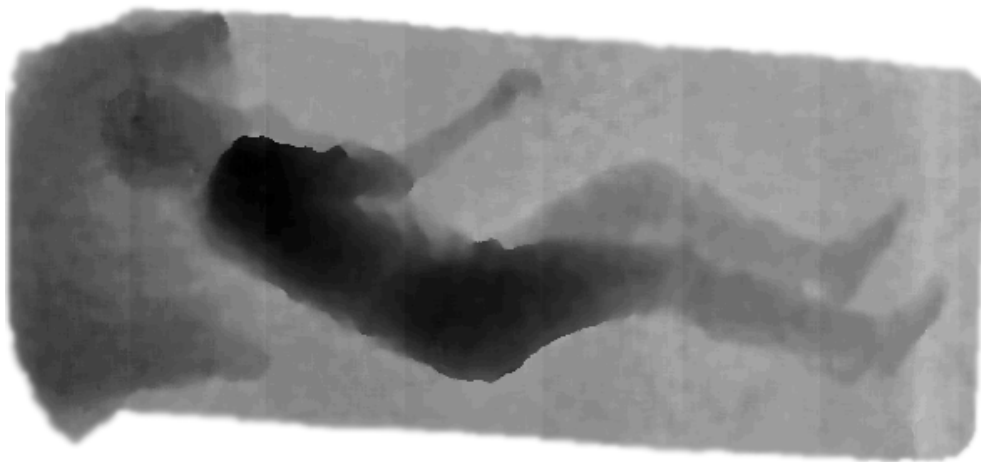


Schlafüberwachung mittels Kinect

Anhand von Tiefenbildern die Bewegungen einer schlafenden Person beurteilen



Semesterarbeit in digitaler Signalverarbeitung

Autoren

Lukas Haldemann und Michael Jost

Betreuer

Prof. Dr. Guido M. Schuster

HSR Hochschule für Technik Rapperswil
23.12.2011

Abstract

Aufgabenstellung

Das Ziel der Arbeit ist die Entwicklung einer Schlafüberwachung mit dem Kinect-Sensor von Microsoft. Anstoss dazu gibt die, in der Pflege von betagten Personen bekannte, Erkrankung an Wundliegen, auch Dekubitus genannt. Der Kinect-Sensor enthält eine Tiefenbildkamera, die eine Verwendung bei Nacht ermöglicht. Anhand der Tiefeninformation sollen die Bewegungen erkannt und auf Relevanz beurteilt werden.

Vorgehen

Mit eigens erstellten Aufnahmen wurde eine repräsentative Datenbank erstellt, in der wichtige relevante Bewegungen abgelegt sind. Dabei wurden vier Grundpositionen festgelegt, in denen sich eine schlafende Person befinden kann; die Bauch-, Halbseiten-, Seiten- und Rückenlage. In der Halbseiten- und Seitenlage muss zwischen links und rechts unterschieden werden. Jede Änderung der Grundposition gilt als relevante Bewegung.

Die Aufnahmen dienen als Grundlage für die Entwicklung eines geeigneten Algorithmus. Bewegungen der Person werden detektiert und bezüglich Art und Relevanz beurteilt. Dazu wird die Grundposition vor und nach der Bewegung bestimmt.

Ergebnisse

Der Algorithmus wurde auf Robustheit und Fehleranfälligkeit getestet, wobei die Bewegungsdetektion äusserst gut funktioniert. Die Erkennung der Seitenlage aus den vier Grundpositionen ist etwas weniger robust, während zwischen linker und rechter Seitenlage zuverlässig unterschieden werden kann. Eine Differenzierung der Rückenlage von der Bauch- und Halbseitenlage wäre nur möglich, wenn die Person nicht unter einer Decke liegen würde. Bauch- und Halbseitenlage können nicht unterschieden werden, da sie sich zu ähnlich sind. Eine zusätzliche Unterscheidung der linken und rechten Halbseitenlage wird damit hinfällig.

Die Beurteilung der Relevanz einer beliebigen Bewegung ist trotzdem möglich. Der dafür gewählte Ansatz ist jedoch noch fehleranfällig.

Ausblick

Es ist möglich, mit dem Kinect-Sensor eine Schlafüberwachung zu realisieren. Die erarbeiteten Ansätze bedürfen aber einer Weiterentwicklung. Lösungen dazu werden in dieser Arbeit aufgezeigt.

Inhaltsverzeichnis

1. Abkürzungen, Definitionen	3
1.1 Abkürzungen	3
1.2 Definitionen	3
2. Einleitung	4
3. Grundlagen	5
3.1 Kinect-Sensor	5
3.1.1 Nachteilige Effekte	6
3.2 Microsoft Kinect SDK	8
3.3 Kinect Recorder	9
3.4 MATLAB	9
3.5 Relevante Bewegungen	10
3.5.1 Liegepositionen	10
3.5.2 Bewegungen	10
3.5.3 Datenbank	11
4. Konzepte	13
4.1 Bewegungen erkennen	13
4.2 Analyse in der Ruhelage	14
4.3 Silhouette auswerten	14
4.4 Rechte und linke Seitenlage unterscheiden	16
5. Implementierung	17
5.1 Programmablauf	17
5.2 Einlesen und Aufbereiten der Tiefendaten	18
5.3 Motion Intensity berechnen	19
5.3.1 Quantisierungsrauschen filtern	19
5.3.2 Kantenrauschen filtern	20
5.4 Motion Intensity auswerten, Bewegungen detektieren	22
5.5 Silhouette ermitteln	24
5.5.1 Verfälschungen der Silhouette korrigieren	24
5.6 Position bestimmen	27
5.6.1 Seitenlage und Flachlage unterscheiden	27
5.6.2 Rechte und linke Seitenlage unterscheiden	28
5.6.3 Bemerkungen zur Flachlage	30
5.7 Bewegungsrelevanz beurteilen	32
5.8 Graphische Benutzeroberfläche	35
6. Tests und Auswertung	37
6.1 Testen des Algorithmus	37
6.2 Kritische Parameter	39

6.3	Beurteilung	39
6.3.1	Stärken	39
6.3.2	Schwächen	40
7.	Schlussfolgerungen und Ausblick	41
8.	Schlusswort	42
9.	Abbildungsverzeichnis	43
10.	Tabellenverzeichnis	44
11.	Literaturverzeichnis	45
12.	Erklärung zur Urheberschaft	46
A	Aufgabenstellung	47

1. Abkürzungen, Definitionen

1.1 Abkürzungen

fps	frames per second
SDK	Software Development Kit
API	Application Programming Interface
NUI	Natural User Interface
GUI	Graphical User Interface
AVI	Audio Video Interleave
MI	Motion Intensity

1.2 Definitionen

Kinect-Sensor

Sensor zur Spielkonsole Xbox von Microsoft. Enthält eine Tiefenbild-Kamera.

Xbox

Spielkonsole aus dem Hause Microsoft.

Framerate

Bildwiederholfrequenz des Videos. Wird in frames per second (fps) angegeben.

Light Coding

Funktionsprinzip der Tiefenbildkamera.

Kinect Recorder

Kleines C# Programm, welches zur Aufnahme von Tiefenbild-Videos des Kinect-Sensors erstellt wurde.

AVI-Video

Video, welches im Audio Video Interleave (AVI) Format abgespeichert wurde. Dies ist ein von Microsoft definiertes Videoformat.

MATLAB

Kommerzielle Software der Firma **The MathWorks**® zur Lösung mathematischer Probleme. Ergebnisse können graphisch dargestellt werden.

2. Einleitung

Problemstellung	In der Pflege von betagten Personen ist das Wundliegen, auch Dekubitus genannt, ein grosses Problem. Bei zu wenig Bewegung im Schlaf können durch die anhaltenden Belastungen der Körperteile Druckstellen entstehen. Die Zellen werden nicht mehr mit den nötigen Nährstoffen versorgt und beginnen abzusterben. In den USA gibt es jährlich 2,5 Millionen Fälle. Die anfallenden Kosten einer Heilung belaufen sich auf ca. 43'000 Dollar. [1]
Bestehende Lösung	Die Firma Compliant Concept, ein Spin-off der ETH Zürich und der EMPA Dübendorf, hat deshalb ein intelligentes Bett entwickelt, welches mittels Drucksensoren die Bewegungen im Schlaf überwacht und die Person bei Bedarf automatisch umlagert. Es soll im Jahr 2012 auf den Markt kommen.
Neuer Ansatz	Ein neuer Ansatz zur Schlafüberwachung besteht darin, die Bewegungen mit einer Tiefenbildkamera zu erkennen. Diese Lösung wäre wesentlich kostengünstiger als eine Druckmessung am Lattenrost des Bettes. Es müsste lediglich eine solche Kamera über dem zu überwachenden Bett angebracht werden.
Ziel der Arbeit	Ziel dieser Arbeit ist es, diesen Ansatz umzusetzen. Dabei wird untersucht, wie gut die Bewegungen erkannt werden können. Insbesondere müssen relevante von nicht relevanten Bewegungen unterschieden werden. Auch gilt es herauszufinden, ob dieser Ansatz funktioniert, wenn die schlafende Person unter einer Decke liegt. Am Ende soll eine Aussage gemacht werden, ob mit diesem Lösungsansatz eine schlafende Person zuverlässig überwacht werden kann.
Kinect-Sensor	Für die Arbeit wird der Kinect-Sensor von Microsoft verwendet, welcher für die Spielkonsole Xbox entwickelt wurde. Er ist in der Lage ein Tiefenbild zu erstellen, welches für jedes Pixel die Entfernung des Objektes zur Kamera enthält. Das Tiefenbild wird im Infrarot-Bereich erstellt, was den Gebrauch bei Dunkelheit ermöglicht.
Software als Endprodukt	<p>Eine Software wertet den Datenstrom der Kamera aus und visualisiert die Resultate. Bei der Analyse der Daten wird die Liegeposition der Person ermittelt und zwischen relevanten und nicht relevanten Bewegungen unterschieden. Relevante Bewegungen führen zu einer Positions- oder Druckveränderung der Person, was zum Beispiel mit einer Drehung vom Bauch auf den Rücken geschieht. Eine nicht relevante Bewegung wäre beispielsweise eine Drehung der Hand.</p> <p>Die Entwicklung geeigneter Algorithmen ist ein zentraler Teil dieser Arbeit. Dafür wird eine annotierte Datenbank mit Aufnahmen diverser Schlafbewegungen erstellt. Da nur mit wenigen, selbst erstellten Aufnahmen gearbeitet werden kann, ist eine Beurteilung des Algorithmus für den breiten Einsatz nur bedingt möglich.</p>

3. Grundlagen

3.1 Kinect-Sensor

Der Kinect-Sensor von Microsoft ist eigentlich ein Zusatz für die Xbox. Er ermöglicht, Videospiele mit den Bewegungen des eigenen Körpers zu steuern. Der Sensor findet in dieser Arbeit Verwendung, da er Tiefenbilder aufnehmen kann und im Vergleich zu anderen Tiefenbildkameras sehr kostengünstig ist. Er enthält ein Mikrofon-Array, eine normale Kamera (mittlere Linse) und eine Tiefenbildkamera (linke und rechte Linse). Zudem enthält er einen Motor um die Neigung des Sensors zu verstellen. Für die Anwendung in einer Schlafüberwachung werden jedoch ausschliesslich die Informationen der Tiefenbildkamera benötigt.

Verwendung



Abbildung 3.1: Kinect-Sensor [2]

Das Funktionsprinzip der Tiefenbildkamera nennt sich Light Coding. Mit einem Laserstrahl und einem Filter (linke Linse) wird ein halb zufälliges Muster von kleinen Punkten im Infrarotbereich auf die Umgebung projiziert. Das reflektierte Muster wird von einer Infrarot-Kamera (rechte Linse) eingelesen. Aus dem gesendeten und empfangenen Licht-Muster, der Linsenverzerrung und dem räumlichen Abstand zwischen Sender und Empfänger kann die Distanz für jedes Pixel berechnet werden. Der Kinect-Sensor liefert für jedes Pixel den Abstand von der Sensorfläche zum Objekt in Millimeter.

Funktionsprinzip

Berechnete Distanz

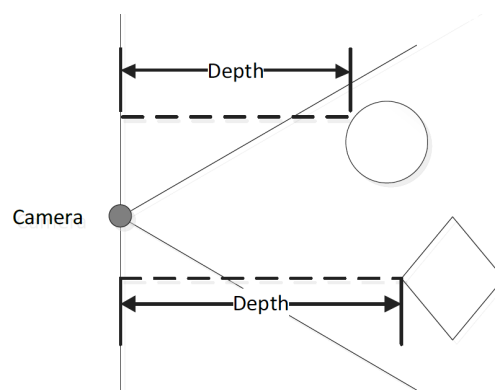


Abbildung 3.2: Berechnete Distanz [3]

Technische Daten
zur
Tiefenbildkamera

Sensor item	Specification range
Range	1.2 - 3.5 meters
Viewing angle	43° vertical, 57° horizontal
Frame rate (depth stream)	30 frames per second (fps)
Resolution (depth stream)	640 x 480 pixels

Tabelle 3.1: Technische Daten [3]

Der Messbereich des Sensors ist effektiv noch grösser als vom Hersteller angegeben. Mit dem Microsoft Kinect SDK konnten Distanzen zwischen 0.8 m und 3.975 m gemessen werden. Darüber wie auch darunter wird der Wert Null ausgegeben.

3.1.1 Nachteilige Effekte

Reflexionen

In Abb. 3.3 ist ein Tiefenbild in Graustufen dargestellt. Man kann sehen, dass die Tiefenbildkamera nicht alle Objekte erkennen kann. Glatte Oberflächen werden nicht erkannt, weil der ausgesendete Laserpunkt nicht zur Infrarot-Kamera reflektiert wird. Es sei denn, die glatte Oberfläche stehe ungefähr rechtwinklig zum Sensor. Dies ist anhand der beiden Monitoren gut zu sehen. Der Linke steht senkrecht zur Kamera und wird erkannt, der Rechte hingegen nicht. Transparente Objekte, wie die Glaswand im Hintergrund werden ebenfalls nicht erkannt. Dies, weil der Laserstrahl durch das Objekt hindurch geht.



Abbildung 3.3: Tiefen- und Videobild

Kleine Objekte

Mit kleinen Objekten hat die Kamera Mühe und kann diese nicht erkennen. Eine minimale Fläche ist notwendig, damit die Objekte korrekt erfasst werden. Bei einer schlafenden Person macht sich das bei den Haaren stark bemerkbar, weil diese keine kontinuierliche Fläche bilden. Die Kamera erkennt dann gar nichts (weisser Fleck), wie in Abb. 3.6 beispielhaft zu sehen ist.

Schmaler Rand

Die Bilder weisen rechts aussen einen Rand von 8 Pixel Breite auf. Die Werte dort liegen alle bei Null. Der Grund für diesen Rand ist nicht bekannt. Er wird die Auswertung der Tiefendaten aber nicht stören. In Abb. 3.9 ist der Rand gut zu erkennen.

Die Tiefenwerte des Sensors weisen einen beträchtlichen Quantisierungsfehler auf. Je weiter ein Objekt von der Kamera entfernt ist, desto grösser wird die Auflösung respektive der Quantisierungsfehler. Die Auflösung Δz kann in Abhängigkeit der Tiefe z approximiert werden. [4]

Quantisierungsfehler

$$\Delta z \approx 0.0033z^2 - 0.0017z \quad (3.1)$$

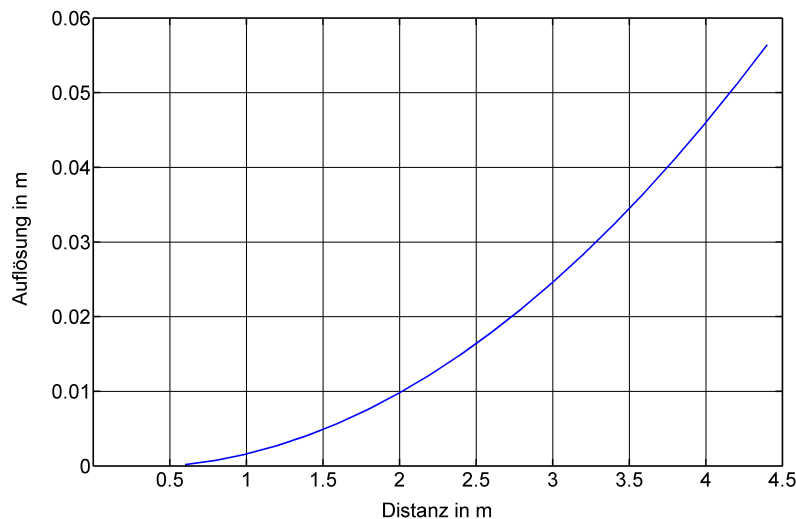


Abbildung 3.4: Tiefenauflösung

Die Tiefenwerte springen demnach wegen der beschränkten Auflösung je nach Tiefe mehr oder weniger stark hin und her. Für ein einzelnes Pixel sieht das über die Zeit aus wie in Abb. 3.5. Wie oft das Pixel springt ist unterschiedlich. Es kann auch Pixel geben, die nicht springen. Ein weiteres Phänomen des Sensors macht sich durch

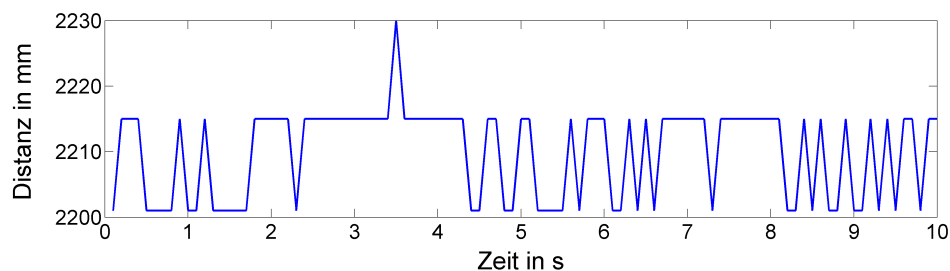


Abbildung 3.5: Tiefensprünge

feine vertikale Linien im Tiefenbild bemerkbar, wie sie in Abb. 3.6 zu erkennen sind. Links und rechts der Linien wird nicht die gleiche Höhe gemessen, obwohl die reale Oberfläche keine solche Sprünge aufweist. Die Position und die Anzahl der Linien sind nicht konstant und können während der Laufzeit ändern. Es scheint jedoch, als wären sie immer gleichmässig über das Bild verteilt. Der Grund für die Linien ist nicht bekannt. Sie gehören aber zum Sensor dazu und sind kein Problem im Sinne einer fehlerhaften Herstellung. Für die Schlafüberwachung stören sie nicht weiter. Die Höhenunterschiede bei den Linien entsprechen dem Quantisierungsfehler des Sensors. Damit muss die Software umgehen können.

Vertikale Linien

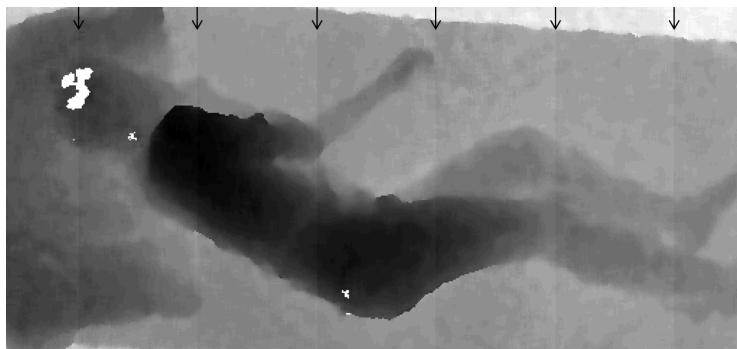


Abbildung 3.6: Vertikale Linien mit Pfeilen markiert

3.2 Microsoft Kinect SDK

Application Programming Interface

Microsoft stellt ein eigenes Software Development Kit (SDK) zur Verfügung, welches ein Application Programming Interface (API) für die Verwendung des Kinect-Sensors enthält. Mit dem API können die Videodaten, Tiefendaten und Audiodaten ausgelesen und der Motor der Kamera angesteuert werden. Für diese Arbeit werden ausschliesslich die Tiefendaten verwendet.

Die Funktionen, welche für die Ansteuerung der Kamera verwendet werden, sind in der Natural User Interface (NUI) Library definiert. Sie stellt die Schnittstelle zur Applikation dar.

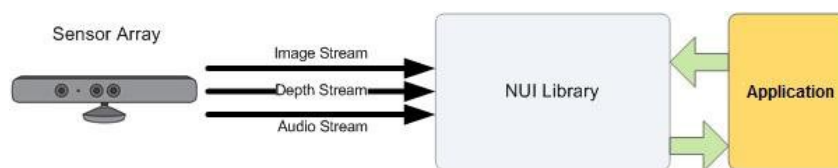


Abbildung 3.7: NUI Library [3]

Das API kann mit den Programmiersprachen C++ oder C# verwendet werden. Da die Inbetriebnahme des Sensors mit C++ wesentlich aufwändiger ist als mit C#, wurde letztere gewählt. Das SDK bietet verschiedene Einstellmöglichkeiten, vor allem betreffend Auflösung und Datenformat. Im Folgenden ist beschrieben, in welcher Form die Daten ausgelesen wurden.

Format der Tiefendaten

Es wurde die höchst mögliche Auflösung von 640 x 480 Pixel verwendet. Ein Down-sampling ist im Nachhinein immer noch möglich. Die Pixel sind von links nach rechts und von oben nach unten zeilenweise nummeriert.

Der Kinect-Sensor sendet die Tiefendaten jedes einzelnen Pixels in einem Array, das Einträge der Grösse eines Bytes enthält. Der Wert zweier Bytes, also 16 Bits, stellt die Tiefe eines Pixels in Millimeter dar, wobei die obersten vier Bits Null sind.



Abbildung 3.8: Format der Tiefenwerte

3.3 Kinect Recorder

Um mit dem Sensor Aufnahmen zu machen, wurde ein Programm namens Kinect Recorder erstellt. Das Programm ist in C# geschrieben und verwendet das Microsoft Kinect SDK. Die Daten werden vom Sensor ausgelesen und fortlaufend in eine dat-Datei geschrieben; also Pixel für Pixel, Frame für Frame, ohne Zusatzinformationen. Für die weitere Bearbeitung der Daten können diese wieder aus der dat-Datei ausgelesen werden.

Programm für
Aufnahmen

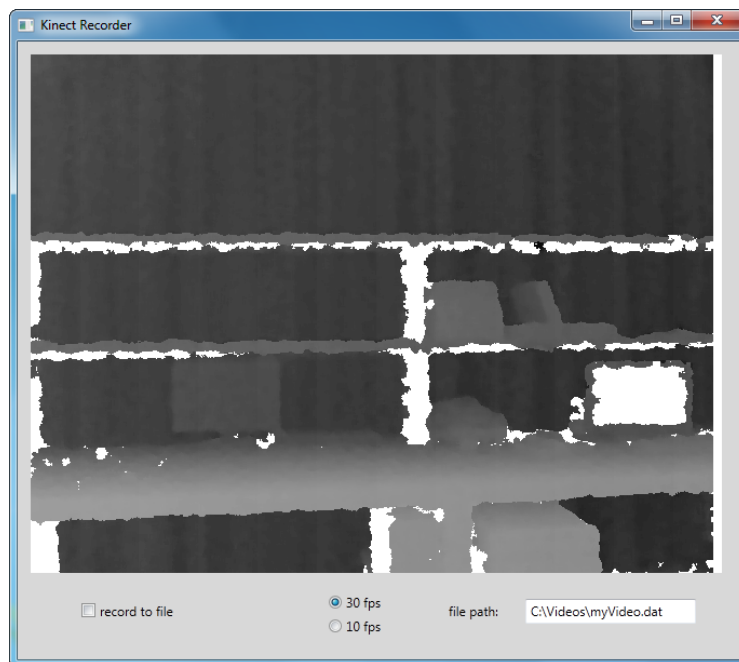


Abbildung 3.9: Kinect Recorder

3.4 MATLAB

Für die Auswertung der Tiefenbilder wurde MATLAB R2011a inklusive Image Processing Toolbox verwendet. Der Grund dafür ist, dass die Tiefenbilder in Form von zweidimensionalen Matrizen sehr einfach verarbeitet werden können. MATLAB stellt viele Funktionen zur Verfügung, welche für die Auswertung der Daten von grossem Nutzen sind. Zudem ermöglicht es Signale, Vektoren oder auch Bilder graphisch darzustellen. Dadurch wird die Entwicklung und Implementierung der Schlafüberwachung wesentlich effizienter als mit einer Hochsprache wie C#. Der einzige Nachteil besteht darin, dass es nicht möglich ist einen Live-Stream vom Sensor einzulesen. Die Verarbeitung erfolgt also nur aus gespeicherten Aufnahmen.

Verwendete
Software

3.5 Relevante Bewegungen

Definition

Es ist zweckmässig, jene Bewegungen als relevant zu definieren, welche eine Veränderung der Druckbelastung auf den Körper bewirken. Wenn beispielsweise nur ein Arm bewegt wird, entlastet dies wohl den Arm bzw. ändern seine Druckstellen, der Rest des Körpers bleibt jedoch in der gleichen Position. Somit ist er immer noch den gleichen Belastungen ausgesetzt. Bei einer Drehung des Körpers hingegen werden ganze Teile entlastet, wodurch sich diese erholen können.

Dekubitusprophylaxe

Um eine Schädigung der Zellen infolge Dekubitus zu vermeiden, ist das Umlagern der Person der wohl wichtigste Aspekt. Dazu genügen grundsätzlich kleine Umlagerungen von wenigen Graden in regelmässigen Abständen. Dazu werden Lagerungshilfsmittel, wie spezielle Kissen, verwendet. Bei einer schlafenden Person sind solche Mikroumlagerungen aber nicht möglich. Ohne Hilfsmittel wird sie nur in bestimmten Grundpositionen liegen.

3.5.1 Liegepositionen

Für eine schlafende Person wurden folgende vier Grundpositionen definiert:



(a) Bauchlage



(b) Halbseitenlage links / rechts



(c) Seitenlage links / rechts



(d) Rückenlage

Abbildung 3.10: Liegepositionen

Die Halbseitenlage wird durch das angewinkelte Bein charakterisiert. Diese Lage wurde deshalb dazugenommen, weil der Brustkorb im Vergleich zur Bauchlage entlastet wird. Ebenso erfährt die Hüfte im Vergleich zur Seitenlage eine Entlastung.

3.5.2 Bewegungen

Relevante Bewegungen

Für die Analyse sind aus den vier Liegepositionen alle möglichen Positionswechsel bestimmt worden. Sie stellen zugleich die relevanten Bewegungen dar. Die Erkennung

soll für die linke und rechte Seitenlage genau gleich funktionieren. Folgende relevante Bewegungen wurden definiert:

Bauchlage	⇔	Halbseitenlage links / rechts
Bauchlage	⇔	Seitenlage links / rechts
Bauchlage	⇔	Rückenlage
Rückenlage	⇔	Seitenlage links / rechts
Rückenlage	⇔	Halbseitenlage links / rechts
Halbseitenlage links / rechts	⇔	Seitenlage links / rechts
Halbseitenlage links	⇔	Halbseitenlage rechts
Seitenlage links	⇔	Seitenlage rechts

Tabelle 3.2: Bewegungen

In den einzelnen Liegepositionen können zusätzlich nicht relevante Bewegungen ausgeführt werden. Diese müssen von den relevanten Bewegungen unterschieden werden. Zu den nicht relevanten Bewegungen gehören:

Nicht relevante
Bewegungen

- Leichtes hin und her Drehen des Körpers
- Bewegungen der Beine und Arme
- Bewegungen des Kopfes

3.5.3 Datenbank

Für die Entwicklung des Algorithmus wurde eine Datenbank aus Tiefenbilddaufnahmen erstellt. Der Grund für das Erstellen einer Datenbank liegt darin, dass für das Testen der Software stets die gleichen Daten verwendet werden. Somit kann beurteilt werden wie sich der Algorithmus verbessert hat. Wenn stattdessen mit Live-Bildern vom Kinect-Sensor gearbeitet würde, könnten die verschiedene Versionen und Ansätze nicht gut verglichen werden.

Sinn und Zweck

Für die Aufnahmen wurden einige repräsentative, relevante Bewegungen ausgeführt. Mit den Videodaten wurde anschliessend die Datenbank erstellt. Dabei sind gewöhnliche Matratzen verwendet worden. Zusätzlich wurden zwei Aufnahmen bei 'realem' Schlafen während einer Nacht gemacht. Es stellte sich aber heraus, dass dabei nur selten Bewegungen stattfanden. Die Aufnahmen wurden deshalb nicht verwendet.

Art der Aufnahme

Für die Speicherung der Tiefendaten wurde der Kinect Recorder verwendet. Da zum Aufnahmezeitpunkt noch nicht bekannt war, welche Auflösung und welche Framerate benötigt würde, sind alle Bewegungen mit maximaler Auflösung von 640 x 480 Pixel und maximaler Framerate von 30 fps aufgenommen worden. Dies ergab eine beträchtliche Datenmenge:

Format

$$\text{Datenmenge} = 640 * 480 \text{ Pixel} * 2 \text{ Bytes/Pixel} * 30 \text{ fps} \approx 1 \text{ GByte/min}$$

Zusätzlich zu den dat-Dateien wurden mit Hilfe eines MATLAB-Skriptes aus den Rohdaten AVI-Videos in Graustufen erzeugt. Dabei musste aus der Distanz eine Graustufe berechnet werden. Damit die liegende Person im Video möglichst gut erkannt werden kann, wurde die Farbauflösung nur im Distanzbereich der Person gemacht.

AVI-Video

Alle Punkte, die weiter weg sind, werden schwarz angezeigt, alle näheren Punkte weiss. Das MATLAB-Skript heisst *binary2avi.m*.

Inhalt

Die Bewegungen wurden einzeln abgespeichert. Vor und nach der Bewegung wurde jeweils ruhig gelegen. Meistens sind in der Aufnahme auch nicht relevante Bewegungen enthalten. Zusätzlich wurde ein kompletter Bewegungsablauf der einzelnen Bewegungen aufgenommen. Es sind längst nicht alle möglichen Bewegungen in der Datenbank enthalten, die vorhandenen sollten für die Entwicklung des Algorithmus aber ausreichen. Diese gestellten Aufnahmen wurden sowohl mit als auch ohne Decke gemacht. Die Grösse der gesamten Datenbank beträgt ca. 24 GByte.

Bewegungen ohne Decke			Länge
Bauchlage	⇒	Halbseitenlage links	25s
Halbseitenlage links	⇒	Seitenlage links	40s
Seitenlage links	⇒	Halbseitenlage links	40s
Halbseitenlage links	⇒	Bauchlage	27s
Bauchlage	⇒	Seitenlage rechts	18s
Seitenlage rechts	⇒	Rückenlage	35s
Rückenlage	⇒	Seitenlage rechts	16s
Seitenlage rechts	⇒	Bauchlage	40s
Bauchlage	⇒	Rückenlage	25s
Rückenlage	⇒	Bauchlage	38s
Bewegungsablauf			310s

Tabelle 3.3: Datenbank ohne Decke

Bewegungen mit Decke			Länge
Bauchlage	⇒	Halbseitenlage links	44s
Halbseitenlage links	⇒	Seitenlage links	35s
Seitenlage links	⇒	Halbseitenlage links	60s
Halbseitenlage links	⇒	Bauchlage	18s
Bauchlage	⇒	Seitenlage rechts	69s
Seitenlage rechts	⇒	Rückenlage	32s
Rückenlage	⇒	Seitenlage rechts	50s
Seitenlage rechts	⇒	Bauchlage	45s
Bauchlage	⇒	Rückenlage	17s
Rückenlage	⇒	Bauchlage	19s
Bewegungsablauf			355s

Tabelle 3.4: Datenbank mit Decke

4. Konzepte

In diesem Kapitel werden die Lösungskonzepte zur Erkennung von relevanten Bewegungen vorgestellt. Die konkrete Umsetzung dieser Konzepte ist in Kapitel 5. beschrieben.

Einleitung

Gute Konzepte sollen folgende Bedingungen erfüllen:

- Sie müssen problemorientiert sein.
- Sie müssen ein komplexes Problem abstrahieren und es auf ein einfacheres zurückführen.
- Sie müssen robust sein, das heisst sie funktionieren in den allermeisten Fällen und sind nicht fehleranfällig.

Beim Finden von Konzepten wurde auf diese Bedingungen geachtet. Im Folgenden sind auch Ansätze erwähnt, welche diesen Bedingungen nicht gerecht und deshalb nicht in Betracht gezogen wurden.

4.1 Bewegungen erkennen

Für die Erkennung einer relevanten Bewegung ist es wichtig zu wissen, wann die Person sich bewegt und wann sie in Ruhe ist. Zudem soll die Bewegungsintensität, also die Stärke der Bewegung, gemessen werden. Für diese Überlegungen ist die Art der Bewegung irrelevant.

Es braucht also ein Mass für die Bewegungsaktivität einer Person. Dazu wird die Differenz zweier aufeinander folgenden Bilder berechnet. Es entsteht ein Differenzbild. Wichtig ist zu erwähnen, dass die absolute Differenz bestimmt wird. Dies, weil für das Mass der Bewegungsaktivität die Bewegungsrichtung irrelevant ist. In Abb. 4.1 kann erahnt werden, ob und wie stark sich eine Person bewegt. Dunkle Stellen markieren dabei Orte mit grosser Differenz.

Berechnung des Differenzbildes

Von diesen Differenzwerten kann noch nicht auf die Intensität einer Bewegung geschlossen werden. Bewegungen, die punkto Schnelligkeit und Grösse des bewegenden Objekts identisch sind können unterschiedliche Resultate liefern.

Intensität bestimmen

Bewegungen parallel zur Kamera ergeben an den Rändern sehr grosse Differenzen, während sie bei Bewegungen senkrecht zur Kamera eher klein ausfallen. Aufgrund dieser Überlegung soll die Anzahl aller Pixel, die sich geändert haben, berechnet werden. Daraus ergibt sich eine Zahl, die beschreibt, wie gross die Fläche ist, die sich bewegt hat. Im Folgenden heisst diese Zahl Motion Intensity (MI).

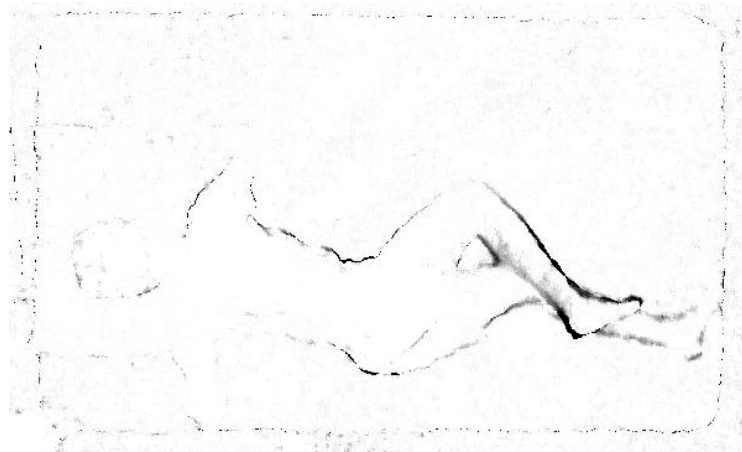


Abbildung 4.1: Differenzbild während einer Bewegung

4.2 Analyse in der Ruhelage

Analyse in der Ruhelage	<p>Wie in Abschnitt 3.5 beschrieben, sind relevante Bewegungen von nicht relevanten zu unterscheiden. Mit einer relevanten Bewegung ist immer eine Positionsänderung, respektive eine Änderung der Druckstellen, verbunden.</p> <p>Es ist also nicht von Interesse, welche Bewegungen eine Person macht. Viel mehr interessiert die Liegeposition einer Person vor und nach der Bewegung.</p>
Vergleich vorher / nachher	<p>Diese Überlegung impliziert folgende Feststellung:</p> <p>Kann der Start und das Ende einer Bewegung erkannt werden, müssen die Liegepositionen vor und nach der Bewegung, das heisst in der Ruhelage, verglichen werden. Liegt die Person nach der Bewegung in einer anderen Position war die Bewegung relevant, andernfalls war sie nicht relevant. Die Abläufe während einer Bewegung werden nicht analysiert. Dies schliesst ein Verfolgen bestimmter Körperteile, auch 'Tracking' genannt, aus.</p>
Kein Tracking	<p>Grundsätzlich wäre es sehr nützlich, durch 'Tracking' die Position wichtiger Körperteile wie Kopf oder Schulter ständig zu wissen. Die grosse Gefahr besteht aber darin, dass der Algorithmus das Körperteil 'verlieren' kann und danach nie mehr 'findet', weil er es nur in einem lokalen Bereich, abhängig von der letzten Position, sucht. Dies wäre für diese Anwendung verheerend.</p> <p>Es ist nun entscheidend, dass Beginn und Ende einer Bewegung sauber erkannt werden können. Dazu wird der Wert der MI ausgewertet. Überschreitet sie eine Eintrittsschranke, markiert dies den Anfang der Bewegung. Unterschreitet sie eine Austrittsschranke, markiert dies das Ende der Bewegung. Mit einer Hysterese kann schnelles Aus- und wieder Einschalten verhindert werden.</p>

4.3 Silhouette auswerten

Silhouette als Werkzeug	<p>Für die Erkennung der Liegeposition einer Person gilt es, typische Eigenschaften des Menschen auszunützen. Von Auge kann aus einem Tiefenbild sehr genau bestimmt werden wie eine Person liegt. Es braucht ein Werkzeug mit dem diese Eigenschaften</p>
-------------------------	--

analysiert werden können. Dazu ist die Silhouette einer Person sehr gut geeignet. Darunter ist folgendes zu verstehen:

Man schaut eine liegende Person von der Seite an und folgt stets ihrer maximalen Höhe. Es wird also ein eindimensionales Höhenprofil der Person erstellt. Abb. 4.2 zeigt eine Silhouette in Seitenlage.

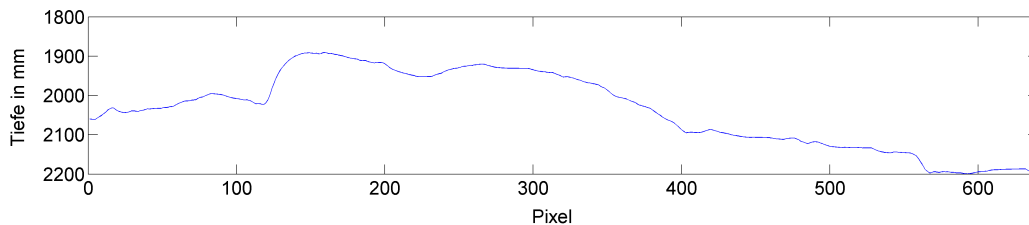


Abbildung 4.2: Silhouette in Seitenlage

Mit Hilfe der Silhouette sollen nun typische Merkmale einer Person in einer bestimmten Liegeposition erkannt werden. In Abschnitt 3.5 sind die verschiedenen Liegepositionen definiert. Als erstes interessiert, ob sich eine Person in Seitenlage oder in einer Flachlage (Bauch-, Halbseiten- oder Rückenlage) befindet. Es gibt ein typisches Merkmal, welches diese beiden Positionen unterscheidet:

In der Seitenlage liegt die Schulter höher als der Kopf. In Flachlage hingegen sind sie auf ähnlicher Höhe.

Seiten- oder Flachlage

Für die Unterscheidung der Rückenlage von der Bauch- oder Halbseitenlage sind verschiedene Ansätze denkbar. Es scheinen sich zwei Merkmale aus der Silhouette anzubieten:

Erkennung der Rückenlage

- Das Gesicht einer Person
- Die Füße sind nach oben gerichtet

Es ist jedoch nicht möglich, anhand dieser beiden Merkmale eine klare Aussage zu machen. Dafür sind sie zu schlecht erkennbar. Auf diese Probleme wird in Abschnitt 5.6.3 näher eingegangen. Ist bekannt, in welcher Seitenlage sich die Person vor der Bewegung befindet, könnte auch aus der Drehrichtung während der Bewegung auf die Rückenlage geschlossen werden. Das Feststellen der Drehrichtung erweist sich aber als sehr schwierig. Die Person kann vor der Drehung noch in die andere Richtung wippen oder sich während der Bewegung auf der Matratze verschieben. Eine Differenzierung der Halbseiten- von der Bauchlage scheint auf diesem Wege ebenfalls nicht möglich zu sein.

Es muss also eine globale Lösung angestrebt werden, die den ganzen Menschen und nicht einzelne Details analysiert. Dazu ist der folgende Ansatz möglich:

Vergleich der Silhouetten

Die Silhouetten einer Person vor und nach einer Bewegung werden verglichen. Sind sie stark verschieden, war die Bewegung relevant. Sind sie ähnlich, war die Bewegung nicht relevant.

Es braucht ein Mass für die Ähnlichkeit zweier Silhouetten. Dazu wird das mittlere Fehlerquadrat berechnet. Damit kann bestimmt werden, ob sich die Person relevant bewegt hat. Eine Aussage, ob sich die Person auf dem Rücken, auf dem Bauch oder in Halbseitenlage befindet, kann jedoch nicht gemacht werden.

4.4 Rechte und linke Seitenlage unterscheiden

Eine Person kann sich in zwei verschiedenen Seitenlagen befinden. Entweder geht die Blickrichtung von der Kamera aus gesehen nach rechts oder nach links. Ein typisches Merkmal der Seitenlage ist, dass die Beine gegenüber dem Oberkörper angewinkelt sind. Andernfalls wäre die Seitenlage instabil, man würde in eine Flachlage 'rollen'. Dieser Winkel kann für die Erkennung der Blickrichtung ausgenutzt werden. Öffnet er sich nach rechts, geht die Blickrichtung nach rechts. Öffnet er sich nach links geht die Blickrichtung nach links. Abb. 3.10(c) zeigt eine Seitenlage mit Blickrichtung nach rechts. Der besagte Winkel ist deutlich erkennbar.

Auch in der Halbseitenlage könnte dieser Winkel wohl für die links/rechts Unterscheidung herangezogen werden. Da jedoch die Halbseitenlage nicht erkannt werden kann, erübrigt sich eine genauere Analyse.

5. Implementierung

5.1 Programmablauf

Aus den in Kapitel 4. beschriebenen Konzepten ergibt sich der Programmablauf. Die einzelnen Teilblöcke sind in den folgenden Abschnitten genauer beschrieben. Zu beachten gilt es, dass die Silhouette eigentlich nur vor und nach einer Bewegung gebraucht würde. Sie wird aber für jedes Frame ermittelt, um sie in der graphischen Benutzeroberfläche anzuzeigen.

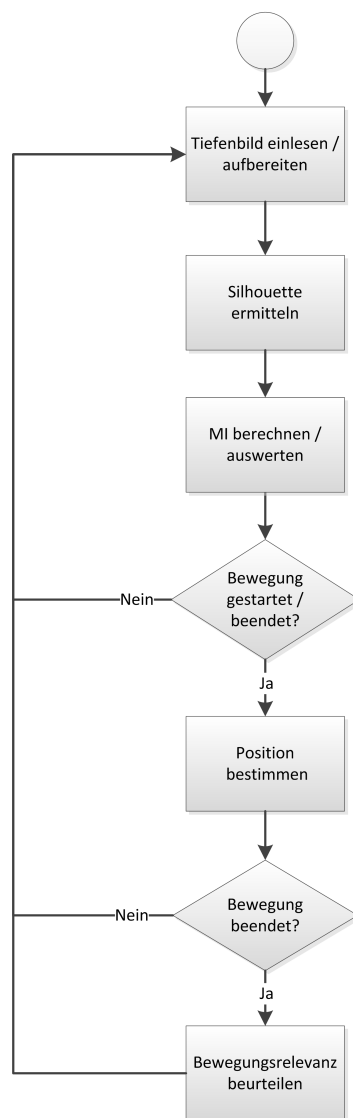


Abbildung 5.1: Programmablauf

5.2 Einlesen und Aufbereiten der Tiefendaten

Framerate

Die Tiefenbilder werden frameweise aus einer dat-Datei, welche nur die Rohdaten enthält, eingelesen. Ist eine tiefere Framerate gewünscht, wird jeweils eine gewisse Anzahl Frames beim Einlesen übersprungen. Während der Entwicklung wurde festgestellt, dass eine Framerate von 30 fps für diese Anwendung zu hoch ist. Eine schlafende Person führt keine so schnellen Bewegungen aus. Zudem wären so nur sehr kleine Änderungen zwischen zwei Frames erkennbar, selbst wenn die Person in Bewegung ist. 10 fps scheint für diese Anwendung ein vernünftiger Wert zu sein. Damit können schnelle Bewegungen problemlos erkannt werden. Die Änderungen zwischen zwei Frames werden zudem deutlicher.

Glättung über die Zeit

Weil die Tiefenwerte, wie in Abb. 3.5 ersichtlich, stark springen, wird auf jedes Pixel ein rekursives, lineares Filter angewendet, um die Bilder über die Zeit etwas zu glätten. Dadurch werden die Bilder während der Bewegung 'verschmiert'. Da sie nur vor und nach einer Bewegung ausgewertet werden, spielt dies keine Rolle. Bis zur Auswertung des Bildes *Im* nach der Bewegung ist dieser Effekt bereits wieder abgeklungen. Der Vorteil ist jedoch, dass das geglättete Bild *ImSmooth* für die Auswertung der Silhouette verwendet werden kann. Sie wird dadurch keine allzu grossen Sprünge mehr aufweisen.

$$ImSmooth_{nT} = ImSmooth_{nT-T} + \alpha \cdot (Im_{nT} - ImSmooth_{nT-T}) \quad (5.1)$$

mit $\alpha = 0.2$

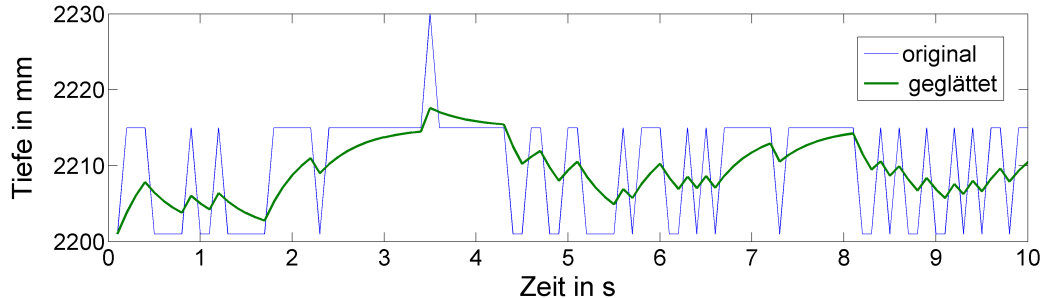


Abbildung 5.2: Glättung eines Pixels über die Zeit

Ungültige Werte ersetzen

Aufgrund der in Abschnitt 3.1.1 beschriebenen Effekte können immer wieder ungültige Tiefenwerte mit dem Wert Null auftreten. Für die Auswertung hätte dies verheerende Folgen. Durch die Glättung würden an diesen Stellen sehr kleine Tiefenwerte, die jedoch grösser als Null sind, auftreten. Weil für die Silhouette aber die kleinsten Tiefenwerte von Interesse sind, ergäbe sich ein verfälschtes Resultat. Deshalb werden vor der Glättung alle Pixel mit dem Wert Null auf den grössten Tiefenwert des gesamten Bildes gesetzt. Dies stimmt noch immer nicht mit dem realen Tiefenbild überein, die Auswertung wird jedoch nicht mehr gestört.

5.3 Motion Intensity berechnen

Als Grundlage für die Berechnung der MI dient das Differenzbild. Es wird gebildet, indem die Differenz Δh der Tiefenwerte h für jedes einzelne Pixel zweier aufeinander folgenden Frames berechnet wird.

Differenzbild
berechnen

$$\Delta h = |h_{nT} - h_{nT-T}| \quad (5.2)$$

Dabei wird immer der Betrag berechnet. Dies, weil das Vorzeichen keine brauchbare Aussage liefert. Bei Bewegungen senkrecht zum Sensor zeigt das Vorzeichen zwar korrekterweise an, ob zum Sensor hin oder davon weg bewegt wird. Bei Bewegungen parallel zum Sensor hingegen befindet sich auf der vorderen Seite des bewegenden Objekts der Bereich mit negativem Vorzeichen und an der hinteren Seite des Objekts der Bereich mit positivem Vorzeichen. In der jeweiligen Bewegung hat das Vorzeichen also eine andere Bedeutung. Es wird deshalb ignoriert.

Der Sensor liefert für Pixel, bei welchen die Tiefenwerte nicht bestimmt werden konnten, den Wert Null. Diese ungültigen Pixel verfälschen das Differenzbild. Darum werden vorgängig alle ungültigen Pixel im Tiefenbild mit einem grossen Wert versehen, der weit ausserhalb des Messbereichs des Sensors liegt, zum Beispiel 8000 Millimeter. Wenn nun ein Pixel im ersten Frame ungültig war und im nächsten gültig, erscheint es im Differenzbild als sehr grosser Wert. Diese unrealistisch hohen Werte werden im Differenzbild gefiltert. So haben ungültige Werte keinen Einfluss mehr.

Ungültige
Tiefenwerte filtern

5.3.1 Quantisierungsrauschen filtern

Der Sensor liefert Tiefenwerte, die stark quantisiert sind. Dies bewirkt grosse Sprünge. Es stellt sich ein Quantisierungsrauschen ein. Der Quantisierungsfehler nimmt mit zunehmender Tiefe zu; siehe dazu Abschnitt 3.1.1. Dieses Rauschen der Tiefenwerte verfälscht das Differenzbild. Als einfache Lösung werden deshalb alle Differenzwerte, die unterhalb des Quantisierungsfehlers liegen, gefiltert.

Quantisierungs-
rauschen filtern

Diese Vorgehensweise hat Nebeneffekte. So werden auch Differenzwerte, die nicht auf das Quantisierungsrauschen zurückzuführen sind, gefiltert. Solche tiefen Differenzwerte sind vor allem bei sehr langsamen oder schwachen Bewegungen zu beobachten. Diese werden also nicht erkannt. Weiter werden flache Bereiche im Tiefenbild, die sich parallel zur Kamera verschieben, nicht erkannt, da die Höhendifferenz zweier Pixel sehr klein ist. Diese ungewollten Nebeneffekte werden in Kauf genommen.

Nebeneffekte

Für die Filterung der Werte muss der Quantisierungsfehler bekannt sein. Mit der Formel 3.1 kann der Quantisierungsfehler in Abhängigkeit der Tiefe approximativ berechnet werden. Dafür muss eine typische Tiefe festgelegt werden. Es wird von einem Worst-Case-Szenario ausgegangen, sodass der Quantisierungsfehler mit grosser Sicherheit gefiltert wird. Es wird angenommen, dass ein Bett maximal 2.4 Meter lang ist. Das Bett soll 80% der Länge des Sensor-Tiefenbildes ausfüllen. Dies ergibt eine Bildlänge l_{Bild} von 3 Meter. Der horizontale Blickwinkel des Sensors beträgt gemäss Tabelle 3.1 $\alpha = 57^\circ$.

Quantisierungsfehler
bestimmen

Der Abstand Δz des Sensors zum Bett ist

$$\Delta z = \frac{\frac{l_{Bild}}{2}}{\tan(\frac{\alpha}{2})} = 2.763 \text{ m} \quad (5.3)$$

In der Formel 3.1 kann nun diese Tiefe eingesetzt werden. Es ergibt sich ein Quantisierungsfehler von 20.49 Millimeter. Dieser Wert wird aufgerundet auf

$$quant_err = 21 \text{ mm}$$

Er entspricht dem maximalen zu erwartenden Quantisierungsfehler für diese Anwendung. Wird der Sensor jedoch höher als 2.763 Meter montiert, muss auch der Wert von *quant_err* nach oben korrigiert werden. Dies sollte jedoch vermieden werden, da ein hoher Wert auch die Bewegungserkennung beeinträchtigt. Der Quantisierungsfehler kann in der Software über die Konstante `QUANT_ERR` eingestellt werden.

5.3.2 Kantenrauschen filtern

Störender Effekt	Es muss nun ein weiterer störender Effekt berücksichtigt werden. Das Tiefenbild einer Person auf einem Bett enthält scharfe Kanten, wie zum Beispiel die Bettkante. Auch wenn die Tiefen sich nicht ändern liefert der Sensor an den Kanten ständig neue Werte. Die ändernden Werte erscheinen natürlich im Differenzbild. Je höher die Kanten, desto höher sind die Werte. Dieses unerwünschte Phänomen heisst im Folgenden Kantenrauschen.
Idee des Filters	Es sind typischerweise immer nur einzelne Pixel oder kleine Bereiche von Pixeln, die ihren Wert ändern. Wenn es nun gelingt, kleine Bereiche im Differenzbild zu erkennen, können diese entfernt werden. Dadurch wird das Kantenrauschen unterdrückt.
Morphologische Operation	Für die Umsetzung wird eine morphologische Operation, das Opening, verwendet. Dabei handelt es sich um das nacheinander Ausführen einer Erosion und einer Dilatation, jeweils mit dem gleichen strukturierenden Element. Durch die Erosion werden alle Strukturen gelöscht, die kleiner sind als das strukturierende Element. Die anschließende Dilatation macht die Erosion für den verbleibenden Rest wieder rückgängig. [5]

Beispiel:

Als Strukturelement `SE` wird ein Kreis mit Radius zwei Pixel verwendet. Abb. 5.3(a) zeigt beispielhaft ein Differenzbild als Bildmatrix `A`. Zellen, die eine grosse Differenz aufweisen sind grau hinterlegt. Abb. 5.3(b) zeigt die Bildmatrix `B` nach Anwendung des Opening Operators. Kleinere Bereiche werden entfernt, während grosse Bereiche bestehen blieben.

Der Opening Operator ist Teil der Image Processing Toolbox von `MATLAB`. Für das Beispiel werden folgende Befehle verwendet:

```
SE = strel('disk',2);
B = imopen(A,SE);
```

Da im Differenzbild einer liegenden Person typischerweise keine Ecken vorkommen, wird als Strukturelement ein Kreis verwendet. Sein Radius ist von der Anzahl Pixel

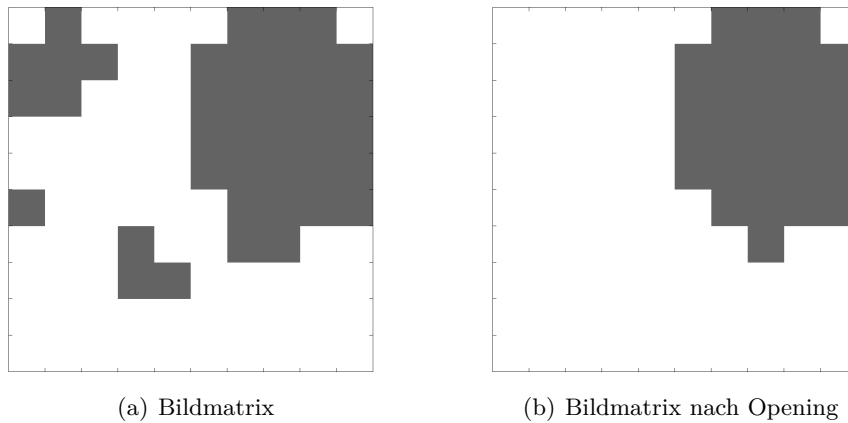


Abbildung 5.3: Opening Operator

im Differenzbild abhängig. Für eine Bildmatrix von 640 x 480 Pixel ist ein Radius von drei Pixel optimal. Wird die Grösse der Bildmatrix geändert, muss auch der Radius geändert werden. Er kann über die Konstante `RADIUS_SE` in der Software eingestellt werden.

Die tatsächlichen Höhendifferenzen bei jedem Pixel des Differenzbildes sind nur begrenzt aussagekräftig. Im Folgenden wird auf die Eigenschaften des Differenzbildes eingegangen.

Der Raum, welcher vom Sensor gemessen wird, kann man sich als dreidimensionales kartesisches Koordinatensystem vorstellen. Die Bewegung einer Person bezüglich des Sensors hat immer drei Komponenten; eine Verschiebung entlang der x-, y- und z-Achse. Im Differenzbild kommen diese Komponenten jedoch unterschiedlich zum Ausdruck.

Bei Bewegungen entlang der z-Achse, also senkrecht zum Sensor, ist es offensichtlich, dass die Höhe des Körperteils, welches in Bewegung ist, zu- oder abnimmt. Somit stellt sich eine Höhendifferenz ein. Bewegt sich eine Person hingegen entlang der x- oder y-Achse, also parallel zum Sensor, ändern die Höhen der Pixel an den Orten wo sich die Person zuvor befunden hat und wo sie sich neu befindet. Nur an den Rändern der Person wird sich eine Höhendifferenz einstellen.

Für Bewegungen parallel zum Sensor kommt ein weiterer Effekt dazu: Bewegungen bei unterschiedlichen Tiefen ergeben unterschiedliche Differenzen. Dieser Effekt ist unerwünscht. Jede gleichartige Bewegung soll dieselbe Auswirkung haben, unabhängig in welcher Tiefe sie stattfindet. Aufgrund dieser Überlegungen macht es wenig Sinn, die tatsächlichen Höhendifferenzen zu verarbeiten. Stattdessen soll die Anzahl aller Pixel, die eine Höhendifferenz aufweisen, bestimmt werden.

Man erhält so die Angabe über die Fläche der Bereiche, die geändert haben. Diese Zahl ist das gewünschte Mass für die Bewegungsaktivität, die Motion Intensity (MI). Formel 5.4 veranschaulicht dies. h_{ij} gibt an, ob an der Stelle ij des Differenzbildes eine Höhendifferenz besteht ($=1$) oder nicht ($=0$).

Parallele /
senkrechte
Bewegungen

Motion Intensity

$$MI = \sum_{i=1}^N \sum_{j=1}^M h_{ij} \quad h_{ij} \in [0, 1] \quad (5.4)$$

mit N Bildzeilen und M Bildspalten

Eigenschaften

Die MI hat folgende positive Eigenschaften:

- Je grösser die Bereiche sind, in denen sich eine Person bewegt, desto grösser ist sie.
- Sie ist unabhängig von der Höhe, in der eine horizontale Bewegung stattfindet.
- Je schneller eine Bewegung parallel zum Sensor ist, desto grösser ist sie.

Folgende negative Eigenschaft ergibt sich:

- Bewegungen unterschiedlicher Geschwindigkeit senkrecht zum Sensor ergeben gleiche Werte, da die Anzahl sich ändernder Pixel gleich bleibt.

Die Berechnung erfolgt in der Funktion `getMotionIntensity()`. Zusätzlich wird die MI mit der Framegrösse skaliert, sodass sie unabhängig von der Bildauflösung ist.

5.4 Motion Intensity auswerten, Bewegungen detektieren

Beginn / Ende bestimmen

Die MI liefert ein Mass für die Bewegungsaktivität der liegenden Person. Mit der Bestimmung der Liegeposition vor und nach der Bewegung kann auf die Bewegungsrelevanz geschlossen werden. Darum muss der Zeitpunkt des Beginns und des Endes einer Bewegung bestimmt werden. Eine einfache Lösung würde darin bestehen, eine Schranke festzulegen. Überschreitet die MI diese, hat die Bewegung begonnen, wird sie unterschritten, hat die Bewegung geendet. Diese Lösung greift jedoch zu kurz. Die Analyse der Videodaten ergab, dass die MI während einer Bewegung auf sehr tiefe Werte fallen kann. Gründe dafür sind das ungewollte Filtern von langsamen Phasen während der Bewegung und die Filterung der Bewegung flacher Objekte. Zudem muss eine Positionsveränderung nicht immer direkt erfolgen. Es kann auch sein, dass während einer Bewegung kurze Pausen eingelegt werden. Die Bewegung sollte aber nicht in Teilbewegungen unterteilt werden. Die gesamte Bewegung ist von Interesse. Daher muss ein robusterer Ansatz gewählt werden.

Top/Bottom-Folger

Aus der MI werden zwei neue Signale gewonnen, ein Top- und ein Bottom-Folger. Der Top-Folger folgt dem Maximum der MI. Kleine Ausreisser nach unten haben keinen Einfluss. Der Bottom-Folger folgt dem Minimum. Eine steigende MI hat nur verzögernd einen Einfluss. Die Folger werden mit einem rekursiven Filter realisiert:

$$y[n] = y[n-1] + \alpha \cdot (r[n] - y[n-1]) \quad (5.5)$$

$r \hat{=}$ Eingangswerte; $y \hat{=}$ Ausgangswerte

Mit dem Parameter α kann der Einfluss der Änderung der Eingangsgrösse auf die Ausgangsgrösse gesteuert werden.

Für die Realisierung des Top-Folgers müssen steigende Eingangswerte stark gewichtet werden. Fallende Eingangswerte dürfen hingegen nicht ins Gewicht fallen.

Der Bottom-Folger muss die umgekehrten Eigenschaften aufweisen.

Aufgrund dieser Tatsache muss das rekursive Filter nicht linear aufgebaut werden. Je nach Vorzeichen der Änderung wird α unterschiedlich gesetzt.

Folgende Werte werden verwendet:

Top-Folger:

$$\begin{aligned} r[n] - y[n-1] > 0 &\Rightarrow \alpha = 0.4 \\ r[n] - y[n-1] < 0 &\Rightarrow \alpha = 0.15 \end{aligned}$$

Bottom-Folger:

$$\begin{aligned} r[n] - y[n-1] > 0 &\Rightarrow \alpha = 0.005 \\ r[n] - y[n-1] < 0 &\Rightarrow \alpha = 0.305 \end{aligned}$$

Als Indikator kann nun die Differenz Δy der Werte des Top- und Bottom-Folgers verwendet werden. Übersteigt diese Differenz einen gewissen Wert, beginnt die Bewegung. Unterschreitet sie einen gewissen Wert, ist die Bewegung beendet. Um diesen Schaltvorgang robuster zu gestalten wird eine Hysterese eingebaut. Nach Auswertung der MI mit dem vorhandenen Videomaterial wurden folgende Werte gewählt:

Schalthysterese

$$\begin{aligned} \Delta y > 10 &\Rightarrow \text{Bewegung beginnt} \\ \Delta y < 5 &\Rightarrow \text{Bewegung endet} \end{aligned}$$

Sie können über die Konstanten `MOTION_START_UPPER_THRESHOLD` und `MOTION_START_LOWER_THRESHOLD` eingestellt werden. Abb. 5.4 zeigt einen möglichen Bewegungsablauf mit Bewegungsdetektion.

Bei starken Bewegungen nimmt die MI hohe Werte an. Wird nun eine solche starke Bewegung beendet, braucht der Top-Folger im Vergleich zu schwachen Bewegungen länger, bis er auf tiefe Werte sinkt. Dies hat zur Folge, dass die Differenz Δy später die untere Schaltschwelle unterschreitet. Aufgrund dessen werden starke Bewegungen gegenüber schwachen später als beendet erkannt. Dieser Unterschied ist aber relativ klein. Es macht sogar Sinn, bei starken Bewegungen etwas länger zu warten, um sicher zu gehen, dass sich die Person wieder in Ruhe befindet.

Eigenschaften der Detektion

Des Weiteren wird, bis zur Detektierung des Bewegungsanfanges, ein ganz kurzer Teil der Bewegung verpasst. In dieser Zeit hat sich die Person aber noch kaum bewegt.

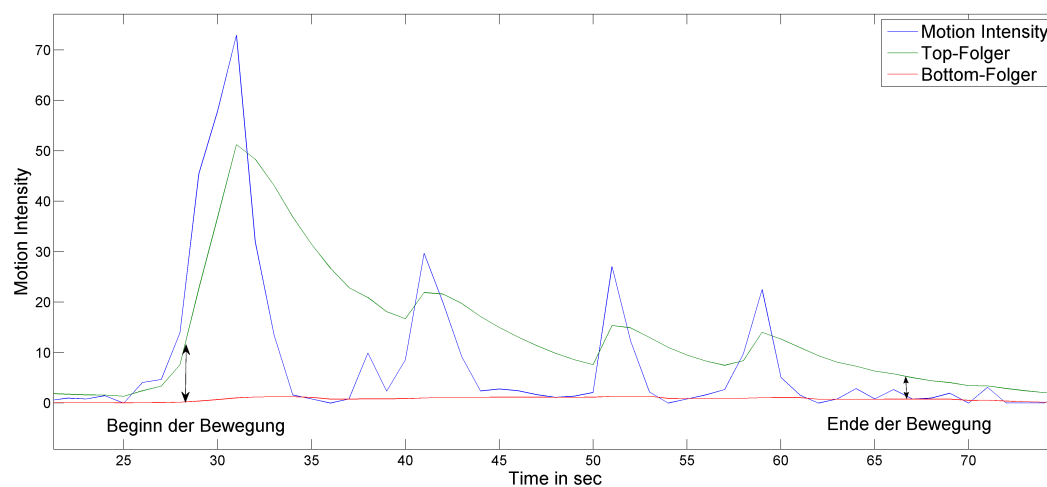


Abbildung 5.4: Bewegungsdetektion mit Top- und Bottom-Folger

5.5 Silhouette ermitteln

Betrachtung von
der Seite

Die Silhouette erhält man, indem eine Person von der Seite beobachtet und stets seiner maximalen Höhe gefolgt wird. Es entsteht ein eindimensionales Höhenprofil. Dabei kann die Silhouette entweder über die Länge oder über die Breite einer Person aufgezeichnet werden. Für die Positionsbestimmung wird die Silhouette allerdings nur über die Länge der Person benötigt.

Finden der
höchsten Punkte

Die Silhouette wird bei jedem Frame neu ermittelt. Als Grundlage dazu dient das geglättete Tiefenbild des Sensors. Das Ziel ist nun, die höchsten Punkte der liegenden Person zu finden. Diese haben den kleinsten Abstand zum Sensor, womit die kleinsten Tiefenwerte gefunden werden müssen. Wie oben erwähnt, wird die Silhouette über die Länge der Person ermittelt. In der Bildmatrix bedeutet dies, über jede Bildspalte das Minimum zu suchen. Diese Vorgehensweise verlangt, dass die liegende Person genau parallel zur x-Achse des Sensorbildes liegen muss. Sobald sie schräg, also mit einem gewissen Winkel zur x-Achse liegt, verfälscht sich die Silhouette. Siehe dazu Abschnitt 5.5.1.

Abb. 5.5 zeigt eine Silhouette bei Rückenlage mit Decke. Dies ist vor allem anhand des Kopfes sichtbar. Das Kinn und die Nase sind hier klar zu erkennen.

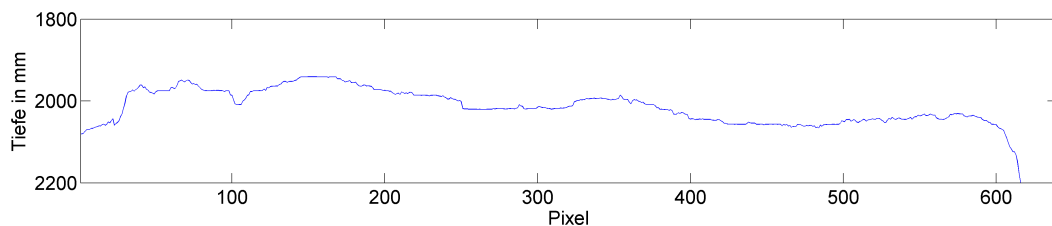


Abbildung 5.5: Silhouette bei Rückenlage mit Decke

Glättung der
Silhouette

Bei genauerem Hinsehen ist erkennbar, dass die Tiefenwerte der Silhouette kleine Sprünge enthalten. Der Grund ist darin zu finden, dass die benachbarten Maximalstellen in der Silhouette an völlig anderen Positionen in y-Richtung liegen können. Dadurch entstehen Sprünge, welche für die spätere Auswertung unerwünscht sind. Sie können mit einem linearen rekursiven Filter entfernt werden.

$$y[n] = y[n-1] + \alpha \cdot (r[n] - y[n-1]) \quad n \in [1, N] \quad (5.6)$$

$$r \hat{=} \text{Eingangswerte}; y \hat{=} \text{Ausgangswerte}$$

$$\text{mit } \alpha = 0.2$$

Abb. 5.6 zeigt die mit dem Filter bearbeitete Silhouette von Abb. 5.5. Die glättende Wirkung des Filters ist gut erkennbar.

5.5.1 Verfälschungen der Silhouette korrigieren

Verfälschung
durch Kissen

Es ist möglich, dass die menschliche Silhouette durch das Kissen verfälscht wird. Das Problem liegt darin, dass die Kissentile links und rechts des Kopfes denselben überragen können. Auf der Silhouette ist dann nicht mehr der Kopf, sondern das Kissen sichtbar.

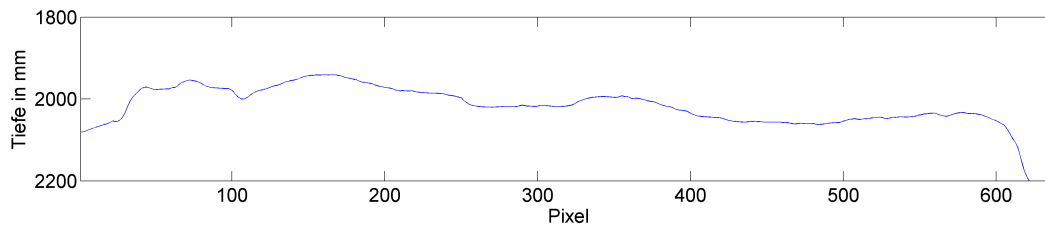


Abbildung 5.6: Geglättete Silhouette bei Rückenlage mit Decke

Daher wurde ein Ansatz geprüft, welcher grosse Sprünge der Silhouette in y-Richtung nicht zulässt. Dies wird erreicht, indem die höchsten Punkte nicht über die ganze y-Achse, sondern nur über einen eingeschränkten Bereich gesucht werden. So kann der höchste Punkt nicht von der Halsgegend auf ein Kissen 'springen'. Die höchsten Punkte würden stattdessen im Bereich des Kopfes gefunden.

Es ist jedoch möglich, dass der höchste Punkt von der Schulter auf das Kissen 'wandert'. In diesem Fall wird er mit grosser Wahrscheinlichkeit nicht mehr zurückfinden, da sein Suchbereich eingeschränkt ist. Aufgrund dieser Tatsache muss der Ansatz verworfen werden. Abb. 5.7 zeigt das Problem auf.

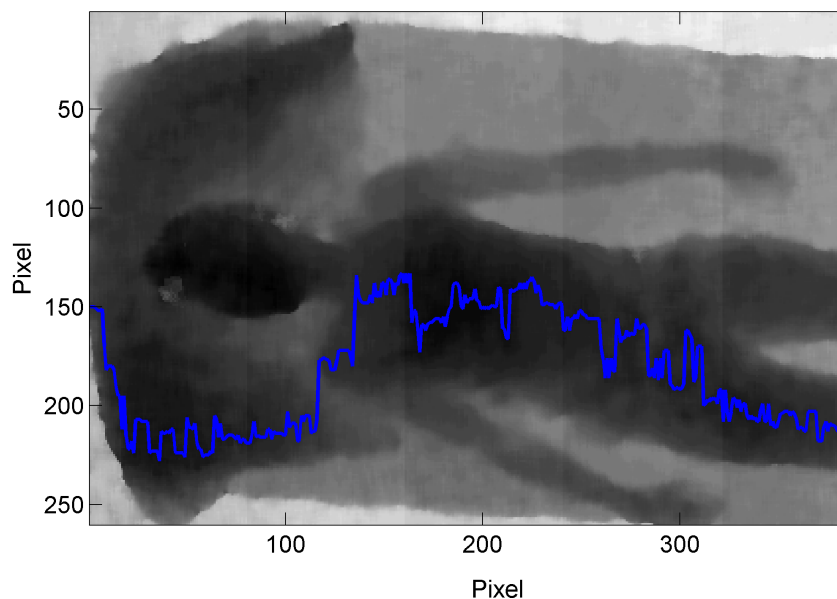


Abbildung 5.7: Falsche Silhouette über Kissen

Ein Lösungsansatz ist, über das ganze Tiefenbild eine optimale Silhouette zu finden. Es handelt sich hierbei um den Dijkstra-Algorithmus. Man bestimmt einen Anfangs- und Endpunkt. Hier wären diese links und rechts des Bildes. Der Algorithmus sucht nun den besten Weg von links nach rechts, indem er aus jeder Bildspalte ein Pixel mit dem nächsten verbindet. Dabei werden den einzelnen Verbindungen Gewichte vergeben. Hier wären dies die absolute Tiefe z und der y-Abstand Δy zwischen zwei Punkten. Je kleiner die Tiefe und je geringer der Abstand zum nächsten Pixel, desto besser. Der optimale Weg wird dort sein, wo die Summe der Gewichte über den ganzen Weg minimal ist. Die Höhen auf diesem Weg ergeben dann die Silhouette.

Dijkstra-
Algorithmus

Der Dijkstra-Algorithmus löst also ein Minimierungsproblem.

$$\min\left\{\sum_{i=1}^M(z_i + \alpha \cdot \Delta y_i)\right\} \quad \text{mit } M \text{ Bildspalten} \quad (5.7)$$

Mit dem Parameter α können die Gewichte optimal aufeinander angepasst werden.

Dieser Algorithmus konnte aus Zeitgründen nicht mehr umgesetzt werden. Er wäre aber ein guter Ansatz, um die Silhouette weniger fehleranfällig gegenüber dem Kissen zu machen.

Verfälschung
durch Schräglage

Es darf nicht davon ausgegangen werden, dass eine Person parallel zum Sensorbild liegt. Dies kann insbesondere zum Problem werden, wenn in Seitenlage der Oberkörper schräg im Bild liegt, was zur Folge hat, dass der Kopf in der Silhouette von der Schulter verdeckt wird. Dieses Problem kann gelöst werden, indem die dominante Achse der Person gefunden wird. Dazu müssen die Eigenwerte und Eigenvektoren der Kovarianzmatrix, die aus der statistischen Bildinformation einer Person gewonnen wird, bestimmt werden. [6]

Die Matrix X enthält die x- und y-Koordinaten aller Punkte der Person.

$$X = \begin{bmatrix} x_1 & x_2 & x_3 & \dots & x_k \\ y_1 & y_2 & y_3 & \dots & y_k \end{bmatrix} \quad \text{wobei } w_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

Davon wird der Mittelwert aller x- und y-Koordinaten berechnet.

$$m = \overline{w_i} = \frac{1}{k} \sum_{i=1}^k w_i \quad (5.8)$$

Für das Eigenwertproblem wird die Kovarianzmatrix C berechnet.

$$C = \frac{1}{k-1} \sum_{i=1}^k (w_i - m)(w_i - m)^T \quad (5.9)$$

Anschließend werden die Eigenwerte und Eigenvektoren von C berechnet. Der Eigenvektor mit dem grössten Eigenwert zeigt in Richtung der dominanten Achse der Person, also in Liegerichtung. Nun kann die Bildmatrix so gedreht werden, dass die Person waagrecht liegt. Die Silhouette des gedrehten Bildes liefert eine wahrheitsgetreue Aussage. Dieser Ansatz konnte aus Zeitgründen nicht mehr implementiert werden.

5.6 Position bestimmen

Aus dem Tiefenbild soll die Seiten- von einer Flachlage unterschieden werden können. Wird eine Seitenlage detektiert, bestimmt der Algorithmus sogleich ob es sich um die linke oder rechte Seitenlage handelt. Diese beiden Unterscheidungen werden in der Funktion `determinePosition()` gemacht.

5.6.1 Seitenlage und Flachlage unterscheiden

Wie in Abschnitt 4.3 beschrieben, dient die grosse Höhendifferenz zwischen Kopf und Schulter als Erkennungsmerkmal der Seitenlage. Es wurde auch in Betracht gezogen, die Differenz der absoluten Schulterhöhe in Seitenlage zur absoluten Brust-/Schulterhöhe in Flachlage als Merkmal herbei zu ziehen. Dieses Merkmal scheint aber weniger robust zu sein, da korpulente Personen gegenüber schlanken eine kleinere Differenz zwischen Schulter- und Brusthöhe aufweisen.

Der Anstieg zwischen Kopf und Schulter hängt wohl von der Postur der Person ab, ist aber bei jedem Menschen vorhanden. Es scheint weniger sinnvoll, das Minima des Halses mit dem Maxima der Schulter zu vergleichen, weil das Minima beim Hals verschwinden kann, wenn die Person etwas schräg im Bett liegt oder die Decke bis an den Kopf hochgezogen wird.

Um die grosse Höhendifferenz zwischen Kopf und Schulter zu finden, müssen diese beiden Positionen zuerst gefunden werden. Dies ist anhand der Silhouette möglich. Dazu werden zuerst alle Maxima in der Silhouette gesucht, wofür die Funktion `findextrema()` verwendet wird. Diese liefert alle lokalen Extrema der Silhouette, von denen nur die Maxima benötigt werden.

Weil die Silhouette nicht perfekt geglättet ist, würden sehr viele Extrema gefunden werden. Die Funktion gibt deshalb nur jene Extrema zurück, welche vorher und nachher eine Mindestanzahl von Pixeln mit gleichem Steigungsvorzeichen haben. Um zu verhindern, dass ein angezogenes Bein in Rückenlage fälschlicherweise als Schulter erkannt wird, werden die Maxima nur im oberen Drittel der Silhouette gesucht. Es scheint ziemlich unwahrscheinlich, dass eine Person weiter unten im Bild liegt, wenn der Kinect-Sensor auf das Bett ausgerichtet wurde. Zudem liegen kleine Personen eher im oberen Teil des Bettes als am unteren Bettrand.

Geeignetes Erkennungsmerkmal

Kopf / Schulter finden

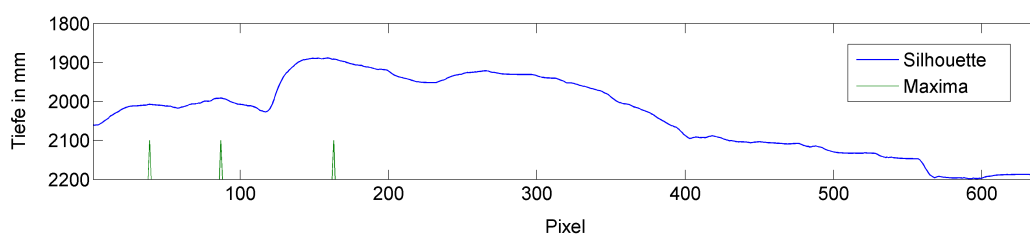


Abbildung 5.8: Silhouette und Maxima

In Abb. 5.8 sind die gefundenen Maxima einer Silhouette visualisiert. Es scheint, dass das letzte Maxima etwas weit hinten erkannt wird. Der Grund liegt darin, dass die Extrema von links nach rechts gesucht werden und ein Maximum erst dort erkannt

wird, wo die Silhouette wieder zu sinken beginnt und nicht schon dort, wo sie flach wird. Dies spielt aber keine Rolle, da für die Auswertung nur die Höhe benötigt wird. Für spätere Untersuchungen reicht diese Genauigkeit der Schulterposition aus.

Position beurteilen

Aus den Maxima kann nun beurteilt werden, ob es sich um eine Seitenlage handelt oder nicht. Dazu werden die Differenzen von einem zum nächsten Maximum gebildet. Wird ein Anstieg gefunden, der grösser ist als `DELTA_HEAD_SHOULDER`, handelt es sich um eine Seitenlage. Andernfalls muss es eine Flachlage sein. Das rechte der beiden Maxima gibt zugleich die Schulterposition `shoulderPos` an. Voraussetzung für eine erfolgreiche Erkennung ist, dass der Kopf in der Silhouette zu sehen ist damit dort ein Maximum gefunden wird.

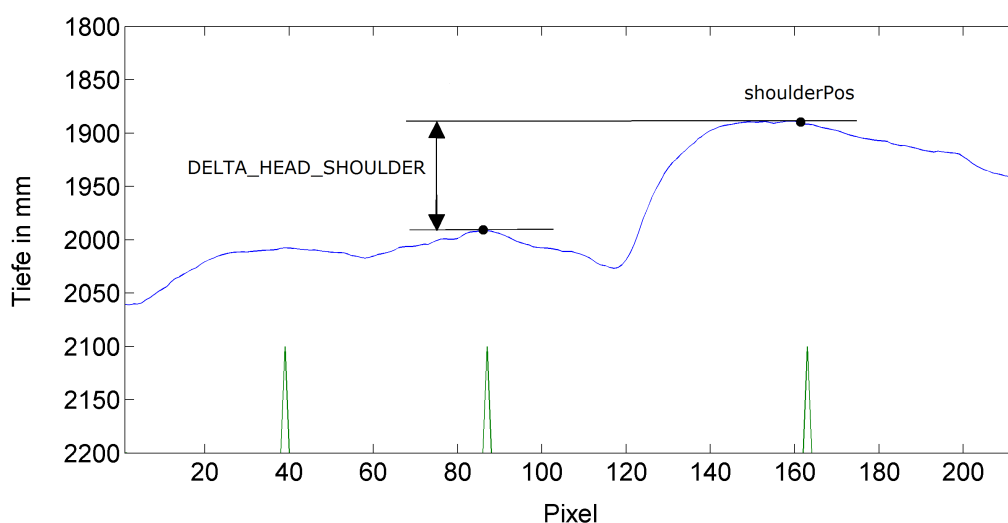


Abbildung 5.9: Seitenlage finden

5.6.2 Rechte und linke Seitenlage unterscheiden

Definition der Seitenlage

Vorerst soll genauer definiert werden, welche Position mit Seitenlage links beziehungsweise Seitenlage rechts gemeint ist. Wir gehen davon aus, dass der Sensor so positioniert ist, dass der Kopf sich am linken Ende des Bildes befindet. In der rechten Seitenlage geht die Blickrichtung der Person nach oben, in der linken nach unten.

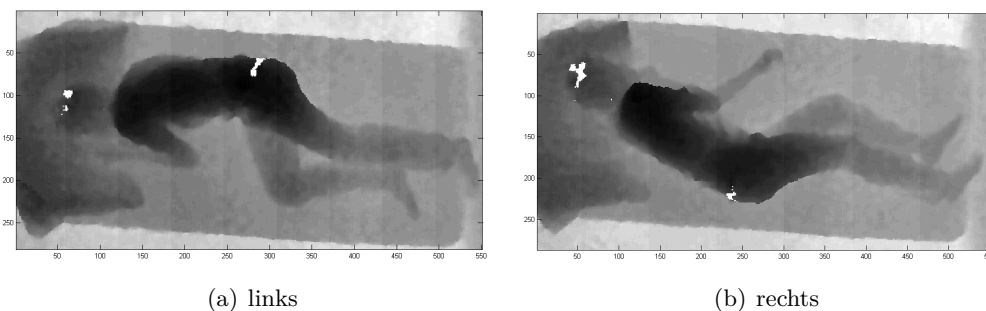


Abbildung 5.10: Seitenlagen

Das Anwinkeln der Beine ist das beste Merkmal, um die beiden Positionen zu unterscheiden. Das heisst der Winkel zwischen dem Oberschenkel und dem Oberkörper ist entscheidend. Die anderen Körperteile sind hier nicht von Bedeutung. Es wäre somit sinnvoll, nur diese beiden Bereiche des Körpers zu analysieren.

Aus der Auswertung der Silhouette ist mit `shoulderPos` die horizontale Position der Schulter bekannt. Von dieser Position aus nach rechts werden alle Punkte im Tiefenbild gesucht, die eine gewisse maximale Höhendifferenz Δh zur Schulter aufweisen. In einer neuen Matrix werden diese Punkte mit einer Eins versehen, die restlichen mit einer Null. Es entsteht ein sogenannter 'Blob'. Mit der Wahl eines geeigneten Δh wird erreicht, dass ein 'Blob' entsteht, der nur den Schulter- und Oberschenkelbereich enthält. Das zu wählende Δh ist die Konstante `BLOB_DEPTH`. Aufgrund der Analyse der Videodaten wurde ein Wert von 150 Millimeter gewählt. Damit sind für Aufnahmen mit und ohne Decke sehr zuverlässige Ergebnisse erzielt worden.

Interessanter
Bereich selektieren

In Abb. 5.11 ist klar erkennbar, dass sich der Winkel zwischen Oberkörper und Oberschenkel nach oben öffnet. Es braucht nun ein geeignetes Werkzeug, um die Richtung dieses Winkels zu erkennen. Dazu wird eine Funktion gesucht, die am besten durch den 'Blob' gelegt werden kann. Hierfür eignet sich ein Polynom zweiter Ordnung, also eine Parabel.

Optimale Parabel
legen

$$y = ax^2 + bx + c \quad (5.10)$$

Wenn sich diese Parabel nach oben öffnet handelt es sich um eine rechte, andernfalls um eine linke Seitenlage. Mit der Methode der kleinsten Fehlerquadrate soll die optimale Parabel gefunden werden. Dies wird in `MATLAB` mit der folgenden Funktion berechnet:

```
p = polyfit(Col,Row,2);
```

`Col` und `Row` sind Vektoren, die alle Indizes der selektierten Bildpunkte enthalten. `p` ist ein Vektor mit den Koeffizienten a , b und c . In Abb. 5.11 ist erkennbar, dass sich der Winkel zwischen Oberkörper und Oberschenkel nach oben öffnet. Es handelt sich demnach um eine rechte Seitenlage. Zusätzlich ist die optimal gelegte Parabel eingezeichnet.

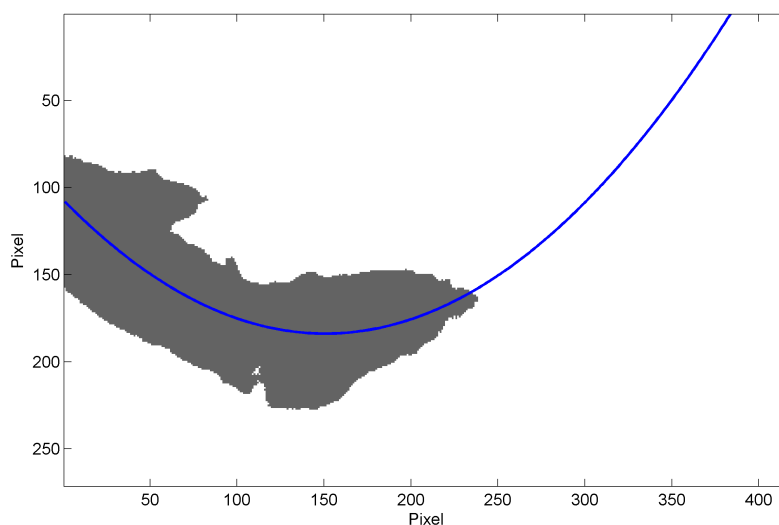


Abbildung 5.11: Schulter- und Oberschenkelbereich mit Parabel

Richtung ablesen Am Koeffizient a kann die Richtung der Parabel abgelesen werden. In der Bildmatrix beginnen die Werte der y-Achse oben. Daher muss auch die Logik für die Richtungsangabe invertiert werden. Ein negatives a ergibt eine nach oben geöffnete Parabel, ein positives a eine nach unten geöffnete Parabel. In Abb. 5.11 hat die Parabel die Form $y = -0.0033x^2 + 1.0181x + 107.1334$. Es handelt sich also um eine rechte Seitenlage. Daraus folgt:

$a < 0 \quad \Rightarrow \quad$ rechte Seitenlage

$a > 0 \quad \Rightarrow \quad$ linke Seitenlage

5.6.3 Bemerkungen zur Flachlage

Mit dem Algorithmus kann wohl eine Flachlage erkannt werden. Eine Aussage darüber, ob sich die Person in Bauch-, Halbseiten- oder Rückenlage befindet, ist jedoch nicht möglich. Um zumindest die Rückenlage von den anderen beiden zu unterscheiden, wurden zwei konkrete Ansätze verfolgt. Leider ist eine zuverlässige Aussage, die sowohl ohne als auch mit Decke funktioniert, bis dato nicht möglich. Im Folgenden werden diese beiden Konzepte kurz erläutert.

Gesicht erkennen

Ungenügende Genauigkeit Das einzige Merkmal, das sich auch mit Decke nicht ändert, ist der Kopf. Optimal wäre daher, den Kopf genauer zu untersuchen, um damit eine Aussage über die Bauch- oder Rückenlage zu machen. Der Kopf befindet sich ungefähr in zwei Meter Distanz vom Sensor. Lieder beträgt der Quantisierungsfehler gemäss Formel 3.1 in dieser Tiefe bereits zehn Millimeter. Bei dieser Ungenauigkeit ist es praktisch unmöglich, die Konturen des Gesichtes zu erkennen. Dazu kommt, dass die Haare wie in Abschnitt 3.1.1 beschrieben, oft zu ungünstigen Werten führen können. Für die Haare können darum keine präzisen Tiefendaten erwartet werden. Vielmehr springen die Werte stark hin und her.

Beispiele Die vier Kopfpositionen in Abb. 5.12 zeigen, dass eine Erkennung schon von Auge ziemlich schwierig ist. Diese Bilder wurden in Ruhelage gemacht und über 40 Frames (vier Sekunden) gemittelt.

Silhouette unzuverlässig Auch die Analyse der Silhouette im Kopfbereich ist nicht zuverlässig. Liegt die Person auf dem Rücken mit dem Gesicht gerade nach oben, ist die Nase gut erkennbar. Sobald sich der Kopf aber leicht zur Seite dreht, verschwindet dieses Merkmal. Die generelle Unebenheit des Gesichtes kann nicht verwendet werden, weil die Haare ebenfalls uneben sind.

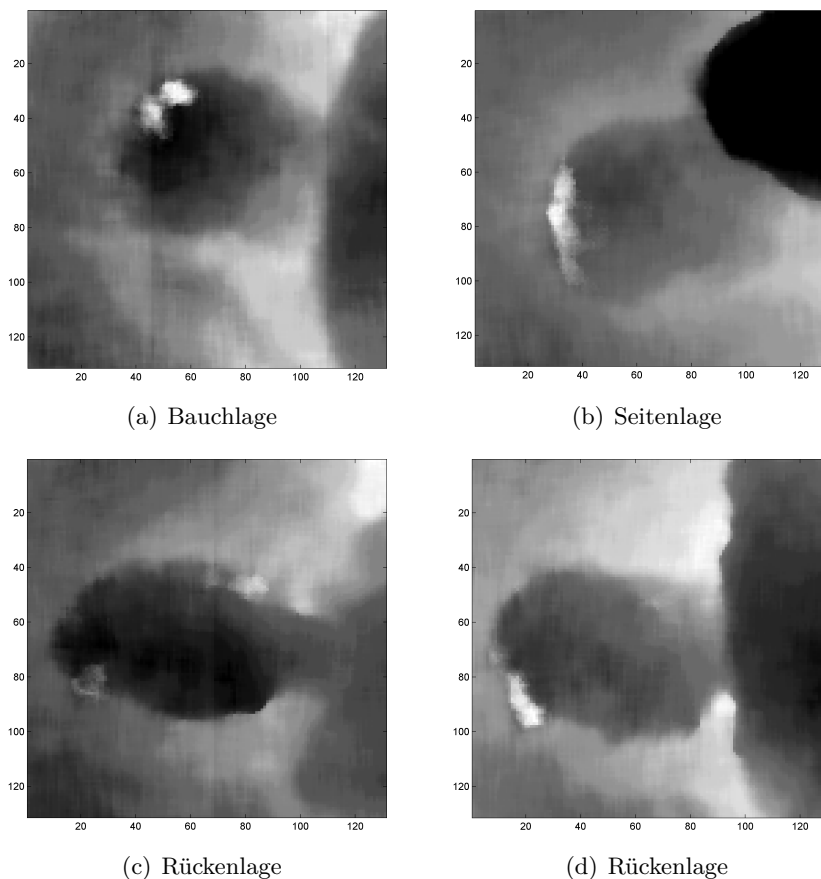


Abbildung 5.12: Kopfpositionen

Füsse / Beine untersuchen

Ein mögliches Merkmal zur Unterscheidung der Rücken- und Bauchlage sind die Füße. In der Rückenlage stehen diese nach oben und ergeben eine markante Überhöhung in der Silhouette. Leider hat die Decke stark glättenden Charakter. Dadurch ist die Silhouette der Rückenlage kaum von der einer Bauchlage zu unterscheiden. Dieses Merkmal würde sich also nur ohne Decke eignen.

Durch die Analyse der Silhouetten wurde festgestellt, dass in Rückenlage der tiefste Punkt der Silhouette nahe beim Fussgelenk liegt. In der Bauchlage liegt er jedoch in der Kniekehle und die Silhouette fällt vom Gesäss her relativ steil ab. Diese Unterschiede sind mit Decke aber nur sehr schwach zu erkennen und für verschiedene Personen wohl nicht gleich aussagekräftig. Deshalb wurden diese Ansätze nicht umgesetzt.

Überhöhung bei
Füssen

Beine als Merkmal

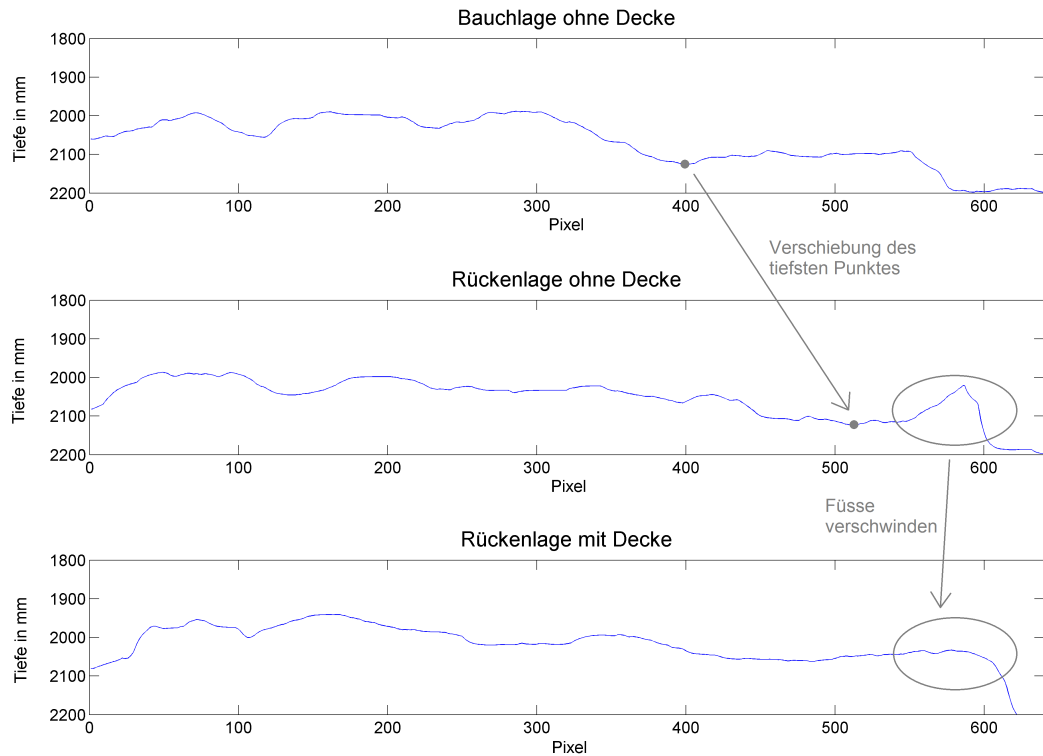


Abbildung 5.13: Vergleich der Füsse und Beine

5.7 Bewegungsrelevanz beurteilen

Berechnung des Fehlerquadrats

Die Silhouette einer liegenden Person kann verwendet werden, um eine generelle Aussage darüber zu machen, ob sich die Person relevant bewegt hat. Dazu werden die Silhouetten vor und nach der Bewegung verglichen. Stimmen sie nicht bis zu einem gewissen Grad überein, kann man von einer relevanten Bewegung ausgehen. Als Mass für die Ähnlichkeit zweier Silhouetten wird das mittlere Fehlerquadrat berechnet. Dazu werden die Höhenunterschiede $|h_{1n} - h_{2n}|$ an jeder Stelle der beiden Silhouetten quadriert. Aus diesen Werten wird der Mittelwert gebildet.

$$sqr_err = \frac{1}{N} \sum_{n=1}^N |h_{1n} - h_{2n}|^2 \quad n \in [1, N] \quad (5.11)$$

Verschiebung beachten

Diese Zahl hat den Vorteil, dass grosse Höhenunterschiede stärker gewichtet werden. Ein negativer Effekt sollte dabei beachtet werden:

Liegt eine Person nach einer Bewegung in derselben Lage, jedoch an einer bezüglich oberer Bettkante anderen Position, wird das Fehlerquadrat gross sein, da die Silhouetten beim Vergleich nicht mehr übereinander liegen. Dies ist unerwünscht, denn die globale Position der Person soll keinen Einfluss auf das mittlere Fehlerquadrat haben. Zur Behebung dieses Umstandes werden die Silhouetten gegeneinander verschoben. Aus den verschiedenen Werten wird das kleinste mittlere Fehlerquadrat ermittelt. Die erhaltene Zahl ist ein wahrheitsgetreues Mass für die Ähnlichkeit zweier Liegepositionen.

Die maximale Verschiebung wird auf 40 Pixel festgelegt. Dieser Wert gilt für eine Framegrösse von 640 x 480 Pixel. Ansonsten muss sie kleiner gewählt werden. Da sich die Silhouetten zueinander verschieben können, ergeben sich $40 \cdot 40 = 1600$ mögliche Kombinationen, die gerechnet werden müssen. Der kleinste erhaltene Wert stellt das kleinste mittlere Fehlerquadrat dar.

Dabei muss zusätzlich darauf geachtet werden, dass nur die Silhouette der Person verwendet wird. Sobald noch Teile des Bettes enthalten sind, können diese die Auswertung stark verfälschen. Auch das Ende der Füße soll in der Silhouette nicht enthalten sein, da dort steile Kanten vorkommen können. Das 'Matching' der Silhouette wird dann an diesen Kanten minimal sein und es ist kein wirklicher Vergleich der ganzen Silhouette gegeben. Aussagekräftig ist aber vor allem die Ähnlichkeit im Schulter- und Hüftbereich.

Daher wird die Silhouette im unteren Drittel des Bildes nach steil abfallenden Stellen überprüft. Werden solche gefunden, ist immer die am nächsten zur oberen Bettkante liegende zu verwenden. An dieser Stelle müssen sich die Füße befinden. Die für das Matching verwendete Silhouette beginnt dann erst 40 Pixel oberhalb dieser Stelle. Sind keine Kanten vorhanden, werden die Silhouetten über die ganze Länge verwendet. In der Kopfgegend wird keine Anpassung vorgenommen. Das Tiefenbild bei der Initialisierung vom Benutzer sauber auf die Bettgrösse anzupassen.

Silhouettenlänge anpassen

Es gilt nun eine geeignete Schranke festzulegen, ab welcher auf eine relevante Bewegung geschlossen werden kann. Sie kann nur auf empirischem Wege gefunden werden. Dazu wurden einige Silhouetten aus dem vorhandenen Videomaterial analysiert. In den Tabellen 5.1 bis 5.4 sind die kleinsten mittleren Fehlerquadrate von relevanten und nicht relevanten Bewegungen aufgelistet.

Analyse der Videodaten

Positionswechsel			min	max
Bauch	⇔	Halbseite	1000	1473
Bauch	⇔	Rücken	728	1019
Rücken	⇔	Halbseite	1745	2364
Seite	⇔	Bauch	3455	8000
Seite	⇔	Rücken	3091	7818
Seite	⇔	Halbseite	2000	4909

Tabelle 5.1: Relevante Bewegungen ohne Decke

Positionswechsel			min	max
Bauch	⇔	Halbseite	727	1818
Bauch	⇔	Rücken	745	1455
Rücken	⇔	Halbseite	1818	2182
Seite	⇔	Bauch	2909	5091
Seite	⇔	Rücken	2182	3818
Seite	⇔	Halbseite	855	3091

Tabelle 5.2: Relevante Bewegungen mit Decke

Positionswechsel			min	max
Bauch	\Leftrightarrow	Bauch	255	327
Rücken	\Leftrightarrow	Rücken	455	455
Halbseite	\Leftrightarrow	Halbseite	142	142
Seite	\Leftrightarrow	Seite	200	1709

Tabelle 5.3: Nicht relevante Bewegungen ohne Decke

Positionswechsel			min	max
Bauch	\Leftrightarrow	Bauch	35	582
Rücken	\Leftrightarrow	Rücken	56	491
Halbseite	\Leftrightarrow	Halbseite	600	600
Seite	\Leftrightarrow	Seite	173	1000

Tabelle 5.4: Nicht relevante Bewegungen mit Decke

Auswertung

Erwartet wurde, dass bei relevanten Bewegungen die gemessenen Werte stets größer sind als bei nicht relevanten Bewegungen. Die Daten zeigen jedoch, dass dies nicht immer erfüllt ist. Die nicht relevante Bewegung Seite \Leftrightarrow Seite hat bei Auswertung ohne Decke einen gemessenen maximalen Wert von 1709. Er ist höher als beispielsweise der minimale Wert der relevanten Bewegung Bauch \Leftrightarrow Rücken. Bei allen anderen Bewegungen zeigen die Daten das erwartete Verhalten.

Folglich können alle Bewegungen, bei denen die Seitenlage Start- oder Endposition ist, nicht auf Bewegungsrelevanz überprüft werden. Dies ist jedoch unproblematisch, da die Seitenlage mit anderen Mitteln, die in Abschnitt 5.6.1 beschrieben sind, erkannt werden kann. Aufgrund der gemessenen Werte kann nun für die restlichen Bewegungen eine geeignete Schranke festgelegt werden.

$$\text{Optimale Schranke} = 670$$

Sie kann über die Konstante `REL_MOTION_THRESHOLD` modifiziert werden.

5.8 Graphische Benutzeroberfläche

Während einer Schlafüberwachung sollen dem Benutzer die wichtigsten Zahlen und Graphiken zur Verfügung gestellt werden. Dazu dient eine graphische Benutzeroberfläche, welche diese Daten in Echtzeit anzeigt. Sie wurde mit MATLAB erstellt.

GUI

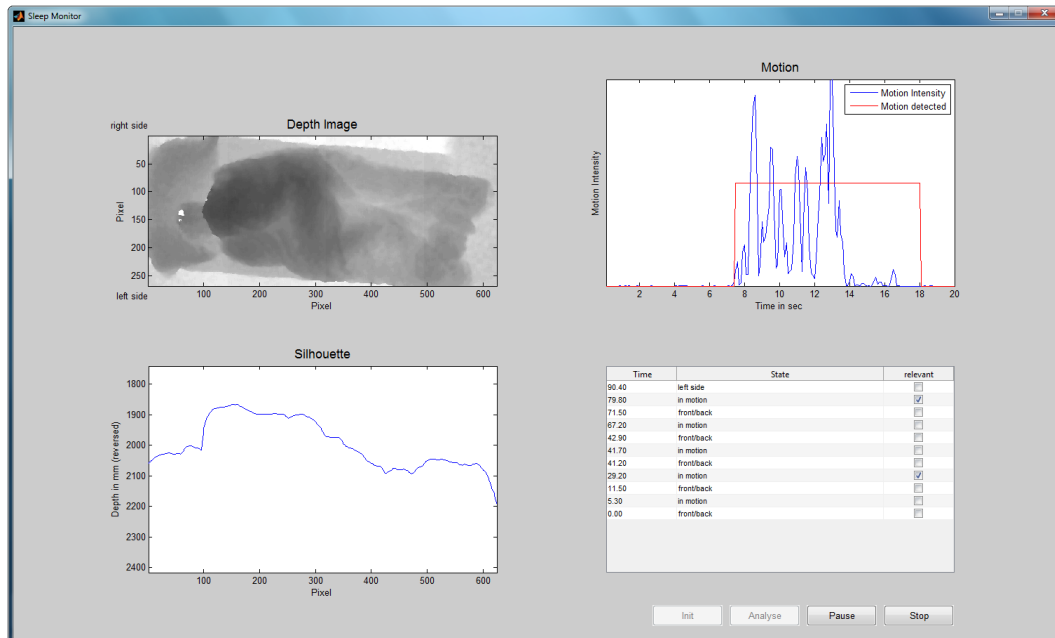


Abbildung 5.14: Benutzeroberfläche während einer Überwachung

Folgende Informationen werden zur Verfügung gestellt:

Dargestellte
Informationen

- Links oben wird das vom Benutzer eingegrenzte Tiefenbild angezeigt. Je größer die Tiefenwerte sind, desto heller wird die Anzeige. Bereiche, die vom Sensor nicht erkannt werden, sind weiss eingefärbt.
- Links unten erscheint die Silhouette. Die Tiefenwerte werden umgekehrt angezeigt, sodass kleine Tiefen höher liegen. Dadurch erscheint eine logische Silhouette der Person, wie man sie von der Seite her betrachtet sehen würde.
- Rechts oben wird die Bewegungsintensität in einem Zeitraum von 20 Sekunden aufgezeichnet. Zudem wird der Start und das Ende einer Bewegung signalisiert.
- Rechts unten werden in einer Tabelle alle Zustände aufgelistet, in der sich die schlafende Person befunden hat. Zuerst steht der aktuelle Zustand. Bei Beginn der Bewegung erscheint 'in motion'. Am Ende der Bewegung wird die neue Position angezeigt. Die Zeitangabe markiert den Zeitpunkt, bei welchem sich der neue Zustand einstellt. War eine Bewegung relevant, wird dies beim Eintrag 'in motion', am Ende der Bewegung, vermerkt.

Durch Drücken des 'Init' Button öffnet sich ein neues Fenster. Das Bild des Sensors kann nun auf die Grösse des Bettes angepasst werden. Mit 'Apply' werden die Einstellungen übernommen. Dieser Initialisierungsvorgang ist vor jeder Schlafüberwachung vorzunehmen. Bei Falscheingabe kann er beliebig oft wiederholt werden.

Initialisierung
durch Benutzer

Programm starten / beenden	Mit den Buttons 'Analyse' und 'Stop' kann die Überwachung gestartet bzw. beendet werden. Nach einem Stopp kann nochmals eine Initialisierung gemacht oder mit 'Analyse' das Video von Neuem analysiert werden.
Programm unterbrechen / fortsetzen	Mit dem Button 'Pause' wird die Überwachung gestoppt. Die Funktion des gleichen Buttons wechselt dann auf 'Resume'. Bei erneutem Drücken wird die Überwachung an gleicher Stelle fortgesetzt. Diese Funktion macht nur deshalb Sinn, weil die Daten aus einer Datei gelesen werden. Bei einer Live-Überwachung würde diese Funktion nicht benötigt.

6. Tests und Auswertung

Dieses Kapitel zeigt auf, wie gut der Algorithmus funktioniert und wo seine Stärken und Schwächen liegen. Es wird beschrieben, mit welchen Parametern die Schlafüberwachung optimiert und mit welchen Ansätzen sie verbessert werden kann.

6.1 Testen des Algorithmus

Der Algorithmus wurde mit allen Aufnahmen aus der in Abschnitt 3.5.3 beschriebenen Datenbank, getestet; sowohl mit als auch ohne Decke. Er wurde nur mit den Einzelbewegungen entwickelt. Der Bewegungsablauf, welcher eine separate Aufnahme ist, zeigt wie gut die Software unbekannte Aufnahmen beurteilen kann.

Testdaten

Bewegungen		✓ / ×	Problem
Einzelbewegungen			
Bauchlage	⇒	Halbseitenlage links	✓
Halbseitenlage links	⇒	Seitenlage links	✓
Seitenlage links	⇒	Halbseitenlage links	✓
Halbseitenlage links	⇒	Bauchlage	× Matching
Bauchlage	⇒	Seitenlage rechts	✓
Seitenlage rechts	⇒	Rückenlage	✓
Rückenlage	⇒	Seitenlage rechts	✓
Seitenlage rechts	⇒	Bauchlage	✓
Bauchlage	⇒	Rückenlage	× Matching
Rückenlage	⇒	Bauchlage	✓
Vorgekommene nicht relevante Bewegungen		✓	
Bewegungsablauf			
Bauchlage	⇒	Halbseitenlage links	× Matching
Halbseitenlage links	⇒	Seitenlage links	✓
Seitenlage links	⇒	Halbseitenlage links	✓
Halbseitenlage links	⇒	Bauchlage	✓
Bauchlage	⇒	Seitenlage rechts	✓
Seitenlage rechts	⇒	Rückenlage	✓
Rückenlage	⇒	Seitenlage rechts	✓
Seitenlage rechts	⇒	Bauchlage	✓
Bauchlage	⇒	Rückenlage	× Matching
Rückenlage	⇒	Bauchlage	✓
Vorgekommene nicht relevante Bewegungen		×	Schulterhöhe, Kissen

Testresultate
ohne Decke

Tabelle 6.1: Resultate ohne Decke

Beschreibung der Fehler

Die Probleme des Matchings sind klar. Wenn der Fehler nicht genügend gross wird, kann keine relevante Bewegung erkannt werden.

In der Halbseitenlage wurde einmal eine nicht relevante Bewegung als relevant erkannt. Grund war, dass zu Beginn der Bewegung fälschlicherweise eine Seitenlage erkannt wurde. In Halbseitenlage liegt die Schulter bereits etwas höher und wegen dem kleinen $\Delta_{\text{HEAD_SHOULDER}} = 50 \text{ mm}$ ist es zur Fehlerkennung gekommen. Wird dieses aber vergrössert, ist es möglich, dass eine Seitenlage gar nicht erkannt wird, wenn die Schulter etwas abgekippt ist.

Einmal wurde eine Seitenlage nicht erkannt, weil der Kopf in der Silhouette vom Kissen verdeckt war.

Testresultate mit Decke

Bewegungen		✓ / ×	Problem
Einzelbewegungen			
Bauchlage	⇒ Halbseitenlage links	✓	
Halbseitenlage links	⇒ Seitenlage links	✓	
Seitenlage links	⇒ Halbseitenlage links	✓	
Halbseitenlage links	⇒ Bauchlage	✓	
Bauchlage	⇒ Seitenlage rechts	✓	
Seitenlage rechts	⇒ Rückenlage	✓	
Rückenlage	⇒ Seitenlage rechts	✓	
Seitenlage rechts	⇒ Bauchlage	✓	
Bauchlage	⇒ Rückenlage	×	Matching
Rückenlage	⇒ Bauchlage	×	Matching
Vorgekommene nicht relevante Bewegungen		×	Matching
Bewegungsablauf			
Bauchlage	⇒ Halbseitenlage links	×	Matching
Halbseitenlage links	⇒ Seitenlage links	×	Kissen
Seitenlage links	⇒ Halbseitenlage links	✓	
Halbseitenlage links	⇒ Bauchlage	×	Matching
Bauchlage	⇒ Seitenlage rechts	✓	
Seitenlage rechts	⇒ Rückenlage	✓	
Rückenlage	⇒ Seitenlage rechts	✓	
Seitenlage rechts	⇒ Bauchlage	✓	
Bauchlage	⇒ Rückenlage	✓	
Rückenlage	⇒ Bauchlage	✓	
Vorgekommene nicht relevante Bewegungen		×	Schulterhöhe, Kissen

Tabelle 6.2: Resultate mit Decke

Beschreibung der Fehler

Wiederum wurden einige relevante Bewegungen nicht erkannt, weil die Silhouetten zu ähnlich waren. Einmal wurde bei einer nicht relevanten Bewegung die Decke etwas gespannt, was die Silhouette erheblich veränderte und daher als relevante Bewegung erkannt wurde. Die Decke senkte sich wohl wieder ab, aber so langsam, dass der Motion Detector längst wieder eine Ruhelage erkannte, bevor sich die ursprüngliche Silhouette eingestellt hatte.

In Bauchlage wurde einmal die Seitenlage erkannt, weil der Arm ganz an den Körper gezogen wurde und die Person dadurch etwas schräg lag. Somit kam die Schulter weiter nach oben und das $\Delta_{\text{HEAD_SHOULDER}} = 100 \text{ mm}$ wurde überschritten.

Das Kissen war wiederum ein Problem. Wenn es den Kopf verdeckt, kann die Seitenlage nicht erkannt werden. Dies war im Bewegungsablauf in der linken Seitenlage der Fall.

6.2 Kritische Parameter

Für die Funktionalität des entwickelten Algorithmus sind einige Parameter entscheidend. Hier sollen die Parameter beschrieben werden, die besonders kritisch sind und bei verschiedenen Aufnahmen unterschiedlich eingestellt werden müssen.

Kritische
Parameter

- **DELTA_HEAD_SHOULDER**
Die Höhendifferenz zwischen Kopf und Schulter ist bei Aufnahmen mit und ohne Decke unterschiedlich zu setzen, da die Decke eine zusätzliche Höhe darstellt. Bei dem vorhandenen Videomaterial wurde mit Decke ein Wert von `DELTA_HEAD_SHOULDER = 100 mm` verwendet. Ohne Decke musste der Wert hingegen auf `DELTA_HEAD_SHOULDER = 50 mm` eingestellt werden.
- **BLOB_DEPTH**
Die Tiefe, welche für die Selektierung des Schulter- und Oberschenkelbereichs nötig ist, kann bei den erstellten Aufnahmen mit und ohne Decke gleich gewählt werden. Ein Wert von `BLOB_DEPTH = 150 mm` war hier optimal.
- **REL_MOTION_THRESHOLD**
Die Schwelle, ab der das kleinste mittlere Fehlerquadrat als relevante Bewegung erkannt wird, ist in besonderem Masse von den konkreten Tiefendaten abhängig. Für die erstellten Aufnahmen wurde ein Wert von `REL_MOTION_THRESHOLD = 670` gewählt.

Wie stark diese Parameter von der aufgenommenen Person abhängen, ist nicht bekannt. Es muss jedoch davon ausgegangen werden, dass insbesondere `REL_MOTION_THRESHOLD` und `DELTA_HEAD_SHOULDER` bei anderen Personen angepasst werden müssen.

Unbekannte
Abhängigkeit

6.3 Beurteilung

In diesem Abschnitt sind die Stärken und Schwächen der entwickelten Schlafüberwachung beschrieben. Zudem werden Ansätze zur Verbesserung des Algorithmus, in Hinblick auf eine mögliche Weiterentwicklung der erarbeiteten Lösung, vorgeschlagen.

6.3.1 Stärken

Die Detektion einer Bewegung anhand der MI ist zuverlässig und robust. Kleine wie auch grosse Bewegungen werden erkannt. Der Start einer Bewegung wird rechtzeitig signalisiert und die Bewegung wird erst dann für beendet befunden, wenn die Person in Ruhe ist. Auch die Angabe über die Intensität der Bewegung ist wahrheitsgetreu.

Bewegungsdetektion

Bei schwachen Bewegungen ergeben sich nur minime Ausschläge der MI. Grosse Bewegungen fallen indes stark aus.

Rechte/Linke
Seitenlage
unterscheiden

Die Unterscheidung der rechten und linken Seitenlage funktioniert sehr gut. Der Ansatz, mit einer optimal gelegten Parabel den Winkel des Schulter- und Oberschenkelbereichs zu erkennen, ist robust und kaum fehleranfällig. Es kann jedoch nicht mit letzter Sicherheit angenommen werden, dass alle Menschen beim Schlaf in der Seitenlage diesen Winkel aufweisen. Für diese Aussage müssten mehr Daten zur Verfügung stehen. Zudem muss BLOB_DEPTH richtig gewählt sein. Erscheint nur der Oberkörper oder noch Teile der Matratze im 'Blob', ist keine korrekte Erkennung mehr garantiert.

6.3.2 Schwächen

Bestimmung der
Silhouette

Die Silhouette kann aufgrund des Kissens verfälscht werden. Dies hat leider einen Einfluss auf die Unterscheidung der Seiten- von einer Flachlage. Vermag das Kissen den Kopf zu verdecken, wird derselbe nicht mehr erkannt. Es ist dann nicht möglich einen Höhenunterschied zwischen Kopf und Schulter zu messen. Zudem wird das Matching der Silhouetten, also die Aussage über die Relevanz einer Bewegung, beeinträchtigt. Der in Abschnitt 5.5.1 beschriebene Dijkstra-Algorithmus wäre ein möglicher Weg, um die Fehleranfälligkeit der Silhouette zu verkleinern.

Weiter ist die Silhouette verfälscht, sobald die liegende Person nicht parallel zur Bettkante liegt. Der Kopf könnte dann hinter den Schultern 'verschwinden'. Um eine wahrheitsgetreue Silhouette zu bestimmen, müsste die Achse der Person gedreht werden. Dies könnte mit einer Eigenwertzerlegung gemäss Abschnitt 5.5.1 beschrieben, realisiert werden.

Matching der
Silhouetten

Der Ansatz, die Bewegungsrelevanz mit dem Matching zweier Silhouetten zu bewerten, ist schwierig zu beurteilen. Eine genaue Analyse verschiedener Silhouetten wäre nötig. Zudem bedingt das Matching, dass nur die Silhouetten des Körpers analysiert werden. Dafür muss die ungefähre Position des Kopfes sowie der Füsse bekannt sein. Diese Erkennung ist jedoch noch nicht robust genug und muss verbessert werden. Weiter werden spezielle Bewegungen, wie das Anziehen der Beine, als relevant erkannt. Dies entspricht aber nicht einer relevanten Bewegung, wie sie in Abschnitt 3.5 beschrieben ist.

Seiten- und
Flachlage
unterscheiden

Die Unterscheidung der Seiten- von einer Flachlage ist vom Höhenunterschied zwischen Kopf und Schulter abhängig. Neben der oben beschriebenen Fehleranfälligkeit der Silhouette können noch andere Faktoren die Aussage verfälschen. Hält eine Person die Schultern in der Seitenlage sehr tief, kann eine Flachlage erkannt werden. Hat eine Person hingegen in Rückenlage die Arme auf die Brust gelegt, kann eine Seitenlage erkannt werden. Der Ansatz ist grundsätzlich gut, weil das Merkmal des Kopf-Schulter Anstiegs eigentlich bei jeder Person sichtbar sein sollte. Die Implementierung ist aber noch zu einfach gehalten, wodurch zu oft falsche Aussagen gemacht werden.

7. Schlussfolgerungen und Ausblick

Das Ziel dieser Arbeit war, anhand einer selbst erstellten Datenbank, einen Algorithmus zur Überwachung einer schlafenden Person zu entwickeln. Abschliessend kann festgestellt werden, dass dieses Ziel zumindest teilweise erreicht wurde. Es ist grundsätzlich möglich, mit dem Kinect-Sensor eine Person zu überwachen und ihre Bewegungen zu beurteilen, selbst wenn diese unter einer Decke liegt. Jedoch sind die einzelnen Funktionalitäten des Algorithmus unterschiedlich robust.

Ziel erreicht?

Die Bewegungsdetektion und die Erkennung der linken und rechten Seitenlage funktionieren sehr zuverlässig. Es ergaben sich für die Aufnahmen aus der Datenbank keine Fehlerkennungen. Die Unterscheidung der Seiten- von einer Flachlage, sowie die generelle Aussage über die Bewegungsrelevanz sind jedoch noch fehleranfällig. Leider wurde kein Merkmal gefunden, um die Rückenlage von der Bauch- oder Halbseitenlage zu unterscheiden. Ohne Decke wäre es gut machbar, mit Decke wird es aus der Ruhelage jedoch kaum möglich sein. Bauch- und Halbseitenlage können zurzeit nicht auseinander gehalten werden, daher wird eine links/rechts Erkennung der Halbseitenlage hinfällig.

Stärken /
Schwächen

Mit einer Verbesserung und Überarbeitung der bestehenden Lösung ist eine robustere Schlafüberwachung möglich. Insbesondere die Ermittlung der Silhouette muss verbessert werden. Einerseits um den Einfluss des Kissens zu eliminieren und andererseits um von der Schräglage der Person bezüglich dem Tiefenbild unabhängig zu werden. Lösungsansätze dazu sind in Abschnitt 5.5.1 beschrieben.

Verbesserungs-
vorschläge

Der Ansatz zur Unterscheidung der Seiten- von einer Flachlage muss noch verbessert werden. Hier ist ein zusätzliches Merkmal nötig, um den Algorithmus robuster zu machen.

Um die Rückenlage von der Bauch- und Halbseitenlage zu unterscheiden könnte ein dynamischer Ansatz (Tracking) weiterhelfen. Mit der Richtung der Seitenlage und der Drehrichtung der Bewegung könnte eine Aussage gemacht werden. Das Erkennen der Drehrichtung ist allerdings kein einfaches Problem.

Auch das Matching der Silhouetten muss noch überarbeitet werden. So ist der Bereich, in dem die Silhouetten verglichen werden, gründlicher auszuwählen, um bessere Resultate zu erhalten.

Der Algorithmus könnte durch die vorgeschlagenen Weiterentwicklungen stark an Zuverlässigkeit gewinnen. Erst dann wäre ein Einsatz in der Pflege von betagten Personen denkbar. Dafür müsste auch ein Test mit diversen Probanden unterschiedlichster Posturen durchgeführt werden. Vorher ist eine gute Aussage über die Robustheit des Algorithmus kaum möglich.

Ausblick

Eine Schlafüberwachung mit einer Tiefenbildkamera scheint aber grundsätzlich umsetzbar zu sein und könnte in Zukunft durchaus zum Einsatz kommen.

8. Schlusswort

Erstellen der Arbeit

Die Erstellung und Analyse der Aufnahmen, sowie die Entwicklung eines geeigneten Algorithmus war spannend und hat uns viel Freude bereitet. Zu Beginn konnten wir uns nicht vorstellen, wie eine Lösung für diese Aufgabenstellung aussehen könnte. Es war interessant, Schritt für Schritt eine Lösung zu erarbeiten, auch wenn dabei immer wieder schwierige Problemstellungen auftauchten und erarbeitete Lösungen verworfen werden mussten.

Dank

Es bleibt uns noch einen Dank all denen auszusprechen, die uns in dieser Arbeit unterstützt haben. Prof. Dr. Guido M. Schuster hat die Arbeit betreut. In den wöchentlichen Meetings konnten immer wieder Probleme diskutiert und Lösungen gefunden werden. Peter Roffler hat sich für die Ausstattung der Arbeitsplätze und die Beschaffung der benötigten Hardware gekümmert. Mathias Hunold war stets unsere erste Ansprechperson, wenn Probleme mit `MATLAB` auftauchten. Wir waren einige Male dankbar für seine Hilfe.

9. Abbildungsverzeichnis

3.1	Kinect-Sensor [2]	5
3.2	Berechnete Distanz [3]	5
3.3	Tiefen- und Videobild	6
3.4	Tiefenauflösung	7
3.5	Tiefensprünge	7
3.6	Vertikale Linien mit Pfeilen markiert	8
3.7	NUI Library [3]	8
3.8	Format der Tiefenwerte	8
3.9	Kinect Recorder	9
3.10	Liegepositionen	10
4.1	Differenzbild während einer Bewegung	14
4.2	Silhouette in Seitenlage	15
5.1	Programmablauf	17
5.2	Glättung eines Pixels über die Zeit	18
5.3	Opening Operator	21
5.4	Bewegungsdetektion mit Top- und Bottom-Folger	23
5.5	Silhouette bei Rückenlage mit Decke	24
5.6	Geglättete Silhouette bei Rückenlage mit Decke	25
5.7	Falsche Silhouette über Kissen	25
5.8	Silhouette und Maxima	27
5.9	Seitenlage finden	28
5.10	Seitenlagen	28
5.11	Schulter- und Oberschenkelbereich mit Parabel	29
5.12	Kopfpositionen	31
5.13	Vergleich der Füße und Beine	32
5.14	Benutzeroberfläche während einer Überwachung	35

10. Tabellenverzeichnis

3.1	Technische Daten [3]	6
3.2	Bewegungen	11
3.3	Datenbank ohne Decke	12
3.4	Datenbank mit Decke	12
5.1	Relevante Bewegungen ohne Decke	33
5.2	Relevante Bewegungen mit Decke	33
5.3	Nicht relevante Bewegungen ohne Decke	34
5.4	Nicht relevante Bewegungen mit Decke	34
6.1	Resultate ohne Decke	37
6.2	Resultate mit Decke	38

11. Literaturverzeichnis

- [1] Intelligentes bett auf dem weg zum markt, 2011. Empa Medienmitteilung, http://www.empa.ch/plugin/template/empa/*/109455, Website besucht am 13.12.2011.
- [2] <http://www.wired.com/gadgetlab/2010/06/kinect-sensor-bar-on-microsoft-store-for-150/>, Website besucht am 22.12.2011.
- [3] Microsoft Research. *Programming Guide*, 2011. Kinect for Windows SDK beta.
- [4] Mikkel Viager. Analysis of kinect for mobile robots. Technical report, Technical University of Denmark (DTU), 2011.
- [5] http://de.wikipedia.org/wiki/Opening_%28Bildverarbeitung%29, Website besucht am 22.12.2011.
- [6] Richard E. Gonzalez, Rafael C. / Woods. *Digital Image Processsing*, chapter 11.4. Pearson Education, Inc., 3. edition, 2008.

12. Erklärung zur Urheberschaft

Erklärung

Wir erklären hiermit an Eides statt, dass wir die vorliegende Arbeit ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt haben; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Ort, Datum

Rapperswil, 23.12.2011

Unterschriften

Lukas Haldemann

Michael Jost

A Aufgabenstellung

Thema: Schlafüberwachung mittels Kinect

Studenten: Lukas Haldemann, Michael Jost
Betreuer: Guido Schuster
Partner: Compliant Concept

Kurzbeschreibung

Compliant Concept hat ein intelligentes Bett entwickelt, welches mittels Drucksensoren die Bewegungen im Schlaf überwacht.

Ziel dieser Arbeit ist es nun, eine solche Schlafüberwachung mittels einer Kinect Kamera zu realisieren. Der Vorteil der Kinect Kamera ist die einfache Nachrüstbarkeit und die 3D Informationen. Ausserdem funktioniert diese Kamera auch in der Dunkelheit und ist somit ideal für eine solche Überwachung geeignet. Die Aufgabe ist somit, den Videostrom in Echtzeit auf relevante Bewegungen zu untersuchen und diese als eine Rate (Bewegungen pro Zeit, Zeit seit letzter Bewegung, etc) darzustellen.

Aufgabenstellung

- Einarbeitung in die existierende Hard- und Software der Kinect Kamera
- Erstellen einer repräsentativen Datenbank für die Algorithmenentwicklung
- Analyse der 3D Daten und Entwicklung eines geeigneten Verfahrens
- Realisierung des entwickelten Verfahrens in Echtzeit und Visualisierung der Resultate

Erwartete Ergebnisse

- Dokumentation der Theorie, der Lösungsansätze, der Lösung und der entwickelten Software
- Eine dokumentierte und annotierte Datenbank
- Eine Echtzeit Software (mit GUI) zur Auswertung und Visualisierung der 3D Daten
- Ein persönliches Laborbuch

Arbeitsweise

- Sie führen ein persönliches Laborbuch, wo Sie aufschreiben wann Sie was für wie lange machen und was die Ergebnisse sind
- Sie schicken vor jeder Sitzung eine Zusammenfassung, welche dokumentiert, was Sie in der letzten Woche gemacht haben.