gegenüber einer festen Unterteilung durchweg unterlegen.

Die Bestimmung der Zyklen aus einem Fenster der Größe 256 und die anschließende Einteilung der Zyklen in Intervalle brachte leider nicht die erhofften Ergebnisse (Tabelle 6.2). Periodischen Aktivitäten, insbesondere *Treppen hinuntersteigen* und *Gehen* weisen zwar gute Erkennungsraten auf, jedoch haben statische Aktivitäten und besonders *Ruhezustand* extrem schlechte Erkennungsraten (Tabelle 6.4a). Insgesamt ist es dem Verfahren mit Einteilung des Fensters in feste Intervalle deutlich unterlegen und eine durchschnittlich 20% schlechtere Gesamterkennungsrate.

Ähnlich verhält es sich, wenn nur der neueste Zyklus aus dem Zeitfenster zur Erkennung herangezogen wird (Tabelle 6.4b). Die Erkennungsrate für statische Aktivitäten beträgt maximal 57%. Da statische Aktivitäten keine Zyklen erhalten wird das gesamte Fenster als Zyklus betrachtet, deshalb muss hier mit einem Fenster der Größe 256 verglichen werden, welches in feste Anzahl Intervalle aufgeteilt wird. Dort wurden Erkennungsraten von 85% erreicht, also über 30% mehr als mit Zyklenerkennung. Für periodische Aktivitäten wird das Verfahren mit einer Fenstergröße von 64 verglichen, da *Gehen*, *Treppen hinaufsteigen* und *Treppen hinuntersteigen* alle eine Zyklendlänge von 50-70 Samples aufweisen. Doch auch hier erwies sich eine Aufteilung in feste Intervalle als besser (Tabelle 6.2).

7 Diskussion

In diesem Kapitel wird auf die Stärken und Schwächen der beiden Klassifikationsverfahren eingegangen und die Auswirkung der Zyklenerkennung diskutiert. Anschließend werden die Klassifikationsergebnisse von Entscheidungsbäumen mit den Ergebnissen der Hidden Markov Modelle verglichen.

An dieser Stelle sei auf den Anhang dieser Arbeit verweisen, wo sich alle Konfusionsmatrizen für die Hidden Markov Modelle und die Entscheidungsbäume finden lassen - alle erstellt mit 10-fold cross validation.

7.1 Entscheidungsbäume

Entscheidungsbäume liefern bei der Verwendung von festen Fenstergrößen ähnlich hohe Erkennungsrate, wie in [25] [11]. Die Erkennungsraten für *Treppen hinaufsteigen* und *Treppen hinuntersteigen* können durch die Transformation auf ein körperfestes Koordinatensystem (Abschnitt 4.4) maßgeblich verbessert werden. Ohne diese Transformation treten häufig Fehlklassifikationen mit *Gehen* auf. Zuvor wurden nur Erkennungsraten von 55% bis 62% erreicht (vgl. [11]), in dieser Arbeit konnten Erkennungsraten von über 90% erzielt werden.

Die Verwendung der Zyklenerkennung bringt eine Verbesserung der Erkennungsrate, insbesondere bei periodischen Aktivitäten. Die Erkennungsraten mit Zyklenerkennung unterscheiden sich für statische Aktivitäten allerdings kaum von den Erkennungsraten mit einer festen Fenstergröße von 256 Samples. Dies war auch nicht anders zu erwarten, da der Beschleunigungsverlauf bei statischen Aktivitäten keine Zyklen enthält und daher das ganze Fenster als Zyklus aufgefasst wird.

7.2 Hidden Markow Modelle

Die Einteilung der Fenster in Intervalle und anschließende Extraktion von Merkmalen aus diesen liefert deutlich bessere Ergebnisse, als die direkte Verwendung von Rohdaten. Im Vergleich mit den periodischen Aktivitäten sind die Erkennungsraten für statische Aktivitäten *Aufzug fahren*, *Rolltreppe fahren* und *Ruhezustand* zwar durchweg schlechter, allerdings schneiden sie im Vergleich zu HMM mit Rohdaten dennoch besser ab.

Die durch die Zyklenerkennung erhoffte Verbesserung der Erkennungsrate trat bei HMM leider nicht ein. Im Gegenteil, die Erkennungsraten waren deutlich schlechter. Dies könnte

daran liegen, dass die Unterteilung bei statischen Aktivitäten zu grob ist. Denn diese enthalten wie bereits gesagt keine Zyklen, es wird also das gesamten Fenster verwendet und in nur wenige Intervalle unterteilt. Bei den periodischen Aktivitäten konnte die Unterteilung ebenfalls nicht so fein gewählt werden, wie bei einer festen Fenstergröße, da *Rennen* einen extrem kurzen Zyklus von 8 bis 10 Samples aufweist und sich daher schlecht unterteilen lässt.

Da die Einteilung in fest Intervalle die besten Ergebnisse bietet soll nur dieses Verfahren mit den Entscheidungsbäumen verglichen werden.

7.3 Vergleich der Verfahren

Für alle Klassifikationsverfahren gilt, dass Fehlklassifikationen fast ausschließlich zwischen den einzelnen statischen (*Aufzug fahren*, *Rolltreppe fahren*, *Ruhezustand*) und dynamischen Aktivitäten (*Rennen*, *Treppen hochsteigen*, *Treppen hinuntersteigen*, *Gehen*) auftreten, d.h. statische Aktivitäten werden nur selten als periodische klassifiziert und umgekehrt.

Da außerdem die Erkennungsraten für die statischen Aktivitäten durchweg schlechter sind, als die der dynamischen Aktivitäten, werden hier beide getrennt betrachtet:

Statische Aktivitäten Die Erkennung von *Rolltreppe fahren* funktioniert für Entscheidungsbäume und HMM mit Intervallen gut und bieten ähnliche Erkennungsraten. Für die beiden übrigen statischen Aktivitäten *Aufzug fahren* und *Ruhezustand* bieten Entscheidungsbäume die besseren Erkennungsraten. Aufgrund ihrer Ähnlichkeit schneiden sie jedoch auch hier schlechter ab. Gut erkennbar sind lediglich die Beschleunigungs- und Bremsphasen, während das dazwischenliegenden Stehphasen auch als solche erkannt werden, d.h. Stehen wird als *Aufzug fahren* klassifiziert und umgekehrt.

Periodische Aktivitäten Die Erkennungsraten für *Rennen* sind durchweg sehr gut, unterscheidet sich aber auch am deutlichsten von den restlichen Aktivitäten. *Treppen hinuntersteigen* hat unter den dynamischen Aktivitäten die niedrigsten Erkennungsraten, was wohl in der Ähnlichkeit zu *Gehen* begründet liegt. HMM bieten für die dynamische Aktivitäten eine um etwa 2-3 % höhere Erkennungsrate im Vergleich zu Entscheidungsbäumen.

Tabelle 7.1 stellt die Gesamterkennungsraten der Entscheidungsbäume den HMM gegenüber. Für die HMM wurde für jede Fenstergröße diejenige Intervalleinteilung ausgewählt, die die besten Ergebnisse liefert. Für kürzere Fenster ist die Erkennungsrate der HMM zwar besser, allerdings leiden sie durch die schlechtere Klassifikationsergebnisse bei statischen Aktivitäten. Für längere Fenster liegen die HMM mit den Entscheidungsbäumen gleichauf und für eine Fenstergröße von 256 Samples (etwa 5 Sekunden) übertreffen die HMM die Entscheidungsbäume sogar, da das Problem der schlechteren Klassifikation der statischen Aktivitäten nicht mehr besteht (Tabelle 7.2).

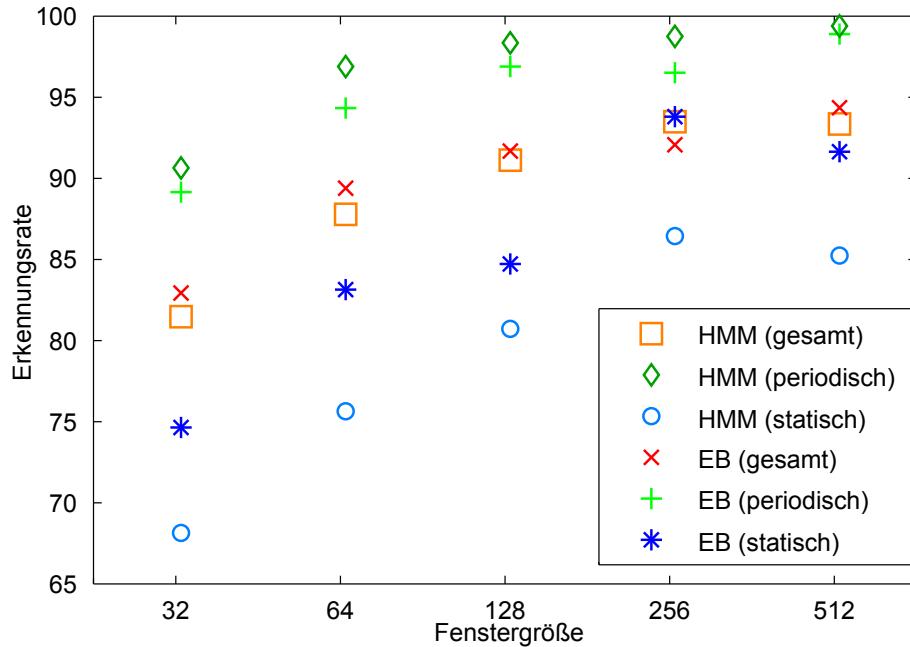


Abbildung 7.1: Vergleich der Erkennungsraten. Jeweils für periodische Aktivitäten, Statische Aktivitäten und Gesamterkennungsraten. Entscheidungsbäume liefern für statische Aktivitäten bessere Ergebnisse, HMM für periodische Aktivitäten. Es sind nur HMM dargestellt, die mit Hilfe von Merkmalen aus der Intervalleinteilung trainiert wurden.

Abbildung 7.1 bietet einen direkten Vergleich der Erkennungsraten der getesteten Verfahren. Da die HMM mit Rohdaten deutlich schlechtere Erkennungsraten im Vergleich zu HMM mit Intervallen bieten, werden sie in Abbildung 7.1 der Übersicht halber nicht mehr aufgeführt.

Zusammenfassend lässt sich sagen, dass das Hidden Markov Modelle bessere Ergebnisse bei dynamischen Aktivitäten aufweisen, während Entscheidungsbäume die besseren Ergebnisse bei statischen Aktivitäten zeigen, insbesondere bei kürzeren Fenstern.

		Fenstergröße					
		32	64	128	256	512	
HMM	Rohdaten	Gesamt	59.00	75.46	81.18	85.55	89.00
	Intervalle	Periodisch	74.94	94.18	97.19	98.50	99.50
		Statisch	38.50	50.54	59.83	68.50	77.00
HMM	Intervalle	Gesamt	80.83	87.5	90.48	92.85	92.71
		Periodisch	90.00	96.25	97.71	98.11	98.75
		Statisch	67.50	75.00	80.08	85.83	84.60
EB		Gesamt	82.29	88.75	91.04	91.43	93.71
		Periodisch	88.51	93.69	96.25	95.97	98.25
		Statisch	74.00	82.50	84.08	83.16	91.00

Tabelle 7.1: Vergleich der Erkennungsraten. Jeweils für periodische Aktivitäten, Statische Aktivitäten und Gesamterkennungsraten. Die besten Erkennungsraten sind hervorgehoben. HMM trainiert mit Merkmalen aus Intervallen weisen bei einem Fenster der Größe 256 die besten Erkennungsraten auf. Bei den übrigen Fenstergrößen sind die Entscheidungsbäume etwas besser. HMM mit Rohdaten sind stets unterlegen.

Aktivität	Erkennungsraten	
	Entscheidungsbaum	HMM
Aufzug fahren	82.00	83.50
Rolltreppe	85.00	91.00
Ruhezustand	89.50	83.00
Rennen	100.00	99.00
Treppe runter	97.00	96.50
Treppe hoch	94.50	98.00
Gehen	92.00	98.95
Gesamt	91.43	92.85

Tabelle 7.2: Vergleich der Erkennungsraten von Entscheidungsbaum und HMM bei einer Fenstergröße von 256 Samples. Für das HMM wird das Fenster in 15 Intervalle aufgeteilt. Bei statischen Aktivitäten liegt das HMM gleichauf mit den Entscheidungsbäumen und können sie insgesamt leicht übertreffen.

8 Zusammenfassung und Ausblick

In dieser Arbeit wurden zwei Verfahren zur Erkennung von Bewegungen anhand von Beschleunigungsdaten vorstellt. Zum einen die häufiger eingesetzten Entscheidungsbäume, zum anderen Hidden Markov Modelle. Zur Datenakquisition wurde eine Anwendung auf dem T-Mobile G1 entwickelt, deren Hilfe Daten gesammelt wurden. Es wurde eine Testsuite in Matlab unter Zuhilfenahme der HMM-Toolbox [14] entwickelt und beide Verfahren mit dem Datensatz getestet.

Dabei dienen die Entscheidungsbäume als Referenzimplementierung, um eine bessere Vergleichbarkeit der Ergebnisse mit anderen Arbeiten zu gewährleisten. Die Auswirkungen der Wahl verschiedener Fenstergrößen, sowie deren Unterteilung wurden betrachtet. Zur Unterteilung kamen verschiedene Verfahren zum Einsatz, darunter auch ein neuer Ansatz, bei dem Zyklen im Beschleunigungsverlauf zur Hilfe genommen werden.

Es wurden einige, bisher kaum betrachtete Aktivitäten, wie *Aufzug fahren*, *Rolltreppe fahren*, *Treppen hinaufsteigen* und *Treppen hinuntersteigen* untersucht und durchweg gute Erkennungsraten erreicht. Hidden Markov Modelle schneiden bei periodischen Aktivitäten besser ab, sind jedoch bei statischen Aktivitäten den Entscheidungsbäumen unterlegen. Insgesamt werden ähnliche Erkennungsraten erreicht, wie in anderen Arbeiten zuvor.

Ausblick

Die Erkennungsrate der Aktivität *Aufzug fahren* könnte mit Hilfe eines zustandsabhängigen Modells verbessert werden. Beispielsweise kann das *Ruhezustand*, bzw. *Stehen* der zwischen einer Beschleunigungs- und einer Bremsphase des Aufzugs als *Aufzug fahren* erkannt werden. Die Akquisition und Aufbereitung der Trainingsdaten für ein solches Modell nimmt jedoch sehr viel Zeit in Anspruch, da weite Teile der gesammelten Daten schlecht verwendbar sind. Beispielsweise entspricht die Zeit zwischen Öffnen und Schließen der Aufzugtüren der Aktivität *Ruhezustand*, bzw. *Stehen* und muss daher entfernt oder zumindest separat betrachtet werden. Falls andere zusammengesetzte Aktivitäten betrachtet werden sollen, muss jedoch für jede ein eigenes Modell verwendet werden. Ein allgemeinere Lösung wäre hier wünschenswert.

Hinsichtlich der Hidden Markov Modelle verbessert eine feinere Aufteilung der Zeitfenster in Intervalle die Erkennungsrate erheblich. Die Nutzung einer noch feineren Intervalleinteilung, als in dieser Arbeit könnte die Erkennungsraten weiter verbessern. Im Gegenzug muss allerdings die Anzahl der verwendeten Merkmale pro Intervall reduziert werden, um den Rechenaufwand in Grenzen zu halten.

8 Zusammenfassung und Ausblick

In dieser Arbeit wurde als Sensorposition die vordere Hosentasche gewählt. Dies ist zwar eine recht häufige Position für das Tragen von Mobiltelefonen und die Erkennung ist weitgehend unabhängig von der genauen Position in der Hosentasche selbst, allerdings ist es dennoch eine Einschränkung. Die Erkennung funktioniert nämlich nicht, falls das Gerät beispielsweise in einem Rucksack getragen wird. In Zukunft muss daher eine Lösung für erarbeitet werden, die eine weitgehende Unabhängigkeit von der Position des Sensors ermöglicht.

Mit dem Aufkommen von Gyroskopen in Smartphone können die Erkennungsraten sicher noch weiter verbessert werden. Durch die mit einem Gyroskop erreichbare, weitgehende Unabhängigkeit der Sensordaten von der Bewegung der Benutzers, können weitere Anwendungsbereiche erschlossen werden, beispielsweise die Navigation innerhalb von Gebäuden. Dafür müssen jedoch die Daten des Gyroskops mit den Daten des Beschleunigungssensors fusioniert werden. Hierfür bietet sich beispielsweise die Verwendung von Kalman Filtern an.

A Anhang

A.1 Weitere Konfusionsmatrizen

Dieser Abschnitt enthält Konfusionsmatrizen der durchgeführten Tests. Für die Tests wurde 10-fold cross-validation verwendet und für jeden Test die Konfusionsmatrix berechnet. Anschließend wurde die Summe für jeden Eintrag und daraus die relative Häufigkeit gebildet. Die Aktivitäten *Treppen hochsteigen* und *Treppen hinuntersteigen* werden aus Platzgründen mit "Hoch" und "Runter" abgekürzt.

A.1.1 Entscheidungsbäume

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	82.63	9.04	8.33	0	0	0	0
	Rolltr.	6.25	90	3.75	0	0	0	0
	Ruhez.	52.21	8.33	39.33	0	0	0.13	0
	Rennen	0	0	0	99.46	0	0.17	0.38
	Runter	0	0	0	0.25	85.5	8.42	5.83
	Hoch	0.04	0	0.04	0	7.46	83.25	9.21
	Gehen	0	0	0	0.08	4.5	11.33	84.08

Tabelle A.1: Konfusionsmatrix für Entscheidungsbaum, Fenstergröße 32

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	83.25	3.75	13	0	0	0	0
	Rolltr.	6.5	88.5	5	0	0	0	0
	Ruhez.	21.88	2.38	75.75	0	0	0	0
	Rennen	0	0	0	100	0	0	0
	Runter	0	0	0	0	96.38	1.5	2.13
	Hoch	0	0	0	0	1.75	80.13	18.13
	Gehen	0	0	0	0	0.5	2.25	97.25

Tabelle A.2: Konfusionsmatrix für Entscheidungsbaum, Fenstergröße 64

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Hoch	Runter	Gehen
Vorliegend	Aufzug	82	10	8	0	0	0	0
	Rolltr.	3	85	12	0	0	0	0
	Ruhez.	10	0.5	89.5	0	0	0	0
	Rennen	0	0	0	100	0	0	0
	Runter	0	0	0	0	97	1.5	1.5
	Hoch	0	0	0	0	94.5	5.5	
	Gehen	0	0	0	0	0	8	92

Tabelle A.3: Konfusionsmatrix für Entscheidungsbaum, Fenstergröße 256

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	89	4	7	0	0	0	0
	Rolltr.	0	90	10	0	0	0	0
	Ruhez.	10	6	84	0	0	0	0
	Rennen	0	0	0	100	0	0	0
	Runter	0	0	0	0	99	1	0
	Hoch	0	0	0	0	1	98	1
	Gehen	0	0	0	0	0	4	96

Tabelle A.4: Konfusionsmatrix für Entscheidungsbaum, Fenstergröße 512

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	67	0	33	0	0	0	0
	Rolltr.	0.5	99	0.5	0	0	0	0
	Ruhez.	14	0.5	85.5	0	0	0	0
	Rennen	0	0	0	100	0	0	0
	Runter	0	0	0	0	90.5	3.5	6
	Hoch	0	0	0	0	2.5	96	1.5
	Gehen	0	0	0	0	1.5	0	98.5

Tabelle A.5: Konfusionsmatrix für Entscheidungsbaum, Neuester Zyklus aus Fenstergröße 256

A.1.2 HMM

Features aus Intervallen

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	70	6	24	0	0	0	0
	Rolltr.	1	57	41	0	1	0	0
	Ruhez.	5	40	53	0	2	0	0
	Rennen	0	3	1	88	8	0	0
	Runter	0	10	2	0	88	0	0
	Hoch	0	4	3	0	3	89	1
	Gehen	0	2	3	0	5	3	87

Tabelle A.6: Konfusionsmatrix für HMM, Fenstergröße 512, Features aus 5 Intervallen mit mit 50 % Überlappung, Struktur: 3 Zustände, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	85	11	4	0	0	0	0
	Rolltr.	0	85	15	0	0	0	0
	Ruhez.	3	46	48	0	3	0	0
	Rennen	0	3	0	96	0	1	0
	Runter	0	6	0	0	94	0	0
	Hoch	0	6	1	0	2	91	0
	Gehen	0	0	0	0	0	3	97

Tabelle A.7: Konfusionsmatrix für HMM, Fenstergröße 512, Features aus 7 Intervallen mit mit 50 % Überlappung, Struktur: 3 Zustände, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	87	4	9	0	0	0	0
	Rolltr.	0	74	26	0	0	0	0
	Ruhez.	1	33	66	0	0	0	0
	Rennen	0	1	0	99	0	0	0
	Runter	0	1	1	0	98	0	0
	Hoch	0	0	0	0	3	97	0
	Gehen	0	0	0	0	0	2	98

Tabelle A.8: Konfusionsmatrix für HMM, Fenstergröße 512, Features aus 11 Intervallen mit mit 50 % Überlappung, Struktur: 3 Zustände, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	90	6	4	0	0	0	0
	Rolltr.	1	72	27	0	0	0	0
	Ruhez.	2	29	69	0	0	0	0
	Rennen	0	6	3	90	1	0	0
	Runter	0	1	0	3	96	0	0
	Hoch	0	0	1	0	4	95	0
	Gehen	0	0	0	0	0	3	97

Tabelle A.9: Konfusionmatrix für HMM, Fenstergröße: 512, Features aus 13 Intervallen mit mit 50 % Überlappung, Struktur: 3 Zustände, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	97	1	2	0	0	0	0
	Rolltr.	0	78	22	0	0	0	0
	Ruhez.	0	21	79	0	0	0	0
	Rennen	0	0	0	100	0	0	0
	Runter	0	0	1	0	99	0	0
	Hoch	0	0	0	0	3	97	0
	Gehen	0	0	0	0	0	1	99

Tabelle A.10: Konfusionmatrix für HMM, Fenstergröße: 512, Features aus 15 Intervallen mit mit 50 % Überlappung, Struktur: 3 Zustände, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	82.5	10.5	6	0	1	0	0
	Rolltr.	2.5	70.5	24	0	3	0	0
	Ruhez.	9	34.5	53.5	0	3	0	0
	Rennen	0	3.5	1	93	2.5	0	0
	Runter	0	6.5	3	0	90.5	0	0
	Hoch	0	0.5	1	0	3	95.5	0
	Gehen	0	0	0	0	0.53	1.05	98.42

Tabelle A.11: Konfusionmatrix für HMM, Fenstergröße: 256, Features aus 5 Intervallen mit mit 50 % Überlappung, Struktur: 3 Zustände, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	82	9	8.5	0	0.5	0	0
	Rolltr.	1.5	80.5	18	0	0	0	0
	Ruhez.	9	27.5	63	0	0.5	0	0
	Rennen	0	1	1	95	3	0	0
	Runter	0	1	2	0	97	0	0
	Hoch	0	2	2	0	2.5	93.5	0
	Gehen	0	0.53	0	0	0.53	1.05	97.89

Tabelle A.12: Konfusionmatrix für HMM, Fenstergröße: 256, Features aus 7 Intervallen mit 50 % Überlappung, Struktur: 3 Zustände, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	78.5	8	13	0	0.5	0	0
	Rolltr.	2.5	90.5	7	0	0	0	0
	Ruhez.	8	19	67	0	6	0	0
	Rennen	0	0	0.5	94.5	5	0	0
	Runter	0	1.5	5.5	0	93	0	0
	Hoch	0	0	1	0	1	98	0
	Gehen	0	0	0.53	0	2.11	0.53	96.84

Tabelle A.13: Konfusionmatrix für HMM, Fenstergröße: 256, Features aus 9 Intervallen mit 50 % Überlappung, Struktur: 3 Zustände, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	78	2.5	19.5	0	0	0	0
	Rolltr.	0.5	91	8.5	0	0	0	0
	Ruhez.	6	12.5	75	0	6.5	0	0
	Rennen	0	0	1.5	96.5	2	0	0
	Runter	0	0.5	5.5	0	94	0	0
	Hoch	0	0	0	0	2	98	0
	Gehen	0	0	0	0	1	0.5	98.5

Tabelle A.14: Konfusionmatrix für HMM, Fenstergröße: 256, Features aus 11 Intervallen mit 50 % Überlappung, Struktur: 3 Zustände, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	80	6.5	13.5	0	0	0	0
	Rolltr.	1.5	87	11.5	0	0	0	0
	Ruhez.	4.5	22.5	70.5	0	2.5	0	0
	Rennen	0	1	2	93	4	0	0
	Runter	0	5.5	1	0	93.5	0	0
	Hoch	0	0	0.5	0	2.5	97	0
	Gehen	0	0	0	0	0	1.58	98.42

Tabelle A.15: Konfusionmatrix für HMM, Fenstergröße: 256, Features aus 13 Intervallen mit 50 % Überlappung, Struktur: 3 Zustände, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	83.5	2.5	13.5	0	0.5	0	0
	Rolltr.	1.5	91	7	0	0.5	0	0
	Ruhez.	0.5	9	83	0	7.5	0	0
	Rennen	0	0	1	99	0	0	0
	Runter	0	2	1.5	0	96.5	0	0
	Hoch	0	0.5	0	0	1.5	98	0
	Gehen	0	0	0	0	0	1.05	98.95

Tabelle A.16: Konfusionmatrix für HMM, Fenstergröße: 256, Features aus 15 Intervallen mit 50 % Überlappung, Struktur: 3 Zustände, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	72.75	5.5	18.75	0	3	0	0
	Rolltr.	1.5	81.25	13.75	0	3.5	0	0
	Ruhez.	4	19	68.75	0	8.25	0	0
	Rennen	0	0.5	2.5	91	6	0	0
	Runter	0.5	6	7.25	0	86.25	0	0
	Hoch	0	0.75	2.25	0	5.75	91.25	0
	Gehen	0.25	0.25	1.25	0	2.5	0.25	95.5

Tabelle A.17: Konfusionmatrix für HMM, Fenstergröße: 128, Features aus 5 Intervallen mit 50 % Überlappung, Struktur: 3 Zustände, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	73	2	19	0	6	0	0
	Rolltr.	6	80	11	0	3	0	0
	Ruhez.	13	14	61	0	12	0	0
	Rennen	1	0	1	93	5	0	0
	Runter	3	4	10	0	83	0	0
	Hoch	1	0	2	0	5	92	0
	Gehen	0	0	0	0	1	2	97

Tabelle A.18: Konfusionmatrix für HMM, Fenstergröße: 128, Features aus 7 Intervallen mit 50 % Überlappung, Struktur: 3 Zustände, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Hoch	Runter	Gehen
Vorliegend	Aufzug	75	0	25	0	0	0	0
	Rolltr.	5	70	20	0	5	0	0
	Ruhez.	5	0	95	0	0	0	0
	Rennen	0	0	0	90	10	0	0
	Runter	5	0	10	0	85	0	0
	Hoch	0	0	0	0	0	100	0
	Gehen	0	0	0	0	0	0	100

Tabelle A.19: Konfusionmatrix für HMM, Fenstergröße: 128, Features aus 9 Intervallen mit 50 % Überlappung, Struktur: 3 Zustände, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Hoch	Runter	Gehen
Vorliegend	Aufzug	71.5	4.5	24	0	0	0	0
	Rolltr.	7	74	19	0	0	0	0
	Ruhez.	12.5	6	81.5	0	0	0	0
	Rennen	0	0	2	96	2	0	0
	Runter	0	3	1	0	96	0	0
	Hoch	0	0	0	0	0.5	99.5	0
	Gehen	0	0.5	0	0	0	2.5	97

Tabelle A.20: Konfusionmatrix für HMM, Fenstergröße: 128, Features aus 11 Intervallen mit 50 % Überlappung, Struktur: 3 Zustände, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Hoch	Runter	Gehen
Vorliegend	Aufzug	71.5	9.25	19	0	0.25	0	0
	Rolltr.	6.25	81.25	12.5	0	0	0	0
	Ruhez.	7	12.75	80	0	0.25	0	0
	Rennen	0.75	1.5	0.5	91.25	6	0	0
	Runter	0	2	0.5	0	97.5	0	0
	Hoch	0.25	0	0	0	0.25	99.5	0
	Gehen	0.25	0	0	0	0	2	97.75

Tabelle A.21: Konfusionmatrix für HMM, Fenstergröße: 128, Features aus 13 Intervallen mit 50 % Überlappung, Struktur: 3 Zustände, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	77.5	5.5	17	0	0	0	0
	Rolltr.	7	86.5	6.5	0	0	0	0
	Ruhez.	12.5	9	78.5	0	0	0	0
	Rennen	0.5	0.5	0	97	2	0	0
	Runter	0.5	1.5	0	1.5	96.5	0	0
	Hoch	0	0	0	0	0	100	0
	Gehen	0.53	0	0	0	0	2.11	97.37

Tabelle A.22: Konfusionmatrix für HMM, Fenstergröße: 128, Features aus 15 Intervallen mit 50 % Überlappung, Struktur: 3 Zustände, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	70.63	6.25	17.5	0	5.25	0.38	0
	Rolltr.	5.38	69.13	21.13	0.13	4.25	0	0
	Ruhez.	9.63	17.38	64.13	0.13	8.5	0.25	0
	Rennen	1.13	1.5	4.75	89.38	3.25	0	0
	Runter	2.88	6.88	12.13	0.13	77.75	0.25	0
	Hoch	1.75	1.25	4.25	0	3.5	89.25	0
	Gehen	0.5	0.5	2.25	0	1.63	0.63	94.5

Tabelle A.23: Konfusionmatrix für HMM, Fenstergröße: 64, Features aus 3 Intervallen mit 50 % Überlappung, Struktur: 3 Zustände, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	72.5	7.25	19.5	0	0.75	0	0
	Rolltr.	3.5	74.75	20.25	0	1.5	0	0
	Ruhez.	8.88	17.38	73.38	0	0.38	0	0
	Rennen	0.13	0.63	2.25	96.13	0.88	0	0
	Runter	1	3.13	3.13	0	92.75	0	0
	Hoch	0.75	0.25	0.5	0	1.25	97.25	0
	Gehen	0.25	0.13	0.63	0	0.63	1.63	96.75

Tabelle A.24: Konfusionmatrix für HMM, Fenstergröße: 64, Features aus 5 Intervallen mit 50 % Überlappung, Struktur: 3 Zustände, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	74.84	5.11	20.05	0	0	0	0
	Rolltr.	2.41	77.59	20	0	0	0	0
	Ruhez.	22.43	0	77.57	0	0	0	0
	Rennen	0	0	2.5	95	2.5	0	0
	Runter	2.5	0	5	0	92.5	0	0
	Hoch	0	0	0	0	2.5	97.5	0
	Gehen	0	0	0	0	0	2.5	97.5

Tabelle A.25: Konfusionmatrix für HMM, Fenstergröße: 64, Features aus 7 Intervallen mit 50 % Überlappung, Struktur: 3 Zustände, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	40.25	20.5	29	1	7.25	1.75	0.25
	Rolltr.	19	45	22.25	4.5	8.75	0.5	0
	Ruhez.	22.75	22.25	49.25	1.25	4.25	0.25	0
	Rennen	2.25	7.5	2.75	78.75	7.75	1	0
	Runter	10	8.75	8.5	2	54.5	13.25	3
	Hoch	2.75	2.25	2.25	1.25	7.5	80.5	3.5
	Gehen	2.5	1	0.75	0	3.75	16	76

Tabelle A.26: Konfusionmatrix für HMM, Fenstergröße: 32, Features aus 3 Intervallen mit 50 % Überlappung, Struktur: 3 Zustände, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	49.75	21.5	22.25	0.5	4.5	1.5	0
	Rolltr.	18.75	69.5	9	0.5	2.25	0	0
	Ruhez.	28.5	18.75	50.75	0.75	0.75	0.5	0
	Rennen	1	1.75	1.25	89	6.5	0.25	0.25
	Runter	4.5	7.5	0.75	2.5	76	5	3.75
	Hoch	1.25	0.25	0	0	3	92.75	2.75
	Gehen	0.5	0.25	0	0	3.25	13	83

Tabelle A.27: Konfusionmatrix für HMM, Fenstergröße: 32, Features aus 5 Intervallen mit 50 % Überlappung, Struktur: 3 Zustände, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Hoch	Runter	Gehen
Vorliegend	Aufzug	65	2.5	30	0	2.5	0	0
	Rolltr.	10	82.5	7.5	0	0	0	0
	Ruhez.	45	0	55	0	0	0	0
	Rennen	0	5	0	95	0	0	0
	Runter	0	7.5	0	2.5	77.5	5	7.5
	Hoch	0	0	0	0	7.5	92.5	0
	Gehen	0	0	0	0	0	5	95

Tabelle A.28: Konfusionmatrix für HMM, Fenstergröße: 32, Features aus 7 Intervallen mit 50 % Überlappung, Struktur: 3 Zustände, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	59	14	16.5	0.5	5.5	4.5	0
	Rolltr.	11.5	39.5	30.5	1.5	9	7.5	0.5
	Ruhez.	15.5	27	49.5	0	6	2	0
	Rennen	2.5	5.5	2.5	81.5	7	1	0
	Runter	7	6.5	9	1.5	73.5	2.5	0
	Hoch	3	6.5	4	0.5	6.5	79.5	0
	Gehen	0.5	2.5	1.5	0	2	2	91.5

Tabelle A.29: Konfusionmatrix für HMM, Neuster Zyklus, Features aus 3 Intervallen mit 50 % Überlappung, Struktur: 3 Zustände, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	73	8.5	13.5	0	4	1	0
	Rolltr.	4.5	49.5	39.5	0	5	1.5	0
	Ruhez.	12	34.5	48.5	0	5	0	0
	Rennen	1	2.5	2	79	14.5	1	0
	Runter	3	10.5	4.5	0.5	79.5	2	0
	Hoch	2.5	4.5	2.5	0	9.5	81	0
	Gehen	0.5	0.5	0	0	3	2.5	93.5

Tabelle A.30: Konfusionmatrix für HMM, Neuster Zyklus, Features aus 5 Intervallen mit 50 % Überlappung, Struktur: 3 Zustände, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	65.5	14.5	19	0	1	0	0
	Rolltr.	18	56.5	22.5	0	2.5	0.5	0
	Ruhez.	17	39	41.5	0	2.5	0	0
	Rennen	0	4	0	86	10	0	0
	Runter	2.5	10.5	4.5	7.5	72	3	0
	Hoch	4.5	5.5	2	0	7	81	0
	Gehen	0.5	0.5	0.5	0	4.5	1.5	92.5

Tabelle A.31: Konfusionmatrix für HMM, Neuster Zyklus, Features aus 7 Intervallen mit 50 % Überlappung, Struktur: 3 Zustände, 1. Ordnung

Rohdaten

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	35.75	21.13	16.63	5.13	10.81	8.5	2.06
	Rolltr.	14	52.63	18.69	5.69	7.12	1.13	0.75
	Ruhez.	24.25	26.75	27.13	4.69	9.38	6.5	1.31
	Rennen	0.81	2.44	0.31	89.75	5.31	1.06	0.31
	Runter	7.5	7.38	2.94	4.44	63.75	10.94	3.06
	Hoch	4.94	0.69	2.38	0.94	17	70.19	3.88
	Gehen	1.63	0.5	0.75	3.13	5.38	12.56	76.06

Tabelle A.32: Konfusionmatrix für HMM, Fenstergröße: 32, Rohdaten, Struktur: 6 Zustände, Mischung aus 3 Gaußverteilungen, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	52.38	16.75	19.5	2.75	5.88	2.5	0.25
	Rolltr.	5.38	60.38	25.37	1.63	7	0.13	0.13
	Ruhez.	19.13	28.88	38.75	3	6.63	3.38	0.25
	Rennen	0.13	0.63	0.25	98.13	0.88	0	0
	Runter	2.25	5.63	3.88	1	85.5	1.63	0.13
	Hoch	0.38	0	0	0	2.5	96.5	0.63
	Gehen	0	0	0	0	0.76	2.66	96.58

Tabelle A.33: Konfusionmatrix für HMM, Fenstergröße: 64 (Körperkoordinaten), Rohdaten, Struktur: 6 Zustände, Mischung aus 3 Gaußverteilungen, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	58	18	12.75	1.5	6.5	3	0.25
	Rolltr.	2.75	79.5	11	0.75	5.75	0.25	0
	Ruhez.	13.5	33	42	1.25	6.5	3	0.75
	Rennen	0	0	0	99.5	0.5	0	0
	Runter	0	5	0.75	0.25	91	2.75	0.25
	Hoch	0	0	0	0	0.5	99.5	0
	Gehen	0	0	0	0	0	1.25	98.75

Tabelle A.34: Konfusionmatrix für HMM, Fenstergröße: 128 (Körperkoordinaten), Rohdaten, Struktur: 6 Zustände, Mischung aus 3 Gaußverteilungen, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	66.5	12	6	2.5	7	5.5	0.5
	Rolltr.	1.5	90.5	6	0	1	0.5	0.5
	Ruhez.	12	28.5	48.5	1	5	5	0
	Rennen	0	0	0	100	0	0	0
	Runter	0	3	1	0	94	2	0
	Hoch	0	0	0	0	0	100	0
	Gehen	0	0	0	0	0	0.5	99.5

Tabelle A.35: Konfusionmatrix für HMM, Fenstergröße: 256 (Körperkoordinaten), Rohdaten, Struktur: 6 Zustände, Mischung aus 3 Gaußverteilungen, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	78	2	10	0	5	5	0
	Rolltr.	0	93	6	0	1	0	0
	Ruhez.	1	25	60	1	7	5	1
	Rennen	0	0	0	100	0	0	0
	Runter	0	2	0	0	98	0	0
	Hoch	0	0	0	0	0	100	0
	Gehen	0	0	0	0	0	0	100

Tabelle A.36: Konfusionmatrix für HMM, Fenstergröße: 512 (Körperkoordinaten), Rohdaten, Struktur: 6 Zustände, Mischung aus 3 Gaußverteilungen, 1. Ordnung

		Vorhergesagt						
		Aufzug	Rolltr.	Ruhez.	Rennen	Runter	Hoch	Gehen
Vorliegend	Aufzug	70	10	8	2	5	5	0
	Rolltr.	2.5	90.5	6	0	1	0	0
	Ruhez.	11	27.5	58.5	0	3	0	0
	Rennen	0	0	0	97.5	2.5	0	0
	Runter	4.5	1	1.5	7	79.5	6.5	0
	Hoch	5	0.5	0.5	0	4	90	0
	Gehen	0.5	0	0	0	0.5	6.5	92.5

Tabelle A.37: Konfusionmatrix für HMM, Fenstergröße: Neuester Zyklus aus 256 (Körperkoordinaten), Rohdaten, Struktur: 6 Zustände, Mischung aus 3 Gaußverteilungen, 1. Ordnung

Literaturverzeichnis

- [1] Practical Guide to Accelerometers. <http://www.sensr.com/>.
- [2] Ling Bao and Stephen S. Intille. Activity Recognition from User-Annotated Acceleration Data. *Pervasive Computing*, pages 1–17, 2004. (Zitiert auf Seite 9)
- [3] L. Breiman, J. H. Friedman, R. Olshen, and C. J. Stone. *Classification and Regression Trees*. Chapman and Hall/CRC, 1984. (Zitiert auf Seite 33)
- [4] Norbert Györbíró, Ákos Fábián, and Gergely Hományi. An Activity Recognition Aystem for Mobile Phones. *Mobile Networks and Applications*, 14:82–91, 2009. (Zitiert auf den Seiten 10 und 35)
- [5] C.W. Han, S.J. Kang, and N.S. Kim. Implementation of HMM-Based Human Activity Recognition Using Single Triaxial Accelerometer. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 93(7):1379–1383, 2010. (Zitiert auf den Seiten 12 und 40)
- [6] Zhenyu He and Lianwen Jin. Activity recognition from acceleration data based on discrete consine transform and svm. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 5041–5044, 2009. (Zitiert auf den Seiten 9 und 10)
- [7] HTC Corporation. *T-Mobile G1 Technical Specification*. <http://www.htc.com/us/-products/t-mobile-g1>. (Zitiert auf Seite 13)
- [8] D.M. Karantonis, M.R. Narayanan, M. Mathie, N.H. Lovell, and B.G. Celler. Implementation of a Real-Time Human Movement Classifier Using a Triaxial Accelerometer for Ambulatory Monitoring. *IEEE Transactions on Information Technology in Biomedicine*, 10(1):156–167, 2006. (Zitiert auf Seite 9)
- [9] Asahi Kasei. AK8976A, Product Anouncement, 2006. <http://www.asahi-kasei.co.jp/asahi/-en/news/2005/e060322.html>. (Zitiert auf den Seiten 13 und 14)
- [10] A.M. Khan, Young-Koo Lee, S.Y. Lee, and Tae-Seong Kim. A Triaxial Accelerometer-Based Physical-Activity Recognition via Augmented-Signal Features and a Hierarchical Recognizer. *IEEE Transactions on Information Technology in Biomedicine*, 14(5):1166–1172, 2010.
- [11] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. Activity Recognition Using Cell Phone Accelerometers. *SIGKDD Explorations Newsletter*, 12:74–82, 2011. (Zitiert auf den Seiten 11 und 45)
- [12] Thorsten Last, Chris Nugent, and Frank Owens. Multi-component Based Cross Correlation Beat Detection in Electrocardiogram Analysis. *BioMedical Engineering OnLine*, 3(1):26, 2004. (Zitiert auf Seite 29)

- [13] David Mizell. Using gravity to estimate accelerometer orientation. In *Proceedings of the IEEE International Symposium on Wearable Computers*, pages 252–253, 2003. (Zitiert auf den Seiten 27 und 28)
- [14] Kevin Murphy. Hidden Markov Model (HMM) Toolbox for Matlab, 1998. <http://www.cs.ubc.ca/~murphyk/~Software/HMM/hmm.html>. (Zitiert auf den Seiten 16 und 49)
- [15] B. Najafi, K. Aminian, A. Paraschiv-Ionescu, F. Loew, C.J. Bula, and P. Robert. Ambulatory System for Human Motion Analysis Using a Kinematic Sensor: Monitoring of Daily Physical Activity in the Elderly. *IEEE Transactions on Biomedical Engineering*, 50(6):711–723, 2003. (Zitiert auf Seite 9)
- [16] Daniel Olguín Olguín and Alex Pentland. Human activity recognition: Accuracy across common locations for wearable sensors. In *Proceedings of the IEEE International Symposium on Wearable Computers*, ISWC 2006, pages 11–13, 2006. (Zitiert auf Seite 10)
- [17] Open Handheld Alliance. *Android Developer Guide*, 2008. <http://developer.android.com>. (Zitiert auf den Seiten 14 und 15)
- [18] Open Handheld Alliance. *Android Open Source Project*, 2008. <http://source.android.com/AndroidQuellcode>. (Zitiert auf Seite 16)
- [19] Geoffroy Peeters. Time variable tempo detection and beat marking. (Zitiert auf Seite 29)
- [20] S.J. Preece, J.Y. Goulermas, L.P.J. Kenney, and D. Howard. A Comparison of Feature Extraction Methods for the Classification of Dynamic Activities From Accelerometer Data. *IEEE Transactions on Biomedical Engineering*, 56(3):871–879, 2009.
- [21] Lawrence R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989. (Zitiert auf den Seiten 37 und 38)
- [22] S. Reddy, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. Determining transportation mode on mobile phones. In *Proceedings of the IEEE International Symposium on Wearable Computers*, pages 25–28, 2008. (Zitiert auf den Seiten 10 und 40)
- [23] Dave Redell. Thinking about accelerometers, 1998. <http://www.lunar.org/docs/LUNARclips-v5/v5n1/Accelerometers.html>.
- [24] Ronit Slyper and Jessica K. Hodgins. Action Capture with Accelerometers. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '08, pages 193–199, 2008. (Zitiert auf Seite 9)
- [25] Jun Yang. Toward physical activity diary: motion recognition using simple acceleration features with mobile phones. In *Proceedings of the 1st international workshop on Interactive multimedia for consumer electronics*, pages 1–10, 2009. (Zitiert auf den Seiten 11, 35 und 45)

Alle URLs wurden zuletzt am 24. Mai 2011 geprüft.

Erklärung

Hiermit versichere ich, diese Arbeit selbständig verfasst und nur die angegebenen Quellen benutzt zu haben.

(Attila Györkös)